<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row 1</td>
<td>Row 2</td>
<td>Row 3</td>
<td>Row 4</td>
<td>Row 5</td>
</tr>
</tbody>
</table>

(Additional rows and columns follow)
GRAVITY DATA PROCESSING PROGRAMS

Woods Hole Oceanographic Institution
Woods Hole, Massachusetts

February 1977
NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.
GRAVITY DATA PROCESSING PROGRAMS

By

CARL BOWIN

WOODS HOLE OCEANOGRAPHIC INSTITUTION
Woods Hole, Massachusetts 02543

January 1977

TECHNICAL REPORT

Prepared for the Office of Naval Research under Contract N00014-74-C0262 NR 083-004.

Reproduction in whole or in part is permitted for any purpose of the United States Government. In citing this report in a bibliography, the reference should be followed by the phrase: UNPUBLISHED MANUSCRIPT.

Approved for public release; distribution unlimited.

Approved for Distribution: John I. Ewing, Chairman
Department of Geology & Geophysics
ABSTRACT

A summary and documentation of a family of computer programs that have been developed by the gravity group at the Woods Hole Oceanographic Institution is presented. The programs provide for format conversion, computation of the regional gravity field from spherical harmonic coefficients, selective data retrieval, graphic display, and construction of two- and three-dimensional structure models and the computation of the gravitational attraction of those models.
INTRODUCTION

This report is a summary and documentation of a family of computer programs that have been developed by the gravity group at the Woods Hole Oceanographic Institution. The programs documented here provide for format conversion, computation of the regional gravity field from spherical harmonic coefficients, selective data retrieval, graphic display, and construction of two- and three-dimensional structure models and the computation of the gravitational attraction of those models.

Many of the programs in this report have been used and modified for more than ten years. During this time six substantially different computer systems have been available to us. These are an Autonetics Recomp II, General Electric 225, IBM 7090 and 7094, IBM 1710 (shipboard), XDS Sigma-7, and Hewlett-Packard 2114, 2116, and 2100 (shipboard). Thus the programs have evolved not only because of changing needs and experience, but also because of different system hardware and software constraints. Artifacts reflecting this evolution are evident in some of the programs.
In the interpretation of gravity data and the creation of structure models of earth features, other geophysical and geological information is important. Gravity potential information alone does not define a unique mass distribution, and hence additional information is required to limit the possibilities. Therefore, we have added the capability for retrieving and displaying other types of data which are available in digital form. At the present time these data types include seismicity, seismic refraction profiles, and location of active volcanoes. The seismic refraction profiles are derived from a compilation of crustal seismic refraction profiles prepared by McConnel Jr. and McTaggart-Cowan of the University of Toronto in 1963 and from five supplements (No. 1 by Gupta and McTaggart-Cowan, 1964; No. 2 by Gertner, 1967; No. 3 by Gertner and Farquhar, 1968; No. 4 by Gertner and Farquhar, 1971; and No. 5 by Gertner and Farquhar, 1972). Supplements numbers 2 through 5 were sponsored by the Federation of Astronomical and Geophysical Services of I.C.S.U. Subsequent to the fifth supplement, financial assistance to the University of Toronto by U.N.E.S.C.O. for this compilation terminated, and unfortunately, this compilation effort has ceased. For our utilization of the seismic refraction compilation,
we find a single record per refraction line a more convenient format than the one- or two-record format prepared by the University of Toronto. We prefer magnetic tape or disc for data storage and accordingly are not limited by the 80 character record length of punched cards. We have incorporated additional data as we have had particular needs.

The file of locations of active volcanoes was originally coded from the Catalogue of Active Volcanoes. Volcanoes on New Zealand were added from Thompson (1964), and those in Alaska and the Aleutian Arc were added from Foster et al. (1966) and Coats (1950). More recently, IAVCEI has prepared data sheets of the post-Miocene volcanoes of the world (IAVCEI, 1975). A deck of cards based on these data sheets was obtained in December 1975 from NGSDC.

Additional data types can be incorporated into our programs relatively easily. Location and certain other characteristics of Deep Sea Drilling Program (DSDP) drill hole sites is a file of interest. We hope that a source for a global compilation of seismic slip mechanisms might be found.

PROGRAMS

A diagram outlining the functions served by the programs documented in this report is given as Figure 1. This diagram
serves as an index to the utilization of the family of programs, and it is intended to be largely self-explanatory. Table 1 lists the programs documented here and provides a summary statement of the purpose of each program. Table 1, together with figure 1, enables the reader to quickly find programs to meet his need.

Documentation for the programs themselves follows the references cited section. The programs are ordered alphabetically, and for each program the characteristics and operational parameters are described first, followed by a section containing listings of the source coding. Subroutines required by these programs are then given alphabetically in the section after the program listings. Standard system routines and those of a normal Fortran subroutine library are not reproduced. Normally, only a source listing is given for each subroutine.
ACKNOWLEDGMENTS

Documentation of programs is a tedious activity, which normally seems to be deferred, awaiting a less busy time. The less busy time is an elusive quantity that never seems to arrive, and in the interim more programs are written and old programs modified to meet new requirements. After awhile, the original programmer or the modifier often has moved on to other pastures and is no longer available to help with documentation. Although I firmly believe in the importance of documenting programs when they are written, I personally have not been very faithful to that belief. Thus the existence of this report is largely due to the efforts of others. I particularly want to thank Allin Folinsbee and Leon Gove for being far more rigorous than I in documenting their programming efforts, Julie Milligan for a major contribution in the early stages of preparation of this report, Carolyn Dean for her efforts in the later stages, and Nan Galbraith, Leon Gove and Christine Wooding for their help in its completion. Allin is now at the Bedford Institute of Oceanography, Halifax, Nova Scotia; Julie is at the University of Auckland in Auckland, New Zealand and Carolyn is teaching high school in Falmouth, Massachusetts.

The preparation of this report was supported by the Office of Naval Research under contract N00014-74-C-0262; NR083-004.
<table>
<thead>
<tr>
<th>PROGRAM DICTIONARY</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ABSTGC</strong></td>
</tr>
<tr>
<td><strong>CHART</strong></td>
</tr>
<tr>
<td><strong>CHARTG</strong></td>
</tr>
<tr>
<td><strong>CONV67</strong></td>
</tr>
<tr>
<td><strong>CR2G</strong></td>
</tr>
<tr>
<td><strong>CRWT3</strong></td>
</tr>
<tr>
<td><strong>DMABLK</strong></td>
</tr>
<tr>
<td><strong>DMAP</strong></td>
</tr>
<tr>
<td><strong>DMOD</strong></td>
</tr>
<tr>
<td><strong>GFLD1</strong></td>
</tr>
<tr>
<td><strong>GFLD2</strong></td>
</tr>
<tr>
<td><strong>GRAFG2</strong></td>
</tr>
<tr>
<td><strong>GRAV1</strong></td>
</tr>
<tr>
<td>Program</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>GSTOG</td>
</tr>
<tr>
<td>G3DCP</td>
</tr>
<tr>
<td>G3DCPRES</td>
</tr>
<tr>
<td>HIG</td>
</tr>
<tr>
<td>LSORT</td>
</tr>
<tr>
<td>MODPLOT</td>
</tr>
<tr>
<td>NOAA</td>
</tr>
<tr>
<td>PROFG</td>
</tr>
<tr>
<td>PROJ4</td>
</tr>
<tr>
<td>RETRIEVE</td>
</tr>
<tr>
<td>SAINT2</td>
</tr>
<tr>
<td>SELSP</td>
</tr>
<tr>
<td>SPFMT</td>
</tr>
<tr>
<td>TALPLOT16</td>
</tr>
<tr>
<td>SUBROUTINES REQUIRED BY THE PRECEDING PROGRAMS:</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>ALDTGETX</td>
</tr>
<tr>
<td>ANOV2GETY</td>
</tr>
<tr>
<td>ANOV3GINOT</td>
</tr>
<tr>
<td>AREAKGINTF</td>
</tr>
<tr>
<td>CALSCGI67F</td>
</tr>
<tr>
<td>CDATEGRIDG</td>
</tr>
<tr>
<td>CHGMTGRID2</td>
</tr>
<tr>
<td>COORRINCEP</td>
</tr>
<tr>
<td>DISAZISW(1)</td>
</tr>
<tr>
<td>DMTORM2DY</td>
</tr>
<tr>
<td>DNAVNAVIN</td>
</tr>
<tr>
<td>DRECNAVOT</td>
</tr>
<tr>
<td>DY2MOBG</td>
</tr>
<tr>
<td>ENDLTBLINE</td>
</tr>
<tr>
<td>EVILPINOT</td>
</tr>
<tr>
<td>EXDTPLANET</td>
</tr>
<tr>
<td>FINDPLOTA</td>
</tr>
<tr>
<td>FLD2RETBY</td>
</tr>
<tr>
<td>GETCRTDM2</td>
</tr>
<tr>
<td>GETFRTODM</td>
</tr>
<tr>
<td>GETGSIMUL</td>
</tr>
<tr>
<td>GETGASPLOT</td>
</tr>
<tr>
<td>GETGCSPOOT</td>
</tr>
<tr>
<td>GETGS2SPOT2</td>
</tr>
<tr>
<td>GETHTIDAL</td>
</tr>
<tr>
<td>GETLVEIBY</td>
</tr>
<tr>
<td>GETMWEIG2</td>
</tr>
<tr>
<td>GETPYBLIKI</td>
</tr>
<tr>
<td>GETSYBLIKO</td>
</tr>
<tr>
<td>GETSYNOT</td>
</tr>
<tr>
<td>GETTV</td>
</tr>
<tr>
<td>DATA TYPE</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>SOURCE</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Conversion Programs</td>
</tr>
<tr>
<td>Master Data File</td>
</tr>
<tr>
<td>Selective data retrieval, Graphic, display, Three-dimensional structure model Programs</td>
</tr>
<tr>
<td>Data Projection</td>
</tr>
</tbody>
</table>

**Figure 1**

Diagram showing data processing flow:
- Data input (DATA TYPE) flows to GRAVITY, which includes WHOI Ship's SEAG1 fmt, WHOI land data fmt, and other land and marine data in various formats.
- GRAVITY data is processed through Regional Gravity field Spherical Harmonic Coefficients and other programs.
- The output flows to SEISMICITY, SEISMIC REFRACTION PROFILES, and ACTIVE VOLCANOES.

Diagram shows:
- Data processing through conversion programs like GSTOG, GSTOG67, CONV67.
- Master Data File includes Gravity Data Library (EDL) GSUM format and selective data retrieval.
- Data projection through PROJ4, SAINT2, MODPLOT, DMOD, TALPLOT16, and MODPLOT.
REFERENCES CITED


Refraction Profiles - A Compilation (Supplement No. 1)
University of Toronto 13, 1964.

I.A.V.C.E.I. Working Group, Data sheets of the Post Miocene

International Association of Volcanology (Ed.), Catalogue of the
Active Volcanoes of the World Including Solfatara Fields,

McConnel, R. K., Jr., and G. H. McTaggart-Cowan, Crustal Seismic
Refraction Profiles: A Compilation, University of Toronto,

and Effective Utilization, Technical Rept. W.H.O.I. 74-33,

Thompson, B.N., Quaternary Volcanism of the Central Volcanic
Region, New Zealand Jour. Geol. and Geophys., Vol. 7,
NAME: ABSTGC
TYPE: Main Program
PURPOSE: To abstract GCON data at a spacing of 10 nautical miles
MACHINE: XDS Sigma 7
PROGRAM CATEGORY: Statistical

DESCRIPTION:
The program processes one degree of GCON data at a time. The data is decoded and if the data falls within the degree square of concern the free air anomaly and height is added to the appropriate 10 nautical mile square value and the position is checked to see if this point is the closest to the center of the square. If it is the closest, the values and the position are retained. When all the data for the degree square is processed the GABS data record is written to the output device.

INPUT:
PARAMETER CARDS (via F:105)
1) Sense switch card
   SSW(46) = 0 no effect
   SSW(46) = 2 process within bounds and use D.L.T.
2) Geographic Bounds (in degrees)
   Top (KDTOP, I5)
   Bottom (KDBOT, I5)
   Left (KDLFT, I5)
   Right (KDRGT, I5)
3) D.L.T. deck if applicable

GCON DATA (via F:1)
Data in GCON format blocked 22 x 50.

OUTPUT:
GABS DATA (via F:2)
Data in GABS format. Data in each physical record is all the data for one degree square. The first logical record of each physical record is the whole-part of the latitude and longitude. Following are 36 logical records, one each for the 10 nautical mile square (see figures 1 and 2).
ABSTGC (continued) page 2

USAGE:

SAMPLE RUN

!JOB
!LIMIT (9T,2)(CORE,20),(TIME,XY)
!MESSAGE I/P tape info
!MESSAGE O/P tape info
!ASSIGN F:1(DEVICE,9T),(SN,XXXX),(IN),(TRIES,10)
!ASSIGN F:2(DEVICE,9T),(SN,YYYY),(OUT),(TRIES,10)
!LOAD (BI),(UNSAT,(312),(3))
    ABSTGC object deck

!RUN
!DATA

    Parameter cards

!EOD

RESTRICTIONS:

If bounds are to be checked a D.L.T. deck must be provided

STORAGE: 16K words

SUBPROGRAMS REQUIRED: ISW, FORTRANIV Library

TIMING: Thru-put time is about 3000 logical records/minute

PROGRAMMER: Lee Gove

ORIGINATOR: Carl Bowin

DATE: 15 October 1975
NAME: CHART
TYPE: Program
PURPOSE: Plot data on Mercator charts
SOURCE LANGUAGE: Sigma-7 Fortran 4
MACHINE: Sigma-7
PROGRAM CATEGORY: Graphical Display
DESCRIPTION:
Plots Mercator chart at specified scale, draws track and annotates with specified parameter.

INPUT:
Input formats: FIXSE, SEAG1, GSUM, MBATR, CALCM, and tabulations of refraction, earthquake, volcano, heat flow data. There is also a user specified format.
Program plots a 1/2-inch fiducial square in lower right corner of chart. If sides of square are offset means pen hit stops or lost registration in course of plotting.

1st card
(20A4) Label- up to 80 characters, plotted vertically on left-hand margin of chart.

2nd card Sense switch options
ISW(0) - ISW(79) (8021) Put Sense switch (0) option in column 80, all others in column corresponding to switch number.

Optional card
If ISW(10) = 1 on card 2, put four-character name of input tape here, format (A4). Using this option (subroutine MOUNT) it is possible to generate a plot tape with several plots per job separated by EOF, from one or more input tapes. This card is never used when input data is in GSUM format. For MOUNT cards for GSUM format, see card(s) seven below.

3rd card (2(312,14,5X), 3125)
Column
1,2 ISTDA Start date for processing, for example
3,4 ISTMO 0204720341 means 2 February 1972 0341Z
5,6 ISTYR If blank, plotting begins with first
7,10 ISTHM record.
### 3rd card continued

<table>
<thead>
<tr>
<th>Column</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>16,17</td>
<td>IENDA</td>
</tr>
<tr>
<td>18,19</td>
<td>IENMO</td>
</tr>
<tr>
<td>20,21</td>
<td>IENYR</td>
</tr>
<tr>
<td>22-25</td>
<td>TENHM</td>
</tr>
<tr>
<td>31-35</td>
<td>ISKP</td>
</tr>
<tr>
<td>36-40</td>
<td>ISFIL</td>
</tr>
<tr>
<td>41-45</td>
<td>IBCKUP</td>
</tr>
</tbody>
</table>

### 4th card (F10.0,215,1X,A4,415)

<table>
<thead>
<tr>
<th>Column</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>SINCH</td>
</tr>
<tr>
<td>15</td>
<td>ITRK</td>
</tr>
<tr>
<td>20</td>
<td>LCNT</td>
</tr>
<tr>
<td>25</td>
<td>NDEG</td>
</tr>
<tr>
<td>27-30</td>
<td>NUMPL</td>
</tr>
<tr>
<td>35</td>
<td>NPTA</td>
</tr>
<tr>
<td>39,40</td>
<td>JFMT</td>
</tr>
<tr>
<td>44,45</td>
<td>NX</td>
</tr>
<tr>
<td>50</td>
<td>NFILE</td>
</tr>
</tbody>
</table>

- 0 will still plot first file.
JFMT = 1, FIXSE Format - navigation

NX = 1 for time, and date at change of date
    = 2 for month
    = 3 for year
    = 4 for day
    = 5 for zone

JFMT = 2, SEAG1 Format - gravity

NX = 1 for time, and date at change of day
    = 2 for water depth in corrected meters
    = 3 for free air anomaly
    = 4 for Bouguer anomaly
    = 5 for speed in knots
    = 6 for heading in degrees
    = 7 for Eotvos correction
    = 8 for Matthews Table number
    = 9 for low order 3 digits of total magnetic field intensity
    = 10 for total regional magnetic field (not implemented)
    = 11 for residual magnetic value (not implemented)
    = 12 for negative speed
    = 13 for negative water depth
    = 14 for negative Eotvos correction
    = 15 for negative Free Air anomaly
    = 16 for negative heading
    = 17 for total magnetic field intensity
    = 18 for uncorrected depth in meters
    = 19 for uncorrected depth in fathoms

JFMT = 3, GSUM Format - gravity summary

NX = 1 for time
    = 2 for source code
    = 3 for elevation
    = 4 for depth
    = 5 for height
    = 6 for Free Air anomaly
    = 7 for Bouguer anomaly
    = 8 for terrain correction
    = 9 for complete Bouguer anomaly
    = 10 for regional Free Air anomaly
    = 11 for observed gravity

JFMT = 4, MBATR Format - bathymetry

NX = 1 for time
    = 2 for corrected depths in fathoms
    = 3 for corrected depths in meters
    = 4 for cumulative distance in kms
    = 5 for heading
    = 6 for speed in knots
    = 7 for uncorrected depths in fathoms
    = 8 for uncorrected depths in meters
CHART Continued, page 4 16
13 November 1972
CHART Page 4

4th card continued

JFMT = 5, CALCM Format - magnetic field
NX = 1 for time
= 2 for calculated regional field
= 3 for anomalous field
= 4 for cumulative distance in kms
= 5 for heading
= 6 for speed
= 7 for observed magnetic field

JFMT = 6, STATN Format - stations
Not implemented

JFMT = 7, SPFMT Format (Bowin format for Univ. of Toronto compilation of seismic refraction data)
NX = 1 for station number
= 2 for height
= 3 for mantle velocity
= 4 for depth to mantle
= 5 for crustal thickness
= 6 for average crustal velocity (CRVW) (Nafe-Drake)
= 7 for column weight (WETN) using Nafe and Drake velocity/density relation
= 8 for column weight (AVWTW) using average crustal density
= 9 for CRVW same as 6 to 8 but using Woollard's
= 10 for WGTW velocity/density relation
= 11 for AVWTW

JFMT = 8, World Seismicity Format
If SSW(16) = 1, then NX value is ignored and ANOV3 plots a spot whose type and size depends upon depth and magnitude of earthquake
If SSW(16) = 0
NX = 1 for date (month, day, year)
= 2 for depth in kilometers
= 3 for magnitude

JFMT = 9, Active Volcanoes
NX = 1 for region code number (from IVA Catalog of Active Volcanoes of the World)
= 2 for height in meters
= 3 for volume and page ((IPT*1000)+IPAGE)
4th card (Contd.)

**JFMT = 10, Heat Flow**

(For key to items 1, 6, and 7, see Simmons and Horai, Journ. Geophys. Res., Vol. 73, p. 6608-6629, 1968)

NX = 1 for catalog sequence number  
= 2 for depth  
= 3 for heat flow  
= 4 for gradient  
= 5 for conductivity  
= 6 for classification code for station  
= 7 for reference number  
= 8 for year

**JFMT = 11, Lunar Data**  
NX = No options implemented yet

**JFMT = 12, User supplied format. Dummy**

Subroutine GETX is in library. User supplies his own as a binary or source deck with job. The following conventions must be followed:

*If NX = 0 no annotations will take place.*  
*If NX = 1 program will annotate with time.*
5th card Format (4I5)

Column
5       KPT = 1 chart magnification factor (usually 1)
10      KHT = Annotation character size in integer multiples
         of 0.07 inch (usually 1)
15      ICTYP = 0 for non-integer degree chart boundaries
         = 1 for integer degree chart boundaries
19,20   IDEC = variable for decimal point in annotation of
         plotted points
         = N, for N DIGITS to right of decimal point
         = 0 for decimal point only
         = -1 for suppressing decimal point

6th card Format (4I5) Values are negative for west and south
If ICTYP = 1 (integer degree boundaries)
Column
  1-5    ITOP = Top boundary of chart
  6-10   IBOT = Bottom boundary of chart
  11-15  ILEFT = Left boundary of chart
  16-20  IRIGT = Right boundary of chart
If ICTYP = 0 (non-integer degree), then enter CHART boundaries
on 4 cards in degrees and minutes Format (I5,F10.5)
Be sure sign of the minutes agrees with the sign of
degrees (e.g., -33-30.0)

DATA:

Data in specified format are loaded in device having unit
reference number 1.

OUTPUT:

Printer:
Listing of inputs
List of dates outside of chart boundaries ("OOB") if SSW(9) is up
List of dates of all data read if SSW(12) is up
Plotter:
Mercator Charts

USAGE: See operating instructions at the end of this section

RESTRICTIONS: None

STORAGE REQUIREMENTS: 18,432 locations

*7th card format (A4) for GSUM formatted data only.
Column 1-4 Input tape serial number, one per card, as many cards
as input tapes. Last tape serial number card must have
EITP in columns 1-4, to signal end of input tape serial
numbers (calls subroutine MOUNT).
SUBROUTINES REQUIRED: Stored in library accounts 305, 312 and 3
GRID2, OLINE, WHR, ANOV2, RETBY, VETBY, GETC, GETF, GETG, GETM, GETS, GETST, GETH, GETP, GETV, GETY, GETL, MOUNT, STAT, ISW, TODAY, POSTAP, SPOT2, ANOV3, FIND, CALCOMP routines.

OPERATIONAL ENVIRONMENT:

Data input device - Unit reference number = 1
12" or 30" Calcomp Plotter

OPERATIONAL CHARACTERISTICS:

Sense Switch Options: Set to zero to decline option

<table>
<thead>
<tr>
<th>SSW(n)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSW(0)</td>
<td>= 1 to draw only the grid</td>
</tr>
<tr>
<td>SSW(1)</td>
<td>= 1 to delete drawing NDEG lines (in GRID2)</td>
</tr>
<tr>
<td>SSW(2)</td>
<td>= 1 if next plot will be on the same grid as this plot, sets pen back at origin</td>
</tr>
<tr>
<td>SSW(3)</td>
<td>= 1 to only annotate date at change of day</td>
</tr>
<tr>
<td>SSW(4)</td>
<td>= 0 for no mark at data point</td>
</tr>
<tr>
<td></td>
<td>= 1 for plotting a circle around data point</td>
</tr>
<tr>
<td></td>
<td>= 2 for plotting a dot at data point</td>
</tr>
<tr>
<td>SSW(5)</td>
<td>= 0 to make degree annotations inside grid (character size 0.07&quot;)</td>
</tr>
<tr>
<td></td>
<td>= 1 to make degree annotations outside grid (character size 0.21&quot;)</td>
</tr>
<tr>
<td></td>
<td>= 2 to make degree annotations outside grid (character size 0.35&quot;)</td>
</tr>
<tr>
<td>SSW(6)</td>
<td>For multiplot runs, = 1 will put on EOF between plots. Useful to PDP-5 operator for restarting in the event of mechanical malfunction of pen</td>
</tr>
<tr>
<td>SSW(7)</td>
<td>= 0 to annotate on right side of track</td>
</tr>
<tr>
<td></td>
<td>= 1 to annotate on left side of track</td>
</tr>
<tr>
<td>SSW(8)</td>
<td>= 1 to suppress plotting of grid</td>
</tr>
<tr>
<td>SSW(9)</td>
<td>= 1 to list points out of bounds on line printer</td>
</tr>
<tr>
<td>SSW(10)</td>
<td>= 1 to call subroutine MOUNT which reads serial number of input tape; not used for GSUM formatted data.</td>
</tr>
<tr>
<td>SSW(11)</td>
<td>= 1 to annotate data points alternately on left and right side of track</td>
</tr>
<tr>
<td>SSW(12)</td>
<td>= 1 to list date of data just read for identification</td>
</tr>
<tr>
<td>SSW(13)</td>
<td>= 1 if two or more plots are to be made from the same file and this is not the last plot. Backs tape up to beginning of file and reinitializes program.</td>
</tr>
</tbody>
</table>
OPERATIONAL CHARACTERISTICS (Contd.)

SSW(16) = 1 to plot a spot for seismicity data whose type and size depends upon the depth and magnitude of the earthquake (ANOV4)

SSW(17) = 0 (seismicity) plots an x for pre-1961 data. Depth and magnitude data pre-1961 are limited. For these points, ANOV4 normally uses a symbol which does not vary in size

= 1 ANOV4 will try to plot varying sized symbols for all data, including pre-1961
= 9 will not plot pre-1961 data at all
This sense switch is used only if SSW(16)=1

SSW(18) = 0 to make annotation at right angles to incremental track (subroutine ANOV2)
= 1 to make annotations horizontally
= 2 to make annotations vertically
= 3 to invert annotations for headings 180 to 269
= 4 to annotate either horizontally or vertically depending on direction of track

SSW(19) = 0 for earth meridional parts from Bowditch
= 1 for meridional parts for spherical planet

SSW(20) = N, (seismicity) for additional size increment in plotting symbols for all data points (ANOV4). (Only if SSW(16)=1)

SSW(21) = N, (seismicity) for size factor by which plotting symbols will vary according to magnitude. If N = 0, then ANOV4 sets N = 2. (Only if SSW(16) = 1)

SSW(27) = 1 for GSUM data to suppress rewind input tape at start

SSW(30) = 1 to read GSUM from 2 cards

SSW(32) = 1 to read SPFMT from 2 cards

SSW(40) = 1 to process GSUM with BOUNDS using DLT

SSW(42) = 1 to read SEISMICITY data in blocked format

SSW(60) = 1 to process GSUM data only with IFFC = 4

SSW(61) = 1 to replace GSUM values with averaged values for FA, BG, ELEV, LAT, LONG

SSW(71) = 1 to annotate every two hours on the hour only
Program Flow:

Tape advances to start date. Program initialization choices are made, plotter draws and annotates Mercator grid, and then data in appropriate format are read and plotted one record at a time if within chart boundaries. If more than one plot is being made the program can be restarted using SSW(13), or by using SSW(6) and by putting a RUN and DATA card and continue with a new set of data cards.

TIMING:

Two to twenty minutes depending upon size of chart, number of intermediate degree lines plotted, and amount of data plotted and annotated.

ERROR MESSAGE DIAGNOSTIC:

<table>
<thead>
<tr>
<th>Message</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOB day, month, year, time</td>
<td>Date point is out of chart boundaries, and SSW(9) is up</td>
<td>Record is skipped, program continues</td>
</tr>
<tr>
<td>EOF day, month, year, time</td>
<td>End of file found on magnetic tape</td>
<td>Job ends or continues to next plot if any</td>
</tr>
<tr>
<td>PARITY ER day, month, year, time</td>
<td>Parity error found</td>
<td>Record is skipped, program continues</td>
</tr>
<tr>
<td>FMT ER day, month, year, time</td>
<td>Unidentified error found</td>
<td>Record is skipped, program continues</td>
</tr>
</tbody>
</table>

PROGRAMMER: Carl Bowin and Hartley Hoskins

ORIGINATOR: Carl Bowin

DATE: Version of 19 October 1972

SEISMICITY CHARTS WITH VARYING SYMBOLS

The type of symbol is determined by depth; size varies with magnitude.
(Subroutine ANOV4, version 15 Apr. 1975)

<table>
<thead>
<tr>
<th>Depth</th>
<th>$M_b$ MAGNITUDE</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Symbol" /> - less than 70 km</td>
<td>$\bigcirc$ - less than 4.5</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /> - 70 to 150 km</td>
<td>$\bigcirc$ - 4.5 to 5.5</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /> - 150 to 300 km</td>
<td>$\bigcirc$ - 5.5 to 6.5</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /> - 300 to 500 km</td>
<td><img src="image" alt="Symbol" /> - greater than 6.5</td>
</tr>
<tr>
<td><img src="image" alt="Symbol" /> - greater than 500 km</td>
<td><img src="image" alt="Symbol" /> - pre 1961 data</td>
</tr>
</tbody>
</table>
NAME: CHARTG

TYPE: Program

PURPOSE: Plot GSUM, GCON, GABS on Mercator charts

SOURCE LANGUAGE: XEROX EXTENDED FORTRAN IV

MACHINE: Sigma-7

PROGRAM CATEGORY: Graphical Display

DESCRIPTION: Plots Mercator chart at specified scale, plots and annotates with specified parameter value.

INPUT:

Input formats: GSUM, GCON, GABS

Plotter registration: Program plots a 1/2 inch fiducial square in lower right corner of chart. If sides are offset, there has been a loss of registration.

Parameter Cards

Card 1 Label - up to 80 characters written vertically on left margin of chart (FORMAT(20A4))

Card 2 Sense Switch Options - (FORMAT(80I1))

put option in card column corresponding to sense switch (SSW(ϕ) in column 80)

SSW(ϕ) = 1 to only draw grid (no input data read)
SSW(1) = 1 to delete all intermediate (NDEG) grid lines
SSW(2) = 1 next plot will be on the same grid
SSW(3) = 1 to annotate only at change of day
SSW(4) = 1 to plot a circle around data point
= 2 to plot a dot at the data point
IN\UT (continued):

SSW(5) = \( \phi \) degree annotation inside grid  
(character size = 0.07")  
= 1 degree annotation outside grid  
(character size = 0.21")  
= 2 degree annotation outside grid  
(character size = 0.35")
SSW(6) = 1 puts EOF between plots
SSW(7) = \( \phi \) annotate on left side of track  
= 1 annotate on right side of track
SSW(8) = 1 to suppress plotting of grid
SSW(9) = 1 list date and time of data out of bounds
SSW(10) = 1 to call mount to read serial  
number of input tape
SSW(11) = 1 to annotate data points alternately  
on left and right side of track
SSW(12) = 1 to list date of data just read
SSW(18) = \( \phi \) annotations at right angles to track  
= 1 annotate horizontally  
= 2 annotate vertically  
= 3 invert annotations for headings  
between 180 to 269  
= 5 to do no annotation
SSW(19) = \( \phi \) for earth meridional points from  
Bowditch  
= 1 for meridional points for spherical  
planet
SSW(25) = 1 to call MOUNT for input tape  
serial number
SSW(30) = 1 to read GSUM from punched-cards
SSW(40) = \( \phi \) process unblocked GSUM (no DLT)  
= 1 process blocked GSUM (no DLT)  
= 2 process blocked GSUM (with DLT)
SSW(46) = \( \phi \) process with bounds  
= 1 to make no check on bounds
SSW(71) = 1 to annotate every two hours on the  
hour

Card 3 START/END Dates (Format(2(3I2,414,1x),3I5))

Column
1,2  Start Day  (ISTDA)
3,4  Start Month  (ISTMO)
5,6  Start Year  (ISTyr)
7,10  Start Time  (ISTHM)
16,17  End Day  (IENDA)
18,19  End Month  (IENMO)
20,21  End Year  (IENYR)
22,25  End Time  (IENHM)
INPUT (continued):

Card 4  
(Format(F10.0,3I5,1X,A4,5I5))

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>Inches/one degree of longitude</td>
<td>SINCH</td>
</tr>
<tr>
<td>15</td>
<td>= 1 to draw track</td>
<td>ITRK</td>
</tr>
<tr>
<td>20</td>
<td>= N to plot every nth point</td>
<td>LCNT</td>
</tr>
<tr>
<td>25</td>
<td>= N to draw every nth grid line</td>
<td>NDEG</td>
</tr>
<tr>
<td>27-30</td>
<td>= number of plot</td>
<td>NUMPL</td>
</tr>
<tr>
<td>35</td>
<td>= N to annotate every nth plotted point</td>
<td>NPTA</td>
</tr>
<tr>
<td>39,40</td>
<td>= 3 to use GSUM</td>
<td>JFMT</td>
</tr>
<tr>
<td>44,45</td>
<td>= N to annotate with nth variable (see following table)</td>
<td>NX</td>
</tr>
<tr>
<td>50</td>
<td>= N to output N files on one grid</td>
<td>NFILE</td>
</tr>
</tbody>
</table>

Table for Selecting NX

<table>
<thead>
<tr>
<th>NX</th>
<th>JFMT 3 (GSUM)</th>
<th>13 (GCON)</th>
<th>14 (GABS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>omits annotation</td>
<td>omits annotation</td>
<td>omits annotation</td>
</tr>
<tr>
<td>1</td>
<td>time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>source code</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>elevation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>depth</td>
<td>depth</td>
<td>average free air</td>
</tr>
<tr>
<td>5</td>
<td>height</td>
<td>height</td>
<td>average elevation</td>
</tr>
<tr>
<td>6</td>
<td>free air anomaly</td>
<td>free air anomaly</td>
<td>central free air</td>
</tr>
<tr>
<td>7</td>
<td>Bouguer anomaly</td>
<td>Bouguer anomaly</td>
<td>central elevation</td>
</tr>
<tr>
<td>8</td>
<td>terrain corrections</td>
<td>abstracted free air</td>
<td>number of points</td>
</tr>
<tr>
<td>9</td>
<td>complete Bouguer</td>
<td>abstracted height</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>regional free air</td>
<td>average free air</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>observed gravity</td>
<td>average free air</td>
<td></td>
</tr>
</tbody>
</table>

Card 5  
(Format(4I5))

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>chart magnification factor (KPT)</td>
</tr>
<tr>
<td>10</td>
<td>annotation character size (KHT)</td>
</tr>
<tr>
<td></td>
<td>in integer multiples of 0.07 inch</td>
</tr>
<tr>
<td>15</td>
<td>= 0 for non-integer chart boundaries (ICTYP)</td>
</tr>
<tr>
<td>19,20</td>
<td>= N for N digits to right of decimal point</td>
</tr>
<tr>
<td></td>
<td>= 0 for decimal point only</td>
</tr>
<tr>
<td></td>
<td>= -1 for suppressing decimal point</td>
</tr>
</tbody>
</table>
INPUT (continued):

Card 6 (Format(4I5))

If ICTYP = 1 integer degree boundaries (Format(4I5))

Column

1-5       Top boundary
6-10      Bottom boundary
11-15     Left boundary
16-20     Right boundary

If ICTYP = 0 non-integer boundaries (Format(I5,F10,5))

Enter one card each for top, bottom, left, right in degrees and minutes.

VALUES ARE NEGATIVE FOR WEST AND SOUTH

Remaining Cards

The remaining cards depend on if DLT's are used and if mount is called.

If neither are used, there are no more cards.

If only mount is called, then there is a card for each input tape of the form.

Column

1-4       mag tape serial number (ITAPE)

And after all tape serial numbers there is a card with ITP from an ID. This signifies end of input tapes.

If D.L.T.'s are used, the D.L.T. deck is inserted immediately after the MOUNT serial number card for the appropriate tape.

OUTPUT:

Printer: Listing of input parameters

Plotter: Mercator charts
USAGE: See operating instructions

RESTRICTIONS: None

STORAGE REQUIREMENTS: 134910 locations

SUBROUTINES REQUIRED:
Stored in library accounts 456, 305, 312, and 3
GRID2, OLINE, WHR, ANOV2, RETBY, VETBY, GETGS, MOUNT, STAT, ISW, TODAY, POSTAP, SPOT2, ANOV3, FIND, CALCOMP routines

OPERATIONAL ENVIRONMENT:
9-track tape drive, card reader, line printer, plotter

OPERATIONAL CHARACTERISTICS:
Program Flow:
Tape advances to start date. Program initialization choices are made, plotter draws and annotates Mercator grid, and then data in appropriate format are read and plotted one record at a time if within chart boundaries.

TIMING: About 1000 pts plotted per minute if the DLT is in use.

ERROR MESSAGE DIAGNOSTIC:

<table>
<thead>
<tr>
<th>Message</th>
<th>Cause</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOB</td>
<td>day, month, year, time</td>
<td>Data point is out of chart boundaries, and SSW(9) = 1</td>
</tr>
<tr>
<td>EOF</td>
<td>day, month, year, time</td>
<td>End of file found on magnetic tape</td>
</tr>
<tr>
<td>PARITY ER</td>
<td>day, month, year, time</td>
<td>Unidentified error found</td>
</tr>
</tbody>
</table>
PROGRAMMER: Carl Bowin, Hartley Hoskins, J.M. Monyet

ORIGINATOR: Carl Bowin

DATE: May 1973

REFERENCES:
NAME: CONV67
TYPE: Main Program
PURPOSE: Convert gravity data to 1967 Geodetic Reference System and the new basic value of gravity at Potsdam, 981260, mgals.
MACHINE: Sigma-7
SOURCE LANGUAGE: Fortran IV
PROGRAM CATEGORY: Data Processing
DESCRIPTION:
CONV67 converts gravity data, in GSUM format and blocked by 50, from the 1930 gravity formula and Potsdam gravity value to the 1967 Geodetic Reference System and new Potsdam gravity values. The program makes use of the Asynchronous I/O available in XDS extended Fortran IV.

INPUT:
   a) GSUM - blocked by 50 at 1930 datum (IREC=1)

OUTPUT:
   a) GSUM - blocked by 50 at IGSN71 datum (IREC=2) and referenced to International Gravity Formula 1967.

USAGE:

!JOB
!MESSAGE (Mag tape info)
!ASSIGN F:1, (DEVICE,9T), (SN,XXX), (IN), (TRIES,10)
!ASSIGN F:2, (DEVICE,9T), (SN,XXX), (OUT), (TRIES,10)
!OAY (BI), (UNSAT,312), (305), (456), (3)
!RUN

RESTRICTIONS:
   a) uses only tapes blocked by 50
   b) must have GINTF (theoretical gravity function) for 1967 datum.
CONV67 continued

SUBPROGRAMS REQUIRED: GINTF

OPERATIONAL CHARACTERISTICS:

SENSE SWITCH OPTIONS - not applicable

PROGRAM FLOW

Using BUFF IN, BUFF OUT, ENCODE and DECODE, CONV67 performs asynchronous I/O while converting observed gravity, Free-Air anomaly and Bouguer anomaly.

ERRORS AND DIAGNOSTIC MESSAGES:

Waiting for Input - the processing has halted temporarily while a block of data is read into memory

Waiting for Output - the processing has halted temporarily while a block of data is written from memory

End of File on ITAPE - end of file mark encountered on input tape

End of File on JTAPE - end of reel foil encountered on output tape, no reel change will be made.

Buffer In Error - a read error has occurred but it is not fatal and processing will continue. Probably will result in some lost records.

Buffer Out Error - a write error has occurred but it is not fatal and processing will continue. Probably will result in some lost records.

PROGRAMMER: Lee Gove

ORIGINATOR: Carl Bowin

DATE: 1 December 1973

REFERENCES:

NAME: CR2G
TYPE: Main Program
PURPOSE: Converts land gravity meter counter readings to observed gravity values
MACHINE: Sigma-7
SOURCE LANGUAGE: Fortran
PROGRAM CATEGORY: Data Processing

DESCRIPTION:
Takes input of a counter reading and converts the counter reading to gravity, also inserts drift and tidal corrections. The program lists the data for each station, punches cards for sorting, and writes a GSUM format file with anomalies calculated in reference to IGF 1930.

INPUT:
Card 1:
Conversion tables for the counter reading to relative milligal values (I2, F7.2) 70 cards

Card 2: Sense switches (8011)
ISW(1) = 0 for printed output of computed values for each station
       = 1 for suppression of printed output
ISW(2) = 0 to punch output for gravity description program (GDS)
       = 1 for suppression of punched output
ISW(4) = 0 to output FILE TWO in GSUM format in preparation for sorting
       = 1 to suppress output into FILE TWO
ISW(5) = 0 for meter drift correction
       = 1 suppression of drift correction

Card 3: IGM(1), IGM(2), DRFTCO, LSRC, IELC, IGC (2A4, 2X, F10.5, 3I5)
IGM - gravity meter used (e.g. L&R G-18)
       If these are both blank, the type of gravity meter will be set to the default value of 'L&R G-18'
DRFTCO - The correction factor for drift of the gravity meter
       If this is blank, or set to 0.0, a drift variation of 0.003 mgal/day is assumed (default value)
LSRC - Source code of GSUM output. Default value is 006 - the source for the G-18 meter.
IELC - Elevation code for GSUM output. Default value is 09.
IGC - Gravity meter code for GSUM output. Default value is 01.
CR2G (continued) page two

INPUT (continued)

These are followed by groups of individual station counter reading cards. Each group is headed by three cards:

Card A: BASEG(1), BASEG(2) (F3.0, F6.2)
The absolute gravity value for the reference station

Card B: DENSE (F4.2)
The assumed crustal density to be used in calculation of the Bouguer anomaly

Card C: Counter reading card for the reference station. Drift is computed starting with the date on this card.

Card D: Counter reading cards for those stations which will be referenced to the station (card C)
Counter reading cards have the following format.
(format of 17 May 1966):
Station number (I4), Day (I2), Month (I2), Year (I2),
Time (I4), Counter reading (F8.3), Latitude degrees (I2), Latitude minutes (F5.2), North or South (A1),
Longitude degrees (I3), Longitude minutes (F5.2),
West or East (A1), Elevation (F7.1), Time Zone (I2),
and Description (32A1).

Card E: Either a counter reading card with all zeros (or blanks) except for the year value (card columns 9 and 10) - signals the end of a group of stations. Program then tries to read a new absolute gravity value (card A above)

OR A card with all zeros (or blanks) - signals the end of input data.

OUTPUT:

A. Unless sense switch (1) equals 1, records of the following format will be output to the line printer, along with a page heading.

STAT = Station number
DATE = Day, month, year, e.g. 10 Dec. 1970 becomes 101270
TIME = Hour, minute
LAT = Latitude
LONG = Longitude
ELEV = Elevation
CR = Counter reading
RELV = Relative value of gravity to counter reading
DIFF = Difference of gravity between two readings
OBSG = Observed gravity
GFREE = Free-air gravity
BOUG = Bouguer gravity
CLS = Tidal correction
HONK = Honkasolo correction
CR2G (continued) page Three

OUTPUT (continued)

TZONE = Time zone corresponding to time
GDATE = Converted GMT date and time
DAYS = Days into the year
TDIFF = Time difference from origin
DRIFT = Drift correction that is being applied

B. Unless sense switch (4) equals 1, a file in GSUM format will be output to unit number 2 in preparation for sorting.

C. Unless sense switch (2) equals 1, cards will be punched for input to gravity description program.

NOTE: A card is not punched for the reference station

RESTRICTIONS: 1) CAUTION: If the drift of the meter is positive the value of DRFTCO must be negative.

Note also that if a value of 0.00 is entered for DRFTCO, a value of 0.003 will be assumed.

2) A maximum of 9000 cards can be input

STORAGE REQUIREMENTS: 1010 decimal words

SUBPROGRAMS REQUIRED: CDATE, CHGMT, GINTF, M2DY, TIDAL

TIMING: Unknown

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: C. Bowin, J. Wolfe, S. Abbot

ORIGINATOR: C. Bowin

DATE: 1 August 1975
NAME: CR2G67

TYPE: Main Program

PURPOSE: Converts land gravity meter counter readings to observed gravity values

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran

PROGRAM CATEGORY: Data Processing

DESCRIPTION:
Takes input of a counter reading and converts the counter reading to gravity, also inserts drift and tidal corrections. The program lists the data for each station, punches cards for sorting, and writes a GSUM format file with anomalies calculated in reference to International Gravity Formula 1967.

INPUT:
Card 1:
Conversion tables for the counter reading to relative milligal values (I2, F7.2) 70 cards

Card 2: Sense switches (8011)
ISW(1) = 0 for printed output of computed values for each station
= 1 for suppression of printed output
ISW(2) = 0 to punch output for gravity description program (GDS)
= 1 for suppression of punched output
ISW(4) = 0 to output FILE TWO in GSUM format in preparation for sorting
= 1 to suppress output into FILE TWO
ISW(5) = 0 for meter drift correction
= 1 suppression of drift correction

Card 3: IGM(1), IGM(2), DRFTCO, LSRC, IELC, IGC (2A4, 2X, F10.5, 3I5)
IGM - gravity meter used (e.g. L&R G-18)
If these are both blank, the type of gravity meter will be set to the default value of 'L&R G-18'
DRFTCO - The correction factor for drift of the gravity meter
If this is blank, or set to 0.0, a drift variation of 0.003 mgal/day is assumed (default value)
LSRC - Source code of GSUM output. Default value is 006 - the source for the G-18 meter.
IELC - Elevation code for GSUM output. Default value is 09.
IGC - Gravity meter code for GSUM output. Default value is 01.
INPUT (continued)

These are followed by groups of individual station counter reading cards. Each group is headed by three cards:

Card A: BASEG(1), BASEG(2) (F3.0, F6.2)

The absolute gravity value for the reference station

Card B: DENSE (F4.2)

The assumed crustal density to be used in calculation of the Bouguer anomaly

Card C: Counter reading card for the reference station. Drift is computed starting with the date on this card.

Card D: Counter reading cards for those stations which will be referenced to the station (card C)

Counter reading cards have the following format.
(format of 17 May 1966):
Station number (I4), Day (I2), Month (I2), Year (I2), Time (I4), Counter reading (F8.3), Latitude degrees (I2), Latitude minutes (F5.2), North or South (Al), Longitude degrees (I3), Longitude minutes (F5.2), West or East (Al), Elevation F7.1), Time Zone (I2), and Description (32Al).

Card E: Either a counter reading card with all zeros (or blanks) except for the year value (card columns 9 and 10) - signals the end of a group of stations. Program then tries to read a new absolute gravity value (card A above)

OR A card with all zeros (or blanks) - signals the end of input data.

OUTPUT:

A. Unless sense switch (1) equals 1, records of the following format will be output to the line printer, along with a page heading.

STAT = Station number
DATE = Day, month, year, e.g. 10 Dec. 1970 becomes 101270
TIME = Hour, minute
LAT = Latitude
LONG = Longitude
ELEV = Elevation
CR = Counter reading
RELV = Relative value of gravity to counter reading
DIFF = Difference of gravity between two readings
OBSG = Observed gravity
GFREE = Free-air gravity
BOUG = Bouguer gravity
CLS = Tidal correction
HONK = Honkasolo correction
OUTPUT (continued)

TZONE = Time zone corresponding to time
gdate = Converted GMT date and time
days = Days into the year
tdiff = Time difference from origin
drift = Drift correction that is being applied

B. Unless sense switch (4) equals 1, a file in GSUM format will be output to unit number 2 in preparation for sorting.

C. Unless sense switch (2) equals 1, cards will be punched for input to gravity description program.

NOTE: A card is not punched for the reference station.

RESTRICTION: If the drift of the meter is positive, the value of DRFTCO must be negative.

Note also that if a value of 0.00 is entered for DRFTCO, a value of 0.003 will be assumed.

2) A maximum of 9000 cards can be input.

STORAGE REQUIREMENTS: 1010 decimal words

SUBPROGRAMS REQUIRED: CDATE, CHGMT, GINTF, M2Dy, TIDAL

TIMING: Unknown

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: C. Bowin, J. Wolfe, S. Abbot

ORIGINATOR: C. Bowin

DATE: 1 August 1975
NAME: CRWT3

TYPE: Main Program

PURPOSE: To calculate the pressure at the base of a crustal column (Kg/cm²).

MACHINE: Sigma 7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Mathematical (equation solving)

DESCRIPTION:

Reads U. of Toronto World Seismic Refraction Compilation at W.H.O.I. SPFMT format. Data may input on cards or mag tape.

Table look-up values for the conversion of compressional seismic velocity to density are entered during initialization. Compensation depth (e.g. 40 km) is also entered during initialization. Seismic refraction data of SPFMT format is read and the pressure at the compensation depth is then calculated. Crustal thickness, average crustal velocity and depth to mantle are also calculated and output in the SPFMT format.

INPUT:

Card 1: Sense switch options: Put sense switch 0 in column 80.

- ISW(0) = 1 to list intermediate values for testing
- ISW(26) = 1 to output on line printer only
- ISW(32) = 1 to read SPFMT data from two cards per record
- ISW(33) = 1 to write SPFMT data on two cards per record

Card 2: ICTAB, DCOMP (I5,F10.0)
- ICTAB = 0 for Nafe-Drake Density Table
- ICTAB = 1 for Woollard Density table
- DCOMP = depth of compensation (Km).

Card 3: Density table cards (10F8.3) 10 values per card

Card(s) 4: (optional) SPFMT data cards, if data is on cards

Card 5: !EOD if data is on cards
OUTPUT: Data can be output either to mag tape or cards, depending on sense switches and control cards. If ISW(0) = 1, values read in and calculated are listed with annotation on the line printer.

USAGE: Assign F:1 to input device; F:2 to output device

RESTRICTIONS:

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: EVIL, EXIT, ISW, NAVIN, PINOT, STAT, TODAY

TIMING: Undetermined

ERRORS AND DIAGNOSTICS: If ISW(0) = 1, the program outputs annotated lists of values read and calculated.

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 2 October 1974
NAFE-DRAKE EXPERIMENTAL RELATIONSHIP
(In Talwani, Sutton, and Worzel, 1959
JGR, v. 14, No. 10, p. 1548)
(Picks by C. Bowin)

<table>
<thead>
<tr>
<th>V_p</th>
<th>P</th>
<th>V_p</th>
<th>P</th>
<th>V_p</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0</td>
<td>4.0</td>
<td>2.39</td>
<td>7.0</td>
<td>3.04</td>
</tr>
<tr>
<td>1.1</td>
<td>0</td>
<td>4.1</td>
<td>2.41</td>
<td>7.1</td>
<td>3.07</td>
</tr>
<tr>
<td>1.2</td>
<td>0</td>
<td>4.2</td>
<td>2.425</td>
<td>7.2</td>
<td>3.10</td>
</tr>
<tr>
<td>1.3</td>
<td>0</td>
<td>4.3</td>
<td>2.44</td>
<td>7.3</td>
<td>3.13</td>
</tr>
<tr>
<td>1.4</td>
<td>0</td>
<td>4.4</td>
<td>2.45</td>
<td>7.4</td>
<td>3.16</td>
</tr>
<tr>
<td>1.5</td>
<td>1.47</td>
<td>4.5</td>
<td>2.48</td>
<td>7.5</td>
<td>3.19</td>
</tr>
<tr>
<td>1.6</td>
<td>1.66</td>
<td>4.6</td>
<td>2.5</td>
<td>7.6</td>
<td>3.22</td>
</tr>
<tr>
<td>1.7</td>
<td>1.73</td>
<td>4.7</td>
<td>2.52</td>
<td>7.7</td>
<td>3.25</td>
</tr>
<tr>
<td>1.8</td>
<td>1.80</td>
<td>4.8</td>
<td>2.53</td>
<td>7.8</td>
<td>3.28</td>
</tr>
<tr>
<td>1.9</td>
<td>1.86</td>
<td>4.9</td>
<td>2.55</td>
<td>7.9</td>
<td>3.31</td>
</tr>
<tr>
<td>2.0</td>
<td>1.92</td>
<td>5.0</td>
<td>2.57</td>
<td>8.0</td>
<td>3.34</td>
</tr>
<tr>
<td>2.1</td>
<td>1.98</td>
<td>5.1</td>
<td>2.59</td>
<td>8.1</td>
<td>3.38</td>
</tr>
<tr>
<td>2.2</td>
<td>2.01</td>
<td>5.2</td>
<td>2.61</td>
<td>8.2</td>
<td>3.42</td>
</tr>
<tr>
<td>2.3</td>
<td>2.03</td>
<td>5.3</td>
<td>2.62</td>
<td>8.3</td>
<td>3.46</td>
</tr>
<tr>
<td>2.4</td>
<td>2.06</td>
<td>5.4</td>
<td>2.64</td>
<td>8.4</td>
<td>3.49</td>
</tr>
<tr>
<td>2.5</td>
<td>2.09</td>
<td>5.5</td>
<td>2.66</td>
<td>8.5</td>
<td>3.525</td>
</tr>
<tr>
<td>2.6</td>
<td>2.11</td>
<td>5.6</td>
<td>2.68</td>
<td>8.6</td>
<td>3.56</td>
</tr>
<tr>
<td>2.7</td>
<td>2.13</td>
<td>5.7</td>
<td>2.7</td>
<td>8.7</td>
<td>3.59</td>
</tr>
<tr>
<td>2.8</td>
<td>2.15</td>
<td>5.8</td>
<td>2.72</td>
<td>8.8</td>
<td>3.63</td>
</tr>
<tr>
<td>2.9</td>
<td>2.18</td>
<td>5.9</td>
<td>2.74</td>
<td>8.9</td>
<td>3.67</td>
</tr>
<tr>
<td>3.0</td>
<td>2.21</td>
<td>6.0</td>
<td>2.77</td>
<td>9.0</td>
<td>3.71</td>
</tr>
<tr>
<td>3.1</td>
<td>2.23</td>
<td>6.1</td>
<td>2.8</td>
<td>9.1</td>
<td>3.74</td>
</tr>
<tr>
<td>3.2</td>
<td>2.24</td>
<td>6.2</td>
<td>2.83</td>
<td>9.2</td>
<td>3.78</td>
</tr>
<tr>
<td>3.3</td>
<td>2.26</td>
<td>6.3</td>
<td>2.85</td>
<td>9.3</td>
<td>3.82</td>
</tr>
<tr>
<td>3.4</td>
<td>2.28</td>
<td>6.4</td>
<td>2.87</td>
<td>9.4</td>
<td>3.85</td>
</tr>
<tr>
<td>3.5</td>
<td>2.30</td>
<td>6.5</td>
<td>2.9</td>
<td>9.5</td>
<td>3.88</td>
</tr>
<tr>
<td>3.6</td>
<td>2.32</td>
<td>6.6</td>
<td>2.93</td>
<td>9.6</td>
<td>3.91</td>
</tr>
<tr>
<td>3.7</td>
<td>2.34</td>
<td>6.7</td>
<td>2.95</td>
<td>9.7</td>
<td>3.95</td>
</tr>
<tr>
<td>3.8</td>
<td>2.36</td>
<td>6.8</td>
<td>2.975</td>
<td>9.8</td>
<td>3.99</td>
</tr>
<tr>
<td>3.9</td>
<td>2.375</td>
<td>6.9</td>
<td>3.01</td>
<td>9.9</td>
<td>4.02</td>
</tr>
</tbody>
</table>

10.0 | 4.06 |
VEL-DENS RELATIONSHIP  
(From Woollard (1959))

<table>
<thead>
<tr>
<th>VEL. Km/sec</th>
<th>( \rho ) gm/cm(^3)</th>
<th>VEL. Km/sec</th>
<th>( \rho ) gm/cc</th>
<th>VEL. Km/sec</th>
<th>( \rho ) gm/cm(^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.62</td>
<td>4.0</td>
<td>2.61</td>
<td>7.0</td>
<td>3.06</td>
</tr>
<tr>
<td>1.1</td>
<td>1.62</td>
<td>4.1</td>
<td>2.62</td>
<td>7.1</td>
<td>3.09</td>
</tr>
<tr>
<td>1.2</td>
<td>1.63</td>
<td>4.2</td>
<td>2.62</td>
<td>7.2</td>
<td>3.12</td>
</tr>
<tr>
<td>1.3</td>
<td>1.66</td>
<td>4.3</td>
<td>2.62</td>
<td>7.3</td>
<td>3.15</td>
</tr>
<tr>
<td>1.4</td>
<td>1.69</td>
<td>4.4</td>
<td>2.62</td>
<td>7.4</td>
<td>3.17</td>
</tr>
<tr>
<td>1.5</td>
<td>1.74</td>
<td>4.5</td>
<td>2.62</td>
<td>7.5</td>
<td>3.20</td>
</tr>
<tr>
<td>1.6</td>
<td>1.81</td>
<td>4.6</td>
<td>2.62</td>
<td>7.6</td>
<td>3.23</td>
</tr>
<tr>
<td>1.7</td>
<td>1.88</td>
<td>4.7</td>
<td>2.63</td>
<td>7.7</td>
<td>3.25</td>
</tr>
<tr>
<td>1.8</td>
<td>2.06</td>
<td>4.8</td>
<td>2.64</td>
<td>7.8</td>
<td>3.28</td>
</tr>
<tr>
<td>1.9</td>
<td>2.18</td>
<td>4.9</td>
<td>2.65</td>
<td>7.9</td>
<td>3.31</td>
</tr>
<tr>
<td>2.0</td>
<td>2.27</td>
<td>5.0</td>
<td>2.66</td>
<td>8.0</td>
<td>3.33</td>
</tr>
<tr>
<td>2.1</td>
<td>2.34</td>
<td>5.1</td>
<td>2.67</td>
<td>8.1</td>
<td>3.36</td>
</tr>
<tr>
<td>2.2</td>
<td>2.39</td>
<td>5.2</td>
<td>2.68</td>
<td>8.2</td>
<td>3.39</td>
</tr>
<tr>
<td>2.3</td>
<td>2.42</td>
<td>5.3</td>
<td>2.69</td>
<td>8.3</td>
<td>3.42</td>
</tr>
<tr>
<td>2.4</td>
<td>2.45</td>
<td>5.4</td>
<td>2.70</td>
<td>8.4</td>
<td>3.45</td>
</tr>
<tr>
<td>2.5</td>
<td>2.49</td>
<td>5.5</td>
<td>2.71</td>
<td>8.5</td>
<td>3.47</td>
</tr>
<tr>
<td>2.6</td>
<td>2.51</td>
<td>5.6</td>
<td>2.73</td>
<td>8.6</td>
<td>3.50</td>
</tr>
<tr>
<td>2.7</td>
<td>2.53</td>
<td>5.7</td>
<td>2.74</td>
<td>8.7</td>
<td>3.53</td>
</tr>
<tr>
<td>2.8</td>
<td>2.55</td>
<td>5.8</td>
<td>2.76</td>
<td>8.8</td>
<td>3.55</td>
</tr>
<tr>
<td>2.9</td>
<td>2.56</td>
<td>5.9</td>
<td>2.79</td>
<td>8.9</td>
<td>3.58</td>
</tr>
<tr>
<td>3.0</td>
<td>2.56</td>
<td>6.0</td>
<td>2.81</td>
<td>9.0</td>
<td>3.61</td>
</tr>
<tr>
<td>3.1</td>
<td>2.57</td>
<td>6.1</td>
<td>2.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>2.58</td>
<td>6.2</td>
<td>2.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.3</td>
<td>2.59</td>
<td>6.3</td>
<td>2.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.4</td>
<td>2.59</td>
<td>6.4</td>
<td>2.90</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td>2.60</td>
<td>6.5</td>
<td>2.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.6</td>
<td>2.60</td>
<td>6.6</td>
<td>2.96</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.7</td>
<td>2.60</td>
<td>6.7</td>
<td>2.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.8</td>
<td>2.61</td>
<td>6.8</td>
<td>3.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.9</td>
<td>2.61</td>
<td>6.9</td>
<td>3.04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
NAME: DMABLK

TYPE: Main Program

PURPOSE: Converts blocked DMA format data to blocked GSUM format data

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:
DMABLK is a modification of DMA which, in turn, is a modification of program ACTG3.

DMABLK reads DMA data blocked by 50 and converts data with elevation codes 1 and 3 to GSUM formatted data blocked by 50. Source code and beginning sequence number are entered at run time; sequence number is output in station number field. Data records with elevation codes other than one and three are output to another tape in DMA format for further processing.

INPUT:
Card 1: NSEQ (I10) - starting sequence number
Card 2: ISORC (I5) - source code for this data

ERRORS AND DIAGNOSTICS:
'WAITING FOR I/P' - input buffer not yet filled when checked
'END OF FILE ON ITAPE' - end of file found on input tape
'BUFFER IN ERROR' - input buffer error detected by ICHECK
'WAITING FOR OUTPUT' - output buffer not yet filled when checked
'END OF FILE JTAPE' - end of reel encountered on output tape
'BAD JKEY' - end of reel encountered on output tape
'ALL DONE'
OUTPUT:

On unit reference number 2: GSUM records for elevation codes 1 and 3.
On unit reference number 3: DMA records for other elevation codes.

The number of records input, records output to each output tape, and ending sequence number are output to line printer.

USAGE:

Assign F:1 to input device; F:2 to output device for GSUM records; F:3 to output device for 'oddball' records (elevation codes other than 1 and 3).

RESTRICTIONS: None

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: ALTD, XEROX Fortran IV Library

TIMING: about 1,000 records per minute

PROGRAMMER: Lee Gove, C. Bowin

ORIGINATOR: C. Bowin

DATE: 31 July 1975
NAME: DMAP

TYPE: Main Program

PURPOSE: Converts digitized position in inches to latitude and longitude (radians)

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:
Digitized x and y coordinate values from a digitizing table for the four corners of a map region and read by the program along with latitude and longitude of each point. From this information the angle of tilt of the map and its scale are determined. Other x and y coordinate data points are then read in and the latitude and longitude of each is calculated to a precision governed by the input value for the variable EPSIL.

INPUT:

Card 1: sense switches

SSW(2) = 1 to list intermediate values
SSW(3) = 1 to list date and SMIN for each data point
SSW(5) = 1 to output SMIN only if greater than EPSIL

Card 2: ITAPE, JTAPE, EFAC, EPSIL (2I5,2F5.2)

ITAPE - unit number for input device
JTAPE - unit number for output device
EFAC - factor (0.1 to 1.03) used on iteration for estimated latitude to converge on true latitude
EPSIL - tolerance (in meridional parts) by which estimated latitude must match meridional parts for true latitude.

Card 3, 4, 5, and 6: ICODE, XC(J), YC(J), N1, N2, LAT(J) LONG(J)

ICODE = 9 for cards 3, 4, 5 and 6 for initialization
J in do loop is = 1 for bottom left corner,
then 2,3,4 counter clockwise around map corners
XC(J) = X coordinate value in inches
YC(J) = Y coordinate value in inches
N1 = not used
N2 = not used
LAT(J) = Latitude
LONG(J) = Longitude
Card 7: ICODE, XP, YP, NDA, NMO, NYR, NHM  
     (1L, 1X, F5.3, 1X, F5.3, 3I3,I5)

ICODE = 5 for data points  
XP = X coordinate value in inches  
YP = Y coordinate value in inches  
NDA = Day  
NMO = Month  
NYR = Year  
     = 0 on terminator card to indicate last data point has been processed.  
NHM = Hours and minutes (24 hours clock)

OUTPUT:

Outputs record containing latitude and longitude for each input data point.

USAGE: Assign input and output devices compatible with ITAPE and JTAPE values entered on card 2.

RESTRICTIONS: None

STORAGE REQUIREMENTS: Undetermined

SUBPROGRAMS REQUIRED: CALSC, DMTOR, ISW, PARTM, RTODM

TIMING:

ERRORS AND DIAGNOSTICS: Undetermined

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 26 September 1975
NAME: DMOD
TYPE: Main Program
PURPOSE: To punch digitized polygon points for crustal models digitized on a digitizing table
MACHINE: Sigma 7
SOURCE LANGUAGE: Fortran IV
PROGRAM CATEGORY: Format Conversion

DESCRIPTION:

This program converts the coordinates for polygon points punched by a digitizing table to the correct format for use in a TALPLO™ run and punches the cards necessary for all polygon points using origin and scale factors input to this program at run time. Input and output are always on cards. For instructions in the use of the digitizing table, see comments under USAGE.

The program initializes by reading sense switches, scale factors and coordinates origin. It sets the origin to the coordinates of first digitized point entered, uses the second digitized point to establish a horizontal reference line and then calculates X and Y distances of all points from the origin using the input scale factors. It punches cards with the adjusted and scaled X and Y coordinates along with the identification number of each polygon; one card for each polygon in which the point occurs.

INPUT:

Card 1: Sense Switch Options: (8011)

ISW(1) = 0 for second point to right (+) of origin
= 1 for second point to left (-) of origin

Card 2: XFAC,YFAC,XORG,YORG (4 F10.0)

XFAC = scale factor in X direction (km/in)
YFAC = scale factor in Y-direction (km/in)
XORG = X-coordinate of origin of model (km)
YORG = Y-coordinate of origin of model (km)

The following cards are all punched at the digitizing table

Card 3: XA,YA,IA,KP1,KP2,KP3 (2F10.3,15,314)

XA X and Y Coordinates from digitizing table from
YA its origin

IA =0 (same format as card(s) 5 below, but not used for this point)
Card(s) 5:  Values from the digitizing table for the polygon points XP,YP,ICODE,KP1,KP2,KP3 (2F10.3,15,3I4)

    XP  X and Y coordinates of the polygon point
    YP

    ICODE =  9 for last point of a polygon
    =  8 for X=-3000 km
    =  7 for X=+3000 km

    (ICODE=7 or 8 is used in this program only. Points with ICODE = 7 or 8 are punched by this program with ICODE = 0).

    KP1  numbers of the polygons for which this point forms a boundary. One output card will be punched for each polygon listed here.

    KP2

    To indicate end of input cards, an additional polygon point card with ICODE set equal to 99 must follow the last digitized point.

OUTPUT:  On line printer: the digitized points

On cards:  Cards in the correct format for use in TALPLOT run. Values punched are X coordinate in km., Y coordinate in km., ICODE, and the number of the polygon for which the card was punched. Cards will usually not be in the correct order and there may be some extra cards (if the first polygon point is not the first polygon point for another polygon that it defines).

USAGE:  A crustal model is prepared which is composed of various polygons of various densities. The polygons are numbered arbitrarily, with the exception of polygon number 1, which is a water layer, and the final polygon, which must be number 99. One point of each polygon is designated the "starting point". Points define the polygons by proceeding clockwise from the starting point and ending exactly at the same point. Polygon points must be arranged in order for input to the TALPLOT program, but need not be digitized in order nor input in order to the program IMOD. Output from program DMOD must be rearranged for output to TALPLOT.

At the digitizing table, the first point digitized must be the origin. The second point is a point on the same X axis as the origin, and is used to establish the horizontal for the model. The remaining points may be digitized in any order. Before lining up a point, ICODE is set in the leftmost thumbwheel switch position on the manual entry switches. ICODE = 9 to indicate the last point of any given polygon. ICODE = 7 will punch a card at the same Y coordinate as the point under the digitizing screen, but
with +3000 km as the X coordinate. ICODE = 8 creates a card with -3000 km as the X coordinate. These are used at the sides of the model to extend the edges of the polygons beyond the area for which gravity will be calculated in order to avoid an edge effect. In addition, the numbers of the polygons for which the given point delineates a boundary are set in the three pairs of thumbwheel switches to the right of the leftmost thumbwheel switch. In the DMOD program, a polygon point coordinate card is punched for each polygon number inserted here.

RESTRICTIONS:

1) When punching the first and last cards for each polygon on the digitizer, make sure that the cards read exactly the same values - otherwise the polygon will not close.

2) Right and down are positive on the model graph. That means that Y coordinates of polygon points below the sea surface are positive numbers.

3) Input and output must be on cards.

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: CALSC, EXIT, ISW

TIMING: Undetermined

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 24 July 1975
NAME: GFLD1
TYPE: Main Program
PURPOSE: Calculation of regional free-air anomaly values for a given region from spherical harmonic coefficients.

MACHINE: SIGMA 7
SOURCE LANGUAGE: FORTRAN IV
PROGRAM CATEGORY: Data Processing

DESCRIPTION:
Area bounds and increment size for region of interest are entered. GFLD1 next reads spherical harmonic coefficients defining a gravitational field. The program then steps across the region defined by the input area bounds calculating the regional gravity field at each position increment. Program outputs the regional free-air anomaly value in the free-air position of SEAG1 format.

INPUT:

Card 1: Sense switches (8011)
   ISW(4) = 1 to list data on high speed printer

Card 2: ITAPE, JTAPE (215)
   ITAPE = input device number (used for input of spherical harmonic coefficients)
   JTAPE = output device number

Card 3: ITOP, IBOT, ILLEFT, IRIGHT, INC (5I5)
   ITOP = integer degree for top area boundary
   IBOT = integer degree for bottom area boundary
   ILLEFT = integer degree for left area boundary
   IRIGHT = integer degree for right area boundary
   INC = integer degree increment for do loop in defining positions at which regional free-air anomalies will be calculated.

Card 4: Spherical harmonic coefficients
   Format (I2,2X,I2,2X,E11.4,2X,E11.4) followed by 2 !EOD card - this input can be on magnetic tape or disc by appropriate value of ITAPE, on card 2 above

OUTPUT:
On unit reference JTAPE. The data in SEAG1 format with regional free-air values in free-air field.
GFLD1

**USAGE:** Assign input and output devices to ITAPE and JTAPE values input on card 2.

**RESTRICTIONS:** None

**STORAGE REQUIREMENTS:** Unknown

**SUBPROGRAMS REQUIRED:** ISW, FLD2, standard Fortran IV Library

**TIMING:**

**ERRORS AND DIAGNOSTICS:**

**PROGRAMMER:**

**ORIGINATOR:**

**DATE:**
NAME: GFLD2

TYPE: Main Program

PURPOSE: Calculates regional free-air gravity anomalies from spherical harmonic coefficients entered at run time for location of input GSUM records.

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Data processing

DESCRIPTION:
GFLD2 reads spherical harmonic coefficients defining a gravitational field. A regional gravity value is computed from the input coefficients at latitude and longitude locations read from input GSUM records.

Program outputs regional free-air value in regional gravity position of GSUM format.

INPUT:
Card 1: Sense switches (80I2)

| ISW(5) | 1 to list intermediate values for BV, COSD on line printer (SUB. FLD2) |
| ISW(12) | 1 to list date identification on line printer (SUB. GINOT) |
| ISW(26) | 1 to output on line printer only (SUB. GINOT) |
| ISW(29) | 1 to process only selected source codes = 2 to skip selected source codes (SUB. GINOT) |
| ISW(30) | 1 for input data on cards (SUB. GINOT) |
| ISW(31) | 1 for output data on cards (SUB. GINOT) |
| ISW(40) | 0 to process without bounds = 1 to process with bounds using the Data Location Table (SUB. GINOT) |
| ISW(60) | 1 to process only Abstracter output (SUB. GINOT) |
| ISW(61) | 1 to replace FA, BG, ELEV, LAT, AND LONG with averaged values (SUB. GINOT) |

Card(s) 2: Spherical harmonic coefficients format (12,2X,12,2X,El1.4,2X,El1.4) followed by a !EOD card

Card 3: (optional) ISRC (16I5) See example for SAO Standard Earth 1969
If ISW(29) does not equal zero, enter here up to 16 source codes to be selected (ISW(29)=2) or skipped (ISW(29)=1).

If input is on magnetic tape:
GFLD2

Card(s) 4: Serial number(s) of input tapes, one per card, in columns 1 to 4 (used by subroutine MOUNT)

Card 5: EITP in columns 1 to 4 - signals end of input tape serial numbers

If input is on cards:

Card(s) 4: data cards in GSUM format-two cards per record

Card 5: !EOD card

If output is to magnetic tape:

Card(s) 6: Serial number(s) of output tapes, one per card in columns 1 to 4 (used by subroutine MOUNT).

Card 7: EOTP in col. 1 to 4 - signals end of output tape serial numbers.

There will be no cards 6 or 7 if output is on cards.

OUTPUT: On unit reference no. 2 - the data in GSUM format, with regional free-air values in regional free-air field.

USAGE: Assign F:1 to input device; F:2 to output device

RESTRICTIONS: None

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: FLD2, GINOT, ISW, MOUNT, STAT, TODAY

Standard Fortran IV Library

TIMING: Undetermined

ERRORS AND DIAGNOSTICS: 'EOF FOUND ON INPUT TAPE'

PROGRAMMER: A. Folinsbee, Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 5 August 1975
NAME: GRAFG2
TYPE: Main Program
PURPOSE: To plot one variable versus another
MACHINE: Sigma - 7
SOURCE LANGUAGE: FORTRAN IV
PROGRAM CATEGORY: Graphical Display

DESCRIPTION:

GRAFG2 creates a graph, plotting one variable against another. It uses input either in GSUM (gravity summary) format, or WHOIG (WHOI lunar gravity) format. It uses latitude and longitude for processing bounds. In addition, it can use either start and end dates or altitude bounds (in lunar format) as further limits to the data processed, if desired.

INPUT:

CARD 1: LABEL (20A4)
        information for plot label

CARD 2: sense switch settings:
        options

    ISW(0) = 1 to output values for testing
    ISW(3) = 1 to plot Lunar gravity (calls GETL)
    ISW(4) = 1 to read SVEC altitude bounds for lunar data and process data only within these altitude bounds.
    ISW(7) = 1 to input new area bounds for next plot
    ISW(8) = 1 to suppress plotting grid
    ISW(10) = 1 to start a new graph
    ISW(12) = 1 to list date identification
    ISW(13) = 1 to annotate plot point with DATAW
    ISW(30) = 1 to read GSUM data from cards
    ISW(34) = 1 to read Lunar data from cards
CARD 3: ITAPE, NX, NY, NZ, NW, IDEC, KPT, KHT (8I5)

ITAPE = input tape device number (must agree with control cards) Should = 105 to read from cards.
NX  = PLT(NX) for X variable
NY  = PLT(NY) for Y variable
NZ  = PLT(NZ) for Z variable
NW  = PLT(NW) for W variable
IDEC = code for decimal point in annotation of DATAW
KPT = plot size factor-varies size of entire plot (should = 1 in standard plot)
KHT = character height factor (varies by multiples of 0.07)

CARD 4: XFAC, YFAC, ZFAC, WFAC, ANGB, XINC, YINC (7F10.0)

XFAC = engineering units per inch on X axis
YFAC = engineering units per inch on Y axis
ZFAC = engineering units per inch on Z axis
WFAC = engineering units per inch on W axis
ANGB = angle for DATAW annotation
XINC = spacing in decimal inches for annotation in x-direction
YINC = spacing in decimal inches for annotation in y-direction

CARD 5: TOP, BOT, DLEFT, RIGT. (4F10.0)

TOP
BOT
DLEFT
RIGT

bounds for graph in engineering units

CARD(s) 6-9: area bounds in degrees and decimal minutes, one per card (I5,F10.0)

(6) IDEG, AMIN (TOP)
(7) " " (BOTTOM)
(8) " " (LEFT)
(9) " " (RIGHT)

Note: The sign of the minutes must agree with the sign of the degrees (e.g. -36 -30.0).
GRAFG2 continued

Values for NX,NY,NZ, and NW depend on input format.

For GSUM format

- 1 for KGHM (time)
- 2 for ISORC (source code)
- 3 for ELEV (elevation)
- 4 for DEPTH (depth)
- 5 for HEIGHT (both depth and elevation in the same parameter)
- 6 for FA (free-air anomaly)
- 7 for BG (Bouguer anomaly)
- 8 for TC (terrain correction)
- 9 for BGCOM (complete Bouguer anomaly)
- 10 for RFA (regional free air)
- 11 for GOBS (observed gravity)
- 12 for HEIGHT/BG

For lunar data (WOLG format)

- 1 for SVEC (vehicle distance from center of mass in km)
- 2 for SVEC-1738.0 (vehicle distance from center of mass in km minus radius)
- 3 for ALTL (laser altitude)
- 4 for (SVEC-ALTL)-1738.0
- 5 for AZ (azimuth)
- 6 for SINC (inclination)
- 7 for STAC (tangential acceleration)
- 8 for SNAC (normal acceleration)
- 9 for FA (free-air anomaly-radial acceleration)
- 10 for THEOR (theoretical gravity)
- 11 for GOBS (observed gravity)
- 12 for ELEV (elevation of topography with reference to radius)
- 13 for ELFL (laser altitude, with reference to radius)
- 14 for BG (Bouguer anomaly)
- 15 for TACEL (total acceleration)
CARD 10: (optional) BSVEC, TSVEC (2F10.3)
spacecraft altitude bounds for lunar data
if ISW (4) = 1.
BSVEC - lower altitude limit
TSVEC - upper altitude limit

CARD 11: ISTDA, ISTMO, ISTYR, ISTHM, IENDA, IENMO,
IENYR, IENHM, ISKP
(3I2, I4, 5X, 3I2, I4, 5X, I5)
Start date for processing
ISTDA - day
ISTMO - month
ISTYR - year
ISTHM - time

End date for processing
IENDA - day
IENMO - month
IENYR - year
IENHM - time
ISKP - number of records to be skipped
at start of job. Much faster than
start date alone.

To avoid checking for start date, use a blank
card. Plotting will then begin with the first
record.

CARD 12: ITAPID (I4)
Serial number of input tape(s), one per card.

CARD 13: EITP in columns 1-4
(signals end of input tape serial numbers)
There will be no cards 12 and 13 if input is on cards.

CARD(s) 14:
additional start and end dates for processing
may be inserted here - last card must have
start date = 99 to end processing.

OUTPUT:
Input parameters are listed on line printer. Graph can be
output either to versatec or to calcomp plotter.
GRAFG2 continued

USAGE:

Any number of additional graphs may be run in the same job, by use of sense switches, and inserting additional data cards 6-11, to process data with new area bounds, or just a new start date (card 11).

RESTRICTIONS:

STORAGE REQUIREMENTS: 30 peak core pages (Core 15), on the limit card

SUBPROGRAMS REQUIRED: DMTOR, EXIT, FIND, GETG, GETL, GRIDG, ISW, NUMBER, PLOT, PLOTS, SETSKP, SKPREC, SPOT, STAT, SYMBOL, TODAY, WHERE

TIMING: Undetermined

ERRORS AND DIAGNOSTICS:

'END DATE PASSED' date

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 2 June 1975
NAME: GRAV1

TYPE: PROGRAM

PURPOSE: Converts data input at format of 8 July 1969 to GSUM format.

MACHINE: SIGMA-7

SOURCE LANGUAGE: FORTRAN IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION: GRAV1 converts data input at format of 8 July 1969 to 128 character GSUM format. Latitude and longitude are output in decimal degrees. Observed gravity can be calculated from FA anomaly, and is referenced to the IGSN-71 datum. The Bouguer anomaly is calculated, using reference density entered at run time. Anomalies may be input either in 1930 or 1967 International Gravity Formula, but all output is in 1967 IGF. If necessary, station numbers are assigned, numerically starting with first input record as 1. Resulting records can be listed on line printer, or intermediate values can be output, if desired. Data may be input and output either on cards or on magnetic tapes, depending upon control cards.

INPUT:

Card 1: Sense switch settings - Punch SSW(0) in column 80.
SSW(0) = 1 for input elevation in feet
= 0 for input elevation in meters
SSW(2) = 1 for input depth in fathoms
= 0 for input depth in meters
SSW(3) = 1 to calculate observed gravity from FA anomaly
SSW(4) = 1 for input data at Potsdam Reference System and 1930 International Gravity Formula
= 0 for input data at IGSN-71 datum and 1967 IGF
SSW(6) = 1 for incorporating terrain correction
      = 0 not to use terrain correction
SSW(8) = 1 to print values of THEO, FELEV, & TH67
SSW(13) = 1 to assign station numbers, numerically,
             starting with first input record as 1
SSW(26) = 1 'to output on line printer only (GINOT)
SSW(31) = 1 to output data on two cards per record
             (GINOT)

Card 2:  CRDEN  (FIO.0)
          CRDEN  - assumed crustal density

Card 3:  If output is to be on mag tape, output tape serial
          number (I4), in columns 1-4, one per card, as
          many cards as necessary

Card 4:  EOTP in columns 1-4. Signals end of output tape serial
          numbers. Not necessary if output is on cards or
          line printer

Card(s) 5:  Data cards if input is on cards

Card 6:  !EOD

OUTPUT:  Assumed crustal density, as input, is output to line
         printer. Records are output to line printer or output
         device depending on sense switch options. Records
         output to line printer begin with the second character
         of the record, and do not include IREC2. In addition,
         intermediate values for theoretical gravity may be
         output.

USAGE:  Assign F:1 to input device; F:2 to output device.

RESTRICtIONS:  None

STORAGE REQUIREMENTS:  21 peak core (pages)
SUBPROGRAMS REQUIRED: AREA, STAT, GINOT, GINTF, GI67F, ISW, NAVIN, EVIL, RTDM2

TIMING: Unavailable

ERRORS AND DIAGNOSTICS: None

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 27 May 1975
NAME: GSTOG

TYPE: Main Program

PURPOSE: Converts data input in SEAG1 or SEAG2 format to 128 character GSUM format.

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:
GSTOG is our standard conversion program. It inputs data in SEAG format and outputs 128-character GSUM format. It tests for invalid observed gravity and invalid free-air anomaly values. If either data parameter is invalid, that record is rejected. Checks for no depth or elevation information and if lacking sets Bouguer anomaly value to invalid code (999.0).

If IREC = 1, input data is at Potsdam system and used IGF 1930 (SEAG1 format).

If IREC = 2, input data is at IGSN71 and used IGF 1967 (SEAG2 format) and proper output will be provided at GSUM format.

INPUT:
Card 1: Sense Switch options -
SSW(26) = 1 to output on line printer only (GINOT)
SSW(31) = 1 to output data on two cards per record (GINOT)

Card 2: ISORC, IDCOD, IELC, IGC, BIAS (4I5,F10.0)
ISORC = source code number
IDCOD = 0 for ID by date
       = 1 for ID by station number
IELC = elevation code
IGC = gravity meter code
BIAS = gravity meter bias (in mgals)

Card 3: NFILE (I5) NFILE = number of files to be input

Card 4: EITP in columns 1-4
GSTOG

Card 5: If output is to be on mag tape, output tape serial number (I4), in columns 1-4, one per card, as many cards as necessary.

Card 6: EOTP in columns 1-4. Signals end of output tape serial numbers. Not necessary if output is on cards or line printer.

OUTPUT:

Input parameters are output to line printer. Number of records output and number of records rejected are output to line printer.

Data records can be output on cards if desired by appropriate use of control cards and sense switches. Records may be output to line printer and if so, begin with the second character of the record and do not include IREC.

USAGE:

Assign F:1 to input device; F:2 to output device

RESTRICTIONS:

STORAGE REQUIREMENTS: 23 peak core pages (core,12)

SUBPROGRAMS REQUIRED: BICOR, EVIL, EXIT, GINOT, ISW, MCVOL, STAT, TODAY, UNPKBY

TIMING: CPU time = 12.9 min. to process 10,500 input records; 8,400 output records.

ERRORS AND DIAGNOSTICS:

PROGRAMMER:

ORIGINATOR: Carl Bowin

DATE: 10 July 1975
NAME: GSTOG67

TYPE: Main Program

PURPOSE: Converts data input in SEAG1 or SEAG2 format to 128 character GSUM format and converts Potsdam system data to IGSN71

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:
GSTOG is our standard conversion program. It inputs data in SEAG format and outputs 128-character GSUM format. It tests for invalid observed gravity and invalid free-air anomaly values. If either data parameter is invalid, that record is rejected. Checks for no depth or elevation information and if lacking sets Bouguer anomaly value to invalid code (999.0).

If IREC = 1, input data is at Potsdam system and used IGF 1930 (SEAG1 format). OUTPUT is at IGSN 1971.

If IREC = 2, input data is at IGSN71 and used IGF 1967 (SEAG2 format) and proper output will be provided at GSUM format.

INPUT:

Card 1: Sense Switch options -
SSW(26) = 1 to output on line printer only (GINOT)
SSW(31) = 1 to output data on two cards per record (GINOT)

Card 2: ISORC, IDCOD, IELC, IGC, BIAS (4I5,F10.0)
ISORC = source code number
IDCOD = 0 for ID by date
        = 1 for ID by station number
IELC = elevation code
IGC = gravity meter code
BIAS = gravity meter bias (in mgals)

Card 3: NFILE (I5) NFILE = number of files to be input

Card 4: EITP in columns 1-4
Card 5: If output is to be on mag tape, output tape serial number (I4), in columns 1-4, one per card, as many cards as necessary.

Card 6: EOTP in columns 1-4. Signals end of output tape serial numbers. Not necessary if output is on cards or line printer.

OUTPUT:

Input parameters are output to line printer. Number of records output and number of records rejected are output to line printer.

Data records can be output on cards if desired by appropriate use of control cards and sense switches. Records may be output to line printer and if so, begin with the second character of the record and do not include IREC.

USAGE:

Assign F:1 to input device; F:2 to output device

RESTRICTIONS:

STORAGE REQUIREMENTS: 23 peak core pages (core,12)

SUBPROGRAMS REQUIRED: BICOR, EVIL, EXIT, GINOT, ISW, MCVOL, STAT, TODAY, UNPKBY

TIMING: CPU time = 12.9 min. to process 10,500 input records; 8,400 output records.

ERRORS AND DIAGNOSTICS:

PROGRAMMER:

ORIGINATOR: Carl Bowin

DATE: 10 July 1975
NAME: G3DCP

TYPE: Main Program

PURPOSE: Computes gravity anomaly (for both flat and curved planetary surfaces), potential field (for flat surfaces), and mass per unit area for a set of polygonal laminae comprising a three-dimensional crustal structure model.

MACHINE: XDS Sigma 7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Modeling

DESCRIPTION:

The program is based on a method developed by Talwani and Ewing (1960). The three-dimensional body is represented by depth contours. The depth contours are replaced by flat polygonal laminae. The gravity anomaly is evaluated for the laminae by a formula expressed in a closed form. A correction is then calculated for the curved surface and this correction is applied to the flat anomaly to arrive at a curved anomaly (see Bowin et al., in press). The anomaly for the entire body is then obtained by a numerical integration for the values of the individual laminae. The Z-axis is chosen positive down. The X and Y axes can lie along any two mutually perpendicular directions in the horizontal plane. The points where the anomaly is to be calculated are called field points. These are read from cards. The anomaly for each lamina is calculated in turn and a numerical integration is then performed to obtain the anomaly for the entire body. In addition the potential field and mass per unit area is also calculated.

INPUT:

Cards (via F:1)

Card 1: Sense switch card

Card 2: COORD Initialization card

Transverse Mercator Coordinate for X (FX)
Transverse Mercator Coordinate for Y (FY)
Latitude in Radians for point (X,Y) (RLAT)
Longitude in Radians for point (X,Y) (RLONG)
Starting switch (IST)
G3DCP continued page 2

Input (continued)

Card 3:

Reference Weight (RFW; F8.1)
Reference Density (RFD; F4.2)
Reference Gravity (RFG; F5.1)

FILE: (via F:3)

The data representing the body as output by G3DCPREP.

OUTPUT:

PRINTER (via F108)

Printed output of various integration and corrections for each lamina.

USAGE:

In spite of indications to the contrary G3DCP, when used with LSORT and G3DCPREP is reasonably straightforward to use.

STEP 1 Definition of the bodies to be used

Each body for which an anomaly will be calculated must be defined lamina by lamina. A file is created for each body which has a group of records of the following form for each lamina.

Record 1

Lamina number (NCNT; I2)
Density (RHO; F10.4)
Vertical distance from origin (Z; F16.6)

Following Records

X coordinate of polygon point (X; F12.5)
Y coordinate of polygon point (Y; F12.5)
Last point flag = 1 for last point in lamina (LSLPT; I1)

STEP 2 Laminae Sorting

The bodies to be used are input to LSORT. This program will check that a user specified minimum lamina thickness is observed and counts the number of lamina per body.
G3DCP continued page 3

OUTPUT (continued)

STEP 3

Combination of bodies into one model

The bodies to be used in the particular model are combined into one file by program G3DCPREP in a format acceptable to G3DCP. At this point a reference density is specified for each body that will be subtracted from the density in the input file.

STEP 4 Calculation of anomalies

The model and field points are input to G3DCP and the anomalies are calculated.

SAMPLE RUN

STEP 1 Definition of bodies

The bodies must be in the following format whether they are produced by hand or by some modeling program

Body 1 (file BOD1)
<table>
<thead>
<tr>
<th></th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
<th>10°00</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2.8700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>39<em>92</em>48</td>
<td>29000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**BODYZ (FILE BOD2)**
**STEP 2** Laminae Sorting (see LSORT documentation)

```
!ASSIGN F:1,(FILE,BOD1),(IN)
!ASSIGN F:2,(FILE,BOD2),(IN)
!ASSIGN F:7,(FILE,SORT1),(OUT),(SAVE)
!ASSIGN F:8,(FILE,SORT2),(OUT),(SAVE)
!ASSIGN F:13,(FILE,SORTCNT),(OUT),(SAVE)
!LOAD(BI),(UNSAT,(305),(312),(3))
    LSORT Binary Deck
!RUN
!DATA
    0.1
    02
!EOD
```

**STEP 3** Combination of Bodies (see G3DCPREP documentation)

```
!ASSIGN F:1,(FILE,SORT1),(IN),(SAVE)
!ASSIGN F:2,(FILE,SORT2),(IN),(SAVE)
!ASSIGN F:13,(FILE,SORTCNT),(IN),(SAVE)
!ASSIGN F:7,(FILE,G3DCIN),(OUT),(SAVE)
!LOAD (BI),(UNSAT,(3))
    G3DCPREP Binary Deck
!RUN
!DATA
    02
    1.03
    3.3
!EOD
```

FILE G3DCIN
STEP 4 Calculation of Anomalies

!ASSIGN F:1,(DEVICE,SI)
!ASSIGN F:3,(FILE,G3DCIN),(IN),(SAVE)
!LOAD (BI),(UNSAT,(514),(456),(305),(312),(3))
!RUN
!DATA
   Sense switch card
   Coord initialization card
   Reference card
   Field point card (s)
   Last field point card has 1 in col.43
!EOD

STORAGE REQUIREMENTS: 30,000 decimal words

SUBPROGRAMS REQUIRED: COORD, STAT, GINOT, PLANET, FORTRAN IV library

PROGRAMMER: Bruce Simon

ORIGINATOR: Carl Bowin

DATE: 1 October 1975


NAME: G3DCPREP

TYPE: Main Program

PURPOSE: To prepare input to program G3DCP

MACHINE: Sigma 7

SOURCE LANGUAGE: Extended Fortran IV

PROGRAM CATEGORY: File management

DESCRIPTION: G3DCPREP prepares the input to the modeling program G3DC from LSORT output.

INPUT:

Cards
Card 1: Number of bodies to be input (NUMBOD;I2)
Cards 2-6: A reference density for each body to be subtracted from the density in the LSORT output (REFD;F10.0)

Files
Files 1-6: 1 file for each of up to 6 bodies to be input to G3DC
File 13: file with laminae count for each body input

OUTPUT:

Files
File 7: a file compatible with G3DC to be used as input to G3DC

USAGE: See G3DCP documentation

SUBROUTINES REQUIRED: Fortran IV Library

PROGRAMMER: Lee Gove

ORIGINATOR: Lee Gove

DATE: 1 October 1975
<table>
<thead>
<tr>
<th></th>
<th>2+80,00</th>
<th>3+95</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+99996</td>
<td>2000</td>
<td>00</td>
</tr>
<tr>
<td>2</td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+80,00</td>
<td>3+24</td>
<td>00</td>
</tr>
<tr>
<td>3</td>
<td>2+8000</td>
<td>3+44</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+44</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+44</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+44</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+44</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+44</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+44</td>
<td>00</td>
</tr>
<tr>
<td>4</td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td>5</td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td>6</td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td>7</td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td>8</td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td>9</td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+63</td>
<td>00</td>
</tr>
<tr>
<td>10</td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td>11</td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
<tr>
<td></td>
<td>2+8000</td>
<td>3+60</td>
<td>00</td>
</tr>
</tbody>
</table>
NAME: HIG

TYPE: Main Program

PURPOSE: To convert gravity data in the format used by the Hawaii Institute of Geophysics (HIG) to GSUM format.

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:
HIG simply reads a record in H.I.G. format and converts the record to a GSUM record with format forward code of 17

INPUT:
Unblocked tape in HIG format on device no. 1

CARDS:
1) Sense switch options - all zero (or blank)
2) Source code (I5)
3) Input tape serial number (4A1)
4) EITP in columns 1 to 4
5) Output tape serial number (4A1)
6) EOTP in columns 1 to 4

OUTPUT:
Unblocked tape in GSUM format on device no. 2

USAGE:
!JOB
!LIMIT
!MESSAGE (I/O mag tape info)
!ASSIGN F:1,(DEVICE, 9T), (SN,xxxx), (IN), (TRIES,10)
!ASSIGN F:2, (DEVICE,9T), (SN,xxxx), (OUT), (TRIES, 10)
!OLAY
!RUN
!DATA
Data Cards
!EOD
RESTRICTIONS:

SUBROUTINES REQUIRED: GINOT STAT FORTRAV IV library

OPERATIONAL CHARACTERISTICS: Simple read-then-write program

PROGRAMMER: Lee Gove

ORIGINATOR: Carl Bowin

DATE: 1 December 1973
NAME: LSORT

TYPE: Main Program

PURPOSE: To edit and sort polygonal laminae of G3DC format

MACHINE: Sigma-7

SOURCE LANGUAGE: Extended Fortran IV

PROGRAM CATEGORY: File Management

DESCRIPTION: LSORT reads, for each of up to 6 bodies, up to twenty laminae. It counts them, discards laminae of thickness less than ZLIM, and prepares files for input to G3DCPREP.

INPUT:

Cards

Card 1: Minimum allowable thickness for a single laminae (ZLIM;F10.0)

Card 2: Number of bodies (files) to be input (NUMBOD;I2)

Files

Files 1-6: 1 file for each of up to 6 bodies

File 13: file of counts of laminae for each body

USAGE: See G3DCP documentation

SUBROUTINES REQUIRED: FORTRAN IV Library

PROGRAMMER: Lee Gove

ORIGINATOR: Lee Gove

DATE: 1 October 1975
NAME: MODPLOT

TYPE: Main Program

PURPOSE: Plots data for preparation of crustal structure models of the earth's crust and plots the output tapes from TALPLOT16

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Graphical display

DESCRIPTION:
This program performs two basic functions. It will plot a profile of data used in the preparation of two-dimensional structure models of the earth's crust as well as plotting the structural models themselves.

Input to this program may include output from a TALPLOT16 run and/or combinations of data output from PROJ4 and/or SAINT2 runs and model polygons. TALPLOT16 is a program that computes the gravitational attraction of two-dimensional structure models. The PROJ4 program projects data from any given area to a selected straight line and outputs a card deck which is then used to plot profiles of the data. The SAINT2 program will interpolate the data output from PROJ4 at regularly spaced intervals (in order to reduce the effects of small local variations) and output data that can also be plotted as profiles.

At the present time, the types of data that the PROJ4 program will process are: GSUM format (containing free-air and Bouguer gravity data, bathymetry and elevation), SPFMT format (containing seismic refraction data), seismicity data, and the model polygon portion of TALPLOT16 input data.

Program operation is determined by sense switch options selected via the first data card and the order in which the various types of data are arranged in the input deck. The JFMT number (input card 7) informs the program what kind of data follows in the input deck. The data terminator cards (input card 9) indicate completion of input of present JFMT type data. Another JFMT card is then read in. If it equals 9, then the job is terminated.
INPUT:

**Card 1:** Values of SSW(I). Format (80I1) See SSW options under USAGE

**Card 2:** XFACT, YFACT, TOP, BOT, BLEFT, RIGT (6F10.0)

- **XFACT** - The number of km's/inch in the x-direction (long axis of plot).
- **YFACT** - The number of km's/inch in the y-direction.
- **TOP** - The upper bound for the model (km). Elevation above sea level is negative, depth below sea level is positive. To avoid boundary effects, the value of TOP should be more negative than the y-coordinate of the point of highest elevation to be plotted.
- **BOT** - The lower bound for the model (km). The value of BOT should be a greater positive number than the base of the model, to avoid truncating the bottom of the model.
- **BLEFT** - The left boundary (km) of the data and model to be plotted.
- **RIGT** - The right boundary (km) of the data and model to be plotted.

**Card 3:** ELFAC, ELDIS, GFAC, GDIS, WPAC, WDIS, PFAC, PDIS (8F10.0)

- **ELFAC** - The scale factor for elevation (km/in).
- **ELDIS** - The distance of the origin of the elevation profile above the origin of the model (inches).
- **GFAC** - The scale factor for gravity; free-air and Bouguer (mgal/in).
- **GDIS** - The distance of the origin of the plot of free-air and/or Bouguer gravity data above the origin of the structure model (inches).
- **WPAC** - The scale factor for the model weight profile (kg/in).
- **WDIS** - The distance of the origin of the weight profile below the base of the model.
- **PFAC** - The scale factor for individual polygon contributions (mg/in).
- **PDIS** - Distance of the origin of the curves (contribution of individual polygons) above the origin of the structure model.
Card 4: HT, DBOT (2F10.0)

HT - Character height multiplication factor in multiples of 0.07" (used in the call to symbol for the plotting of the anomaly curves). If HT = 0 a default value of 3 is assumed. (0.21”).

DBOT - The distance that the weight curve is supposed to be plotted above the bottom of the plot.

Omit the following cards when plotting only the output from TALPLOT16.

Card 5: A label card containing 80 columns of alphanumeric data. This will appear at the beginning of the plot. Include this label card only if SSW(1) = 1. If SSW(1) = 0 or 2, then the label is obtained from the TALPLOT16 output tape.

Card 6: Crustal structure section (CSS) identification card (same card as used in the PROJ4 run).

CSS ID, ANG, DMAXM, ILAT, RILTM, ILONG, RILDM, LABEL (8A1, 2X, 2F10.0, I4, F6.2, I4, F6.2, 30A1)

CSS ID - Crustal structure section identification number (CSS-NNN).

ANG - The angle between the vertical and the straight line to which the data has been projected in the PROJ4 program.

DMAXM - The greatest distance (km) from the reference point for which data will be accepted.

ILAT - The latitude, in degrees, of the reference point for the line.

RILTM - The latitude, in decimal degrees, of the reference point.

ILONG - The longitude, in degrees, of the reference point.

RILDM - The longitude, in decimal degrees, of the reference point.

LABEL - A label containing up to 30 alpha-numeric characters.
MODPLOT, page 4

Card 7: JFMT (J-format) (II)

<table>
<thead>
<tr>
<th>JFMT</th>
<th>For</th>
<th>JFMT Data Terminator (II)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (or 2)</td>
<td>GSUM</td>
<td>2 - &quot;8&quot; cards (column 1)</td>
</tr>
<tr>
<td>3</td>
<td>SPFMT</td>
<td>2 - &quot;8&quot; cards</td>
</tr>
<tr>
<td>4</td>
<td>SEISMICITY</td>
<td>1 - &quot;8&quot; card</td>
</tr>
<tr>
<td>5</td>
<td>MODEL POLYGONS</td>
<td>none</td>
</tr>
<tr>
<td>6</td>
<td>TALPLOT16 INPUT</td>
<td>none</td>
</tr>
</tbody>
</table>

Note: 5 and 6 both read TALPLOT16 input. JFMT=6 reads from card 1 of TALPLOT16. JFMT=5 reads from after card 5 of TALPLOT16. In either case, only the model polygons will be plotted.

A. If input is on cards:

Card 8: data cards for the JFMT
Card 9: data terminator cards (see card 7)

B. If input created by the PROJ4 program is on mag tape, and JFMT = 1 or 2 for GSUM format:

Card 8: input tape serial number (I4)

C. If input created by the PROJ4 program is on tape, but is some format other than GSUM, there will be no cards 8 or 9. Data termination cards are already on the mag tape.

Repeat cards 7 - 9 for each JFMT data type to be plotted.

Card 10: JFMT = 9 to terminate the job (II)

OUTPUT

1) on line printer: the input parameters
2) on Calcomp plotter: profiles of the data and/or two-dimensional plot model input to the program.

USAGE:

Sense Switch Settings

SSW(1) = 0 to plot only from TALPLOT16 output tape
        = 1 to plot PROJ4 data, and/or model polygons: reads JFMT.
        = 2 to plot both TALPLOT16 output tape and PROJ4 data
SSW(2) = 1 to plot Bouguer anomaly in addition to the free-air
SSW(3) = 1 to plot height from GSUM data
SSW(4) = 1 to plot elevation. The input values are on the
         TALPLOT16 output tape (in meters). Use only if
elevation data was input to previous TALPLOT16 run.
SSW(5) = 1 to plot observed and calculated gravity.
SSW(6) = 1 to plot weight
SSW(7) = 1 to plot contribution of each polygon.
SSW(8) = 1 to plot two-dimensional Bouguer anomaly in TALPLOT16 output.
SSW(13) = 1 to print intermediate values.
SSW(14) = 1 to plot only a dot for GSUM free-air values, rather than a continuous line.
Be sure that SSW(14) = 1 if plotting from unsorted PROJ4 output.
SSW(30) = 1 to read GSUM data on 2 cards
SSW(32) = 1 to read SPFMT data on 2 cards
SSW(36) = 1 to read Seismicity data from cards
SSW(44) = 1 to read Seismicity data in blocked format
The calculated anomaly plot is marked by asterisks.

RESTRICTIONS:
Elevation above sea level is negative, depth below sea level is positive.

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: ANOV3, DISAZ, DMTOR, DNAV, EVIL, EXIT, EXTD, GINOT, INCEP, ISW, PINOT, PLOT, PLOTS, SPLOT, SPOT, STAT, SYMBOL, WHERE, YINOT, STANDARD FORTRAN IV LIBRARY.

TIMING: TALPLOT16 output of approximately 200 polygon points will take just under two minutes of computer time. For other data types, the timing is undetermined.

ERRORS AND DIAGNOSTICS:

Message                           Error
"Plot too wide, width = VIT"      Distance of model origin above the bottom (right) of paper is too great. The value of VIT must be less than 29. (VIT = DBOT + WDIS + YWIDE/ YFACT) where (YWIDE = TOP-BOT)

PROGRAMMER: Carl O. Bowin

ORIGINATOR: Carl O. Bowin

DATE: 12 June 1974
NAME:  NOAA

TYPE:  Main Program

PURPOSE:  To convert gravity data in NOAA format to WHOI GSUM format

MACHINE:  Sigma-7

SOURCE LANGUAGE:  Fortran IV

PROGRAM CATEGORY:  Format Conversion

DESCRIPTION:

NOAA uses asychronous I/O to speed up the conversion of NOAA data to GSUM. The GSUM record has a format forward code of 16. The input tape is blocked by 20 and the output is blocked by 50. Input gravity data is assumed to be referenced to International Gravity Formula 1930. Conversion to IGF 1967 is made in another program.

INPUT:

1) Mag-tape in NOAA format blocked by 20
2) Cards
   1) Source Code(I5)

OUTPUT:

1) Mag tape in WHOI GSUM format blocked by 50

USAGE:

!JOB
!MESSAGE (Mag tape info)
!ASSIGN F:1,(DEVICE,9T),(SN,xxxx),(IN),(TRIES,10)
!ASSIGN F:2,(DEVICE,9T),(SN,xxxx),(OUT),(TRIES,10)
!OLAY
!RUN
!DATA
   Data card
!EOD

RESTRICTIONS:

a) NOAA tape must be blocked by 20
b) GSUM tape blocked by 50 with format forward code = 16
NAME: PROFG

TYPE: Main Program

PURPOSE: Profiles GSUM data

MACHINE: XDS Sigma 7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Graphical Display

DESCRIPTION:

This program profiles various parameters in GSUM format. By using GINOT as the input routine all the options in GINOT are available.

INPUT:

1: GSUM data (via F:1)
   GSUM data on tape or cards, blocked or unblocked

2) Cards:

   Card 1  Sense switch card
   SSW(5) = 0 no effect
           = 1 to select station
   SSW(14) = 0 no effect
            = 1 process only if data after start date
   SSW(25) = 0 no effect
            = 1 to call MOUNT
   SSW(29) = 0 no effect
            = 1 process only specified source codes
            = 2 skip specified source codes
            = 3 process only specified format-forward codes
            = 4 skip specified format-forward codes
   SSW(30) = 0 no effect
            = 1 input GSUM on cards
   SSW(40) = 0 input unblocked with no DLT
            = 1 input blocked X50 with no DLT
            = 2 input blocked X50 with DLT
   SSW(41) = 0 output on tape unblocked
            = 1 output on tape blocked X50
   SSW(46) = 0 no effect
            = 1 process only within specified geographic bounds
   SSW(47) = 0 no effect
            = 1 process only within specified time interval
            = 2 skip data within specified time interval
PROFG continued

INPUT continued

Card 2  GINOT initialization card
Card 3  Number of nautical miles, kilometers, or hours per inch on plot (DIFAC;F10.0)
        Engineering units per inch in Y direction (YFAC;F10.0)
        Number of points between time annotation (LCNT;I5)
        0 For nautical miles, 1 for kilometers (MIKM;I5)
        Number of plots (NPLLOT;F5)
        Number of files (NFILE;I5)

Card 4  Upper limit for plotting data value in eng units (ULIM;F10.0)
        Lower limit for plotting data value in eng. units (BLIM;F10.0)
        Distance limit in inches for plotting data (DLIM;F10.0)
        1 for distance along track  0 for time along track
        (IXDIR; I5)

Card 5  Allowable distance for length of plot before reinitialization
        (XALOW; F10.0)
        Inches to move before reinitialization (DMOVE;F10.0)

Card 6  Starting day for processing (LIMDA; I5)
        Starting month for processing (LIMMO; I5)
        Starting year for processing (LIMYR; I5)
        Starting hour and minute for processing (LIMHM; I5)

Card 7  Station number to be selected (MSTA; I5)

OUTPUT:

1) Printer (via F:108)
   All input parameters are output to the line printer along
   with error messages
2) Plot tape (via F:PLOT)
   A plot tape for plotting by the 30" Calcomp plotter

USAGE:

!JOB
!LIMIT (9T,1), (7T,1), (CORE, 20), (TIME, X)
!MESSAGE GSUM data tape info
!MESSAGE plot tape info
!ASSIGN F:1 to GSUM file
!ASSIGN F:PLOT, (DEVICE, 7T), (SN, PLT1), (BIN), (UNPACK), (TRIES, 10)
!LOAD (EF, (PROFG, 456), (STAT, 3)), (UNSAT, (305), (312), (3))
!RUN
!DATA
!EOD

RESTRICTIONS: None
PROFG continued

STORAGE REQUIREMENTS: 15K

SUBPROGRAMS REQUIRED: GINOT, FIND, ISW, CDATE, MCVOL, SPOT

TIMING: Undetermined

ERRORS AND DIAGNOSTICS

    DLIM (date) distance limit exceeded
    ULIM (date) upper limit exceeded
    BLIM (date) bottom limit exceeded
    END DATE PASSED end of processing

PROGRAMMER: Carl Bowin and Lee Gove

ORIGINATOR: Carl Bowin

DATE: 2 October 1975
NAME: PROJ4

TYPE: Main Program

PURPOSE: To project data onto a given line providing the data is within a given area and within a given distance from the line.

MACHINE: Sigma-7

SOURCE LANGUAGE: FORTRAN IV

PROGRAM CATEGORY: Data analysis

DESCRIPTION: This program projects several data formats to a straight line defined by input latitude and longitude of a point, and bearing from the north (0 to 360 degrees). The program can process several different formats in the same run, the order for processing being determined by a format choice input card (card 6).

The data can be output to a tape to be used in MODPLOT, TALPLOT16 and/or SAINT 2 runs when JFMT = 1 or 2 (GSUM format).

The different data formats that the program will process are:

- GSUM - Gravity summary format, containing free-air and Bouguer gravity values, elevation and bathymetry.
- SEAG1 or 2 - Sea gravity data format, containing corrected gravity values plus magnetics. SEAG data may be input in 1939 IGF and referenced to Potsdam (SEAG1), or in 1967 IGF and referenced to the IGSN71 (SEAG2). GSUM data may be input in 1939 IGF and referenced to Potsdam (IREC=1), or in 1967 IGF and referenced to the IGSN71 (IREC=2). For either SEAG or GSUM input, output is GSUM with IREC=2.
- SPFMT - Seismic refraction data format.
- SEISMICITY - U.S. Coast and Geodetic Survey data format.
- ACTIVE VOLCANOES - File on active volcanoes key-punched from "Catalogue of the Active Volcanoes of the World" and supplementary references.
INPUT:

Card 1: Crustal structure section (CSS) identification card containing projection parameters.

CSS ID, ANG, DMAXM, ILAT, RILTM, ILONG, RILDM, LABEL
(1G1, 2F10.0, I4, F6.2, I4, F6.2, 30A1)

CSS ID - Crustal structure section identification number (CSS-NNN).

ANG - The angle between the vertical and the straight line to which the data will be projected in the PROJ4 program (0 to 360 degrees).

DMAXM - The greatest distance in nautical miles from the reference point for which data will be accepted.

ILAT - The latitude, in degrees, of the reference point for the line. South is negative.

RILTM - The latitude, in decimal minutes, of the reference point. Note: if degrees are negative minutes must be negative also.

ILONG - The longitude, in degrees, of the reference point. West is negative.

RILDM - The longitude, in decimal minutes, of the reference point. Note: if degrees are negative minutes must be negative also.

LABEL - A label containing up to 30 alpha-numeric characters.

Cards 2-5: Four boundary cards (I4, F6.2)

2) Degrees Minutes (top latitude)
3) " " (bottom latitude)
4) " " (left longitude)
5) " " (right longitude)

Card 6: Selection of order that data formats are processed:
(9I5) 1st JFMT, 2nd JFMT, 3rd JFMT, ..., 9th JFMT.

JFMT = 1 for GSUM format.
JFMT = 2 for SEAG1 or 2 input, output is GSUM format with IREC=2.
JFMT = 3 for SPFMT format.
JFMT = 4 for SEISMICITY data format.
JFMT = 7 for Active Volcanoes.
Card 7 and following vary depending on the JFMT. Repeat cards 7 and greater for each selected format in the order given on card 6. The program terminates when JFMT = 0 or blank, or when start date = 99.

A. for JFMT = 1  
GSUM format  
(uses subroutine GINOT)

Card 7: Sense switch options. Put sense switch (0) option in column 80, all others in column corresponding to switch number.
ISW (3) = 1 to output GSUM data onto a disk file (unit 20).
ISW (4) = 1 to output intermediate values for checking (subroutine PROJ).
ISW (12) = 1 to list date identification.
ISW (26) = 1 to output on high speed printer only.
ISW (27) = 1 to suppress rewind of ITAPE at start of job.
ISW (27) = 2 to suppress rewind of JTAPE at start of job.
ISW (27) = 9 to suppress rewind of both ITAPE and JTAPE.
ISW (29) = 1 to read and test for selected source code numbers to be processed.
ISW (29) = 2 to read and test for selected source code numbers to be skipped.
ISW (30) = 1 for input data on cards.
ISW (31) = 1 to output data on cards.
ISW (40) = 0 to process without the DLT.
ISW (40) = 1 to process with bounds using the data location table (DLT).
ISW (60) = 1 to process only data with IFFC = 4, ABSTRACTOR output.
ISW (61) = 1 to replace free-air, Bouguer, elevation, latitude and longitude with averaged values.

Card 8: (optional)  
If ISW (29) = 1 or 2, enter ISRC numbers to be processed or skipped (16I5).

Card 9: (optional)  
If data is input on tape, serial number of input tapes – one per card (A4). If data is not input on tape, ISW (30) must equal 1.
Card 10: (optional) If there are card(s) 9, EITP in columns 1-4.

Card 11: (optional) If data is to be output to mag tapes, serial number of output magnetic tapes, one per card (A4). If no output tapes, then no serial number cards and either ISW (31) or ISW (3) must equal 1.

Card 12: (optional) If there are card(s) 11, EOTP in columns 1-4. If no output tapes, then no Card 12.

Card 13: Start and end dates, ISKP card

ISTDA, ISTMO, ISTYR, ISTHM, IENDA, IENMO, IENYR, IENHM,
ISKP (312, I4, 5X, 312, I4, 5X, 315)
Start date for processing
ISTDA (day)
ISTMO (month)
ISTYR (year)
ISTHM (time of day)
End date for processing
IENDA (day)
IENMO (month)
IENYR (year)
IENHM (time of day)

ISKP Number of records to be skipped at the start of the job, is much faster than using start date only.

Use a blank card if start and end dates not used. If this card is blank, plotting begins with the first record.

B. for JFMT = 2

SEAG1 or 2, format input, GSUM output with \texttt{INEC=2} (uses SINOT and GINOT)

Card 7: Sense switch options. Same as for JFMT = 1.

C. for JFMT = 3

SPFMT format (uses subroutine PINOT)

Card 7: Sense switch options.
ISW (4) = 1 to output intermediate values for checking (subroutine PROJ).
ISW (26) = 1 to output to line printer.
ISW (32) = 1 to read SPFMT data on two cards.
ISW (33) = 1 to write SPFMT data on two cards.
Card 8: Start and end dates and ISKP. (Same as Card 13 on JFMT = 1 above).

b. for JFMT = 4 SEISMICITY format (uses subroutine YINOT)

Card 7: Sense switch options.
ISW (4) = 1 to output intermediate values for checking (subroutine pros).
ISW (26) = 1 to output to line printer.
ISW (44) = 1 to read blocked input.
ISW (45) = 1 to write blocked output

Card 8: Start and end dates and ISKP (Same as card 13 on JFMT = 1 above).

E. for JFMT = 7
at present there is no volcano routine.

OUTPUT:

On unit 2:
1) JFMT number record (Il, 20X).
2) projected data at appropriate format.
3) two records with a numeral '8' in column 1 (Il, 20x). Only one '8' record for seismicity and volcano data.
This sequence is repeated for each selected JFMT.

On unit 20 (disk): GSUM data for subsequent processing by SAINT 2 if SSW(3) = 1 (Only applies when JFMT = 1 or 2).
On unit 108 (line printer): initialization parameters.

USAGE: Assign cards vary depending on format. There should be a separate assign card for each JFMT input.
Unit 2 is for output of projected data
Unit 3 is for GSUM format input
Unit 4 is for SEAG1 format input; output is GSUM with IREC=2
Unit 5 is for SPFMT format input
Unit 6 is for SEISMICITY format input
Unit 9 is for Active Volcano data input
Unit 20 is for disk storage of GSUM data if ISW(3) = 1.

To stop processing make start day = 99. If start day = 99, the program goes to 1000 and checks input format choices for format code number of zero.
RESTRICTIONS: South and west are negative.
Degrees and minutes must have the same sign
(e.g. 47°45.0' or -32° - 30.0').

STORAGE REQUIREMENTS: Unknown.

SUBPROGRAMS REQUIRED: AREAK, DNAV, ENDO, ENDLT, EVIL, EXIT, FIND, GINOT,
ISW, MCVOL, MOUNT, NAVIN, NAVOT, OBG, PINOT, PKBY,
PROJ, SETSKP, SINOT, SKPRE, STAT, TODAY, UNPKBY,
YBLKI, YBLKO, YINOT.

TIMING: Undetermined.

ERRORS AND DIAGNOSTICS:

MESSAGE
Error in SKPRE, IND = 'IND'

ERROR
number of records to be skipped = 0.

PROGRAMMER: Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 24 February 1975
NAME: RETRIEVE

TYPE: XDS Manage Processor Program

PURPOSE: To extract data from a data base file according to user specified search criteria.

DESCRIPTION:
Retrieves data from the data files using a selective criterion applied to any part of the data record as described in the dictionary. This selection may involve a secondary file called "matchine file" to which the main data file is compared. The Manage Reference Manual gives detailed information on the use of the program.

USE:

There is a tape containing information on world seismicity (hypocenter data) from 1900 through October 1974, which is available for general use. The tape will be updated periodically, and will always be assigned the same number. The serial number if #L20.

Because of the amount of data on this tape (about 90,000 earthquakes), we recommend you retrieve the data in the area you wish to plot, using the RETRIEVE processor, before charting. A sample RETRIEVE run is included here. Input bounds are in order TOP, BOTTOM, LEFT, RIGHT. South and west are negative. For more details see the XEROX MANAGE processor manual. If a listing of the data retrieved is desired, it can be dumped using FTDUMP. For details see the Handbook for Computer Users.

The program CHART can give you plots annotated with date, depth in km, or magnitude; or as in the sample run, it can plot symbols whose type and size varies with depth and magnitude. For additional details, see the DDL Documentation for CHART.
Because of the way this program 'finds' data: what you see is not what you get.

For example, suppose you are trying to retrieve all data within the bounds 47, 45, -118, -116: that is, all data in squares marked X.

The retrieve program, however, uses your input bounds to calculate keys to match against a key in the data on the input tape. The sort keys are created by adding 90 to the latitude and 180 to the longitude, in order to insure positive numbers for the codes for all points of the earth. The sort key for latitude 47 would be 137, but this code refers to data in the degree square to the north of the given latitude line. Similarly, the sort key for longitude -118 would be 62, but this number refers to the area of the degree square to the east of the given longitude line. Thus, in the example given, the code 137062 refers to the square marked Y₁.

If your input bounds are 47, 45, -118, -116, the data that will find its way to your output is all data in squares marked X and all data in squares marked Y.

Therefore, you must be aware of this quirk and learn to live with it, or you must hedge on your input bounds. To get only that data in squares marked X, your input bounds would be 46, 45, -118, -117.
JOB NAME, LABEL
LIMIT (TIME, 10), (CORE, 30), (GT, 2)
MESSAGE XXXX ON GT, NEW, **ADJUST**
MESSAGE @120 ON GT
ASSIGN FILE (FILE, RETBO), (OUT), (SAVE)
LOAD (EF, (FILE, 514), (UNSAT), (3))
RUN
DATA
ASSIGN F:SORTIN (FILE, RETBO), (IN)
ASSIGN F:SORTOUT (FILE, MATCH), (OUT)
ISORT
U000001001
A00000101
ASSIGN M:EI (FILE, MATCH), (IN)
IFTRUMP HEAD
***MATCHING FILE FOR RETRIEVE FROM BOUNDS***
ASSIGN F:STRFILE, (DEVICE, GT), (INS), ($20), (IN)
ASSIGN F:NONREPT, (DEVICE, GT), (OUTSN, XXXX), (OUT)
ASSIGN F:SLCFILE, (FILE, MATCH), (IN)
ASSIGN F:DICT, (FILE, DICT, 456), (IN)
RETRIEVE
SE 1 A CGSDATA MATCH S U E
SE 1 C 01 LT10KEY EQ MALAT10 AND
SE 1 C 02 LTKEY EQ MALA11 AND
SE 1 C 03 LG10KEY EQ MALON10 AND
SE 1 C 04 LGKEY EQ MALON1 END
IPCL
DELETE RETBO
END

00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
11111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111
22222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222222
33333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333
44444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444444
55555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555555
66666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666666
77777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777777
88888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888
99999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999999
NAME: SAINT2
TYPE: Main Program
PURPOSE: To interpolate data at even intervals
MACHINE: Sigma-7
SOURCE LANGUAGE: Fortran IV
PROGRAM CATEGORY: Data Analysis

DESCRIPTION:

This program operates on GSUM format output of PROJ4, a projection program. SAINT2 will sort the data by distance along the projected line, if necessary, and then interpolate the values at even intervals of distance, if desired. Interpolation is accomplished by filtering with a polynomial function, certain parameters of which are entered at run time.

The phrase "interpolated GSUM from SAINT2" is entered in the free-field portion of the GSUM output record.

INPUT:

Card 1: Sense switch values (put a "1" in column corresponding to switch number of desired option)

ISW(1) = 1 to sort data
ISW(2) = 1 to punch sorted data
ISW(3) = 1 to print sorted data
ISW(4) = 1 to interpolate data
ISW(5) = 1 to output interpolated data in GSUM format (see ISW(11))
ISW(6) = 1 to punch interpolated free-air anomaly in format for TALPLOT
ISW(7) = 1 to print interpolated data and summary of job.
ISW(8) = 1 to print input data
ISW(9) = 1 to punch interpolated elevation data in format for TALPLOT
ISW(10) = 1 to read GSUM data from magnetic tape
ISW(11) = 0 to punch interpolated GSUM on cards (two cards per record)
           = 1 to write interpolated GSUM on magnetic tape
Card 2: XKM, XSCAL, NPTS (F5.1, F5.1, I5)

XKM = distance between interpolated points in kilometers.
XSCAL = maximum distance in kilometers for which points are given unit weight in interpolation algorithm. Beyond this weight drops as X/XSCAL. Good first approximation: XSCAL = 3*XKM.
NPTS = number of points on either side of output point to be used in interpolation filtering function. Default value is 4.

Card 3: Crustal structure section (CSS) identification card (same card as used in PROJ4 run).

CSS ID, ANG, DMAXM, ILAT, RILTM, ILCNG, RILDM, LABEL (10A1, 2F10.0, I4, F6.2, I4, F6.2, 30A1)

CSS ID = crustal structure section identification number
ANG = the angle between the vertical and the straight line to which the data has been projected.
DMAXM = the greatest distance from the reference point for which data will be accepted, in nautical miles.
ILAT = the latitude, in degrees, of the reference point for the line. North is positive.
RILTM = the latitude, in decimal minutes, of the reference point (note: if degrees are negative, minutes must be negative, too).
ILONG = the longitude, in degrees, of the reference point. West is negative.
RILDM = the longitude, in decimal minutes, of the reference point (see note Re: RILTM).
LABEL = a label containing up to 30 alpha-numeric characters.

Card(s) 4: If ISW(10) is not set, GSUM format input data cards (output of PROJ4).

Card 5: If data is on cards, a terminator is needed. Either a !EOD card or two cards with a '9' in column 1.
OUTPUT:

On line printer: input parameters and listings controlled by sense switches.

On card punch: sorted data, GSUM format, TALPLOT elevation input, and/or TALPLOT free-air anomaly input, as requested.

On magnetic tape: Interpolated GSUM format, if desired.

USAGE:

Assign F:1 to input device (either magnetic tape or card reader)
Assign F:2 to final output tape.
Assign F:3 to RAD output file for sort
Assign F:4 to RAD input file for sort
   (These must have the same file name. SAINT2 writes the file and then sorts it.)
Assign F:5 to RAD output file for output of sort. (OUT) should be on the assign card. It will be changed to (IN) by the sorter, so that SAINT2 can read it in.
Here are sample assign cards:

ASSIGN F1,(DEVICE,S1)
ASSIGN F2,(DEVICE,9T),(OUT),(SN,GN30),(TRIES,10)
ASSIGN F3,(FILE,FILE1),(OUT),(SAVE)
ASSIGN F4, (FILE,FILE1),(IN),(SAVE)
ASSIGN F5,(FILE,FILE3),(OUT),(SAVE)

RESTRICTIONS: Maximum number of input records read is 999. The number of interpolated records will be ≤ 1000.

STORAGE REQUIREMENTS: 1575 decimal words. Requires (CORE,16) on limit card.

SUBPROGRAMS REQUIRED: AMAW, AMIW, ATSM, CLOFIL, DISAZ, DMTOR, EQN, OPIN, PICK, SETAL, THORT, WT, WTSET

TIMING: less than 3 minutes for under 150 records input, 200 records output, with two listing options selected.

ERRORS AND DIAGNOSTICS: NONE

PROGRAMMERS: Jack Wolfe, Carl Bowin, Allin Folinsbee

ORIGINATOR: Carl Bowin

DATE: 2 October 1974
NAME: SELSP

TYPE: Main Program

PURPOSE: Outputs data selected according to any of several criteria

MACHIN?: XDS Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Data selection

DESCRIPTION: Program CRWT3 (which see) outputs University of Toronto World Seismic Refraction Compilation in W.H.O.I. SPFMT format. SELSP will select from among this output those records satisfying the required criteria, such as quality, geographic area, geologic province, depth to mantle, or water depth. Records which satisfy the criteria specified on the input cards are output in SPFMT format. A record key of 9 causes that record to be skipped.

INPUT:

Card 1: Sense switch options (80 Il) Put sense switch 0 option in column 80, others in column corresponding to switch number.

ISW(26) = 1, to output on line printer only (sub. PINOT)
ISW(32) = 1, to read data from two cards per record (sub. PINOT)
ISW(33) = 1, to write data onto two cards per record (sub. PINOT)

Card 2: (6I5, 2F 10.0)

JMET = 1 to sort on the basis of quality
IMET = 1 degree of quality required
        = 2 for less-reliable interpretation
        = 3 for unreliable interpretation
JPROV = 1 to sort on the basis of province
IPROV = is the province number to be selected (see references)
IAR = 1 to sort on the basis of area bounds
IDMP = 1 to sort on the basis of mantle depth
SMIND Minimum depth to mantle in kilometers
SMAXD Maximum depth to mantle in kilometers
INPUT (continued)

Card 3:  (I5, 2F10.0)

IDP = 1  to sort on the basis of water depth
=-1  to sort on basis of elevation above sea level
(land data)

DMIN  minimum water depth or elevation in kilometers
DMAX  maximum water depth or elevation in kilometers
(All positive values)

Cards 4-7 (optional)
Area bounds (I5, F10.0)
If IAR=1, enter top bound, bottom, left and right, in integer degrees, and minutes with decimal point, one bound per card.

Data cards (optional):
If sense switch 32 is set to one, SPFMT cards should follow, two cards per record.

OUTPUT:

To line printer: input parameters and counters of records read and written

If ISW(26)= 1, selected output records

On output device: records which satisfy selection criteria, which may be output on cards or magnetic tape.

USAGE:

Assign F:1 to input device. Unless ISW(26) or ISW(33) = 1, assign F:2 to an output tape.

RESTRICTIONS:

STORAGE REQUIREMENTS: 468 decimal words; (CORE,9) is required on LIMIT card.

SUBPROGRAMS REQUIRED: ARCK, ARLIM, EXIT, PINOT

TIMING: 5 charge units to select 100 records from 2300 (output to lister).

ERRORS AND DIAGNOSTICS: None
REFERENCES:


PROGRAMMER: CARL BOWIN

ORIGINATOR: CARL BOWIN

DATE: 6 August 1975
NAME: SPFMT

TYPE: Main Program

PURPOSE: Converts seismic refraction data in University of Toronto World Seismic Refraction Compilation format to W.H.O.I. SPFMT format

MACHINE: Sigma-7

SOURCE LANGUAGE: Fortran IV

PROGRAM CATEGORY: Format conversion

DESCRIPTION:

SPFMT converts seismic refraction column data in University of Toronto World Seismic Refraction Compilation format (one or two cards per profile) to W.H.O.I. SPFMT format (each 120 character output record is one profile). Input data is assumed to be on cards.

INPUT:

- card 1: Sense switches (blank card)
  
  No sense switches have been implemented for this program, available for possible future use.

- card 2: data cards

- card 3: blank card to signal end of input data. Without this card, the last input card may not be output.

OUTPUT: On device no. 2 - seismic refraction column data in SPFMT format.

USAGE: Assign F:1 to input device; F:2 to output device

RESTRICTIONS: NONE

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: EVIL, ISW, STAT

TIMING: Not determined
SPFMT

ERRORS AND DIAGNOSTICS: 'EDF FOUND, NREC =
NREC is the number of records output

PROGRAMMER: John Woodside, Carl Bowin

ORIGINATOR: Carl Bowin

DATE: 6 August 1975
NAME: TALPLOT 16
TYPE: Main program
PURPOSE: To compute gravity anomalies for a set of two-dimensional polygons
MACHINE: Sigma-7
SOURCE LANGUAGE: Fortran IV
PROGRAM CATEGORY: Data analysis

DESCRIPTION:

This program uses Talwani's method for calculating the gravity anomaly for two-dimensional polygons. It sums the contributions from the set of polygons and compares the calculated result to the observed gravity which is read in at the start of the program. The calculated gravity is referenced to the observed gravity at a specified point by subtracting the value of \((g_{\text{calculated}} - g_{\text{observed}})\) at specified point from the calculated values at each point.

The program calculates the RMS difference between the calculated and observed gravity anomaly. If desired the program will modify specified points of the last polygon that is read in so as to reduce the RMS difference. This is done in the following way. For each point in the polygon that is to be modified, the effect of a change of \(z\) for this point on the gravity observed at each field point is calculated. This is done by computing the effect of thin triangles on the gravity at the field points.

The if we have \(N_{\text{FIELD}}\) field points and \(N_{\text{VAR}}\) variable polypoints (note that \(N_{\text{VAR}} \leq N_{\text{FIELD}}\)) we end up with a set of over-determined equations for the change in depth of each of the variable polypoints. These \(N_{\text{FIELD}}\) equations are reduced to \(N_{\text{VAR}}\) equations by the standard summation method.
DESCRIPTION (continued)

The equations are solved, and the appropriate modifications are made to each of the polypoints. The new gravity contribution of the polygon is calculated, along with the RMS error. We then go back and calculate a new set of variational parameters, and then another polygon model, and so on. This loop continues until one or the other of the following two things happens: 1. the number of new polygons calculated exceeds IMAX or, 2. the RMS error decreases by less than 0.5 mgals. When this happens the program prints out the new final polygon, prints out some graphs of the observed and calculated gravity, and then stops. It also calculates the crustal weight for a 1 cm² column down to the base of the model. If the new polygon displaces more than one polygon, above or below, the gravity and mass calculation will be incorrect.

INPUT:

First Card:

A card containing 80 columns of alphanumeric data. This will appear at the start of the job and at start of plot. Format (20A4)

Second Card: Sense Switches (80I1)

Put ISW(0) option in column 80 and the other options in columns corresponding to the switch number.

Sense Switch Settings

ISW(1) = 0 to write onto unit 2
= 1 not to write onto unit 2
ISW(2) = 0 sets elevation for all field points = 0
= 1 reads in elevation cards (kilometers)
= 2 reads in elevation cards (meters)
ISW(3) = 0 to print intermediate data for each polygon
= 1 does not print
ISW(4) = 0 to plot intermediate data for each polygon
= 1 does not plot
ISW(5) = 0 not to plot of elevation on final plot
= 1 to plot elevation on final plot
ISW(6) = 1 to adjust last polygon to fit gravity data
ISW(7) = 1 to print intermediate data for the fitting part of the program (debugging only)
INPUT (continued)

ISW(8) = 1 to disable the punching of new polygon points
ISW(9) = 1 NOT to write intermediate data for each polygon onto unit 2 (JTAPE)
ISW(12) = 1 to correct calculated weights with an elevation factor of (elev(km) * 2.67 * 100. Use when model does not contain that part of topography above sea level.
ISW(13) = 1 to print out results of input polygon before varying the boundary

Third Card: RDENS, RWGT, RHOD, REFX, FXI, DELFX, M, NFER, IMAX
(6F10.2, I10, 2I5)

RDENS - The reference density that is subtracted from the density of the polygons, (usually 0.0).
RWGT - The weight that is to be subtracted from the crustal weight calculated by the program. The value should be the normal weight of a normal column of material 1 cm² down to the bottom of the model in kg.
RHOD - The difference in density that is used by the model adjusting program. It is equal to the density of the layer below the adjustable boundary minus the density of the layer above the adjustable boundary.
REFX - The value of the X coordinate at which the calculated gravity is set equal to the observed gravity.
NFER - half length of the filter used to smooth the residuals before modifying the lower polygon boundary. If = 0 no smoothing is done.
FXI - The X coordinate for the first field point.
DELFX - Distance in Km between field points
M - The number of field points (must be .LE. to 200)
IMAX - The maximum number of times the boundaries of the last polygon will be adjusted (5 is a good number).

FOURTH Card: Observed gravity
The observed gravity values, five per card (5F10.1) (a total of M values). If the observed gravity is not known the value of the observed gravity should set to 990.

Fifth Card: (Optional: if ISW(2) = 1) elevations
Elevation of the field points in km, five per card (5F10.1) (a total of M values)
Then comes a group of cards that is repeated for each polygon

repeated for each polygon
LNO, RHORK (I5,F10.3)
XX,ZZ,ICODE,IAL (2F10.2, 2I1) a card like this for each polygpoint.

LNO - The number of the polygon
= 1 this is a water layer, and the program will calculate the 2-D Bouguer anomaly
= 99 this is the last polygon in the model
= any other number for ordinary polygon
RHORK - The density of the polygon (in gm/cc)
XX,ZZ - The X and Z coordinates of the polygon points. Note that Z is positive down and X is positive to the right (the coordinates are in km)
ICODE = 9 to indicate that this is the last point in the polygon.
= blank if not the last point.
Note that the last polygon point must have the same coordinates as the first polygon point, and that the polygon should be given in a clockwise order.
IAL - Used only by the boundary altering part of the program.
= 1 to indicate that this polygon point is to be varied. For stability the number of points to be varied should be less than 1/4 the number of observed gravity values, although the number of points can be equal to the number of observed gravity values if a perfect fit is desired. Also, all the points in a boundary should not be varies as this will lead to instability in the model. The final restriction is that the first or last point in a polygon must not be varied. If you wish to vary these points the polygon points should be reordered.
OUTPUT:

Onto URN 2 if ISW(1) .EQ. 0

a complete duplication of the input, along with the
results. This tape is used as input for the program
MODPLOT.

On line printer (108)

the input parameters, and numerical tabulations and
plots of the results. After each polygon the contri-
bution of that polygon is printed and plotted, if desired.

The values tabulated are:

K - the field point number
FX(K), FZ(K) - the coordinates of the field point
ANOMALY - the anomaly contribution in mgals., and a blank
column containing the weight contribution in kg.

After all polygons have been computed we get the RMS
difference between the observed and calculated gravity, ignoring
those points for which observed gravity is not known.

Then comes the number of points which are used in the
computation of the RMS value.

If the boundary is to be varied these values appear
for each iteration.

After the final new polygon has been determined the program
prints out the new polygon points.

Then comes the tabulation of the final results

K, FX(K), FZ(K), as above

ANOMALY - unreferenced gravity anomaly
CALC REF- the calculated value referenced to the
appropriate field point
RESIDUAL- the difference between the calculated reference
value and the observed value
OBS ANOMALY - observed gravity
WEIGHT - the weight of a cm² column down to the bottom of
the model
WGT DIFF - the weight - RWGT
WEIGHTTEST - this column will be a constant number provided
the bottom of the model is flat and there are no
holes or overlaps of the polygon. If this column
is not constant then one of the above conditions
has occurred, and there is probably an error in one
or more of the polygons.
OUTPUT (continued)

Then follows a line printer plot of the crustal weight, and a line printer plot of the observed, and theoretical gravity.

Onto punch (106) - new polygons points if these are calculated.

USAGE:

Input is assumed to be on cards - assign F:1 to card reader; F:2 to output device.

A crustal model is prepared which is composed of various polygons of various densities. The polygons are numbered arbitrarily, with the exception of polygon number 1, which is a water layer, and the final polygon; which must be number 99. One point of each polygon is designated the "starting point". Points define the polygons by proceeding clockwise from the starting point and ending exactly at the same point. Polygon points must be arranged in this order for input to the TALPLOT program.

RESTRICTIONS:

1. Input is assumed to be on cards
2. There can be no more than
   200 field points per model
   150 polygon points per polygon
   20 variable polygon points
3. Polygon points must be arranged in order clockwise from starting point, and the final point of each polygon must be exact by the same as the starting point. Holes or overlaps between polygons will cause strange and exciting gravity variations. To check for these, see column marked WEIGHTEST in the line printer output.

STORAGE REQUIREMENTS: Unknown

SUBPROGRAMS REQUIRED: PLOTA, PLOTER, ISW, SIMUL, WEIG2

TIMING: Unknown

ERROR AND DIAGNOSTICS: None

PROGRAMMER: Allin Folinsbee

ORIGINATOR: Allin Folinsbee

DATE: 20 July 1975
PROGRAM ARSTGC
PROGRAM TO ABSTRACT GCBN DATA BY 10 MIN AVERAGES
VERSION OF 5 NOV 75 TO CORRECT CENTER LAT AND LONG

ORIGINAL VERSION 20 AUG 75

DIMENSION IEBFIN(666)
DIMENSION IEBFOT(1184)
DIMENSION KBDE(300),KDLAT(300),KLATMIN(300)
DIMENSION KLDGEC(300),KLONGDEC(300)
DIMENSION KDLNG(300),KLONMIN(300),KFA(300),KELDEP(300)
DIMENSION KAVFA(6,6),NPTS(6,6),KAVEL(6,6)
DIMENSION H8AFAR(6,6),KCENLAT(6,6),KCENLNG(6,6)
DIMENSION KCENEL(6,6),KCENFAM(6,6)

CONTINUE

100 C CONTINUE

101 C CONTINUE

102 CONTINUE

103 CONTINUE

104 CONTINUE

105 CONTINUE

READ BOUNDS
READ(NCARD,1003) KDTBF,KCBET,KDLFT,KCRGT
KDTBF=KDTBF+200
KCBET=KCBET+200
KDLFT=KDLFT+200
KCRGT=KCRGT+200
BLTFLT KDTBF,KCBET,KDLFT,KCRGT

1003 FORMAT(2015)

110 CONTINUE

IF(ISW(46),EE) GO TO 200

READ(NBR,EGL,ECL) NBR,NOLTLT,NOLTLGR,NDLTGL
BLTFLT NBR,NOLTLT,NOLTLGR,NDLTGL

C CHECK BOUNDS

1004 FORMAT(5X,16.3I3)

IF((NDLTGL*GT*KDTBF),BR,(NDLTLT*LT*KDBET)) GO TO 150
IF((NDLTLT*LT*KDLFT)*ANC((NDLTLGR*LT*KDLFT)), GO TO 150
IF((NDLTGL*LT*KCRGT) AND ((NDLTLGR*LT*KCRGT)) GO TO 150
GO TO 200
11C  OUTSIDE BOUNDS
11D CONTINUE
11E ULTPUT KBR
11F IF(NBR,LT, (300 MIN)) NIN+1 NBR=NBR+1 G6 TO 110
11G NBR=NBR-1 (300 MIN)
11H CALL BUFFER IN(ITAPE,O IBLFIN(1), 165C IKEY, N1)
11I 151 CONTINUE
11J GO TO (152#155, 153, 154) IKEY
11K 152 ULTPUT "WAITING"; GO TO 151
11L 153 ULTPUT "ERROR WHILE SKIPPING"; GO TO 900
11M 154 ULTPUT "ERROR WHILE SKIPPING"; GO TO 900
11N 155 CONTINUE
11O N1=0
11P GO TO 150
11Q INPUT LOGIC
11R CONTINUE
11S IF(NBR=EG+C+AND+ISW(48)*+G2) G6 TO 110
11T IF(NIN=LT+1) GO TO 220
11U ULTPUT N1
11V 210 CONTINUE
211 CALL BUFFER IN(ITAPE, O IBLFIN(1), 165C IKEY, N1)
212 GO TO (211+215, 213+214) IKEY
213 ULTPUT "WAITING"; GO TO 210
214 ULTPUT "ERROR ON INPUT"; GO TO 500
215 CONTINUE
216 N1=N1+1/25
217 DEOCE(N1)*2, IBLFIN(1) NC
218 1001 FORMAT (30C11+13, 1213, 12, 16, 15)
219 N1=0
220 100 FORMAT 1, 11, 300
221 CLATIN=FLBAT(KLATDEC(I1)/100)*60.
222 KLATMIN=IFIX(DLATMIN)
223 KLNGM1N=FLBAT(KLNGDEC(I1)/100)*60.
224 KLONGMIN=IFIX(DLONGMIN)
225 CONTINUE
226 CONTINUE
227 IF(NIN=1) G6 TO 225
228 IF(ISM=*4) G6 TO 225
229 IF(KURLAT+KURAT/499) G6 TO 265
230 IF(KURLAT+KURLAT AND KLONGKURLAT+KULONG) G6 TO 300
231 CONTINUE
232 CONTINUE
233 PREPARING LAST DEGREE SQUARE FOR 0/P
234 CONTINUE
235 CONTINUE
236 C EOCDE(1184, 100C IBLFST(I), NE)
237 C KURLAT KURLAT, O C C O C O
238 C KURLAT, KURLAT, KURLAT, KURLAT, KURLAT, KURLAT
239 C KURLAT, KURLAT, KURLAT, KURLAT, KURLAT, KURLAT
240 C KURLAT, KURLAT, KURLAT, KURLAT, KURLAT, KURLAT
241 C KURLAT, KURLAT, KURLAT, KURLAT, KURLAT, KURLAT
242 260 CONTINUE
CALL BUFFER BLT(WRONG) CONTINUE
KURLAT=KCLAT(NIN)
KURLONG=KCLONG(NIN)
1002 FORMAT(37(16,16,212,216,14))
C CLEARING ARRAYS BEFORE NEW DEGREE SQUARE
CB 280 I=1,6
CB 27C J=1,6
KAVFA(I,J)=0
KAVEL(I,J)=0
NPTS(I,J)=0
KCENEL(I,J)=0
KCENLAT(I,J)=0
KCENLONG(I,J)=0
270 CONTINUE
CB 280 CONTINUE
SAME DEGREE SQUARE
1111 CALL BUFFER BLT(WRONG) CONTINUE
1112 KURLAT=KCLAT(NIN)
1113 KURLONG=KCLONG(NIN)
1114 1002 FORMAT(37(16,16,212,216,14))
1115 C CLEARING ARRAYS BEFORE NEW DEGREE SQUARE
1116 CB 280 I=1,6
1117 CB 27C J=1,6
1118 KAVFA(I,J)=0
1119 KAVEL(I,J)=0
1120 NPTS(I,J)=0
1121 KCENEL(I,J)=0
1122 KCENLAT(I,J)=0
1123 KCENLONG(I,J)=0
1124 270 CONTINUE
1125 CB 280 CONTINUE
1126 SAME DEGREE SQUARE
1127 IF (KAVFA(I,J)+KAVEL(I,J)+KELDEL(NIN))
1128 KAVEL(I,J)*KAVEL(I,J)*KELDEL(NIN)
1129 KAVFA(I,J)*KAVFA(I,J)*KFA(NIN)
1130 KAVFA(I,J)*KAVFA(I,J)*KFA(NIN)
1131 IF (KAVFA(I,J)+KAVEL(I,J)+KELDEL(NIN))
1132 KAVEL(I,J)*KAVEL(I,J)*KFA(NIN)
1133 KAVFA(I,J)*KAVFA(I,J)*KFA(NIN)
1134 IF (KAVFA(I,J)+KAVEL(I,J)+KELDEL(NIN))
1135 KAVEL(I,J)*KAVEL(I,J)*KFA(NIN)
1136 KAVFA(I,J)*KAVFA(I,J)*KFA(NIN)
1137 CONTINUE
1138 CONTINUE
1139 CB 280 CONTINUE
1140 CB 27C J=1,6
1141 KAVFA(I,J)=0
1142 KAVEL(I,J)=0
1143 NPTS(I,J)=0
1144 KCENEL(I,J)=0
1145 KCENLAT(I,J)=0
1146 KCENLONG(I,J)=0
1147 DIST=SQR((KLATMIN(NIN)-(J*10+5))**2+(KLONGMIN(NIN)-(J*10+5))**2)
1148 IF (DISTLT+BWFA(I,J)) -BWFAR(I,J)*DIST
1149 * KCENLAT(I,J)*KLATDEC(NIN)
1150 * KCENLONG(I,J)*KLONGDEC(NIN)
1151 * KCENEL(I,J)*KFA(NIN)
1152 GO TO 200
1153 900 CONTINUE
1154 STOP
1155 END
**Highest Error Severity:** 0 (No Errors)

<table>
<thead>
<tr>
<th></th>
<th>DEC Words</th>
<th>HEX Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generated Code:</td>
<td>631</td>
<td>CC277</td>
</tr>
<tr>
<td>Constants:</td>
<td>10</td>
<td>C0000A</td>
</tr>
<tr>
<td>Local Variables:</td>
<td>10800</td>
<td>C2A30</td>
</tr>
<tr>
<td>Temps:</td>
<td>1</td>
<td>C0001</td>
</tr>
<tr>
<td>Total Program:</td>
<td>11442</td>
<td>C2CB2</td>
</tr>
</tbody>
</table>
SUBROUTINES USED: GRID2, SLINE, WR, STAT, ISW, SPOT2, CALCM, ROUTINES, RETBY,

SENSE switch options

PLAT LABEL

TIME INTERVAL - READS START AND END DATE. IF TAPE INPUT, TAPE CAN BE PRE-

POSITIONED BY SPECIFYING RECORDS TO BE SKIPPED. IF NO END DATE SPECIFIED,

NO TEST MADE FOR END DATE.

POINT FORMATS 1. SCALE IN INCHES PER DEGREE LONGITUDE 2. CONNECT PLOTTED

POINTS 3. PLOT EVERY NTH POINT 4. PLOTTING EVERY NTH GRID LINE

PLOT NUMBER 6. VALUE TO BE ANNOTATED 7. ANNOTATE EVERY NTH POINT 8. FORMAT

PLOT FORMATS CONTINUED TO CHARACTER HEIGHT (.0007 INCH) 3. INTEGER

OR NON-INTEGER CHART BOUNDS 4. DIGITS AFTER DECIMAL MARKER PLOTTED

GRID BOUNDARIES - TOP, BOTTOM, LEFT, AND RIGHT EDGES - IN DEGREES AND MINUTES

UP TO NOT READ ANY INPUT DATA, PLAT ONLY GRID

UP TO DELETE DRAWING NODE GRID LINES (IN GRID2)

IF NEXT PLOT WILL BE ON THE SAME GRID AS THIS PLOT.

ALSO, UP TO ADD CURRENT VELOCITIES TO SHIP'S VELOCITY

FOR current = 2 AND BLK = 0 (IN SUB, GETO2)

UP TO ANNOTATE ONLY AT CHANGE OF DAY

FOR NO MARK AT DATA POINT

@ FOR PLOTTING A CIRCLE AROUND DATA POINT

@ FOR PLOTTING A CIRCLE AROUND DATA POINT

@ TO MAKE DEGREE ANNOTATIONS INSIDE GRID (CHARACTER SIZE 0*07 INCH)

@ TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0*21 INCH)

@ FOR MULTIPLE RUNS, UP TO ANNOTATE ON LEFT SIDE OF TRACK

@ FOR MULTIPLE RUNS, UP TO ANNOTATE ON LEFT SIDE OF TRACK

@ SUPPRESS PLOTTING OF GRID

UP TO LIST DATE AND TIME OF DATA BUT OF CHART BOUNDARIES

UP TO CALL SUBROUTINE ROLL WHICH READS IN SERIAL NUMBER OF INPUT TAPE

USED FOR PLOTS HAVING TWO OR MORE INPUT TAPE

UP TO ANNOTATE POINTS ALTERNATELY ON LEFT AND RIGHT SIDES OF TRACK

UP TO LIST DATE OF DATA JUST READ FOR IDENTIFICATION

UP IF TWO OR MORE PLOTS ARE BEING MADE FROM THE SAME TAPE AND THIS IS
NOT THE LAST PLOT TAPE WILL BE POSITIONED AT BEGINNING OF CURRENT
FILE EVEN IF END OF FILE PASSED.

SSW(15) = 1 FOR ADDITIONAL SIZE INCREMENT IN PLOTTING SYMBOLS IF
SSW(16) = 1 UP TO PLOT SPOT FOR SEISMICITY DATA PROPORTIONAL TO MAG AND DEPTH
SSW(17) = 0 TO PLOT AN X FOR SEISMICITY DATA PRE-1961
SSW(18) = 1 TO MAKE ANNOTATIONS AT RIGHT ANGLES TO INCREMENTAL TRACK (AN4V2)
SSW(19) = 0 TO MAKE ANNOTATIONS HORIZONTALLY
SSW(20) = 1 TO MAKE ANNOTATIONS VERTICALLY
SSW(21) = 1 TO INVERT ANNOTATIONS FOR HEADINGS 180 TO 269
SSW(22) = 0 FOR EARTH MERIDIONAL PARTS FROM BOWDITCH
SSW(23) = 1 FOR MERIDIONAL PARTS FOR SPHERICAL PLANET
SSW(24) = 1 (SEISMICITY) FOR ADDITIONAL SIZE INCREMENT FOR ALL DATA PTS,
SSW(25) = 1 (SEISMICITY) FOR SIZE FACTOR BY WHICH PLOTTING SYMBOLS WILL
VARY ACCORDING TO MAGNITUDE. IF A = 0, THEN AN4V4 SETS N = 2.
SSW(26) = 1 TO READ DATA IN ASCII CODING RATHER THAN EBCDIC
FOR CFMT = 19 AND 5 WITH BLOCK = 0
SSW(27) = N TO ANNOTATE EVERY N HBLRS BN THE HBLR

DIMENSION IBUF(1000), LABELP(20), NW(4), IEND(4)
INTEGER BLOCK
II8T = 108
IYR = 0
MFILE = 0
CALL ABORTSET(310S,15)
C PRINT DATE AND TIME OF JOB ON HEADING
C CHECK DAY(NOW)
WRITE(II8T,13) NOW
13 FORMAT(1x,4A4)
WRITE(II8T,9)
9 FORMAT(1x,8A4)
CALL SETSKP(INC)
CALL PLOTS(IBLF, 1900)
CALL PLOT(2000)
CALL STAT
C MOVE PEN IN FROM EDGE AND ALONG PAPER SO THAT IF LETTERING OUTSIDE GRID WILL
C NOT ROLL OFF EDGE.
C NOT RUN OFF EDGE.
CALL PLOT(1*0,0,5,3)
CALL STAT
CALL MAT(ITAPE,NAME)
WRITE (T1OUT,784) NAME
784 FORMAT( 'USING INPUT TAPE NUMBER: 1X,4A4)
C POSITION LABEL DEPENDING WHETHER ANNOTATION INSIDE OR OUTSIDE GRID
PLACE=0+80
IF (ISKPt.5) EG=1 PLACE=-1-35
CALL SYMBOL(PLACE,1@0+1@LABELP,9@0+80)
C ANNOTATE DATE CHART MADE IN LOWER LEFT CORNER OF PLOT
CALL SYMBOL(PLACE=0+8B+C@0+16)
CALL PLOT(1@0+0@0,3)
C
C CARD 3 TIME INTERVAL OF DATA, AND IF TAPE INPUT = POSITION ON TAPE.

INIT=
JDA*0
JMB*0
JBYR*0
JH*0
NY=1
NZ=1
N=1
IEBD*0
IGAP*0
IAGAP=0
NPTS=0
IPCT=0
IPCE=1
RAUGD=57.29578
DEG=1.7=53929E=2
READ(11.2) ISTDA,ISTMB,ISTYR,ISTHM,IENTA,IENTB,IENTR,IENTH,ISKP,
1ISFIL,ISDCF
2 FORMAT(12,14,6X,312,14,5X,315)
WRITE(1111U7,6365) ISTDA,ISTMB,ISTYR,ISTHM,IENTA,IENTB,IENTR,IENTH,ISKP,
1ISFIL,ISDCF
6365 FORMAT(START DATE 1312,14,5X,312,14,5X,315,RECORD
1 IS SKIPPED TO START OF INTERVAL (ISKP), 1,14,7X,FILES SKIPPED TO
2START OF INTERVAL (ISFIL), 1,14)
6 FORMAT(1, SERIES OF OVERLAPPING CHARTS BEING MADE, OVERLAP OF NEXT
1 CHART ON THIS CHART WILL BE 1,",",1 RECORDS,"
SPACING FILES ON MULTIPLE INPUT TAPES
IF(ISFIL+EG=0) GO TO 8
CALL SKPFIL(ITAPE,ISFIL,IFWD)
GO TO (997,1111,997.997,997) IN
SPACING RECORDS ON INPUT TAPE
IF(ISKP+EG=0) GO TO 8
IF(ISKP=GT=1) CALL SKPREC(ITAPE,ISKP,1FWD)
IF(ISKP=LT=1) ISKP=ISKP; CALL SKPREC(ITAPE,ISKP,1REV)
GO TO (997,997,997,999,999) IN
CARD 4 PLOT FORMAT PARAMETERS
8 READ(11.3) SINCX,ITRK,LCNT,NDEG,NMPL,NPTA,IFMT,NX,NFILE
3 FORMAT(F12=0,315,1X,4,4X,15)
IF(IFMT+EG=2) IFMT+2
IF(IFMT+EG=4) IFMT+3
IF(IFMT+EG=6) IFMT+4
IF(IFMT+EG=8) IFMT+1
IF(IFMT+EG=0) OUTPUT + 1), READ ROUTINE IS NOT GET(21)
WRITE(1111U7,64983) SINCX,ITRK,LCNT,NDEG,NMPL,NPTA,IFMT,NX
64983 FORMAT(10x,TRACK POINTS CONNECTED (ITRK),12,12,4X,1PLOTTING EVERY NTH POI
2NT (LCNT),12,12,4X,1PLOTTING EVERY NTH DEGREE LINE (NDEG),12,12,4X,
3PLT NUMBER (NMPL),12,12,4X,1ANNOTATE EVERY NTH PLOTTED POINT (NP
4TA),12,12,4X,1DATA FORMAT (JFMT),12,12,4X,1VALUE ANNOTATED (NX),1,
WRITE(1111U7,6493) NFILE
6493 FORMAT(I) NUMBER OF FILES BEING OUTPUTTED ON SAME GRID (NFILE)
181. C WFM SPECIFIES THE FORMAT OF THE DATA
182. C 1 *** FIX DATA
183. C 2 *** SEAGI DATA
184. C WFM 3 *** OSLM DATA
185. C 4 *** MBATR DATA
186. C 5 *** CALCH DATA
187. C 6 *** STATN DATA
188. C 7 *** SPFMT DATA
189. C 9 *** VOLCABES
190. C 10 *** HEAT FLOW
191. C 11 *** LLAR DATA
192. C 12 *** FLEXIBLE (SUPPLIED BY USER)
193. C
194. C NSKIP=LCTN+1
195. C NSTBP=LCTN
196. C MSTBP=PTA
197. C CARD 5 PLOT FORMAT PARAMETERS
198. C READ(IN,5) KPT,KHT,JICYP,IDECS, BLOCK
199. C 4 FORMAT(ISG)
200. C WRITE(3,B849) KPT,KHT,JICYP,IDECS, BLOCK
201. C 6499 FORMAT(MAGNIFICATION FACTOR (KPT)=1,12.1X, CHARACTER HEIGHT *0.07
202. C S INCH (KHT)=1.12.4X,NB=INTEGER OR INTEGER CHART BOUNDS (ICYP)
203. C 2.12.2/20.0 DECIMAL POINT IN ANNOTATION (IDECS)=1,12,
204. C 9 USING THE DDL'S BLOCKED TAPES (NB/YES) *.0/1,1,03,0/7
205. CIDECS=N FOR N DIGITS TO RIGHT OF DECIMAL PT. IN SUBR ANNV2 ANNNOTATION
206. C *0 DECIMAL PT ONLY
207. C *1 SUPPRESS DECIMAL POINT
208. C ZZ = KPT
209. C S INCH = S INCH + ZZ
210. C ZHT/KPT+KHT
211. C CARD 6 PLOT BOUNDARIES
212. C (CARDS 6 TO 9 IF NON-INTEGER BOUNDS)
213. C ITPB, IBBT, ILFFT, IRIGHT READ IN BY RETBY
214. C WRITE(3,B6364)
215. C 6364 FORMAT(I) CHART BOUNDARIES (10,10, EAST AND NORTH POSITIVE)
216. C IF(JICYP)=125.15.1210
217. C 21G CALL RETBY
218. C A DATA,1EDG,1IN,1IBUT,1ITAPE,NUMPL,DART,RLANG,K4G4M,IA4AP,LCN
219. C R,RAEG,DEG4EA,KBEG2,IEG2,IPDEG2,RAEG2,RTOP,ITOP,RB0T,IB0T,RLFET,
220. C LLEFT,RRIGHT,IRIGHT,SLST,SLGK,SPK,FB0T,FTOP,FLEFT,FRIGHT,NDG,
221. C SLAT,SLONG,SLOMP,XX,YY,INIT,XOLD,YOLD)
222. C 00 TO 23C
223. C DEGREES AND MINUTES FOR RTOP, RB0T, RLEFT, RRIGHT READ IN BY
224. C ARLIM AS CALLED BY RETBY
225. C 205 CALL VETBY
226. C A DATA,1EDG,1IN,1IBUT,1ITAPE,NUMPL,DART,RLANG,K4G4M,IA4AP,LCN
227. C R,RAEG,DEG4EA,KBEG2,IEG2,IPDEG2,RAEG2,RTOP,ITOP,RB0T,IB0T,RLFET,
228. C LLEFT,RRIGHT,IRIGHT,SLST,SLGK,SPK,FB0T,FTOP,FLEFT,FRIGHT,NDG,
229. C SLAT,SLONG,SLOMP,XX,YY,INIT,XOLD,YOLD)
230. C 23C CONTINUE
231. C C INITIALIZE GET02
232. C
233. C IF(FMT/=EG) 1 GO TO 23B
234. C 1 FMT /=EG) 4 GO TO 23B
235. C 2 FMT/=EG) 5 GO TO 23B
236. C 5 BLOCK GT C
237. C ) CALL GET02(IITAPE,x)
238. C 0 ) CALL GET02(IITAPE,x)
239. C 0 CALL GET02(IITAPE,x)
240 9 DATA X, DATAY, DATAZ, DATAW, ITRK, LCNT, NPTA, IDEC,
241 8 IEG, IAGAP, IFMT, INYR, BLOCK)
242 4 IF (ISW(19)>0) G0 TO 240
243 4C T3 SET ORIGIN - MERIDIONAL PARTS FROM BOUNDARY
244 4 TO AB(S(RB8T))*2 + 0.45*C*DEGRA
245 4B & (ALSG(SIN(A)/COS(A)))*G0.4342945
246 4 DATAX, DATAY, DATAZ, IDENT
247 1) = (G0.025*(SIN(ABS(RB8T))*AS3))
248 4B G8 TO 15
249 4C CONTINUE
250 4 TO SET ORIGIN - MERIDIONAL PARTS FOR SPHERICAL PLANET
251 4A = AB(S(RB8T))/2.0 + (4.5 * DEGRA)
252 4B = (ALSG((SIN(A)/COS(A))
253 4B G0MP = 3.4374748
254 15 IF (RBT) 20730
255 2C G0MP = 8B9MP
256 3C IF (ITYP) 222232
257 5C SUBROUTINE 8L1NE FOR GRID ENCLOSED BY NO# INTEGER DEGREES
258 22 CALL 8LINE(ZZ, ZHT;
259 4A IDAT A, IDAT B, IN, I NPUT, H AP, NPL, DAT A, R LEA T, R LON G, KG HE, IAG A P, LC N
260 4B R DEG, DEGRA, KDE G2, ID EGA, RDEG2, RT8P, RT8P, RBT, RGT, P LET, SL A T, SLONG,
261 4C SLAT, SLONG, BETMP, XX, YY, INIT, XLD, YLD)
262 4C CHECKING IF ONLY DRAWING OF GRID WANTED
263 26 IF (ISW(0)) 310310391
264 2C SUBROUTINE GRID2 FOR GRID ENCLOSED BY WHOLE DEGREES
265 32 CALL GRID2(ZZ, ZHT, NUMPL, DEGRA, DEG2, DEGRA, DEG2, RT8P, RT8P, RBT, RGT, P LET,
266 1 SLAT, SLONG, BETMP)
267 4C CHECKING IF ONLY DRAWING OF GRID WANTED
268 27 IF (ISW(0)) 310310391
269 2C END OF INITIALIZATION, BEGIN PLOTTING POINTS
270 2C CHECKING IF TRACK POINTS SHOULD BE CONNECTED
271 25 IF (ITRK) 404045
272 4C IP = 3
273 4G G8 TO 70
274 45 IF (IP) 555550
275 4G IP = 3
276 28 G8 TO 70
277 55 IF (ITYP) 656560
278 6C IP = 3
279 6G G8 TO 70
280 6S IP = 2
281 4C PLOT DATA POINT
282 7C SLAT = R LAT
283 7LON G = R LON G
284 7C CALL WRH(DEGRA,DEG2,DEG8,REALT,SLIN C, SMP, SLAT, SLONG, BETMP, XX, YY)
285 7C CALL PLOT(XX, YY, IP)
286 7C IP = IP+1
287 7C IF(ISW(16)) 7373710
288 71C CALL ANSV4(XX, YY, DATAZ, DATAW, YR)
289 73 G8 TO 72
290 4C CHECK FOR TYPE OF DATA POINT ANNOTATION MADE IN SPOT2
291 73 CALL SPOT2(XX, YY)
292 74 DATA = DAT A
293 74 IF (NX) 717271
294 71 MSTOP = MSTOP+1
300  IF(MSTBP)871,871,72
301  871 CONTINUE
302  C TEST FOR TIME ANNOTATED AT 2- HOUR INTERVALS
303  C
304  IF(ISHM(71)) 872,874,872
305  872 IF(JCDA = JBD(1)) 875,877,875
306  873 IF(LNATU(OH/100) = LNATUJUH/100)) 874,872,874
307  874 ICT = ICT + 1
308  875 ICT D 0
309  878 CONTINUE
310  CALL ANV2(ZZ,ZHT,NX,JCDA,MB,YR,WJH,JBD,XX,YY,INIT,DATAS,IDE);
311  MSTOP=NPTA
312  72 INIT = 0
313  74 SLAT = RLAT
314  75 SLONG = RLONG
315  76 QUAL = QUAL
316  77 OYRE = OYRE
317  78 WOHM = WOHM
318  79 NPYSS = NPYSS
320  DATA P@INT PLOTTED, INPUT NEXT DATA POINT
321  CONTINUE
322  C DATA POINT PLOTTED, INPUT NEXT DATA POINT
323  C INPUT DATA FORMATS: 308 FIXSE 308 SEAS1 311 QSUM 308 MBATR 308 CALCM
324  C 314 STA DATA 340 SPFM 342 SEISMICY 344 VOLCANES 346 HEAT FLB
325  C 348 LUNAR DATA 350 FLEXIBLE (USER SUPPLIED)
326  C 310 GO TO (308,308,311,308,314,340,342,344,346,348,350,350) JFMT
327  308 CONTINUE
328  IF(BLOCK EG 0, AND), JFMT EG 2) CALL GETS(ITAPE,
329  $ NX,NY,NZ,NW,DATAX,DATAY,DATAZ,DATAW,RLAT,RLONG,
330  $ JDA,YR,JHM,IDE) / GO TO 78
331  CALL GETG(ITAPE,1
332  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
333  $ JHM,IDE) / GO TO 78
334  346 CALL GETH(ITAPE,1
335  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
336  $ JHM,IDE) / GO TO 78
337  350 CALL GETX(ITAPE,1
338  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
339  $ JHM,IDE) / GO TO 78
340  351 CALL GETY(ITAPE,1
341  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
342  $ JHM,IDE) / GO TO 78
343  352 CALL GETV(ITAPE,1
344  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
345  $ JHM,IDE) / GO TO 78
346  353 CALL GETL(ITAPE,1
347  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
348  $ JHM,IDE) / GO TO 78
349  354 CALL GETA(ITAPE,1
350  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
351  $ JHM,IDE) / GO TO 78
352  355 CALL GETB(ITAPE,1
353  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
354  $ JHM,IDE) / GO TO 78
355  356 CALL GETC(ITAPE,1
356  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
357  $ JHM,IDE) / GO TO 78
358  359 CALL GETD(ITAPE,1
359  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
360  $ JHM,IDE) / GO TO 78
361  362 CALL GETE(ITAPE,1
362  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
363  $ JHM,IDE) / GO TO 78
364  365 CALL GETF(ITAPE,1
365  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
366  $ JHM,IDE) / GO TO 78
367  368 CALL GETG(ITAPE,1
368  $ JDA,YR,JHM,RLAT,RLONG,NX,NY,NZ,NW,
369  $ JHM,IDE) / GO TO 78
IF (IE0=1) 85*,1212*,1212

1212 N, FILE=FILE+1

IF (FILE=LE*0) GO TO 12

INIT = 1

IE0=0

FILE=FILE+1

BLLUT 'PLOTTING CONTINUES FROM NEXT FILE'

GO TO 310

C END OF DATA FILE - CHECK SENSE SWITCHES FOR NEXT STEP

C CHECKING IF ANOTHER PLOT FOLLOWS OR TO EXIT

12 IF (ISW(2)=EQ*0 AND ISW(13)=EQ*0 AND IBCKUP*EQ*0) GO TO 91

IF (ISW(2)=EQ*1) CALL PLOT(0*0*0*0*0*3)

SENSE SWITCH 13 CHECKS WHETHER ANOTHER PLOT IS TO BE MADE FROM SAME FILE

IF (ISW(13)) 80*81*80

ICBACK=0

IF (IE0=EQ*1) ICBACK=1

IF (FILE=NE*0) ICBACK=FILE+1

CALL SKPFIL(ITAPE,IBACK,IR)

C SEQUENTIAL PLOTS FROM SAME FILE WITH (IBCKUP) OVERLAP

81 IF (IBCKUP*EQ*0) GO TO 96

IF (IBCKUP*EQ*1) CALL SKPFIL(ITAPE,IBCKUP,IR)

IF (IE0=EQ*1) CALL SKPFIL(ITAPE,IR)

C ESTABLISH ORIGIN OF NEW PLOT

96 IF (ISW(2)*EQ*1) GO TO 95

XINCRE=ABS(RDEG)*RDEG*SINCH*4*00

CALL PLOT(XINCRE) GO TO 79

C PUT RLA AND DATA CARDS FOLLOWING LAST DATA DECK TO REINITIALIZE PROGRAM.

TEN=79

C CLOSE PLOT TAPE AND END OF LABEL FOR PLOT=5 OPERATOR

DATA (IEND(1),IEND(4),/END OF CHART /GB) /

C COMPLETE BOTTOM AND RIGHT SIDES OF FIDUCIAL HALF-INCH SQUARE DRAWN IN GRID

XINCRE=ABS(RDEG)*RDEG*SINCH*4*00

CALL PLOT(XINCRE) GO TO 95

C CHECK IF DATE IS WITHIN SPECIFIED TIME INTERVAL

FLAG IS A FLAG ALLOWING THE FIRST CALL TO FIND IF WE HAVE ALREADY FOUND THE STARTING DATE

IF (FLAG*NE*0) GO TO 82

CALL FIND(ISTDA,ISTMB,ISTYR,ISTHM,IDA,JMD,JYR,JHM,INDK)

IF (INDK*EQ*1) GO TO 310

IF (FLAG) 82

IF (IE0=EQ*0) GO TO 851

CALL FIND(IE0A,IE0B,IE0VR,IE0NR,IE0MH,IDA,JMD,JYR,JHM,INDK)
IF(INDB=EQ.1) G6 TO 995

C CHECKING IF DATA WITHIN CHART BOUNDARIES

851 NSTEP=LCNT

86 IF(RT6P=RLAT) 100,100,86

88 IF(RLAT=R66T) 100,88,98

90 IF(RRIGT=RL6NG) 100,100,35

C DATA OUTSIDE OF BOUNDS

C WRITING OUT DATE IF DATA POINT IS OUT OF BOUNDS

100 INIT=1

410 WRITE(IOUT,995) JDA, JMB, YR, JHM

420 FORMAT('66B 1,312,1X,14')

995 WRITE(IOUT,996) JDA, JMB, YR, JHM

996 FORMAT('END DATE PASSED',2X,312,1X,14)

G6 TO 12

C ERROR MESSAGES IF MISTAKE IN TAPE FILE OR RECORD SPACING,

999 WRITE(IOUT,999) IND

998 FORMAT('ERROR IN SKPREC, IND=',12)

G6 TO 91

997 WRITE(IOUT,994) IND

994 FORMAT('ERROR IN SKPFIL, IND=',12)

G6 TO 91

END
### Local Variables (1133 Words)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000C IBUF</td>
<td>003E8 LABELP</td>
</tr>
<tr>
<td>00016E IIBUT</td>
<td>00407 JPMT</td>
</tr>
<tr>
<td>00040C I</td>
<td>00400 INZ</td>
</tr>
<tr>
<td>000412 JBCA</td>
<td>00413 JMB</td>
</tr>
<tr>
<td>000418 NW</td>
<td>00419 IEDB</td>
</tr>
<tr>
<td>00041E RADEG</td>
<td>0041F CEORA</td>
</tr>
<tr>
<td>00042E IENOA</td>
<td>00425 IENMB</td>
</tr>
<tr>
<td>00042A IBCUP</td>
<td>0042B SINC</td>
</tr>
<tr>
<td>00043C NPTA</td>
<td>00431 WFM</td>
</tr>
<tr>
<td>000436 MSFBP</td>
<td>00437 KPT</td>
</tr>
<tr>
<td>00043C ZHT</td>
<td>00430 IDATA</td>
</tr>
<tr>
<td>000442 KDEG2</td>
<td>00443 IDEG2</td>
</tr>
<tr>
<td>000448 RBF7</td>
<td>0044A RLEFT</td>
</tr>
<tr>
<td>00045E SLTG</td>
<td>0045F SLKG</td>
</tr>
<tr>
<td>00045A FRIGT</td>
<td>00455 SLAT</td>
</tr>
<tr>
<td>00045A XLD</td>
<td>00458 YLD</td>
</tr>
<tr>
<td>000460 CATAX</td>
<td>00461 DATAY</td>
</tr>
<tr>
<td>000466 IP</td>
<td>00467 ICT</td>
</tr>
<tr>
<td>00046C INDK</td>
<td>00468 ICTL</td>
</tr>
</tbody>
</table>

### Blank Common (0 Words)

### Intrinsic Subprograms Used:

<table>
<thead>
<tr>
<th>Subprogram</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
</tr>
<tr>
<td>ALOG</td>
</tr>
<tr>
<td>CBS</td>
</tr>
<tr>
<td>FLAT</td>
</tr>
<tr>
<td>SIN</td>
</tr>
</tbody>
</table>

### External Subprograms Required:

<table>
<thead>
<tr>
<th>Subprogram</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABORTSET</td>
</tr>
<tr>
<td>ANDV2</td>
</tr>
<tr>
<td>AN024</td>
</tr>
<tr>
<td>ANOV4</td>
</tr>
<tr>
<td>EXIT</td>
</tr>
<tr>
<td>FIND</td>
</tr>
<tr>
<td>GET</td>
</tr>
<tr>
<td>GETG</td>
</tr>
<tr>
<td>GETH</td>
</tr>
<tr>
<td>GETL</td>
</tr>
<tr>
<td>GETO2</td>
</tr>
<tr>
<td>GRID2</td>
</tr>
<tr>
<td>GETV</td>
</tr>
<tr>
<td>GETX</td>
</tr>
<tr>
<td>GETY</td>
</tr>
<tr>
<td>GETZ</td>
</tr>
<tr>
<td>I56</td>
</tr>
<tr>
<td>HBLAT</td>
</tr>
<tr>
<td>MREL</td>
</tr>
<tr>
<td>BLINE</td>
</tr>
<tr>
<td>PLOT</td>
</tr>
<tr>
<td>PLOTS</td>
</tr>
<tr>
<td>REBY</td>
</tr>
<tr>
<td>SETH</td>
</tr>
<tr>
<td>SETSKP</td>
</tr>
<tr>
<td>SFRFIL</td>
</tr>
<tr>
<td>SKFREC</td>
</tr>
<tr>
<td>SPSBT</td>
</tr>
<tr>
<td>STAT</td>
</tr>
<tr>
<td>SYMBOL</td>
</tr>
<tr>
<td>TDBAY</td>
</tr>
<tr>
<td>VETBY</td>
</tr>
<tr>
<td>VHR</td>
</tr>
<tr>
<td>M10</td>
</tr>
<tr>
<td>F1101</td>
</tr>
<tr>
<td>F1102</td>
</tr>
<tr>
<td>F1103</td>
</tr>
<tr>
<td>F1104</td>
</tr>
<tr>
<td>F1105</td>
</tr>
<tr>
<td>F1106</td>
</tr>
<tr>
<td>FIL08</td>
</tr>
<tr>
<td>M108</td>
</tr>
<tr>
<td>H16C</td>
</tr>
<tr>
<td>9ALOG</td>
</tr>
<tr>
<td>BCDRCDEE</td>
</tr>
<tr>
<td>9BCDREAD</td>
</tr>
<tr>
<td>9BCDWRIT</td>
</tr>
<tr>
<td>9COS</td>
</tr>
<tr>
<td>9ENDB1L</td>
</tr>
<tr>
<td>9INITIAL</td>
</tr>
<tr>
<td>918DATA</td>
</tr>
<tr>
<td>918USA</td>
</tr>
<tr>
<td>9IT08</td>
</tr>
<tr>
<td>9PRINT</td>
</tr>
<tr>
<td>9IN</td>
</tr>
</tbody>
</table>

### Local Variables (1133 Words)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000C IBUF</td>
<td>003E8 LABELP</td>
</tr>
<tr>
<td>00016E IIBUT</td>
<td>00407 JPMT</td>
</tr>
<tr>
<td>00040C I</td>
<td>00400 INZ</td>
</tr>
<tr>
<td>000412 JBCA</td>
<td>00413 JMB</td>
</tr>
<tr>
<td>000418 NW</td>
<td>00419 IEDB</td>
</tr>
<tr>
<td>00041E RADEG</td>
<td>0041F CEORA</td>
</tr>
<tr>
<td>00042E IENOA</td>
<td>00425 IENMB</td>
</tr>
<tr>
<td>00042A IBCUP</td>
<td>0042B SINC</td>
</tr>
<tr>
<td>00043C NPTA</td>
<td>00431 WFM</td>
</tr>
<tr>
<td>000436 MSFBP</td>
<td>00437 KPT</td>
</tr>
<tr>
<td>00043C ZHT</td>
<td>00430 IDATA</td>
</tr>
<tr>
<td>000442 KDEG2</td>
<td>00443 IDEG2</td>
</tr>
<tr>
<td>000448 RBF7</td>
<td>0044A RLEFT</td>
</tr>
<tr>
<td>00045E SLTG</td>
<td>0045F SLKG</td>
</tr>
<tr>
<td>00045A FRIGT</td>
<td>00455 SLAT</td>
</tr>
<tr>
<td>00045A XLD</td>
<td>00458 YLD</td>
</tr>
<tr>
<td>000460 CATAX</td>
<td>00461 DATAY</td>
</tr>
<tr>
<td>000466 IP</td>
<td>00467 ICT</td>
</tr>
<tr>
<td>00046C INDK</td>
<td>00468 ICTL</td>
</tr>
</tbody>
</table>
### Highest Error Severity: 0 (No Errors)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Words</strong></td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td><strong>Generated Code</strong></td>
<td>1678</td>
<td>0068E</td>
</tr>
<tr>
<td><strong>Constants</strong></td>
<td>44</td>
<td>0002C</td>
</tr>
<tr>
<td><strong>Local Variables</strong></td>
<td>1133</td>
<td>0046D</td>
</tr>
<tr>
<td><strong>Temps</strong></td>
<td>3</td>
<td>00003</td>
</tr>
<tr>
<td><strong>Total Program</strong></td>
<td>2858</td>
<td>0082A</td>
</tr>
</tbody>
</table>
SUBROUTINES USED: GRID, BLINE, HR, STAT, ISW, SPRT, CALCNUM Routines, RETRY, VETEC, TODAY, ARLLIN, ENDB, EVIL, DMYR, FIND, ANSIV, RSTAP

PLT LABEL
SENSE SWITCH OPTIONS
TIME INTERVAL READS START AND END DATE, IF TAPE INPUT, TAPE CAN BE PRE-POSITIONED BY SPECIFYING RECORDS TO BE SKIPPED, IF NOT END DATE SPECIFIED
NO TEST MADE FOR END DATE.
PRINTS: 1. PLOT EVERY NTH PRINT, 2. PLOTTING EVERY NTH GRID LINE
5. PLOT NUMBER & VALUE TO BE ANNOTATED, 7. ANNOTATE EVERY NTH POINT, 8. FORMAT
PLOT FORMATS CONTINUED, 1. CHARACTER HEIGHT (#0.07 INCH), 3. INTEGER OR NON-INTEGER CHART BOUNDS, 4. DIGITS AFTER DECIMAL PRINTED PLOTTED
BOUNDARIES = TOP, BOTTOM, LEFT, AND RIGHT EDGES = IN DEGREES AND MINUTES

SSW (0) UP TO NOT READ ANY INPUT DATA, PLOT ONLY GRID
SSW (1) UP TO DELETE DRAWING NODE LINES
SSW (2) UP IF NEXT PLOT WILL BE ON THE SAME GRID AS THIS PLOT.
SSW (3) UP TO ANNOTATE ONLY AT CHANGE OF DAY
SSW (4) = 0 FOR NO MARK AT DATA POINT
= 1 FOR PLOTTING A CIRCLE AROUND DATA POINT
= 2 FOR PLOTTING A DOT AT DATA POINT
SSW (5) = 0 TO MAKE DEGREE ANNOTATIONS INSIDE GRID (CHARACTER SIZE 0.07 INCH)
= 1 TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.21 INCH)
SSW (6) = 0 FOR MULTIPLAT RUNS, UP WILL PUT AN EOF BETWEEN PLOTS, USEFUL TO PREVENT DATA BLEEDING IN THE EVENT OF MECHANICAL MALFUNCTION OF PEN.
SSW (7) UP TO ANNOTATE ON LEFT SIDE OF TRACK
SSW (8) UP TO SUPPRESS PLOTTING OF GRID
SSW (9) UP TO LIST DATE AND TIME OF DATA BUT OF CHART BOUNDARIES
SSW (10) UP TO CALL SUBROUTINE MOUNT WHICH READS IN SERIAL NUMBER OF INPUT TAPE
USED FOR PLATS HAVING TWO OR MORE INPUT TAPES

SSM(11) UP TO ANNIMATE POINTS ALTERNATELY ON LEFT AND RIGHT MARGIN OF THE MAP.

SSM(12) UP TO LIST DATE OF DATA, JUST READ FOR IDENTIFICATION.

SSM(13) UP IF TWO OR MORE PLATS ARE BEING MADE FROM THE SAME TAPE AND THIS IS

NOT THE LAST PLAT, TAPE WILL BE POSITIONED AT BEGIN OF FILE IF FILE PASSED.

FILE EVEN IF END-OF-FILE PASSED.

SSM(18) #0 TO MAKE ANNOTATIONS AT RIGHT ANGLES TO INCREMENTAL TRACK (LONGITUDINAL)

#1 TO MAKE ANNOTATIONS HORIZONTALLY

#2 TO MAKE ANNOTATIONS VERTICALLY

#3 TO INVERT ANNOTATIONS FOR HEADINGS, #4 TO PA8

SSM(19) #0 FOR EARTH MERIDIONAL PARTS FROM EARTH WHIC

#1 FOR MERIDIONAL PARTS FOR SPHERICAL PLANET

SSM(71) N TO ANNIMATE EVERY N HOURS

C

DIMENSION BUF(100), LABELP(20), NW(4), IEND(4)

COMMON BUF, LABELP, NW, IIN, IIBUF

1 MFILE, INZ, IFLAG, I, ZERB, NW

2 MAG, INZ, ITAPE, NAME, PLACE, INIT

3 XLD, YLD, JDA, JMA, JAYR, JAHM

4 SIN, ZC, IN, IEND, IGA, IGA

COMMON NPTS, ISTD, ISTD, ISTD, ISTD, ISTD, ISTD

1 JENO, JENY, JENH, JSKI, ISKI, ISKI

2 RAEK, DEGR, SINH, ITRK, LCNT, LDEG

3 MUP, NX, NPTA, JFMT, FILE, NSKI

4 NSTOP, NSTAK, KPT, KRT, ICTYP, IDEC

COMMON ZZ, ZHT, IDATA, DATA, RLAT, RLANG

KOBH, KOBH, KOBH, KOBH, KOBH, KOBH

1 ITP, RRT, JHT, RLEFT, RRIGHT

3 IRIGHT, SLTX, SLGY, SMP, FBOT, FBOT

4 FLIGHT, SLGT, SLTY, SLAG, XX

COMMON YY, A, B, IP, DATA, DATA

1 DATAX, JDA, JMA, JAYR, JAHM

2 RLGA, DATAX, KGDA, KGMB, KGMB

3 TRAC, XINC, IND

C

IIN = 105

IIBUF = 108

MFILE = 0

PRINT DATE AND TIME OF MAP IN HEADING

CALL TODAY(NOW)

WRITE(IIBUF,13) NOW

13 FORMAT(1X,4A4)

OUTPUT (PROGRAM CHART VERSION OF 04 FEB 76)

CALL SETSKP(IND)

CALL PLOT(LBUF, BUF)

C MOVE PEN IN FRAME EDGE AND ALONG PAPER SO THAT OUTSIDE GRID LETTERING WILL

NOT RUN OFF EDGE.

CALL PLOT(150, 50, 50)

CALL STAT

79 IFLAG = 0

CARD 1 PLOT LABEL (20A4)

C PUT LABEL ON LEFT MARGIN OF PLOT. SHIP, CRUISE, DATE, AND AREA, 80 CHAR.

HEAD(IN, END, 93, ERR, 91) LABELP(1), I=120

IF (ISW(5) .EQ 4) PLACE=1, 35

6 FORMAT(20A4)

WRITE(IIBUF,7) LABELP(1), I=120

7 FORMAT(20A4)
CARD 2

SENSE SWITCH OPTIONS = SM1 TO SM9 IN COLUMNS 1 TO 79, SM10 IN COLUMN 80

121.
122. IF(ISM(19).EQ.0) OUTPUT MERIDIONAL PARTS FROM DITCH
123. IF(ISM(19).EQ.1) OUTPUT MERIDIONAL PARTS FROM SPHERICAL PLANET

124. IF(ISM(10).EQ.1) OUTPUT HEADING 5765 NAME WHEN HAVING ITAPE NAME

125. WRITE(INPUT,5764) NAME

126. FORMAT USING INPUT TAPE NUMBER! 1X,A4)

127. FORMAT(AC)

128. POSITION LABEL DEPENDING WHETHER ANNOTATION INSIDE OR OUTSIDE GRID

129. PLACE*0*X0

130. CALL SYMBOL(PLACE:0.5,0.14,LABELP,90,0,30)

131. ANNOTATE DATE CHART MADE IN LOWER LEFT CORNER OF PLAT

132. CALL SYMBOL(PLACE:0.48,0.07,LOWA:0.07)

133. CALL PLAT(0.0,0.0,0.1)

134. CARD 3 TIME INTERVAL OF DATA, AND IF TAPE INPUT = POSITION ON TAPE.

135. INIT=

136. X#=

137. Y#=

138. Z#=

139. READ(IIN,2) ISTDA,ISTMB,YSTICK,ISTYM,ISTHM,ISTMB,IENTM,IENTP,IFNM,ISKP,

140. IF(ISFIL.EQ.0) GO TO 11

141. READ(TIN,2) ISTDA,ISTIC,ISTRIBUTION,INDEGS,NUMPL,NPTA,UFMT,NX,NTLK

142. FORMAT(F5.0,6I5,4A5)

143. WRITE(IOUT,5496) SINCH,ITRK,LCNT,INDEG,NUMPL,NPTA,JFMT,NX,NFIL,ITRLK

144. WRITE(IOUT,6498) CHART SCALE (SINCH) = 1.77 X INCHES PER DEGREE LATITUDE:

145. 4X/TRACK POINTS CONNECTED (ITRK) 11,12,14,16,PLOTTING EVERY NTH POINT

146. 14X/PLACING EVERY NTH DEGREE LINE (INDEG) 11,12,14,16

147. 3X/PLACING NUMBER (NOMPL) 11,12,14,16	ANNOTATE EVERY NTH PLOTTED POINT (NP

148. DATA FORMAT (JFMT) 11,12,14,16,VALUE ANNOTATED (NX)
180* WRITE((INUT,6493)) NFILE
181* 6493 FORMAT('1 NUMBER OF FILES BEING PUTPUTED ON SAME GRID (NFILE):')
182* C JFMT 3 *** GSUM DATA
183* C
184* NSKIP*LCA = 1
185* NSTB*LCA = 1
186* MSTB*LCA = 1
187* CARD 5 PLOT FORMAT PARAMETERS
188* READ(INUT,KPT,KHT,ICTYTP,IDECD
189* 4) FORMAT(415)
190* WRITE((INUT,6499)) KPT,KHT,ICTYTP,IDECD
191* 6499 FORMAT(\"MAGNIFICATION FACTOR (KPT)=\",12,4X,\"CHARACTER HEIGHT =\",12,4X,\"NON-INTEGER OR INTEGER CHART BRUNDS (ICTYTP)\",12,4X,\"DECIMAL POINT IN ANNOTATION (IDECD)\")
192* C
193* IDECD=VARIABLE FOR DECIMAL POINT IN ANG ANNOTATION
194* IDECD FOR N DIGITS TO RIGHT OF DECIMAL PT,
195* =0 DECIMAL PT ONLY
196* =1 SUPPRESS DECIMAL POINT
197* ZZ = KPT
198* SINCX = SINCX = ZZ
199* 2* = KPT = KHT
200* CARD 6 PLOT BOUNDARIES (CARDS 6 TO 9 IF NON-INTEGER BRUNDS)
201* ITOP, RTOP, LEFT, RIGHT READ IN BY RETRY
202* WRITE((INUT,6364))
203* 6364 FORMAT('CHART BOUNDARIES! ,20X, \"EAST AND NORTH POSITIVE\")
204* IF(ICTYTP)205,205,210
205* 205 CALL RETRY
206* A IADATA, IEDO, IN, IOUT, ITAPE, NUMPL, DATA, RLAT, RLNG, KTHGM, IAGAP, LCN
207* H, RADEG, DEGDA, KSTDG2, IDEG2, FDEG2, RDEG2, RTOP, ITOP, RTOP, ILEFT, RLFT,
208* C ILEFT, IRIGHT, IRIGHT, SLTK, SLKG, SINCX, SMMP, FBOT, FTOP, FLEFT, FRIGHT, NDG,
209* D SLAT, SLNG, Bampton, XX, YY, INIT, XBLD, YBLD;
210* G9 TO 230
211* DEGREES AND MINUTES FOR RTOP, RBOT, LEFT, RIGHT READ IN BY RETRY
212* ARLIM AS CALLED BY VETBY
213* 214
214* CALL VETBY
215* A IADATA, IEDO, IN, IOUT, ITAPE, NUMPL, DATA, RLAT, RLNG, KTHGM, IAGAP, LCN
216* H, RADEG, DEGDA, KSTDG2, IDEG2, FDEG2, RDEG2, RTOP, ITOP, RTOP, ILEFT, RLFT,
217* C ILEFT, IRIGHT, IRIGHT, SLTK, SLKG, SINCX, SMMP, FBOT, FTOP, FLEFT, FRIGHT, NDG,
218* D SLAT, SLNG, Bampton, XX, YY, INIT, XBLD, YBLD;
219* 220
220* DEGREES AND MINUTES FOR RTOP, RBOT, LEFT, RIGHT READ IN BY RETRY
221* ARLIM AS CALLED BY VETBY
222* 230
230 A = ABS(RRTA)2*0+45*DEGRA
231* B = (ALRG(SIN(A)/COS(A)))0+452945
232* BATOM=79.1574+30 *B=(23+2693*ABS(RRTA)
233* 234 1))((0+505*ABS(SIN(A)ABS(RRTA))))**3)
235* IF(RRTA) 20.30+30
236* BATOM = BATOM
237* 237
237* 1F(ICYTP)27,22,32
238* C SUBROUTINE ALINE FOR GRID ENCLOSED BY NON-INTEGER DEGREES
239* C CALL ALINE(22Z,2MT,
240* 22
241* C CALL ALINE(22Z,2MT,
242* 235 CHECKING IF ONLY DRAWING OF GRID WANTED
243* C IF(ISW(0))30,320,91
244* C SUBROUTINE GRIDP FOR GRID ENCLOSED BY WHOLE DEGREES
240 CALL GRID2(Z2,ZHT,NUMPL,DEGRA,FDEG2,RDEG2,RLEF,m,SLAT,SLONG,BATMP)
241 1 RLEF,m,LEFT,RIGHT,RIGHT,SLlsen,SLAT,FBAT,FLAT,FLAT,RIGHT,RIGHT,
242 2 SLAT,SLONG,BATMP)
243 C CHECKING IF ANLY DRAWING OF GRID WANTED
244 C IF(ISW(0))310,310*91
245 C ENDF OF INITIALIZATION, BEGIN PLOTTING POINTS
246 C CHECKING IF TRACK POINTS SHOULD BE CONNECTED
247 C IF(ITRK)40,40,45
248 40 IP=3
249 45 IF(INIT) 55,55,50
250 50 IP=3
251 55 IF(IGAP) 65,65,60
252 60 IP=3
253 65 IP=2
254 C PLOT DATA POINT
255 70 SLAT = SLAT
256 75 SLONG = SLONG
257 80 CALL WHR(DEGRA,FDEN2,RDEG2,RLEF,SLAT,SLONG,BATMP,XX,YY)
258 C CALL PLOT(XX,YY,IP)
259 C IPCT=IPCT+1
260 C IF(ISW(16))876,876,710
261 710 CALL ABY33(XX,YY,DATAZ,DATAW)
262 C CHECK FOR TYPE OF DATA POINT ANNOTATION MADE IN SPATS
263 C CONTINUE
264 876 C CONTINUE
265 877 CALL SPATS(XX,YY)
266 C IF(ISW(18))879,879,72
267 879 IF(ISW(18))879,879,72
268 90 DATA=DATA
269 91 IF(NX) 71,72,71
270 92 IF(MSPTP=MSPTP-NWIN
271 93 IF(MSPTP)=871*871*72
272 94 C CONTINUE
273 C TEST FOR TIME INTERVAL ANNOTATION
274 C IF(ISW(71))872*872*872
275 872 IF(JDA=JDA) 875*875*875
276 875 IF(FLAT(JMM/100)=FLOT(JMM/100)) 874*874*874
277 874 ITCT=ITCT+1
278 877 ITCT=1
279 878 CONTINUE
280 C CONTINUE
281 C C CONTINUE
282 C IF(ISW(71))872*872*872
283 872 IF(JDA=JDA) 875*875*875
284 875 IF(FLAT(JMM/100)=FLOT(JMM/100)) 874*874*874
285 874 ITCT=ITCT+1
286 877 ITCT=1
287 878 CONTINUE
288 879 C CONTINUE
289 878 C CONTINUE
290 879 C CONTINUE
291 879 C CONTINUE
292 879 C CONTINUE
293 879 C CONTINUE
294 879 C CONTINUE
295 879 C CONTINUE
296 879 C CONTINUE
297 879 C CONTINUE
298 879 C CONTINUE
299 879 C CONTINUE
**END IF DATA - CHECK SENSE SWITCHES FOR NEXT STEP**

**C CHECKING IF ANOTHER PLOT FOLLOWED OR TO EXIT**

**IF(ISW(2).EQ.0.AND.ISW(13).EQ.0.AND.IBACKUP.EQ.0) GO TO 91**

**C SENSE SWITCH 13 CHECKS WHETHER ANOTHER PLOT IS TO BE MADE FROM SAME FILE**

**IF(ISW(13)).GT.0.AND.90**

**80 IBACK=0**

**IF(IEND.EQ.1) IBACK=1**

**IF(NFILE.EQ.0) IBACK=NFILE+1**

**C SENSE SWITCH 14 CHECKS WHETHER ANOTHER INPUT TAPE IS TO BE MOUNTED**

**81 IF (ISW(14)).NE.83,83**

**87 WRITE(IOUT,6496)**

**FORMAT(*CHANGE TO NEXT INPUT TAPE*)**

**C SEQUENTIAL PLOTS FROM SAME FILE WITH (IBACKUP) OVERLAP**

**83 IF(IBACKUP.EQ.0) GO TO 96**

**84 IBACKUP=IBACKUP+1**

**85 IF(IEND.EQ.0) CALL SKPRECI(TAPE,IBACKUP,IREVI)**

**86 IF(IEND.EQ.1) CALL SKPFILE(TAPE,IBACKUP,IREVI)**

**C ESTABLISH ORIGIN OF NEW PLOT**

**96 IF (ISW(2).EQ.1) GO TO 95**

**98 XINCRE=RDG2*RADEG*SINCH+4**

**99 CALL PLAT(XINCRE,0,**

**50 IF(ISW(6).EQ.1) CALL PLAT(*1,000,000,999*) GO TO 94**

**C PUT RUN AND DATA CARDS FOLLOWING LAST DATA CLICK TO REINITIALIZE PROGRAM**

**GO TO 79**

**C CLOSE PLOT TAPE AND END OF JOB LABEL FOR PDP-8 OPERATOR**

**END IF DATA (IEND(1),IEND(8)) END OF CHART JOB**

**91 XINCRE=ABS(RDGO)*RADEG*SINCH+100**

**C CURVE BOUNDARIES AND RIGHT SIDES OF FIDUCIAL HALF-INCH SQUARE DRAWN IN GRID**

**100 CALL PLAT(XINCRE,0,0,0,999)**

**101 CALL PLAT(0,0,0,0,999)**

**102 CALL PLAT(0,0,0,5,9,3)**

**103 CALL SYMBO(0,0,0,0,0,42,END,90,0,16)**

**104 CALL PLAT(4,0,0,0,0,999)**
94 IF(ISW(10) NE 1) CALL MREL(1)
95 WRITE(IIOUT,97)PCT
97 FORMAT(1 NUMBER POINTS PLOTTED = I, I8)
98 CALL EXIT
99 C SKIPPING POINTS. IF EVERY POINT NOT TO BE PLOTTED.
100 IF(NSKIP) 101, 102
101 CALL SKPREC(ITAPE,NSKIP).
102 GO TO (999,186,186,185)
103 185 GO TO (999,186,12,999,12,999)IND
104 186 CONTINUE
105 IF(JMT=EQ.13) GO TO 852
106 IF(JMT=EQ.14) GO TO 852
107 C CHECK IF DATE IS WITHIN SPECIFIED INTERVAL
108 C IFLAG IS A FLAG TO ALLOW SKIPPING THE FIRST CALL TO FIND
109 C IF WE HAVE ALREADY FOUND THE STARTING DATE
110 IF(IFLAG .NE. 0) GO TO 82
111 CALL FIND(ISTD,ISTMB,ISTHM,JDA,JMB,JYR,JHM,INDK)
112 IF(INDK.EQ.1) GO TO 310
113 IFLAG=1
114 82 CONTINUE
115 IF(IVENR.EQ.0) GO TO 851
116 CALL FIND(IENDA,IENMB,IENHM,JDA,JMB,JYR,JHM,INDK)
117 IF(INDK.EQ.1) GO TO 995
118 851 CONTINUE
119 852 CONTINUE
120 C CHECKING IF DATA WITHIN CHART BOUNDS
121 NSTP=LCNT
122 IF(RTSP=RLAT) 100,100,96
123 100 IF(RLAT=RBST) 100,88,88
124 88 IF(RLANG=RLNG) 100,90,90
125 90 IF(RRIGT=RLONG) 100,100,92
126 92 GO TO 35
127 C DATA WITHIN BOUNDS
128 99 GO TO 35
129 C DATA OUTSIDE OF BOUNDS
130 C WRITING OUT DATE IF DATA POINT IS OUT OF BOUNDS
131 100 INIT=1
132 101 IF(ISH(9)) 410,310,410
133 410 WRITE(IIOUT,420)JDA,JMB,JYR,JHM
134 420 FORMAT(1868,1312,1X,14)
135 48 GO TO 310
136 995 WRITE(IIOUT,996)JDA,JMB,JYR,JHM
137 996 FORMAT(16 END DATE PASSED:2X,312,1X,14)
138 401 GO TO 12
139 C ERROR MESSAGES IF MISTAKE IN TAPE FILE OR RECORD SPACING.
140 WRITE(IIOUT,998) IND
141 998 FORMAT(1ERROR IN SKPREC, IND=I,12)
142 997 WRITE(IIOUT,997) IND
143 994 FORMAT(1ERROR IN SKPFILE, IND=I,12)
144 CALL EXIT
145 END
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE</td>
<td>1324</td>
<td>0052C</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>44</td>
<td>0002C</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>7</td>
<td>00007</td>
</tr>
<tr>
<td>TEMPS</td>
<td>3</td>
<td>00003</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>1378</td>
<td>00562</td>
</tr>
</tbody>
</table>

(NO MEMORY PROTECTION)

(PLUS BLANK COMMON)
1. 1 FORMAT('X', 'PROGRAM CONV67 VERSION 17 JAN 74')
2. PROGRAM CONV67
3. VERSION 17 JAN 74 TO SPEED THINGS UP BY DECODING FEWER ITEMS
4. PROGRAM TO CONVERT TO 1967 GRAVITY SYSTEM
5. AND NEW GEODETIC REFERENCE
6. ORIGINAL VERSION 1 DEC 73 BY LEE GAVE
7. DIMENSION IBUFIN(32,50,2), IBUFOT(32,50,2)
8. DIMENSION IA(35)
9. WRITE (108,1)
10. IRE=2
11. ITAPE=1
12. JTAPE=2
13. KTAPE=108
14. IFLIP=1
15. JFLIP=1
16. KFLIP=1
17. LAT=0
18. NIN=50
19. NBUT=0
20. IBUTSW=0
21. DEGRA=1.745329E+2
22. KI=1; KO=2
23. C BUFFER LOGIC FOR I/P
24. CALL BUFF IN(ITAPE=0, IBUFIN(1,1,1FLIP),1600)
25. CONTINUE
26. IF(NIN.LT.50) GO TO 90
27. CONTINUE
28. CALL ICHECK(ITAPE,KEY,HI)
29. GO TO (20,50,30,40) IKEY
30. CALL BUFFER IN(1/P: I/P) IE8D=C
31. CONTINUE
32. GO TO 15
33. GO TO (20) IE8D=1
34. CONTINUE
35. NIN=50
36. IF(IE8D.NE.1) CALL BUFFER IN(ITAPE=O, IBUFIN(1,1,1FLIP),1600)
37. C INPUT LOGIC
38. CONTINUE
39. C EQIT LOGIC
40. CONTINUE
41. IF (NI.EQ.1600) GO TO 95
42. GO TO 60 PROCESSING
43. NIN=50
44. GO TO 999
45. CONTINUE
46. DECIDE(72,1001, IBUFIN(1,1, NIN, NFLIP)) ND IREC1,196RC,KGDA,KGMB,
47. 1 KGYR,KGWM,DLAT,DLONG,ELEV,K977,BSG,1DIP,FA,AG,TC,IE5C
48. C EDIT LOGIC
49. C
CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF THE 1967 INTERNATIONAL GRAVITY FORMULA AND NEW GEODETIC REFERENCE SYSTEM

CALL OBG(K977,OBSS,OBSS,K1)
OBSS=OBSS-14.0
CALL OBG(K977,OBSS,OBSS,K0)
RLAT=DLAT*DEGRA
DG3=2*(13*8*(SIN(ABS(RLAT))**2))
IF (FA<LT.99G) FT=FA+CG
IF (BG<LT.99G) BG=BG+DG
100 CONTINUE

OUTPUT LOGIC
300 CONTINUE
NOUT=NOUT+1
G0=301 I1932,1
IBUF=IBUF(NOUT,NFLIP)=IBLFIN(I,NIN,NFLIP)
301 CONTINUE
ENCOD(72,1001,IBUF,IBUF,FLIP,NOUT,F,NFLIP,NFLIP)
ENCYR,KGHx,DLAT,DLOG,ELEV,K977,OBSS,DEP,FA,BG,TC,IELC
305 CONTINUE
IF(NOUT<LT.50) G0 TO 10
309 CONTINUE
BUFFER LOGIC FOR B/P
310 CONTINUE
IF(IOUTMNE.1) IOUTM1) G0 TO 350
JKEY=JCHECK(JTAPE)
G0 TO (320.350.330.340) JKEY
320 OUTPUT WAITING FOR B/P / IEBD=0
G0 TO 310
330 OUTPUT ENC. OF FILE JTAPE / IEBD=1
G0 TO 999
340 OUTPUT 'BUFF OUT ERROR! / IEBD=1
G0 TO 999
350 CONTINUE
NOUT=0
KFLIP=JFLIP
JFLIP=3+JFLIP
CALL BUFF OUT(JTAPE,O,IBUFST(1,1,JFLIP),1600)
102 G0 TO 10
103 C END OF JOB
104 C
105 C
106 C CONTINUE
107 C CONTINUE
108 JKEY=JCHECK(JTAPE)
109 G0 TO (920.950.930,940) JKEY
110 C OUTPUT WAITING FOR B/P / IEBD=0
111 G0 TO 910
112 C OUTPUT BAD JKEY / IEBD=1
113 G0 TO 960
114 OUTPUT 'BUFF OUT ERROR! / IEBD=1
115 G0 TO 960
116 C CONTINUE
117 JWDSS=NOUT+32
118 CALL BUFF OUT(JTAPE,0,IBUFST(1,1,JFLIP),JWD)
119 C CONTINUE
120. END FILE JTAPE
121. OUTPUT 'ALL DONE'
122. C FORMATS
123. C
124. 98 FORMAT(I4,32A4)
125. 1001 FORMAT(I1,14,312,14,2F9.4,F7.2,13,F4.3,15,2F6.1,F4.1,I2)
127. END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>R</td>
<td>SPRAG</td>
<td>000000</td>
<td>0</td>
<td>BLUFFOUT</td>
<td>R</td>
<td>SPRAG</td>
<td>000000</td>
<td>0</td>
<td>DLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>01930</td>
<td>0</td>
</tr>
<tr>
<td>FA</td>
<td>R</td>
<td>SCALAR</td>
<td>01943</td>
<td>1</td>
<td>I</td>
<td>I</td>
<td>ARRAY</td>
<td>01900</td>
<td>35</td>
<td>ICHECK</td>
<td>I</td>
<td>SPRAG</td>
<td>01946</td>
<td>1</td>
</tr>
<tr>
<td>IA</td>
<td>I</td>
<td>ARRAY</td>
<td>01900</td>
<td>35</td>
<td>IKEY</td>
<td>I</td>
<td>SCALAR</td>
<td>01932</td>
<td>4</td>
<td>IKEY</td>
<td>I</td>
<td>SCALAR</td>
<td>01932</td>
<td>4</td>
</tr>
<tr>
<td>ICHECK</td>
<td>I</td>
<td>SPRAG</td>
<td>01946</td>
<td>1</td>
<td>IREC2</td>
<td>I</td>
<td>SCALAR</td>
<td>01923</td>
<td>1</td>
<td>JFLIP</td>
<td>I</td>
<td>SCALAR</td>
<td>01928</td>
<td>1</td>
</tr>
<tr>
<td>IREC</td>
<td>I</td>
<td>SCALAR</td>
<td>01934</td>
<td>1</td>
<td>JDNS</td>
<td>I</td>
<td>SCALAR</td>
<td>0194C</td>
<td>1</td>
<td>KGNS</td>
<td>I</td>
<td>SCALAR</td>
<td>0193E</td>
<td>1</td>
</tr>
<tr>
<td>KI</td>
<td>I</td>
<td>SCALAR</td>
<td>01930</td>
<td>1</td>
<td>K977</td>
<td>I</td>
<td>SCALAR</td>
<td>01940</td>
<td>1</td>
<td>KI</td>
<td>I</td>
<td>SCALAR</td>
<td>01930</td>
<td>1</td>
</tr>
<tr>
<td>NA</td>
<td>I</td>
<td>SCALAR</td>
<td>01939</td>
<td>1</td>
<td>NOUT</td>
<td>I</td>
<td>SCALAR</td>
<td>0192D</td>
<td>1</td>
<td>SBG</td>
<td>R</td>
<td>SPRAG</td>
<td>01948</td>
<td>1</td>
</tr>
<tr>
<td>BLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>01948</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Local Variables (6477 Words):**

- 000000 IBUFF
- 01926 KTAPE
- 01926 IN
- 01932 IKEY
- 01993 ESC
- 01933 ELEV
- 0193B BG
- 01944 IBUFF
- 01944 ICHECK
- 01944 BLUFFOUT
- 01944 SBG
- 01944 JDNS
- 01944 SCALAR
- 01944 INT
- 01944 COMMON
- 01944 SUBPROG
- 01944 EXTERNAL

**Blank Common (0 Words)**

**Intrinsic Subprograms Used:**

- ABS
- SIA

**External Subprograms Required:**

- IBUFF
- BLUFFOUT
- ICHECK
- SBG
- JDNS
- SCALAR
- INT
- COMMON
- SUBPROG
- EXTERNAL
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE</td>
<td>415</td>
<td>0019F</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>7</td>
<td>00007</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>6477</td>
<td>0194D</td>
</tr>
<tr>
<td>TEMPS</td>
<td>1</td>
<td>00001</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>6900</td>
<td>01AF4</td>
</tr>
</tbody>
</table>
1. PROGRAM CR2G
2. VERSION OF 20 MAR 1973 TO ADD INPUT OF ELEV AND G METER CODES
3. MODIFICATION BY C. BOWIN
4. OUTPUT "PGM CR2G OF 20 MAR 1973"
5. May 72 - BY S. ABBOT TO CORRECT OUTPUT LISTING, AND TO
6. OUTPUT GSUM DATA WITH LAT AND LON IN DECIMAL DEGREES
7. ALSO TO CLEAN UP THE COMMENTS AND SO ON
8. MOD 18 JAN 71 - S. ABBOT TO CORRECT
10. INPUT
11. JOB INITIALIZATION
12. 1) GRAVITY METER TABLES == 70 CARDS
13. 2) SENSE SWITCH CARD == 125 CARDS SSX(0) - CC 80
14. 3) IGM(1), IGM(2), DRFTC, LSRC (2A,2X,F10.5,15)
15. IGM == NUMBER AND/0R MODEL OF GRAVITY METER USED
16. FOR INSTANCE LSRC = 18
17. DRFTC == DRIFT CORRECTION FOR GRAVITY METER == F10.5
18. LSRC == SOURCE CODE FOR GSUM OUTPUT DATA
19. NOTE IF DRFTC = 0 THEN THE ASSUMED VALUE FOR THE GRAVITY METER
20. DRIFT (ASDFC) WILL BE USED, UNLESS SSX(5) IS ON
21. NOTE IF IGM(1) AND IGM(2) ARE BOTH BLANK, THE GRAVITY METER TYPE
22. WILL BE SET TO THE DEFAULT TYPE OF 'LSR G=18'
23. NOTE: IF LSRC = 0 THE SOURCE CODE WILL BE SET TO THE DEFAULT
24. VALUE == LSRC
25. THESE ARE FOLLOWED BY GROUPS OF INDIVIDUAL STATION COUNTER
26. READING CARDS. EACH GROUP IS HEADED BY 3 CARDS:
27. 1) THE ABSOLUTE GRAVITY VALUE FOR THE REFERENCE STATION
28. BASEG(1), BASEG(2)
29. 2) THE CRUSTAL DENSITY IN GM PER CM TO BE USED IN
30. THE PRODUCTION OF THE BOURGEOIS ANOMALY == F10.2 dense
31. 3) THE COUNTER READING CARD FOR THE REFERENCE STATION
32. FORMAT FOR COUNTER READING DATA IS THAT OF 17 MAY 1966
33. THESE ARE FOLLOWED BY COUNTER READING CARDS FOR THE REST OF THE
34. MEASUREMENTS THAT ARE TO BE TIED TO THE REFERENCE MEASUREMENT.
35. A COUNTER READING CARD WITH ALL ZEROS EXCEPT FOR THE
36. YEAR VALUE (CC 9,10) WILL CAUSE THE PGM TO BRANCH TO READ
37. NEW CARDS FOR BASEG AND DENSE AND THE REF STATION
38. A CARD WITH ALL ZEROS INCLUDING YEAR WILL GO TO E.O.J.
39. DO NOT HAVE AN ALL-ZEROS CARD FOLLOWING A CARD WITH ONLY ONE YEAR.
40. SENSE SWITCH OPTIONS
41. SSX(1) OFF, FOR PRINTED OUTPUT OF COMPUTED VALUES FOR EACH STATION
42. ON, FOR SUPPRESSION OF PRINTED OUTPUT
43. SSX(2) OFF, TO PUNCH OUTPUT TO INPUT TO GRAVITY DESCRIPT. PGM, (GDS)
44. ON, TO SUPPRESS PUNCHED OUTPUT
45. SSX(4) OFF, TO OUTPUT GSUM FORMATTED DATA TO 'TAPE
46. ON, TO SUPPRESS GSUM FORMAT OUTPUT
47. SSX(5) OFF, FOR GRAVITY METER DRIFT CORRECTION
48. ON, FOR SUPPRESSION OF DRIFT CORRECTION
49. CR2G010
50. CR2G020
51. CR2G030
52. CR2G040
53. CR2G050
54. CR2G060
55. CR2G070
56. CR2G080
57. CR2G090
58. CR2G010
59. CR2G020
**VARIABLE DEFINITIONS**

- **LDT, LDY, LYT, TIME** - DATE AND TIME OF READING. IF LOCAL TIME IS USED, KTZ SHOULD ALSO BE ENTERED. IF GMT, KTZ ALWAYS = 0.
- **HD, HLD, HMT** - DATE AND TIME OF READING. IF GMT (OR LOCAL IF KTZ = 99).
- **IDAT, IMET, IYT** - DATE AND TIME OF REFERENCE STATION READING (IN GMT OR LOCAL IF KTZ = 99).
- **CRN** - GRAVITY METER COUNTER READING (IN UNITS).
- **ELEV** - ELEVATION OF GRAVITY METER (IN METERS).
- **KTZ** - THE TIME ZONE CORRECTION. IF KTZ = 99, IT INDICATES THAT KTZ WAS NOT AVAILABLE. THUS, CLS AND HONK = 0.
- **DATE/TIME** - DATE AND TIME OF READING (IN GMT). IF LOCAL TIME IS USED, KTZ SHOULD ALSO BE ENTERED. IF GMT, KTZ ALWAYS = 0.
- **KD, KG, KHM** - LOCAL TIME OF READING. IF GMT, KTZ IS ALWAYS = 0.
- **DIAT, DMT** - DATE AND TIME OF REFERENCE STATION READING (IN GMT OR LOCAL IF KTZ = 99).
- **ASDF** - ASSUMED DRIFT FOR GRAVITY METER.
- **BASEG** - TOTAL FIELD GRAVITY VALUE AT STATION OF REFERENCE. THE VALUE IS READ IN WITH A FORMAT OF F3.0, F6.2 FROM WHICH THE BASE FOR INPUT AND BASE7 FOR COMPUTATION IS FORMED.
- **VALM** - GRAVITY METER CALIBRATION TABLES ARRAY. THIS IS THE TABLE USED TO LOOK-UP OR CONVERT A GRAVITY METER COUNTER READING TO AN EQUIVALENT RELATIVE MILLIGAL VALUE. THE COUNTER READING IS READ WITH A FORMAT OF F8.3, FOR EXAMPLE 3572.256.

**INITIALIZATION**

- **NOTE**: LOCAL = KTZ * GMT. I.E. VALUES WEST OF GREENWICH ARE PLUS.
- **DESC**: DESCRIPTION OF GRAVITY STATION SITE IN ALPHABETIC FORMAT.
- **CRFTCD** - THE CORRECTION FACTOR FOR DRIFT OF THE GRAVITY METER. IF DRFTCD * 0, WE ASSUME A DRIFT RATE OF 0.003 MGAL/DAY.
- **ISN(5)** - THE PROGRAM DOES NOT MAKE DRIFT CORRECTION.
- **ADFT** - ASSUMED DRIFT FOR GRAVITY METER.
- **NOTE**: IF THE GRAVITY METER DRIFT IS NEGATIVE, THE CORRECTION FOR DRIFT IS A NEGATIVE NUMBER.
- **BASEG7** - TOTAL FIELD GRAVITY VALUE AT STATION OF REFERENCE. THE VALUE IS READ IN WITH A FORMAT OF F3.0, F6.2 FROM WHICH THE BASE FOR INPUT AND BASE7 FOR COMPUTATION IS FORMED.
- **VALM** - GRAVITY METER CALIBRATION TABLES ARRAY. THIS IS THE TABLE USED TO LOOK-UP OR CONVERT A GRAVITY METER COUNTER READING TO AN EQUIVALENT RELATIVE MILLIGAL VALUE. THE COUNTER READING IS READ WITH A FORMAT OF F8.3, FOR EXAMPLE 3572.256.
DIMENSION KDATE(4)
DIMENSION VALM(70),BASEG(2),DESC(31),IGM(2)
DOUBLE PRECISION DTD
DOUBLE PRECISION DEC,RLAT,RAD,RLNG
DATA WEST,SOUTH/1H,1S,1/
* ASSUMED VALUES FOR L&R G=18 METER
DATA IBLNK,ILR,IG18/1,ILR,IG18/1
ASDFT = .003
* GET DATE OF RUN
CALL TODAY(KDATE)

IN=105
IN=108
JTAPE = 106
ICOUNT = 0
IPAGE = 0
IREC = 1
IDEP = 0
RFA = 0.0
IREGC = 0
TCORR=99.9
LELC=10
LGC=01
IFGC=3
IFBC=0

READ IN GRAVITY METER CALIBRATION TABLE
DO 210 K = 1,70
READ (IN=150) J,TABLE
VALM(J)=TABLE
210 CONTINUE

* INITIALIZE SENSE SWITCHES
INN = (IST=2)

READ GRAVITY METER TYPE, DRIFT VALUE, AND SOURCE CODE VALUE
IF VALUE READ FROM CARD IS 0, USE ASDFT UNLESS SS(N) IS ON.
READ (IN=530) IGM(1),IGM(2),DRFTCO,LSRC,IELC,IGC
IF(DRFTCO.EQ.0) DRFTCO=ASDFT
IF(IN=5*EQ.1) DRFTCO=0.0
IF (IGM(1).EQ.IBLNK.AND.IGM(2).EQ.IBLNK)
1 IGM(1) = ILR; IGM(2) = IG18
IF (LSRC .EQ. 0) LSRC = LSRC
IF(IELC.EQ.0) IELC=LELC
IF(IGC.EQ.0) IGC=LGC

* WRITE OUT JOB INITIALIZATION VALUES
C WRITE (1OUT,5200) IPAGE,KDATE
C WRITE (1OUT,5285) IGMA(1),IGMA(2),DRFTCO,LSRC,IELC,IGC
DB 29% II = 1,70
C WRITE (1OUT,5290) II,VALM(II)
C 29% CONTINUE
C************
C* COMPUTATION OF OBSERVED GRAVITY AND ANOMALIES*
C* BEGIN A GROUP OF MEASUREMENTS*
C************
C* READ BASE GRAVITY*
C AND CONVERT INTO UNITS COMPATIBLE WITH BOTH SYSTEMS
C 300 CONTINUE
C READ ((IN,5320,END=910),BASEG(1),BASEG(2))
C READ ((IN,5330,END=910),DENSE)
C BASEG=((BASEG(1)*9770)+10000)*BASEG(2)
C BASE=BASEG(1)
C 300 CONTINUE
C READ COUNTER READING CARDS FOR INDIVIDUAL STATIONS
C THE FIRST CARD READ IS THE ONE FOR THE REFERENCE STATION
C DRIFT IS COMPUTED FROM DATE ON THIS FIRST CARD.
C ALL OTHER READINGS ARE REFERENCED TO THIS MEASUREMENT.
C
C DB 890 I = 1,9000
C READ ((IN,5405,END=910),1,LSTAT,LDAY,MB,LYR,LTME,CRN,LAT,RLATM,SBRN,LONG,
C IPCS=0
C CLS=.99
C MBNK=.99
C CR=CRN
C NCR=CRN
C
C* CHECK TO SEE IF HAVE NEW REFERENCE STATION OR CALL EXIT
C (STMT #180 IS EXIT) #300 IS START OF NEW GROUP
C
C IF(NCR)500*500=20
C 500 IF(LYR)415=180*415
C 415 I=1
C GBT8 300
C
C* SEARCH TABLES FOR GRAVITY VALUE CORRESPONDING TO COUNTER READING
C
C 420 TURC=CR*0.01
C LC=TURC
C CI=LC
C CI=CI+100
C C2=CR+CI
C RELM=VALM(LC)+(C2*01*(VALM(LC+1)-VALM(LC)))
C
C* CALCULATE LAT AND LON IN RADIANS AND IN DECIMAL DEGREES
C
C RDEG=LAT
C DEC=RLATM*1.666666E-2
240. \text{PLAT=RDEG*DEC} \\
241. \text{CLAT = RLAT} \\
242. \text{RAD=PLAT*1.7453293D-2} \\
243. \text{PLAT=RAD} \\
244. \text{C} \\
245. \text{RDEG*LONG} \\
246. \text{DEC=RLOM*1+666666666D-2} \\
247. \text{RLONG=RDEG + DEC} \\
248. \text{DLON = RLONG} \\
249. \text{RLONG*RLONG*1+7453293 D=2} \\
250. \text{C} \\
251. \text{• IF KTZ IS EQUAL TO 99 MEANS HAVE NOT MADE OR LOOKED UP THE} \\
252. \text{TIME ZONE CORRECTION; THERE THE FOLLOWING CALCULATIONS} \\
253. \text{ARE NOT NEEDED BECAUSE WE CANNOT CALCULATE THE TIDAL OR} \\
254. \text{HOMMLE CORRECTIONS WITHOUT IT} \\
255. \text{C} \\
256. \text{IF(KTZ,NE*99) GO TO 610} \\
257. \text{KGDAL=LDAY} \\
258. \text{KGMB=HB} \\
259. \text{KGYR=LYR} \\
260. \text{KGMH=LTIME} \\
261. \text{I*D} \\
262. \text{GOTO 630} \\
263. \text{C} \\
264. \text{UNTINUE} \\
265. \text{KTT=KTZ} \\
266. \text{CALL CHHKT(LDAY,MB,LYR,LTIME,KTT,KGDA,KGMB,KGYR,KGMH,KTZ)} \\
267. \text{ADAY = KGDA} \\
268. \text{ADAY = ADAY / 2400} \\
269. \text{ADAY = ADAY + FLOAT (I)} \\
270. \text{C} \\
271. \text{NORTH LAT OR EAST LON IS POSITIVE} \\
272. \text{SOUTH LAT OR WEST LON IS NEGATIVE} \\
273. \text{C} \\
274. \text{IF (RLAT) 62006192620} \\
275. \text{619 IF (RLONG) 620,630,620} \\
276. \text{CONTINUE} \\
277. \text{C} \\
278. \text{IF(SBRR,NE=SUBTH) RLAT=RLAT; DLAT=DLAT} \\
279. \text{IF(MORE=EQ=WEST) RLONG=RLONG; DLON=DLON} \\
280. \text{C} \\
281. \text{• CALCULATE CLS AND HONK VALUES} \\
282. \text{C} \\
283. \text{IHR=KGMH/100} \\
284. \text{IMIN=KGMH/IHR/100} \\
285. \text{CALL TJDAL(RLAT,RLONG,KGYR, IHR,IMIN,CLS,HONK,DTD)} \\
286. \text{RELH=RELH+CLS+HONK} \\
287. \text{C} \\
288. \text{630 CONTINUE} \\
289. \text{C} \\
290. \text{IF (I=1) 440*440*450} \\
291. \text{C} \\
292. \text{• NEW REFERENCE STATION (* FIRST CARD OF GROUP) PROCESSING} \\
293. \text{C} \\
294. \text{440 REF=RELH} \\
295. \text{IDA1=KGDAL} \\
296. \text{IMB1=KGMB} \\
297. \text{IYR1=KGYR} \\
298. \text{ITM1=KGMH} \\
299. \text{LSTA=LSTAT}
1PCSV1
LCNT = 0
IMAGE = 1

C CALCULATE DRIFT

C 450 DIFFR=REL=REF
CALL CDATE(IDA,IM1,IMR1,ITM1,KGDA,KGMB,KGYR,KGHM,TIMD)
DIFFR=(TIMD/240)*DFRTC

C GET G8BS7=0.0 DIFFR=DRIFT
ICOUNT=ICOUNT+1

C PUT G8BS7 = 977000: INTO OUTPUT UNITS
ICON=G8BS7/1000.
TEMP=ICON=1000
G8BS=G8BS7=TEMP

C COMPUTE FREE-AIR AND BOUGUER ANOMALIES
X=R*2.*RAD
C8=G8BS7(x)
F8ELV=((0.35065+0.00022*C2R)*ELEV)*((ELEV+0.001)**2)*0.072
RA1=RAD
GFR8E=G8BS7=INTF(RA1)+F8ELV
BELEV=0.04185*DENSE*ELEV
G8BUG=G8FREE=BELEV

C CHECK TO SEE IF LAT AND LON = 0
58 IF ELEV = 0
59 IF THEY DO SET GFREE AND G8BUG EQUAL TO 999.0

C IF(LAT) 2050,2051,2050
2051 IF(RLATM) 2050,2051,2050
2052 IF(LAT) 2050,2052,2050

C IF(LONG) 2050,2051,2050
2051 IF(RLONG) 2050,2051,2050
2052 IF(LONG) 2050,2052,2050

C IF (ELEV) 2054,2053,2054
2053 GFREE=999.0
2054 G8BUG=999.0
2055 CONTINUE

C NSTATNLSTAT

C COMPUTE GSUM SORT KEY FIELDS
PLAT = DLAT + 90
LAT8 = PLAT
PLON = DLON + 180
LONG = PLON

C ***************

C OUTPUT THE DESIRED INFORMATION
C * PUNCH OUTPUT FOR INPUT TO STATION DESCRIPTION PROGRAM
361* C IPCS PREVENTS US FROM PUNCHING THE REFERENCE STATION WHEN
362* IT IS THE FIRST CARD OUT
363* IF (IPCS.EQ.1) GOTO 835
364* IF (ISW(2)) 835,825,835
420* WRITE (IOUT,5001)  
421* END  
422* IF(ISW(14)*GT(0)) GO TO 24  
423* ENDFILE KTAPE  
424* 5950 WRITE (IOUT,5950)  
425* REWIND KTAPE  
426* 24 CONTINUE  
427* IPAGE = 0  
428* WRITE (IOUT,5200) IPAGE,KDATE  
429* STOP  
430* C  
431* C ************************************  
432* C * FORMATS  
433* C * C ************************************  
434* C  
435* C 5001 FORMAT (1H1)  
436* 5150 FORMAT (12,F7*2)  
437* 5200 FORMAT (T2!PAGE!14,T3!DATE OF RUN = '14A4)  
438* 5230 FORMAT (2A4,2X,F10.5,5X,SOURCE CODE = '14)  
439* 5285 FORMAT (1 'ELEV CODE = '14,5X,METER CODE = '14)  
440* 5290 FORMAT (1 'TABLE 15(12,14,F7*2))  
441* 5320 FORMAT (F3*0,F6*2)  
442* 5330 FORMAT (F*2)  
443* 5405 FORMAT (14!312,14,F8*3,12,F5*2,A1,13,F5*2,A1,F7*1,13,F1A1)  
444* 5475 FORMAT(14!312,14,F8*3,12,F5*2,A1,F7*1)  
445* 5595 FORMAT ('/REFERENCE STATION */15,10X/READING OF '13(12/1)'14,  
446* 1 5X,METER */2A4,5X/SOURCE CODE */14)  
447* 5653 FORMAT (T2!*** NEW REFERENCE STATION ***/1)  
448* 1 T2(**********************************************************************)  
449* 5856 FORMAT ('REFERENCE GRAVITY */'13,F6*2,7X/REL MGAL */'F11.3,  
450* 1 5X/DENSE */F5*2,9X/DRFTCG */F6*4/)  
451* 5858 FORMAT ('STATION */10X/DATE */10X/TZ */14X/LATITUDE */2X/CTR RDNG,  
452* 1 4X/GPREE */3X/CLS */5X/GOFF */12X/DESCRIPTION */1)  
453* 5865 FORMAT ('/REFERENCE GRAVITY */13,F6*2,7X/REL MGAL */'F11.3,  
454* 1 5X/DENSE */F5*2,9X/DRFTCG */F6*4/)  
455* 5868 FORMAT ('/REFERENCE GRAVITY */13,F6*2,7X/REL MGAL */'F11.3,  
456* 1 5X/DENSE */F5*2,9X/DRFTCG */F6*4/)  
457* 5869 FORMAT ('/REFERENCE GRAVITY */13,F6*2,7X/REL MGAL */'F11.3,  
458* 1 5X/DENSE */F5*2,9X/DRFTCG */F6*4/)  
459* 5871 FORMAT (11,14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
460* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
461* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
462* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
463* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
464* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
465* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
466* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
467* 1 14,F3*0,F4*2,F3*0,F6*2,F5*3,F4*2,F4*2)  
468* C  
469* END
**LOCAL VARIABLES (221 WORDS):**

<table>
<thead>
<tr>
<th>00000</th>
<th>KDATE</th>
<th>00004</th>
<th>VALM</th>
<th>00004A</th>
<th>BASEG</th>
<th>00004C</th>
<th>DESC</th>
<th>00006B</th>
<th>IGM</th>
<th>00006D</th>
<th>DTD</th>
</tr>
</thead>
<tbody>
<tr>
<td>000070</td>
<td>DEC</td>
<td>000072</td>
<td>Rlat</td>
<td>00007A</td>
<td>RAD</td>
<td>000076</td>
<td>RLNG</td>
<td>000078</td>
<td>WEST</td>
<td>000079</td>
<td>SOUTH</td>
</tr>
<tr>
<td>00007A</td>
<td>IBLNK</td>
<td>00007B</td>
<td>ILR</td>
<td>00007C</td>
<td>I18</td>
<td>00007D</td>
<td>LSRCB</td>
<td>00007E</td>
<td>ASDFT</td>
<td>00007F</td>
<td>INN</td>
</tr>
<tr>
<td>000080</td>
<td>IBUT</td>
<td>000081</td>
<td>JTAPE</td>
<td>000082</td>
<td>KTAPE</td>
<td>000083</td>
<td>ICOUNT</td>
<td>000084</td>
<td>IPAGE</td>
<td>000085</td>
<td>IREC</td>
</tr>
<tr>
<td>000086</td>
<td>IDSP</td>
<td>000087</td>
<td>RPA</td>
<td>000088</td>
<td>IREGC</td>
<td>000089</td>
<td>TCORB</td>
<td>00008A</td>
<td>LELC</td>
<td>00008B</td>
<td>LGC</td>
</tr>
<tr>
<td>00008C</td>
<td>PPC</td>
<td>00008D</td>
<td>IFBC</td>
<td>00008E</td>
<td>K</td>
<td>00008F</td>
<td>J</td>
<td>000090</td>
<td>TABLE</td>
<td>000091</td>
<td>INN</td>
</tr>
<tr>
<td>000092</td>
<td>DRFTF</td>
<td>000093</td>
<td>LSRC</td>
<td>000094</td>
<td>IELC</td>
<td>000095</td>
<td>IGC</td>
<td>000096</td>
<td>II</td>
<td>000097</td>
<td>DENSE</td>
</tr>
<tr>
<td>000098</td>
<td>BASG7</td>
<td>000099</td>
<td>IBASE</td>
<td>00009A</td>
<td>I</td>
<td>00009B</td>
<td>LSTAT</td>
<td>00009C</td>
<td>LDAY</td>
<td>00009D</td>
<td>MB</td>
</tr>
<tr>
<td>00009E</td>
<td>LRY</td>
<td>00009F</td>
<td>LTIME</td>
<td>0000A0</td>
<td>CRN</td>
<td>0000A1</td>
<td>LAT</td>
<td>0000A2</td>
<td>RLATM</td>
<td>0000A3</td>
<td>SBRN</td>
</tr>
<tr>
<td>0000A4</td>
<td>LSN</td>
<td>0000A5</td>
<td>LBM</td>
<td>0000A6</td>
<td>MORE</td>
<td>0000A7</td>
<td>ELEV</td>
<td>0000A8</td>
<td>KTZ</td>
<td>0000A9</td>
<td>1K</td>
</tr>
<tr>
<td>0000AA</td>
<td>IPCS</td>
<td>0000AB</td>
<td>CLS</td>
<td>0000AC</td>
<td>HBNK</td>
<td>0000AD</td>
<td>CR</td>
<td>0000AE</td>
<td>NCR</td>
<td>0000AF</td>
<td>TCR</td>
</tr>
<tr>
<td>0000B0</td>
<td>LC</td>
<td>0000B1</td>
<td>CI</td>
<td>0000B2</td>
<td>C2</td>
<td>0000B3</td>
<td>RELM</td>
<td>0000B4</td>
<td>RDEG</td>
<td>0000B5</td>
<td>DLT</td>
</tr>
<tr>
<td>0000B6</td>
<td>DLLN</td>
<td>0000B7</td>
<td>KGDA</td>
<td>0000B8</td>
<td>KGM</td>
<td>0000B9</td>
<td>KGVR</td>
<td>0000BA</td>
<td>KGHM</td>
<td>0000CA</td>
<td>IMIN</td>
</tr>
<tr>
<td>0000BB</td>
<td>KKT</td>
<td>0000BD</td>
<td>NTZ</td>
<td>0000BE</td>
<td>ADAY</td>
<td>0000BF</td>
<td>IHR</td>
<td>0000BC</td>
<td>IM</td>
<td>0000CD</td>
<td>REF</td>
</tr>
<tr>
<td>0000C2</td>
<td>IDA1</td>
<td>0000C3</td>
<td>IM01</td>
<td>0000C4</td>
<td>IYR1</td>
<td>0000C5</td>
<td>JTM1</td>
<td>0000C6</td>
<td>LSTA</td>
<td>0000C7</td>
<td>LCNT</td>
</tr>
<tr>
<td>0000C8</td>
<td>DIFFR</td>
<td>0000C9</td>
<td>TMD</td>
<td>0000CA</td>
<td>DRFT</td>
<td>0000CB</td>
<td>GBS7</td>
<td>0000CC</td>
<td>ICBN</td>
<td>0000CD</td>
<td>TEMP</td>
</tr>
<tr>
<td>0000CE</td>
<td>GBS</td>
<td>0000CF</td>
<td>IG</td>
<td>0000D0</td>
<td>X</td>
<td>0000D1</td>
<td>CBR</td>
<td>0000D2</td>
<td>FELEV</td>
<td>0000D3</td>
<td>RA1</td>
</tr>
<tr>
<td>0000D4</td>
<td>FREE</td>
<td>0000D5</td>
<td>BELAY</td>
<td>0000D6</td>
<td>GBBUG</td>
<td>0000D7</td>
<td>NSTATN</td>
<td>0000D8</td>
<td>PLAT</td>
<td>0000D9</td>
<td>LTKEY</td>
</tr>
</tbody>
</table>

**BLANK COMMON (0 WORDS)**

**INTRINSIC SUBPROGRAMS USED:***

| CBS | FLOAT |

**EXTERNAL SUBPROGRAMS REQUIRED:***

<table>
<thead>
<tr>
<th>CDATE</th>
<th>CHKMT</th>
<th>GINTF</th>
<th>ISW</th>
<th>M2D</th>
<th>TIDAL</th>
<th>TODAY</th>
<th>F1101</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1102</td>
<td>F1103</td>
<td>F1104</td>
<td>F1105</td>
<td>F1106</td>
<td>F1108</td>
<td>MID8</td>
<td>MIDC</td>
</tr>
<tr>
<td>95CORCDEE</td>
<td>98CDREAD</td>
<td>98CWRIT</td>
<td>9CDS</td>
<td>9DTOR</td>
<td>9ENDFILE</td>
<td>9ENDI5L</td>
<td>9 INITIAL</td>
</tr>
<tr>
<td>915DATA</td>
<td>915LSOA</td>
<td>GITR</td>
<td>SPRINT</td>
<td>9REWIND</td>
<td>9RTBI</td>
<td>9STOP</td>
<td></td>
</tr>
</tbody>
</table>

**HIGHEST ERROR SEVERITY: 0 (NO ERRORS)**

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>00000</td>
</tr>
<tr>
<td>00001</td>
<td>00001</td>
</tr>
<tr>
<td>00002</td>
<td>00002</td>
</tr>
<tr>
<td>00003</td>
<td>00003</td>
</tr>
<tr>
<td>00004</td>
<td>00004</td>
</tr>
<tr>
<td>00005</td>
<td>00005</td>
</tr>
<tr>
<td>00006</td>
<td>00006</td>
</tr>
<tr>
<td>00007</td>
<td>00007</td>
</tr>
<tr>
<td>00008</td>
<td>00008</td>
</tr>
<tr>
<td>00009</td>
<td>00009</td>
</tr>
<tr>
<td>0000A</td>
<td>0000A</td>
</tr>
<tr>
<td>0000B</td>
<td>0000B</td>
</tr>
<tr>
<td>0000C</td>
<td>0000C</td>
</tr>
<tr>
<td>0000D</td>
<td>0000D</td>
</tr>
<tr>
<td>0000E</td>
<td>0000E</td>
</tr>
</tbody>
</table>

| GENERATED CODE | 1001 | 003E9 |
| CONSTANTS | 31 | 0001F |
| LOCAL VARIABLES | 221 | 0000D |
| TEMPS | 1 | 00001 |
| TOTAL PROGRAM | 1254 | 004E6 |
PROGRAM CR2G67

ORIGINAL VERSION 2 OCT 75
MOD OF CR2G TO CALCULATE IGSN 67 VALUES
VERSION OF 20 MAR 1973, TO ADD INPUT OF ELEV AND G METER CODES
MODIFICATION BY C. BOWIN
OUTPUT ' *** PGM CR2G OF 20 MAR 1973!

* MAY 72 -- BY S. ABBOTT TO CORRECT OUTPUT LISTING, AND TO
OUTPUT GSUM DATA WITH LAT AND LON IN DECIMAL DEGREES
ALSO TO CLEAN UP THE COMMENTS AND 80 BA
* MOD 18 JAN 74 -- S. ABBOTT TO CORRECT
********* REWRITTEN IN A HOPEFUL MANNER BY J. WOLFE JUNE E 1969

* INPUT
* JOB INITIALIZATION

1) GRAVITY METER TABLES == 70 CARDS
2) SENSE SWITCH CARD == 6011 (SSW(1) = CC 80)
3) IG(1), IG(2), DRTFCO, LSRC (2A+2X+10+5,15)
   IG == NUMBER AND/OR MODEL OF GRAVITY METER USED
   FOR INSTANCE == LS 8+18
   DRTFO == DRIFT CORRECTION FOR GRAVITY METER == F10+5
   LSRC == SOURCE CODE FOR GSUM OUTPUT DATA
   * NOTE: IF DRTFCO = ON, THE ASSUMED VALUE FOR THE GRAVITY METER
   DRIFT (ASDT) WILL BE USED, UNLESS SSW(5) IS ON
   * NOTE: IF IG(1) AND IG(2) ARE BOTH BLANK, THE GRAVITY METER TYPE
   WILL BE SET TO THE DEFAULT TYPE OF 1LS 8+18
   * NOTE: IF LSRC = 0, THE SOURCE CODE WILL BE SET TO THE DEFAULT
   VALUE == LSRC

* THESE ARE FOLLOWED BY GROUPS OF INDIVIDUAL STATION COUNTER
READING CARDS. EACH GROUP IS HEADED BY 3 CARDS;
1) THE ABSOLUTE GRAVITY VALUE FOR THE REFERENCE STATION --
   F300F6+= BASEG(1), BASEG(2)
   2) THE CRUSTAL DENSITY IN GM PER CU CM TO BE USED IN
   THE CALCULATION OF THE BAUGUER ANOMALY -- 4X2 = DENSE
   3) THE COUNTER READING CARD FOR THE REFERENCE STATION

FORMAT FOR COUNTER READING DATA IS THAT OF 17 MAY 1966

* THESE ARE FOLLOWED BY COUNTER READING CARDS FOR THE REST OF THE
MEASUREMENTS THAT ARE TO BE TIED TO THE REFERENCE MEASUREMENT.

* A COUNTER READING CARD WITH ALL ZEROS EXCEPT FOR THE
YEAR VALUE (CC 910) WILL CAUSE THE PGM TO BRANCH TO READ
NEW CARDS FOR BASED AND DENSE AND THE REF STATION
A CARD WITH ALL ZEROS INCLUDING YEAR WILL GO TO E000.
DO NOT HAVE AN ALL-ZEROS CARD FOLLOWING A CARD WITH ONLY THE YEAR.

* SENSE SWITCH OPTIONS

SSW(1) OFF FOR PRINTED OUTPUT OF COMPUTED VALUES FOR EACH STATION
ON FOR SUPPRESSION OF PRINTED OUTPUT

SSW(2) OFF TO PUNCH OUTPUT FOR INPUT TO GRAVITY DESCRIPTIVE, PGM, IGSN
ON TO SUPPRESS PUNCHES OUTPUT

SSW(3) OFF TO OUTPUT GSUM FORMATTED DATA TO IKTAPE
ON TO SUPPRESS GSUM FORMAT OUTPUT
**VARIABLE DEFINITIONS**

- **LDAY, M8, LRY, LTIME = DATE AND TIME OF READING** - IF LOCAL TIME IS USED, KTZ SHOULD ALSO BE ENTERED. IF GMT, KTZ ALWAYS = 0.
- **KGD, KGB, KGY, KGMM = DATE AND TIME OF READING** - IN GMT (OR LOCAL IF KTZ = 99).
- **LOCAL (IF KTZ = 99)**
- **IDAI, IMD1, IDRI, IDTM = DATE AND TIME OF REFERENCE STATION READING (IN GMT)**
- **CRN = GRAVITY METER COUNTER READING (IN UNITS)**
- **ELEV = ELEVATION OF GRAVITY METER (IN METERS)**
- **KTZ = THE TIME ZONE CORRECTION** - IF KTZ = 99, IT INDICATES THAT KTZ WAS NOT AVAILABLE, THUS CLS AND HONK = 0.
- **DATE/TIME MAY BE ENTERED AS GMT, IN WHICH CASE KTZ WILL ALWAYS BE EQUAL TO ZERO**.
- **IF TIME OF READING IS IN LOCAL TIME BUT TIME ZONE IS NOT KNOWN, KTZ MAY BE ENTERED AS 99, IN WHICH CASE A TIME CORRECTION WILL NOT BE MADE AND HTN AND KTZ WILL NOT BE CALCULATED**.

**NOTE:** LOCAL + KTZ ≠ GMT. I.E., VALUES WEST OF GREENWICH ARE PLUS. DESC = DESCRIPTION OF GRAVITY STATION SITE IN ALPHA NUM FORMAT.

- **DRFCTR = THE CORRECTION FACTOR FOR DRIFT OF THE GRAVITY METER**.
- **IF DRFCTR = 0,00, WE_ASSUME A DRIFT_RATE OF 0.003 MGALs/DAY**
- **IF ISK(5) EQUALS 1 PROGRAM DOES NOT MAKE DRIFT CORRECTION**
- **ASDFT = ASSUMED DRIFT FOR GRAVITY METER**

**NOTE:** IF THE GRAVITY METER DRIFT IS NEGATIVE, THE CORRECTION FOR DRIFT IS A POSITIVE NUMBER.

- **BASEG = TOTAL FIELD GRAVITY VALUE AT STATION OF REFERENCE**
- **THE VALUE IS READ IN WITH A FORMAT OF F3.0, F6.2 FROM WHICH INBASE FOR OUTPUT AND BSG7 FOR COMPUTATION IS FORMED**.
- **BGS87 HAS 977000 SUBTRACTED FROM IT FOR OUTPUT, CONVERT TO F3.62 AFTER ADDING 977000**.

- **LSRCD = DEFAULT SOURCE CODE FOR GSUM OUTPUT DATA**
- **DLAT, DLOB = LATITUDE AND LONGITUDE IN DECIMAL DEGREES**

**NOTE:** LOCATIONS NORTH AND EAST ARE CONSIDERED AS POSITIVE; SOUTH AND WEST ARE CONSIDERED NEGATIVE. (THIS IS THE EXACT OPPOSITE OF THE TIME ZONE CONVENTION).

- **JTACE = UNIT REF. NO. FOR PUNCHED CARD OUTPUT (SSW(2) OPTION)**
- **KTAPE = UNIT REF. NO. FOR GSUM FORMAT OUTPUT (SSW(4) OPTION)**
- **IPCS = PUNCH CARD SKIP THIS IS DONE FOR THE GRAVITY DESCRIPTION**

**NOTE:** LOCAL + KTZ ≠ GMT. I.E., VALUES WEST OF GREENWICH ARE PLUS. DESC = DESCRIPTION OF GRAVITY STATION SITE IN ALPHA NUM FORMAT.

- **VALM = GRAVITY METER CALIBRATION TABLES ARRAY**
- **THIS IS THE TABLE USED TO [LOCAL UP] OR CONVERT A GRAVITY METER COUNTER READING TO AN EQUIVALENT RELATIVE MILLIGAL VALUE**.
- **THE COUNTER READING IS READ WITH A FORMAT OF F8.3, FOR INSTANCE 3572.256**.
- **THE HIGH-ORDER TWO DIGITS = IN THIS CASE '35' = ARE USED AS THE ARRAY INDEX**. THE VALUE STORED IN VALM(35) IS THE EQUIVALENT MILLIGAL VALUE FOR A COUNTER READING OF 3500.000, SO WE INTERPOLATE A VALUE BETWEEN VALM(35) AND VALM(36) AND ARRIVE AT A RELATIVE MILLIGAL VALUE FOR 3572.256.

---

**151**
C * INITIALIZATION

C **********************

DIMENSION KDATE(4)
DIMENSION ValH(70),BASEG(2),DESC(31),IGM(2)
DOUBLE PRECISION DTD
DOUBLE PRECISION DEC,RLAT,RAD,RLNG
DATA WEST,SOUTH/"W",15/!

C * ASSUMED VALUES FOR L&R G=18 METER

DATA IBLNK,ILR,IG18/'L&R ','G=18'/!
OUTPUT 'PROGRAM CR2G67 VERSION 2 OCT 75'
LSRCD = 006
ASDFD = +003

C * GET DATE OF RUN

C CALL TBADY (KDATE)

IN*105
IBUT*108
JTAPE=106
KTAPE = 2
ICBUNT=0
IPAGE = 0

C IREC=1
IDEP = 0
RFA = 0*0
IREGC = 0
TCBRR=99.9
ELC=09
LGC*01
IFFC=3
IFBC=0

C * READ IN GRAVITY METER CALIBRATION TABLE

DB 210 K = 170
READ (IN*5150) J,TABLE
VALM(J)=TABLE
210 CONTINUE

C * INITIALIZE SENSE SWITCHES

INN = (ISW(2))

C * READ GRAVITY METER TYPE, DRIFT VALUE, AND SOURCE CODE VALUE

C IF VALUE READ FROM CARD IS 0, USE ASDFT UNLESS SSW(5) IS ON.

READ (IN*5230) IGM(1),IGM(2),DRFTCO,LSRCD,IELC,IGC
IF(DRFTCO=EQ=0) DRFTCO=ASDFT
IF(ISW(5)=EQ=1) DRFTCO=0
IF (IGM(1)=EQ=ILR,IGM(2)=IG18)
1 IGM(1) = ILR; IGM(2) = IG18
IF (LSRCD=EQ=0) LSRCD = LSRCD
IF (IELC=EQ=0) IELC=ELC
153

IF (IGC EQ 0) IGC = LGC

158

C * WRITE OUT JOB INITIALIZATION VALUES
C

WRITE (180,5240) IPAGE, DATE
WRITE (180,5260) IGM(1), IGM(2), DFTCB, LSRC, IELC, IGC
DO 294 II = 1, 70
WRITE (180,5290) II, VALM(II)
294 CONTINUE

C

C ********************************************
C * COMPUTATION OF OBSERVED GRAVITY AND ANOMALIES
C
C * BEGIN A GROUP OF MEASUREMENTS
C ********************************************

C * READ BASE GRAVITY AND CONVERT INTO UNITS COMPATIBLE WITH BOTH SYSTEMS
C

300 CONTINUE
READ (11N,53200) BASEG(1), BASEG(2)
READ (11N,53300) DENSE
BASE7 = ((BASEG11+9772)*1000)*BASEG(2)
BASE = BASEG(1)

C * READ COUNTER READING CARDS FOR INDIVIDUAL STATIONS
C THE FIRST CARD READ IS THE ONE FOR THE REFERENCE STATION
C DRIFT IS COMPUTED FROM DATE ON THIS FIRST CARD
C ALL OTHER READINGS ARE REFERENCED TO THIS MEASUREMENT
C

DO 890 I = 1, 9900
READ (IIIN,54000) DESC(IK), IK = 1, 31
IPCS = 0
CLS = 99
HBNK = 99
CR = CRN
NCR = CRN

C * CHECK TO SEE IF HAVE NEW REFERENCE STATION OR CALL EXIT
C (STMT #180 IS EXIT) #300 IS START OF NEW GROUP
C

IF (NCR) = 500*500*420
500 IF (LYR) = 15*180*180
415 I = 1
GOTO 300

C * SEARCH TABLES FOR GRAVITY VALUE CORRESPONDING TO COUNTER READING
C

420 TUCR = CR = 0*01
LC = TUCR
CLI = LC
C2 = CR

C * CALCULATE LAT AND LON IN RADIANS AND IN DECIMAL DEGREES
C
C RDEG=LAT
242. DEC=RLAT*1.6666666E2
243. LAT=RDEG+DEC
244. DLAT=RLAT
245. RAD=RLAT*(1.7453293D2)
246. C RDEG=LONG
247. DEC=RLNG*1.66666666D2
248. RLANG=RDEG+DEC
249. DLong=RLNG
250. RLANG=RLNG*1.7453293D2

251. C IF KTZ IS EQUAL TO 99 MEANS HAVE NOT MADE OR LOOKED UP THE
252. TIME ZONE CORRECTION. THERE THE FOLLOWING CALCULATIONS
253. ARE NOT NEEDED BECAUSE WE CANNOT CALCULATE THE TIDAL OR
254. HONKLE CORRECTIONS WITHOUT IT

255. C IF(KTZ=NE=99) GO TO 610
256. KGDAD=LDAY
257. KGMM=MB
258. KGYR=LYR
259. KGHM=LI
260. DLAY=ADAY / 2400
261. ADAY=ADAY+FLOAT(ID)
262. C IF NORTH LAT OR EAST LON IS POSITIVE
263. SOUTH LAT OR WEST LON IS NEGATIVE

264. C IF (RLAT) 620/619/620
265. 619 IF (RLONG) 620/630/620
266. 620 CONTINUE

267. C IF(SearnEQ=SOUTH) RLAT=RLAT // DLAT=DLAT
268. IF(MoreEQ=WEST) RLANG=RLONG // DLONG=DLONG

269. C CALCULATE CLS AND HONK VALUES
270. IMH=KGHM/100
271. IMin=KGHM*100
272. CALL TIDAL(RLAT,RLONG,KGYR,ID,IMH,IMIN,CLS,HONK,DTH)
273. RELM=RELH*CLS+HONK

274. C CONTINUE
275. 630 CONTINUE
276. IF(1=1)1/40/44/45
277. C NEW REFERENCE STATION (* FIRST CARD OF GROUP); PROCESSING
278. 440 REF*RELH
279. IDA=KGDAD
280. IMD=KGMM
IYR1*KGYR
ITM1*KGHM
LSTA=LSTAT
IPCS=1
LCNT = 0
IPAGE * 1
C * CALCULATE DRIFT
C
450 DIFFRELH=REF
CALL CDATE(IDAY,IMOH,YR,ITM,KGDA,KGMB,KGYR,KGMH,TIMD)
DRIFT=(TIMD/24.0)*DRFTC5
GGBS7=BASE7+DIFF*DRIFT
COUNT=COUNT+1
C
C * PUT GGBS7 * 977000 INTO OUTPUT UNITS
C
ICON=GGBS7/1000
TEMP=ICON/1000
GGBS=UGBS7+TEMP
C
C * COMPUTE FREE-AIR AND BOUGUER ANOMALIES
C
x=2*RAD
C2R=CASE(X)
FELEV=((0.30855+0.00022*C2R)*ELEV)+((ELEV=0.001)**2)*0.072
RA1=RAD
GFREE=GGBS7*G167F(RA1)*FELEV
BELEV=0.04185*DENSE*ELEV
GBUG=GFREE-BELEV
C
C * CHECK TO SEE IF LAT AND LON = 0
C
OR IF ELEV # 0
C
IF THEY DO SET GFREE AND GBUG EQUAL TO 999.0
C
C
IF (LAT) = 0
C
IF (LATM) 0, 2051, 2052
C
IF (LONG) = 0
C
IF (RLMT) 2054, 2055, 2056
C
C CONTINUE
C
C
C * COMPUTE GSUM SORT KEY FIELDS
C
PLAT = DLAT + 90
LTKEY = PLAT
PLAN = DLBN + 180
LKEY = PLAN
IAKEY = 0
C
C
C * OUTPUT THE DESIRED INFORMATION
C  ***************
C PUNCH OUTPUT FOR INPUT TO STATION DESCRIPTION PROGRAM
C IPCS PREVENTS US FROM PUNCHING THE REFERENCE STATION WHEN
C IT IS THE FIRST CARD OUT
C
C IF (IPCS.EQ.1) G070 835
C IF (ISW(2)) 835,825,835
C 825 WRITE (JTAPE,5825) LYT,M0,LDAY,LTIME,LSTAT,IGN(1),IGN(2),
C 1 IGNBS,LSTA,BASE,BASEG(2),RLAT,SRNR,RLONG,HERE,ELEV
C CONTINUE
C
C LISTING OF CALCULATED VALUES
C
C IF (ISW(1)) 869,851,869
C 851 IF (LCNT) 852,852,860
C
C PRINT PAGE HEADING
C
C 852 WRITE (IBUT,5001) 854
C IF (IPAGE = 1) 853,860
C WRITE (IBUT,5053)
C CONTINUE
C
C WRITE (IBUT,5020) IPAGE,KDATE
C WRITE (IBUT,5885) LSTA, IDA1,IM01,LYR1,ITM1,IGN(1),IGN(2),LSRC
C WRITE (IBUT,5856) BASE,BASEG(2),REF, DENSE, DRTCB
C WRITE (IBUT,5858)
C IF (IPAGE = 1) 851,857,860
C LCNT = 39
C
C 860 WRITE (IBUT,5860)
C 1 LSTAT,LDAY,M0,LYR,LTIME,KTZ,RLATM,SRNR,CR,
C 2 GFREE,CLS,DIFF,DESC
C WRITE (IBUT,5866)
C 1 IGNBS,KGOA,KGMA,KGMR,KGHM,ELEV,LONG,RLBM,HERE,RELH,
C 2 GBUG,HENK,DRIFT,TIMD,ADAY
C LCNT = LCNT + 3
C CONTINUE
C
C OUTPUT AT GSUM FORMAT TO KTAPE
C FORMAT FORWARD CODE = 3
C
C IF (ISW(4)) 889,871,889
C 871 WRITE (KTAPE,5871) IREC,LSRC,KGOA,KGMB,KGMR,KGHM,DLAT,DLBN,ELEV,
C 1 IGNBS,IDEP,GFREE,GBUG,TCBRR,IELC,IGC,RAF,IREC,
C 2 IFCC,CLS,HBNK,CHNP (DESC(11),11,11,16),KSTATN,IFEC,
C 3 LKEY,LOGKEY,IAKEY
C CONTINUE
C
C STMT. #890 IS THE END OF THE READ DO-LOOP
C
C CONTINUE
C
C END OF JOB
C
C ***************
C
C
180 CONTINUE
181 CONTINUE
910 CONTINUE
WRITE (IBUT,5001)
OUTPUT ICBANT
IF(ISK(4).EQ.0) GO TO 24
ENDFILE KTAPE
5950 WRITE (IBUT,5950)
REWIN KTAPE
24 CONTINUE
1 WRITE (IBUT,5200) IPAGE,KDATE
STOP

C ***************
C * FORMATS *
C *
C ***************
C
5001 FORMAT (1X)
5150 FORMAT (12,F7.2)
5200 FORMAT (T2,PAGE:1,4,T3:DATE OF RUN = 1,4A4)
5230 FORMAT (2A4,2X,F10.15,15,15)
5885 FORMAT (1,G=RETER = 12A'4' DRFTC6 = 'F10.5',5X,SURCE CODE = '14'
1 ELEV CODE = '14',5X,G METER CODE = '14')
5290 FORMAT (1,TABLE:1,15(12,F15.8))
5320 FORMAT (F30,F6.2)
5330 FORMAT (F4.2)
5405 FORMAT (14,3I2,1I4,F8.3,12,F5.2,A13,F5.2,A1,F7.1,13,3I1)
5825 FORMAT(3I2,1I4,1X,14,2A4,13,F6.2,1X,14,13,F6.2,2,(F9.6,A1))F7.1)
5855 FORMAT (/1 REFERENCE STATION #15X READING OF 
1 5X METER = '2A4,5X,SURCE CODE = '14')
5853 FORMAT (T2 === NEW REFERENCE STATION ===) /
1 T2=**************
5856 FORMAT (1,REFERENCE GRAVITY = '13',F6.2,27X,REL MGR = 'F11,3')
1 5X,DENSE = '15,F5.2,9X,DRFTC0 = 'F6.4')
5858 FORMAT (1,REFERENCE STATION = '10X,DAT1 = '14X,LATTITUDE = '2X,CTR RING'
1 4X,GRE,E3X,CLS = '5X,10X,DESCRIPTION'/
1 1,S5,GRAY,5X,GMT DATE = '8X,ELEV = '3X,LONGDOTE = '2X,REL MGR = '13,'
3 4X,GBBG',G2X,8NCK = '2X,ACUM DFT = '7X,10X,DA-BY = 'YR')/
5860 FORMAT (1,REFERENCE STA7ION = '14X,7X,3,121,1'14X,6X,13,2X,13,1X,F5.2,2,A13X,
1 21X,F7.2,2X,F4.2,2X,F8.2,8X-F10.2,2X,F8.4)
1 2) /
1 21X,F7.2,2X,F4.2,2X,F8.2,8X-F10.2,2X,F8.4)
1 2)
1 /
5871 FORMAT (1,REFERENCE STATION = '11',F6.2,14,2F3.4,F7.2)
1 13,F6.2,15,2F7.1,F9.1,212,F6.1,11)
1 12,2F4.2,F7.2,16A1,14,12)
1 21X,12)
1 21X,12)
1 /
1 /
5950 FORMAT (1,WRITE END OF FILE)
1 C
END
LOCAL VARIABLES (221 WORDS):

00000 KDATE 00004 VALH 00044A BASEG
00070 DEC 00072 RLAT 00074 RAD
0007A IBLNK 00078 IHR 0007C IJ18
00080 IBUT 00081 UTAPE 00083 TCBUT
00086 IDEP 00087 RFA 00088 IREG
0008C IFHC 0008D IFBC 0008E J
00092 DRTFCB 00093 LSRC 00094 IELC
00098 BASG7 00099 IBASE 0009A I
0009E Lyr 0009F LTIME 000A0 CRN
000A4 IONG 000A5 IMLH 000A6 WERR
000A8 IPCS 000AB CLS 000AC MBNK
000B0 LC 000B1 CL 000B2 C2
000B6 DLEN 000B7 KGDA 000B8 KGB
000BC KTT 000BD NTZ 000BE ADAY
000C2 IDA1 000C3 IMO1 000C4 IYR1
000C8 DIFFR 000C9 TIMD 000CA DRIFT
000CE GBBS 000CF IG 000D0 X
000DA GFREE 000DF BELEV 000D6 GBUG
000DA PLN 000DB LGKEY 000DC IKEY

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

COS FLOAT

EXTERNAL SUBPROGRAMS REQUIRED:

CDATE CHGTM GI67F IGW MDY TIDAL T0DAY F1101
F1102 F1103 F1104 F1105 F1106 F1108 MIDB M18C
9BCDRE DE 9BCDREAD 9BCDWRIT 9CD8 9D78R 9ENDFILE 9END10L 9INITIAL
916DATA 916USA. 91T8R 9PRINT SREWIND 9RT81 9STOP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC HEX
WORDS WORDS
----- -----

GENERATED CODE: 1014 003F6
CONSTANTS: 30 0001E
LOCAL VARIABLES: 281 0000D
TEMPS: 1 00001
----- -----
TOTAL PROGRAM 1266 004F2
**Compiled 1 Apr 1972**

```
1. C PROGRAM CRN3

2. C VERSION OF 7 MARCH 1972. TO USE PINOT

3. C

4. C DIMENSION TAB(100), IDESC(6), VEL(8), THICK(8), X(8)

5. C DIMENSION NB(4)

6. C

7. C PROGRAM CRN3, CALCULATES PRESSURE AT BASE OF CRUSTAL

8. C COLN (KG/CM²)

9. C

10. C SS(0) UP TO LIST INTERMEDIATE VALUES FOR TESTING

11. C SS(26) UP TO SET JTAPE = 108 AND IREC = 0

12. C SS(32) UP TO READ SPFMT DATA ON TAB CARDS

13. C SS(33) UP TO WRITE SPFMT DATA ON TAB CARDS

14. C

15. C USES SUBROUTINES EVIL, ISW, STAT

16. C

17. C

18. C

19. C ITAPE = URN FOR SEISMIC DATA INPUT

20. C JTAPE = URN FOR DATA OUTPUT

21. C ITAPE = 1

22. C JTAPE = 2

23. C

24. C

25. C IIN = 105

26. C IIB = 108

27. C NOUT = 0

28. C

29. C PRINT DATE AND TIME OF JOB ON HEADING

30. C CALL TODAY(NW)

31. C WRITE(IIB, 13) NW

32. C 13 FORMAT(1X, 4A4)

33. C INIT = ISW(-2)

34. C CALL STAT

35. C KG = 1H9

36. C ISTAB = 0

37. C WRITE (IIB, 600)

38. C 600 FORMAT (/ 'PROGRAM CRN3, VERSION OF 7 MARCH, 1972' //)

39. C

40. C KK = 0

41. C CALL PINOT(ITAPE, JTAPE, KK, ISTA, KEY, LAT, LATM)

42. C 1 LON, LON, KEV, VEL, THICK, IHANT, AELEV, N2, N3, N4, M, YR, IDESC

43. C 2 CRV, STIK, CRV, GTN, AVTN, CNV, GTN, AVTN

44. C

45. C DCOMP = DEPTH OF COMPENSATION IN KM

46. C ICTAB = 0 FOR NAPE DRAKE = 1 FOR WOOLARD DENSITY TABLE

47. C READ (IIN, 2) ICTAB, DCOMP

48. C 2 FORMAT (I5, F10.0)

49. C OUTPUT ICTAB, DCOMP

50. C READ IN 1C VALUES PER CARD

51. C READ (IIN, 3) TAB

52. C
```
3 FORMAT (1CF8.3)

C 100 VALUES ENTERED

C READING OF SURFACE WORLD SEISMIC REFRACTION COMPILATION

10 CONTINUE

CALL FIND(ITAPE,TAPE,KK,ISTA,KEY,LT,LM,LAT,LMAT,KNS,

1 LON,LON,KEY,VEL,THICK,IMANT,NELEV,N1,N2,N3,N4,MT,IXK,DESC,

2 DINE,THIK,CRVN,NGT,AVTN,CRV,N,NIGHT)

IF (KK=1) 120,150,120

120 CONTINUE

IF (IST(0)) 161,16,16

16 WRITE(11,OUT,17) ISTA,KEY,VMANT,ELEV,N1,N2,N3,N4

17 FORMAT(1READ,3X,15,13,F5.1,F7.1,3X,41)

C CHECKING IF KEY = 9

13 IF (KEY<9) 120,10,20

20 CONTINUE

ELEV=NELEV

CLE+ELEV*0.01

VMANT=(FLOAT(IMANT))*0.1

NCT=8

43 IF (IMANT<50) 60,50

50 IF (N1-2) 70,60,70

C SEA SEISMIC PROFILE

60 DINE = ELEV

70 NGT = 1.03*ELEV*100.0

GO TO 20

C LAND SEISMIC PROFILE

80 DINE = NELEV

90 NGT = 0.0

100 WRITE(11,OUT,11) DINE,NGT

11 FORMAT(1DINE=",F4.2,=",NGT=",F10.2)

3 X = C*0

28 SX = 0.0

30 STHIK = 0.0

39 DJ 86 J=1,NCT

40 KK = VEL(J)*10.0

91 DENS = TAB(KK)

92 NGT = NGT*(DENS*THICK(J)*100.0)

93 DINE = DINE + THICK(J)

94 X(J) = VEL(J)*THICK(J)

95 SX = SX + X(J)

96 STHIK = STHIK + THICK(J)

100 IF (IST(0)) 90,90,90

101 AI WRITE(11,OUT,11) KK,THICK(J),DENS,NGT,DINE,

102 X(J),SX,STHIK

103 IF (J=5,1) 105,105

105 FORMAT(10D0,12.14,2X,F4.1,2X,F4.2)

106 CONTINUE

107 CVEL = SX/STHIK

108 KK = CVEL*10.0*B

109 CROEN = TAB(KK)
106. \( KK = VMANT \times 10 \times 10^5 \)
107. \( DENS = TAB(KK) \)
108. \( A = CRDEN \times THIK \times 100 \times 0 \)
109. \( R = DENS \times (DCOMP \times DINE) \times 100 \times 0 \)
110. \( IF(15K(0) = 0,95,90) \)
111. \( WRITE(110UT,92) ; \) CRDEN, DENS, A, B
112. \( 92 \text{ FORMAT}(C25, 10, 2, 3, 10, 1, F5, 2, 4) \)
113. \( 93 \text{ KGT = KGT } + B \)
114. \( AVG K = KATN + A + B \)
115. \( AVG T = KATN + A + B \)

C SETTING UP FOR PROPER OUTPUT
117. \( IF(ICTAB) 810, 820, 810 \)
118. \( 810 \text{ CRVL } = CRVEL \)
119. \( WGTN = WGT \)
120. \( AVATN = AVWGT \)
121. \( GO TO 850 \)
122. \( 820 \text{ CRVN } = CRVEL \)
123. \( WGTN = WGT \)
124. \( AVATN = AVWGT \)
125. \( CONTINUE \)

C BUTFLT RESULTS
126. \( 850 \text{ CONTINUE } \)
127. \( KK = 2 \)
128. \( CALL FINAT((TAPE, K, KTAPE, KY, LAT, LATM, KNS, \)
129. \( 1 \text{ LON, LAT, KEV, THICK, IMANT, NELEV, N2, N3, N4, MET, YR, DESC, } \)
130. \( 2 \text{ DINEN, THIK, CRVN, WGTN, AVATN, CRVN, WGTN, AVATN } \)
131. \( \text{ SORT } + 1 \)
132. \( \text{ GO TO } 10 \)
133. \( 540 \text{ WRITE } (110UT, 545) ; \text{ NUT } + 1 \)
134. \( 545 \text{ FORMAT } (1, E4F ) \text{ FIND ON INPUT TAPE } 1, 110 \)
135. \( 546 \text{ NUT } = 0 \)
136. \( \text{ END FILE } \text{ TAPE } \)
137. \( \text{ END } \)
138. \( 999 \text{ CALL EXIT } \)
139. \( \text{ END } \)
INTRINSIC SUBPROGRAMS USED:

FL9AT

EXTERNAL SUBPROGRAMS REQUIRED:

<table>
<thead>
<tr>
<th>EXIT</th>
<th>ISX</th>
<th>PINDI</th>
<th>STAT</th>
<th>TODAY</th>
<th>F:101</th>
<th>F:102</th>
<th>F:103</th>
</tr>
</thead>
<tbody>
<tr>
<td>9ENDFILE</td>
<td>9ENDFIL</td>
<td>9INITIAL</td>
<td>910DATA</td>
<td>918LUSA</td>
<td>911BR</td>
<td>9PRINT</td>
<td>9RT2</td>
</tr>
</tbody>
</table>

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>GENERATED CODE: 461</td>
<td>001C3</td>
</tr>
<tr>
<td>(NO PRIMARY PROTECTION)</td>
<td></td>
</tr>
<tr>
<td>CONSTANTS: 9</td>
<td>00009</td>
</tr>
<tr>
<td>LOCAL VARIABLES: 182</td>
<td>00066</td>
</tr>
<tr>
<td>TEMPS: 0</td>
<td>00000</td>
</tr>
<tr>
<td>TOTAL PROGRAM: 642</td>
<td>00282</td>
</tr>
</tbody>
</table>
PROGRAM DMA

PROGRAM TO READ BLOCKED BY 50 DMA SOURCE TAPES

AND CONVERT THEM TO BLOCKED BY 50 GSUM TAPES (IFFC=9)

HYBRID OF PROGRAM DMA AND CONV67

VERSION 20 MAY 75 TO REMOVE DOUBLE BUFFER

VERSION 12 MAY 75 TO ZERO VARIABLES FOR CP-V

VERSION 10 DEC 74 TO CORRECT TEST BUGS AND ZEROING BUT

VERSION 30 SEPT 74 TO CORRECT SPELLING OF ISLAT IN READ

VERSION 25 MAY 74 BY Gove

DIMENSION IA(35)

DIMENSION I5(UF,1,32,50)

DIMENSION 12(9), 12(35)

DOUBLE PRECISION GBORS

INTEGER A, B, C

INTEGER THREE, FOUR, FIVE

INTEGER SIX, SEVEN, EIGHT, NINE, DEE

DATA GBA, TBA, IA, IB / 1, 2 /

DATA THREE, FOUR, FIVE / 1, 4, 5 /

DATA SIX, SEVEN, EIGHT / 6, 7, 8 /

DATA NINE, DEE / 9, 0 /

EXTERNAL PRECISION GBS

INTEGER ANE, TWO

INTEGER TRREE, FOURS, FIVE

INTEGER SIX, SEVEN, EIGHT, NINE, DEE

DATA BNE, TBW / 4, 19, 2 /

DATA TRREE, FOURS, FIVE / 3, 21, 4 /

DATA SIX, SEVEN, EIGHT / 6, 7, 8 /

DATA NINE, DEE / 9, 0 /

OUTPUT 'DMA-GLK VERSION 20 MAY 75'

OUTPUT 'AN DEC 12 74 IT WAS DISCOVERED THAT'

OUTPUT 'THE LOGIC IN PROGRAM DMA DID NOT HANDLE'

OUTPUT 'ELEVATION CODES OTHER THAN 1 AND 3 CORRECTLY'

OUTPUT 'IT WAS ALSO DISCOVERED THAT DMA-GLK DID NOT ZERO'

OUTPUT 'ELEV AND IDEP. TO FIX THIS A TEMPORARY'

OUTPUT 'VERSION OF DMA-GLK WAS MADE'

OUTPUT 'THIA VERSIO WRITE 1 AN 3 IN GSUM '

OUTPUT 'BUT JUST PASSES ALL OTHER ELEV CODES TO TAPE'

OUTPUT 'IN, DMA FORMAT ALL GSUM B/P IS GARD'

OUTPUT 'LEE GAVE DEC 13 74'

ITAPE = 1

JTAPE = 2

LTAPE = 3

NRECD = 0

IODHA = 0

IN = 105

IOUT = 103

IREC2 = 2

ITAPE = 1

JTAPE = 2

KTAPE = 108

ICNT = 0

NIN = 50

NRUT = 0

INPUT = 0

DEG = 0

K = 1

K = 2

NRECD = 0

A = 3

CLAT = 1

OLANG = 1

HEIGHT = 0

SAKE Y = 0

IDEIF = 0
INPUT STARTING SEQUENCE NUMBER FOR ID

READ(IN,INSEQ)
8 FORMAT(11C)
OUTPUT NSEQ
READ (IN,SEQ) ISORC
5CH FORMAT (15)
OUTPUT ISORC
KGDA*0
KGMB*0
KGYA*0
KGMM*0
IDIF=0
KGDAH=0
KGB=0
KGYRA=0
IELC*0
IGC=0
IREGIC=0
IFFC=9
IFBC=0
IPFS=0
BUF LRG IC FOR I/P
C
10 CONTINUE
IF(NIN.LT.50) GO TO 90
NIN=0
CALL BUFF IN(ITAPE,IBUFIN(1,1),1050)
15 CONTINUE
CALL ICHECK(ITAPE,IKEY,NI)
GO TO (20*50,30*40) IKEY
20 OUTPUT 'WAITING FOR I/P'; IEAD=0
GO TO 15
30 OUTPUT 'END OF FILE ON ITAPE'; IEAD=1
GO TO 50
40 OUTPUT 'BUFFER IN ERRORIZED'; IEAD=1
GO TO 999
50 CONTINUE
C
INPUT LOGIC
C
9G CONTINUE
NIN=NIN+1
IF(NIN.EQ.1050) GO TO 95
C
GOING TO FBF PROCESSING
C
NINCHK=NIN+21
IF(NINCHK.GT.NI) GO TO 999
95 CONTINUE
ELEV=0*0
IDEF=0
FA=999*0
BG=999*0
TC=999.9
NRECRD=NRECRD+1
DECIDE (84*500,IBUFIN(1,NIN),ND)
A IGEBC,ISLAT,LAT,LSLAT,LSLG,LONG,ALONG,IELEV,IELU,
A ELEV,DEPIN,GOBS,
A FA,PRG,SRCE,IBASE,IBR,ISEQ
500 FORMAT(2X,11X,A1,12,F4.2,1X,A1,13,F4.2,1X,A1,14,F7.1,1X,F5.1,1X)
A F5*1,1X, F5*1,1X, A4*1,1X, A4*1,1X, A4*1,1X)

EDIT LOGIC

TB OUTPUT RECORD SEQUENCE NUMBER IN STATION NUMBER FIELD

A=NSEG
KGyr=A*0.0001
E=KGyr*10000
KGHM=A-B

GROUPING VARIABLES FOR OUTPUT UNDER ARRAY IA
ENCODE(35,410,12) SOURCE,IBASE,IRE,ISEG,I4LEV

CALL UNPKBY(12,IM,35)
DB 420 J=1,35
IA(J)=ISL(IN(J),24)

CONTINUE

CALCULATE LAT, LONG, AND KEYS

PLAT=FLOAT(LAT)+(ALAT/60)*0
IF(ISLAT*EQ-NEG)DLAT=-DLAT
CLONG=FLOAT(LONG)+(ALONG/60)*0
IF(ISLNG*EQ-NEG)DLONG=-DLONG

PLAT=DLAT+90*0,ALTKEY=PLAT
PLONG=DLONG+180*0,ALTKEY=PLONG

CHECKING GEOGRAPHIC COORDINATE CODE
IF(IGAN*EQ-2)WRITE(10,510)SOURCE,ISEG,G6 TB 99

FORMAT(' GEO CODE = 1, STOPPED PROCESSING AT ',A4,2X,A4)

CONVERT ELEVATION TO METERS DEPENDING ON CODE
IF(ISLEV*EQ-2)ELEV=ELEV/3*281
IF(ISLEV*EQ-3)ELEV=ELEV/19*686
IF(ISLEV*EQ-6)G6 TB 5215
IF(ISLEV*EQ-THREE)IDEP=ELEV,ELEV=0,G6 TB 5215
IMBAL=IMBAL+1
WRITE(LTAPe5555) (IBUFIN(KK*NN),KK*1,21)

FORMAT(21A4)
G6 TB 10

5215 CONTINUE
IF(GABS - 0*05) 521,521,524

521 K977 = 0
99SG = 0*0
G6 TP 528

524 GABS=GBS*97600*0,0
CALL 88ALL(K577,99SG,GBS,K9)

CHECKING FOR VALID HEIGHT
528 CALL ALTO(ELEV,IDEp,HEIGT,KK)

IF(KK=9)550,530,550

530 G6=999*0

OUTPUT RSM RECORD
550 CONTINUE

OUTPUT LOGIC
300 CONTINUE
NOUT=NBUT+1
ENCOD(12A,1001,IBUF0T(1,NOUT),NC) IREC2, ISRC, K3DA, KGMM,
1 KGHA, KGMM, K1AT, CLNG, ELEV, K977, BSQ, IDEP, FA, RG, TC, IEIC,
2 IGRFA, IREC3, IFFC, IA, IFBC, LTKEY, LGKEY, IAKF
305 CONTINUE
IF(NBUT*LT*50) GO TO 10

C BUFFER LOGIC FAR O/P

C
310 CONTINUE
JKEY=ICHECK(JTAPE)
130 GO TO (320,350,330,340) JKEY
320 OUTPUT 'WAITING FOR O/P' ; IEBD=0
131 GO TO 310
330 OUTPUT 'END OF FILE JTAPE' ; IEBD=1
200 GO TO 999
340 OUTPUT 'BUFF OUT ERROR' ; IEBD=1
201 GO TO 999
350 CONTINUE
203 CONTINUE
204 NOUT=0
205 CALL BLFF OUTF(JTAPE,0,IBUF0T(1,1),1600)
206 GO TO 10
207 CONTINUE

C END OF JOB.C

999 CONTINUE
910 CONTINUE
JKEY=ICHECK(JTAPE)
212 GO TO (920,950,930,940) JKEY
214 OUTPUT 'WAITING FOR O/P' ; IEBD=0
215 GO TO 910
216 OUTPUT 'BAD JKEY' ; IEBD=1
217 GO TO 950
218 OUTPUT 'BUFF OUT ERROR' ; IEBD=1
219 GO TO 950
950 CONTINUE
220 JADS=NOUT*32
222 CALL BLFF OUTF(JTAPE,0,IBUF0T(1,1),JADS)
223 CONTINUE
960 CONTINUE
98 CONTINUE
225 END FILE JTAPE
226 END FILE JTAPE
227 NSEGS*NSEG=1
228 WRITE(IOUT,1090) NREC, NSEG
229 1090 FORMAT(END DMA RUN, DATA POINTS WRITTEN = '18,
230 LAST SEQUENCE NO. = ' ,I10)
231 WRITE(IOUT, 1092) IREC0
232 WRITE(IOUT, 1091) IODBAL
233 OUTPUT 'ALL DONE'
234 CALL EXIT
235 CONTINUE
236 CONTINUE
238 98 FORMAT(1X,32A4)
239 1001 FORMAT(14143212142F944F72213F622152F621F414)
240.      212, F6,1,1,12,35A1,1X,1,213,12)
241.      1091 FORMAT(3X,15,1X, 'ODDBALL RECORDS WRITTEN!')
242.      1092 FORMAT(3X,15,1X, 'RECORDS READ!')
243.      END
1V907

«666

Ol¢

€
Ory
4gtoo
qgte00
0@s

g
3g100
Bs

=
89
137

33ans I

I

9NO1g 4

XIS
I

1

S378vixva

yacs0

vaelo
96200

yIvDS

3vyv00
4£9v00
vav00
48700

Egvoo
6qv00
agvoo
3av00

egy00

Zavoo
64v00

wIVOS
aav90
a
90agS
NuNSix3
NIMLNI9eu%qs
y¥Iv7S
33v00
yiv9s 4v00
*
AVayY
g&*00
wivgs
33v00
yiv3s
vaqvoo
wis
43¥00
y1v9s
€avoo

03v60

y1¥35

sivas

ytvos

ytv3s

yIv9s5

ytv9s
g3v00

wivos

638700
wivos
T3yv00
#gv00
wivjs
y1¥os vav00
00800

33700
YIy3s5

yqv9s
Tgv0d

yIV9S

13090

a

IT
gyd3yn
I

3dvi4 1

3nIN

wIVDS
seo
¥

IN

:

8yADx 1
wiwv9s
Iom
I
iv4

wtv9s
WO»
|

evaox
wivos
I

f

wtv5s
I3avit

3dyil I

iv4sI I
I

39341 I

Eavoe
LHoTa

wiv9s
IA3xI
anel

gaveu
6avor

6182)

Jvvo0 253y1
3nd
agy90

NOT 4
41x3
Lvels 8
4913 ¥
3svel |
I
tegnel
41301 I
3131 I
ges! I
I
291

Bavin
Savi

NS3ix3

#3700
A

Nugix
Tt

vI
1H913
v4
aos

q
17844N8
N1d3q

,vudv
00000

G4v00

ytvos
=

dvvo0
S3v00

ytvos
ytvoS

avs

«ytvos

a

9Zv00

*5v00

YeoVOO

a
69v06

LndI
Q YOSaN
3NIN

t(Scgm

COT
Geg
Teg
ySz0O

3
5084S
9084S
yWw3S

T3v09
agvoo
Sgv00

O60T

6re00
Z3¢d0C

O24

OTe

€0T00

0¢
Ot

462e0C
ggtuc
gttOo

y1V5S

NIN
aél
34g

4te0o

3vvo0
aayHL
der00
Q4nal

T,E0t

O0€6
42g
00g

1

On€
gds
€4200
49200
Zgt0O

of

c¢

739v1

48

va

savv00

age00

SOs

42t00

z29t00

ZI azyoo

Ned

3400
99200
O¢e
g¢

gvv00
gmL

K3H - >

387

yl
erees

eay00
00000
egvd0
faee

3WYN

cere

altv
43%g@
NI
vuo3d 48

24v00
EavoO

Og,

243000
O02

ygl

93s!

mt

Geno
nHo»

t»

A397

4N@N

13av7

Ogee
3y200

gacOO
gnf09

X34

Omi

NIN
AaWL4

93SN

vat
3n@

33uNes

987

zetoo

7398V1

wyqwos
atvos

X3H

YTyIs

367

avv0O0
E4v00

ayvos

29v00
ytvos

EPL eS

Sdmr

Aare
51s!

“SiNGI

NI
36351
Jay!
AJ131
d3q!
wI3H31

N3A3S
3ygVvir

SSV49 3gqi

v

O0¢€s

2601

32200

yOEOO

T8TOO
40S

849139v7

One
66
O¢

Sauem

3390

g9€00

DJy¥eo
i] @9y00
ifun

62200
ete0O
jetoo
X3H

989

A9v00

«ytwos

«yIVoS

OgvoO

ytvoS

3V¥V¥OO
T3v00)

ytvos
ytvos
wtvoS

Avety
98v00

Jagat

99

s1V9S

@4v00
agsv00
d8V00
93v00

alos

atvoS
ytvos

8aqvoo

Sqavoo
20,00

98adS
ytvoS
yAVvoS
ssv1p

4WYN

SNOT

cece

O96

gtes

Oss
OTS
Océ
o0e

Deuds
aivos
aivos
wiVOS

Sauds

Seco

ga¢0U

tqzoo
¢0z00
£200
€be0u

34100
3807
X3H

€2200

ywIVoS
g3v00
a1v3s
y1V9S

yI1V3S

a3v00 =ytvIs

ytvoIS

yqvos
wtvos

TOgdO)

94¥00
EJv00

atvos
wVIS
3ytVos

eYtvoS

yavos

Javoo
63v00
Zavoo

#3v00

98udS

O4v00
NYu3ix3

x34

NYIix3
A £3v00)
A 33¥00)
397

sgcs
€s290

INC

Aveuy

Ogvoo
IgvOG
9ey¥CO

3WYN

NI4ngl

339
Saxe

89v00
23v00

84v00

y¥1V9S

33v00
44V00

wivas

yIVIS
99udS
ytVoS
yqv3s
w1vos

w1Vos
w1v9s
BTVoS

a1v3s
AVUSY
ytvIs

4a1v5s

Ttavoo

NIQLNI

cav0o0

ssv43

ywvos
yqvos
yqvos

~
3aal4

3gat

>

>>>

r>rr>r>

>>

>>

>>>

yIVOS
s3v00
w1vos

34v00

gav00

T4v00
gqv00
03v00
€2000
qavo0
yvv0o
ogvoo

€3¥00°
savoo

a1¥9S
wivos
y1vos

4 I a .-) I Q I I I I I I 1 I I I I 1 I I I I I I I I I I

€2000
OOP

PPP
SO PF

PRP FPF
8 et ot ot ett

8 et et et iet tet ot etet et ot
08 8

PO et th ht
o
oO
oo

#Qqvoo

yav00

x3H

iyayv

138v1

BqVOS

93v00
3300
3807

cease
oose

8
tt

4° ae!
eee

ee

a) a: 4

>
moe
rrr
op rr
>

>>>
>>>
>>
>>

>>

>>

mt et ot ot ot ot ot tet et et
8 et AD) et et tt
8
et het
8
wt wet wt et ID wt

t t

3avoo

>>>>7>>7>7>>7>7>>7>>

ml
Wedel

Nu3ix3
2qvoo

Nyalx3

>>>>>>>
>>>

9av00
azv0o

>>

>>>

HONIN

N3A3S
24eSI
inet!233u1
aoge
4v4d
93Nfr
sge9 4nD1 n31
vax 4Ady 9N@7 334YN
A313
dvd
3al4
4sI
QN
ppt
»»
2!
AayV1 4tar 3441
AGAGNN
OWHO» 446>
334
NIJSNGI1

A A A
2avoo
c3ayvoc
W3v00

Squdm

ot
wtet

TAD) ot ett ot et etetmtetet
O etet ot tet et ot
ot
wt tet ot ot et

330

>>>

OLT

xa™
387

e orc eeatePa ae miter


<table>
<thead>
<tr>
<th>DEC</th>
<th>00AC4</th>
<th>DEGRA</th>
<th>00AC5</th>
<th>KK</th>
<th>00AC6</th>
<th>KK</th>
<th>00AC7</th>
<th>KB</th>
<th>00AC8</th>
<th>NREC</th>
<th>00AC9</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX</td>
<td>00AC4</td>
<td>DEGRA</td>
<td>00AC5</td>
<td>KK</td>
<td>00AC6</td>
<td>KK</td>
<td>00AC7</td>
<td>KB</td>
<td>00AC8</td>
<td>NREC</td>
<td>00AC9</td>
<td>A</td>
</tr>
<tr>
<td>WORDS</td>
<td>00AC4</td>
<td>DEGRA</td>
<td>00AC5</td>
<td>KK</td>
<td>00AC6</td>
<td>KK</td>
<td>00AC7</td>
<td>KB</td>
<td>00AC8</td>
<td>NREC</td>
<td>00AC9</td>
<td>A</td>
</tr>
</tbody>
</table>

Blank Common (0 Words)

Intrinsic Subprograms Used:

- FLOAT 1SL

External Subprograms Required:

- ALTD 1FLD 1BUFF 1BUFFOUT 1EXIT 1ICW 1CHECK 1GB 16 1UNP 1KBY 1F1101
- 1F1102 1F1103 1F1104 1F1105 1F1106 1F1107 1MIDB 1MIC
- 1GBREAD 1GBWRITE 1GDB 1GDB 1GENDFILE 1GEND 1BL 9INITIAL 9IBDATA
- 1GETDATE 1GITER 1GPRINT 1GRT 1GSTOP

Highest Error Severity: 0 (No Errors)

<table>
<thead>
<tr>
<th>DEC</th>
<th>00AC4</th>
<th>DEGRA</th>
<th>00AC5</th>
<th>KK</th>
<th>00AC6</th>
<th>KK</th>
<th>00AC7</th>
<th>KB</th>
<th>00AC8</th>
<th>NREC</th>
<th>00AC9</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX</td>
<td>00AC4</td>
<td>DEGRA</td>
<td>00AC5</td>
<td>KK</td>
<td>00AC6</td>
<td>KK</td>
<td>00AC7</td>
<td>KB</td>
<td>00AC8</td>
<td>NREC</td>
<td>00AC9</td>
<td>A</td>
</tr>
<tr>
<td>WORDS</td>
<td>00AC4</td>
<td>DEGRA</td>
<td>00AC5</td>
<td>KK</td>
<td>00AC6</td>
<td>KK</td>
<td>00AC7</td>
<td>KB</td>
<td>00AC8</td>
<td>NREC</td>
<td>00AC9</td>
<td>A</td>
</tr>
</tbody>
</table>

Generated Code: 898

Constants: 17

Local Variables: 2819

Temps: 1

Total Program: 3719

03E87
PROGRAM DMAP
VERSION OF 8 DEC 1975 TO CHECK HEMISPHERES
VERSION 27 OCT 1975
VERSION AUGUST 1968
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION XC(4),YC(4)
DIMENSION EDEG(8),BDIF(8)
DIMENSION LAT(4),LANG(4)
OUTPUT (D)MAP - VERSION 8 DEC 1975
CONVERTS DIGITIZED POSITION IN INCHES
TO LATITUDE AND LONGITUDE
NO COMMON REQUIRED
ISH CHOICE DATA CARD GOES BEFORE ITAPE ETC. CARD
USES SUB CALCSCI(ARG), FUNC PARTM(ARG), FUNC ISW(N),
SUB RTODM(ARG), FUNC DMTB6(ARG)
SSW(2) UP TO LIST INTERMEDIATE VALUES
SSW(3) UP TO LIST DATE AND SM IN FOR EACH DATA POINT
SSW(5) UP TO OUTPUT SM IN ONLY IF GREATER THAN EPSIL
NYR=0 WILL TERMINATE PROGRAM
START INITIALIZATION FOR SIGMA 7
II N=105
I INOUT=108
INIT = ISW (=2)
END INITIALIZATION FOR SIGMA 7
DEGRA=1.745329E-2
IHEM=IHEMS=0
ITYPE=0
IGAL=0
N60=0
MAF=0
ITAPE = URN FOR DATA INPUT
JT APE = URN FOR DATA OUTPUT
FFAC = FACTOR (0.1 TO 1.00) USED IN ITERATION FOR
ESTIMATED LATITUDE TO CONVERGE ON TRUE
LATITUDE.
EPSIL = TOLERANCE (IN MERIDIONAL PARTS) BY WHICH
ESTIMATED LATITUDE MUST MATCH MERIDIONAL
PARTS FOR TRUE LATITUDE.
READ(N60) ITAPE, JT APE, FFAC, EPSIL
6 FORMAT(215,F5.2,F5.2)
READ IN DATA
8 READ(ITAPE,150) IC8DE,XP,YP,NDA,NMB,NYR,NHM
15 FORMAT(1,1X,F5.3,1X,F5.3,313,15)
CHANGING SIGN FOR WESTERN OR SOUTHERN HEMISPHERE
IF(IHEMS *GT*0) XP=-1*0*XP
IF(IHEMS *GT*0) YP=-1*0*YP
IF(ICY8DE=9150)20,50
SETTING MAP COORDINATE AND SCALE
20 XC(1)*XP,YC(1)*YP,JLAT(1)=NMB,II HE M90=NYR
CHANGING SIGN FOR WESTERN OR SOUTHERN HEMISPHERE
IF(LONG(1)=LT*0) XC(1)=-1*0*XC(1) IHEM=9
IF(LAT(1)=LT*0) YC(1)=-1*0*YC(1) IHEM=9
DO 25 J=2,4
READ(ITAPE,150) IC8DE,XC(J),YC(J),N1,N2,LAT(J),LONG(J)
CHANGING SIGN FOR WESTERN OR SOUTHERN HEMISPHERE
IF(LONG(J)=LT*0) XC(J)=-1*0*XC(J)
60 CONTINUE
61 MAF=MAF+1
62 C DETERMINING AVERAGE SIN AND COS OF ANGLE OF TILT OF MAP
63 CALL CALSC(XC(1),YC(1),XC(2),YC(2),SC(1),CC(1))
64 CALL CALSC(YC(2),XC(3),YC(3),XC(2),SC(2),CC(2))
65 CALL CALSC(XC(4),YC(4),XC(3),YC(3),SC(3),CC(3))
66 CALL CALSC(YC(1),XC(4),YC(4),XC(1),SC(4),CC(4))
67 AS = (SC(1)+SC(2)+SC(3)+SC(4))/4.0
68 AC = (CC(1)+CC(2)+CC(3)+CC(4))/4.0
69 C LIST CALCULATED SIN AND COS OF ANGLE OF TILT OF MAP
70 WRITE(11,10) MAF,AS,AC
71 10 FORMAT('MAP = ',I4,'  SIN A = ',F8.6,'  COS A = ',F8.6)
72 IF(IS#(1))26 30,26
73 26 WRITE(11,20) SC(1),SC(1),SC(1),SC(1),SC(1),SC(1)
74 20 FORMAT(4F10.6)
75 C ROTATION OF AXES TO CALCULATE MAP SCALE
76 30 X=(XC(2)-XC(1))
77 Y=(YC(4)-YC(1))
78 X=(X+(Y**2))/((X**2)+(Y**2))
79 C DETERMINING LONITUDE OF DATA POINT
80 C EQLG=FLAT*(X/SINH)
81 C IEGL=CEQLG
82 C RLOM(CEGL-A)*60.0
83 C DETERMINING LATITUDE OF DATA POINT
84 C YY=YY
85 C FINISHED CALCULATING SCALE AND PARAMETERS FROM COORDINATE POINTS
86 GO TO 8
87 C MAIN CALCULATION PORTION FOR MAP DATA POINTS
88 IF(NR)50 100,50
89 100 IF(50)50 100,50
90 50 XP=XP-XC(1)
91 YP=YP-YC(1)
92 X=(XP+AC)+(YP+AS)
93 Y=-1.0*(XP+AS)+(YP+AC)
94 C DETERMINING LONGITUDE OF DATA POINT
95 DEGL=FLAT+(X/SINH)
96 ILNG=DEGL
97 A=ILNG
98 RLBM=(DEGL-A)*60.0
99 C DETERMINING LATITUDE OF DATA POINT
100 YY=YY
101 C
51. C MAKING FIRST ESTIMATE OF LATITUDE OF DATA POINT
52. DEGE = DEG + (PH-BTMP)/AVMP)
53. 60 RDEGE = DEG - DEG
54. CALMP = CTRM(RDEGE)
55. KNT = KNT + 1
56. C FINDING DIFFERENCE BETWEEN ACTUAL AND MERIDIONAL PARTS FOR
57. C ESTIMATED LATITUDE
58. DIFMP = PM - CALMP
59. IF(ISW(2) = 61, 65, 61)
60. WRITE(IIBUT, 62) KNT, PM, DIFMP, X3, Y3, SINCH,
61. 1 SMR, AVMP, X2, Y2, BBTMP, T6MP
62. FORMAT(I4, 6F10.2, 6F10.2, 6F10.2)
63. 65 ADIF = DABS(DIFMP)
64. EDEG(KNT) = RDEGE
65. BDIF(KNT) = ADIF
66. C EPSIL IS MERIDIONAL PARTS FOR 0.1 MINUTE OF ARC
67. IF(KNT = 8) 68, 80, 80
68. C MAKING NEW ESTIMATE OF LATITUDE FOR DATA POINT
69. 68 RDEGE = DEG + ((DIFMP/AVMP)*FFAC)
70. GO TO 60
71. C SELECTING MINIMUM ADIF, CALLED SMIN
72. 80 SMIN = BDIF(1)
73. RDEGE = EDEG(1)
74. D9 82 I = 2 + 8
75. IF(BDIF(I) = SMIN) 81, 82, 82
76. 81 SMIN = BDIF(I)
77. RDEGE = EDEG(1)
78. 82 CONTINUE
79. IF(ISW(3) = 83, 84, 83
80. WRITE(IIBUT, 83) NDA, NMB, NMR, NHR, SMIN
81. 83 FORMAT(I12, 15, F7.2)
82. GO TO 84
83. IF(SMIN = EPSIL) 84, 85, 85
84. IF(ISW(5) = 83, 84, 83
85. CALL RTDMM(RDEGE, ILAT, RLATM)
86. C OUTPUT POSITION OF DATA POINT
87. WRITE(JTAPE, 85) NDA, NMB, NMR, NHR, ILAT, RLATM,
88. 85 FORMAT(I12, 14, 9X, I3, F6.2, I4, F6.2, 2X, I2, I1, 15X, I1)
89. GO TO 8
90. 86 WRITE(IIBUT, 101)
91. 101 FORMAT(I12, 15, F7.2)
92. END FILE JTAPE
93. REWIND JTAPE
94. END
...ANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:
DABS

INTERNAL SUBPROGRAMS REQUIRED:
CALSC  DMTR  ISM  PARTM  RT6DM  FI101  FI102  F1103
F1104  F1105  F1106  F1108  M100  M10C  9BCDREAD  9BCDWRIT
9DSORT  9DT61  SENDFILE  SEND16L  9INITIAL  916DATA  91T89  9PR;

GREATEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE</td>
<td>544</td>
<td>00220</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>14</td>
<td>0000E</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>182</td>
<td>000B6</td>
</tr>
<tr>
<td>TEMPS</td>
<td>2</td>
<td>00002</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>742</td>
<td>002E6</td>
</tr>
</tbody>
</table>
PROGRAM DMBD

VERSION OF 6 OCT 1972, TO ALSO HANDLE CASE WHERE SECOND CARD
TO THE LEFT OF THE ORIGIN POINT (FIRST CARD)

VERSION OF 15 OCT 1971, CORRECTS FORMAT ERROR

VERSION OF 6 APRIL 1971, OUTPLTS POLYGON NUMBER IN OUTPUT CARDS

VERSION OF 28 MARCH 1971, ALLOWS FOR MAKING END CARDS

VERSION OF 8 FEB 1971

PROGRAM DMBD, FOR DIGITIZING POLYGONS FOR CRYSTAL MODELS

SSW(1) = 0 FOR SECOND POINT TO RIGHT (+) OF ORIGIN
SSW(1) = 1 FOR SECOND POINT TO LEFT (-) OF ORIGIN

ON DIGITIZING TABLE, ICBDE IS SET IN LEFTMOST THUMBWHEEL SWITCH
POSITION ON MANUAL ENTRY SWITCHES
POLYGON NUMBERS ARE SET IN THE THREE PAIRS OF SWITCHES
TO THE RIGHT OF THE LEFTMOST SWITCH

INPUT DATA CARDS =
1 VALUES FOR XFAC & YFAC IN KM'S / INCH AND
KM VALUES OF ORIGIN OF MODEL (x10^-0)
2 X AND Y VALUES FROM DIGITIZING TABLE FOR ORIGIN
3 X AND Y VALUES FROM DIGITIZING TABLE FOR A POINT AT SAME Y
LEVEL AS ORIGIN
4 X AND Y VALUES FROM DIGITIZING TABLE FOR POLYGON CORNERS

SET ICBDE = 9 FOR INDICATING LAST CARD OF POLYGON FOR TALPLOT PROG
SET ICBDE = 8 FOR X = -2000 KM
SET ICBDE = 7 FOR X = +2000 KM

ICBDE IS RESET BY PROGRAM TO ZERO
LAST DATA CARD SHOULD HAVE ICBDE = 99

OUTPUT : DMBD OF 6 OCT 1972

IN= 105
INOUT= 108
TAPE= 106
INIT = ISW(=2)
READ(IN,12) XFAC,YFAC,XBRG,YBRG
FORMAT(4F10.0)
READ(IN,2C) XA,YA,IA,KP1,KP2,KP3
FORMAT(2F10.3,15B314)
READ(IN,2C) XB,YB,IB,KP1,KP2,KP3
CALL CAL5C(XA,YA,XB,YB,AS,AC)
OUTPUT AS,AC
50 READ(IN,2C) XP,YP,ICBDE,KF1,KF2,KF3
IF(ICBDE=90)60 ; 60 ; 999
60 CONTINUE
XP= XP-XA
YP= YP-YA
X= (XP * AC) + (YP * AS)
Y= (X+0 * (XP=AS)+(YP*AC)
XKM= (X*XFAC) + XBRG
YKM= (Y*YFAC) + YBRG
YKM= -1.0 * YKM
IF(ISW(1))68,6K,64
REVERSE SIGN OF X DISTANCE SINCE SECOND LEVELING POINT WAS TO THE LEFT OF ORIGIN

60 C
61 C
62 C
63 C
64 64 XKM=-XKM
65 65 YKM=-YKM
66 66 CONTINUE
67 67 IF (ICBDE=8) 74, 70, 74
68 70 XKM=-3000*C
69 70 ICBDE=0
70 70 GB TO 100
71 71 IF (ICBDE=7) 90, 76, 90
72 76 XKM=+3000*C
73 73 ICBDE=0
74 74 GB TO 100
75 90 CONTINUE
76 100 IF (KP1) 11C, 11C, 105
77 105 WRITE(*, TAPE, 22) XKM, YKM, ICBDE, KP1
78 110 IF (KP2) 12C, 12C, 115
79 115 WRITE(*, TAPE, 22) XKM, YKM, ICBDE, KP2
80 120 IF (KP3) 13C, 13C, 125
81 125 WRITE(*, TAPE, 22) XKM, YKM, ICBDE, KP3
82 130 CONTINUE
83 22 FORMAT(2F10.2, 1I5, 20X, I5)
84 24 FORMAT(2F10.2, 13, 31, 10)
85 26 WRITE(*, BLK, 24) XKM, YKM, ICBDE, KP1, KP2, KP3
86 50 GB TO 50
87 999 CALL EXIT
88 END
COMPILED 6 DEC 73

1. CS PROGRAM GFLD:
2. C VERSION OF 22 SEPT 1972, TO CORRECT PROGRAM NAME IN OUTPUT
3. C SOURCE OF OCT 19, 1970
4. C
5. C PROGRAM GFLD, SPHERICAL HARMONIC CALCULATIONS IN A
6. C GIVEN REGION
7. C
8. C INCLUDING READING OF C, S
9. C SS(4) UP TO LIST DATA ON TTY DURING RUN
10. C
11. C OUTPUTS REGIONAL FA VALUE IN FREE-AIR POSITION OF
12. C SEA1 FORMAT
13. C
14. C USES SUBROUTINES ISW, FLD2
15. C
16. C
17. C
18. 999 IN = 105
19. IOUT = 108
20. C
21. C OUTPUT : GFLD1 RUN, VERSION OF 22 SEPT 1972:
22. C II = ISW(=2)
23. C READ (II, 8) ITAPE, JTAPE
24. C FORMAT (215)
25. C IOUT, ITAPE, JTAPE
26. C IREC = 1
27. C KG = 22
28. C KK = 0
29. C5600 WRITE (IOUT, 9)
30. C5 FORMAT (6F5.0, I8T, ILEFT, IRIGHT, INC)
31. C5 READ (II, 9) ITAPE, JTAPE, ILEFT, IRIGHT, INC
32. C5600 CALL FLD2(KK, ITAPE, JTAPE, ILEFT, IRIGHT, REG)
33. C5 IAREA = 0
34. C5 ISUV = 0
35. C5 ISTA = 0
36. C5 FLEV = 0.0
37. C5 S77 = 0.0
38. C5 DEPTH = 0.0
39. C5 FA = 999.0
40. C5 BG = 999.0
41. C5 TC = 99.0
42. C5 ACRM = 999.0
43. C5 DEGA = 745329E-2
44. C5 NS = 123B
45. C5 NE = 127B
46. C5 JL = 116B
47. C5 JS = 128B
48. C5 JE = 1058
53. C6  OW=127B
54. C6  CONTINUE
55. 395 DB 402 I=IBOT,ITOP,INC
56. DB 402 J=LEFT,RIGHT,INC
57. DLAT=1
58. DLON=2
59. RLAT=DLAT*DEGRA
60. RLON=DLON*DEGRA
61. KK=1
62. CALL FLD2(KK,ITAPE,RLAT,RLONG,REG)
63. 306 IF(ISW(4))3306,330A,3306
64. 3306 WRITF(IOUT,307)DLAT,DLON,REG
65. 307 FFORMT('DLAT=',F7.2,' DLON=',F7.2,' REG=',F8.3)
66. 3306 FA*REG
67. KFA=FA+10.0
68. C6 445 CALL ENDB
69. C OUTPUT AT SEAG1 FORMAT
70. IF(ISW(26)|EQ.1 OR JTAPE*EQ.108) IREC=6 / JTAPE=108
71. WRITF(JTAPE,12)IREC,KGYR,RLAT,RLONG,KFA
72. 12 FFORMT('11.4X,12.7X,2F9.6,17X,15.35X)
73. CONTINUE
74. WRITF(IOUT,410)
75. 410 FFORMT('THIS RUN COMPLETED//')
76. ENDFILE JTAPE
77. STOP
78. END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG</td>
<td>R</td>
<td>SCALR</td>
<td>00018</td>
<td>V</td>
<td>1</td>
<td>BCGCH</td>
<td>R</td>
<td>SCALR</td>
<td>0001A</td>
<td>V</td>
<td>1</td>
<td>DEGRA</td>
<td>R</td>
<td>SCALR</td>
</tr>
<tr>
<td>DEPTH</td>
<td>R</td>
<td>SCALR</td>
<td>00016</td>
<td>V</td>
<td>1</td>
<td>DLAT</td>
<td>R</td>
<td>SCALR</td>
<td>0001E</td>
<td>V</td>
<td>1</td>
<td>DLB8N</td>
<td>R</td>
<td>SCALR</td>
</tr>
<tr>
<td>ELEV</td>
<td>R</td>
<td>SCALR</td>
<td>00013</td>
<td>V</td>
<td>1</td>
<td>FA</td>
<td>R</td>
<td>SCALR</td>
<td>00017</td>
<td>V</td>
<td>1</td>
<td>FLD2</td>
<td>SPRG</td>
<td>EXTERN</td>
</tr>
<tr>
<td>GBS8</td>
<td>R</td>
<td>SCALR</td>
<td>00015</td>
<td>V</td>
<td>1</td>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>0001C</td>
<td>V</td>
<td>1</td>
<td>IAREA</td>
<td>I</td>
<td>SCALR</td>
</tr>
<tr>
<td>I88T</td>
<td>I</td>
<td>SCALR</td>
<td>00009</td>
<td>V</td>
<td>1</td>
<td>II</td>
<td>I</td>
<td>SCALR</td>
<td>00002</td>
<td>V</td>
<td>1</td>
<td>IIN</td>
<td>I</td>
<td>SCALR</td>
</tr>
<tr>
<td>ILEFT</td>
<td>I</td>
<td>SCALR</td>
<td>00001</td>
<td>V</td>
<td>1</td>
<td>IRIGHT</td>
<td>I</td>
<td>SCALR</td>
<td>0000A</td>
<td>V</td>
<td>1</td>
<td>INC</td>
<td>I</td>
<td>SCALR</td>
</tr>
<tr>
<td>ISURV</td>
<td>I</td>
<td>SCALR</td>
<td>00005</td>
<td>V</td>
<td>1</td>
<td>IS</td>
<td>I</td>
<td>SCALR</td>
<td>00006</td>
<td>V</td>
<td>1</td>
<td>ISTA</td>
<td>I</td>
<td>SCALR</td>
</tr>
<tr>
<td>IT8P</td>
<td>I</td>
<td>SCALR</td>
<td>00008</td>
<td>V</td>
<td>1</td>
<td>J</td>
<td>I</td>
<td>SCALR</td>
<td>00010</td>
<td>V</td>
<td>1</td>
<td>JAVA</td>
<td>I</td>
<td>SCALR</td>
</tr>
<tr>
<td>KFA</td>
<td>I</td>
<td>SCALR</td>
<td>00020</td>
<td>V</td>
<td>1</td>
<td>KG</td>
<td>:</td>
<td>SCALR</td>
<td>00006</td>
<td>V</td>
<td>1</td>
<td>KK</td>
<td>I</td>
<td>SCALR</td>
</tr>
<tr>
<td>K977</td>
<td>I</td>
<td>SCALR</td>
<td>00014</td>
<td>V</td>
<td>1</td>
<td>REG</td>
<td>R</td>
<td>SCALR</td>
<td>0000F</td>
<td>V</td>
<td>1</td>
<td>RLET</td>
<td>R</td>
<td>SCALR</td>
</tr>
<tr>
<td>SLG3</td>
<td>R</td>
<td>SCALR</td>
<td>0000E</td>
<td>V</td>
<td>1</td>
<td>TC</td>
<td>R</td>
<td>SCALR</td>
<td>00019</td>
<td>V</td>
<td>1</td>
<td>RLET</td>
<td>R</td>
<td>SCALR</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (33 WORDS):**

- 00000 IIN
- 00006 KG
- 00007 KK
- 00008 ITB
- 00009 I88T
- 00010 ILEFT
- 00011 IRIGHT
- 00012 I88T
- 00013 ELEV
- 00014 K977
- 00015 GBS8
- 00016 DEGRA
- 00017 FLD2
- 00018 I88T
- 00019 SCALR
- 00020 TCH
- 00021 BCGCH
- 00022 DEGRA
- 00023 FLD2
- 00024 ITB
- 00025 KG
- 00026 SCALR
- 00027 TCH
- 00028 BCGCH
- 00029 DEGRA
- 00030 FLD2
- 00031 ITB
- 00032 KG
- 00033 SCALR
- 00034 TCH
- 00035 BCGCH
- 00036 DEGRA
- 00037 FLD2
- 00038 ITB
- 00039 KG
- 00040 SCALR
- 00041 TCH
- 00042 BCGCH
- 00043 DEGRA
- 00044 FLD2
- 00045 ITB
- 00046 KG
- 00047 SCALR
- 00048 TCH
- 00049 BCGCH
- 00050 DEGRA

**EXTERNAL SUBPROGRAMS RECIPIENTS:**

- FLD2
- IT2
- F101
- F102
- F103
- F104
- F105
- F106
- 9100DATA
- 9100F
- 9100P
- 9100F
- 9100P
- 9100F
- 9100P
- 9100F
- 9100P
- 9100F
- 9100P
- 9100F
- 9100P
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERMS</td>
<td>276</td>
<td>00114</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>33</td>
<td>00021</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>7</td>
<td>00007</td>
</tr>
<tr>
<td>GENERATED CODE</td>
<td>234</td>
<td>000EA</td>
</tr>
</tbody>
</table>

(NO MEMORY PROTECTION)
PROGRAM GFLD2

VERSION OF 4 JANUARY 1972
VERSION OF 13 DEC. 1971, TO READ COEFFICIENTS FROM CARDS
VERSION OF 26 NOV. 1971, FOR READ AND WRITE OF GSUM FILE

PROGRAM GFLD, SPHERICAL HARMONIC CALCULATION IN A
GIVEN REGION

INCLUDING READING OF C,S
SSh(4) UP TO LIST DATA ON TTY DURING RUN

OUTPUTS REGIONAL FREE-AIR VALUE IN REGIONAL POSITION OF GSUM

USES SUBROUTINES ISw, FLD2, GIN8

DIMENSION IA(35)
DIMENSION NOw(4)
CALL STAT
IIw = 105.
IIBUT = 108
ITAPE=1
JTAPE=2.
KTAPE=105
NOUT=0

PRINT DATE AND TIME OF JOB ON HEADING
CALL TODAY(NBH)
WRITE(IIBUT,13) NBH

13 FORMAT(1X,4A4)

IF (NBH<2) 12025120

CALL GIN8IITAPE,JTAPE,KGDA,KGBA,
1 KGYR,KGHM,IDIF,ISORC,RLAT,RLBG,ELEV,K977,8BSG,
2 IDEP,FA,BG,TC,IELC,IGC,REGC,IPFC,IA,IFBC)

12C CONTINUE

CALL FLD2(KK, ITAPE, RLAT, RLONG, REG)

CONTINUE

READ INPUT DATA

CONTINUE

10C CONTINUE

CALL GIN8IITAPE,JTAPE,KGDA,KGBA,
1 KGYR,KGHM,IDIF,ISORC,RLAT,RLONG,ELEV,K977,8BSG,
2 IDEP,FA,BG,TC,IELC,IGC,REGC,IPFC,IA,IFBC)

IF (KK<3) 120120

CONTINUE

CALL FLD2(KK, ITAPE, RLAT, RLONG, REG)

REG
IREGC = 1 FOR SAG MODEL EARTH 1969 COEFS FOR INTL GRAVITY FORMULA

C

IREGC=1

KK=-2

CALL GINAT ITAPE, ITAPE, KK, KGDA, KGMB,

1 KGVR, KGHM, JIDF, ISBRC, RLAT, RLONG, ELEV, K977, OBSG,

2 IDEP, FA, BG, TC, IELC, IGC, RFA, IREGC, IFFC, IA, IFBC)

NOUT=NOUT+1

G8 TO 100

WRITE (INPUT, 545) NOUT

545 FORMAT ('E6F FOUND ON INPLT TAPE ', I10)

NOUT=0

END FILE I TAPE

STOP

END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>LBC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>LBC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>LBC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BG</td>
<td>R</td>
<td>SCALR</td>
<td>0003D V</td>
<td>1</td>
<td></td>
<td>ELEV</td>
<td>R</td>
<td>SCALR</td>
<td>00038 V</td>
<td>1</td>
<td></td>
<td>FA</td>
<td>R</td>
<td>SCALR</td>
<td>0003C V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FL02</td>
<td>SPRB</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td></td>
<td>GIN0T</td>
<td>SPRB</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td></td>
<td>IA</td>
<td>I</td>
<td>ARRAY</td>
<td>00000 V</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>IDEP</td>
<td>I</td>
<td>SCALR</td>
<td>00038 V</td>
<td>1</td>
<td></td>
<td>IDIF</td>
<td>I</td>
<td>SCALR</td>
<td>00036 V</td>
<td>1</td>
<td></td>
<td>IELC</td>
<td>I</td>
<td>SCALR</td>
<td>0003F V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IFBC</td>
<td>I</td>
<td>SCALR</td>
<td>00044 V</td>
<td>1</td>
<td></td>
<td>IFCC</td>
<td>I</td>
<td>SCALR</td>
<td>00043 V</td>
<td>1</td>
<td></td>
<td>IGC</td>
<td>I</td>
<td>SCALR</td>
<td>00040 V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IIN</td>
<td>I</td>
<td>SCALR</td>
<td>00027 V</td>
<td>1</td>
<td></td>
<td>I10T</td>
<td>I</td>
<td>SCALR</td>
<td>00026 V</td>
<td>1</td>
<td></td>
<td>INIT</td>
<td>I</td>
<td>SCALR</td>
<td>0002D V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IREGC</td>
<td>I</td>
<td>SCALR</td>
<td>00042 V</td>
<td>1</td>
<td></td>
<td>ISBRC</td>
<td>I</td>
<td>SCALR</td>
<td>00037 V</td>
<td>1</td>
<td></td>
<td>ISH</td>
<td>I</td>
<td>SPRB</td>
<td>EXTERN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITAPE</td>
<td>I</td>
<td>SCALR</td>
<td>00029 V</td>
<td>1</td>
<td></td>
<td>JTAPE</td>
<td>I</td>
<td>SCALR</td>
<td>0002A V</td>
<td>1</td>
<td></td>
<td>KGDA</td>
<td>I</td>
<td>SCALR</td>
<td>00032 V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>KGHM</td>
<td>I</td>
<td>SCALR</td>
<td>00035 V</td>
<td>1</td>
<td></td>
<td>KTAPE</td>
<td>I</td>
<td>SCALR</td>
<td>00033 V</td>
<td>1</td>
<td></td>
<td>KGYR</td>
<td>I</td>
<td>SCALR</td>
<td>00034 V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>KK</td>
<td>I</td>
<td>SCALR</td>
<td>0002E V</td>
<td>1</td>
<td></td>
<td>N52</td>
<td>I</td>
<td>ARRAY</td>
<td>00023 V</td>
<td>4</td>
<td></td>
<td>K877</td>
<td>I</td>
<td>SCALR</td>
<td>00039 V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>NBUT</td>
<td>I</td>
<td>SCALR</td>
<td>0002C V</td>
<td>1</td>
<td></td>
<td>N872</td>
<td>I</td>
<td>ARRAY</td>
<td>00022 V</td>
<td>4</td>
<td></td>
<td>B98</td>
<td>R</td>
<td>SCALR</td>
<td>0003A V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>REG</td>
<td>R</td>
<td>SCALR</td>
<td>00031 V</td>
<td>1</td>
<td></td>
<td>RFA</td>
<td>R</td>
<td>SCALR</td>
<td>00041 V</td>
<td>1</td>
<td></td>
<td>LAT</td>
<td>R</td>
<td>SCALR</td>
<td>0002F V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>RLONG</td>
<td>R</td>
<td>SCALR</td>
<td>00030 V</td>
<td>1</td>
<td></td>
<td>STAT</td>
<td>SPRB</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td></td>
<td>TC</td>
<td>R</td>
<td>SCALR</td>
<td>0003E V</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**LABEL HEX LBC ** | **LABEL HEX LBC ** | **LABEL HEX LBC ** | **LABEL HEX LBC ** | **LABEL HEX LBC ** | **LABEL HEX LBC ** |
| 13   | 00018 | 100   | 00051 | 120  | 00071 | 155  | 0009A | 540 | 0009B | 545  | 000A0 |

**LOCAL VARIABLES (69 WORDS):**

- 00029 IA 00023 NW
- 00027 INN 00028 II6UT 00029 ITAPE 0002A JTAPE
- 00000 KTAPE 0002C NBUT 0002D INIT 0002E KK 0002F RLAT 00030 RLONG
- 00031 REG 00032 KGDA 00033 KGM 00034 KGYR 00035 KGMH 00036 IDIF
- 00037 ISBRC 00038 ELEV 00039 K977 0003A BB9 0003B I10P 0003C FA
- 0003D BG 0003E TC 0003F IEIC 00040 IGC 00041 RFA 00042 IREGC

**BLANK COMMON (0 WORDS)**

**EXTERNAL SUBPROGRAMS REQUIRED:**

- FLD2 GIN0T ISW
- STAT TODAY FI102 FI104 FI106
- 916 USA SPRINT 9STOP
- 9BCWRIT 9ENDFILE 9END10L 9INITIAL 9I0DATA
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>GENERATED CND:</td>
<td>176</td>
<td>00080</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>1</td>
<td>00001</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>69</td>
<td>00045</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>69</td>
<td>00000</td>
</tr>
<tr>
<td>TOTAL PROGRAM:</td>
<td>246</td>
<td>000F6</td>
</tr>
</tbody>
</table>

(NO MEMORY PROTECTION)
PROGRAM GRAF2

OUTPUT 'GRAF2 RUN, VERSION OF 11 AUG 75'

-----------------------------------------------------------------

VERSION 11 AUG 75 TO ADD OUTPUT OF NUMBER OF RECORDS
VERSION 11 APRIL 75; ADD TEST FOR SPACECRAFT ALTITUDE LUNAR DATA
VERSION 9 APRIL 75; TO ADD SSW(3) FOR INPUT OF LUNAR DATA
VERSION OF 13 JULY 1973, TO MOVE LABEL AWAY FROM JOB NO.
VERSION OF 27 APRIL 1973; ADDING LABEL AND NOW DATE
PROGRAM GRAF2, PLOTS ONE VARIABLE VS ANOTHER
ALSO HAS BEGINNING AND END DATE CHECK LOGIC

SSW(0) = 1 TO OUTPUT VALUES FOR TESTING
SSW(3) = 1 TO USE GETL FOR INPUT OF LUNAR DATA
SSW(4) = 1 TO READ SPACECRAFT ALTITUDE BOUNDS FOR LUNAR DATA,
(AFTER READING AREA BOUNDS), AND TO PROCESS ONLY
DATA WITHIN THESE BOUNDS
SSW(7) = 1 TO INPUT NEW AREA BOUNDS FOR NEXT PLOT
SSW(8) = 1 TO SUPPRESS PLOTTING GRID
SSW(10) = 1 TO START A NEW GRAPH
SSW(12) = 1 TO LIST DATA IDENTIFICATION
SSW(13) = 1 TO ANNOTATE PLOT POINT WITH DATA
TO STOP THE RUN AFTER LAST DATE BLOCK HAS BEEN PROCESSED,
SET ISTDA OF NEXT START/END DATE CARD = 99

DIMENSION IBUF(1000)
DIMENSION NB(4)
DIMENSION LABEL(20)
INREC = 0
IGREC = 0
IAREC = 0
IPREC = 0

PROGRAM TESTS FOR AREA AND GRAPH LIMITS
USES GRIDGSPBT, ISW, STAT, GETG, GETL
ENDIB, EVIL, SHTV, AND CALCOMP Routines

IIN = 105
IOUT = 108
C READ IN INFORMATION FOR PLOT LABEL
INPT=IIN
5021 READ(INPT,5022) LABEL
5022 FORMAT(20A4)
WRITE(IOUT,5023) LABEL
5023 FORMAT(1X,20A4)
C INITIALIZE PSEUDO-SWITCHES AND PLOTTER ROUTINE
CALL PLOTS (IBUF, *100C)
INIT = ISW(-2)
PLNUM=1*0
C PRINT DATE AND TIME OF JOB ON HEADING
5024 CALL TODAY(NOW)
5025 WRITE(IOUT,11) NOW
5026 11 FORMAT(1X,4A4)
CALL STAT
CALL SETSKP(IND)

C
NX = PLT(NX) FOR X VARIABLE
C
NY = PLT(NY) FOR Y VARIABLE
C
NZ = PLT(NZ) FOR Z VARIABLE
C
NW = PLT(NW) FOR W VARIABLE
C
XFAC = ENGINEERING UNITS PER INCH ON PLOT FOR X DIRECTION
C
YFAC = ENGINEERING UNITS PER INCH ON PLOT FOR Y DIRECTION
C
ZFAC = SAME FOR Z DIRECTION
C
WFAC = SAME FOR W DIRECTION
C
ANGB = ANGLE FOR DATAW ANOTATION
C
IDEC = PLOTTER CONTROL CODE FOR DECIMAL POINT IN DATAW ANOT
C
XINC = SPACING IN DECIMAL INCHES FOR ANOT IN X DIRECTION
C
YINC = SPACING IN DECIMAL INCHES FOR ANOT IN Y DIRECTION
C
READ (IIN, 2) ITAPE, NX, NY, NZ, NW, IDEC, KPT, KHT
C
2 FORMAT (8I5)
C
OUTPUT ITAPE, NX, NY, NZ, NW, IDEC, KPT, KHT
C
READ (IIN, 3) XFAC, YFAC, ZFAC, WFAC, ANGB, XINC, YINC
C
3 FORMAT (7F10.0)
C
OUTPUT XFAC, YFAC, ZFAC, WFAC, ANGB, XINC, YINC
C
ZX*KPT
C
ZHT=KPT*KHT
C
XINC*XINC*ZZ
C
YINC*YINC*ZZ
C
XFAC*XFAC*ZZ
C
YFAC*YFAC*ZZ
C
ZFAC*ZFAC*ZZ
C
WFAC*WFAC*ZZ
C
READ (IIN, 4) TBOP, BBT, DLEFT, RIGT
C
4 FORMAT (4F10.0)
C
OUTPUT TBOP, BBT, DLEFT, RIGT
C
TBOP=TBOP/YFAC
C
BBT=BBT/YFAC
C
DLEFT=DLEFT/XFAC
C
RIGT=RIGT/XFAC
C
LN=1
C
READ (IIN, 5) IDEG, AMIN
C
5 FORMAT (15s F10.0)
C
RADN=DEGTR(IDEG,AMIN)
C
GN TB(110,120,130,140)*LN
C
11C RTBOP*RADN
C
LN=2
C
GN TB 6
C
12C RBST*RADN
C
LN=3
C
GN TB 6
C
13C RLEFT*RADN
C
LN=4
C
GN TB 6
C
14C RIGT*RADN
C
402 CONTINUE
C
READ SPACECRAFT ALTITUDE (SVEC) BOUNDS IF SSW(4) #1
C
IF (ISW(4)#E0=0) GB TB 4G7
C
READ (IIN#403) BSVEC,TVEC
C
BSVEC IS LOWER ALTITUDE LIMIT, TVEC IS UPPER
0 FORMAT (2F10•3)
 1 C SET NZ = THAT FOR SVEC
 2 NZ = 1
 3 IFLAG = 0
 4 READ(IN,9)STDASISTD MB,ISTYR,ISTHM,IE NDA,IE NMB,IE NYR,IE NHM,IS KP
 5 9 FORMAT (312•I4,5X,312•I4,5X,15)
 6 IF (ISTDA•99) 404,305,305
 7 WRITE (IBUT,363) STDASISTD MB,ISTYR,ISTHM,IE NDA,IE NMB,IE NYR
 8 1 IE NHM,IS KP
 9 6365 FORMAT ('GRAFG91 START DATE ',312•I4, ' END DATE ',312•I4, ' CON T -
 10 1•I4)
 11 IF (IS KP•EG•C) GO TO 8
 12 CALL SKPREC (ITAPE, IS KP)
 13 8 CONTINUE
 14 WRITE (IBUT,7) PLNUM
 15 7 FORMAT ('START PLOT NUMBER = ',F4•0)
 16 CALL WHERE (XBRG, YBRG)
 17 CALL PLT (XBRG, YBRG, -3)
 18 ANNOTATING PLOT NUMBER
 19 CALL NUMBER (0,0,0,0,0,14•PLNUM, 0,0,1)
 20 ANNOTATE PLOT WITH LABEL AND DATE
 21 CALL SYMBOL (-2•0,0,0•O•14•LABEL, 90•80)
 22 CALL PLT (0,0,0,0,0,14•PLNUM, 0,0,1)
 23 MOVING PEN TO ORIGIN OF PLOT
 24 YT = (YBT/YFAC)
 25 IF (YT) 205, 210
 26 YT = 0
 27 XT = (CLEFT/XFAC)
 28 IF (XT) 215, 220
 29 XT = 0
 30 XT = XT + 10
 31 YT = YT + 10
 32 CALL PLT (XT, YT, 3)
 33 CALL GRID (Z2, HGT, XFAC, YFAC, T80, BBT, DLEFT, RIGT)
 34 PLNUM = PLNUM + 1
 35 CONTINUE
 36 START NEW PLOT
 37 IF (ISK•(10)) 24, 26
 38 CALL PLT (8•0,0,0,3)
 39 GO TO 10
 40 CONTINUE
 41 IF (ISK•(3)•EG•C) GO TO 302
 42 CALL GETL (ITAPE, NX, NY, NZ, NH, DATA X, DAT AY, DATA Z, DATA W, RLAT, 
 43 1 PLNG, JDA, JMB, JVR, JHM, IE BD)
 44 GO TO 304
 45 CALL GETG (ITAPE, NX, NY, NZ, NH, DATA X, DAT AY, DATA Z, DATA W, RLAT, 
 46 1 PLNG, JDA, JMB, JVR, JHM, IE BD)
 47 304 IF (IE BD) 350, 350, 305
 48 E0F OR E0T ENCOUNTERED DURING READ - - - GUT
 49 305 CALL PLT (XX, YY, 999)
 50 WRITE (IBUT,1020) INREC, IREC, IAREC, IPREC
 51 1020 FORMAT ('NUMBER RECORDS INPUT = ',1•I/, 
 52 'NUMBER RECORDS OUT OF GRAPH BOUNDS = ',1•I/, 
 53 'NUMBER RECORDS OUT OF AREA BOUNDS = ',1•I/)
180. 3 'NUMBER RECORDS PLOTTED * I,1)
181.  CALL EXIT
182.  C CHECKING FOR BEGINNING DATE
183.  CONTINUE
184.  INREC = INREC + 1
185.  KGDA = JDA
186.  KGMB = JMB
187.  KGYR = JYR
188.  KGHM = JHM
189.  CONTINUE
190.  C IFLAG IS A FLAG TO ALLOW SKIPPING THE FIRST CALL TO FIND
191.  C IF WE HAVE ALREADY FOUND THE STARTING DATE
192.  C
193.  IF(IFLAG.NE.0) GO TO 182
194.  CALL FIND(ISTD,ISTM,ISTYR,ISTHM,KGDA,KGMB,KGYR,KGHM,INDK)
195.  IF(INDK.EQ.-1) GO TO 300
196.  IFLAG = 1
197.  CONTINUE
198.  IF(IENYR.EQ.0) GO TO 851
199.  CALL FIND(IENTA,IENTM,IENTYR,IENTHM,KGDA,KGMB,KGYR,KGHM,INDK)
200.  IF(INDK.EQ.1) GO TO 995
201.  851 CONTINUE
202.  852 CONTINUE
203.  C CHECKING IF DATA WITHIN CHART BOUNDARIES
204.   85 IF(ISW(0))102,104,102
205.  102 WRITE(110UT,103)RLAT,RLONG,RTOP,RBOT,RLEFT,RRIGHT
206.  103 FORMAT(6E12.5)
207.  104 IF(RTBP.RLAT)100,100,86
208.  86 IF(RLAT.RBGT)100,88,88
209.  88 IF(RLENG.RLEFT)100,90,90
210.  9C IF(RRIGHT.RLONG)100,100,92
211.  C DATA WITHIN BOUNDS
212.  92 GO TO 35
213.  C DATA OUT OF AREA BOUNDS
214.  10C IAREC = IAREC + 1 GO TO 890
215.  35 CONTINUE
216.  C CHECKING IF LUNAR DATA IS WITHIN ALTITUDE BOUNDS
217.  C
218.  IF (ISW(4).EQ.0) GO TO 45
219.  IF ((DATAZ*BSEC).LE.0) GO TO 890
220.  IF ((TSVEC=DATAZ).LE.0) GO TO 890
221.  45 XX = DATA/XFAC
222.  YY = DATAY/YFAC
223.  C CHECKING IF DATA IS IN GRAPH BOUNDS
224.  IND = 0
225.  IF(TTOPY = YY)50,50,52
226.  5C YY = TTOPY
227.  5C IND = 1
228.  52 IF(YY = BBTY)54,54,60
229.  54 YY = BBTY
230.  54 IND = 1
231.  6C IF(XX = DLEFX)62,62,66
232.  62 XX = DLEFX
233.  62 IND = 1
234.  66 IF(RIGHTX = XX)68,68,69
235.  68 XX = RIGHTX
236.  68 IND = 1
237.  69 IF IND = 1 DATA POINT IS OUT OF GRAPH BOUNDS
238.  69 IF(IND.EQ.1) IGREC = IGREC + 1 GO TO 890
CALL PLOT(XX,YY,3)
CALL SPOT(XX,YY)
IPREC = IPREC + 1
IF(ISY(13))82,890,82
XT = XT + XINC
YT = YT + YINC
CALL NUMBER (XT, YT, HGT, DATAH, ANGB, IDEC)
CALL PLOT(XX,YY,3)
89C CONTINUE
IGC IF(IENY=EQ=0) GO TO 951
CALL FIND(IENDA, IENMA, IENYR, IENHM, KGDA, KGMB, KGYR, KGHM, INDK)
IF(IDK=EQ=0) GO TO 94C
GO TO 150
XRGBX = XRGBX + 50
YTBOTY
CALL PLOT(XT, YT, 3)
IF(ISW(7))402,402,6
951 GO TO 150
WRITE(IIBUT, 996) KGDA, KGMB, KGYR, KGHM
996 FORMAT('END DATE PASSED', 2X, 3I2, 14)
CALL PLOT(XX, YY, 999)
CALL EXIT
WRITE(IIBUT, 998) IND
998 FORMAT('ERROR IN SPREC, IND', 13)
CALL PLOT(XX, YY, 999)
CALL EXIT
GO TO 15
15C END
<table>
<thead>
<tr>
<th>INC</th>
<th>DTAE</th>
<th>NX</th>
<th>NY</th>
<th>NZ</th>
<th>NW</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC</td>
<td>00469</td>
<td>004CA</td>
<td>004CB</td>
<td>004CC</td>
<td>004CD</td>
</tr>
<tr>
<td>IDAC</td>
<td>00410</td>
<td>00411</td>
<td>00412</td>
<td>00413</td>
<td>00414</td>
</tr>
<tr>
<td>WAC</td>
<td>00416</td>
<td>00417</td>
<td>00418</td>
<td>00419</td>
<td>00420</td>
</tr>
<tr>
<td>AGT</td>
<td>00422</td>
<td>00423</td>
<td>00424</td>
<td>00425</td>
<td>00426</td>
</tr>
<tr>
<td>BAY</td>
<td>00428</td>
<td>00429</td>
<td>0042A</td>
<td>0042B</td>
<td>0042C</td>
</tr>
<tr>
<td>ARC</td>
<td>0042E</td>
<td>0042F</td>
<td>00430</td>
<td>00431</td>
<td>00432</td>
</tr>
<tr>
<td>IENDA</td>
<td>00434</td>
<td>00435</td>
<td>00436</td>
<td>00437</td>
<td>00438</td>
</tr>
<tr>
<td>VORB</td>
<td>0043A</td>
<td>0043B</td>
<td>0043C</td>
<td>0043D</td>
<td>0043E</td>
</tr>
<tr>
<td>DATAZ</td>
<td>00440</td>
<td>00441</td>
<td>00442</td>
<td>00443</td>
<td>00444</td>
</tr>
<tr>
<td>JUR</td>
<td>00446</td>
<td>00447</td>
<td>00448</td>
<td>00449</td>
<td>0044A</td>
</tr>
<tr>
<td>KGMS</td>
<td>0044C</td>
<td>0044D</td>
<td>0044E</td>
<td>0044F</td>
<td>00450</td>
</tr>
</tbody>
</table>

**Blank Common (0 Words)**

**External Subprograms Required:**

<table>
<thead>
<tr>
<th>CMYOR</th>
<th>EXIT</th>
<th>FIND</th>
<th>GETG</th>
<th>GETL</th>
<th>GRIDG</th>
<th>ISW</th>
<th>NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLOT</td>
<td>FLOT</td>
<td>SETSKP</td>
<td>SKPESC</td>
<td>SPET</td>
<td>STAT</td>
<td>Symbol</td>
<td>TODAY</td>
</tr>
<tr>
<td>WHERE</td>
<td>F101</td>
<td>F102</td>
<td>F103</td>
<td>F104</td>
<td>F105</td>
<td>F106</td>
<td>F107</td>
</tr>
<tr>
<td>MICE</td>
<td>F10C</td>
<td>9BCDREAD</td>
<td>9BCDRW</td>
<td>9ENDIVAL</td>
<td>9INITIAL</td>
<td>910DATA</td>
<td>910USA</td>
</tr>
<tr>
<td>SITOR</td>
<td>SPRINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Highest Error Severity: 0 (No Errors)**

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>821</td>
<td>00335</td>
</tr>
<tr>
<td>21</td>
<td>00015</td>
</tr>
<tr>
<td>1103</td>
<td>0044F</td>
</tr>
<tr>
<td>0</td>
<td>00000</td>
</tr>
<tr>
<td>1945</td>
<td>00799</td>
</tr>
</tbody>
</table>
C
PROMPT 640, 1
C
OUTPUT FOR GRAY J75 4PRL 1975
C
MODIFIED BY C BROWNE FROM FLUG CRON 4F 1 SEPT 1971
C
GRAY AT GRAYJ, REMEMBERS PRINCIPAL FACTS AT ENTRANCE OF 1 JULY 1969
AND Completes the 132 CHARACTER GUN FORMAT. OBSERVED
GRAY IS CALCULATED IF ISW(3) = 1
C
OUTPUT IS AT IGSN71 DATUM AND ANOMALIES CALCULATED WITH
INTERNATIONAL GRAVITY FORMULA 4F 1967.
C
ASSUMES PRISM DENSITY IS ENTERED AT RUN TIME
CALCULATES ANOMALY AT DENSITY ENTERED AT RUN TIME
C
DIMENSION TA(36)
C
G5.(7) = 1 FOR INPUT ELEV IN FEET
G5.(8) = 1 FOR INPUT ELEV IN METERS
G5.(9) = 1 FOR INPUT DEPTH IN FATHOMS
G5.(10) = 1 FOR INPUT DEPTH IN METERS
G5.(11) = 1 TO CALCULATE OBSERVED GRAVITY FROM ANOMALY
G5.(12) = 1 INPUT DATA IS ALREADY AT IGSN71 AND INT. GRAY FROM 1967
G5.(13) = 0 INPUT DATA AT ETSOAM SYSTEM AND INT. GRAY FROM 1950
G5.(14) = 1 FOR INCORPORATING TERRAIN CORRECTION
G5.(15) = 0 FOR NOT USING TERRAIN CORRECTION
G5.(16) = 1 TO PRINT VALUES OF TII' AND FELEV
G5.(17) = 1 TO ASSIGN A STATION NUMBER NUMERICALLY STARTING WITH
FIRST INPUT RECORD AS 1
G5.(18) = 1 TO OUTPUT ON LINE PRINTER ONLY (GINAT)
G5.(19) = 1 TO OUTPUT ON LINE PRINTER ON 2 CARDS EACH (GINAT)
G5.(20) = 1 IF DEPTH = 0, THEN PROGRAM ASSUMES DATA IS FOR A
LAND STATION AND USES ELEV IN EBOUGER COMPUTATION
GRAY ELEV AND DEPTH VALUES ARE IN METERS
G5.(21) = 0 USES STAT, GINAT, G167F, ISW, NAVIN, RT67F
C
***********************************************************************
C
ITAPE = UKN FOR INPUT
C
JTAPB = UKN FOR OUTPUT
C
ITAPBB = 0 FOR OUTPUT
C
ITAPB = 2
C
***********************************************************************
C
 II. = 106
INPUT = 105
CALL STAT
C
INPUT = 1SW(-2)
C
XCI= XIP= XPI= XP= IP= IC= 0
CALL I"
60 IREGC=0
61 IFCC=0
62 IF=CA
63 DB 1 J=1,35
64 IA(J)=1H
65 1 CONTINUE
66 C GROWN = ASSUMED CRUSTAL DENSITY
67 READ (IN, 2) GROWN
68 2 FORMAT (F10.5)
69 OUTPUT GROWN
70 C SET SSA(301) = 150 GINBOX WILL NOT TRY TO READ ETP CARD!
71 II = ICHG(301)
72 CALL GINBT(ITAPE,ITAPE,KK,KGDA,KGMB,KGRR,KGM,LAT,
73 1 KGYR,KGHR,IDIF,ISRRC,RLAT,RLNG,ELEV,K977,9BSG,
74 2 IDEF,FAC,FIC,TC,IELC,IGC,IFA,IREC,IFCC,IA,IFBC)
75 50 CONTINUE
76 51 READ (ITAPE,14) ISBRC,KGDA,KGMB,KGRR,KGM,KGYR,KGHR,LAT,
77 1 RLNG,RLAT,KEK,ELEV,K977,9BSG,
78 2 IDEF,TC,IELC,IGC,FAC,FIC,ABG,ABCM
79 80 F (MAT,14) ELEV = (13F6+17F5+2A113F5+2A17F7+2 = 1)
80 1 13F6+17F5+2A113F5+2A17F7+2
81 82 CALL STAT(1)
83 CALL EVIL(IOUT,1,IBAD,KGDA,KGMB,KGRR,KGHR)
84 IF (IBAD) GO TO 50, 53, 55
85 55 END FILE ITAPE
86 WRITE(IOUT,54) NCT
87 54 FORMAT(TEND OF PROCESSING, NCT = 179)
88 CALL EXIT
89 53 CONTINUE
90 IF (ISW(131)=1) ISN = ISN+1; KGHR = ISN
91 1 KGDA = 0; KGMB = 0; KGYR = 0
92 70 DEPT = IDEF
93 CALL MAVIN(LAT,RLATM,KNS,RLNG,RLAT,KEK,RLAT,RLNG)
94 THE0 = GINTF(RLAT)
95 THE7 = G162F(RLAT)
96 DIF = (K977 - 977) * 1000
97 98 C
99 C CONVERSION FEET TO METERS
100 40 ELEV = ELEV * 0.304801
101 42 FLEV = ((0*30+55 + 0.0022*C6S(2*RLAT)) * ELEV)
102 1 = (FLEV*0.001)**2.0)**0.072
103 FA = AFA
104 IF (ISW(3)=0) ZGGS = 9BSG + DIF + 508
105 IF (ISW(4)=0) ZGGS = (AFA - FLEV) + THE7 + 508
106 C CALCULATE OBSERVED GRAV IN 1930 IGF
107 C
108 C REHS = (AFA - FLEV) + THEB
109 50 TO 510
111 565 IF (ISW(4)=0) GO TO 550
112 REHS = ZGGBS
113 C
114 C CHANGE TO IGSN 71
115 510 ZGGBS = REHS - 14.0
116 320 TGGBS = ZGGBS * 0.001
117 IGGBS = TGGBS
118 VVAT = BT + TSGS
120 1B8G = ZG8BS - A
121 1C  CALCULATE FREE AIR ANOMALY
122 1C
123 1C
124 55C FA = (ZG8BS - TH)7 + FELEV
125 IF (ISK (3) 13) 45 43
126 43 WRITE (BP1 "45, TH, FELEV, TH 67"
127 44 FORMAT (3F10.2)
128 45 IF (ISK (2) 16, 20Q, 46
129 1C CONVERTING FATHOMS TO METERS
130 46 DEPTH = DEPTH * 18288
131 1DPP = DEPTH
132 1C CHECKING IF WATER STATION
133 CONTINUE
134 IF (DPP) 165 160 165
135 1C NO, ASSUME IT IS A LAND STATION
136 160 DENS = CRDEN
137 1C TICH = ELEV
138 GO TO 170
139 1C YES, IT IS A WATER STATION
140 165 DENS = CRDEN = 1.03
141 1C TICH = DEPTH
142 170 FG = FA + TO 041553 DENSTICH
143 1C INCORPORATE TERRAIN CORRECTION
144 IF (ISK (16) 176 174 76
145 74 TG = 0.9
146 80 BGCMM = 999 0
147 TG TO 80
148 76 IF (TG 99 7179 78 74
149 78 BGCMM = 999 0
150 GO TO 80
151 79 BGCMM = BG + TG
152 1C Rounding BGCMM
153 IF (BGCMM 1279 280 280
154 279 BGCMM = BGCMM = 0.05
155 GO TO 80
156 280 BGCMM = BGCMM = 0.05
157 1C Rounding DEPTH
158 80 IF (FAFA 201 220 220
159 201 FA + FA = 0.05
160 GO TO 280
161 220 FA + FA = 0.05
162 1C Rounding BBOGUER
163 280 IF (B255 260 260
164 255 FG + 5 = 3.05
165 GO TO 380
166 260 BGG = 3.05
167 380 CONTINUE
168 KK = 2
169 CALL GINBT (ITAPE, JTAPE, KK, KGDA, KGMB, 170 * GPR, KGHR, IDIF, IS0RC, RLAT, RLNG, ELEV, KG, ZBSG, 171 1 JDF, FA, FE, TC, CSEL, IGE, RFA, RGC, IFFC, IA, IFBC)
172 NCT NCT + 1
173 GO TO 50
174 END
### External Subroutines Required:

<table>
<thead>
<tr>
<th>Evil</th>
<th>Exit</th>
<th>Ginot</th>
<th>Gintf</th>
<th>G167f</th>
<th>ICHG</th>
<th>ISW</th>
<th>NAVIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1101</td>
<td>F1102</td>
<td>F1103</td>
<td>F1104</td>
<td>F1105</td>
<td>F1106</td>
<td>F1107</td>
<td>F1108</td>
</tr>
</tbody>
</table>

### Highest Error Severity: 7 (Number of Errors)

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>423</td>
<td>021A7</td>
</tr>
</tbody>
</table>

### Generated Code:
- 423
- 021A7

### Constants:
- 1
- 0001A

### Legal Variables:
- 26
- 0001C

### Total Program:
- 52
- 0021E
PROGRAM GSTOG

PROGRAM GSTOG, CONVERTS SEAG1 FORMAT TO GSUM FORMAT

VERSION 25 JUNE 75 TO ADD OUTPUT STATEMENT
VERSION 13 JAN 75 TO READ IREC=2
VERSION OF 4 OCT 1971, TO USE GINBT FOR OUTPUT
VERSION OF 20 JANUARY 1971

SIGMA 7 VERSION HANDLES SINGLE REEL INPUT ONLY

SSW(12) UP TO LIST DATA IDENTIFICATION

USES BICOR, STAT, ISW, EVIL

CALLS STAT, MCVOL, GETGEE FROM ACCOUNT 3 LIBRARY

DIMENSION MAG2(2)
DIMENSION MAG1(5), MAG2(5), IBLK(21)
DIMENSION IA(35), IZ(9), IH(35)
DIMENSION NWW(4)
IN = 105
OUT = 108

PRINT DATE AND TIME OF JOB ON HEADING
CALL TODAY(NOW)
WRITE(OUT, 13) NOW

OUTPUT: GSTOG VERSION OF 25 JUNE 75

IREC1=1

FOR MULTIFILE MAGTAPE INPUT
NEF=1

FOR WRITING EBF ON JTAPE
IFLAG=1

CALL STAT
IN = ISW(=2)
NZER=0
KGDA = NZER
KGM8=NZER
KGYR=NZER
KGM8=NZER

***********************
ITAPE = INPUT TAPE
JTAPE = OUTPUT TAPE
ITAPE = 1
JTAPE = 2

***********************

IDIF=0
ELEV=0
TC=99.9
RF=0
IREG=0
IFCC=7
IFBC=0
NREC = 0
ICHECK = 0
NOUT = 0
DEGRA = 1.745329E-2

C ISORC = SOURCE NUMBER
IDCD = 0 FOR ID BY DATE, 1 FOR ID BY STA NO.
IELC = ELEVATION CODE
IGC = G-METER CODE
BIAS = MGAL BIAS
NFILE = NUMBER OF FILES TO BE INPUT

READ (IIN, 2) ISORC, IDCD, IELC, IGC, BIAS
FORMAT (15, F10.0)
OUTPUT ISORC, IDCD, IELC, IGC, BIAS
READ (IIN, 5) NFILE
FORMAT (15)
WRITE(IOUT, 7786) NFILE
FORMAT(1GST56; NFILE=1, I5)

IF(IDCD) 230 226 230
KGDA = 0
KGM = 0
KGYR = 0

C CONTINUE
K = 0
CALL GIN8(TAPE, ITAPE, KM, KGDA, KGM, KGYR, KGDA, KGMB)
1 KGYR, KGMB, IDIF, ISORC, R LAT, R LNG, ELEV, K977, BS 6
2 IDEP, F A, BG, TC, IELC, IGC, RFA, IREC, IFFC, IA, IFBC

C READ INPUT DATA
CBINTILE
READ(ITAPE, 360) IREC, KGDA, KGM, KGYR, KGMB, IDIF,
1 RLAT, R LNG, KVE, K977, KGR, KFA, KBG, KCVN,
2 KCV, KCD, MDC, MTC, MAG(1), MAG(2), KETV
FORMAT(I1,312, 14, I3, 2 F9.6, 215, 13, I4, 51S, 13)
CALL STAT(I)
CALL EIVL(IOUT, I, IBAD, KGDA, KGM, KGYR, KGMB)
IBAD = 0, PARITY OR FORMAT ERROR
IBAD = 9, END OF FILE OR END OF TAPE ENCOUNTERED IF (IBAD) 50, 53, 575
IF (IREC) 50, 53, 575
IF (IREC NE 1) G0 TO 60
IF (ICHECK EQ 1) G0 TO 70
BLT PUT**** INPUT IS IN 1930 IGF *****
ICHECK = 1
G0 TO 70
IF (IREC EQ 5) 60, 62, 50
READ(TAPE, 64) IREC9, IZ, IZ1, ITEST
FORMAT(I12, I13, I14)
IF (IEST EQ 65631580, 56558C
566 WRITE(IOUT, 570)
57C FORMAT(1EBR)
CS PAUSE 400
cs G0 TO 50
EVIL HAS FOUND AN END OF FILE MARK ON INPUT

IF (NEF = NFILE) NEF = NEF + 1
CALL MVCBL (ITAPE)
GO TO 50
END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED

IF (IFLAG - NE) GO TO 50
END FILE TAPE
WRITE (IOUT, 579) NREQ, NOUT
WRITE (IOUT, 601) NOUT
GO TO 575
END OF PROCESSING

NUMBER OF RECORDS REJECTED BECAUSE OF INVALID GRAVITY =
NUMBER OF RECORDS OUTPUT =
CALL EXIT
WRITE (IOUT, 665) 68, 65, 68
WRITE (IOUT, 601)
GO TO 575
END FILE TAPE
WRITE (IOUT, 69)
FORMAT ('IREC1=91')
GO TO 50
CONTINUE
KGDA = KGDA
KGMB = KGMB
KGYR = KGYR
KGHM = KGHM

CHECKING GRAVITY FOR INVALID DATA

IF (KAF = 0) NREQ = NREQ +1 GO TO 50
IF (KAF = 9999) NREQ = NREQ +1 GO TO 50
BSG = FLOAT (IBG) * 0.1
FAL = FLOAT (KFA) * 0.1
BG = FLOAT (KGB) * 0.1
IDEP = KCDM
CALL BICER (K977, 8BSG, BIAS)
IF (FA = 990.0) 85, 87
FA = FA + BIAS
IF (BG = 990.0) 88, 90, 90
BG = BG + BIAS
CONTINUE

GROUPING VARIABLES FOR OUTPUT UNDER ARRAY IA
ENC0DE (35, 40, 12) KV9, KVE, KCY, KCY, KETV9, KTMDC, MT
FORMAT (415, 15, 13, 12, 5X)
CALL UNPKBY (121, IW35)
DB 420 J = 1, 35
IA(J) = 13 (IW(J), 24)
CONTINUE

KK = 2
CALL GINOT (ITAPE, JTAPE, KK, KGDA, KGMB)
1 KGYR, KGHM, IDIF, ISRC, RLAT, RLON, ELEV, K977, 8BSG
2 IDEP, FA, BG, TC, IELC, IGC, RFA, IREGC, IFFC, IA, IFBC)
NBUT = NOUT + 1
GO TO 50

WRITE END OF FILE RECORD

END FILE TAPE
WRITE (IOUT, 330) KGDA, KGMB, KGYR, KGHM
FORMAT ('DATA BEFORE E8R= ', 313, 15)
CALL EXIT
END
BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

FLOAT ISL

EXTERNAL SUBPROGRAMS REQUIRED:

BICER EVIL EXIT GINOT ISM KCVBL STAT TODAY
UNPKBY F1101 F1102 F1103 F1104 F1105 F1106 F1108
FIO M10C 98CDREAD 98CCPRINT 98CDE 9ENCODE SENDFILE SENDIOG GINIT
910DATA 910LLSA 910R 9PRINT 9STOP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>487</td>
<td>C01E7</td>
</tr>
</tbody>
</table>

CONSTANTS: 6

LOCAL VARIABLES: 176

TEMPS: 0

TOTAL PROGRAM: 669
PROGRAM GST6G67
PROGRAM GST6G67 MOD OF GST6G TO G8 1967 FORMULA AND 71 DATUM
CHANGE WHERE NECESSARY
MOD FROM GST6G OF 25 JUNE 75
PROGRAM GST6G, CONVERTS BEAG1 FORMAT TO GSUM FORMAT

VERSION OF 10 MAR 76 TO FIX IRECC FOR I/P AT 67
VERSION OF 5 AUG 75 TO DB 67 FORMULA CONVERSION
AND TO WRITE GSUM 8/P BLOCKED BY 50
OUTPUT GST6G67 VERSION 10 MAR 76!
VERSION 25 JUNE 75, TO ADD OUTPUT STATEMENT
VERSION 13 JAN 75 TO READ IRECC2
VERSION OF 4 OCT 1971, TO USE GINST FOR OUTPUT
VERSION OF 20 JANUARY 1971

SIGMA 7 VERSION HANDLES SINGLE REEL INPUT ONLY

SSW(12) UP TO LIST DATA IDENTIFICATION

DIMENSION IBUFST(3250)
DIMENSION MAG(2)
DIMENSION MAG1(5), MAG2(5), IBLK(21)
DIMENSION IA(35), IZ(9), IH(35)
DIMENSION NOW(4)
IN = 105
IOUT = 108
KI*1
K0=2

PRINT DATE AND TIME OF JOB ON HEADING
CALL TODAY(NOW)
WRITE(IOUT,13) NOW
13 FORMAT(1X,4A4)

RADEG57=2958

215 IREC=1

FOR MULTIFILE MAGTAPE INPUT
NEF=1

FOR WRITING EBF ON JTAPE
IFLAG=1
NRUT=0
INN = IWH(2)
IAKEY=0
NIN=0
NZERO=0
KGDAO = NZERO
KGM80 = NZERO
KGYRO = NZERO
KGMRO = NZERO

ITAPE = INPUT TAPE
JTAPE = OUTPUT TAPE
ITAPE = 1
JTAPE = 2
The code snippet begins with a comment block that seems to describe the purpose of the program. It appears to be a scientific or computational program, likely related to data processing or analysis. The program includes variable definitions, data input and output formats, and conditional statements that control the flow of the program.

The code structures include comments indicating logical operations and data manipulations. It's not immediately clear what the program's specific tasks are without understanding the overall context and purpose. The presence of variables like `ISORC`, `TELC`, `IGC`, `BIAS`, and `NFILE` suggests it deals with data input and output, possibly in a scientific or engineering context.

The program flow involves reading input data from a file, performing calculations, and writing results to another file. The presence of conditional statements like `IF (REC1.EQ.2) GO TO 70` and `IF (REC1.EQ.1) OUTPUT 'IREC NE 1 OR 2'` indicates decision-making processes. The program also includes mathematical operations, such as trigonometric calculations, which are typical in scientific computations.

Overall, the program seems to be a part of a larger data processing system, possibly for numerical or scientific analysis.
C  END OF FILE

575  IF (NEF .NE. NFILE) 576, 577, 577

576  NEF = NEF + 1
577  CALL MCVOL (ITAPE)
578  GO TO 575
579  C  END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
577  IF (IFLAG NE 1) GO TO 578
578  C  LAST BUFFER OUT
578  IF (NROUT EQ 0) GO TO 1577
579  CALL BUFFER OUT (UTAPE, IBUFST (1, 1), NROUT + 32)
580  CONTINUE
581  C  END FILE UTAPe
582  WRITE (IBUT, 579) NREQ, NOUT
583  FORMAT ('END OF DATA - REQUIRED NO. OF FILES NOW PROCESSED)
584  GO TO 575
585  IF (NROUT EQ 0) CALL BUFFER OUT (UTAPE, IBUFST (1, 1), NROUT + 32)
586  CONTINUE
587  C  CHECKING GRAVITY FOR INVALID DATA
588  C  IF (K977 LE 0) NREQ = NREQ + 1 GO TO 50
589  IF (KFA GE 9900) NREQ = NREQ + 1 GO TO 50
590  DLAT = DLAT + 90.01 LTKEY = DLAT
591  DLONG = DLONG + 180.01 LGKEY = DLONG
592  BS = FLOAT (180.0) * 0.1
593  FA = FLOAT (9999) * 0.1
594  BS = FLOAT (K977) * 0.1
595  BS = BS + BIAS
596  CONTINUE
597  C  GROUPING VARIABLES FOR OUTPUT UNDER ARRAY IA
598  CALL GDE (35, 410, IZKVN, IZKVE, IZKCVE, IZKEV, IZMTG, MTG)
599  IF (NROUT EQ 50) CALL BUFFER OUT (UTAPE, IBUFST (1, 1), 1600, JKEY)
600  GO TO 1578
601  IF (NROUT NE NROUT + 1) CALL BUFFER OUT (UTAPE, IBUFST (1, NROUT + 1), NE)
<table>
<thead>
<tr>
<th>DEC Words</th>
<th>HEX Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>544</td>
<td>00220</td>
</tr>
<tr>
<td>15</td>
<td>0000F</td>
</tr>
<tr>
<td>1790</td>
<td>006FE</td>
</tr>
<tr>
<td>1</td>
<td>00001</td>
</tr>
<tr>
<td>2350</td>
<td>0092E</td>
</tr>
</tbody>
</table>
PROGRAM G30CP

VERSION OF 1 MAY 75 TO ZERO EVERYTHING IN SIGHT
VERSION OF 11 APR 75 FOR XEROX CPV COMPATIBILITY
VERSION OF 75 TO INCREASE PRINT SPACE FOR RH=Z
VERSION 5 FEB 75 TO EASE INTERFACE WITH I/P PREP
BUS VERSION OF 22/7/74 TO COMPUTE POTENTIAL FROM BODY
OUTPUTS HEIGHT OF SEA (+ COV) AND MGAL DIFFERENCE
VERSION OF 20 MAY 74 TO INCREASE RESOLUTION OF Z
VERSION OF 23 APRIL 73 TO INCREASE RESOLUTION OF PRINTED VALUE OF
WEIGHT AND WEIGHT

VERSION OF 29 MARCH 73 TO CHANGE SS= 19 & 20 TO 385, 39
VERSION 20 MARCH 1973 TO CALL PLANET FOR RADIUS (LAG)
3D PLAG: FIELD PRINTS ARE GIVEN BY K*K*K* CANTOURS ARE GIVEN
BY "", THE VERTICES FOR EACH CANTOUR ARE GIVEN BY I1,I2,I3
DIFFERENT FOR EACH CANTOUR AND IS GIVEN BY I*E(M), MID(M) IS 1
FOR EACH CANTO
SOME SWITCH OPTIONS
SS= (385) FOR SPHERICAL EARTH
=1 FOR SPHERICAL MOON
=2 FOR SPHERICAL MARS
=3 FOR SPHERICAL SATURN
COMMON U(I)U(10),UH(20),UOH(20),UHZ(20),UHM(20),UH2M(20)
COMMON XX(I),YY(I),LZ(I),BEL(102),SEL(102),E(102),SIGMA(102)
COMMON UZ(I),UL(30),LT(30),G4(102),UZT(30),IMM(30)
COMMON III(I),RNM(102),ZEE(I),ZEE(102),MID(I)
COMMON F(I),PDEL(I),PDEL(102),PDEL(102)
DOUBLE PRECISION RVDR,ANG,RZ,TT,TT
INTEGER RD
C02C I=1 TO
PEL(I)*0=0
BEL(I)*0=0
SEL(I)*0=0
EL(I)*0=0
E(I)*0=0
F(I)*0=0
G(I)*0=0
III(I)*0
MID(I)*0
PDEL(I)*0=0
PDELF(I)*0=0
RNM(I)*0=0
SIGMA(I)*0=0
V(I)*0=0
ZEE(I)*0=0
ZEEF(I)*0=0
CONTINUE
C02C I=1 TO
IMM(I)*0
LT(I)*0=0
UL(I)*0=0
UZT(I)*0=0
LZU(I)*0=0
CONTINUE
C02C I=1 TO
IMM(I)*0
CONTINUE
C02C I=1 TO
IMM(I)*0

CONTINUE
60. \( U_{III}(I,J)=0.0 \)
61. \( U_{III}(I,J)=0.0 \)
62. \( U_{III}(I,J)=0.0 \)
63. \( U_{III}(I,J)=0.0 \)
64. \( 24 \) **CONTINUE**
65. \( 23 \) **CONTINUE**
66. \( D8=25, I=1,20 \)
67. \( D8=26, I=1,20 \)
68. \( D8=27, I=1,10 \)
69. \( X(I,J,K)=0.0 \)
70. \( Y(I,J,K)=0.0 \)
71. \( 27 \) **CONTINUE**
72. \( 26 \) **CONTINUE**
73. \( 25 \) **CONTINUE**
74. \( D8=28, I=1,20 \)
75. \( XX(I)=0.0 \)
76. \( YY(I)=0.0 \)
77. \( 28 \) **CONTINUE**
78. \( A=0.0 \)
79. \( AA=0.0 \)
80. \( ALPH1=0.0 \)
81. \( ALPH2=0.0 \)
82. \( ANM=0.0 \)
83. \( ANM1=0.0 \)
84. \( B=0.0 \)
85. \( BB=0.0 \)
86. \( BDC=0.0 \)
87. \( BETA1=0.0 \)
88. \( BETA2=0.0 \)
89. \( BG=0.0 \)
90. \( C=0.0 \)
91. \( CC=0.0 \)
92. \( D=0.0 \)
93. \( DELT1=0.0 \)
94. \( DELT2=0.0 \)
95. \( DBG=0.0 \)
96. \( C9G=0.0 \)
97. \( DBGGS=0.0 \)
98. \( DBGS=0.0 \)
99. \( DRA=0.0 \)
100. \( DZZ=0.0 \)
101. \( EGA=0.0 \)
102. \( ELEV=0.0 \)
103. \( EGA=0.0 \)
104. \( EMM=0.0 \)
105. \( FAG=0.0 \)
106. \( FELZ=0.0 \)
107. \( FX=0.0 \)
108. \( FY=0.0 \)
109. \( FZ=0.0 \)
110. \( GAMM1=0.0 \)
111. \( GAMM2=0.0 \)
112. \( GM=0.0 \)
113. \( I=0 \)
114. \( IBEGG=0 \)
115. \( IDEP=0 \)
116. \( IDIF=0 \)
117. \( IFLC=0 \)
118. \( IFBC=0 \)
| 120. | GGC=0 |
| 121. | IC=0 |
| 122. | IN=0 |
| 123. | INCARD=0 |
| 124. | INIT=0 |
| 125. | ISUT=0 |
| 126. | IRMA=0 |
| 127. | IS8RC=0 |
| 128. | ITAPE=0 |
| 129. | ITST=0 |
| 130. | JTAPE=0 |
| 131. | JST=0 |
| 132. | K=0 |
| 133. | KDA=0 |
| 134. | KGM=0 |
| 135. | KGMO=0 |
| 136. | KGY=0 |
| 137. | K=0 |
| 138. | KZ=0 |
| 139. | K77=0 |
| 140. | L=0 |
| 141. | LD=0 |
| 142. | M=0 |
| 143. | MM=0 |
| 144. | M=0 |
| 145. | MB=0 |
| 146. | M=0 |
| 147. | MRS=0 |
| 148. | MUN=0 |
| 149. | NGB=0 |
| 150. | NGB=0 |
| 151. | GBS=0 |
| 152. | P=0 |
| 153. | PAMS=0 |
| 154. | PAREZ=0 |
| 155. | PBE=0 |
| 156. | PC=0 |
| 157. | PDENS=0 |
| 158. | PFELZ=0 |
| 159. | PIE=0 |
| 160. | PMASS=0 |
| 161. | PSFELZ=0 |
| 162. | PSI=0 |
| 163. | RAO=0 |
| 164. | RAFA=0 |
| 165. | RHCZ=0 |
| 166. | RHBZ=0 |
| 167. | RKM=0 |
| 168. | RLOT=0 |
| 169. | RLNS=0 |
| 170. | RVAR=0 |
| 171. | RVR=0 |
| 172. | R=0 |
| 173. | R=0 |
| 174. | R=0 |
| 175. | S=0 |
| 176. | SFELZ=0 |
| 177. | SHCZ=0 |
| 178. | SHBZ=0 |
| 179. | SIGA=0 |
IF (MN) 30 40 3
IF (M = 2) 5 9 4 0 5

READ (1X, 1030) UMID (MN, BD), URHB (MN, BD), UZEE (MN, BD), UDM
1030 FORMAT (1H, 12, 16, 4, 16, 6, 6)

WRITE (15, 1031) UMID (MN, BD), URHB (MN, BD), UZEE (MN, BD), UDM
1031 FORMAT (1, 12, 16, 4, 16, 6, 6)

IN = 0

IF (MN = "") 3 4 0 3

3 IF (M = 2) 5, 4, 5
CONTOUR DATA //128H K XGDC140

SIGMA A RH0*Z V DELTA PRIME DELTA /1H /

IF(IS(1)==0) G0 TO 17

CALL GIBRT(1)TAPE,JTAPE,Kx,KO,Fx,Fy,Fz,FAG,LDP

1078 FORMAT(1H1,J2,F8.2,F12.2, F12.2)

17 READ(INCAR,1023) FX, FY, FZ, FAG, LDP

123 FORMAT(3F24.15)

1090 ALPH1 = X(M1,1,BD) = FX

1091 R1 = SQRT(ALPH1 * ALPH1 + BET1 ** 2)

1110 IF (R1) 1100 1105 1100

1110.LI = III(M,1,1,M1,1,1) = XM1,1,1,1,1

1111 FORMAT(1H1, //129H VERTECES=129 DEPTH=72,11H DENSITY=125000)

1214=2/101H X(I) Y(I) Z(I) A C D PARFEZ //1H )

1311 IF (P) 1350 1350 1350 1351

1314 IF (P) 1350 1350 1350 1351

1317 IF (P) 1350 1350 1350 1351

1319 IF (P) 1350 1350 1350 1351

1323 READ(INCAR,1023) FX, FY, FZ, FAG, LDP
1156 S = *i.
1157 GO TO 1170
1160 S = 1.
1167 EMK = BETA1 + ALPH2 = BETA2 + ALPH1
1180 IF (EMK) 1190 1350 1200
1190 N = -1.
1200 GO TO 1210
1206 IF(Z1) 1210 1210 1210
1210 PSI*EXP((Z/SQRT(F(P**2*Z**2)))
1212 AA = GAMM1*GAMM2*DEL1*DEL2
1220 IF (AA) 1220 1220 1230
1226 A = N * 1* 570796327
1232 GO TO 1240
1238 A = N * (ATAN(F(SQRT(F(1* AA* 2 ))) / AA) + 3*141592654)
1244 GO TO 1240
1249 A = N * (ATAN(F(SQRT(F(1* AA* 2 ))) / AA) + 1240 1240 1240
1250 EQ = B*C
1256 GO TO 1270
1262 GO TO 1270
1268 B = (ATANF((SQRT(F(1* AA* 2 ))) / AA) + 1274 1274 1274
1270 C = 1*570796327
1276 C = 1*570796327
1282 IF (CC) 1282 1282 1282
1288 CC = (PSI * ( EGA + GAMM1 + TAU*DEL11))
1294 IF (CC) 1294 1294 1294
1300 EQ = B*C
1306 IF (CC) 1306 1306 1306
1312 IF (CC) 1312 1312 1312
1318 CC = 1*570796327
1324 C = 1*570796327
1330 C = 1*570796327
1336 C = 1*570796327
1342 C = 1*570796327
1348 C = 1*570796327
1354 C = 1*570796327
1360 C = 1*570796327
1366 C = 1*570796327
1372 C = 1*570796327
1378 C = 1*570796327
1384 EQ = B*C
1390 A = C
1396 B = 0
1402 C = 0
1408 D = 0
1414 IF(A) 1370 1390 1370
1420 IRMA = D8G*DB8G*DB8G*DB8G*A*B*C*D*PARFEZ
1426 WRITE(14UT, 1385)
1432 WRITE(14UT, 1398)
218

9998 FORMAT(4E18.7) G3DC2170
161 G3DC2190
1380 SFELZ*SFELZ*SFELZ*PSFELZ*PSFELZ G3DC2190
SIGA*SIGA*SIGA G3DC2210
LAG C G3DC2220
K*FKM*ZEE(1,1) G3DC2230
DRA*CBELE(RA) G3DC2240
DZZ*CBELE(ZZ) G3DC2250
RVAR*RVAR G3DC2260
1398 T*S1*ZGRTRF(Z**2+RVAR**2) G3DC2270
PSFELZ*PSFELZ*PSFELZ*PSFELZ*PSFELZ
SIGA*SIGA*SIGA
L**2 C
K*FKM*ZEE(1,1) G3DC2280
DRA*CBELE(RA) G3DC2290
DZZ*CBELE(ZZ) G3DC2300
RVAR*RVAR G3DC2310
G8 TN 1398 G3DC2320
1393 T*TV*STU G3DC2330
RVAR*RVAR*8*SIGN(A1)/(R11+R21)*SIGN(A2)) G3DC2340
RVAR*RVAR*8 G3DC2350
G9 TN 1398 G3DC2360
1394 T*TV*STU G3DC2370
RVAR*RVAR*2*SIGN(A2)/(R11+R21)*SIGN(A2)) G3DC2380
G9 TN 1398 G3DC2390
1395 T*TV*STU G3DC2400
G9 TN 1398 G3DC2410
1396 T*TV*STU*(A/96) G3DC2420
STV=STV+TV G3DC2430
1400 ALP=ALP+2 G3DC2440
BETA=BETA G3DC2450
GAM=GAM+2 G3DC2460
DELT1=DELT2 G3DC2470
R1=R2 G3DC2480
1410 CONTINUE G3DC2490
1420 IF(SIGA)20122022203 G3DC2500
201 IF(SIGA1100011204265,265 G3DC2510
205 SFELZ*SFELZ*SIGA G3DC2520
208 G3DC2530
G3DC2540
201 G3DC2560
G3DC2570
204 G3DC2580
G3DC2590
201 G3DC2600
G3DC2610
G3DC2620
207 G3DC2630
G3DC2640
G3DC2650
G3DC2660
G3DC2670
G3DC2680
G3DC2690
G3DC2700
G3DC2710
480.  SIGA(M) = SIGA
481.  1430 CONTINUE
482.  IF(LA1650 = 1)1660
483.  1600 M = 1
484.  486.  MID(M) = MID
485.  111(M) = 1
486.  ZEE(M) = ZE
487.  R = (ZEE(M) + ZEE(M1))
488.  SIF(M) = 0
489.  V = (V)VL
490.  E(1) = V
491.  F(1) = V
492.  G6 TO 1620
493.  1610 M = 2
494.  1620 IF(T1, 1630 = 1620)
495.  1630 M = M + 1
496.  499.  MIG(M) = MID(M)+1
497.  111(M) = 1
498.  ZEE(M) = ZE
499.  R = (ZEE(M) + ZEE(M1))
500.  SIF(M) = 0
501.  V = (V)VL
502.  E(1) = V
503.  F(1) = V
504.  G6 TO 1432
505.  1432
506.  PDEL(M) = 0
507.  DELP(M) = 0
508.  DELP(M+1) = 0
509.  512.  DELP(M) = 0
513.  DELP(M+1) = 0
514.  ANM(M)
515.  BNL(M) = 0
516.  BNL(M+1) = 0
517.  BNL(M+2) = 0
518.  ANM(M+1)
519.  PDEL(M) = 0
520.  PDEL(M+1) = 0
521.  PDEL(M+2) = 0
522.  PDEL(M+3) = 0
523.  DEL(M+1) = (V(M) - ((ZEE(M) - ZEE(M+1))/ZEE(M)+ZEE(M+2))) * F630C360
524.  1130 C = ZEE(M+2) = 2 * ZEE(M) - ZEE(M1) + V(M1) * ((ZEE(M) - ZEE(M+1))/ZEE(M)+ZEE(M+2))
525.  PDEL(M+1) = ((ZEE(M) - ZEE(M+1)) * 3) / (ZEE(M) - ZEE(M+1))
526.  527.  47(M+2) = ((ZEE(M) - ZEE(M+1))/6)
528.  BNL(M+1) = (ZEE(M) + ZEE(M+1))/ZEE(M)
529.  1170 C = ZEE(M+2) = 2 * ZEE(M) - ZEE(M1) + V(M1) * ((ZEE(M) - ZEE(M+1))/ZEE(M)+ZEE(M+2))
530.  ZEE(M+1)/ZEE(M+1) = ZEE(M+1) * (3 * ZEE(M+2) - 2 * ZEE(M+1) - ZEE(M1) - ZEE(M+2))
531.  ZEE(M+1) + F(M+2) = ((ZEE(M) - ZEE(M+1)) * 3) / (ZEE(M) - ZEE(M+1))
532.  PDEL(M+1) = (F(M)/((ZEE(M) - ZEE(M+1))/ZEE(M)+ZEE(M+2)) * 1130 C = ZEE(M+2) = 2 * ZEE(M) - ZEE(M1) + V(M1) * ((ZEE(M) - ZEE(M+1))/ZEE(M)+ZEE(M+2))
533.  ZEE(M+1)/ZEE(M+1) = ZEE(M+1) * (3 * ZEE(M+2) - 2 * ZEE(M+1) - ZEE(M1) - ZEE(M+2))
534.  ZEE(M+1) + F(M+2) = ((ZEE(M) - ZEE(M+1)) * 3) / (ZEE(M) - ZEE(M+1))
535.  47(M+1) = (ZEE(M) - ZEE(M+1))/60
536.  DELP(M+2) = (V(M) - ((ZEE(M) - ZEE(M+1))/ZEE(M)+ZEE(M+2)) * 3) / (ZEE(M) - ZEE(M+1))
537.  47(M+1) = (ZEE(M) - ZEE(M+1))/60
210. \[ \text{ANIMAL} = \text{E}_{12+4}, 	ext{ANT} = \text{E}_{12+4} \]
211. \[ \text{FLAT SEASURF} = \text{E}_{12+6}, \text{HEIGHT METERS} = \text{E}_{12+4} \]
212. \[ \text{FLAT SEASURF} = \text{E}_{12+6}, \text{HEIGHT METERS} = \text{E}_{12+4} \]
213. \[ \text{FLAT SEASURF} = \text{E}_{12+6}, \text{HEIGHT METERS} = \text{E}_{12+4} \]
214. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
215. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
216. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
217. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
218. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
219. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
220. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
221. \[ \text{WRITE}(\text{OUT}, \text{1990}) (\text{V}(M), \text{N} = \text{M} + \text{MP}) \]
BLANK COMMON (10717 WORDS):

EXTERNAL SUBPROGRAMS REQUIRED:

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

GENERATED CODE: 2865
CONSTANTS: 36
LOCAL VARIABLES: 139
TEMPS: 12
TOTAL PROGRAM: 3074
PROGRAM G3DC PREP
VERSION 5 FEB 75 TO USE REFERENCE DENSITY
VERSION OF 23 MAY 74 TO INCREASE RESOLUTION OF Z
VERSION 1 MAR 74 TO RENUMBER LAMINAE
PROGRAM TO PREPARE I/P TO G3D
DIMENSION ICARD(60),NSL(10)
DIMENSION REFD(10)
OUTPLT 'G3DCPREP VERSION 5 FEB 75'
LBP=1
ITAPE=0
JTAPE=7
IIN=105
10C READ(IIN,1004) NUMBBD
D0 105 I=1[NUMBBD
READ(IIN,1006) REFD(I)
10S CONTINUE
D0 110 I=1[NUMBBD
READ(I3,10C4) NSLCNT
NSL(I)=NSLCNT
11C CONTINUE
D0 500 I=1[NUMBBD
NCNTP=0
ITAPE=ITAPE+1
IF(NUMBBBD) WRITE(JTAPE,1005) NSL(I),LBP; G0 TO 130
WRITE(VTAPE,1004) NSL(I)
13C CONTINUE
READ(ITAPE,1002,END=400) NCNT,RHO,Z
NCNTP=NCNTP+1
RHO=RHO=REFD(I)
WRITE(JTAPE,1002) NCNTP,RHO,Z
WRITE(JTAPE,1003) X,Y,LSLPT
18C IF(NE=1) G0 TO 18C
G0 TO 170
17C CONTINUE
CONTINUE
CONTINUE
STOP
C *********
C FORMATS
C *********
1001 FORMAT(80A1)
1002 FORMAT(12,F10.4,F16.6,F6.3)
1003 FORMAT(2F10.5,I1)
1004 FORMAT(12)
1005 FORMAT(12,28X,I1)
1006 FORMAT(F10.0)
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>00069</td>
<td>V</td>
<td>I</td>
<td>I</td>
<td>ARRAY</td>
<td>00000</td>
<td>V</td>
<td>80</td>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>00067</td>
</tr>
<tr>
<td>ITAPE</td>
<td>I</td>
<td>SCALR</td>
<td>00066</td>
<td>V</td>
<td>JTAPE</td>
<td>I</td>
<td>SCALR</td>
<td>00066</td>
<td>V</td>
<td>1</td>
<td>LBP</td>
<td>I</td>
<td>SCALR</td>
<td>00064</td>
</tr>
<tr>
<td>LSLPT</td>
<td>I</td>
<td>SCALR</td>
<td>00071</td>
<td>V</td>
<td>NCNT</td>
<td>I</td>
<td>SCALR</td>
<td>00066</td>
<td>V</td>
<td>1</td>
<td>NCNTP</td>
<td>I</td>
<td>SCALR</td>
<td>00068</td>
</tr>
<tr>
<td>NBL</td>
<td>I</td>
<td>ARRAY</td>
<td>00050</td>
<td>V</td>
<td>NSLCNT</td>
<td>I</td>
<td>SCALR</td>
<td>00064</td>
<td>V</td>
<td>1</td>
<td>NUBBD</td>
<td>I</td>
<td>SCALR</td>
<td>00068</td>
</tr>
<tr>
<td>REFC</td>
<td>R</td>
<td>ARRAY</td>
<td>00050</td>
<td>V</td>
<td>RHB</td>
<td>R</td>
<td>SCALR</td>
<td>00064</td>
<td>V</td>
<td>1</td>
<td>X</td>
<td>R</td>
<td>SCALR</td>
<td>00064</td>
</tr>
<tr>
<td>Y</td>
<td>I</td>
<td>SCALR</td>
<td>00070</td>
<td>V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (114 WORDS):**

- 00000 ICARD 00050 NSL 0006A REFC 0006A LBP 0006A NCNT 0006A NCNTP 0006A NCNT
- 0054D RHB 006E I 0004E X 00070 Y 00071 LSLPT

**BLANK COMMON (0 WORDS)**

**EXTERNAL SUBPROGRAMS REQUIRED:**

- F1101  F1102  F1103  F1104  F1105  F1106  F1108  MIDB 9STP
- 9BCDRD 9BCDREAD 9BCDWRIT 9ENDIOL 9INITIAL 9IBDATA 9PRINT

**HIGHEST ERROR SEVERITY: 0 (08 ERRORS)**

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>146</td>
<td>00092</td>
</tr>
<tr>
<td>0</td>
<td>00000</td>
</tr>
<tr>
<td>114</td>
<td>00072</td>
</tr>
<tr>
<td>0</td>
<td>00000</td>
</tr>
<tr>
<td>260</td>
<td>00104</td>
</tr>
</tbody>
</table>
1. C PROGRAM HIG
2. C FOR INITIAL CONVERSION OF 1.1.G. GRAVITY DATA TO WH01 GSUM
3. C INITIAL VERSION BASED ON PROGRAM DMA
4. C INITIAL VERSION 1 DECEMBER 1973
5. C DIMENSION IA(35), NPA(30)
6. C DIMENSION IZ(9), I*(35)
7. C DATA IS*/ 5*
8. C DATA IHE/ 5
9. C ITAPE=1
10. C ITAPE=2
11. C IIN=106
12. C INPUT lC
13. C OUTPUT lPROGAM HIG VERSION 2 DEC 73
14. C CALL STAT
15. C J=ISK(-2)
16. C KGRA=1.74532/E=2
17. C R=DEGF=57.29572
18. C K=4
19. C E=1
20. C E=2
21. C CAN=2.67=1.03
22. C DENS=1.64
23. C NREC=0
24. C ELEV=0
25. C READ(JIN,1001) ISRC
26. C 1001 FORMAT(15)
27. C OUTPUT lSRC
28. C CALL LINEIT(ITAPE, ITAPE, KK, KGDA, KGMB,
29. C KYR, KG, JDIF, ISORC, RLAT, RLON, ELEV, K7, RBSG,
30. C 1 I=DEP, FA, TG, TC, EL, IC, RFA, IREC, IFCC, IA, IFBC)
31. C IFCC=17
32. C ISG=0
33. C IREC=0
34. C IFBC=0
35. C READ(ITAPE,1002) IINSC, ISIGFA, ISIGBG
36. C 1002 FORMAT(15x,14x,12x,12)
37. C READ(ITAPE,1003)(NAME(1),1,1,60)
38. C 1003 FORMAT(15)
39. C IF ELEV > ELEV KEY=1
40. C CNTINUE
41. C READ(ITAPE,1004 END=999) ILB, ITR, IRS, ISTA, IATD, DLAT, NRS,
42. C ILANG, DLONG, NEBR, IELV, IGV, IOR, IELEV, KEY, IYH, IAF,
43. C IELEV, IELEV T, IFA
44. C 1004 FORMAT(13x,A1,14x,1x,12,F4.2, A1I3,F4.2, A1I7I3,15x,13x,16)
45. C IF ELEV > ELEV CNTINUE
46. C C CHECKING APPARATUS CODE
47. C NAPF=1APF=1
49. C 510 OUTPUT lAPPARATUS CODE NOT IMPLEMENTED
50. C CNTINUE
51. C ELEV=FLOP(ELEV)*1
52. C IF (IELEV>KEY+0.0 AND IELEV+NE+4) ELEV>DELEV GR TO 560
53. C DELEV=ELEV
DEP=FLAT(IDEP)
C CONTINUE
NL=FLAT/60
NL=FLAT(ILATO)
FLAT=FLAT(ILATM)
IF(NGRS*EG*IS) FLAT=FLAT
DLONGM=DLONGM/60
DLONGC=FLOAT(ILONGD)
DLONGC=DLONGC+DLONGM
IF(NE8R*EG*IS) DLONG=DLONG
FA=FLAT(IFIA)*1
BC*FA+(C4165+DENS*FLAT)
ZG1=FLAT(NG1)*1000
ZG2=FLAT(NG2)*1001
K977=NG1
VBSC=ZG2
KGYR=ISTA
ENCDE(35,1005,1Z) HSGC,ISIGFA,ISIBG,IL6C,ITH,ISER,IAIP,IELEV,1
IELEV
1005 FORMAT(14E12,13,2A2,311)
CALL UNFKEY(14,10,35)
D3=420
IA=ISL*(0)*24
CALL UNFKEY(14,10,35)
RREC=RREC+1
GO TO 50
C CONTINUE
CALL GINAT(ITAPE,JTAPE,K0,K0DA,K0HM,KGYR,K0HM,ISIF,IS6RC,
TC,IELC,IGCRFA,IREGC,IFHC,IA,IFHC)
C END OF FILE
GO TO 50
C CONTINUE
ENDFILE JTAPE
BLTFLT 999
BLTFLT 'ALL DBNE'
STF
END
PROGRAM LSORT

PROGRAM TO SORT AND EDIT LUNL O/P
VERSION 29 APRIL 75 TO CHANGE TEST IN LINE 21
VERSION OF 23 MAY 74 TO INCREASE RESOLUTION OF Z
VERSION OF 9 MAR 74 TO MAKE BETTER E8F CHECK
VERSION 1 MAR 74 TO STOP IF ISLCNLT GT 20

ORIGINAL VERSION 21 FEB 74

ITAPE=0
JTAPE=6
KTAPE=13

11. IN=105
12. OUTPUT 'PROGRAM LSORT VERSION 6F 29 APRIL 75'
13. READ(IN,1004) ZLIM
14. OUTPUT ZLIM
15. READ(IN,1001) NUMBD
16. CH 50C I=1, NUMBD
17. ITAPE=ITAPE+1
18. JTATE=JTATE+1
19. ISLCNT=0
20. ZST=990
21. READ(ITAPE,1002)END=400 NCNT,RHO,Z
22. IF(ZLIM EQ ZLIM) AND ISLCNT GT 0 AND RHO*EQ C CO) GO TO 400
23. ZCH=ABS(Z+7ST)
24. IF(ZCH+LZLIM) 0 TO 600
25. ZST=Z
26. ISLCNT=ISLCNT+1
27. IF(ISLCNT GT 20) OUTPUT 'THE MANY LAMINATE ISLCNT, I: GO TO 999
28. WRITE(ITAPE,1002) NCNT,RHO,Z
29. READ(ITAPE,1003) X,Y,LSLFT
30. WRITE(ITAPE,1003) X,Y,LSLFT
31. IF(LSLFT*E*1) GO TO 80
32. GO TO 73
33. WRITE(ITAPE,1001) ISLCNT
34. 500 CONTINUE
35. GO TO 999
36. 600 CONTINUE
37. READ(ITAPE,1003) X,Y,LSLFT
38. IF(LSLFT*EQ 1) GO TO 7C
39. GO TO 600
40. 995 STOP
41. C
42. C FORMATS
43. C ***
44. 1001 FORMAT (12)
45. 1002 FORMAT (12F10.4,F16.6,F6.3)
46. 1003 FORMAT (F12.5,
47. 1004 FORMAT (F10.0)
48. END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>R</td>
<td>SPRG</td>
<td>0001</td>
<td>1</td>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>0002</td>
<td>1</td>
</tr>
<tr>
<td>ISLCNT</td>
<td>I</td>
<td>SCALR</td>
<td>0003</td>
<td>1</td>
<td>ITAPE</td>
<td>I</td>
<td>SCALR</td>
<td>0004</td>
<td>1</td>
</tr>
<tr>
<td>KTAPE</td>
<td>I</td>
<td>SCALR</td>
<td>0005</td>
<td>1</td>
<td>LSFT</td>
<td>I</td>
<td>SCALR</td>
<td>0006</td>
<td>1</td>
</tr>
<tr>
<td>NUMBBD</td>
<td>I</td>
<td>SCALR</td>
<td>0007</td>
<td>1</td>
<td>RHO</td>
<td>R</td>
<td>SCALR</td>
<td>0008</td>
<td>1</td>
</tr>
<tr>
<td>ZLIM</td>
<td>R</td>
<td>SCALR</td>
<td>0009</td>
<td>1</td>
<td>ZST</td>
<td>R</td>
<td>SCALR</td>
<td>000A</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>0034</td>
<td>0000</td>
<td>80</td>
<td>0063</td>
<td>0000</td>
<td>400</td>
<td>0067B</td>
<td>000Z</td>
<td>500</td>
<td>0080</td>
<td>000A</td>
</tr>
<tr>
<td>1001</td>
<td>0093</td>
<td>0000</td>
<td>1002</td>
<td>0095</td>
<td>0000</td>
<td>1003</td>
<td>006C</td>
<td>0000</td>
<td>1004</td>
<td>00A0</td>
<td>0000</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (16 WORDS):**

000000 ITAPE 000001 JTAPE 000002 KTAPE 000003 IIN 000004 ZLIM 000005 NUMBBD
000006 I 000007 ISLCNT 000008 ZST 000009 NCNT 00000A RHO 00000B Z

**BLANK COMMON (0 WORDS)**

**INTRINSIC SUBPROGRAMS USED:**

ABS

**EXTERNAL SUBPROGRAMS REQUIRED:**


**HIGHEST ERROR SEVERITY: 0 (NO ERRORS)**

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>105</td>
</tr>
</tbody>
</table>

**GENERATED CODE:**

105 00005
100 00001
16 00010
5 00000

**TOTAL PROGRAM:** 112 00076
PROGRAM MODPLOT

PLOTS DATA FOR PREPARATION OF, AND CONSTRUCTED,
STRUCTURE MODELS OF EARTH'S CRUST
MAKES PLOTS OF OUTPUT FROM TALPLOT 15, PROJ4, AND/OR SAINT2

VERSION 7 APRIL 75 TO ZERO VARIABLES
VERSION 26 FEB 1975, TO ADD USE OF PINOT AND YINOT
VERSION OF 3 FEB 1975, SOGINOT WILL NOT TRY TO READ ESTP SERIAL NO. CARDS
VERSION OF 11 DEC 1974 TO ADD HANDLING OF PROJ4 OUTPUT TO 9T TAPE
VERSION OF 26 SEPT 1974 TO CHANGE AND IMPROVE DOCUMENTATION
VERSION OF 4 FEB 1973, TO CHANGE GSUM READ TO DECIMAL DEGREES
VERSION OF 12 OCT 1972, TO CORRECT ERROR IN PLOTTING
POLYGON MODEL FROM CARDS WITH JFMT = 5
VERSION OF 1 FEB 1972, TO INCLUDE SOURCE NAME AND DOCUMENTATION
VERSION OF 1 SEPT 1971, TO INCORPORATE MODIFICATIONS FROM TALPLOT OUTPUT
VERSION OF 1 APR 1971, TO PLOT 2-D BOUGUER ANOMALY FROM TALPLOT OUTPUT

DIMENSION BUF(1000), LABEL(20)
DIMENSION CXL(100), CYL(100), BGA(200), FX(200), SSELZ(200)
DIMENSION DCT(200)
DIMENSION KSN(80), FZ(200)
DIMENSION IDESC(6), VEL(8), THICK(8)
DIMENSION BRG20(200)
DIMENSION JA(10), JB(30)
DIMENSION IA(35)
DATA NSH, NSH, '11', 'W', 1/

WHEN USING OUTPUT FROM PROJ4 ON MAG TAPE,
JFMT NUMBER CARDS MUST STILL BE INPUT ON CARDS.
IF JFMT = 1 AND DATA IS ON MAG TAPE, THEN ESTP CARDS HAVE TO BE
INCLUDED FOR SUBROUTINE MOUNT, FOLLOWING THE JFMT=1 CARD.

SSW(1)=0 TO PLOT ONLY FROM TALPLOT OUTPUT TAPE
*1 TO PLOT PROJ4 DATA, INCLUDING MODEL POLYGONS
*2 TO PLOT BOTH TALPLOT OUTPUT TAPE AND PROJ4 DATA
SSW(2)=1 TO PLOT BOUGUER ANOMALY IN ADDITION TO THE FREE-AIR
SSW(3)=1 TO PLOT HEIGHT FROM GSUM DATA
SSW(4)=1 TO PLOT ELEVATION, (INPUT VALUES ARE IN METERS)
SSW(5)=1 TO PLOT OBSERVED * CALC. GRAVITY
SSW(6)=1 TO PLOT WEIGHT
SSW(7)=1 TO PLOT CONTRIBUTION OF EACH POLYGON
SSW(8)=1 TO PLOT 2-D BOUGUER ANOMALY IN TALPLOT OUTPUT
SSW(11)=1 TO PRINT INTERMEDIATE VALUES
SSW(14)=1 TO PLOT ONLY A DOT FOR GSUM FREE-AIR VALUES, RATHER
THAN A CONTINUOUS LINE
SSW(30)=1 TO READ GSUM DATA ON 2 CARDS
SSW(32)=1 TO READ SEISMICITY DATA FROM CARDS

USES INCEP, EXTD, ISH, SPLYT, SPOT, GINOT, DISAZ

INITIALIZATION SECTION

CALL STAT
RFACT=1
I=18W(-2)
C SETTING SSW(31)=1 SO GINOT WILL NOT TRY TO READ ESTP SERIAL NO. CARDS
I=ICHG(31,1),
CALL PLOTS(IBUF=1000)
OUTPUT MODPLOT VERSION OF 4 NOV 1975

1  IN=105
2  OUT=108
3  ITAPE=1
4  JTAPE=2
5  IGSUM=0
6  L=99
7  KFXN=1
8  ILOEP=0
9  DEGR1=745329=02
10  RDEG=57.29578
11  I=0
12  JFMT=0
13  IYIN=0
14  IPIN=0
15  D8 100 1=1200
16  G8A(I)=0.0
17  Fx(I)=0.0
18  Fz(I)=0.0
19  SSELZ(I)=0.0
20  DMST(I)=0.0
21  BG2D(I)=0.0
22  CONTINUE
23  AI=0.0
24  ASG=0
25  ANGB=0.0
26  D6 101 1=100
27  CXL(I)=0.0
28  CYL(I)=0.0
29  CONTINUE
30  DISTKM=0.0
31  D8 102 1=1.8
32  VEL(I)=0.0
33  THICK(I)=0.0
34  CONTINUE
35  J1=0
36  J2=0
37  J3=0
38  J4=0
39  J5=0
40  J6=0
41  J7=0
42  J8=0
43  K1=0
44  K2=0
45  K3=0
46  K4=0
47  K5=0
48  K6=0
49  K7=0
50  K8=0
51  XFact = NUMBER KM'S/INCH IN X DIRECTION (LONG AXIS OF PLOT)
52  YFact = NUMBER OF KM'S IN Y DIRECTION
53  XWide = KM WIDTH OF PLOT IN X DIRECTION
54  YWide = KM WIDTH OF PLOT IN Y DIRECTION
55  Top, Bot, Left, Right = KM VALUE FOR THOSE
ABOVE SEA LEVEL = NEGATIVE
FOR OTHER CURVES THAN MODEL, DEPTH = NEGATIVE, ELEV = POSITIVE

BOUNDARIES OF MGT PLOT
ELFAC, GFA, WFA, PFA = ARE THE SCALE FACTORS FOR ELEVATION

(KMS/IN), GRAVITY(MGAL/IN), WEIGHT(KG/IN) AND
INDIVIDUAL POLYGON CONTRIBUTION (MG/IN) = 1
ELDIS, GDIS, PDIS = ARE THE DISTANCE IN INCHES OF THE
ORIGINS OF THE CURVES ABOVE THE ORIGINS OF THE MODEL
WDIS = DISTANCE OF WEIGHT CURVE BELOW BOTTOM OF MODEL
ORIGIN OF MODEL IN Y DIRECTION = DBOT + WDIS + (BOT/YFA)

READ(IN,11) EFACT, YFACT, TOP, BOT, BLEFT, RIGT
OUTPUT XFAXT, YFACT, TOP, BOT, BLEFT, RIGT
TOP = TOP
BOT = BOT
ICHT = 0
ISTR = 0
READ(IN,11) ELFAC, ELDIS, GFA, GFA, WFA, WFA, PFA, PFA
READ(IN,11) HT, DBOT
C HT = CHARACTER HEIGHT MULTIPLICATION FACTOR (USED IN THE CALL TO
SYMBOL FOR THE PLOTTING OF THE ANOMALY CURVES
C IF HT IS EQUAL TO ZERO A DEFAULT VALUE OF 3 IS ASSUMED
C DBOT IS THE DISTANCE THAT THE WEIGHT CURVE IS SUPPOSED TO BE
PLOTTED ABOVE THE BOTTOM OF THE PLOT (RIGHT SIDE OF PLOTTER)
OUTPUT ELFAC, ELODIS, GFA, GDIS, WFA, WDIS, PFA, PDIS, HT
1 DBOT
IF (HT.EQ.0) HT = 3
HT = HT + 0.5
XWIDE = RIGT - BLEFT
YWIDE = TOP - BOT
SL = (XWIDE/XFACT) + 0.5
SW = (YWIDE/YFACT) + 0.5
CALL WHERE( XORG, XORG, YFACT)
CALL PLOT( XORG, YORG, RFACT)
CALL SYMBOL(0.0,0.0,28,2,0,0)
WCHAR (0.0,0.0,28,2,0,0,2)
5018 INPT = 105
5019 GO TO 5021
5021 READ(INPT, 5022) LABEL
5022 FORMAT(20A4)
5023 FORMAT(1X, 20A4)
5024 CALL SYMBOL(0.0,0.0,28, LABEL, 90, 80)
C VIT = DISTANCE OF MODEL ORIGIN ABOVE BOTTOM (RIGHT) OF PAPER
VIT = DBOT + WDIS + YWIDE/XFACT
IF (VIT.GE.29.1) OUTPUT (PLOT 766 WIDE, WIDTH = VIT, OUTPUT VIT)
STOP
IF (BLEFT) 5025, 5026, 5024
XRT = 3 + 0
5026 CALL PLOT( XRT, VIT = 3)
XX = 0 + 0
YY = 0 + 0
CALL SYMBOL( O.0, O.0, O.0, 14, 9, O.0, 0.1)
C CALCULATING COORDINATES OF CENTRAL POINT
RX = (XWIDE*0.5) + (-1*O.5*BLEFT)
CONVERTING ORIGIN TRANSLATION VECTORS TO INCHES

RY = -(YWIDE*0.5) + (TOP))

RX = RX/XFACT
RY = RY/YFACT

OUTPUT INTERMEDIATE VALUES
WDIS = -YWIDE/YFACT-WDIS

IF(ISW(13)) 402, 698, 408
402 WRITE((16UT,404)XWIDE,YWIDE,SL,SW,RX,RY,XFACT,YFACT)
404 FORMAT((IV=5/2E12.5/2E12.5/2E12.5)

PLOT ORIGIN IS AT 0,0 OF STRUCTURE MODEL

CONTINUE

INITIALIZATION IS COMPLETE

C READ CSS VALUES (FIRST DATA CARD USED IN PROVA)

READ(IN,699)JA,ANG,DMAXM,ILAT,RILTM,ILONG,RILGM,JB

FORMAT(10A12,2F10.0,4F6.4,2F6.2,30A1)
WRITE((16UT,699)JA,ANG,DMAXM,ILAT,RILTM,ILONG,RILGM,JB

RIL=CM2TR(ILAT,RILTM)
RILG=CM2TR(ILONG,RILGM)

IF(ISW(1))=115,115,405

CONTINUE

READ(ITAPE,501)KSW

FORMAT(B011)
READ(ITAPE,120)RDENS,RHGT,RHDO,REFX,FXI,DELFX,M,IMAX

FORMAT(6F10.3)
READ(ITAPE,135)GSA(I),I=1,M

FORMAT(F510.1)
IF(KSW(2)=552,553,552
CONTINUE

READ(ITAPE,135)(FZ(I),I=1,M)

CONTINUE

NOW START READING POLYGONS

READ(ITAPE,150)LN8,RHORK

FORMAT(I5,F10.3)
I = 1

CONTINUE

READ(ITAPE,168)(XX,YY,ICODE

FORMAT(2F10.2,11)

MAKING DEPTHS NEGATIVE

YY = -YY

CONVERTING COORDINATES TO INCHES

XX = XX/XFACT
YY = YY/YFACT

SHIFTING COORDINATES TO CENTER OF PLOT

CXL(I) = XX-RX
CYL(I) = -(YY*R-YY)
CX = CXL(I)
CY = CYL(I)
IF(I=1)190,190,200

CONTINUE

CXP = CXL(I=1)
CALL INCER(SL, SW, CX, CY, CXP, CYP, BX, BY, A1, A2, B1, B2)

IF(ISW(13))406, 205, 406

WRITE(180T, 408) SL, SW, CX, CY, CXP, CYP, BX, BY, A1, A1, B1, B2

FORMAT(1V, 1X, 4E10.5, 6E10.5)

IF(A1=9990.0)210, 248, 248

C PLOT MODEL SEGMENT

210 XX = A1*RX

YY = B1*RY

IPEN = 3

CALL PLOT(XX, YY, IPEN)

XX = A2*RX

YY = B2*RY

IPEN = 2

CALL PLOT(XX, YY, IPEN)

248 IF(ICODE=9)250, 260, 250

250 I=1

GO TO 162

256 IF((JFMT*EQ*5) .OR. (JFMT*EQ*6)) GO TO 272

258 NCDE = 1

260 IF(LN6=1)1260, 1190, 1260

265 C CHECKING IF POLYGON NUMBER = 1

266 IF(ISW(8))1200, 1200, 1200

1200 DO 1220 XI=KFXN,M

READ(ITAPE, 1206) K, FX(K), SSELZ(K), BGI0D(K)

1206 FORMAT(I5, F10.3, 2, 10X, F10.2)

C THIS SECTION PLOTS INTERMEDIATE DATA FOR EACH POLYGON

1207 IF(ISW(7))1210, 1210, 1210

1210 XX=FX(K)/XFAC

YY=SSELZ(K)/PFAC*PDIS

YP=YY*VIT

1220 IF((YP*GE*29) .OR. (YP*LE*0)) NCDE=11 GO TO 1220

1225 CALL SYMBOL(XX, YY, HT, ICHAR, Q=0, NCDE)

1230 CONTINUE

1234 NCDE = 2

1238 CONTINUE

GO TO 271

1260 IF(ISW(1))1270, 1270, 1270

1270 IF((JFMT*EQ*5) .OR. (JFMT*EQ*6)) GO TO 271

1275 IF((IFMT*EQ*5) .OR. (IFMT*EQ*6)) GO TO 271

1280 C NEXT TWO IF STATEMENTS ASSUME PLOTTING OF MODEL POLYGONS

1284 IS ONLY BEING DONE FROM TALSPLT OUTPUT ON MAG TAPE

1287 C ILOOP IS .EQ. 1 ONLY WHEN WE ARE PLOTTING THE MODIFIED POLYGON

1289 IF(ILoop*EQ*1) GO TO 302

1293 IF(KSW(9)*EQ*1) GO TO 271

1297 DO 270 III=KFXN,M

READ(ITAPE, 263) K, FX(K), SSELZ(K)

263 FORMAT(I5, F10.2, 10X, F10.2)

268 C THIS SECTION PLOTS INTERMEDIATE DATA FOR EACH

C POLYGON

290 IF(ISW(7))554, 555, 554

294 XX=FX(K)/XFAC

YY=SSELZ(K)/PFAC*PDIS

YP=YY*VIT

299 IF((YP*GE*29) .OR. (YP*LE*0)) NCDE=11 GO TO 270

2995 CALL SYMBOL(XX, YY, HT, ICHAR, Q=0, NCDE)

2999 NCDE = 2

270 CONTINUE
238

300 271 CONTINUE
301 ICHAR=ICHAR+1
302 272 IF(LNB=L) 145,290,145
303 290 CONTINUE
304 291 IF(ISW(1))291,291,700
305 291 IF(KSW(6)+EG=0) G6 TO 300
306 300 G6 TO 160
307 300 IF((JFMT+EG=5) * OR (JFMT+EG=6)) G6 TO 330
308 302 DB 310 K*KFXN, M
309 302 READ(1TARE, 304) JFX(K), SSELZ(K), DWGT(K)
310 304 FORMAT(15, F10.2, 20X, F10.2, 36X, F16.0)
311 310 CONTINUE
312 310 C (C) PLOT ELEV, GGA, WEIGHT, SSELZ, AS REQUIRED BY SENSE
313 310 C (C) SENSE SWITCHES
314 315 NCDE = -1
316 315 IF(ISH(4))556,557,556
317 556 DB 590 K*KFXN, M
318 556 XX*KFX(K)/FACT
319 556 YY*FZ(K)/FACT+ELFA+ELD
320 556 YP=YY*VIT
321 556 IF(YPS+GE29, OR, YPS+LE0) NCDE=11 G6 TO 590
322 556 CALL SYMBOL(XX, YY, HT, O, C, O, NCDE)
323 556 NCDE=-2
324 556 CONTINUE
325 556 CONTINUE
326 557 CONTINUE
327 557 C (C) PLOT BBS, ERVEP + CALCULATED G
328 558 NCDE = -1
329 558 DB 599 K*KFXN, M
330 558 XX*KFX(K)/FACT
331 558 YY*GFA+DIS
332 558 YP=YY*VIT
333 558 IF(YPS+GE29, OR, YPS+LE0) NCDE=11 G6 TO 599
334 558 CALL SYMBOL(XX, YY, HT, O, C, O, NCDE)
335 558 NCDE=-2
336 559 CONTINUE
337 559 NCDE = -1
338 560 IPEN=3
339 560 DB 592 K*KFXN, M
340 560 XX*KFX(K)/FACT
341 560 YY*GFA+DIS
342 560 YP=YY*VIT
343 560 IF(YPS+GE29, OR, YPS+LE0) NCDE=11 IPEN=3 G6 TO 592
344 560 CALL SYMBOL(XX, YY, HT, O, C, O, NCDE)
345 560 CALL PLD(T, XX, YY, IPEN)
346 560 NCDE=-2
347 560 IPEN=2
348 560 CONTINUE
349 560 CONTINUE
350 560 C (C) CHECK IF PLOT 2-D BOUGUER ANOMALY
351 559 CONTINUE
352 559 CONTINUE
353 559 IF(ISH(8)) 588,588,57C
354 570 IPEN=3
355 570 DB 588 K*KFXN, M
356 570 XX*KFX(K)/FACT
357 570 YY*GFA+DIS
358 570 CALL SYMBOL(XX, YY, HT, O, C, O, NCDE)
360 IF (YPGE.29 .OR. YPG LE.0) IPEN=3 G0 TO 585
361 CALL PLOT (XX,YY,IPEN)
362 IPEN=2
363 585 CONTINUE
364 588 NCDE = 1
365 561 DO 93 K=KFXNM
366 588 XXFX(K)XFAC
367 568 YY=DWGT(K)/KFACT+WDIS
368 569 IF (YPGE.29 .OR. YPG LE.0) NCDE=1 G0 TO 593
369 588 CONTINUE
370 593 CONTINUE
371 562 CONTINUE
372 330 WRITE (110UT,332)
373 332 FORMAT ('LAST POLYGON OF MODEL PLOTTED')
374 333 CALL PLOT (XX,YY,999)
375 334 CALL EXIT
376 70C ITAPE = 105
377 C SETTING UP CONSTANTS FOR PLOTTING SPfmt COLUMNS
378 C BDIST = LENGTH OF TICK LINE IN INCHES
379 C ANGB = ANGLE IN DEGREES OF TICK LINE FROM HORIZONTAL
380 C UPWARD ANGLE = + ANGLE
381 C DOWNWARD ANGLE = - ANGLE
382 C YFAC = KM PER INCH FOR PLOTTING COLUMN
383 C SEP = SEPARATION DISTANCE IN INCHES BETWEEN COLUMNS
384 C DEGRA = 1/7453296=2
385 C RADEG = 57.29578
386 C RANGB = ANGB = DEGRA
387 C AC = COS(RANGB)
388 C AS = SIN(RANGB)
389 C AS=1.0*AS
390 C KDA=0
391 C KMB=0
392 C KYS=0
393 C BDIST=0.25
394 C ANGB=0.0
395 C YFAC=YFACT
396 C KGDA=0
397 C KGMB=0
398 C KGYS=0
399 C ZHT=1.0
400 C HGT=0.07
401 C ********************************************
402 C JFMT = 1 FOR GSUM FORMAT
403 C JFMT = 2 FOR GSUM FORMAT
404 C JFMT = 3 FOR SPfmt DATA
405 C JFMT = 4 FOR SEISMICITY DATA
406 C JFMT = 5 FOR MODEL POLYGONS
407 C JFMT = 6 FOR TALPLB T INPUT
408 C JFMT = 9 TO TERMINATE JOB
409 C ********************************************
410 701 READ (JIN,702) JFMT
411 702 FORMAT (1)
412 703 OUTPUT JFMT
413 704 G0 TO (71C,710,750,800,145,115,333,333,996) JFMT
420. C PLOTTING GSM DATA
421. 710 CONTINUE
422. IPEN=3
423. 712 CONTINUE
424. IF(ISW(30).EQ.0)ITAPE=1 GO TO 713
425. C INITIALIZE GINOT IF THIS IS THE FIRST READ FOR GINOT
426. C 713 IF(ISUM.EQ.1)GO TO 714
427. 9K=0
428. CALL GINOT(ITAPE,UTAPE,KK,KGDA,KGM0,
429. 1 KGYR,KGHM,1DIF,ISSRC,RLAT,RLONG,ELEV,K977,6BSG,
430. 2 IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
431. ISUM=1
432. 714 KK=1
433. CALL GINOT(ITAPE,UTAPE,KK,KGDA,KGM0,
434. 1 KGYR,KGHM,1DIF,ISSRC,RLAT,RLONG,ELEV,K977,6BSG,
435. 2 IDEP,FA,BG,TC,IELC,IGC,RFA,IREGC,IFFC,IA,IFBC)
436. IF(KK.EQ.8)GO TO 700
437. IF(KK.EQ.9)GO TO 999
438. KGD0=KGDA
439. KGMO=KGM0
440. KGYR=KGYR
441. KGHM=KGMH
442. 73 DEPTH=IDEP
443. IF(IDEP).GT.78,74,78
444. 74 HEIGT=ELEV
445. GO TO 64
446. 78 HEIGT = DEPTH
447. 64 CONTINUE
448. C CONVERSION HEIGT FROM METERS TO KM
449. HEIGT=HEIGT*0.001
450. 720 YY=DIS*(FA/GFAC)
451. XX=DISTKM/FACT
452. YP=YY+VIT
453. IF(YP.GE.29. OR. YP.LE.0) IPEN=3 GO TO 783
454. IF(ISW(4)).GT.721,721,722
455. PLOTTING A SMALL CIRCLE FOR FREE-AIR
456. 721 CALL PLOT(XX,YY,IPEN)
457. IPEN=2
458. GO TO 723
459. C PLOTTING ONLY A DOT FOR FREE-AIR
460. 723 CALL PLOT(XX,YY,3)
461. CALL PLOT(XX,YY,2)
462. CALL PLOT(XX,YY,3)
463. GO TO 723
464. C CHECKING IF ALSO PLOT BOUGUER ANOMALY
465. 725 XT=XX
466. YT=DIS*(BG/GFAC)
467. YP=YT+VIT
468. IF(YP.GE.29. OR. YP.LE.0) GO TO 728
469. CALL PLOT(XT,YT,3)
470. CALL SPOT(XT,YT)
471. CALL PLOT(XX,YY,3)
472. C CHECKING IF ALSO PLOT HEIGHT
730 XT=XX
731 Y=HEIGHT/ELFAC
732 YP=YT+YIT
733 IF(YP,GE,29. OR. YP,LE,0) GO TO 712
734 CALL PLOT(XT,YT3)
735 CALL SPOT(XT,YT)
736 CALL PLOT(XX,YF3)
737 GO TO 712
738 C PLOTTING SPFFT DATA
739 750 IF(ISN(32),EQ,0) ITAPE = 1
740 C READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
741 C INITIALIZING PINOT, IF THIS IS FIRST READ
742 IF (IPIN,NE,1) GO TO 782
743 KK = 0
744 CALL PINOT (ITAPE, ITAPE, KK, IST, KEY, LAT, LATH, KNS,
745 1 LON, LOM, KEN, VEL, THICK, IMANT, NELEV, N1, N2, N3, N4, MET, IYR, IDESC,
746 2 DINE, STH, CRVN, WGTN, AVWN, CRVH, AVGH, AVHM)
747 IPIN = 1
748 752 KK = 1
749 CALL PINOT (ITAPE, ITAPE, KK, IST, KEY, LAT, LATH, KNS,
750 1 LON, LOM, KEN, VEL, THICK, IMANT, NELEV, N1, N2, N3, N4, MET, IYR, IDESC,
751 2 DINE, STH, CRVN, WGTN, AVWN, CRVH, AVGH, AVHM)
752 IF (KK,NE,8) GO TO 700
753 IF (KK,NE,9) GO TO 999
754 20 CONTINUE
755 VEL (1) = (FLOAT(J1))*0.1
756 VEL (2) = (FLOAT(J2))*0.1
757 VEL (3) = (FLOAT(J3))*0.1
758 VEL (4) = (FLOAT(J4))*0.1
759 VEL (5) = (FLOAT(J5))*0.1
760 VEL (6) = (FLOAT(J6))*0.1
761 VEL (7) = (FLOAT(J7))*0.1
762 VEL (8) = (FLOAT(J8))*0.1
763 THICK (1) = (FLOAT(K1))*0.1
764 THICK (2) = (FLOAT(K2))*0.1
765 THICK (3) = (FLOAT(K3))*0.1
766 THICK (4) = (FLOAT(K4))*0.1
767 THICK (5) = (FLOAT(K5))*0.1
768 THICK (6) = (FLOAT(K6))*0.1
769 THICK (7) = (FLOAT(K7))*0.1
770 THICK (8) = (FLOAT(K8))*0.1
771 VMANT = (FLOAT(IMANT))*0.1
772 ELEV = NELEV
773 ELEV = ELEV*0.01
774 50 IF(IN.EQ,2) 70,60,70
775 C SEA SEISMIC PROFILE
776 60 VELWS = 1.5
777 LATM = ELEV
778 GO TO 80
779 C LAND SEISMIC PROFILE
780 70 VELWS = 0.0
781 LATM = 0.0
782 C MAIN PLOTTING LOOP
783 80 RLATM = LATM
784 RLM = LRM
785 RLAT = DMTR(LATM,RLATM)
786 RLong = DMTR(LONG,RLM)
787 IF(KNS,NE,KNS) 54, 52, 54
540* 52 RLAT = +RLAT
541* 54 IF (RKE+NEW) 58, 56, 58
542* 56 RLNG = +RLNG
543* 58 CONTINUE
544* C DETERMINE DISTANCE FROM ORIGIN
545* GO TO 40
546* 451 XX*DISTKM*FACT
547* YY = 0.0
548* CALL PLOT(XX,YY,3)
549* CALL SPLT(IST,YLAT,YLON,THICK,VEL,HATT,VHAN,XX
550* 1 YFAC,ZHT,HGT,AC,AS,ANG,BDIST)
551* GO TO 750
552* C PLOTTING SEISMICITY DATA
553* 800 CONTINUE
554* IF(ISH(36)*EQ.0)ITAPE=1
555* C INITIALIZING YINOT, IF THIS IS FIRST READ
556* IF(YIN*EQ.1) GO TO 801
557* KK = 0
558* CALL YINOT(ITAPE,ITAPE,KK,
559* 1 ISR,ISRG,KDA,KMB,KYR,KHE,SEC,CLAT,KSN,DLON,KWE,DEPT,AMAG,IMB,
560* 2 ISR,ISRG,KDA,KMB,KYR,KHE,SEC,CLAT,KSN,DLON,KWE,DEPT,AMAG,IMB,
561* 3 ICE,IAUTH,IGH,NNP,ISG,ILM,IIS,IS2)
562* C
563* 564* 801 KK = 1
565* C
566* CALL YINOT(ITAPE,ITAPE,KK,
567* 1 ISR,ISRG,KDA,KMB,KYR,KHE,SEC,CLAT,KSN,DLON,KWE,DEPT,AMAG,IMB,
568* 2 ISR,ISRG,KDA,KMB,KYR,KHE,SEC,CLAT,KSN,DLON,KWE,DEPT,AMAG,IMB,
569* 3 ICE,IAUTH,IGH,NNP,ISG,ILM,IIS,IS2)
570* IF(KK*EQ.8) GO TO 700
571* IF(KK*EQ.9) GO TO 999
572* C DETERMINE DISTANCE FROM ORIGIN
573* KGDA=KDA
574* KMB=KMB
575* KYR=KYR
576* KGHE=KGHE
577* KGDA=KDA
578* KMB=KMB
579* KYR=KYR
580* KGHE=KGHE
581* KLM=KLM
582* CALL DNAV(DLAT,KSN,DLON,KWE,RLAT,RLONG,KL)
583* GO TO 40
584* 82C XX*DISTKM*FACT
585* YY*DEPT*FACT
586* YPB=YPB
587* IF(YPB*GE.29. OR. YPB*LE.0) GO TO 800
588* CALL PLOT(XX,YY,3)
589* CALL ANG3(XX,YY,DEPT,AMAG)
590* GO TO 800
591* 40 CONTINUE
592* CALL DISDZ(RLAT,RLONG,RLAT,RLONG,1,AB,DISKMC)
593* IF(ANG) 44, 42
594* 42 IF(A=135) 48, 46
595* 44 IF(A=225) 48, 46
596* 46 DISKMC=DISKMC
597* 48 CONTINUE
598* IF(DISKMC.LT.BLEFT. OR. DISKMC.GT.RIOT) 1088 = 1088 +1
599* GO TO (720,720,451,820) JFMT
243

600*    OUTPUT '0001
601*    GO TO (712,712,752,800)JFMT
602*    WRITE(IOUT,997)
603*    FORMAT('JFMT = 91)
604*    FORMAT(1WO,'END OF PROCESSING',/)
605*    15,'DATA POINTS OUT OF PLOT BOUNDS')
606*    WRITE(IOUT,998) 100B
607*    GO TO 333
608*    END
**WIGHEST ERROR SEVERITY**: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE</td>
<td>1896</td>
<td>00768</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>42</td>
<td>0002A</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>2805</td>
<td>00AF5</td>
</tr>
<tr>
<td>TEMPS</td>
<td>1</td>
<td>00001</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>4744</td>
<td>01288</td>
</tr>
</tbody>
</table>
PROGRAM NOAA
VERSION OF 8 JUN 74 TO 06 LAMBERT
VERSION OF 0 JAN 1974 TO REMOVE ABORT FOR BUFF IN ERROR
FOR INITIAL CONVERSION OF C, AIG AND USGS 1971 IDOE
INITIAL VERSION 10 DECEMBER 1973

HESKANIAN G METER

DIMENSION IBUFIN(20,50,2), IBUTF(32,50,2)
DIMENSION IA(35), CRUISE(6)
DIMENSION IZ(9), Iw(35)

ITAPE=1
JTAPE=2
IN=105
IBUT=108
IREC=1
KK=0
KI=1
KB=-2
IFLIP=1
KFLIP=1
NFLIP=1
ICNT=0
NREC=0
ELEV=0
ISOUT=0
NIN=50
NSUT=0
READ(IN,1005) ISRC
WRITE(OUT,1002) ISRC
IFFC=16

THETA=30
DEGRA=1+745329E-2
RADERG=57*29378
IENDKEY=0
IELC=5
IREGC=0
IPBC=0
DENS=2+67*1+03
DENS=1+64

INPUT 'PROGRAM NOAA VERS OF 8 JUN 74'

BUFFER LOGIC FOR I/P

CALL BUFF IN(ITAPE,0,IBUFIN(I,1,1,IFLIP),1000)
10 CONTINUE
IF (NIN<LT*50) GO TO 90
15 CONTINUE
CALL ICHECK(ITAPE,IKEY,NI)
GO TO (20/N50/30440) IKEY
20 OUTPUT 'WAITING FOR I/P', IEDD=0
GO TO 15
30 OUTPUT 'END OF FILE ON ITAPE', IEDD=1
GO TO 50
40 OUTPUT 'BUFF IN ERROR'
50 CONTINUE
NIN=0
NFLIP=IFLIP
IFLIP=3-IFLIP
IF(IE8D*NE*1) CALL BUFF IN(TAPE,0,IBUF,11,IFLIP),1000

C INPUT LOGIC

90 CONTINUE

C NIN=NIN+1

IF (IE8D*NE*1) G0 TO 95

C GOING TO EOF PROCESSING

IF(IENDKEY*EQ*1) G0 TO 999

C NINCH=NIN+20

IF(NINCH*EQ*NI) IENDKEY=1

CONTINUE

DECIDE(80,1003,IBUF11,NIN,NFLIP),ND)

1 (CRUISE(I),I+1,B),ITIMEZ,IYR,IMG,IDA,IHR,IMIN,

2 DLAT,DLONG,NAVPT,NAVYP,ICURAZ,ICURAZ,ITMCFATH,ICRRM,

3 IMATHZ8,IMAG,IREMAG,IFA

C EDIT LOGIC

75 IF (NAVPT*NE*0) G0 TO 10

74 IF (IFAS*EQ*0) G0 TO 10

73 IF (ITIMEZ*NE*0) CALL CHGMT(IDA,IMG,ITYR,KHM,KTZ,KGDA,KGMS,KGYR,

72 1 KGHA/NTZ);KTZ=9;G0 TO 100

82 KDGA=IDA

81 KGMS=IMG

80 KGHA=ITYR

81 CONTINUE

80 ICURVEL=INT(CURVEL*100)

83 FLAT=DLAT*DEGRA

82 RLST=DLONG*DEGRA

81 THE8*GINT(PLAT)

80 DIA=FLAT(IFA)*1

79 FA*DFA

78 GABS=THB*DFA

77 CALL BBG(K77,BB9G,BB9S,K8)

76 IDP=ICRRM

75 DMIN=FLAT(IMIN)

74 DMINT=DMIN+1

73 JMIN=INT(DMIN)

72 DMINT=INT(DMIN)+DJMIN

71 KH=MJHN*+IHR+100

70 KGHA=KH

72 DZ=FLAT(ITIMEZ)

71 KTZ=INT(DTZ*1)

70 DEP=FLAT(IDEP)

69 BFA=O*O185*DEN=DEP

68 IF(IDEP*EQ*0) BG=999*0

67 PLAT=DLAT+90,FLAT=PLAT

66 PLTGG=DLONG+180,ILHKEY=PLAT

65 IAKY=0

64 ENCRD(35,1000*1Z),(CRUSS(I),I+1,B),KTZ,MINT,NAVPT,ICURAZ,

63 ICURVEL,ICURAZ,IMATHZ8,IMAG,IREMSAG

62 CALL UNPKBY1Z,1W,35)

61 DB 120 I=1,35

60 IA(I)*ILS(1W(I)*24)

120 CONTINUE

C OUTPUT LOGIC
300 CONTINUE
NBUT=NBUT+1
ENCOD(128,1001,IBUTF8T(1,NBUT,JFLIP),ND)IREC,ISREC,<GRA,GR
1.KGYR,KISH4,OLAT,DLONG,ELEV,K77,8BSG,IDEP,FA,BG,TC,IELC,IGC,
2.RPA,IREG,IFLC,IA,IFRC,LTKEY,LGKEY,IAKEY
305 CONTINUE
IF(NBUT.LT.50)GO TO 10
310 CONTINUE
IF(NBUT.WE.1)NBUT=1;GO TO 350
JKEY=CHECK(JTAPE)
GO TO (320,350,330,340) JKEY
320 OUTPUT 'WAITING FOR O/P ';IEBD=0
GO TO 310
330 OUTPUT 'END OF FILE JTAPE';IEBD=1
GO TO 999
340 OUTPUT 'BUFF BUT ERROR! ';IEBD=1
GO TO 940
350 CONTINUE
NBUT=0
KFLIP=JFLIP
JFLIP=-JFLIP
CALL BUFF BUT(JTAPE,0,IBUTF8T(1,1,KFLIP),1600)
GO TO 10
399 CONTINUE
910 CONTINUE
IF(JKEY.EQ.9)JKEY=CHECK(JTAPE)
GO TO (920,950,930,940) JKEY
920 OUTPUT 'WAITING FOR O/P ';IEBD=0
GO TO 910
930 OUTPUT 'BAD JKEY! ';IEBD=1
GO TO 960
940 OUTPUT 'BUFF BUT ERROR! ';IEBD=1
GO TO 960
950 CONTINUE
JWDS=NBUT*50
CALL BUFF BUT(JTAPE,0,IBUTF8T(1,1,JFLIP),JWDS)
960 CONTINUE
END FILE JTAPE
OUTPUT NREC
OUTPUT 'ALL DONE'
1001 FORMAT(11,14,312,14,2F9.4,F7.2,13,F6.2,15,2F6.1,F4.1,
12,212,F6.1,11,12,35A,1X,11,213,12)
1002 FORMAT(X,'THIS RUN PROCESSED SOURCE CODE',15)
1003 FORMAT(8A1,15,312,1X,12,13,F8.4,F9.4,211,13,F4.1,1X,215,12,1X,315)
1004 FORMAT(8A1,311,213,15,12,215)
1005 FORMAT(15)
END
BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:
FLOAT  IDINT  ISL

EXTERNAL SUBPROGRAMS REQUIRED:
BUFFIN  BUFFOUT  CHGHT  GINTF  ICHECK  08G  UNPKBY  F1101
F1102  F1103  F1104  F1105  F1106  F1108  M105  M18C
9BCDREAD  9BCDWRITE  9DEC9DE  9ENC9DE  SENDFILE  SEND10L  91INITIAL  918DATA
910LUSA  910R  9PRINT  9R101  9STOP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC  HEX
WORDS  WORDS  -----  -----  

GENERATED CODE:  617  00269
CONSTANTS:  13  000000
LOCAL VARIABLES:  5373  014FD
TEMPS:  1  000001

TOTAL PROGRAM:  6004  01774
PROGRAM PROFG

DIMENSION PLT(15)
DIMENSION IA(35)

VERSION OF 2 OCT 75 TO CALL GINT
PROGRAM PROFG, PLTS PROFILE OF SELECTED VARIABLE
READ FROM GSUM FORMAT

X AXIS PROPORTIONAL TO EITHER DISTANCE OR TIME ALONG TRACK
IF LCNT=9000 OR GREATER THEN TIME ANNOTATION
IS DELETED
HAS DATA AND DISTANCE LIMITS AS INPUT
FOR SELECTION OF SINGLE STATION SSW(5) = 1 AND MSTA READ IN
AS THE STATION NUMBER TO BE SELECTED
USES GINT, FIND, ISW, STAT, EVIL, CDATE, MCVL, SPOT,
SSW(5)=1 TO SELECT SINGLE STATION NUMBER
SSW (14) = UP TO CALL FIND

DIMENSION IBUF(1000)

IN = 105
INOUT = 108
KK=0
JTAPE=2
K=1
INNT=ISW(-2)
CALL GINT(ITAPE,JTAPE, KK)
CALL PLTS(IBUF, *1000)
WRITE (INOUT,600)

600 FORMAT(/1, PROGRAM PROFG VER 2 OCT 75/)

NE8F = 0
XX=0.0
INIT=1

DIFAC = NUMBER OF N MILES, KM, OF HOURS PER INCH ON PLOT
YFAC = ENGINEERING UNITS PER INCH ON PLOT FOR Y DIRECTION
LCNT = TIME ANNOTATION EVERY LCNT POINTS PLOTTED
MIKM = 0 FOR NAUTICAL MILES, 1 FOR KILOMETERS
NFILE = NO. OF INPUT REELS TO PROCESS
FORMAT (2F10.0, 415)
OUTPUT DIFAC, YFAC, LCNT, MIKM, NPLT, NFILE
ULIM = UPPER LIMIT FOR PLOTTING DATA VALUE IN ENG UNITS
BLIM=BOTTOM LIMIT FOR PLOTTING DATA VALUE IN ENG UNITS
DLIM = DISTANCE LIMIT IN INCHES FOR PLOTTING DATA POINT
IXDIN=1 FOR X AXIS PROPORTIONAL TO DISTANCE ALONG TRACK
C  *2 FOR X AXIS PROPORTIONAL TO TIME ALONG TRACK
60  READ (IN,6) ULIM, BLIM, DLIM, IXDIR
61  FORMAT (3F10.0, I5)
62  C XALOW = ALLOWABLE INCHES FOR LENGTH OF PLOT
63  BEFORE REINITIALIZATION
64  C DMOVE = INCHES TO BE SPACED BEFORE REINITIALIZATION
65  C
66  READ (IN,8) XALOW, DMOVE
67  FORMAT (2F10.0)
68  OUTPUT ULIM, BLIM, DLIM, IXDIR, XALOW, DMOVE
69  IF (SW(16) .EQ. 129, 129, 129)
70  128 READ (IN,9) LIMDA, LIMMB, LIMYR, LIMHM
71  9 FORMAT (4I5)
72  OUTPUT LIMDA, LIMMB, LIMYR, LIMHM
73  111 CALL FIND (LIMDA, LIMMB, LIMYR, LIMHM, KGDA, KGMB, KGVR, KGHM, INDK)
74  IF (INDK .EQ. 111) 129, 129
75  129 CONTINUE
76  READ (IN,9) MSTA
77  CALL WHERE (XORG, YORG, RFAC)
78  CALL PLAT (XORG, YORG, DFAC)
79  IF (8990-LCNT=130, 132, 132)
80  130 NCNT=0
81  GO TO 134
82  132 NCNT=132
83  134 IF (MK=14, 13, 14)
84  13 CONV=0.53359
85  GO TO 52
86  14 CONV=1.0
87  50 CONTINUE
88  15 CONTINUE
89  CALL GINOT (ITAPE, JTAPE, KI, KGDA, KGMB, KGVR, KGHM, IDF, IS8RC,
90  * RLA, RLBM, REL, K977, MBSG, IDEP, FA, BG, TC, IELC, IGC, RFA, IREGC,
91  * IFCC, IA, IFGC)
92  52 CONTINUE
93  IF (KI*EQ*9) 44 TO 44
94  IF (IS8(5)*EQ*1) GO TO 70
95  IF (KGHM*EQ*MSTA) GO TO 70
96  GO TO 52
97  44 OUTPUT 'PLOTTING COMPLETED'
98  CALL PLAT (XX, YY, 999)
99  CALL EXIT
100  70 DEPTH=IDEP
101  IF (IDEP=78, 74, 78)
102  74 HEIG=REL
103  GO TO 80
104  78 HEIG = *DEPTH
105  80 A*K977=977
106  AXA=1000.0
107  GHSB*GBSG=A
108  BGCM*BG+TC
109  100 PLT(1)=KGHM
110  PLT(2)=IS8RC
111  PLT(3)=ELEV
112  PLT(4)=DEPTH
113  PLT(5)=HEIG
114  PLT(6)=FA
115  PLT(7)=BG
116  PLT(8)=TC
117  PLT(9)=BGCM
118  PLT(10)=RFA
119  PLT(11)=GBMS
120* KDA=KDA
121* KMB=KGB
122* KRY=KYR
123* KHM=KHM
124* DAY=KDA
125* YMD=KGM8
126* YEAR=KGYR
127* HOUR=KHM
128* XLAT=XLAT
129* YLONG=LONG
130* DATA=PLT(NPLOT)
131* IF(INIT=1)25,30,25,
132* 30 DISTM=0.0
133* TIMD=0.0
134* KDA=KDA
135* KMB=KGB
136* KRY=KYR
137* KHM=KHM
138* INIT=0
139* IPEN=3
140* GO TO 50
141* 25 TLAT=ABS(XLAT)
142* RADI=6371229.0
143* DLTDI=(XLAT*XLAT8)*RADI
144* DLGD1=(XLONG*XLNG8)*RADI*COS(TLAT)
145* DISTM=SQR((ABS(DLTDI))**2+(ABS(DLGD1))**2)
146* CALL COATE(KDA,KMB,KRY,KHM,TIMD)
147* 350 TLAT=XLAT
148* XLNG=XLNG
149* KDA=KDA
150* KMB=KGB
151* KRY=KYR
152* KHM=KHM
153* YY=DATA/YFAC
154* IF(IDIR=1)54,56,54
155* 54 DIS=TIMD/DIFAC
156* GO TO 58
157* 56 DIS=(DISTM=0.001*CMNY)/DIFAC
158* 58 IF(DIS=DLIM)365,365,360
159* 360 XX=XX+3.0
160* WRITE(IIUT,61)KDA,KMB,KRY,KHM
161* 61 FORMAT('DLIM',313,15)
162* CALL PLAT(XX,25)
163* GO TO 25
164* 365 IF(DATA=ULIM)368,366,366
165* 366 WRITE(IIUT,61)KDA,KMB,KRY,KHM
166* 67 FORMAT('ULIM',313,15)
167* GO TO 160
168* 167 WRITE(IIUT,168)KDA,KMB,KRY,KHM
169* 168 FORMAT('BLIM',313,15)
170* 160 XX=XX+DIS
171* CALL PLAT(XX,0.0,3)
172* IPEN=3
173* GO TO 15
174* 368 IF(BLIM=DATA)369,167,167
175* 369 XX=XX+DIS
176* C CHECKING WITHIN ALLOWABLE PLOT DISTANCE
177* 170 IF(XX=XAL8W)180,172,172
178* 172 XX=XX+DMOVE
179* IPEN=3
180.* CALL PLOT(XX,YY,IPEN)
181.* CALL WHERE(XORG,YORG,RFAC)
182.* CALL PLOT(XORG,0.0,3)
183.* XX=0.0
184.* CALL PLOT(XX,YY,IPEN)
185.* CALL SPOT(XX,YY)
186.* IPEN=2
187.* GO TO 72
188.* 180 CALL PLOT(XX,YY,IPEN)
189.* CALL SPOT(XX,YY)
190.* IPEN=2
191.* 72 CONTINUE
192.* IF(NCNT=LCNT)120,300,120
193.* 300 CALL NUMBER(XX,*5.0,0.07,DAY,90.0,1)
194.* CALL NUMBER(XX,*4.8,0.07,YMB,90.0,1)
195.* CALL NUMBER(XX,*4.6,0.07,YEAR,90.0,1)
196.* CALL NUMBER(XX,*4.4,0.07,HOUR,90.0,1)
197.* CALL PLOT(XX,YY,3)
198.* NCNT=1
199.* GO TO 15
200.* 120 NCNT=NCNT+1
201.* OUTPUT NCNT
202.* GO TO 15
203.* END
### BLANK COMMON (0 WORDS)

### INTRINSIC SUBPROGRAMS USED:

<table>
<thead>
<tr>
<th>ABS</th>
<th>CBS</th>
<th>SWRT</th>
</tr>
</thead>
</table>

### EXTERNAL SUBPROGRAMS REQUIRED:

<table>
<thead>
<tr>
<th>ID</th>
<th>EXIT</th>
<th>FIND</th>
<th>GINBT</th>
<th>ISM</th>
<th>NUMBER</th>
<th>PLT</th>
<th>PLTTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>EXIT</td>
<td>FIND</td>
<td>GINBT</td>
<td>ISM</td>
<td>NUMBER</td>
<td>PLT</td>
<td>PLTTS</td>
</tr>
<tr>
<td>SPRT</td>
<td>WHERE</td>
<td>F1101</td>
<td>F1102</td>
<td>F1103</td>
<td>F1104</td>
<td>F1105</td>
<td>F1106</td>
</tr>
<tr>
<td>F1108</td>
<td>MIDL</td>
<td>-MILC</td>
<td>9HCHEAD</td>
<td>9HCWR</td>
<td>9CB9</td>
<td>SEND10</td>
<td>9INITIAL</td>
</tr>
</tbody>
</table>

### HIGHEST ERROR SEVERITY: 1 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WB9DS</td>
<td>WB9DS</td>
</tr>
</tbody>
</table>

### GENERATED CODE:

<table>
<thead>
<tr>
<th>ID</th>
<th>611</th>
<th>00262</th>
<th>(NO MEMORY PROTECTION)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>21</td>
<td>00015</td>
<td></td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>1136</td>
<td>00470</td>
<td></td>
</tr>
<tr>
<td>TEMPS</td>
<td>1</td>
<td>00001</td>
<td></td>
</tr>
</tbody>
</table>

### TOTAL PROGRAM: 1769 | 00659 |
PROGRAM PROJ4

VERSION 1 FEB 1973, TO OUTPUT DECIMAL DEGREES IN 1A FIELD
VERSION 25 AUGUST 1972, TO UPDATE YIN@8 CALLING ARGUMENTS
VERSION 14 MARCH 1972, ADDING IMANT TO ARG LIST TO PIN@8
VERSION OF 29 AUGUST 1971

PROGRAM PROJ4, PROJECTS SEVERAL DATA FORMATS TO A STRAIGHT LINE
CAN PROCESS DIFFERENT FORMATS IN SAME RUN
ON FORMAT CHOICE IN PUT CARD, PUT JFMTS IN ORDER DESIRED FOR
PROCESSING DURING RUN

JFMT ALLOWS CHOICE OF DATA FORMAT TO BE PROCESSED
JFMT = 1 FOR GSUM FORMAT
 = 2 FOR SEAG1 INPUT FORMAT, OUTPUT IS GSUM FORMAT
 = 3 FOR SPFMT FORMAT
 = 4 FOR SEISMICITY DATA FORMAT
 = 7 FOR ACTIVE VOLCANES

ITAPE IS SET TO (JFMT + 2)
UTAPE = 2
KTAPE=20

SHOULD HAVE ASSIGN CARDS FOR UNITS 2,3,4,5,6,9,20

UNIT 2 IS FOR OUTPUT OF PROJECTED DATA
UNIT 3 FOR GSUM
UNIT 4 FOR SEAG1
UNIT 5 FOR SPFMT DATA
UNIT 6 FOR SEISMICITY DATA
UNIT 9 FOR ACTIVE VOLCANES DATA
UNIT 20 FOR DISK STORAGE OF GSUM DATA IF ISW(3)=1

TO STOP PROCESSING MAKE START DAY=99
IF START DAY = 99, PROGRAM GOES TO 1000 AND CHECKS INPUT FORMAT
LABELS FOR FORMAT CODE NUMBER OF ZERO

SSW(3) = 1 TO OUTPUT GSUM DATA INTO A DISK FILE, ONLY
APPLIES WHEN JFMT = 1 OR 2

DIMENSION IA(35),JF(10)
DIMENSION IDESC(6),VEL(8),THICK(8),X(8)
DIMENSION N8H(4)
DIMENSION IZ(9),IW(35)

CALL STAT
NWBN=1
IIN=105
IOUT=108
JTAPE=2
JFCT=1
JREC=0
I8=8
IREC1=1
DEGRA=1,745329E-02
RADEG=57.29578
PRINT DATE AND TIME OF JOB ON HEADING
CALL TODAY(NOW)
WRITE(IOUT,13) NOW
FORMAT(1X,4A4)
WRITE(IOUT,16)
FORMAT(1PR8J4 RUN, VERSION OF 6 MAY 1974+)
C
INITIALIZING PROJECTION REQUIREMENTS
C
CALLPR8J(KK,RLAT,RLONG,PLAT,PLONG,DISR,DIST)
C
READING ORDER THAT DATA FORMATS ARE TO BE PROCESSED
C
READ(IIN,20)JFMT,JF(2),JF(3),JF(4),JF(5),JF(6),JF(7),JF(8),JF(9)
FORMAT(915)
C
START OF READING NEW DATA FORMAT BEGINS HERE WITH SSW CARD
C
INIT*ISw(-2)
CALL SETSKP(IND)
WRITE(IOUT,25) JFMT
FORMAT(JFMT = '1,14)
ITAPE=FMT=2
GB TO (40,50,60,70,80,90,100,1000,110) JFMT
K8=0
CALLGINMT(ITAPE,ITAPE,KK,KGDA,KGM8,
KGYR,KGHR,IDIF,IS8RC,RLAT,RLONG,ELEV,K87,88SG,
IDEF,FA,BG,TC,IELC,IGC,RAF,IREGC,IFFC,IA,IFBC)
GO TO 100
READ(IIN,55)IS8RC,IELC,IGC
FORMAT(315)
ELEV=0.0
I8G=82
TC=99.9
RAF=0.0
IREGC=0.0
IFFC=0.0
IFBC=0
KK=0
CALL SINGT(ITAPE,ITAPE,KK,KGDA,KGM8,KGYR,KGHR,IDIF,RLAT,
106*  
1  RLBNG, KVN, KVE, K977, KGR, KFA, KBG, KCVN, KCVE,
107*  
2  KDPM, MTDG, MT, MAG1, MAG2, KETV8,
108*  
CALL  GINST (ITAPE, JTAPE, K, KGDA, KGMB,
109*  
1  KYR, KGHM, IDIF, ISORC, HLAT, RLEN, ELEV, K977, BBSG,
110*  
2  IDEPFA, BG, TC, IELC, IGC, RFA, IREGC, IFFC, IA, IFBC)
111*  
GO TO 100
112*  
80 CONTINUE
113*  
KK=0
114*  
CALL  FInsT (ITAPE, JTAPE, K, ISTA, KEY, LAT, LATM, KNS,
115*  
1  LENG, LBM, KEV, VEL, THICK, IMANT, NELEV, N1, N2, N3, N4, MET, IYR, IDESC,
116*  
2  DINE, STHIK, CRV, MGNT, AVTN, CRVW, KGTW, AVTH)
117*  
GO TO 100
118*  
90 CONTINUE
119*  
KK=0
120*  
CALL  YINST (ITAPE, JTAPE, K, ISS, ISRE, KD, KM, SEC, PLAT, KSN, DBN, KWE, DEPT, AMAG, IMB,
121*  
2  ISUS, INT, IUS, IMT, ISCH, IYRC, IYDR, INANT, IMG, IFEG, IMS, ISAP, IZH,
122*  
3  IC, IMG, IAUTH, IGHY, NPP, 180, 1LM, IS1, IS2)
123*  
GO TO 100
124*  
C FOR PRESENT THERE IS NO VOLCANIC STREAM
125*  
110 GO TO 1000
126*  
100 CONTINUE
127*  
C READING START & END DATE & ISKP CARD FOR EACH INPUT FORMAT
128*  
IFLAG =
129*  
READ(IIN, 2) ISTDA, ISTMO, ISTYR, ISTHM, IENDA, IEMP, IENMR, IENHM, ISKP
130*  
2 FORMAT (312*14s5x*312*14s5x, 15)
131*  
WRITE (11UT, 6365) ISTDA, ISTMO, ISTYR, ISTHM, IENDA, IEMP, IENMR, IENHR
132*  
1 IENHM, ISKP
133*  
6365 FORMAT (PROJ: START DATE '1312141', END DATE '1312141', ISKP=
134*  
11+14)
135*  
IF (ISKPE .EQ. 0) GO TO 8
136*  
CALL SKPRES (ITAPE, ISKP)
137*  
GO TO (999, 999, 999, 999, 999) IND
138*  
8 CONTINUE
139*  
C CHECK IF END OF PROCESSING
140*  
C BY CHECK IF ISTDA=99
141*  
IF (ISTDA .EQ. 99) GO TO 992
142*  
9 CONTINUE
143*  
C INITIALIZATION NOW COMPLETE
144*  
145*  
READ IN DATA
146*  
147*  
150 GO TO (200, 300, 400, 500, 1000, 1000, 530) JFM
148*  
149*  
200 KK=1
150*  
CALL  GINST (ITAPE, JTAPE, K, KGDA, KGMB,
151*  
1  KYR, KGHM, IDIF, ISORC, HLAT, RLEN, ELEV, K977, BBSG,
152*  
2  IDEPFA, BG, TC, IELC, IGC, RFA, IREGC, IFFC, IA, IFBC)
153*  
IF (KK=9) 210, 1000, 210
154*  
210 GO TO 186
155*  
300 KK=1
156*  
CALL  SInSt (ITAPE, JTAPE, K, KGDA, KGMB, KYR, KGHM, IDIF, HLAT,
157*  
1  RLBNG, KVN, KVE, K977, KGR, KFA, KBG, KCVN, KCVE,
158*  
2  KDPM, MTDG, MT, MAG1, MAG2, KETV8)
159  IF (KK=9) 310 TO 1000 TO 310
160  310 DB$ = 10GR
161  GS$ = 10GR
162  FA*FLAT(KFA)*0.1
163  BG*FLAT(KBG)*0.1
164  GO TO 186
165  400 KK = 1
166  CALL PIN8 ITAPE, JTAPE, KK, IST, KEY, LAT, LATM, KNS,
167          1 LANG8 LB8, K1N, VEL8 THICK, IHANT, ILEV, N1, N2, N3, N4, IRET, IYR, IDESC,
168          2 DIN8 STIK, CRV8N, GT8N, AV8TN, CRV8, GT8W, AV8WN
169  IF (KK = 9) = 10 TO 1000 TO 410
170  410 RLATM = LATM
171  RLM = LOM
172  CALL NAVIN(LAT, RLATM, KNS, L898, RLM, KEW, RLAT, RL898)
173  GO TO 186
174  500 KK = 1
175  CALL Y8NT ITAPE, JTAPE, KK,
176          1 IS8, ISRP, KDA, KB8, K1R, KHM, SEC, DLAT, KSN, D89N, KWE, DEPT, A89G, IMB,
177          2 IS8, I8TS, IDIAS, ITS, IS8, I8EICH, IV8LC, I89T, I8G, IFEG, IMS, IASP, IZH,
178          3 I8C, I8G, I8AUTH, I8HY, I8PP, I8D, ILM, IS8, IS82
179  IF (KK = 9) = 15 TO 1000 TO 510
180  510 K8L = 0
181  CALL D8NAV(DLAT, KSN, DL89N, KWE, RLAT, RL898, KKL)
182  GO TO 186
183  530 GO TO 1000
184  186 CONTINUE
185  IF (KK = 9) 190 TO 992 TO 190
186  C CHECKING FOR ERR ON INPUT DATA
187  190 JDA = KDA
188  JMB = K8M
189  JYR = K1R
190  J8HP = KHM
191  C I8FL8G IS A FLAG TO ALLOW SKIPPING THE FIRST CALL TO FIND
192  C IF WE HAVE ALREADY FOUND THE STARTING DATE
193  C
194  IF (IFLAG NE 0) GO TO 82
195  CALL FIND(I8STD, J8TH, I8TR, I8NM, JDA, JMB, JYR, JHM, INDK)
196  IF (INDK, EQ = 1) GO TO 190
197  I8FL8G = 1
198  82 CONTINUE
199  IF (IY8R EQ 0) GO TO 851
200  CALL FIND(I8END, I8N8M, I8NYR, I8NM, JDA, JMB, JYR, JHM, INDK)
201  IF (INDK, EQ = 1) GO TO 995
202  851 CONTINUE
203  852 CONTINUE
204  C
205  C START PROCESSING
206  C
207  C
208  C
209  C KK = N998
210  CALL PROJ(KK, RLAT, RL898, PL89N, PL898, DIS8, DIST)
211  C CHECKING IF DATA POINT IS WITHIN AREA AND DISTANCE FROM PROJ LINE
IF (KK+2) 599, 150, 599
CONTINUE

JREC=JREC+1
X=ISR*6371+0
OUTPUT X

OUTPUT DATA

G9 T6 (600, 600, 700, 800, 1000, 1000, 830) JFMT

600 KK=2

LOGIC FOR PUTTING RLAT AND RLANG VALUES IN ARRAY 'IA'

GROUPING VARIABLES FOR OUTPUT UNDER ARRAY 'IA'

DLAT=RLAT*RADEG
DLON=RLONG*RADEG
ENCODE(350, 608, 12) DLAT, DLON

FORMAT(2F9.4, 17X)
CALL UNPKBY(12, IA, 35)
DB 609
J=1, 35
IA(J)=ISL(J)-1

CONTINUE

PUTS RLAT AND RLONG IN POSITION OF PLAT AND PLANG

CALL GIN&T(ITAPE, TAPE, KK, KGDA, KGM8, KGMB, KGCR, ISBC, PLAT, PLANG, ELEV, K977, BSGS,
1 IDEF, FA, BG, TC, IELC, IGCarFA, IREGC, IFFC, IA, IFBC)
IF (ISR(ki) 900, 900, 612
WRITE(ITAPE, 12) IREC, ISBC, KGDA, KGM8, KGCR, KGMB,
1 PLAT, PLANG, ELEV, K977, BSGS, IDEF, FA, BG, TC, IELC, IGCarFA, IREGC, IFFC, IA, IFBC

FORMAT(1, 14, 31, 14, 2F9.6, F7.2, 13, F6.2, I5, 2F6.4, F4.1,
1 212, F5.1, 1, 11, 12, 35A1, 12)
G9 T6 900

700 KK=2

KDEC*C
CALL NAVST(PLAT, PLANG, LAT, RLATM, KNS,
1 LONG, RLONM, KEW, KDEC)
LAT*RLATM
LON*RLONM

CALL FINST(ITAPE, TAPE, KK, ISTA, KEY, LAT, LATM, KNS,
1 LONG, LONM, KEV, THICK, IMANT, ELEV, N1, N2, N3, N4, MET, IYR, IDESC,
1 DINE, STH, KCRV, KGT, AVHTN, CRVR, KGT, AVHTW)
G9 T6 900

800 KK=2

KLS=1
CALL DNAV(DLAT, KSN, DLON, KEW, PLAT, PLANG, KL)
CALL YINST(ITAPE, TAPE, KK,
1 ISR, KDA, KMB, KYR, KL, SEC, DLAT, KSN, DLON, KHE, DEPT, AMAG, IMB,
2 ISBS, INTS, DIS, ITSU, IS1CH, IVELC, INANT, ICHG, IFEG, IMS, IASP, IZH,
3 ICE, IMB, IAUTH, ICHG, APP, ICHG, ILM, IS1, IS2)
G9 T6 900
IF (IENYRESEQO) GoTo 951
CALL FIND(IENDA, IENMB, IENYR, IENMH, JDA, JMB, JYR, JHM, INDK)
IF (INDK.EQ.0) GoTo 100
GoTo 150
GoTo 150
WRITE (I116UT,993) JIREC,
FERMAT('END OF PROCESSING, RECORDS OUTPUT = ',18)
CALL EXIT
WRITE (CTI@UT2996) JDAs UMOaJYRa UHM :
FORMAT('END DATE PASSED!,2X,312,14)
CALL EXIT
WRITE (II8UT,998) IND
FORMAT('ERROR IN SPREC,IND=',IND)
CALL EXIT
C
END OF AN INPUT STREAM
C
CONTINUE
C
WRITING AN B ON OUTPUT STREAM FOR INPUT TO MODPLOT PROGRAM
C
IF JFMT = SEISMICITY OR VOLCANOES WRITE ONLY ON B, OTHERWISE WRITE TWO B'S
C
IF (JFMT=4) 1002,1008,1002
1002 IF (JFMT=7) 1007,1008,1007
1007 WRITE (UTAPE,1001) 18
1008 WRITE (UTAPE,1001) 18
1005 JFCT=JFCT+1
1005 IF (JFMT) 1994,994,1005
1004 IF (JFMT=JFCT) 1994,994,1005
1004 JFCT=JFCT
C
WRITING JFMT NUMBER ON OUTPUT FOR INPUT TO MODPLOT PROGRAM
C
C
WRITE (UTAPE,1001) JFMT
C
FORMAT(11,20X)
C
GoTo 18
C
END
INTRINSIC SUBPROGRAMS USED:

FLOAT  ISL

EXTERNAL SUBPROGRAMS REQUIRED:

DNAV  EXIT  FIND  GINBT  ISM  NAVIN  NAVS  PINBT
PBOJ  SETSKP  SINGT  SKPREC  STAT  TODAY  UNPKG  YINBT
F1101  F1102  F1103  F1104  F1105  F1106  F1108  MIDD
M10C  9BCDREAD  9BCDWRIT  9ENCODE  9SENDIL  9INITIAL  9IDDATA  9IDLUSA
9170R  9RT01

HIGHEST ERROR SEVERITY: 0 (No Errors)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>GENERATED CODE: 972 033CC (No Memory Protection)</td>
<td></td>
</tr>
<tr>
<td>CONSTANTS: 7 00007</td>
<td></td>
</tr>
<tr>
<td>LOCAL VARIABLES: 259 00103</td>
<td></td>
</tr>
<tr>
<td>TEMPS: 0 00000</td>
<td></td>
</tr>
<tr>
<td>TOTAL PROGRAM: 1238 004D6</td>
<td></td>
</tr>
</tbody>
</table>
PROGRAM SAINT2, VERSION OF 7 AUG 1975, TO PROVIDE FOR MAGNETIC TAPE TAPE INPUT AND OUTPUT OF GSUM RECORDS.

VERSION OF 7 AUG 1974, TO IDENTIFY XSCAL.

VERSION OF 6 FEB 1973, TO REALLY CHANGE GSUM OUTPUT TO DEC DEG.

VERSION OF 3 SEPT 1972, TO CHANGE PCS DATA INPUT FORMAT AND TO ALLOW GSLY INPUT ON DEVICES OTHER THAN CARDS.

VERSION OF 21 AUGUST 1972, TO INPUT AND OUTPUT GSUM IN DEC DEG.

MOD MAY 20, 1971 BY FOLINSBEE TO FIT UNDER NEW MONITOR ARRAYS INTO COMMON.

PROGRAM SAINT2, MODIFIED FROM SAINT ON 27 NOV 1970 BY C. BROWIN.

THIS IS A PROGRAM WRITTEN TO INTERPOLATE DATA AT EVEN INTERVALS.

DIMENSION A(10), NAME(5), SCALE(5), AMINV(5), AMAXV(5), Y(5)

COMMON RLAT(1000), RLBG(1000), GAV(1000), FAIR(1000), BUG(1000),

DIMENSION KEY(60)

DIMENSION JAI(10), VAL(10), ARG(10)

DIMENSION JA(10), OBT(10), VAL(640), ARG(40)

DIMENSION JA(10), OBT(90), VAL(640)

DOUBLE PRECISION G

DATA NAME/GAV, FAIR, BUG, DEP, ELEV/

DATA KEY/UCC81, UCC1, '0', '06', '01', '01', '01', '01'

DATA EPS/G=5

SSK(1) = 1 TO SORT DATA

SSK(2) = 1 TO PUNCH SORTED DATA

SSK(3) = 1 TO PRINT SORTED DATA

SSK(4) = 1 TO INTERPOLATE VALUES

SSK(5) = 1 TO PUNCH INTERPOLATED DATA IN GSUM FMT

SSK(6) = 1 TO PUNCH INTERPOLATED FREE-AIR ANOMALY DATA IN TALPLOT FMT

SSK(7) = 1 FOR PRINTED INTERPOLATED DATA AND SUMMARY OF JOB

SSK(8) = 1 FOR PRINTING HEADING AND VALUES OF INPUT DATA

SSK(9) = 1 TO PUNCH INTERPOLATED ELEVATION DATA IN TALPLOT FMT

SSK(10) = 1 TO READ GSUM DATA FROM MAGNETIC TAPE

SSK(11) = 1 TO WRITE INTERPOLATED DATA TO GSUM FMT ON MAGNETIC TAPE.

LAST INPUT GSUM DATA CARD SHOULD HAVE A 9 IN COLUMN 1.

SETUP INPUT AND OUTPUT DEVICES

IIN=105

IN=105

IOUT=108

OUT=108

IPUNCH = 106

TAPE=2

ITAPE=1

IFILE=43

IFILE=4

IFILE=3

GID=0+0

OUTPUT 'PROGRAM SAINT2, VERSION OF 7 AUG 1975.'

DEGRA=1.74329E+02

RACEG=57.29578

INIT=ISW(-2)

ENTER OPTIONS FOR PROCESSING.
5 READ(I1N,500,END=99) KKM, XSCAL, NPTS
6 FORMAT(F5*1,F5*1,15)
7 C
8 C KKM = KM DISTANCE BETWEEN INTERPOLATED DATA POINTS
9 C XSCAL = MAX DISTANCE FOR WHICH POINTS ARE GIVEN UNITY WEIGHT.
10 C BEYOND THIS DROPS LIKE KKM/XSCAL.
11 C
12 IF(NPTS.EQ.C) NPTS=4
13 NIM=2*NPTS
14 WRITE(IOUT,650) KEY
15 650 FORMAT(T50, 'INPUT PARAMETERS'/
16 A 'SORTING KEY USED/3(20A/4)')
17 OUTPUT KKM,XSCAL,NPTS
18 C
19 C INITIALIZING FUNCTION WT
20 C
21 IWTSET(XSCAL)
22 C READING PCS PARAMETERS AS INPUT TO SUBROUTINE PROJ
23 READ(I1N,200) JA, ANG, DMAX, ILAT, RILTN, RILNM, JB
24 200 FORMAT(5A10, F10.0, 5F10.0, 2F10.0)
25 WRITE(IOUT,200) JA, ANG, DMAX, ILAT, RILTN, RILNM, JB
26 RILT=DMAX(RILTN, RILNM)
27 RILG=DMAX(RILTN, RILNM)
28 IF(JA.EQ.1) WRITE(IOUT,604)
29 604 FORMAT(T50, 'INPUT DATA'/
30 'A RECORD LATITUDE LONGITUDE GRAVITY FREE AIR BUOYER DE' BPTY ELEVATION TCBR IELC/1 NUMBER RADIANS RADIANS MG
31 520 FORMAT(5A10, F10.0, 5F10.0, 2F10.0)
32 WRITE(IOUT,520) VAs ANGS CMAXM, TLATARILTM, LLONG, RILOM, JB
33 READ (ITAPE,520) VAs ANGS CMAXM, TLATARILTM, LLONG, RILOM, JB
34 WRITE (IOUT,604)
35 CONTINUE
36 WRITE(IOUT,520) VAs ANGS CMAXM, TLATARILTM, LLONG, RILOM, JB
37 READ (ITAPE,520) VAs ANGS CMAXM, TLATARILTM, LLONG, RILOM, JB
38 WRITE(IOUT,604)
39 CONTINUE
40 IREC=IREC+1
41 IF(JA.EQ.1) WRITE(IOUT,604)
42 WRITE(IOUT,520)
43 READ (ITAPE,520) VAs ANGS CMAXM, TLATARILTM, LLONG, RILOM, JB
120 IF (T2 > T1 + T4 + T3) GO TO 34
121 C0 37 I1 \NVAR
122 IF (RLONG(I) < LT.BLONG) BLONG=RLONG(I) / BLAT=RLAT(I)
123 CALL CONTINUE
124 CONTINUE
125 C0 31 I1 \N8
126 CALL DISTZ(RLAT(I),RLONG(I),RLAT(I),RLONG(I),BLAT,RLAT(I))
127 IF (ANG) \G4.4.4.4.2
128 IF (AZ=135)/(48)/(48)/(48)/(48)
129 IF (AZ=225)/(48)/(48)/(48)/(48)
130 DISTKM=DISTKM
131 CONTINUE
132 DIST(1)\DISTKM *10000*0
133 CONTINUE
134 IF (ISH(1) \G 1 OR GREATER SORTS \O OTHERWISE
135 IF (ISH(1) \G 0) GO TO 3C
136 C0 11 I1 \NVAR
137 G=GRV(I)+977000*
138 WRITE(IFILE1,503)
139 1 IREC,RLAT(I),RLONG(I),ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC,
140 2 DIST(I)
141 503 FORMAT(1,14X,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6)
142 CONTINUE
143 SORTING PART
144 END FILE IFILE1
145 REMIND IFILE1
146 OUTPUT 1,1="SORTING TAKING PLACE==1"
147 CALL CLOFIL(IFILE1)
148 IN SHORT ARGUMENT LIST
149 2ND ARG, 4 IS UNIT NO. FOR INPUT
150 3RD ARG, 5 IS UNIT NO. FOR OUTPUT
151 CALL THORT(KEY,4,5,IC0DE)
152 OUTPUT IC0DE
153 CALL OPIN(IFILE3)
154 OUTPUT 1=*END OF SORT==1"
155 C0 13 I1 \N8
156 READ(IFILE3,503)
157 1 IREC,RLAT(I),RLONG(I),ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC,
158 2 DIST(I)
159 503 FORMAT(1,14X,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6)
160 CONTINUE
161 1=CHOOSE TO PRINT AND PUNCH DATA
162 C ISH(2) \G 1 TO PUNCH DATA
163 IF (ISH(2) \G 0) GO TO 3C
164 C0 14I1 \N8
165 G=GRV(I)+977000*
166 DLAT=RLAT(I)*RADEG
167 DLONG=RLONG(I)*RADEG
168 WRITE(IPUNCH,502)
169 1 IREC,DLAT,DLONG,ELEV(I),G,DEPTH(I),FAIR(I),BOUG(I),TC,IELC
170 CONTINUE
171 14 CONTINUE
172 30 CONTINUE
173 C IF (ISH(3) \G 1 WANT PRINTED OUTPUT OF SORT
174 IF (ISH(3) \G 0) GO TO 3S
175 WRITE(OUT,602)
176 602 FORMAT(1/4,14X,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6,2F9.6)
177 A1 RECORD LATITUDE LONGITUDE GRAVITY FREE AIR BOUGUER DE
178 BPTH ELEVATION DISTANCE KM/1 NUMBER RADIANS RADIANS MG
BALSMGALSMGALSMTSMTSFROM#1

C832I=1:N8
G=GRV(I)/SGC*10
ADIST=DIST(I)/100000
WRITE(I8UT,6D3)
A(1,101),RLONG(I),G,FAIR(I),BUGU(I),DEPT(I),ELEV(I),ADIST
603 FORMAT('1/1,1X, 15,4X, F9.6,2X,F9.6,1X,F9.6,1.4X,F6.1,2X,F6.1,3X,F6
A=0.5X,F6.0/3X,F8.1)
CONTINUE
C IF ISK(4)=1 WILL INTERPOLATE
35 IF(ISK(4)=4) GO TO 50
C CALCULATING HEIGHT AND PLACING IT IN ELEV FIELD
GO TO 450
C HEIG=ADIST(I)
GO TO 450
45C CONTINUE
C IF(ISK(7)=EG(I)) WRITE(I8UT,610)
CONTINUE
C OIF ISWK EG, WILL SAYS a
35 IFC ISW(4),EG, WRITE(TOUT,610)
CONTINUE
C DETERMINING NEAREST FIELD POINT TO FIRST MEMBER OF DATA ARRAY
IZERB=DIST(1)/10000*xKm
X=10000+IZERB*xKm
ISW(5)=OUTPUT IN GSUM PUNCH
ISW(6)=OUTPUT IN TALPLT PUNCH
ISW(7)=PRINTED OUTPUT, INTERPOLATED
44C CONTINUE
IF(X.GT.CIST(NG) AND 0) 614 CONTINUE
614 FORMAT('X')
C FOR USE IN ATSM AND PICK THE SORTED ARRAY DIST MUST BE STORED
C IN ORDER FROM THE SMALLEST TO LARGEST (IE. DIST(I))
C MUST BE < OR = DIST(*1)
1F(ARG(DIST(IN)=DIST(IN))=X) RRLON=RLONG(IN))
1/RRLAT=RLAT(IN)/DIST(IN))
1/RRLAT=RLAT(IN)/DIST(IN))
1/RRLAT=RLAT(IN)/DIST(IN))
1/RRLAT=RLAT(IN)/DIST(IN))
IN=106
NORD=IN
CALL ATSM(X,DIST(IN),RRLAT(RNLAT(IN))
INORD
CALL ATSM(X,DIST(IN),ARG,IAL,NORD)
C THIS SECTION IS CALCULATING THE SUM OF THE WEIGHTS OF 3 OR 4
C CLOSEST VARIABLES AND THE SUM OF ALL THE WEIGHTS
S3=WT(ARG(1)*X)*WT(ARG(2)*X)*WT(ARG(3)*X)
S4=S3+WT(ARG(4)*X)
S4=EX
DB 664 IX,5,NDIM
S4=5*EX
CONTINUE
C WE START BUT WITH NORD (ORDER OF POLYGON) ≠ 3. THEN IF THE FOLLOWING
C CONDITIONS ARE NOT MET WE REDUCE THE ORDER OF THE POLYGON
C THESE CONDITIONS ARE ONLY ESTIMATES AND SHOULD PROBABLY BE
C CHANGED AS FURTHER EXPERIENCE IS GAINED WITH THE PROGRAM.
C
NORD = 3
246. IF (S4 ≤ 4 AND S3 ≤ 4) NORD = 2
247. IF (S3 ≤ 4 AND S3 ≤ 4) NORD = 1
248. IF ((S4 ≤ 4) AND (S4 ≤ 4)) NORD = 2
249. IF ((S4 ≤ 4) AND (S4 ≤ 4)) NORD = 1
250. CALL SETAL (GRV (IZ), JAL, VAL, NDIM, NORD, A)
251. CALL EGN (X, ARG, VAL, NDIM, NORD, A)  
252. GEF = A (1)
253. CALL SETAL (FAIR (IZ), JAL, VAL, NDIM, NORD, A)
254. CALL EGN (X, ARG, VAL, NDIM, NORD, A)  
255. FF = A (1)
256. CALL SETAL (EBUG (IZ), JAL, VAL, NDIM, NORD, A)
257. CALL EGN (X, ARG, VAL, NDIM, NORD, A)  
258. BB = A (1)
259. CALL SETAL (ELEV (IZ), JAL, VAL, NDIM, NORD, A)
260. CALL EGN (X, ARG, VAL, NDIM, NORD, A)  
261. EE = A (1)
262. FATP = FF
263. C CONVERT ELEVATION TO KMS FOR OUTPUT AT TALPLOT ELEV INPUT
264. IF (HINT (I) = 0.001)
265. IF (ISW (S) = 0) GO TO 705
266. C OUTPUT INTERPOLATED VALUES AT GSUM FMT ON TWO CARDS
267. C CONVERT TO DECIMAL DEGREES
268. DRLAT = RRLAT = RADI
269. DRLON = RRLON = RADI
270. IF (ISW (11) = 0) GO TO 699
271. WRITE (TAPE, 636) DRLAT, DRLON, EE, GD, DD, FF, BB
272. 696 FORMAT ( 11X, 4X, 2F9.4, 2F9.4, 6F5.2, 2F6.1)
273. GO TO 705
274. 699 WRITE (IPUNCH, 700) DRLAT, DRLON, EE, GD, DD, FF, BB
275. 700 FORMAT ( 11X, 4X, 2F9.4, 2F9.4, 6F5.2, 2F6.1 /
276. 1 13X, 'INTERPOLATED GSUM FROM SAINT !)
277. 705 X = 10000
278. 41 IF (ISW (7) = 0) WRITE (ICUT, 611) IG, FF, BB, DD, EE, XD, NORD, RRLAT, RRLON
279. 611 FORMAT ( 11X, IG, 5X, X = 10000
280. 612 FORMAT ( 11X, IG, 5X, X = 10000
281. 613 FORMAT ( 11X, IG, 5X, X = 10000
282. 614 FORMAT ( 11X, IG, 5X, X = 10000
283. 615 FORMAT ( 11X, IG, 5X, X = 10000
284. 616 FORMAT ( 11X, IG, 5X, X = 10000
285. 617 FORMAT ( 11X, IG, 5X, X = 10000
286. 618 FORMAT ( 11X, IG, 5X, X = 10000
287. 619 FORMAT ( 11X, IG, 5X, X = 10000
288. 620 FORMAT ( 11X, IG, 5X, X = 10000
289. 621 FORMAT ( 11X, IG, 5X, X = 10000
290. 622 FORMAT ( 11X, IG, 5X, X = 10000
291. 623 FORMAT ( 11X, IG, 5X, X = 10000
292. 624 FORMAT ( 11X, IG, 5X, X = 10000
293. 625 FORMAT ( 11X, IG, 5X, X = 10000
294. 626 FORMAT ( 11X, IG, 5X, X = 10000
295. 627 FORMAT ( 11X, IG, 5X, X = 10000
296. 628 FORMAT ( 11X, IG, 5X, X = 10000
297. 629 FORMAT ( 11X, IG, 5X, X = 10000
298. 630 FORMAT ( 11X, IG, 5X, X = 10000
299. 631 FORMAT ( 11X, IG, 5X, X = 10000
300. 632 FORMAT ( 11X, IG, 5X, X = 10000
301. 633 FORMAT ( 11X, IG, 5X, X = 10000
302. 634 FORMAT ( 11X, IG, 5X, X = 10000
303. 635 FORMAT ( 11X, IG, 5X, X = 10000
304. 636 CONTINUE
305. 436 CONTINUE
306. IF (ISW (6) = 0) WRITE (IPUNCH, 615) (FATP (I), I = 1, NUMIT)
307. 615 FORMAT (5F10.1)
308. IF (ISW (5) = 0) WRITE (IPUNCH, 620) (HINT (I), I = 1, NUMIT)
309. 620 FORMAT (5F10.3)
310. 621 CONTINUE
311. 50 CONTINUE
312. 156 CONTINUE
313. GO TO 5
314. 99 CONTINUE
315. STOP
316. END
PROGRAM SELSP

VERSION OF 11 AUG 75 (ORIGINAL)

OUTPUT 'PROGRAM SELSP = VERSION OF 11 AUG 75'

FOR SELECTING SPFMT OUTPUT OF CRWT3 (DERIVED FROM PROGRAM SBRT3)

PROGRAM SBRT3, FOR SORTING OUTPUT OF CRWT2

DIMENSION IDESC(6), VEL(8), THICK(8)

INITIALIZATION

IN = 105
IBUT = 108
ITAPE = 11, JTAPE = 2
NREC = 0
IBREC = C

SENSE SWITCH CARD IS REQUIRED BY PINOT

INIT = ISW(2)

KK = 0

CALL PINOT (ITAPE, JTAPE, KK, ISTA, KEY, LAT, LATM, KNS, LNS, LBM, N1, N2, N3, N4, N5, MET, IYR, IDESC, DINE,

25THICK, CRVN, AVTN, CRVT, WGTN, AVWTN, CRVT, WGTN, AVWTN)

ISTAG = 0

KS = 1 KS

NNS = 1 HS

NEW = 1 HW

READ (IN, 2) IMET, IMET, IPREV, IPREV, WP, IMOP, SIMND, SMAX

2 FORMAT (6F10.0)

3 IF (IDP = INDICATOR FOR SORT ON WATER DEPTH OR ELEVATION)

4 DMIN = MINIMUM WATER DEPTH OR ELEVATION

5 DMAX = MAXIMUM WATER DEPTH, OR ELEVATION

6 READ (IN, 3) IDP, DMIN, DMAX

7 3 FORMAT (15F10.0)

8 IF (IPREV < IMET) CALL ARLIM (IN, ISUB, RTOP, RBOT, LEFT, RIGHT)

9 IF (IDP = IDP) CALL ARLIM (IN, ISUB, RTOP, RBOT, LEFT, RIGHT)

10 CONTINUE

CONTINUE

CALL PINOT (ITAPE, JTAPE, KK, ISTA, KEY, LAT, LATM, KNS, LNS, LBM, N1, N2, N3, N4, N5, MET, IYR, IDESC, DINE,

25THICK, CRVN, AVTN, CRVT, WGTN, AVWTN, CRVT, WGTN, AVWTN)

NREC = NREC + 1

END OF FILE OR END OF TAPE ENCOUNTERED DURING READ

IF (KK = KEY = 9) GO TO 999

C CHECKING IF KEY = 9

18 IF (KEY =? 9) 20 10 20

20 CONTINUE

C CHECKING FOR SBRT CHECKS

ISTAG = ISTA

IF (IMET = IMET) 310, 310, 310

30C IF (IMET = MET) 10, 310, 10

31C IF (IPREV = IPREV) 315, 320, 315

31E NPREV = N1000 + N200 + N30 + N4
IF (IPROVE=NPREV) 10, 320, 10
320 IF (IMDP) 325, 330, 325
325 IF (DINE=SMIND) 10, 327, 327
327 IF (SMAXD=DINE) 10, 330, 330
330 IF (IAR) 335, 360, 335
335 RLATM=RLATM
360 RLM=RLM
361 RLAT=DMTR(LAT,RLATM)
362 RLONG=DMTR(LONG,RLGM)
354 IF (KEW=NEW) 358, 356, 358
356 RLONG=RLONG
358 CALL ARCK(RLAT,RLONG,RTOP,RGBT,RLEFT,RRIGT,IND)
360 IF (IND) 10, 360, 10
363 IF (IDP) 364, 370, 362
362 IF (N1*EG>2) ELEV=NELEV GO TO 366
366 GO TO 10
364 IF (N1*EG>2) GO TO 10
368 IF (ELEV>DMIN) 10, 368, 366
370 CONTINUE
374 CALL BNT(RLAT,RLONG,RTOP,RGBT,RLEFT,RRIGT,IND)
380 IF (IND) 10, 380, 380
384 CALL ARCK(RLAT,RLONG,RTOP,RGBT,RLEFT,RRIGT,IND)
388 CONTINUE
390 WRITE (10, 600) NRECs, JAREC
600 FORMAT ('NUMBER OF RECORDS INPUT=', I10)
100 CONTINUE
HIGHEST ERROR SEVERITY: C (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GENERATED CODE: 370 0C172
CONSTANTS: 5 00005
LOCAL VARIABLES: 80 00000
TEMPS: 2 00002

TOTAL PROGRAM: 457 001C9
**PROGRAM SPFMT**

**VERSION OF 21 AUGUST 1971, TO READ NEW CARD SET READED JULY 71**

**DIMENSION IDESC(6), VEL(8), THICK(8), X(8)**

**DIMENSION UDESC(6)**

**PROGRAM SPFMT, CONVERTS SEISMIC REFRACTION COLUMN FORMAT FROM U OF TORONTO WORLD SEISMIC REFRACTION COMPILED**

**FORMAT TO SPFMT FORMAT**

**EACH PROFILE IN ONE 120 CHARACTER RECORD**

**USES SUBROUTINES ISW, STAT, EVIL**

**-------------------------------**

**ITAPE # URN FOR SEISMIC DATA INPUT**

**JTAPE # URN FOR SPFMT DATA OUTPUT**

**ITAPE = 1**

**JTAPE = 2**

**-------------------------------**

**NOTE**

**IN ORDER TO OUTPUT THE LAST DATA CARD, A BLANK CARD MUST FOLLOW THE DATA CARDS.**

**-------------------------------**

**IIIN = 105**

**IIOUT = 108**

**CALL STAT**

**INN = ISW(-2)**

**WRITE (IIOUT, 605)**

**605 FORMAT (/ 'PROGRAM SPFMT! //')**

**OUTPUT, 'VERSION OF 21 AUGUST 1971!**

**ISTAB = 0**

**ND = 0**

**NREC = 0**

**IFST = 0**

**-------------------------------**

**L9 = 9**

**LC = 0**

**L3 = 3**

**L4 = 4**

**KNUM = #**

**L9 = 1H9**

**L0 = 1H0**

**L3 = 1H3**
L4 = 1H4
KNUM = 1H0
L2=1H2
L6=1H6
L8=1H8
LK=1HK
LM=1HM
LP=1HP
LR=1HR
LT=1HT
LV=1HV
LX=1HX
LZ=1HZ
LG=1HW
IREC1 = 1
DINE = .000
RTHIK = 0.00
CRVN = 0.00
WGTN = 0.00
AVWTN = 0.00
CRVM = 0.00
WGTM = 0.00
AVWTM = 0.00

C
KDA = 0
KM6 = 0
KYR = 0
C READING U OF TORNTB WORLD SEISMIC REFRACTION COMPILATION
999 CONTINUE
READ(ITAPE,12)ISTA,JKEY,JLAT,JLATM,JKNS,DLONG,DLAM,
1 JKEV,V1,V2,V3,T3,4,T4,VMATJ,ELEVJ,
2 JN3,JN4,JMET,JYR,DESC
12 FORMAT(14,A1,12.I12,A1,13,12.A14,F2,12,F3,10)
call stat(1)
call evil(iout,i,ibad,kda,km6,kyr,ista)
90 IF (IBAD) 999,13 995
91 13 CONTINUE
92 C CHECKING FOR KEY CODE FOR SECOND CARD
93 C
94 C
95 IF(JKEY=L2)501,550,501
96 501 IF(JKEY=L6)502,550,502
97 502 IF(JKEY=L8)503,550,503
98 503 IF(JKEY=LM)504,550,504
99 504 IF(JKEY=LP)505,550,505
100 505 IF(JKEY=LT)506,550,506
101 506 IF(JKEY=LV)507,550,507
102 507 IF(JKEY=LT)508,550,508
103 508 IF(JKEY=LV)509,550,509
104 509 IF(JKEY=LX)510,550,510
105 510 IF(JKEY=LX)511,550,511
106. IF(\texttt{KEY} = \texttt{LZ}) GOTO 512
107. IF(\texttt{KEY} = \texttt{LQ}) GOTO 513
108. GOTO 301
109. IF(\texttt{ISTA} = \texttt{ISTA0}) GOTO 444
110. \texttt{ISTA0} = 0
111. GOTO 999
112. CONTINUE
113. CONTINUE
114. \texttt{KEY} = \texttt{KEY1}
115. KEY = \texttt{KEY2}
116. \texttt{LAT} = \texttt{LAT1}
117. \texttt{LATM} = \texttt{LATM1}
118. \texttt{KNS} = \texttt{KNS1}
119. \texttt{LONG} = \texttt{LONG1}
120. \texttt{LONM} = \texttt{LONM1}
121. \texttt{KEY} = \texttt{KEY2}
122. \texttt{VMANT} = \texttt{VMANT1}
123. \texttt{VMANT} = \texttt{VMANT1000}
124. \texttt{ELEV} = \texttt{ELEV1}
125. \texttt{ELEV} = \texttt{ELEV1000} + \texttt{SIGN(RND,ELEV)}
126. \texttt{K1} = \texttt{K11}
127. \texttt{K2} = \texttt{K12}
128. \texttt{K3} = \texttt{K13}
129. \texttt{K4} = \texttt{K14}
130. \texttt{MET} = \texttt{MET1}
131. \texttt{YR} = \texttt{YR1}
132. \texttt{DB} = \texttt{DB1}
133. \texttt{K3} = \texttt{K136}
134. \texttt{K3} = \texttt{K134}
135. CONTINUE
136. \texttt{IND} = 2
137. CONTINUE
138. \texttt{IND} = 2
139. \texttt{FST} = 0
140. CONTINUE
141. \texttt{J} = \texttt{VEL(1)} + 10.0 + 0.5
142. \texttt{J} = \texttt{VEL(2)} + 10.0 + 0.5
143. \texttt{J} = \texttt{VEL(3)} + 10.0 + 0.5
144. \texttt{J} = \texttt{VEL(4)} + 10.0 + 0.5
145. \texttt{J} = \texttt{VEL(5)} + 10.0 + 0.5
146. \texttt{J} = \texttt{VEL(6)} + 10.0 + 0.5
147. \texttt{J} = \texttt{VEL(7)} + 10.0 + 0.5
148. \texttt{J} = \texttt{VEL(8)} + 10.0 + 0.5
149. \texttt{J} = \texttt{THICK(1)} + 10.0 + 0.5
150. \texttt{J} = \texttt{THICK(2)} + 10.0 + 0.5
151. \texttt{J} = \texttt{THICK(3)} + 10.0 + 0.5
152. \texttt{J} = \texttt{THICK(4)} + 10.0 + 0.5
153. \texttt{J} = \texttt{THICK(5)} + 10.0 + 0.5
154. \texttt{J} = \texttt{THICK(6)} + 10.0 + 0.5
155. \texttt{J} = \texttt{THICK(7)} + 10.0 + 0.5
156. \texttt{J} = \texttt{THICK(8)} + 10.0 + 0.5
157. \texttt{J} = \texttt{THICK(9)} + 10.0 + 0.5
158. \texttt{J} = \texttt{THICK(10)} + 10.0 + 0.5
159. k3 = THICK(3) *10.0 + 0.5
160. k4 = THICK(4) *10.0 + 0.5
161. k5 = THICK(5) *10.0 + 0.5
162. k6 = THICK(6) *10.0 + 0.5
163. k7 = THICK(7) *10.0 + 0.5
164. k8 = THICK(8) *10.0 + 0.5
165. WRITE(JTAP,990) IREC, ISTA, KEY, LAT, LATH, KN, LNG, LBM,
166. 1 K1, J1, J2, J3, J4, J5, J6, J7, J8, K8
167. 2 IMANT, NELEV, N1, N2, N3, N4, MET, IYR, IDESC, DINE, STH, CRV
168. 3 WGTN, AVHTN, CRV, WGTW, AVWTH
170. 1 I1, I2, 6A2, 2F4, 1, F3 = 1, 2F6, 0, 1, F3 = 1, 2F6, 0, 5X
171. 1 NREC = NREC + 1
172. C SETTING VELOCITY AND THICKNESS ARRAYS = ZERO
173. DO 602 I = 1, 8
174. VEL(I) = 0.0
175. THICK(I) = 0.0
176. CONTINUE
177. G9 TO (330/610)IND
178. ISTAB = 0
179. G9 TO 999
180. 301 IF (KEY = -9) 310, 305, 310
181. 305 IF (V1 = 0.01) 999, 999, 310
182. 310 IF (IFST) 312, 320, 312
183. 312 IND = 1
184. G9 TO 600
185. 320 IFST = 1
186. 330 CONTINUE
187. C C CONVERT READ VALUES TO THOSE FOR LAYERS 1 = 4
188. C
189. ISTA = ILEST
190. KEY = JKEY
191. LAT = JLAT
192. LATH = JLATH
193. KN = JKN
194. LNG = JLONG
195. LBM = JLBM
196. KEW = JKKEW
197. VMANT = VMATJ
198. IMANT = VMANT = 10.0
200. ELEV = ELEV
201. NELEV = ELEV - 100.0 + (SIGN(RND, ELEV))
202. N1 = JN1
203. N2 = JN2
204. N3 = JN3
205. N4 = JN4
206. MET = JMET
207. IYR = JIYR
208. D9 340 I = 1, 6
209. IDESC(I) = JDESC(I)
210. CONTINUE
211. 340 VEL(I) = V1
VEL(2)  = V2
VEL(3)  = V3
VEL(4)  = V4
THICK(1) = T1
THICK(2) = T2
THICK(3) = T3
THICK(4) = T4

ISTAG = JIST

C READ NEW INPUT CARD
GO TO 999

995 END FILE JTAPE
WRITE(16UT,996)NREC
996 FORMAT(15OF FOUND, NREC = ', I8)

CALL EXIT
END
LOCAL VARIABLES (135 WORDS):

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>JDSC</td>
<td>0006</td>
</tr>
<tr>
<td>000025</td>
<td>UTAPE</td>
<td>0026</td>
</tr>
<tr>
<td>000038</td>
<td>NREC</td>
<td>0032</td>
</tr>
<tr>
<td>000031</td>
<td>KNUM</td>
<td>0032</td>
</tr>
<tr>
<td>000037</td>
<td>LP</td>
<td>0038</td>
</tr>
<tr>
<td>000030</td>
<td>K4</td>
<td>0054</td>
</tr>
<tr>
<td>000055</td>
<td>T2</td>
<td>0056</td>
</tr>
<tr>
<td>000056</td>
<td>ELEVJ</td>
<td>0056</td>
</tr>
<tr>
<td>000041</td>
<td>ULV</td>
<td>0062</td>
</tr>
<tr>
<td>000067</td>
<td>KNS</td>
<td>0068</td>
</tr>
<tr>
<td>000066</td>
<td>ELEVL</td>
<td>0068</td>
</tr>
<tr>
<td>000073</td>
<td>MET</td>
<td>0074</td>
</tr>
<tr>
<td>000079</td>
<td>J4</td>
<td>0074</td>
</tr>
<tr>
<td>00007F</td>
<td>K2</td>
<td>0080</td>
</tr>
<tr>
<td>000005</td>
<td>K8</td>
<td>0086</td>
</tr>
</tbody>
</table>

000016 X
000028 INN
000029 ISTAD
00002A RMD
00002B L0
000034 L8
000035 LK
000038 LK
0000C L2
000042 HDTN
000043 H4
000040 AVTH
000042 CRV
000047 KDA
000048 KMN
000049 J3
000050 J4
000056 J5
000058 J6
000052 J1
000053 J1
000059 J4
000060 J4
00005E JN4
000064 KEY
000066 LAT
000068 VMANT
00006C IMANT
000001 X
000029 JDSC
00002A ITAPE
00002B L0
000034 L8
000035 LK
000038 LK
0000C L2
000042 HDTN
000043 H4
000040 AVTH
000042 CRV
000047 KDA
000048 KMN
000049 J3
000050 J4
000056 J5
000058 J6
000052 J1
000053 J1
000059 J4
000060 J4
00005E JN4
000064 KEY
000066 LAT
000068 VMANT
00006C IMANT

BLANK COMMON (0 WORDS)

INTRINSIC SUBPROGRAMS USED:

EXTERNAL SUBPROGRAMS REQUIRED:

<table>
<thead>
<tr>
<th>Name</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVIL</td>
<td></td>
</tr>
<tr>
<td>EXIT</td>
<td></td>
</tr>
<tr>
<td>ISW</td>
<td></td>
</tr>
<tr>
<td>STAT</td>
<td></td>
</tr>
<tr>
<td>F101</td>
<td></td>
</tr>
<tr>
<td>F102</td>
<td></td>
</tr>
<tr>
<td>F103</td>
<td></td>
</tr>
<tr>
<td>F104</td>
<td></td>
</tr>
<tr>
<td>SENDOL</td>
<td></td>
</tr>
<tr>
<td>GINITL</td>
<td></td>
</tr>
<tr>
<td>GISDATA</td>
<td></td>
</tr>
<tr>
<td>GISUSA</td>
<td></td>
</tr>
<tr>
<td>GPRINT</td>
<td></td>
</tr>
<tr>
<td>GRTSI</td>
<td></td>
</tr>
<tr>
<td>GSTOP</td>
<td></td>
</tr>
<tr>
<td>SENDFILE</td>
<td></td>
</tr>
</tbody>
</table>

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>Type</th>
<th>Dec Words</th>
<th>Hex Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEC</td>
<td>562</td>
<td>00232</td>
</tr>
<tr>
<td>HEX</td>
<td>00016</td>
<td>00001</td>
</tr>
<tr>
<td>WORDS</td>
<td>135</td>
<td>0087</td>
</tr>
<tr>
<td>TEMPS</td>
<td>00001</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL PROGRAM: 720 00230

(GENERIC CODE)
286

COMPILED 24 April 1975

1. PROGRAM TALPLOT 16
2. OUTPUT: TALPLOT 16 RUN: VERSION OF 8 APRIL 75.
3. VERSION 8 APRIL 75 TO ZERO VARIABLES.
4. VERSION 8 APRIL 1971, CHANGING TALPLOT 15 TO 16 AND
   CORRECTING ERRORS IN 3D BOUGUER ANOMALY (BG0D) CALCULATION.
5. BOTH VALUES WERE ONLY CORRECT IN PAST WHEN ROENS EQUALLED
   THE VALUE 2pD.
6. VERSION OF NOV 15 TO CORRECT OUTPUT ON TAPE OF NEW FIELD POINTS
7. SUCH THAT PLOTTING PROGRAM TERMINATES CORRECTLY.
8. MODIFIED OCT 14 TO CORRECT HEIGHT BUG AND TO REMOVE PUNCHING OF
   MODIFIED POINTS. TIMER FEATURE ADDED.
9. VERSION OF SEPT 28 CORRECTED MT CALCULATION FOR ELEVATION
10. AND IMPLEMENTED FILTERING OF RESIDUALS.
11. MODIFIED SEPT 14, 1971, BY APOLINSBEE.
12. TO FILTER THE RESIDUAL BETWEEN OBSERVED AND THEORETICAL GRAVITY.
13. MODIFIED JUNE 2 71 TO USE ELEVATIONS IN CRUSTAL MT CALCULATION.
14. VERSION OF 8 APRIL 71, WRITES COMPLETE BOUGUER ON TAPE:
15. LAST CHANGE FEB 3 71 TO READING OF ELEVATIONS.
16. LAST CHANGE JAN 19, 71 TO REMOVE 9GA BUG:
17. LAST CHANGE REMOVE 9GA SET TO OF BUG ON JAN 19, 71, APOLINSBEE.
18. TALPLOT 15 FROM TALPLOT 14 ON OCT 23, 1970 BY APOLINSBEE.
19. THIS IS A PROGRAM TO DETERMINE LAND AND SEA DATA.
20. ELEVATION MUST FOLLOW 9GA DAT, BEGINNING WITH A NEW CARD IN FORMAT.
21. TYPICAL USE ON SIGMA 7: 7 TRACK MAGNETIC TAPE.
22. OUTPUT REPRODUCES INPUT AND ALSO RESULTS.
23. ASSIGN 9 TRACK MAG TAPE TO UNIT NUMBER 2.
24. LAST BODY POINT IN EACH POLYGON MUST HAVE A 9 IN COL. 21
25. WEIGH EXPECTS THE DIMENSION OF X, Y, Z TO BE 3, 4, 5.
26. THE HEIGHT IS CALCULATED FOR THE INPUT POLYGONS, 2 AND 3.
27. FOR THE MODIFIED POLYGONS.
28. ISP(1) = 0 WRITE TAPE.
29. ISP(1) = 1 NO WRITE TAPE.
30. ISP(2) = 0 SETS ELEV.
31. ISP(2) = 1 READS IN ELEV IN KM.
32. ISP(3) = 0 PRINTS INTERMEDIATE DATA FOR EACH POLYGON.
33. ISP(3) = 1 NO PRINT OF INTERMEDIATE DATA.
34. ISP(4) = 0 PLOT INTERMEDIATE DATA.
35. ISP(4) = 1 NO PLOT.
36. ISP(5) = 0 NO PLOT OF ELEVATION.
37. ISP(5) = 1 PLOT ELEVATION.
38. ISP(6) = 1 TO ADJUST LAST CURVE TO FIT GRAVITY DATA.
39. SS(7) UP FOR OUTPUT DURING DEBUGING ONLY.
40. ISP(3) = 0 TO NOT WRITE INTERMEDIATE DATA FOR EACH POLYGON.
41. JTAPE.
BOe €

OLE C ESSSW(44) UP TO USE ELEVATION VALUES IN CRUSTAL WT CALCULATION

C IF SSSW(44) IS UP THEN THE GRAVITY VALUES READ IN SHOULD

BE COMPLETE BOURGER ANOMALIES

IATE #1 MEANS THAT THIS POINT WILL BE VARRIED TO COMPUTE A BEST FITTAUDD400

BOE €

C THIS SHOULD BE USED WHEN THE GRAVITY ANOMALIES ARE BOURGER ANOMALIES

AND THUS THE TOP OF THE MODEL IS AT SEA LEVEL. A DENSITY OF 2.67

C IS USED IN MAKING THE AT CORRECTION

C SSSW(44) #1 TO OUTPUT RESULTS OF INPUT POLYGONS BEFORE

ALTERNATE THE VARIABLE BOUNDARY POINTS

C THE POLYGON THAT IS TO BE VARRIED MUST BE THE LAST POLYGON TO BE TAUN0410

C THE POINT TO BE VARRIED MUST NOT BE THE FIRST OR Last POINT IN THE TAUN0420

C POLYGON

C IMAX IS THE MAXIMUM NUMBER OF MODELS THAT WILL BE CALCULATED

IMPO IS THE NUMBER OF MODELS THAT HAVE BEEN CALCULATED

LOGICAL PAR*FALSE*

COMMON FER(200)*FER(100110)

DIMENSION LABEL(20) TAUN00460

DIMENSION COME(5) TAUN00460

DIMENSION FXZ(200),FZ(200)

DIMENSION PDZLZ(200)*SSZLZ(200)*Z0(11)*Z0(1)*Z2(200)

1 DATA(200)*RESA(200)*TEST(200), DSU(200)

DIMENSION ARRAY (200),SUM(200),XSW(200),PSW(200),SUMK(200)

DIMENSION PDEL(20)*X5(20)

DIMENSION ALT(10),AA(200,21),KK(5),Z15,PDELZ(20)

DIMENSION YGGA(200)

DIMENSION STSUM(200)

EQUIVALENCE(ARRAY(11),FX(1),ARRAY(12),SSZLZ(1)),

1 (ARRAY(13)*RESA(1),ARRAY(14),YGGA(1))

100 FORMAT(5F5.1) TAUN00570

CALL TIC
d 0 1 K=1,200

1 FER(K),0

ILM=0

98 IMEST

99 IFIRST=0

100 FS=1,E70

101 IMD=0

102 PDEL=1

103 D(2)= DEL

104 D(+)+DEL

105 KK(1)=1

106 KK(3)=1

107 KK(5)=1

108 HCO=0

109 D(1)=0

110 D(3)=0

111 D(5)=0

112 KK(3)=0

113 KK(4)=0

114 D(1)=1*100

115 RESA(1)=0

116 FX(1)=0

117 FZ(1)=0

118 PDELZ(1)=0

119 SSZLZ(1)=0

120
```
101 CONTINUE
102 CONTINUE

C NOTE THAT THE LAST POLYGON MUST HAVE NUMBER 99

C HPG = REFERENCE DENSITY FOR GRAVITY CALCULATIONS
C WHT = REFERENCE HEAVY FOR MASS CALCULATIONS
C

C HPG = REFERENCE DENSITY FOR GRAVITY CALCULATIONS
C WHT = REFERENCE HEAVY FOR MASS CALCULATIONS

C
```
I*1+1
IF (ICRUE=9) G01 X10 Y01
410 CONTINUE
412 IF (TEST .EQ.1) G0 TO 811
CALL GETOP (XZ,NZ,FZ,MUX,MYR,KTEST,DSU)
811 CONTINUE
413 IF (K(3).EQ.0) PRINT 47
414 C PRINT POINT D3 LU8P
415 D" D54.0<12<15XN0
416 C PELZ*0
417 C PELZ*0 PRINTS D3 LU8P
418 C D*1 G00 I*1
1010
205 EAXX.*X(I) = FX(K)
202 ZFX.*X(I) = FZ(K)
209 CALL CAMP
304 CONTINUE
305 IF (LUMP.EQ.1) PDELZ(K) = RH0*(13.34*SDELZ-PCON(K)/RHO)*SDELZ(K)*13.34*RHO*SDELZ
306 CONTINUE
309 SDELZ(K) = SSELZ(K)+PDELZ(K)
505 IF (L*1) G0107 X10 Y1010
702 CONTINUE
509 CONTINUE
501 TD = XRA(K)+PDELZ(K)*(2.57*RHO)/RHO
207 PRINT 5057 X/K,FZ(X),FX(K),PDELZ(K)
307 PRINT 5057 X/K,FZ(X),FX(K),PDELZ(K)
709 CONTINUE
401 CONTINUE
407 IF (ISX(I).EQ.1) G0 TO 7009
408 WRITE (JTAPIE,5007) K,FX(K),FZ(K),PDELZ(K),BGTD
409 X TO 7009
7009 CONTINUE
421 CONTINUE
427 IF (ISX(I).EQ.1) G0 TO 423
CALL PLTER(PDELZ,YBAR)
433 CONTINUE
430 IF (LENX) G043 X60
431 REFC + REFC + SSELZ(J)
432 SSELZ(K) = SSELZ(K) + REFC
428 CONTINUE
425 SREFC = REFC + REFC
426 TPSX0
427 ARHLS*0
430 REFC*0
CALCULATING THE RMS ERROR

IF (SGA(K) < GE*90%) GOTO 4422

IF (IFER+LT+1) IFER(K) = RESA(K) GOTO 4029

IF (F(K)*Q)

CONTINUE

IF (KHA(K) < GE*90) GOTO 4424

FI A(K) IFER(K)*RESA(K) = KHM

IF (IFER(K) = RESA(K) = KHM)

CONTINUE

IF (F(K)*Q) CONTINUE

PRINT (14,4422) IMординates RESF

IF (10) EQE 1 AND IFIRST 0 GOTO 439

CONTINUE

C saving out of model altering part of program

CONTINUE

IF (IFER(K) = RESA(K) = KHM)

GOTO 437

C COMPUTATION OF D/22 FOR THE REFERENCE POINT

10 765: I=I+1

IF (IALTE1=EQ10) GOTO 7650

562: SDELE2

563: 20 7640: DUM*15

564: TMDOD

565: x=IF(K)*F(K)

566: X=ESAGA(K)*F(K)+D(I1)*SIGN(1+ARG)

567: CALL CIMP

568: CONTINUE

569: "CH*CH=1"
RFDEL(MCH)=13*3+RHDD*SDELZ

IF(15*7*EQ.1) WRITE(11,6701) KCODE, K, RFDEL(KD), KD, MCH)

CONTINUE

STORING THE OLD VALUES OF PDELZ

1 IF(15*7*EQ.1) GO TO 434

3 G8 432 K*K*FAN,M

PDELZ(K)

G8 7835 K*K*FAN,M

IF(50A(K) .GE. 900) GO TO 7850

IF(K .NE. JREF) GO TO 7850

CONTINUE

FIELD POINT 08 Loop

C IF(15*7*EQ.1) WRITE(11,6701) KCODE, K, RFDEL(KD), KD, MCH)

C COMPUTING 0/07 FOR EACH CHANGEABLE POLY POINT

C IF(FALT+1) .EQ. 0) GO TO 7830

SDELZ=0

G8 7820 11=1*5

EXXX(1)(K)(I)+FX(K)

ZERO=1+ (K)(I) .EQ. 0)

100M

C CALL CMP

7820 CONTINUE

3 SDELZ(KM)+13*3+RHDD*SDELZ+RFDEL(MCH)

7830 CONTINUE

KCODE1

1 IF(15*7*EQ.1) WRITE(11,6701) KCODE, K, SDELZ(KD), KD, MCH)

5-contINUE

50 Z835 I1=1*MCH

Z831 I1=1*MCH

AA(I1,IP)+AA(I1,IP)+SDELZ(IP)+SDELZ(I1)

CONTINUE

AA(I1,CH+1)=AA(I1,CH+1)+FER(K)*SDELZ(I1)

CONTINUE

CONTINUE

IF(HAVE N\n
FINISHED SETTING UP THE NORMAL EQUATIONS

C B P = , +1

C NOTE THAT NRC IS THE NUMBER OF COLUMNS IN THE MATRIX AA
CALCULATE THE TRUE VALUES OF THE POLYGON POINTS

IF (TEST-EN+1) WRITE (NEAR POLY POINTS) WRITE (FINAL VERSION)

IF (TEST-EN+1) WRITE (TAPE, X(N), Z(N), IC08E)

CONTINUE
440.  OUTPUT 2: DIFFERENCE BETWEEN THEORETICAL AND OBSERVED GRAVITY
441.  OUTPUT 3: OBSERVED GRAVITY
442.  OUTPUT 4: ELEVATION IN 100S OF METERS
443.  IF(ISW(13) NE 1) AND IFIRST*EQ.0) GOTO 4423
444.  993 CONTINUE
445.  C STORE THE VALUE OF X
446.  IF(ISW(6) NE 1) GO TO 2916
447.  4E(D[114]*X3**3+X*7011) IF(REX=0)
448.  IF(ISW(679) BE TO 7011
449.  INITIAL
450.  IMPED
451.  IF(ISW(679)=EQ.00) RRS=0
452.  IF(ISW(679)=LE 650) RRS=1
453.  630 X=DELZ(K)-PSDELZ(K)
454.  C DELZ(K) IS THE SUMMED VALUE OF ALL POLYGON CONTRIBUTIONS
455.  C EXCEPT THAT DUE TO THE CHANGED PART OF THE MODEL
456.  OUTPUT NEW RECALCULATING THE MODEL USING NEW VARIABLE POLYPINT
457.  KG
458.  GO TO 699
459.  C IS USED TO SET EQUAL TO 1 TO INDICATE THAT
460.  C WE ARE READING AN ADDITIONAL SET OF POINTS FOR THE LAST
461.  C POLYGON TO SEE THE EFFECT OF USING DIFFERENT
462.  C VARIABLE POLYGON POINTS
463.  7011 CONTINUE
464.  CALL TIC(TIME)
465.  OUTPUT TIME
466.  STOP
467.  SUBROUTINE COMP...
468.  IF(EQZ)210*240*240...
469.  210 IF(EQQ)120*230*270...
470.  220 THEBSATAM(EQQ)EQZ+3*1415927...
471.  230 THEBSATAM(EQQ)EQZ+3*1415927...
472.  240 IF(EQQ)250*260*270...
473.  250 THEBSATAM(EQQ)EQZ+3*1415927...
474.  260 THEBSATAM(EQQ)EQZ+3*1415927...
475.  270 THEBSATAM(EQQ)EQZ+3*1415927...
476.  280 THEBSATAM(EQQ)EQZ+3*1415927...
477.  290 THEBSATAM(EQQ)EQZ+3*1415927...
478.  300 THEBSATAM(EQQ)EQZ+3*1415927...
479.  310 THEBSATAM(EQQ)EQZ+3*1415927...
480.  320 THEBSATAM(EQQ)EQZ+3*1415927...
481.  330 THEBSATAM(EQQ)EQZ+3*1415927...
482.  340 THEBSATAM(EQQ)EQZ+3*1415927...
483.  350 THEBSATAM(EQQ)EQZ+3*1415927...
484.  360 THEBSATAM(EQQ)EQZ+3*1415927...
485.  370 THEBSATAM(EQQ)EQZ+3*1415927...
486.  380 THEBSATAM(EQQ)EQZ+3*1415927...
487.  390 IF(EQQ)350*360*360...
488.  350 DTHETAM=EQQ...
489.  360 IF(EQQ)350*360*360...
490.  370 THEBSATAM(EQQ)EQZ+3*1415927...
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C = 1.0 * (LEF-E) * ALU_3(3R/8)</td>
</tr>
<tr>
<td>2</td>
<td>CFL2 * (H+C)</td>
</tr>
<tr>
<td>3</td>
<td>4.1</td>
</tr>
<tr>
<td>4</td>
<td>2002</td>
</tr>
<tr>
<td>5</td>
<td>ZEE * ZFEE</td>
</tr>
<tr>
<td>6</td>
<td>RRR</td>
</tr>
<tr>
<td>7</td>
<td>THETA * THETA</td>
</tr>
<tr>
<td>8</td>
<td>RETURN</td>
</tr>
<tr>
<td>9</td>
<td>END</td>
</tr>
<tr>
<td>DEC WORDS</td>
<td>HEX WORDS</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
</tr>
<tr>
<td>GENERATED CODE:</td>
<td>1946</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>35</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>4155</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>8</td>
</tr>
<tr>
<td>TOTAL PROGRAM:</td>
<td>6144</td>
</tr>
</tbody>
</table>
SUBROUTINE ALTIE(ELEV, IDEP, HEIGT, KK)

VERSION 8F 13 JANUARY 1971

SUBROUTINE ALTIE, RETURNS VALUE OF HEIGT (NEGATIVE BELOW SEA
LEVEL). KK RETURNS = 9 IF NEW DATA RECORD SHOULD BE READ, OTHERWISE KK = 0.

KK = 0
A = ABS(ELEV)
IF (A = 0.004) 100, 100, 200

C ELEV = ZERO

C CHECKING DEPTH

IF (IDEP) 110, 110, 130

KK = 9
GO TO 990

C ELEV NOT ONE

GO TO 990

GO TO 990

C ELEV NOT ZERO

IF (ELEV) 210, 100, 210

GO TO 990

RETURN

END
### LOCAL VARIABLES (2 WORDS)

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTD</td>
<td>R</td>
<td>SCALR</td>
<td>00001 V</td>
<td>1</td>
</tr>
<tr>
<td>IDEP</td>
<td>I</td>
<td>SCALR</td>
<td>00003 V DUMMY</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELEV</td>
<td>R</td>
<td>SCALR</td>
<td>00002 V DUMMY</td>
<td></td>
</tr>
<tr>
<td>KK</td>
<td>I</td>
<td>SCALR</td>
<td>00005 V DUMMY</td>
<td></td>
</tr>
</tbody>
</table>

### BLANK COMMON (0 WORDS)

### ENTRY POINTS

- COOC ALTD

### INTRINSIC SUBPROGRAMS USED

- ABS

### EXTERNAL SUBPROGRAMS REQUIRED

- 9108
  - 9SETUP

### HIGHEST ERROR SEVERITY: C (39 ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>CO021</td>
</tr>
<tr>
<td>1</td>
<td>CO001</td>
</tr>
<tr>
<td>2</td>
<td>CO002</td>
</tr>
<tr>
<td>E</td>
<td>CO005</td>
</tr>
<tr>
<td>41</td>
<td>CO029</td>
</tr>
</tbody>
</table>

- GENERATED CODE: 33
- CONSTANTS: 1
- LOCAL VARIABLES: 2
- TEMPS: 5
- TOTAL PROGRAM: 41
SUBROUTINE ANGVE2(2Z,2HT,AX,KGDA,KGM,KGY,RKGHM,KGDA,XX,YY,INIT)
1 DATA, IEC)
C 24 JULY 1974 - TO ADD HORIZ AND VERT ANNOTATION
C
C SUBROUTINE ANGVE2 ANNOTATES PLOTTED POINT WITH DATA VALUE, TIME, OR DATE AND TIME
C
C SS*(3) UP TO ANNOTATE ONLY AT CHANGE OF DATE
C SS*(7) UP TO ANNOTATE ON LEFT SIDE OF TRACK
C SS*(11) UP TO ANNOTATE ALTERNATELY ON LEFT AND RIGHT SIDES OF TRACK
C SS*(18) C TO HAVE ANNOTATION AT RIGHT ANGLES TO INCREMENTAL TRACK
C 1 TO ANNOTATE HORIZONTALLY
C 2 TO ANNOTATE VERTICALLY
C 3 ON TRACK HEADING 091 TO 269, TO INVERT ANNOTATION
C 4 TO ANNOTATE EITHER HORIZ OR VERT DEPENDING UPON DIRECTION
C
USES CALCMEF SUBROUTINES AND IS*
DIMENSION FM(4), BATH(2), DAY(2)
C THETA IS THE INCREMENTAL TREND OF TRACK
C AX XX X XLD
C AY YY Y YLD
C OPTION TO PLOT VALUES HORIZONTALLY OR VERTICALLY
C IF(ISW(18) GT 0) GM TO 110
C THETA=G
C IF(ABS(AY - C*CH) .GE. 000058358) GO TO 110
C 110 THETA=1.57079
C IF(AABS(AX - C*CH)) 400058358
C 111 CONTINUE
C IF(ISW(18) EQ 4) GM TO 29
C DETERMINE ANGLE ALONG WHICH TO ANNOTATE
C IF(AY) .GE. .5151
C 50 THETA=ABS(AY/AX)
C THETA=1.57079+ATAN(THETA)
C IF(AX*GT*0) THETA=THETA
C GO TO 59
C 51 THETA=ABS(AX/AY)
C THETA=ATAN(THETA)
C IF(AX*GT*0) THETA=THETA
C 49 THETA=ABS(AX/AY)
C 43 THETA=ATAN(THETA)
C 44 IF(AX*GT*0) THETA=THETA
C 45 IF(ISW(18) EQ 3 AND THETA*GT*1.57079) THETA=THETA+3.14159
C 46 IF(ISW(18) EQ 3 AND THETA*LT*1.57079) THETA=THETA+3.14159
C GO TO 58
C 29 BX=ABS(AX*XLD)
C BY=ABS(AY*YLD)
C 50 IF(BX=BY) 30, 36, 40
C ANNOTATE VERTICALLY
C 52 THETA=1.57079
C GO TO 58
C ANNOTATE HORIZONTALLY
C 36 THETA=C*0
C 55 BB THETA-ThETA=57.20657
C CHECK IF DISTANCE INCREMENT ALONG TRACK FROM LAST DATA POINT IS SUFFICIENT
C THAT NEXT ANNOTATION DOES NOT OVERPRINT LAST
C TAX=SQRT(AX*AX+AY*AY)
IF(ISW(1)) 71,71,70
61. 70 TAX=TAX+0.04*2HT
62. 71 TAX=TAX+0.08*2HT
63. C IF TAX IS NEGATIVE, INCREMENT IS INSUFFICIENT TO ANNOTATE, RETURN
64. IF(TAX)*00*92*92
65. 92 IF(ISW(11)+NE+1) G6 TO 6C
66. 91 KBUNT=KBUNT+1
67. ISIDE=ISIDE+1
68. IF(ISIDE) 93*93*93
69. 90 IF(ISW(7)) 94*94*93
70. 93 OFSET=0*34
71. 90 G6 TO 95
72. 94 OFSET=0*34
73. 95 XXT*OFSET=GCS(THATA)
74. YTT=OFSET*GCS(THATA)
75. AXT=1*ZHT*GCS(THATA)
76. B=C*0*14*ZHT*GCS(THATA)
77. =C*0*14*ZHT*GCS(THATA)
78. =C*0*14*ZHT*GCS(THATA)
79. XXT*XT*XT
80. YTT*YTT
81. 10C IF(NX**AE+1) G6 TO 320
82. C CHANGE HOUR AND MINUTE FORMAT FROM (14) TO (411) FORMAT 86 TO PRINT FOUR
83. C DIGITS
84. H^1=(KGDA/1000)
85. H^2=(KGDA*H^1)*1000/100
86. H^3=H^2
87. H^4=(KGDA=H^2)*1000*10
88. H^5=H^4
89. H^6=H^5
90. H^7=H^6
91. C CHANGE DAY AND MONTH FORMATS FROM (12) TO (211) 86 TO PRINT TWO DIGITS.
92. C ANNIMATE DATA POINT
93. DAY(1)*KGDA/10
94. DAY(2)*KGDA*(DAY(1)*10)
95. BTH(1)*KGDA/10
96. BTH(2)*KGDA=(BTH(1)*1000)
97. C ANNIMATE DATA POINT
98. 28C IF(KGDA+NE=KGDA+AND+ISW(3)+EG+1) G6 TO 291
99. 291 CALL NUMBER(XT,YT,HGT,DAY(1)+THETA+01)
100. 291 XT=XT+C
101. YT=YT+C
102. CALL NUMBER(XT,YT,HGT,DAY(2)+THETA+1)
103. XT=XT+B
104. YT=YT+A
105. CALL NUMBER(XT,YT,HGT,BTH+1)+THETA+1)
106. XT=XT+C
107. YT=YT+C
108. CALL NUMBER(XT,YT,HGT,BTH+2)+THETA+1)
109. XT=XT+B
110. YT=YT+A
111. YEAR=YEAR
112. CALL NUMBER(XT,YT,HGT,YEAR,THETA+1)
113. IF(ISW(3)+EG+1) G6 TO 34C
114. XT=XT+(2*0+8)
115. YT=YT+(2*0+A)
116. 29C IF(ISW(3)+EG+1) G6 TO 34C
117. CALL NUMBER(XT,YT,HGT,H^1+THETA+1)
118. XT=XT+C
120: Y T = Y T + C
121: CALL NUMBER (X T, Y T, H G T, H M (2), T H E T A, = 1)
122: X T = X T + C
123: Y T = Y T + C
124: CALL NUMBER (X T, Y T, H G T, H M (3), T H E T A, = 1)
125: X T = X T + C
126: Y T = Y T + C
127: CALL NUMBER (X T, Y T, H G T, H M (4), T H E T A, = 1)
128: G 8 T 0 3 4 0
129: CALL NUMBER (X T, Y T, H G T, D A T A, T H E T A, I D E C)
130: C R E T U R N F E N T 0 1 D A T A P O I N T
131: 3 4 C CALL P L O T ( X X , Y Y , 3)
132: 3 4 2 X 8 L D = X X
133: Y 8 L D = Y Y
134: 4 0 C R E T U R N
135: 8 5 K B L N T = 0
136: T H E T A = C
137: T H E T A = 0
138: X 8 L D = C
139: Y 8 L D = 0
140: G 0 T 0 6 0
141: E N D
### Highest Error Severity: 0 (No Errors)

<table>
<thead>
<tr>
<th>DECODED CODES</th>
<th>DEBCODED CODES</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE:</td>
<td>462</td>
<td>C01CE</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>23</td>
<td>C0C17</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>32</td>
<td>C0C20</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>17</td>
<td>C0C11</td>
</tr>
<tr>
<td>TOTAL PROGRAM:</td>
<td>534</td>
<td>C0216</td>
</tr>
</tbody>
</table>
SUBROUTINE ANOV3(XX,YY,CEPT,AMAG)

VERSION OF 15 DEC 1971, ADD SIZE VARIABLE AND CHANG
DEPTH LIMITS FOR SHALLOW EPICENTERS
SUBROUTINE ANOV3, TO MAKE VARIABLE SIZED SYMBOLS FOR
EPICENTER DATA DEPENDING UPON DEPTH AND MAGNITUDE

CEPT = DEPTH IN KM
AMAG = MAGNITUDE (MAXIMUM IS 7.5)

DATA ISTRT/0/
IF (ISTRT) 15,5,15
OUTPUT 'ANOV3, VER 15 DEC 1971'
SIZE = 1,0
SIZE = 2,0
OUTPUT SIZE
ISTRT = 1

END OF INITIALIZATION

15 IF (CEPT <= 70) (20,20,22
20 INTEG = 1
23 GB TO 50
24 IF (CEPT <= 150) (24,24,26
24 INTEG = 2
26 GB TO 50
27 IF (CEPT <= 300) (28,28,30
28 INTEG = 5
29 GB TO 50
30 IF (CEPT <= 500) (32,32,34
32 INTEG = 12
32 GB TO 50
33 INTEG = 0
34 GB TO 50
35 IF (AMAG <= 5) (52,52,54
36 HF = 1,0
37 GB TO 100
38 IF (AMAG <= 5) (56,56,58
39 HF = 2,0
40 GB TO 100
41 IF (AMAG <= 5) (60,60,62
42 HF = 3,0
43 GB TO 100
44 HF = 4,0
45 GB TO 100
46 HGT = 0, C7*HF*SIZE
47 CALL SYMBOL(XX,YY,HGT,INTEG,0,0,=1)
48 RETURN
49 END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>APAG</td>
<td>R</td>
<td>SCALR</td>
<td>00009</td>
<td>V DUM</td>
<td>ANBV3</td>
<td>R</td>
<td>SCALR</td>
<td>00000</td>
<td>V</td>
<td>1</td>
<td>ANBV3</td>
<td>SPRBG</td>
<td>00000</td>
<td>P</td>
</tr>
<tr>
<td>DEPT</td>
<td>I</td>
<td>SCALR</td>
<td>00003</td>
<td>V</td>
<td>ISTRT</td>
<td>I</td>
<td>SCALR</td>
<td>00001</td>
<td>V</td>
<td>1</td>
<td>HGT</td>
<td>R</td>
<td>SCALR</td>
<td>00005</td>
</tr>
<tr>
<td>INTEG</td>
<td>I</td>
<td>SCALR</td>
<td>00003</td>
<td>V</td>
<td>XX</td>
<td>R</td>
<td>SCALR</td>
<td>00006</td>
<td>V DUM</td>
<td>1</td>
<td>SIZE</td>
<td>R</td>
<td>SCALR</td>
<td>00002</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>SPRBG</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YY</td>
<td>R</td>
<td>SCALR</td>
<td>00007</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (6 WORDS):**

| C0000 ANBV3 | C0001 ISTRT | 00002 SIZE | 00003 INTEG | 00004 HF | 00005 HGT |

**BLANK COMMON (0 WORDS)**

**ENTRY POINTS:**

| 00000 ANBV3 |

**EXTERNAL SUBPROGRAMS REQUIRED:**

| SYMBOL | F11C8 | 9ED16L | 910DATA | 9PRINT | 9SETUPN |

**HIGHEST ERROR SEVERITY:.0 (N9 ERRORS)**

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>95</td>
<td>C005F</td>
</tr>
<tr>
<td>14</td>
<td>C000E</td>
</tr>
<tr>
<td>6</td>
<td>C0006</td>
</tr>
<tr>
<td>5</td>
<td>C0005</td>
</tr>
<tr>
<td>120</td>
<td>C0078</td>
</tr>
</tbody>
</table>
SUBROUTINE AREAK(CLAT,CLONG,IAKEY)
VERSION OF 1 DECEMBER 1971

VERSION OF 18 OCT 1971, DUMMY ROUTINE

SUBROUTINE AREAK, GIVES VALUE TO CODE IAKEY WHICH MAY
SUBSEQUENTLY BE USED IN SORTING GSUM RECORDS INTO A
SO DESIGNATED BY DIFFERENT VALUES FOR IAKEY

IAKEY = 0
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>HEX</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>HEX</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>HEX</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREAK</td>
<td>SPRBG</td>
<td>00000</td>
<td>F</td>
<td>AREAK</td>
<td>R SCALR</td>
<td>00000</td>
<td>V</td>
<td>DLAT</td>
<td>UNUSED</td>
<td>00000</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>CLANG</td>
<td>LINDEX*00002</td>
<td>V DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (1 WORD): COCCC AREAK

BLANK COMMON (0 WORDS)

ENTRY POINTS: COCCC AREAK

EXTERNAL SUBPROGRAMS REQUIRED: 9SETUP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE CODE: 9 COCCC9</td>
<td></td>
</tr>
<tr>
<td>CONSTANT S: 0 COCCC0</td>
<td></td>
</tr>
<tr>
<td>LOCAL VARIABLES: 1 COCCC1</td>
<td></td>
</tr>
<tr>
<td>TEMPS: 4 COCCC4</td>
<td></td>
</tr>
<tr>
<td>TOTAL PROGRAM: 14 COCCC0E</td>
<td></td>
</tr>
</tbody>
</table>
1. SUBROUTINE CALSC(A, B, C, D, SC, CC)
2. C SUBROUTINE CALSC, DETERMINES SIN AND COS OF ANGLE
3. C OF TILT OF DIGITIZED MAP
4. R=SGRT((C-A)**2+(D-B)**2)
5. SC=(D-B)/R
6. CC=(C-A)/R
7. RETURN
8. END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALR</td>
<td>00002</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>CALSC</td>
<td>R</td>
<td>SCALR</td>
<td>00003</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>C</td>
<td>R</td>
<td>SCALR</td>
<td>00004</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>SGRT</td>
<td>R</td>
<td>SPR6G</td>
<td>00005</td>
<td>V</td>
<td>DUMMY</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (2 WORDS):**
- 00002 CALSC
- 00004 R

**BLANK COMMON (1C WORDS)**

**ENTRY POINTS:**
- 00002 CALSC

**INTRINSIC SUBPROGRAMS USED:**
- SGRT

**EXTERNAL SUBPROGRAMS REQUIRED:**
- 9SETUP
- 959RT

**HIGHEST ERROR SEVERITY:** C (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>26</td>
<td>C001C</td>
</tr>
<tr>
<td>C</td>
<td>C000C</td>
</tr>
<tr>
<td>2</td>
<td>C0002</td>
</tr>
<tr>
<td>8</td>
<td>C0008</td>
</tr>
<tr>
<td>38</td>
<td>C0026</td>
</tr>
</tbody>
</table>

**TOTAL PROGRAM:** 38
SUBROUTINE CDATE (IDA1, IM01, IYR1, IMH1,
               1 IDA2, IM02, IYR2, IMH2, TIMD)
C
  C * 2 DEC 1970 /2200 -- S.ABBOTT
  C * K-F ASA BASIC FORTRAN (EXTEDED)
  C * MODIFIED FOR SIGMA 7 -- 20 DEC 71
  C
  C * PURPOSE: COMPARES TWO DATES AND RETURNS THE TIME
  C * DIFFERENCE IN DECIMAL HOURS (TIMD) ;
  C * 'TIMD' WILL BE NEGATIVE IF DATE 1 IS AFTER DATE 2.
  C
  C * EQUIVALENT TO SUBROUTINE CDATE; EXCEPT THAT THE HOUR-MINUTE
  C * ARGUMENTS ARE SUPPLIED AS INTEGER NUMBERS.
  C
  C * THERE ARE NO DATE LIMITS FOR INPUT DATA
  C
  AHM1 = IMH1
  AHM2 = IMH2
  CALL NDH (IDA1, IM01, IYR1, AHM1, IC1, T1)
  CALL NDH (IDA2, IM02, IYR2, AHM2, IC2, T2)
C
  C * CALCULATE TIME DIFFERENCE IN DECIMAL HOURS
  C
  TIMD = (ID2 - ID1)
  TIMD = TIMD * 24
  TIMD = TIMD + (T2-T1)
C
  RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHM1</td>
<td>R</td>
<td>SCALR</td>
<td>00001 V</td>
<td>AHM2</td>
<td>R</td>
<td>SCALR</td>
<td>00002 V</td>
<td>IDA1</td>
<td>I</td>
<td>SCALR</td>
<td>*00007 V</td>
</tr>
<tr>
<td>CDATE</td>
<td>R</td>
<td>SCALR</td>
<td>00000 V</td>
<td>IDA2</td>
<td>I</td>
<td>SCALR</td>
<td>*00008 V</td>
<td>IDM1</td>
<td>I</td>
<td>SCALR</td>
<td>*00003 V</td>
</tr>
<tr>
<td>ID1</td>
<td>I</td>
<td>SCALR</td>
<td>00003 V</td>
<td>IMO1</td>
<td>I</td>
<td>SCALR</td>
<td>*00008 V</td>
<td>IMO2</td>
<td>I</td>
<td>SCALR</td>
<td>*00004 V</td>
</tr>
<tr>
<td>IYR1</td>
<td>I</td>
<td>SCALR</td>
<td>*00009 V</td>
<td>IYRE</td>
<td>I</td>
<td>SCALR</td>
<td>*00006 V</td>
<td>ND1</td>
<td>S</td>
<td>PROG</td>
<td>EXTERN</td>
</tr>
<tr>
<td>TIMC</td>
<td>R</td>
<td>SCALR</td>
<td>*00006 V</td>
<td>T1</td>
<td>R</td>
<td>SCALR</td>
<td>00007 V</td>
<td>T2</td>
<td>R</td>
<td>SCALR</td>
<td>00008 V</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (7 WORDS):

- COCOCC CDATE
- COCOCC AHM1
- COCOCC AHM2
- COCOCC ID1
- COCOCC ID1
- COCOCC ID2

BLANK COMMON (0 WORDS)

ENTRY POINTS:

- COCOCC CDATE

EXTERNAL SUBPROGRAMS REQUIRED:

- ND1
- S108
- 9SETUP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td>COCOCE</td>
</tr>
<tr>
<td>1</td>
<td>COCO01</td>
</tr>
<tr>
<td>7</td>
<td>COCO07</td>
</tr>
<tr>
<td>10</td>
<td>COCO0A</td>
</tr>
<tr>
<td>64</td>
<td>COCO4C</td>
</tr>
</tbody>
</table>

TOTAL PROGRAM
SUBROUTINE CHGTM(KDA, KMB, KYR, KHM, KTZ, KGDA,
1   KGMB, KGYR, KGHM, NTZ)
DIMENSION MBDAY(12)

C SUBROUTINE CHGTM TO DETERMINE GMT DATE AND TIME
FROM LOCAL TIME
C THE SIGN OF THE TIME ZONE DIFFERENCE IS TO GAIN
C FROM GMT TIME TO THE LOCAL TIME.  THUS IF
C GMT = 18:00, AND LOCAL = 14:00, KTZ = +04:

MBDAY(1)=31
MBDAY(2)=28
MBDAY(3)=31
MBDAY(4)=30
MBDAY(5)=31
MBDAY(6)=30
MBDAY(7)=31
MBDAY(8)=31
MBDAY(9)=30
MBDAY(10)=31
MBDAY(11)=30
MBDAY(12)=31

KGHM=KHM-(KTZ*100)

IF(KGHM)110,126,128
KGHM=2400+(KHM-(KTZ*100))
KGDA=KDA=1
A=KYR
B=KYR/4
A=A/4*0
IF(A-B)112,122,112
IF(KGDA)114,114,120
KGMB=KMB=1
IF(KGMB)116,116,120
KGYR=KYR
KGDA=MBDAY(KMBO=1)
GB T6 150
KGMB=12
KGDA=MBDAY(KGMB)
KGYR=KYR=1
GB T6 150
KGMB=KMB
KGYR=KYR
GB T6 150
IF(KMB)112,123,112
IF(KDA)112,124,112
KGDA=MBDAY(KMBO=1)+1
KGMB=KMB=1
KGYR=KYR
GB T6 150
KGDA=KCA
KGMB=KMB
KGYR=KYR
GB T6 150
IF(KGHM)2400,126,131,130
KGHA=CO00
KGDA=KDA=1
A=KYR
B=KYR/4
A=A/4*0

C FOR...
IF (KGDA = MODAY (KMB))

KGMB = KMB
KGYR = KYR
G8 TO 150
KGDA = 1
KGMB = KMB + 1
IF (KGMB = 13)
KGYR = KYR
G8 TO 150
KGMB = 1
KGYR = KYR + 1
G8 TO 150
IF (KMB = 2)
IF (KGDA = 29)
KK = KTZ + 100
KGRM (KHM - KK) = 240C
G8 TO 132
KTZ = KTZ
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R SCALR</td>
<td>000CD</td>
<td>B</td>
<td>R SCALR</td>
<td>000CE</td>
<td>C</td>
<td>R SCALR</td>
<td>000CF</td>
</tr>
<tr>
<td>CHGMT</td>
<td>R SCALR</td>
<td>000CD</td>
<td>KDA</td>
<td>I SCALR</td>
<td>*00010</td>
<td>KGDA</td>
<td>I SCALR</td>
<td>*00015</td>
</tr>
<tr>
<td>KGHM</td>
<td>I SCALR</td>
<td>*00018</td>
<td>KGMP</td>
<td>I SCALR</td>
<td>*00016</td>
<td>KGYR</td>
<td>I SCALR</td>
<td>*00017</td>
</tr>
<tr>
<td>KHM</td>
<td>I SCALR</td>
<td>*00013</td>
<td>KMM</td>
<td>I SCALR</td>
<td>*00011</td>
<td>KMB</td>
<td>I SCALR</td>
<td>*00011</td>
</tr>
<tr>
<td>KTZ</td>
<td>I SCALR</td>
<td>*00014</td>
<td>KYR</td>
<td>I SCALR</td>
<td>*00012</td>
<td>MBDAY</td>
<td>I ARRAY</td>
<td>00001</td>
</tr>
<tr>
<td>ATZ</td>
<td>I SCALR</td>
<td>*00019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (16 WORDS):**

- 00000 CHGMT
- 00001 MBDAY
- 000CD A
- 000CE B
- 000EF KK

**BLANK COMMON (0 WORDS)**

**ENTRY POINTS:**

- 00000 CHGMT

**EXTERNAL SUBPROGRAMS REQUIRED:**

- SITOR
- 9SETUP

**HIGHEST ERROR SEVERITY: C (NO ERRORS)**

- DEC WORDS: 185
- HEX WORDS: C0C8
- CONSTANTS: 3
- LOCAL VARIABLES: 16
- TEMPS: 11
- TOTAL PROGRAM: 215 C0CD
1. **SLERALTINE** C88RR(X,Y,RLONG,FLAT,IST,UTST)
2. **CHANGES LAT AND LONG TO TRANSVERSE MERCATOR AND VICA VERSA**
3. **VERSION OF JUNE 1972**
4. **MODIFIED BY BRUCE SIMBA**
5. IN=105
6. IOLT=108
7. IF(IIST)=2*2/15
8. IF(IIST)=1/1*102
9. 1 CONTINUE
10. ITST=1
11. E2=76*2+5799*2
12. DTR=3*14+152+663*180
13. RTD=1*2/DTR
14. ATRCD=360C+RTD*3C+713114
15. AD=37806C+4
16. AD=11132089
17. A*AD/DTR
18. B=16216+944
19. C1=2*10397
20. D=0*2273
21. E=E*CC033
22. G1=125*23932E=10
23. AS=4*813681E=5
24. 101 CONTINUE
25. READ(IIN=66C1) CME,CMM,RLONG,RLAT,ISRM,XZ,YZ
27. WRITE(110UT,607) CME,CMM,RLONG,RLAT,ISRM,XZ,YZ
28. 6C7 FORMAT(C'M1,F4.0', 'DEG',',F6.3','MINS',',P2','F4.0','DEG',',F6.3','MINS',',P2','F4.0')
29. ISR=1/161 XZ*1/F6.0 YZ*1/F6.0
30. XZ*1/1000 YZ*1/1000
31. P2*P2+P2/66
32. CMM=CMM/66
33. CMM=CMM/66
34. CMM=CMM/66
35. RPZ=RPZ+DTR
36. IF(ISR)=5/6/5
37. R=FLBAT(ISR=1)/FLBAT(ISR)
38. GO TO 7
39. 5 CONTINUE
40. ELB=A+RPZ+BSIN(Z+2*RPZ)+C*SIN(4*RPZ)+D*SIN(6*RPZ)
41. 1+E*SIN(8*RPZ)
42. ELB=ELB+R
43. R=AD=56+60*CSS(2*RPZ)+1*2+CSS(4*RPZ)
44. RPZ=RPZ+DTR
45. RETURN
46. 102 CONTINUE
47. IFLG=2
48. X*X*1000
49. Y*Y*1000
50. XP=X*X
51. YP=Y*Y
52. P3=RPZ+YP/RM
53. RETURN
54. 10 CONTINUE
55. ELA=*A+P1*BSIN(Z+2*RP1)+C*SIN(4*RP1)+D*SIN(6*RP1)
56. 1+E*SIN(8*RP1)+R
57. CP=ARC9D/R/SGRT((1-E2*(SIN(P1)))**1)**3
58. YN=ELA-ELB
59. CPA=(YP-YA)/DY
P1=P1+DPN
IFLG=IFLG-1
IF(IFLG) 11 11 10
CONTINUE
11

CC
P1 IS NOW THE TABULAR LATITUDE
CC
T=(1+*E2(SIN(P1))**2)
C=TAN(P1)**2/G1

C
WE ARE NOW CALCULATING THE RHO Z FACTOR
SG1=(xF*(xP**3)*T/R)**2/2+242+4694**E+12/R
CELP=((SG1)**2*C/3600+1)*CTR
P1=F1+CELP

P1 IS NOW THE TRUE LATITUDE
T=(1+*E2(SIN(P1))**2)
RN=AA/SQRT(T)
DELL1=SIN(SG1/RN)/COS(P1)
DELLR=AR SIN(DELL1)
RLO NG=RDELLR+RC M

R LAT=P1
X*X/1000*
Y*Y/1000*
RETURN

DELLR=RC M+RLO NG
DELL1=SIN(DELLR)
P1=R LAT

F2=P1
T=(1+*E2(SIN(P2))**2
RN=AA/SQRT(T)
SG1=RA*AR SIN(COS(P2)*DELL1)
C8=88 I=1/3
C=TAN(P2)**2/G1
CELP=((SG1)**2*C/3600+1)*DTR

P2=P1+CELP
P1=P2

ELN=((A+P1)*8*SIN(4*P1)+C*SIN(4*P1)+D*SIN(6*P1)
Y=ELN-EL0+Y2
SG=SG1

T=(1+*E2(SIN(P1))**2
xF=SG**3*(T/R)**2/242+4694**E+12
X=X*P1+X
X*X/1000*
Y*Y/1000*
RETURN

END
EXTERNAL SUBPROGRAMS REQUIRED:

F1101  F1102  F1103  F1104  F1105  F1106  F1108  9ASIN  9SORT
9BCDREAD  9BCDDRWIT  9CBS  918CATA  9178R  9SETUPN  9SIN  9SORT

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS GENERATED CODE</td>
<td>457</td>
<td>C01C9</td>
</tr>
<tr>
<td>WORDS CONSTANTS</td>
<td>23</td>
<td>C0017</td>
</tr>
<tr>
<td>WORDS LOCAL VARIABLES</td>
<td>47</td>
<td>C002F</td>
</tr>
<tr>
<td>WORDS TEMPS</td>
<td>11</td>
<td>C0008</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>538</td>
<td>C021A</td>
</tr>
</tbody>
</table>
SUBROUTINE CISAZ(ALAT,ALON,BLAT,BLON,IFRD,AZMTH,BAKAZ,DISKM,DISOG)

C THIS SUBROUTINE CALCULATES THE DISTANCE FROM a POINT A TO A POINT B
C AN ALSO THE AZIMUTH CLOCKWISE FOR THE NORTH FROM A TO B
C ALAT AND ALON ARE THE COORDINATES OF POINT A
C BLAT AND BLON ARE THE COORDINATES OF POINT B
C
REMEmBER:!!
THE POSITION COORDINATES
  - IS SOUTH AND WEST AND + IS NORTH AND EAST
C
IF IRED IS AN OPTION TO INPUT EITHER DEGREES OR RADIANS FOR THE
C IEEE 1984 COMPUTER BY JOHN FAIRBURN; IT HAS BEEN REVISED FOR THE
C TH IBM 360 AT MIT BY JACK WOLFE. NOW IT HAS BEEN REVISED FOR THE
C SIGMA -7 COMPUTER AT NASA HABLE (BY JACK WOLFE) THE INEFFICIENCY
C OF STORAGE ALLOCATION IS DUE TO THE MOST RECENT PROGRAMMERS
C (JACK WOLFE) LINES AND NEGLECT TO CLEAN UP ALL THE JUNK
C
DIMENSION TH(2),PHI(2),DIST(2),AZ(2),AZINV(2)

I=1
K=2
TH(1)=ALAT
TH(2)=BLAT
PHI(1)=ALON
PHI(2)=BLON
IF(IFRD) 3C,31/30

30 CONTINUE
THG = ATAN(+99328*TAN(TH(K)))
C = SIN(PHI(K))
E = -COS(PHI(K))
F = -COS(THG)
A = F*E
B = C*SIN(THG)*D
C = SIN(THG)
G = =E
H = C*E
THG = ATAN(+99328*TAN(TH(1)))
C1 = SIN(PHI(1))
E1 = -COS(PHI(1))
F1 = -COS(THG)
C1 = SIN(THG)
A1 = F1*E1
B1 = -F1*C1
G1 = -C1*E1
F1 = C1*D1
SC = A1*B1 + C*C1
SD = SQRT((A*A1)**2 + (B*B1)**2 + (C*C1)**2) * (A+A1)**2 + (B+B1)**2)
1**2 + (C+C1)**2/4/0
XDEG(1) = ATAN(SC/SC)*67*2957795
IF (SC) 51,52/53

53 IF (SC) 1,2,3
54 1 XDEG (1) = XDEG (1) + 180*C
55 3 IF (SS) 3,4/5
56 2 IF (SS) 3,4/5
57 A2(1) = ATAN(SS/SC)*67*2957795
58 3 IF (SS) 3,4/5
59 3 IF (SC) 6, 7, 8
60 6  AZ(I) = AZ(I) + 180°0 0430
61 6  G0 T6 4 0440
62 7  AZ(I) = AZ(I) + 360°0 0450
63 7  G0 T6 4 0460
64 5  IF (SC) 8, 4 4 0470
65 8  AZ(I) = AZ(I) + 180°0 0480
66 6  SS = (4A+G1)*2 + (B=E1)*2 + C*2 = 2*0 0490
67 7  SC = (4A+G1)*2 + (B=H1)*2 + (C+1)*2=2*0 0500
68 8  AZINV(I) = ATAN(SS/SC) + 2*95779 0510
69 13  IF (SS) 13 14 5 15 0520
70 16  IF (SC) 16 17 17 0530
71 16  AZINV(I) = AZINV(I) + 180°0 0540
72 7  G0 T6 14 0550
73 7  AZINV(I) = AZINV(I) + 360°0 0560
74 7  G0 T6 14 0570
75 15  IF (SC) 18 14 14 0580
76 18  AZINV(I) = AZINV(I) + 180°0 0590
77 14  EL = +13726770C2 + 393372732E + 03 0600
78 7  EC = +13726770C2 + 02 0610
79 1 1 1C + EL 0620
80 80  AL = TAN(T+H(I))/((1+TAN(T+H(I)))) + EC * SQR((T+H(I)) + 2)
81 88  B = PH(I) - PH(I) 0640
82 12  A12 = ATAN(SIN(4L)/((AL + COS(4L))) + SIN(T+(H(I)))) 0660
83 84  ED = EL1((COS(T+H(I)) + COS(A12))*2 + (SIN(T+(H(I))))*2) 0670
85 85  ED2 = ED2 = ED2 0680
86 86  ED3 = ED3 0690
87 87  C0 = 10 + ED/4C*0 - 30 + ED2/64 + ED2 0700
88 88  C2 = ED3 + ED3E + ED3 + ED3 + ED3 + ED3 + ED3 0710
89 89  C4 = ED2 + ED2 + ED2 + ED2 + ED2 + ED2 + ED2 0720
90 90  TH2 = T+(H(I)) + 2E 0730
91 91  TH2 = T+(H(I)) + 2E 0740
92 92  VI = EXP((133C289 E*01 + (380544 E*01 + (732368 E=3)*CAS(TH2)) 0760
93 1 1 + ((175 E*0) + CAS(20 + TH2) = (17*9 + CAS(30 + TH2)) 0770
94 94  V2 = EXP((133C289 E*01 + (380544 E*01 + (732368 E=3)*CAS(TH2)) 0770
95 1 1 + ((175 E*0) + CAS(20 + TH2) = (17*9 + CAS(30 + TH2)) 0780
96 96  Z1 = V1 + (100 + EC) SIN(T+(H(I))) 0790
97 97  Z2 = V2 + (100 + EC) SIN(T+(H(I))) 0800
98 98  Y2 = V2 + (100 + EC) COS(T+(H(I))) 0810
99 99  Y2 = V2 + (100 + EC) COS(T+(H(I))) 0820
100 100  L1 = ATAN(TAN(T+H(I)))*SQR((10*E)*E + CAS(A12)) 0830
101 101  L2 = ATAN((10*E)*E + CAS(A12)) + (10*E)*E + L2 0840
102 102  L2 = ATAN((10*E)*E + CAS(A12)) + (10*E)*E + L2 0850
103 103  B0 = V1*SQR((10*E)*E + CAS(A12)) + (10*E)*E + L2 0860
104 104  DIST1 = (B0 = (U2 + U1)*C2 + (SIN(U2 + U1)*C2 + (SIN(U2 + U1)) + C4*(SIN(U2 + U1)) 0870
105 105  DIST1 = (B0 = (U2 + U1)*C2 + (SIN(U2 + U1)) + C4*(SIN(U2 + U1)) 0880
106 106  DIST1 = ABS(DIST1); 0890
107 107  TEST = DIST1 - 111*0*XECG(I) 0900
108 108  IF (ABS(TEST) > 100 + C) 25 + 30 + 130 + 130 0910
109 361  L2 = U2 + 151592605E + 01 0920
110 110  DIST1 = (B0 = (U2 + U1)*C2 + (SIN(U2 + U1)) = SIN(U2 + U1)) + C4*(SIN(U2 + U1)) + C4*(SIN(U2 + U1)) 0930
111 111  DIST1 = (B0 = (U2 + U1)*C2 + (SIN(U2 + U1)) + C4*(SIN(U2 + U1)) + C4*(SIN(U2 + U1)) 0940
112 112  CONTINUE 0950
113 113  C = BAKAZ = AZINV(I) 0960
114 114  RETURN 0970
115 115  END 0980

LOCAL SUBPROGRAMS DEFINED:

INTRINSIC SUBPROGRAMS USED:

EXTERNAL SUBPROGRAMS REQUIRED:

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>651</td>
<td>0028B</td>
</tr>
<tr>
<td>28</td>
<td>001C</td>
</tr>
<tr>
<td>59</td>
<td>003B</td>
</tr>
<tr>
<td>20</td>
<td>0014</td>
</tr>
<tr>
<td>758</td>
<td>002F6</td>
</tr>
</tbody>
</table>
FUNCTION DMTOR (KD, AM)

FUNCTION TO CONVERT DEGREES AND MINUTES TO RADIANS

NOTE! IF BOTH KD AND AM DO NOT HAVE SAME SIGN AND KD IS NOT C, THEN RADIANS WILL BE WRONG

115

DMTOR = KD
A = DMTOR + (AM/60.0)
CMTOR = A * 1.745329E+2
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALR</td>
<td>CCCC1</td>
<td>V</td>
<td>1</td>
<td>AM</td>
<td>R</td>
<td>SCALR</td>
<td>CCCCC3</td>
<td>V DUMMY</td>
<td>KD</td>
<td>I</td>
<td>SCALR</td>
<td>CCCCC2</td>
</tr>
<tr>
<td>DMTOR</td>
<td>R</td>
<td>SCALR</td>
<td>CCCCC</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (2 WORDS):

CCCC CMTOR 00001 A

BLANK COMMON (0 WORDS)

ENTRY POINTS:

CCCC CMTOR

EXTERNAL SUBPROGRAMS REQUIRED:

SITOR SSETUP2

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>CCCC1</td>
</tr>
<tr>
<td>2</td>
<td>CCCC2</td>
</tr>
<tr>
<td>2</td>
<td>CCCC2</td>
</tr>
<tr>
<td>3</td>
<td>CCCC3</td>
</tr>
<tr>
<td>23</td>
<td>CCCC17</td>
</tr>
</tbody>
</table>

GENERATED CODE: 16
CONSTANTS: 2
LOCAL VARIABLES: 2
TEMPS: 3
TOTAL PROGRAM: 23
SUBRALTIME_DNAV(DLAT, KSN, DLON, KME, RLAT, RLONG, KK)

C VERSION 3 FEB 1974, CORRECT KEYPLNCH ERROR

C SUBRALTIME_DNAV, CONVERTS ANOTATED DECIMAL DEGREES TO RADIANS

C IF KK=0

C CONVERTS RADIANS TO ANOTATED DECIMAL DEGREES

C IF KK=-1

C ANOTATION IS N, S, W, OR E.

DEGPA = 1.745329E-2
RADI = 67.29576
JE=1HA
JS=1HS
KL=1H

IF (KK) 10C, 50, 50

C CONVERT DECIMAL DEGREES TO RADIANS

50 RLAT=DLAT*DEGRA
RLONG=DLON*DEGRA

C CONVTLT LATITUDE

70 RLAT=RLAT
75 IF (KKE=JK) 85, 80, 85

C WEST LONGITUDE

80 RLONG = -RLONG
85 CONTINUE

C CONVERT RADIANS TO ANOTATED DECIMAL DEGREES

10C ALAT = ABS(RLAT)
ALONG = ABS(RLONG)

11C CLAT = ALAT*RADI

C ALAT*RADI

33 IF (RLAT) 1442, 432, 432
43C KSN = JS
45 GO TO 435
46 CJKE = JK
47 IF (RLONG) 442, 442, 442
48 C WARK
49 GO TO 445
40 C RETURN
41 C END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>R</td>
<td>SPRG</td>
<td></td>
<td></td>
<td></td>
<td>DEGRA</td>
<td>R</td>
<td>SCALR</td>
<td></td>
<td></td>
<td></td>
<td>DNAV</td>
<td>R</td>
<td>SCALR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNAV</td>
<td>R</td>
<td>SCALR</td>
<td>00002</td>
<td>V</td>
<td>1</td>
<td>ALAT</td>
<td>R</td>
<td>SCALR</td>
<td>00008</td>
<td>V</td>
<td>1</td>
<td>DLA</td>
<td>R</td>
<td>SCALR</td>
<td>00008</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>WS</td>
<td>I</td>
<td>SCALR</td>
<td>00006</td>
<td>V</td>
<td>1</td>
<td>JA</td>
<td>I</td>
<td>SCALR</td>
<td>00006</td>
<td>V</td>
<td>1</td>
<td>WW</td>
<td>I</td>
<td>SCALR</td>
<td>00007</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>RLAT</td>
<td>R</td>
<td>SCALR</td>
<td>*0000F</td>
<td>V</td>
<td>DUMMY</td>
<td>KHE</td>
<td>I</td>
<td>SCALR</td>
<td>*0000E</td>
<td>V</td>
<td>DUMMY</td>
<td>RLONG</td>
<td>R</td>
<td>SCALR</td>
<td>*00010</td>
<td>V</td>
<td>DUMMY</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (11 WORDS):

- 0000C CCNAV
- 00001 IFKK
- 00006 WE
- 00007 WM
- 00002 DEGRA
- 00003 RADEG
- 00004 JN
- 00005 JS

Blank common (0 Words)

ENTRY POINTS:
- 0000C CCNAV

INTRINSIC SUBPROGRAMS USED:
- ABS

EXTERNAL SUBPROGRAMS REQUIRED:
- 9SETUP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

- DEC WORDS
- HEX WORDS

- GENERATED CODE: 70
- CONSTANTS: 6
- LOCAL VARIABLES: 11
- TEMPS: 8
- TOTAL PROGRAM: 95
SUBROUTINE DREC(VN, VE, RLAT, RLONG, TDIFF)

CC THIS SUBROUTINE TAKES A POSITION (RLAT, RLONG)
AND DR S USING VELOCITIES (VN, VE), AND TIME
DIFFERENCE (TDIFF IN HOURS) TO A NEW POSITION
WHICH IS STORED IN (RLAT, RLONG)
WRITTEN BY A. FOLINSBEE

USES: REARR

R = REARR(RLAT)
SPECK = +5144444
TSEC = TDIFF*3600.

RLAT = RLAT+(VN*SPECK*TSEC/R)
BLAT = ABS(RLAT)
RLONG = RLONG+VE*SPECK*TSEC/(R*COS(BLAT))

RETURN
END
**Local Variables (in):**

- **Name** | **Type** | **Class** | **Dec** | **Abs**
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC

**ENTRY POINTS:**

- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC

**Intrinsic Subprograms Used:**

- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC

**External Subprograms Required:**

- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC
- DEC   | DEC   | DEC   | DEC   | DEC

**Highest Error Severity:** 0 (No Errors)

**Total Program Size:** 35000
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>R</td>
<td>SPROG</td>
<td>INTRIN</td>
<td></td>
<td></td>
<td></td>
<td>BLAT</td>
<td>R</td>
<td>SCALR</td>
<td>000000</td>
<td>V</td>
<td>1</td>
<td></td>
<td>CBS</td>
<td>R</td>
<td>SPROG</td>
<td>INTRIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DREC</td>
<td>R</td>
<td>SCALR</td>
<td>00000</td>
<td>V</td>
<td>1</td>
<td></td>
<td>DREC</td>
<td>R</td>
<td>PROG</td>
<td>00000</td>
<td>P</td>
<td></td>
<td></td>
<td>R</td>
<td>R</td>
<td>SCALR</td>
<td>00001</td>
<td>V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>REARM</td>
<td>R</td>
<td>SPROG</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td></td>
<td>BLAT</td>
<td>R</td>
<td>SCALR</td>
<td>*00007</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td>RLONG</td>
<td>R</td>
<td>SCALR</td>
<td>*00008</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
</tr>
<tr>
<td>SPECK</td>
<td>R</td>
<td>SCALR</td>
<td>00002</td>
<td>V</td>
<td>1</td>
<td></td>
<td>TDF</td>
<td>R</td>
<td>SCALR</td>
<td>*00009</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td>TSEC</td>
<td>R</td>
<td>SCALR</td>
<td>00003</td>
<td>V</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>VE</td>
<td>R</td>
<td>SCALR</td>
<td>*00006</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td>VNM</td>
<td>R</td>
<td>SCALR</td>
<td>*00005</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Local Variables (5 Words):**
- 00000 DREC
- 00001 R
- 00002 SPECK
- 00003 TSEC
- 00004 BLAT

**Blank Common (0 Words)**

**Entry Points:**
- 00000 DREC

**Intrinsic Subprograms Used:**
- ABS
- CBS

**External Subprograms Required:**
- REARM
- 9CO3
- 9SETUP

**Highest Error Severity:** 0 (No Errors)

<table>
<thead>
<tr>
<th>DEC</th>
<th>WORDS</th>
<th>HEX</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>00004</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00002</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00005</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00007</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>00032</td>
<td></td>
</tr>
</tbody>
</table>

**Total Program:** 56
SUBROUTINE CY2M(ID, IY, MB, IDAY)
C CY2M TAKES CONSECUTIVE DAYS AND THE YEAR AND CHANGES THEM INTO DAYS & MONTHS
DIMENSION MYDAY(13)
DATA MYDAY/1,32,60,91,121,152,182,213,244,274,305,335,365/
C DETERMINE IF LEAP YEAR
A = IY
B = IY/4
A = A / 4
IF (A = B) 12 * 10 * 12
LEAP = 1
G0 TO 13
LEAP = 0
CO 14 I = 3 * 13
MYDAY(I) = MYDAY(I) + LEAP
CONTINUE
CO 15 I = 1 * 12
IF (MYDAY(I) + 1 - ID) 15, 16, 16
MO = I
IDAY = ID - MYDAY(I) + 1
G0 TO 17
CONTINUE
CONTINUE
C RETURN MO = 0 AND IDAY = 0 IF ID GT 365 + LEAP
MO = C
IDAY = 0
CO 18 I = 3 * 13
MYDAY(I) = MYDAY(I) + LEAP
CONTINUE
RETURN
END
LOCAL VARIABLES (18 WORDS):

COCC C2M  COCC1 MYDAY  COCC6 A  COCC0 F B  COCC0 I LEAP  COCC11 I

BLANK COMMON (0 WORDS)

ENTRY POINTS:

COCC C2M

EXTERNAL SUBPROGRAMS REQUIRED:

SITOR  SSETUP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC  HEX
WORDS  WORDS

GENERATED CODE: 64 COCC4 C
CONSTANTS: 2 COCC0 2
LOCAL VARIABLES: 18 COCC1 2
TEMPs: 5 COCC0 5
TOTAL PROGRAM: 89 COCC8 9
SUBROUTINE ENDLT(ICNT, CLATB, CLAB8, DL8LE, DLBR1, IDL, I11, IAR)
VERSION 8F 10 MAY 72
BUTPUL '1ENDLT 8F 10 MAY 72'
BUTPUL CLATB, CLAB8, DL8LE, DLBR1

C-----
DIMENSION IA(2C), IB(S)
CATA ITERI, IBL/ITEITP/, 1/

C-----
IIN=105
ICNT=1
RADEG = 57.29578

C-----
IF (IAR.EQ.C) G8T8100

CALL ARLIM(IIN, IBLT, CLATB, CLAB8, DL8LE, DLBR1)

CLA8 = DLATB.*RADEG
CLAB8 = DLAB8.*RADEG
DL8LE = DL8LE.*RADEG
DLBR1 = DLBR1.*RADEG

LA101 = LAT8P/10 + LA102/10
LA103 = LA101/10

C 1CC CONTINUE

C-----
ALAT8P = CLATB + 90 + LAT8P = ALAT8P
ALAB8T = CLAB8 + 90 + LAB8T = ALAB8T
AL8LE = DL8LE + 18C + L8LE = AL8LE
ALBR1 = DLBR1 + 18C + LBR1 = ALBR1

LA101 = LAT8P/10 + LA102/10
LA103 = LA101/10

L0101 = L8LE/10 + L0102 = L8RI/10

C-----
C8101 = LA101 + LA102/10
C8102 = L0102 + L0101/10
C8103 = 90 + 90 + 1
C8104 = L8RI + 10 + L8RI + 10 + L
IF ((LA8P.LT.L8RI) .OR. (LA8P.GT.LAT8P)) G8T8104
IF ((LB8P.LT.L8LE) .OR. (LB8P.GT.LBR1)) G8T8104
WRITE (115, (105)) LAC, LBC

103 FORMAT(2I3)
104 CONTINUE
105 CONTINUE
106 CONTINUE
107 CONTINUE
108 CONTINUE

C-----
C2 CONTINUE

C42 FORMAT(2I4)
C43 READ(IIN, 1) IA
C44 1 FORMAT(20A4)

C-----
IF (IA(1).EQ.ITERI) WRITE (11, 12) (IA(I), I=1, 5) ; G8T83
IF (IA(1).EQ.IBL) ICNT = ICNT+1
WRITE (ICNT, 12) (IA(I), I=1, 5) ; G8T82

C-----
C8101 = 14
C8102 = 5*(I-1)+1
C8103 = 5*(I-1)+1
C8104 = 5*(I-1)+1
C8105 = 5*(I-1)+1
C8106 = 5*(I-1)+1

C8107 = 15
C8108 = 16

11 CONTINUE
WRITE (ICL, 12) IB
12 FORMAT(2A4)
13 CONTINUE
66:       GOTO2
61:        C -----
62:        3 REWIND 1DL & REWIND 1LI & RETURN
63:        END
### LOCAL VARIABLES (20 WORDS):

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>COCC0</td>
<td>ENLT</td>
<td>COCCI</td>
<td>IA</td>
<td>0015</td>
<td>IB</td>
<td>COC1A</td>
<td>ITERI</td>
<td>0001</td>
<td>IB</td>
<td>COCC1</td>
<td>IB</td>
<td>COCC1</td>
<td>IB</td>
<td>0001</td>
</tr>
<tr>
<td>COCC1</td>
<td>RADEG</td>
<td>COCC1</td>
<td>IBUT</td>
<td>0016</td>
<td>ALETP</td>
<td>COCC2</td>
<td>LATEB</td>
<td>0022</td>
<td>ALETP</td>
<td>COCC2</td>
<td>LATEB</td>
<td>COCC2</td>
<td>LATEB</td>
<td>0022</td>
</tr>
<tr>
<td>COCC2</td>
<td>ALETP</td>
<td>COCC2</td>
<td>LATEB</td>
<td>0025</td>
<td>ALETP</td>
<td>COCC3</td>
<td>ALORI</td>
<td>0025</td>
<td>ALETP</td>
<td>COCC3</td>
<td>ALORI</td>
<td>COCC3</td>
<td>ALORI</td>
<td>0025</td>
</tr>
<tr>
<td>COCC3</td>
<td>LORI</td>
<td>COCC3</td>
<td>LORI</td>
<td>0026</td>
<td>LORI</td>
<td>COCC4</td>
<td>LORI</td>
<td>0026</td>
<td>LORI</td>
<td>COCC4</td>
<td>LORI</td>
<td>LORI</td>
<td>0026</td>
<td>LORI</td>
</tr>
<tr>
<td>COCC4</td>
<td>LORI</td>
<td>COCC4</td>
<td>LORI</td>
<td>0027</td>
<td>LORI</td>
<td>COCC5</td>
<td>LORI</td>
<td>0027</td>
<td>LORI</td>
<td>COCC5</td>
<td>LORI</td>
<td>LORI</td>
<td>0027</td>
<td>LORI</td>
</tr>
<tr>
<td>COCC5</td>
<td>LORI</td>
<td>COCC5</td>
<td>LORI</td>
<td>0028</td>
<td>LORI</td>
<td>COCC6</td>
<td>LORI</td>
<td>0028</td>
<td>LORI</td>
<td>COCC6</td>
<td>LORI</td>
<td>LORI</td>
<td>0028</td>
<td>LORI</td>
</tr>
<tr>
<td>COCC6</td>
<td>LORI</td>
<td>COCC6</td>
<td>LORI</td>
<td>0029</td>
<td>LORI</td>
<td>COCC7</td>
<td>LORI</td>
<td>0029</td>
<td>LORI</td>
<td>COCC7</td>
<td>LORI</td>
<td>LORI</td>
<td>0029</td>
<td>LORI</td>
</tr>
<tr>
<td>COCC7</td>
<td>LORI</td>
<td>COCC7</td>
<td>LORI</td>
<td>0030</td>
<td>LORI</td>
<td>COCC8</td>
<td>LORI</td>
<td>0030</td>
<td>LORI</td>
<td>COCC8</td>
<td>LORI</td>
<td>LORI</td>
<td>0030</td>
<td>LORI</td>
</tr>
<tr>
<td>COCC8</td>
<td>LORI</td>
<td>COCC8</td>
<td>LORI</td>
<td>0031</td>
<td>LORI</td>
<td>COCC9</td>
<td>LORI</td>
<td>0031</td>
<td>LORI</td>
<td>COCC9</td>
<td>LORI</td>
<td>LORI</td>
<td>0031</td>
<td>LORI</td>
</tr>
</tbody>
</table>

### ENTRY POINTS:

COCC0 ENLT

### EXTERNAL SUBPROGRAMS REQUIRED:

- ARDUP: F:101
- GA: F:102
- GAC: F:103
- GAC: F:104
- GAC: F:105
- GAC: F:106
- GAC: F:108

### HIGHEST ERROR SEVERITY: 0 (NO ERRORS)
SUBROUTINE EVIL (IOUT, IBAD, KDA, KMB, KYR, KHM)
C SUBROUTINE EVIL CHECKS STATUS INDICATOR, WRITES ERROR MESSAGES,
AND RETURNS INDICATOR FOR BAD READS (IBAD)
C IBAD=0, ALL OK
CC
CC
IBAD=0
GO TO (11C, 120, 130, 14C, 150, 160), 1
12C WRITE (IOUT, 122) KDA, KMB, KYR, KHM
122 FORMAT ('EBF FOUND',3I3I5)
CS PAUSE 122
IBAD=2
GO TO 110
13C WRITE (IOUT, 132) KDA, KMB, KYR, KHM
132 FORMAT ('EBF FOUND',3I3I5)
CS PAUSE 132
IBAD=2
GO TO 110
14C WRITE (IOUT, 142) KDA, KMB, KYR, KHM
142 FORMAT ('PARITY ER',3I3I5)
1BAC=1
GO TO 110
22C FORMAT ('FMT ER',3I3I5)
IBAD=1
GO TO 110
26C WRITE (IOUT, 162) I, KDA, KMB, KYR, KHM
262 FORMAT ('ER I=',12,3I3I5)
IBAD=1
11C RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>HEX</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>HEX</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>HEX</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVIL</td>
<td>SPRG</td>
<td>CCCC F</td>
<td></td>
<td>EVIL</td>
<td>R SCALR</td>
<td>CCCC V</td>
<td></td>
<td>I</td>
<td>I SCALR</td>
<td>#00002 V DUMMY</td>
<td></td>
</tr>
<tr>
<td>IBAC</td>
<td>I SCALR</td>
<td>*CCCC3 V DLPMY</td>
<td></td>
<td>JIBLT</td>
<td>I SCALR</td>
<td>*CCCC1 V DUMMY</td>
<td></td>
<td>KDA</td>
<td>I SCALR</td>
<td>#00004 V DUMMY</td>
<td></td>
</tr>
<tr>
<td>KMR</td>
<td>I SCALR</td>
<td>*CCCC7 V DLPMY</td>
<td></td>
<td>KM6</td>
<td>I SCALR</td>
<td>*CCCC5 V DUMMY</td>
<td></td>
<td>KVR</td>
<td>I SCALR</td>
<td>#00006 V DUMMY</td>
<td></td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (1 WORD):**

CCCCC EVIL

**BLANK COMMON (0 WORDS)**

**ENTRY POINTS:**

CCCCC EVIL

**EXTERNAL SUBPROGRAMS REQUIRED:**


**HIGHEST ERROR SEVERITY:** 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE:</td>
<td>105 CCCC6C</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>0 CCCC00</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>1 CCCC1</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>8 CCCC03</td>
</tr>
</tbody>
</table>

**TOTAL PROGRAM:** 118 CCCC76
SUBROUTINE EXTD (CX, CXF, CY, CYF, EX, EY, IND)

IND = C
D = ((CX - CXF)**2) + ((CY - CYF)**2)
A = ((CX - EX)**2) + ((CY - EY)**2)
B = ((CXF - EX)**2) + ((CYF - EY)**2)

IF (A = D) GOTO 20, 20, 10
IND = 1
RETURN

IF (B = D) GOTO 30, 30, 25
IND = 1
RETURN

END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALR</td>
<td>CCC02 V</td>
<td>1</td>
<td>B</td>
<td>R</td>
<td>SCALR</td>
<td>CCC03 V</td>
<td>1</td>
<td>EX</td>
<td>R</td>
<td>SCALR</td>
<td>*00008 V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>BY</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC09 V</td>
<td>CLMPY</td>
<td>CX</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC04 V</td>
<td>DUMMY</td>
<td>CXP</td>
<td>R</td>
<td>SCALR</td>
<td>*00005 V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>CY</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC06 V</td>
<td>CLMPY</td>
<td>CYP</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC07 V</td>
<td>DUMMY</td>
<td>D</td>
<td>R</td>
<td>SCALR</td>
<td>00001 V</td>
<td>1</td>
</tr>
<tr>
<td>EXTC</td>
<td>R</td>
<td>SCALR</td>
<td>CCC00 V</td>
<td>1</td>
<td>EXT</td>
<td>SPRB</td>
<td>CCC00 P</td>
<td></td>
<td>IND</td>
<td>I</td>
<td>SCALR</td>
<td>*0000A V</td>
<td>DUMMY</td>
<td></td>
</tr>
</tbody>
</table>

**LABEL HEX LBC HEX WORDS**

<table>
<thead>
<tr>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>CCC2A</td>
<td>20</td>
<td>CCC2D</td>
<td>25</td>
<td>CCC3C</td>
<td>30</td>
<td>CCC32</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (4 WORDS):**

- CCC00 EXTC
- 000C1 C
- 000C2 A
- 000C3 B

**BLANK COMMON (C WORDS):**

**ENTRY POINTS:**

- CCC00 EXTC

**EXTERNAL SUBPROGRAMS REQUIRED:**

- SSETLFA

**HIGHEST ERROR SEVERITY:** C (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**GENERATED CODE:** 51

**CONSTANTS:**

- 0

**LOCAL VARIABLES:**

- 4

**TEMP:**

- 5

**TOTAL PROGRAM:** 64
SUBROUTINE FIND(LIMCA, LIMMB, LIMYR, LIMHM, INDCA, INDMB, INYR, INHM, INDIC)

C INDICATES WHETHER INPUT DATE <, =, > LIMIT DATE
C NO COMPARISON REQUIRED
C LIMCA, LIMMB, LIMYR, LIMHM ARE LIMIT DAY, MONTH, YEAR, TIME
C INDCA, INDMB, INYR, INHM ARE INPUT DAY, MONTH, YEAR, TIME
C INDIC IS INDICATOR
C
C IF LIMYR = 99, NO COMPARISON IS MADE
C IF LIMYR = 999, ONLY YEARS ARE COMPARED
C IF LIMCA = 999, ONLY YEARS AND MONTHS ARE COMPARED
C IF LIMHM = 9999, DATES ARE COMPARED BUT TIMES ARE IGNORED
C NOTE: ALL FOUR LIMITS SHOULD BE GIVEN
C
C INDICATOR SETTINGS:
C INDIC = -1 MEANS INPUT DATE IS BEFORE LIMIT DATE
C INDIC = 0 MEANS INPUT DATE IS EQUAL TO LIMIT DATE
C INDIC = +1 MEANS INPUT DATE IS AFTER LIMIT DATE
C
C IF(LIMYR=99)100,300,100
C IF(INYR=LIMYR)199,101,201
C IF(INMB=LIMMB)102,202,102
C IF(INHM=LIMHM)103,203,103
C IF(INCA=LIMCA)104,204,104
C IF(INHM=LIMHM)105,205,105
C IF(INCA=LIMCA)106,206,106
C 199 INDIC = -1
C RETURN
C 200 INDIC = 0
C RETURN
C 201 INDIC = +1
C RETURN
C END
### LOCAL VARIABLES (1 WORD):

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIND</td>
<td>0000F</td>
</tr>
</tbody>
</table>

### BLANK COMMON (0 WORDS)

### ENTRY POINTS:

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIND</td>
<td>0000F</td>
</tr>
</tbody>
</table>

### EXTERNAL SUBPROGRAMS REQUIRED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLAS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIND</td>
<td>SPRING</td>
<td>0000</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>INDIC</td>
<td>SCALAR</td>
<td>0004</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>INTO</td>
<td>SCALAR</td>
<td>0007</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>LIMP</td>
<td>SCALAR</td>
<td>0002</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

### HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### GENERATED CODE:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASSES</th>
<th>DEC LBC</th>
<th>HEX LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIND</td>
<td>SPRING</td>
<td>0000</td>
<td>0000</td>
<td>0000F</td>
</tr>
<tr>
<td>INDIC</td>
<td>SCALAR</td>
<td>0004</td>
<td>0004</td>
<td>00048</td>
</tr>
<tr>
<td>INTO</td>
<td>SCALAR</td>
<td>0007</td>
<td>0007</td>
<td>00010</td>
</tr>
<tr>
<td>LIMP</td>
<td>SCALAR</td>
<td>0002</td>
<td>0002</td>
<td>0002E</td>
</tr>
</tbody>
</table>

### TOTAL PROGRAM: 60 WORDS
SUBROUTINE FLD2(KK,ITAPE,RLAT,RLONG,REG)

VERSION 6 MAY 1974

TO IMPLEMENT MODIFICATIONS BY FOLMSEBE OF 24 APRIL 73

SUBROUTINE FLD2, CALCULATES REGIONAL FREE-AIR
ANOMALIES FROM SPHERICAL HARMONIC COEFFICIENTS
ENTERED AT RUN TIME


DIMENSION CF(25),FM(25),FN(25)

SSW(4) UP TO LIST INTERMEDIATE VALUES
SSW(5) UP TO LIST BINARY

TO CHANGE ORDER OF COEFFICIENTS REPLACE DIMENSION
STATEMENTS BY ORDER + 1, AND SET ADIM = ORDER + 1

A BLANK CARD MUST FOLLOW COEFFS TO INDICATE THEIR COMPLETION

KK*1 FOR INITIAL ENTRY TO FLD2
KK=1 HENCEFORTH

USES F4LIBS FLOAT, SQRT, SIN, COS, ATAN

IF(KK)500,999,500

CONTINUE

NDIM=25

II = 105

IIBLT = 108

REFG=9800000*C

OUTPUT 'FLD2'

WRITE(IIBLT,1)

1 FORMAT (10,11), M, C(N/M), S(N/M)!

ISET=C

MAXN=C

CM X 324 N=1,NDIM

CN X 324 M=1,N

C(N,M)=0.

S(N,M)=0.

CONTINUE

CONTINUE

READ(ITAPE,302,END=30) N,M,CTEMP,STEMP

IF(N) 30,30,25

302 FORMAT (12*2X,2X,E11.4,2X,E11.4)

CM(N+1,M+1)=CTEMP

SN+1,M+1=STEMP

WRITE(IIBLT,3) N,M,C(N+1,M+1),S(N+1,M+1)

FORMAT(1X,20) I5*15*15*2E11.4

IF(N) 325,325,26

55 FORMAT(1X,M=MAXN) 325,325,26

MAXN=M

GO TO 325

CONTINUE

C NDIM SHOULD BE SAME AS SIZE OF DIMENSIONED ARRAYS

MAX1=MAXN+1

C ISET=C INDICATES FIRST CALL TO PROGRAM

REnormalize COEFF IF THIS IS FIRST CALL

IF (ISET) 500,2,500
2 CONTINUE
  ISET=1
  DB 17 N=1, NDIM
  DB 17 M=2, NDIM
  F(N,M)=1*C

17 CONTINUE
  BAR(1,1)=1*C
  DB 20 N=2, NDIM
  F(N)=N
  FM(N)=N-1
  BAR(N,M)=BAR(N-1,M)*FLAT/(2*N-3)/FLOAT(N-1)
  J=2
  DB 20 M=2*N
  BAR(N,M)=BAR(N,M-1)*SGRT(FLAT/(N-1)*J)/FLOAT(N+2)

C C
C BAR(N,M) ARE FACTORS TO RENORMALIZE CJS
C
74  C J=1
78  2C CONTINUE
79    DB 21 N=2*MAX1
80    DB 21 M=1,N

81  C C
C NOW FINISHED WITH BAR, WILL USE LATER TO STORE CONST FOR
C RECURSION RELATION
C
85  C
86  P(1,1)=1*
87    SP(1)=C*
88    CF(1)=1*
89    RAD=672957795E+02
90    A=6378.388
91    FLAT=1.0-1.0*C/297*
93    A2=A**2
94    A4=A**4
95    B2=(A*FLAT)**2
96    A2B2=A2*(1*-FLAT**2)
97    A4E=A4*(1*-FLAT**4)
98  C C
C BAR(2,1)=C*
C BAR(2,2)=C*
100  C
101    DB 24 N=3, NDIM
102    DB 24 M=1,N

103  C 24 CONTINUE
104  RETURN
105  C WE HAVE NOW SET UP MOST CONSTANTS ARRAYS, ETC.
106  C THE PROGRAM COMES TO THIS POINT IF IT HAS BEEN ENTERED
107  C PREVIOUSLY
108  C C
109  50C SINLA=SIN(RLAT)
110   CP(2)=COS(RLON)
111   SP(2)=SIN(RLON)
112    DB 51 N=3, MAX1
113    SP(M)=SP(2)*CP(M-1)+CP(2)*SP(M-1)
114    CP(M)=CP(2)*CP(M-1)-SP(2)*SP(M-1)
115  51 CONTINUE
116  C
117    SINLA2=SINLA**2
118    DEN=A2-A2B2*SINLA2
119    DENS=SGRT(DEN2)
120
120. \text{FAC} = R / A2
121. \text{THETA} = \text{ATAN} (\text{FAC} \times \text{SIN} / (1 \times 30 \times \text{SQR} (1 - \text{SIN} 2)))
122. \text{R} = \text{SQR} ((\text{A}4 \times \text{A}4 \times \text{SIN} 2) / \text{DEN} 2)
123. \text{CT} = \text{SIN} (\text{THETA})
124. \text{ST} = \text{COS} (\text{THETA})
125. \text{AR} = 1 + \text{C}
126. \text{AR} = \text{AR} \times 2
127. \text{RV} = \text{C} + \text{C}
128. \text{CB} = 4 \times \text{X} + \text{MAX} 1
129. \text{AR} = \text{AR} \times \text{AR}
130. \text{CB} = 5 \times \text{X} + \text{Y}
131. \text{IF} (\text{N} \times \text{M}) = 112 \times 111 \times 112
132. \text{F} (\text{N}, \text{M}) = \text{ST} \times \text{F} (\text{N} - 1, \text{M} - 1)
133. \text{GO TO} 113
134. \text{IF} (\text{N} \times \text{M}) = 2011 \times 2012 \times 2011
135. \text{GO TO} 112
136. \text{GP} = 1
137. \text{GO TO} 2012
138. \text{GP} = \text{F} (\text{N} \times \text{M})
139. \text{GO TO} 2010
140. \text{F} (\text{N}, \text{M}) = \text{CT} \times \text{F} (\text{N} - 1, \text{M}) - \text{BAR} (\text{N}, \text{M}) \times \text{CF}
141. \text{FNM} = \text{F} (\text{N}, \text{M}) \times \text{AR}
142. \text{TEF} = \text{C} (\text{N}, \text{M}) \times \text{CF} (\text{M}) + \text{S} (\text{N}, \text{M}) \times \text{SP} (\text{M})
143. \text{BV} = \text{BV} \times \text{TEMP} \times \text{FLBAT} (\text{N} - 2) \times \text{FN}
144. \text{CONTINUE}
145. \text{REG} = \text{BV}
146. \text{REG} = \text{REG} \times \text{REFG}
147. \text{IF} (\text{M} \times \text{M}) = \text{306} \times 306 \times 403
148. \text{WRITE} (\text{IIBOUT} \times \text{G5}, \text{RV} \times \text{COS} C)
149. \text{RD} = \text{MAT} (\text{RV} = 1 \times \text{E} - 1 \times 4 \times 2 \times, \text{COS} C = 1 \times \text{FAC} \times 3)
150. \text{RETURN}
151. \text{END}
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>ADR</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ATAN</td>
<td>R</td>
<td>SPRKG</td>
<td>1</td>
<td>1</td>
<td>A2</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A0</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>A2B</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>BAR</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CMS</td>
<td>R</td>
<td>SPRKG</td>
<td>1</td>
<td>1</td>
<td>C</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>CT</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>CP</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DETA</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>CP</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>F2</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>DEN</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>F22</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>F22</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLAT</td>
<td>R</td>
<td>SPRKG</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FL2</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>SPRKG</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FP</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>I1A</td>
<td>I</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>IS1</td>
<td>I</td>
<td>SPRKG</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ISn</td>
<td>I</td>
<td>SPRKG</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MAX1</td>
<td>I</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MAX1</td>
<td>I</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>XXX</td>
<td>I</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RAC</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>RAC</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>FLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SIN</td>
<td>I</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SIN</td>
<td>I</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SP</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SP</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>STEPP</td>
<td>R</td>
<td>SCALAR</td>
<td>1</td>
<td>1</td>
<td>FLD</td>
<td>R</td>
<td>ARRAY</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX</th>
<th>LOC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LOC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LOC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LOC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C000C</td>
<td>2</td>
<td>2</td>
<td>C007F</td>
<td>5</td>
<td>525</td>
<td>C0286</td>
<td>26</td>
<td>325</td>
<td>C0365</td>
<td>72</td>
<td>325</td>
<td>C0365</td>
<td>72</td>
</tr>
<tr>
<td>3</td>
<td>C000D</td>
<td>26</td>
<td>3</td>
<td>C000B</td>
<td>2</td>
<td>55</td>
<td>C000B</td>
<td>26</td>
<td>325</td>
<td>C000B</td>
<td>2</td>
<td>55</td>
<td>C000B</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>C001A</td>
<td>32</td>
<td>5</td>
<td>C001A</td>
<td>32</td>
<td>55</td>
<td>C000B</td>
<td>26</td>
<td>325</td>
<td>C0036</td>
<td>72</td>
<td>325</td>
<td>C0036</td>
<td>72</td>
</tr>
<tr>
<td>3</td>
<td>C0018</td>
<td>32</td>
<td>5</td>
<td>C0018</td>
<td>32</td>
<td>55</td>
<td>C000B</td>
<td>26</td>
<td>325</td>
<td>C0036</td>
<td>72</td>
<td>325</td>
<td>C0036</td>
<td>72</td>
</tr>
<tr>
<td>599</td>
<td>C0043</td>
<td>2C10</td>
<td>2C11</td>
<td>C0197</td>
<td>2C12</td>
<td>C0197</td>
<td>2C12</td>
<td>C0197</td>
<td>2C11</td>
<td>C0197</td>
<td>2C12</td>
<td>C0197</td>
<td>2C12</td>
<td></td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (2637 WORDS):**

- **C00CC FLC2**
- **C00DE CP**
- **C002C REFQ**
- **C0A2E MAX**
- **C0A3B A2**
- **C0A3E SNA2**
- **C0A4 CT**
- **C0A4A PAY**

**BLANK COPPER (C WORDS)**

**ENTRY POINTS:**

- **C00CC FLC2**

**INTRINSIC SUBPROGRAMS USED:**

- **ATAN, COS, FLOAT, SIN, SORT**
EXTERNAL SUBPROGRAMS REQUIRED:

<table>
<thead>
<tr>
<th>ISH</th>
<th>FS102</th>
<th>FS103</th>
<th>FS104</th>
<th>FS105</th>
<th>FS106</th>
</tr>
</thead>
<tbody>
<tr>
<td>9ATAN</td>
<td>9BCDARIT</td>
<td>9CBS</td>
<td>9ENDBL</td>
<td>9SADATA</td>
<td>9ITOR</td>
</tr>
</tbody>
</table>

HIGHEST ERROR SEVERITY: C (NO ERRORS)

<table>
<thead>
<tr>
<th>Generated Code</th>
<th>Dec</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Words</td>
<td>479</td>
<td>C01DF</td>
</tr>
<tr>
<td>Constants</td>
<td>8</td>
<td>C0008</td>
</tr>
<tr>
<td>Local Variables</td>
<td>2637</td>
<td>C0A4C</td>
</tr>
<tr>
<td>Temps</td>
<td>9</td>
<td>C0009</td>
</tr>
<tr>
<td>Total Program</td>
<td>3133</td>
<td>CCC3C</td>
</tr>
</tbody>
</table>
SUBROUTINE GETC (ITAPE, NX, NY, NZ, DATA, RLAT, RLONG, KGDA, KGMB, KGMR, KGMM, IEBC)

C SUBROUTINE GETC FOR READING MAGNETICS AT CALCM FORMAT

CHANGED 17 JUNE 1971 BY CM WOOGING TO ANNSTATE SAS MAG

C DIMENSION PLT(7)
IEBC=0
IIBL=108
CALL END
READ (ITAPE, 16) KGDA, KGMB, KGMR, KGYR, KGMM, DLAT, DLONG, DATA, B, FLD, DIS, DIR,
16 FORMAT (1X, I2, 1X, 2I2, 1X, F6.1, F7, 1X, F8, 3X, 1X, 2F6, 2X, 1X, 1X),
1F3, 0, F4, 1
CALL STAT(I)
CALL EVC(IICUT, I, IBD, KGDA, KGMB, KGYR, KGMM)
IF (IBAD) 14, 30, 65
IEBC=1
RETURN
30 READ (RLAT, 108) (1*0/57.29578)
READ (RLONG, 108) (1*0/57.29578)
PLT(1) = KGMM
PLT(2) = B
PLT(3) = FLC
PLT(4) = DIS
PLT(5) = DIR
PLT(6) = SPD
PLT(7) = DATA
KGDA = KGDA
KGMB = KGMB
KGYR = KGYR
KGMM = KGMM
SELECT POINT TO BE PLOTTED
IF (NX) 110, 120, 110
110 DATA = PLT(NX)
120 DATA = PLT(NY)
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>R</td>
<td>SCALAR</td>
<td>00010</td>
<td>V 1</td>
<td>DATA</td>
<td>R</td>
<td>SCALAR</td>
<td>00010</td>
<td>V 1</td>
<td>DATA</td>
<td>R</td>
<td>SCALAR</td>
<td>00010</td>
<td>V 1</td>
</tr>
<tr>
<td>DATA</td>
<td>R</td>
<td>SCALAR</td>
<td>00010</td>
<td>V 1</td>
<td>DATAY</td>
<td>R</td>
<td>SCALAR</td>
<td>00010</td>
<td>V 1</td>
<td>DATA</td>
<td>R</td>
<td>SCALAR</td>
<td>00010</td>
<td>V 1</td>
</tr>
<tr>
<td>DIR</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
<td>DIS</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
<td>DLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
</tr>
<tr>
<td>CLR</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
<td>ENC10</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
<td>EVIL</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
</tr>
<tr>
<td>FLG</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
<td>GETC</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
<td>SPR0G</td>
<td>R</td>
<td>SCALAR</td>
<td>00020</td>
<td>V 1</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>IBAC</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>IEDD</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>IIBL</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>ITAPE</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>KGD A</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>KGD</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>KGD</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>KGD</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>KGMD</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>KGMD</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>KGMD</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>KGVR</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>KGVR</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>KGVR</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>N</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>N</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>N</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>N</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>N</td>
<td>I</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>PLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>PLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>PLAT</td>
<td>R</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
<tr>
<td>TOT</td>
<td>R</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>TOT</td>
<td>R</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
<td>TOT</td>
<td>R</td>
<td>SCALAR</td>
<td>00030</td>
<td>V 1</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (23 WORDS):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCCC1</td>
<td>GETC</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>GETC</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>GETC</td>
<td></td>
<td>00217</td>
<td>15</td>
</tr>
<tr>
<td>CCCC1</td>
<td>PLT</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>PLT</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>PLT</td>
<td></td>
<td>00217</td>
<td>15</td>
</tr>
<tr>
<td>CCCC1</td>
<td>DIS</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>DIS</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>DIS</td>
<td></td>
<td>00217</td>
<td>15</td>
</tr>
<tr>
<td>CCCC1</td>
<td>DIR</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>DIR</td>
<td></td>
<td>00217</td>
<td>15</td>
<td>CCCC1</td>
<td>DIR</td>
<td></td>
<td>00217</td>
<td>15</td>
</tr>
</tbody>
</table>

EXTERNAL SUBPROGRAMS REQUIRED:

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDE</td>
<td>E</td>
<td>&quot;E&quot;</td>
<td>00217</td>
<td>15</td>
<td>ENDE</td>
<td>E</td>
<td>&quot;E&quot;</td>
<td>00217</td>
<td>15</td>
<td>ENDE</td>
<td>E</td>
<td>&quot;E&quot;</td>
<td>00217</td>
<td>15</td>
</tr>
<tr>
<td>SITB</td>
<td>S</td>
<td>&quot;S&quot;</td>
<td>00217</td>
<td>15</td>
<td>SITB</td>
<td>S</td>
<td>&quot;S&quot;</td>
<td>00217</td>
<td>15</td>
<td>SITB</td>
<td>S</td>
<td>&quot;S&quot;</td>
<td>00217</td>
<td>15</td>
</tr>
</tbody>
</table>

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)
SUBROUTINE GETF (ITAPE, NX, NY, NZ, NW, CATAX, CATAY, DATAZ, DATAW, IECB)
10. C
11. C
12. C
13. C
14. C
15. C
16. C
17. C
18. C
19. C
20. C
21. C
22. C
23. C
24. C
25. C
26. C
27. C
28. C
29. C
30. C
31. C
32. C
33. C
34. C
35. C
36. C
37. C
38. C
39. C
40. C
41. C
42. C
43. C
44. C
45. C
46. C
47. C
48. C
49. C
50. C
51. C
52. C
53. C
54. C
55. C
56. C

DIMENSION PLT(8)

IEBC = 0
IIBLT = 108
IIBLT = 2
ISE ISW(15)) = 13, 14, 13
PAUSE 15
CALL ENDO
11 READ ITAPE, KGDA, KGMB, KGYR, KGHM, ITDF, LAT,
19. 1 RLATM, LGBK, RL8, K79, K8, K1, K2, K3
21. 1 6 FORMAT (I12, 14, 1X, 13, 1X, 13, 1X, 14, 1X, 14, 1X, 1X, 1X, 1X, 311)
23. CALL STAT(I)
24. CALL EVIL(IIBLT, I, IBAD, KGDA, KGMB, KGYR, KGHM)
25. IF (IBAD) 14, 30, 65
26. IEBD = 1
RETURN
30. IF (ISW(12)) = 55, 60, 55
31. WRITE(IIBLT, 56) KGDA, KGMB, KGYR, KGHM
32. IF (NLFYR) 70, 65, 70
33. CONTINUE
34. C
35. C
36. C
37. C
38. C
39. C
40. C
41. C
42. C
43. C
44. C
45. C
46. C
47. C
48. C
49. C
50. C
51. C
52. C
53. C
54. C
55. C
56. C

SELECTING DATA TO BE PLOTTED
50. IF(NX) = 110, 120, 110
51. 11C CATAX = PLT(NX)
52. 12C CATAY = PLT(NY)
53. CATAZ = PLT(NZ)
54. CATAW = PLT(NW)
55. RETURN
56. END
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9-12</td>
<td>GetG version Feb 15 76. Changed input format for backarc code manage problem. MODIF for new ABSTG format handling.</td>
<td></td>
</tr>
<tr>
<td>15-20</td>
<td>Modif on Nov 12/71 by PANGET U.M. To include: use of data location table, use of label tapes. MBC alg 16/1971 by FOLINSbee to correct error on calling args of GBLKI. Version of June 29/71 does not write EGF back output device.</td>
<td></td>
</tr>
<tr>
<td>21-23</td>
<td>Version of June 29/71 to read (or white) blocked data by a FOLINSbee. Version of April 16/71 to optionally suppress rewind of ITAPE and JTape with complete argument list, ISRC moved to come after date.</td>
<td></td>
</tr>
<tr>
<td>26-27</td>
<td>SUBROUTINE GETG, FOR GELF FORMATTED DATA</td>
<td></td>
</tr>
<tr>
<td>29-34</td>
<td>Version with designation of input magnetic tapes by use of subroutine MOUNT.</td>
<td></td>
</tr>
<tr>
<td>35-37</td>
<td>SSW(12) to list date identification. SSW(27) to suppress rewind of tapes at start of job. SSW(29) = 1 to read and test for selected source code numbers to be processed. SSW(29) = 2 to read and test for selected source code numbers to be skipped.</td>
<td></td>
</tr>
<tr>
<td>38-40</td>
<td>SSW(30) to suppress rewind of tapes at start of job. SSW(31) to suppress rewind of ITAPE. SSW(40) to process with bounds using CLT. SSW(60) to process only data with IFFC=4, abstracted output. SSW(61) to replace FAM, ELEV, LAT, LONG with averaged values.</td>
<td></td>
</tr>
<tr>
<td>41-43</td>
<td>USES ENDIS(DUMMY), EVIL, STAT, ISK. Assume ISK and STAT initialized in MAIN program.</td>
<td></td>
</tr>
<tr>
<td>46-53</td>
<td>DIMENSION IDTIN(20), IBK(20), ITK(20), IDENS(20), IDESC(17,20), IDTIN(20), IBK(20), ITK(20), IDENS(20), IDESC(17,20), EQUVALENCE (CLAP1, DLAP1), (CLAM1, DLAM1), (CLBR1, DLBR1), (CLBR2, DLBR2),</td>
<td></td>
</tr>
<tr>
<td>54-58</td>
<td>DATA ITLM/40/, IA(35), IAFMT(9), IA(35), IAFMT(9), ISRC(16), DATA IFLAG/C.</td>
<td></td>
</tr>
</tbody>
</table>
C IF( IFLAG.IE.0 ) G0T050
C IFLAG=1
C
C ------ GETG INITIALISATION LOGIC
C
C IN=105
C I18LT = 108
C IED0=0
C IDISC = 100
C KI = 1
C KG = -2
C NEF = 1
C FILE = 1
C
C DEGRA = 1.745329E+2
C RADEG=57.29578
C
C BLPUT ' GETG VERSION 25 JULY 75 FOR 67 C FORMULA'
C IF(ISW(40).NE.0) CALL ENCLT(J,DLATB,DLAB,DLAB,DLAB,DLAB)
C * CALL SETSKP(INICDA) J IDLT=0
C * J NEF=1 J IFILE=1 J BLPUT NEF,FILE
C NERB=0
C KGDB=NZERO
C KGFB=NZERO
C KGGB=NZERO
C KGHB=NZERO
C NRECT = NO. OF RECORDS NOW WRITTEN ON PRESENT OUTPUT TAPE
C NEF = NO. OF FILE NOW BEING PROCESSED
C NREC1 = 1
C
C IF(ISW(40).NE.0) G0T0140
C IF(ISW(30).NE.0) G0T0140
C
C READ (IN46) IDTIN(J),IBK(J),ITK(J),IDENS(J),
C (IDESC(K),K=1,5)
C 406 FORMAT(A4,1X,A1,1X,111X,13,17A4)
C IF(IDTIN(J).NE.ITERI) J+1, GO TO 405
C NEF=1
C IFILE=IFILE+1
C BLPUT NEF,FILE
C
C ------ CHECK SSW(29) TO SEE IF SOURCE CODE NUMBERS
C ------ ARE TO BE READ FOR DATA SELECTION
C
C IF(ISW(29).NE.0) G0T0100
C IF(ISW(30).NE.0) G0T0100
C
C READ(IN,9CC)SRC
C 90C FORMAT(161)
C IF(ISW(29).EQ.1) WRITE(IN8LT,912)SRC, G0T0140
C WRITE(IN8LT,913)SRC
C 913 FORMAT(1MO,10X,'SKIPPED SOURCE CODES = ',161)
C 912 FORMAT(1MO,10X,'SELECTED SOURCE CODES = ',161)
C
C ------
C CONTINUE

C CONVERTING FROM THE POTS DAM REF. NO. TO THE IGSN-71 REF. SYSTEM
C
C NX = 11 TO PLOT OBSERVED GRAVITY
C IF (NX = NE - 11) GO TO 81
C IF (IPRC = 2) GO TO 81
C CALL BGC (K977, 6BSG, 6BS, K1)
C GO TO 8BS = 8BS + 140
C CONTINUE

C BGC0M = BG + TC
C PLT(1) = KGMM
C PLT(2) = ISBC
C PLT(3) = ELEV
C PLT(4) = DEPHT
C PLT(5) = HEIGT
C PLT(6) = FA
C PLT(7) = BG
C PLT(8) = TC
C PLT(9) = BGC0M
C PLT(10) = RFA
C PLT(11) = 6BS
C PLT(12) = HEIGT/8G
C IF (IPRC = 60) GO TO 81
C CONTINUE

C IF (IPRC = 61) 1109, 1109, 1105
C SET FA, BG, AND ELEV = AVERAGED VALUES FROM ABSTRACTER OUTPUT
C SET LAT AND LONG TO VALUES AT CENTER OF GRID AREA
C CONTINUE
324 IF(IPRE*EG*NMAX)G0T325
301 IBEG=IPRE+1
302 G0T336
303 325 IBEG=1
304 C ------- NINF = INDEX FIRST REC. TO READ
305 C ------- NSLP = INDEX LAST REC. TO READ
306 .326 NFIR = NMAX*(NBLO+1)
307 NINF=NFIR+IBEG
308 NSUP=NFIR + ILAST
309 C ------- AVOID TRYING TO READ REC. WHICH ARE ALREADY PROCESSED
310 IF(NINF+LT+NFRE)NINF=NPRE
311 NSKIF=NINF+APRE
312 CALL SKFREC(ITAPE,NSKIP,'FWD!')
313 G0T333(330,330,331,332,333)INDICA
314 33C CONTINUE
315 APRE=NSLP+1
316 ICNREC = NSUP-NINF +1 ; ICNT = 0
317 821 IF(ICNT+LT+ICNREC)IDL+1 ; G0T337
318 IDLT=0 ; G0T322
319 87C READ(ITAPE,REC1,ISORC,KGDA,KGMB,KGVR,KGHM,
320 1 DLAT,DLONG,ELEV,K977,BSG,DEP,FAC,TG,IELC,IGC,
321 2 RFA,IREC,IFFC,IA,IFBC,LTKEY,LLKEY,IAKEY
322 ICN = ICNT+1
323 G0T301
324 343 WRITE(I18UT,344)
325 344 FORMAT('TEST AREA ALREADY PROCESSED!'/)
326 IEO=1 ; RETURN
327 331 WRITE(I18UT,345)
328 345 FORMAT('INCORRECT CLT TABLE = FOUND EOF WHILE PROCESSING
329 * RECORDS'/)
330 IEO=1 ; RETURN
331 WRITE(I18UT,346)
332 346 FORMAT('INCORRECT CLT TABLE = FOUND END OF TAPE WHILE
333 * SKIPPING RECORDS'/)
334 IEO=1 ; RETURN
335 WRITE(I18UT,347)
336 347 FORMAT('ERROR CONDITION WHILE SKIPPING RECORDS'/)
337 IEO=1 ; RETURN
338 END
**HIGHEST ERROR SEVERITY:** 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE:</td>
<td>1130</td>
<td>C046A</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>19</td>
<td>C0C13</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>1046</td>
<td>C0C16</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>18</td>
<td>C0C12</td>
</tr>
<tr>
<td>TOTAL PROGRAM:</td>
<td>2213</td>
<td>C08A5</td>
</tr>
</tbody>
</table>
SUBROUTINE GETGA(ITAPE,NX,DATAX,NY,DATAY,NZ,DATAZ,
KTP,PRBT,PLFT,RRGT,RLAT,RLONG,IEBD)

DIMENSION IBUFIN(184)
DIMENSION KAVFA(36),KAVEL(36),KPTS(36)
DIMENSION KCENEL(36),KCENFA(36)
DIMENSION KCENLAT(36),KCENLANG(36)
DATA INIT/0000/
IF(INIT*NE.0) GA TO 200

C INITIALIZATION

"OUTPUT 'SUBROUTINE GETGA VERSION' OF 25 OCT 75"
INIT = 1
IKEY = 105
NIN = 37
IEBD = 0
RADEG = 57.2458
DEG = 1.745382
KDTAP=RTAP+RADEG+200
KDRAT=RRAT+RADEG+200
KDLFT=RLFT+RADEG+200
KDRGT=RRGT+RADEG+200

200 CONTINUE
IF(NIN*LT*37) GA TO 250
CALL BUFFER IN(ITAPE,0,IBUFIN(1),296,IKEY,NI)
210 CONTINUE
GB TO (211,215,213,214) IKEY
211 OUTPUT 'WAITING! GA TO 210'
213 OUTPUT 'ESF GN, ITAPE! GA TO 300'
214 OUTPUT 'ERROR GN INPUT! GA TO 300'
CONTINUE
215 CONTINUE

DECWUL(1184,1001,IBUFIN(1),ND)

KLAT,KLANG,KDUM1,KDUM2,KDUM3,KDUM4,KDUM5,
(KCENEL(1),KCENFA(1),KCENLAT(1),KCENLANG(1))
(KAVEL(1),KAVFA(1),KPTS(1),I=1,36)
NIN = 0
IF((KLAT*LT*KDTAP)*90-(KLAT*LT*KDRAT)) NIN=371 GA TO 200
IF((KLANG*LT*KDLFT)*90-(KLANG*LT*KDRGT)) NIN=371 GA TO 200
250 CONTINUE
GB TO (310,320,330,340,350) (NX*3)
300 CONTINUE

NIN=NIN+1
IF(NIN=EQ.37) GA TO 200
IF(KPTS(NIN)*EQ.0) GA TO 310
DATA*FLBAT(KAFLA(NIN)/10)
450 CONTINUE

C AVERAGE FREE AIR
310 CONTINUE

C AVERAGE ELEVATION
320 CONTINUE

NIN=NIN+1
IF(NIN=EQ.37) GA TO 200
IF(KPTS(NIN)*EQ.0) GA TO 320
DATA*FLBAT(KAFLA(NIN))
510 CONTINUE

C CENTRAL FREE AIR
330 CONTINUE

NIN=NIN+1
IF(NIN=EQ.37) GA TO 200
IF(KPTS(NIN)*EQ.0) GA TO 330
DATA*FLBAT(KCENFA(NIN)/10)
580 CONTINUE

C AVERAGE FREE AIR
CENTRAL ELEVATION

340 CONTINUE
   NIN=NIN+1
   IF(NIN.EQ.37) GO TO 200
   IF(KPTS(NIN).EQ.0) GO TO 340
   DATA=FLAT(KCENEL(NIN))
   GO TO 400

350 CONTINUE
   NIN=NIN+1
   IF(NIN.EQ.37) GO TO 200
   IF(KPTS(NIN).EQ.0) GO TO 350
   DATA=FLAT(KPTS(NIN))

400 CONTINUE
   RLAT=(KLAT+FLAT(KCENLAT(NIN)))/100.*200.*DEGRA
   RLANG=(KLANG+FLAT(KCENLANG(NIN)))/100.*200.*DEGRA
   RETURN

END OF FILE

END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFERIN</td>
<td>SPRG</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td></td>
<td>DATAX</td>
<td>R</td>
<td>SCALR</td>
<td>005B3</td>
<td>V</td>
<td>DUMMY</td>
<td>DATAY</td>
<td>UNUSE</td>
<td>005B5</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
</tr>
<tr>
<td>DATAZ</td>
<td>UNUSED</td>
<td>005B7</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td>GETGA</td>
<td>R</td>
<td>SCALR</td>
<td>005A1</td>
<td>V</td>
<td>1</td>
<td>FLBTA</td>
<td>R</td>
<td>SPRG</td>
<td>INTR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETGA</td>
<td>SPRG</td>
<td>00000</td>
<td>D</td>
<td></td>
<td></td>
<td>IBUFIN</td>
<td>I</td>
<td>ARRAY</td>
<td>00001</td>
<td>V</td>
<td>11B4</td>
<td>IEBD</td>
<td>I</td>
<td>SCALR</td>
<td>005B6</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>INIT</td>
<td>I</td>
<td>ARRAY</td>
<td>00090</td>
<td>V</td>
<td></td>
<td>ITAPE</td>
<td>I</td>
<td>SCALR</td>
<td>005B1</td>
<td>V</td>
<td>DUMMY</td>
<td>KAVFA</td>
<td>I</td>
<td>ARRAY</td>
<td>005A1</td>
<td>V</td>
<td>36</td>
</tr>
<tr>
<td>KAVFA</td>
<td>I</td>
<td>ARRAY</td>
<td>005A1</td>
<td>V</td>
<td>36</td>
<td>KAVFA</td>
<td>I</td>
<td>ARRAY</td>
<td>005A1</td>
<td>V</td>
<td>36</td>
<td>KAVFA</td>
<td>I</td>
<td>ARRAY</td>
<td>005A1</td>
<td>V</td>
<td>36</td>
</tr>
<tr>
<td>KCENLAT</td>
<td>I</td>
<td>ARRAY</td>
<td>00555</td>
<td>V</td>
<td>36</td>
<td>KCENLAT</td>
<td>I</td>
<td>ARRAY</td>
<td>00579</td>
<td>V</td>
<td>36</td>
<td>KDALT</td>
<td>I</td>
<td>SCALR</td>
<td>005A3</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>KDALT</td>
<td>I</td>
<td>SCALR</td>
<td>005A4</td>
<td>V</td>
<td>1</td>
<td>KDALT</td>
<td>I</td>
<td>SCALR</td>
<td>005A4</td>
<td>V</td>
<td>1</td>
<td>KDALT</td>
<td>I</td>
<td>SCALR</td>
<td>005A4</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>KDOM4</td>
<td>I</td>
<td>SCALR</td>
<td>005A6</td>
<td>V</td>
<td>1</td>
<td>KDOM4</td>
<td>I</td>
<td>SCALR</td>
<td>005A6</td>
<td>V</td>
<td>1</td>
<td>KDOM4</td>
<td>I</td>
<td>SCALR</td>
<td>005A6</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>KLANG</td>
<td>I</td>
<td>SCALR</td>
<td>005AA</td>
<td>V</td>
<td>1</td>
<td>KLANG</td>
<td>I</td>
<td>SCALR</td>
<td>005AA</td>
<td>V</td>
<td>1</td>
<td>KLANG</td>
<td>I</td>
<td>SCALR</td>
<td>005AA</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>ND</td>
<td>I</td>
<td>SCALR</td>
<td>005A8</td>
<td>V</td>
<td>1</td>
<td>ND</td>
<td>I</td>
<td>SCALR</td>
<td>005A8</td>
<td>V</td>
<td>1</td>
<td>ND</td>
<td>I</td>
<td>SCALR</td>
<td>005A8</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>NX</td>
<td>I</td>
<td>SCALR</td>
<td>005B2</td>
<td>V</td>
<td>DUMMY</td>
<td>NX</td>
<td>I</td>
<td>SCALR</td>
<td>005A8</td>
<td>V</td>
<td>1</td>
<td>NX</td>
<td>I</td>
<td>SCALR</td>
<td>005B2</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>RAENG</td>
<td>R</td>
<td>SCALR</td>
<td>005A0</td>
<td>V</td>
<td>1</td>
<td>RAENG</td>
<td>R</td>
<td>SCALR</td>
<td>005A0</td>
<td>V</td>
<td>1</td>
<td>RAENG</td>
<td>R</td>
<td>SCALR</td>
<td>005A0</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>RLFT</td>
<td>R</td>
<td>SCALR</td>
<td>005B9</td>
<td>V</td>
<td>DUMMY</td>
<td>RLFT</td>
<td>R</td>
<td>SCALR</td>
<td>005B9</td>
<td>V</td>
<td>DUMMY</td>
<td>RLFT</td>
<td>R</td>
<td>SCALR</td>
<td>005B9</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>RSTP</td>
<td>R</td>
<td>SCALR</td>
<td>005BB</td>
<td>V</td>
<td>DUMMY</td>
<td>RSTP</td>
<td>R</td>
<td>SCALR</td>
<td>005BB</td>
<td>V</td>
<td>DUMMY</td>
<td>RSTP</td>
<td>R</td>
<td>SCALR</td>
<td>005BB</td>
<td>V</td>
<td>DUMMY</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (1457 WORDS):

00000 GETGA 00001 IBUFIN 004A1 KAVFA 004C5 KAVEL 004E9 KPTS 0050D Kcies-EL
00531 KCENLA 00555 KCENLAT 00579 KCENLIN 0059D INIT 0059E NCARD 0059F NIN
00540 RAENG 005A1 DEGRA 005A2 KDTP 005A3 KDBBT 005A4 KDLFT 005A5 KDRTL
005A6 IKEY 005A7 ND 005A8 NI 005A9 KLAB 005AA KLBNG 005AB KDOM1
005AC KDOM2 005AD KDOM3 005AE KDOM4 005AF KDOM5 005B0 1

BLANK COMMON (10 WORDS)

ENTRY POINTS:

00000 GETGA

INTRINSIC SUBPROGRAMS USED:

FLBTA

EXTERNAL SUBPROGRAMS REQUIRED:

BUFFERIN F1108 9DECIDE 9ENBL 9SIDATA 9ISTEP 9PRINT
9SETUPN 9STSP
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATE CODE</td>
<td>284</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>8</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>1457</td>
</tr>
<tr>
<td>TEMPS</td>
<td>16</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>1785</td>
</tr>
</tbody>
</table>
SUBROUTINE GETGC(ITAPE, NX, DATAx, NY, DATAY, NZ, DATAZ, 
RTOP, RBBT, RLFT, RRGB, RLAT, RLON, IEQU)

VERSION OF 10 JAN 76 TO DE NO PROPERLY

AND TO IMPLEMENT SSW 46

ORIGINAL VERSION 13 SEPT 75

DIMENSION IRAFIN(6,300)

DIMENSION KADE(300), KDLAT(300), KDLONG(300), KELDER(300), 

DIMENSION KAFA(10,10), KAVEL(10,10), NPTS(10,10)

DIMENSION HWFAR(10,10), KCENLAT(10,10), KCENLON(10,10)

DIMENSION KCNCEL(10,10), KCNENFA(10,10), KCNENLAT(10,10), KCNENLON(10,10)

DIMENSION NAVFA(100), NAVEL(100)

EQUIVALENCE (KAVFA, NAVFA)

EQUIVALENCE (KAVEL, NAVEL)

EQUIVALENCE (KCNCEL, KCNENLAT)

EQUIVALENCE (KCNEL, KCNENEL)

EQUIVALENCE (KCNFA, KCNENFA)

EQUIVALENCE (KCNLON, KCNENLON)

DATA INIT/0000/

DATA END/'EITP'

IF(INIT=NE) GS TS 200

IF(INIT=NE) GS TS 200

AUTPUT 'SUBROUTINE GETGC VERSION 10 JAN 76'

INIT=1

CONTINUE

100 CONTINUE

NCAHRD=105

NDLT=100

KURLAT=999

KURFLONG=999

NRET=0

IEQU=0

DB 107 I=1,10

DB 106J=1,10

108 KAVFA(I,J)=0

KAVEL(I,J)=0

NPTS(I,J)=0

KCENEL(I,J)=0

KCENEL(I,J)=0

KCNFA(I,J)=0

KCNFA(I,J)=0

KCNLON(I,J)=999

KCNLON(I,J)=999

43 106 CONTINUE

44 107 CONTINUE

45 NRET=0

46 ILAST=0

47 NBR=0

48 NI=0

49 RADIUS=57,2958

50 DEGRA=1,7453E=2

51 IF(ISW(25) EQ 0) GB TO 120

52 READ(NCARD,1001) NTAPESN

53 1001 FORMAT(20A4)

54 IF(NTAPESN EQ EITP) GB TO 999

55 CALL MOUNT(ITAPE, NTAPESN)

56 WRITE(NPRINT,1002) NTAPESN

57 1002 FORMAT(1X, 'TAPE S/N', A4, ' MOUNTED!')

58 120 CONTINUE

59 KOTOP=RTOP, RADIUS=200
KDBUT=RBAT=RADEG=200
KDLFT=RLFT=RADEG=200
KDRGT=RRGT=RADEG=200

110 CONTINUE
CALL BUFFER IN(ITAPE,0,IBUFIN(1,1),1650,KEY,NI)
G9 TO (111,115,113,114) KEY
111 OUTPUT "WAITING" G9 TO 110
113 HPUTEND OF FILE ON INPUT' G9 TO 900
114 OUTPUTERR ON INPUT' STOP
115 CONTINUE
70 NI=(NI*2)/2
71 DECODE(NI*22,1005,IBUFIN(1,1),NI)
72 IF(KDE(I),KDLAT(I),KDLANG(I),KELDEP(I),KFA(I),I=1,NI)
73 IDCODE=0
74 N=0
75 IF(NX=EQ.4) GA TO 200
76 IF(NX=EQ.5) GA TO 200
77 IF(NX=EQ.6) GA TO 200
78 IF(NX=EQ.7) GA TO 200
79 IF(NX=EQ.8) READ(NDET=1003,FND=900) NBR,NDET,NLTLG,NLTLGL
80 CONTINUE
200 FORMAT(10(IX=16))
201 FORMAT(1X=15,44)
202 IF(ILAST=EQ.1) GA TO 910
203 IF(NDET=GT.7) GA TO 685
204 IF(IS=EQ.6) GA TO 205
205 IF(NH=EQ.0) GA TO 205
206 IF(NH+E=EQ.0) READ(NDET=1003,FND=900) NBR,NDET,NLTLG,NLTLGL
207 CONTINUE
1003 FORMAT(5X=14,31)
C CHECK BOUNDS
91 IF(NLTLG,EQ.5,KEY) GA TO 500
92 IF(NLTLG,EQ.4,KDFK) GA TO 204
93 50 TO 50
94 CONTINUE
95 IF(NLTLG,EQ.5,KEY) GA TO 500
96 C WITHIN BOUNDS
97 CONTINUE
205 IF(NLTLG,EQ.5,KEY) GA TO 220
206 CONTINUE
CALL BUFFER IN(ITAPE,0,IBUFIN(1,1),1650,KEY,NI)
101 91 TO (211,215,213,214) KEY
102 OUTPUT "WAITING" G9 TO 210
103 HPUTEND OF FILE ON INPUT' G9 TO 900
104 OUTPUTERR ON INPUT' STOP
105 CONTINUE
210 FORMAT(10(IX=16,15))
C
106 IF(NLTLG,EQ.5,KEY) GA TO 220
107 IF(NLTLG,EQ.4,KDFK) GA TO 220
108 IF(KDE(I),KDLAT(I),KDLANG(I),KELDEP(I),KFA(I),I=1,NI)
109 IDCODE=0
110 CONTINUE
1005 FORMAT(300(I1),15,16,15))
111 N=0
112 CONTINUE
113 IF(IDCODE+EQ.1)
114 IF(KDE(I)+EQ.2,1005,IBUFIN(1,1),NI)
115 IF(KDE(I),KDLAT(I),KDLANG(I),KELDEP(I),KFA(I),I=1,NI)
116 IDCODE=0
117 N=N+1
118 NBR=NBR+1
119 IF(KLAT+KDLAT(NI)/100
C

120. KL=KL+1
121. IF (ISW(46).EQ.1) G0 TO 230
122. IF (KLT*GT*KDRAT) G0 TO 200
123. IF (KLT*KDLFT) ARI (KLB+GT*KDRAT)) G0 TO 200
124. 230 CONTINUE
125. G0 TO (240,250,260,270,300,600,600,600,600,600,600,600,600,600)
126. C

127. 240 CONTINUE
128. IF (KELDEP(NIN) GE 0) G0 TO 200
129. DATA=FLOAT(KELDEP(NIN))
130. G0 TO 480
131. C

132. 250 CONTINUE
133. IF (KELDEP(NIN) LE 0) G0 TO 200
134. DATA=FLOAT(KELDEP(NIN))
135. G0 TO 480
136. C

137. 260 CONTINUE
138. DATA=FLOAT(KFA(NIN)/10)
139. G0 TO 480
140. C

141. 270 CONTINUE
142. OUTPUT 'BAUGURE CALC NOT IMPLEMENTED! STOP
143. 280 CONTINUE
144. IF (NZ*EQ.0) G0 TO 480
145. G0 TO (340,350,360,370) (NY=3)
146. 340 CONTINUE
147. IF (KELDEP(NIN) GE 0) G0 TO 200
148. DATA=FLOAT(KELDEP(NIN))
149. G0 TO 380
150. 350 CONTINUE
151. IF (KELDEP(NIN) LE 0) G0 TO 200
152. DATA=FLOAT(KELDEP(NIN))
153. G0 TO 340
154. 360 CONTINUE
155. DATA=FLOAT(KFA(NIN)*10)
156. G0 TO 380
157. 370 CONTINUE
158. 380 CONTINUE
159. IF (NZ*EQ.0) G0 TO 480
160. G0 TO (440,450,460,470) (NZ=3)
161. 440 CONTINUE
162. IF (KELDEP(NIN) GE 0) G0 TO 200
163. DATA=FLOAT(KELDEP(NIN))
164. G0 TO 480
165. 450 CONTINUE
166. IF (KELDEP(NIN) LE 0) G0 TO 200
167. DATA=FLOAT(KELDEP(NIN))
168. G0 TO 480
169. 460 CONTINUE
170. DATA=FLOAT(KFA(NIN)*10)
171. G0 TO 480
172. 470 CONTINUE
173. 480 CONTINUE
174. KLNT=(1FLT(KDLAT(NIN))/100)+200)*DEGRA
175. KLAT=(1FLT(KDLAT(NIN))/100)+200)*DEGRA
176. RETURN
177. C

178. 500 CONTINUE
179. IF (NOLTLR*LT*KDLFT) AND (NOLTLT*LT*KDBAT))
190.  * OUTPUT 'PAST ROUNDS' & GO TO 900
191.  C OUTSIDE BOUNDS
192.  502 CONTINUE
193.  IF(NBR.LT.(300-NIN))
194.  * IIOC0DF=1
195.  * NIN=NIN+NBR NBR=0 GO TO 200
196.  CALL BUFFER IN((TAPE=0,IBUFIN(1,1),1650,IKEY,NI)
197.  1012 FORMAT(1X,5I5,34)
198.  NIN=NIN+1
199.  505 CONTINUE
200.  GO TO (511,515,513,514) IKEY
201.  511 OUTPUT WAITING & GO TO 505
202.  513 OUTPUT EOF FOUND WHILE SKIPPING; GO TO 900
203.  514 OUTPUT 'ERROR WHILE SKIPPING'
204.  C CONTINUE
205.  NIN=0
206.  GO TO 502
207.  C ABSTRACT DATA BEFORE PLOTTING
208.  600 CONTINUE
209.  IF(KURLAT.EQ.999) GO TO 668
210.  IF((KLAT.EQ.KURLAT) OR (KLANG.EQ.KURLANG)) GO TO 700
211.  C NEW DEGREE SQUARE
212.  C PREPARING PRIOR DEGREE SQUARE FOR OUTPUT
213.  DB 660 I=1,10
214.  DO 650 J=1,10
215.  KAVFA(I,J)*KAVFA(I,J)/NPTS(I,J)
216.  KAVEL(I,J)*KAEL(I,J)/NPTS(I,J)
217.  650 CONTINUE
218.  660 CONTINUE
219.  665 CONTINUE
220.  IF(NRET.GT.100) GO TO 668
221.  IF((N.X.EQ.3) DATA=FLOAT(NCENFA(NRET))*0.11 GO TO 667
222.  IF((N.X.EQ.9) DATA=FLOAT(NCENEL(NRET))*GO TO 667
223.  IF((N.X.EQ.12) DATA=FLOAT(NAVFA(NRET))*0.11 GO TO 667
224.  IF((N.X.EQ.13) DATA=FLOAT(NAVEL(NRET)) I GO TO 667
225.  667 CONTINUE
226.  IF(DATA.EQ.0) GO TO 665
227.  KAV=(FLOAT(NCENLAT(NRET))/100.*200.)*DEGRA
228.  KAVEL=(FLOAT(NCENLONG(NRET))/100.*200.)*DEGRA
229.  RETURN
230.  668 CONTINUE
231.  KURLAT=KLAT
232.  KURLANG=KLANG
233.  NRET=0
234.  C CLEARING ARRAYS BEFORE NEW DEGREE SQUARE
235.  DO 680 I=1,10
236.  DO 670 J=1,10
237.  KAVFA(I,J)=0
238.  KAVEL(I,J)=0
239.  NPTS(I,J)=0
240.  KCENEL(I,J)=0
241.  KCENFA(I,J)=0
242.  HINFH(I,J)=0
243.  670 CONTINUE
244.  680 CONTINUE
SAME DEGREE SQUARE

C 700 CONTINUE
C DLAT=FLAT(KDLAT(NIN))/100*
C DLONG=FLAT(KDLONG(NIN))/1000*
C DECLAT=FLAT(IFIX(DLAT))*0.001
C DECLONG=FLAT(IFIX(DLONG))*0.001
C KDECLAT=IFIX(DECLAT*1000)
C KDECLONG=IFIX(DECLONG*1000)
C I=IFIX(DELAT+10)+1
C J=IFIX(DLONG+10)+1
C KAVEL(I,J)=KAVEL(I,J)+KELDEP(NIN)
C KAVFA(I,J)=KAVFA(I,J)+KFA(NIN)
C NPTS(I,J)=NPTS(I,J)+1
C DIST=SQRT(KDECLAT*1+10*5)**2+(KDECLONG*10*5)**2
C IF(DIST<HOWFAR(I,J)) HOWFAR(I,J)=DIST
C KCENLAT(I,J)=KDLAT(NIN)
C KCENLONG(I,J)=KDLONG(NIN)
C KCENFA(I,J)=KFA(NIN)
C KCENEL(I,J)=KELDEP(NIN)
C GB TO 200 PREPARING LAST DEGREE SQUARE
C 900 CONTINUE
C IF(NX<10) GB TO 999
C I=LAST+1
C D8 906 I*1,10
C D8 905 J*1,10
C KAVFA(I,J)=KAVFA(I,J)/NPTS(I,J)
C KAVEL(I,J)=KAVEL(I,J)/NPTS(I,J)
C 905 CONTINUE
C 906 CONTINUE
C OUTPUT LAST DEGREE SQUARE
C 910 CONTINUE
C NRET=NRET+1
C IF(NRET<100) GB TO 999
C IF(NX=8) DATA=FLAT(NCENFA(NRET))*0+1) GB TO 967
C IF(NX=08) DATA=FLAT(NCENEL(NRET)) GB TO 967
C IF(NX=13) DATA=FLAT(NAVFA(NRET))*0+1) GB TO 967
C IF(NX=3) DATA=FLAT(NVAEL(NRET)) GB TO 967
C 967 CONTINUE
C IF(DATA=0) GB TO 910
C RLAT=((FLAT(NCENLAT(NRET))/100*)*200+1)*DEGRA
C RLONG=((FLAT(NCENLONG(NRET))/100*)*200+1)*DEGRA
C RETURN
C 999 CONTINUE
C I=1
C RETURN
C END
BLANK COMMON (0 WORDS)
ENTRY POINTS:
00000 GETGC

INTRINSIC SUBPROGRAMS USED:
FLOAT IFIX SORT

EXTERNAL SUBPROGRAMS REQUIRED:
BUFFERIN ISK MOUNT F1101 F1102 F1103 F1104 F1105
F1106 F1108 9ACORDEE 9ACDREAD 9ACDWRITE 9DECHOE 9ENDML 9INCDATA
911HR 9PRINT 9RTAI 9SETUPE 9SURT 9STOP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DECODED</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>997</td>
<td>003E5</td>
</tr>
<tr>
<td>16</td>
<td>00010</td>
</tr>
<tr>
<td>418</td>
<td>0102A</td>
</tr>
<tr>
<td>16</td>
<td>00010</td>
</tr>
</tbody>
</table>

TOTAL PROGRAM: 5167 0142F
SUBROUTINE GETGS(I TAPE, AX, DATA, X, IFMT, RTBF, RBBT, ILF, HRIG, HRT, 
1 ALAT, RLAT, KGCA, KGM, KGCR, KHR, KEG, IEC, IBLK)

VERSION 30 JAN 75, CORRECT READ FOR ISM(29)
VERSION 26 JUNE 74, FINAL TOCHES CONVERSION TO 67 G FORMULA
VERSION 12 JUNE 74
VERSION OF 10 AUGUST 1973, START CONVERSION TO 1967 G FORMULA
VERSION 30 MAY 73

SSW(12) UP TO LIST DATE IDENTIFICATION
SSW(27) UP TO SUPPRESS REWRITE OF TAPES AT START OF JOB
SSW(29) = 1 SUPPRESS REWRITE OF ITAPE
SSW(29) = 1 READ AND TEST FOR SELECTED SOURCE CODE
     NUMBERS TO BE PROCESSED
SSW(29) = 2 READ AND TEST FOR SELECTED SOURCE CODE
     NUMBERS TO BE SKIPPED

SSW(40) UP TO PROCESS WITH BOUNDS USING CLT
     0 = PROCESS WITHOUT BOUNDS
     1 = PROCESS WITH BOUNDS USING THE DATA LOCATION TABLE

USES ENDG(CUMMY), EVIL, STAT, ISM
ASSUME ISM AND STAT INITIALIZED IN MAIN PROGRAM

DIMENSION IGSL(12), IFMT(40)
DIMENSION ABLF(50), ABLI(128), ABLK(128)
DIMENSION IS(20), IL(20)
DIMENSION ITLN(20), ITPL(20)
DIMENSION KVAF(20), KVPM(20)
DIMENSION ICR(20), IFMT(20)
DIMENSION ICR(20), IFMT(20)
DIMENSION IVAR(5)
DIMENSION IDTN(20), IBK(20), ITK(20), IDP(20), IDESC(17,20)
DIMENSION IDTN(20), IBK(20), ITK(20), IDP(20), IDESC(17,20)
DIMENSION PLT(20), IA(35), IAFM(9), IASH(35), ISRC(16)

DATA ITER, EIPT, /
DATA IIL, /
DATA IFLAG, IF, FINO, IFLCI, C/ 
DATA IGSL, /1, 1
DATA IVAR(3), ILN(3), IP(3)
DATA IVAF(4), ILN(4), IP(4)
DATA IVAF(5), ILN(5), IP(5)
DATA IVAF(6), ILN(6), IP(6)
DATA IVAF(7), ILN(7), IP(7)
DATA IVAF(8), ILN(8), IP(8)
DATA IVAF(9), ILN(9), IP(9)
DATA IVAF(10), ILN(10), IP(10)
DATA IVAF(11), ILN(11), IP(11)
DATA IVAF(12), ILN(12), IP(12)
DATA IVAF(13), ILN(13), IP(13)
DATA IVAF(14), ILN(14), IP(14)
DATA IVAF(15), ILN(15), IP(15)
DATA IVAF(16), ILN(16), IP(16)
DATA IVAF(17), ILN(17), IP(17)
DATA IVAF(18), ILN(18), IP(18)
DATA IVAF(19), ILN(19), IP(19)
DATA IVAF(20), ILN(20), IP(20)
DATA IVAF(21), ILN(21), IP(21)
DATA IVAF(22), ILN(22), IP(22)
DATA IVAF(23), ILN(23), IP(23)
DATA IVAF(24), ILN(24), IP(24)
DATA IVAF(25), ILN(25), IP(25)
DATA IVAF(26), ILN(26), IP(26)
DATA IVAF(27), ILN(27), IP(27)
DATA IVAF(28), ILN(28), IP(28)
DATA IVAF(29), ILN(29), IP(29)
DATA IVAF(30), ILN(30), IP(30)
DATA IVAF(31), ILN(31), IP(31)
DATA IVAF(32), ILN(32), IP(32)
DATA IVAF(33), ILN(33), IP(33)
DATA IVAF(34), ILN(34), IP(34)
DATA IVAF(35), ILN(35), IP(35)
DATA IVAF(36), ILN(36), IP(36)
DATA IVAF(37), ILN(37), IP(37)
DATA IVAF(38), ILN(38), IP(38)
DATA IVAF(39), ILN(39), IP(39)
DATA IVAF(40), ILN(40), IP(40)
DATA IVAF(41), ILN(41), IP(41)
DATA IVAF(42), ILN(42), IP(42)
DATA IVAF(43), ILN(43), IP(43)
DATA IVAF(44), ILN(44), IP(44)
DATA IVAF(45), ILN(45), IP(45)
DATA IVAF(46), ILN(46), IP(46)
DATA IVAF(47), ILN(47), IP(47)
DATA IVAF(48), ILN(48), IP(48)
DATA IVAF(49), ILN(49), IP(49)
DATA IVAF(50), ILN(50), IP(50)
DATA IVAF(51), ILN(51), IP(51)
DATA IVAF(52), ILN(52), IP(52)
DATA IVAF(53), ILN(53), IP(53)
DATA IVAF(54), ILN(54), IP(54)
DATA IVAF(55), ILN(55), IP(55)
DATA IVAF(56), ILN(56), IP(56)
DATA IVAF(57), ILN(57), IP(57)
DATA IVAF(58), ILN(58), IP(58)
DATA IVAF(59), ILN(59), IP(59)
C

KCMD*256

C NRECT = NB. OF RECORDS NOW WRITTEN ON PRESENT OUTPUT TAPE
C NEF = NB. OF FILE NOW BEING PROCESSED
C NRECT = 0
IREC = 1

C

OUTPLT 'GETGS VERSION 3C JAN 75 FOR 67 C FORMULA!

C

CHECK SSW(29) TO SEE IF SOURCE CODE NUMBERS
ARE TO BE READ FOR DATA SELECTION

C

IF(ISW(29)*EG*0)G8T01405

C

READ(IN,9CC)ISRC

9CC

FORMAT(1615)

IF(ISW(31)*EG*1)WRITE(11BLT,912)ISRC ; G8T01405

WRITE(11BLT,913)ISRC

913 FORMAT(IHC,ICX), 'SKIPPED SOURCE CODES = ',1615)

912 FORMAT(IHC,ICX), 'SELECTED SOURCE CODES = ',1615)

C

405

IF(ISW(30)*EG*1)CALL ENCLT(J,CLATO,CLABB,CLBE,CLBRO,IDL,IL)D,0)

* ) CALL SETSKP(INDICA) ; IDLT=0

* ) OUTFILE.* = 1

* ) G8 TO 1410

IF(ISW(30)*EG*1)G8T01410

404

READ(IN,406)IDTIN(0),ITK(J),IDENS(J),

1 (IDESC(K),K=1,4)

406

FORMAT(A4,1X,4X,1X,1X,1X,1X,1X,1X,1X,1X,17A4)

IF(IDTIN(J)*NE*ITER1) =*+11 G8 TO 405

151

NEF1 =

152

IFILE.* = 1

153

OUTFILE.* =

C

1410

IF(ISW(31))16,10,2414

10 IF(ISW(40)*EG*C)G8T0811

READ(IDC,4C6,IDTIN(1),ITK(1),IDENS(1),

* (IDESC(K),K=1,17) ; IMDL*C

811

CALL MOUNT(ITAPE,1D041)

OUTFILE 'INPUT TAPE MOUNTED'

WRITE(11BLT,1413)IDTIN(1),ITK(1),IDENS(1),

1 (IDESC(K),K=1,17)

1413

FORMAT(A4,1X,1X,1X,1X,1X,1X,1X,1X,1X,1X,17A4)

168

IF(IDTIN(U)*NE*ITER1)IEBD=1 ; RETURN

166

OUTFILE '-------------------------'

167

IF(ISW(27)*EG*C)G8T02414

168

REWIND ITAPE

169

IF(IPTXHEG)PRINT920 ; IEBD=1 ; RETURN

170

92C FORMAT(IHC,5A), 'IFMT NOT EQUAL TO 3 , GETG CAN ONLY PROCESS GSUM

171

* DATA*/

172

C82414=1,4C

173

IF(PTX1)*IGSU(1)

174

924 CONTINUE

175

IF(NLXLT,2)G8T02414

176

PRINT921;AX;IEBD=1 ; RETURN

177

921 FORMAT(IHC,5A,1X,1X,1X,1X,1X,1X,1X,1X,1X); THIS IFMT X0 NOT YET INCLUDED/

178

C

179

IFMS1*IFMT(E) ; IFMS2*IFMT(7)
C 414 CONTINUE

PRINT@15,IFMT

FARMAT(' INPUT IREC = /',I1X,20A4,'/1X,20A4/',
* 6X,('-------------',/)

C -------- GETG INPUT LOGIC

C 50 CALL ENDB

IF(ISIN(*0) .NE. *C)*GT8601

C 52 CALL BLFIG(I Tape,IE N D,1,0,
*IBLK,JK2,IFMT,BLI,BLF,BLI,BLK,ICNT)

IF(IE ND .EQ. 1) I = 2 ; G0 T0 710

C IF(NX .EQ. D) GET8650

DECODE(KMAX,IFMT,BLI,IREC,IS06,KGDA,KGP6,KGYR,KGHM,DLAT,DLNG,
* LTKEY,LOGKEY

IF(NX .EQ. *1) VAR = KGHM ; G0 T0 970

IF(NX .EQ. *2) VAR = IS0RC

GT8970

C 950 DECODE(KMAX,IFMT,BLI,IREC,IS06,KGDA,KGP6,KGYR,KGHM,DLAT,DLNG,
* (IRVAR(*1) .EQ. *1)NRVAR(NX) ;
* LTKEY,LOGKEY

IF(IFLG1 .EQ. *1) G0 T0 373

IFLOG = *1

WRITE(I100,374) IREC1

C 374 FARMAT(' INPUT IREC = /',I12)

IF (IREC .EQ. *1) OUTPUT ' CONVERTING TO 67 FORMULA IN THIS RUN;

IF (IREC .EQ. *2) OUTPUT ' INPUT ALREADY IN 67 FORMULA'

C 373 CONTINUE

IF(NX .NE. *S)*GT8953

IF(RVAR(2) .EQ. *C) VAR = RVAR(1) ; G0 T0 870

VAR = RVAR(2) ; G0 T0 870

GT8970

C 953 IF(NX .NE. *9)*GT8954

VAR = RVAR(1) + RVAR(2) ; G0 T0 870

C 954 IF(NX .NE. *11)*GT8955

NX = *11 TO FLOAT OBSERVED GRAVITY

C 230 C THIS ROUTINE CANNOT PLOT OBSERVED GRAVITY

A = RVAR(1) .EQ. 977 * A = A .EQ. 10CC .O

C 323 A = RVAR(2) .EQ. *A + G0 T0 870

C 955 IF(NX .LT. *L)*GT8956

IF(RVAR(1) .EQ. *10)*PRINT87 ; IEOQ = *1 ; RETURN

C 957 FARMAT(' 890 C , IFFC CODE NOT EQUAL TO 1C */')

A9R1 .EQ. *1/RVAR(2) ; A9R1 .EQ. A9R1/2*

IAX = LTKEY .EQ. 89 ; IX = LOGKEY .EQ. *80

ALV = FLOAT(IAX) ; BLV = FLOAT(IX)
CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF THE 1967 INTERNATIONAL GRAVITY FORMULA
THIS SUB DECADES ONLY THAT VARIABLE TO BE PLOTTED.
NX DETERMINES THE VARIABLE.

1. OUTPLT ' CANNO' PLOT OBSERVED GRAVITY',
2. STOP

TO AVOID PLOTTING INVALID DATA POINTS

IF((NX*EG<6 | 8R|)
   1. NX*EG=7 | 8R|
   2. NX*EG=9 | 8R|
   3. NX*EG=10 | 8R|
   4. NX*EG=14 | 8R|
   5. NX*EG=15 | AND|
   6. VAR=GT 990'C)
   7. 09 TO 50

CORRECTION FOR FREE AIR OR BOURSIER FOR 67 G FORMULA
\[ RLAT = DLAT + DEGRA \]
\[ CG=3.2* (13.6* (SIN(ABS(RLAT)))*2) \]

IF((NX*EG=6 | 8R|
   1. NX*EG=7 | 8R|
   2. NX*EG=9 | 8R|
   3. NX*EG=10 | 8R|
   4. NX*EG=14 | 8R|
   5. NX*EG=15 | AND|
   6. TREC1 *EG= 1)
   7. VAR = VAR + DG

PLT(NX)*VAR
10C CONTINUE
41C CONTINUE
KGDA*KGDA
KGMAB*KGMAB
KGYR*KGYR
KGMAC*KGMAC

IF((NX)*1100=1200=1100
110C DATA*PLT(NX)
X BLTPUT DATAX
120C RETURN

GETG DLT INPUT LOGIC

501 IF(IDLT*EG=1)G07B514
502 IF(IDCHA*EG=1)G07B507

INPUT AN ELEMENT OF LIST OF DEGREES NEEDED
READ(IIL*,5C3=END=550)LA10C,LAT,C,L810C,L9NC
502 FORMAT(2(12,11))

507 CONTINUE | ICHA*0
507 CONTINUE | ICHA*0

INPUT A MEMBER OF DLT
READ(IDL*,5C3,END=540)ICEN=INBR,LA10C,LAT1,L8101L9NC,L8102L9N2
SC3 FORMAT(A4,1X,16,3(I2,11))
C
C ------ TESTING DLT FOR CONSISTENCY
C IF(LB1CNE.LB1C2)PRINTE4 & OUTPUT LB1C,LB1C2 & STOP
C SC4 FORMAT(1HC,10X,'ERROR IN DLT',/,'2CX,'LB1C NOT EQUAL TO LB1C2',/)
C IF(IECA.EQ.IEB)GOT8530
C IF(IECA.EQ.ITER)GOT8540
C
C LB1C*LB1C2 / ILN=LB1C-LB1C2+1
C
C ------- SIMPLT SCANNING OF DLT AND MATCHING LIST
C
C SC5 KEY1*LA1OT-LA1CC / KEY2*LB1C-LB1C2
C KEY3*LT=LTAC
C
C CB506I=1,ILN
C LBN=LB1C1=I+1
C KEY4=LBN-LBNC
C IF(KEY1)52G*51C+506
C 51C IF(KEY2)52G*511+506
C 511 IF(KEY3)52G*512+506
C 512 IF(KEY4)52G*513+506
C SC6 CONTINUE
C
C ------- SKIPPING UNNECESSARY RECORDS
C CB5CSI*INBR
C CALL BLFGET(ITAPE,1END,11,
C * IBLK,32,KMAX,KN,1ST,ILT,BLI,BLK,ICNT)
C 5CS CONTINUE
C 50S GOTO 5C7
C
C ------- ERROR CONDITIONS
C SC8 PRINT34 & IE6D*1 & RETURN
C SC44 FORMAT(1HC,10X,'INCORRECT DLT * FOUND EOF WHILE SKIPPING REC',/)
C
C ------- CONDITIONAL BRANCH TO READ PROCESS
C SC13 ICND*0 / IDLT=1 / OUTPUT 'PROCESS',INBR & G0T852
C 514 IF(ICND.EQ.INBR)IDLT=0 & G0T8501
C 50T852
C
C ------- INPUT AN ELEMENT OF MATCHING LIST
C SC2 READ(LI),5C2(END=550)LA1CC,LTAC,LB1C,LBNC
C G0T8505
C
C ------- COMPLETION MESSAGE
C SC3 PRINT551 & IE6D*1 & OUTPUT ICNT & RETURN
C SC41 FORMAT(1HC,10X,'AREA PROCESSED = STOP',/)
C
C ------- CHANGE TAPE REEL
C SC6 IDLT=0 / ICHA*1 & G0T8578
C
420* C ----- END OF CLT REACHED
421* C
422* 540 PRINTS41 ; IE80=1 ; RETURN
423* 541 FORMAT(1HO,1CX,'END OF CLT REACHED = STOP')
424* END
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>GENERATED CODE</td>
<td>1276</td>
<td>C04FC</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>20</td>
<td>C0014</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>3375</td>
<td>C0D2F</td>
</tr>
<tr>
<td>TEMPS</td>
<td>20</td>
<td>C0014</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>4691</td>
<td>C1253</td>
</tr>
</tbody>
</table>
SLBROBUTINE GETH(ITAPEsNX,Y,Z,W,CATAX,CATAY,CATAZ,CATAW,
1 RLAT,RLONG,KGDA,KGMB,KGYR,KGHM,IEBD)

VERSION OF 6 JUNE 1972, ALLOWS FOR LETTERED STATION NUMBERS
VERSION OF 22 MAR 1972, IGNORES COL 1 AND CHECKS IF LAT
AND LONG ARE ZER
VERSION OF 18 MAR 1972 -- FIRST GENERATION OF ROUTINE

SLBROBUTINE GETH, FOR READING HEAT FLOW DATA

1. DIMENSION PLT(8),M(6)
2. IIN = 105
3. IIBUT = 108
4. IEBD=0
5. KGDA=0
6. KGMB=0
7. CONTINUE
8. READ (ITAPE,4) IN8,AN8,P,STAID,LAT,A1,ILATH,KNS,LNG,A2,ILOM,KEW,
9. IHEIT,GRAD,CBND,HF,ICC,IREF,IR Y
10. FORMAT (X,XA,BXY,AX,XB,ZB,XY)
11. CALL STAT(I)
12. CALL EVIL (IIBUT,IEBD,KGDA,KGMB,KGYR,KGHM)
13. IF (IBAD 4) 410 53 900
14. CONTINUE
15. KGYR=IYR
16. KGMB=IN8
17. RLATM=ILATH
18. RLOM=ILOM
19. CHECKING FOR ZERO LAT AND LONG
20. IF (LAT)7C60 70
21. IF (ILATM)7C62 70
22. IF (LNG)7C64 70
23. IF (ILOM)7C410 70
24. CALL NAVIN (LAT,RLATM,KNS,LNG,RLOM,KEW,RLAT,RLONG)
25. KGDA=KGDA
26. KGMB=KGMB
27. KGYR=KGYR
28. KGHR=KGHR
29. PLT(1)=IN8
30. PLT(2)=IHEIT
31. PLT(3)=HF
32. PLT(4)=GRAD
33. PLT(5)=CBND
34. PLT(6)A=M(1)A1000000M(2)A1000000M(3)A1000000M(4)A1000000M(5)A1000000M(6)
35. PLT(7)=IREF
36. PLT(8)=IYR
37. IF (NX)80 85 80
38. DATA=PLT(NX)
39. DATA=IEBD=0
40. RETURN
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>R</td>
<td>SCALR</td>
<td>0012</td>
<td>V</td>
<td>A1</td>
<td>R</td>
<td>SCALR</td>
<td>0015</td>
<td>V</td>
<td>A2</td>
<td>R</td>
<td>SCALR</td>
<td>0019</td>
<td>V</td>
</tr>
<tr>
<td>C0D</td>
<td>R</td>
<td>SCALR</td>
<td>001E</td>
<td>V</td>
<td>DATA</td>
<td>R</td>
<td>SCALR</td>
<td>0003</td>
<td>V</td>
<td>DATA</td>
<td>R</td>
<td>SCALR</td>
<td>0003</td>
<td>V</td>
</tr>
<tr>
<td>DATAY</td>
<td>R</td>
<td>SCALR</td>
<td>0003</td>
<td>V</td>
<td>GETT</td>
<td>R</td>
<td>SPRB</td>
<td>0000</td>
<td>P</td>
<td>EVIL</td>
<td>R</td>
<td>SPRB</td>
<td>EXTN</td>
<td></td>
</tr>
<tr>
<td>H/F</td>
<td>R</td>
<td>SCALR</td>
<td>0010</td>
<td>V</td>
<td>IBD</td>
<td>I</td>
<td>SCALR</td>
<td>0023</td>
<td>V</td>
<td>IBAD</td>
<td>I</td>
<td>SCALR</td>
<td>0024</td>
<td>V</td>
</tr>
<tr>
<td>I/IN</td>
<td>I</td>
<td>SCALR</td>
<td>0026</td>
<td>V</td>
<td>IIBL</td>
<td>I</td>
<td>SCALR</td>
<td>0010</td>
<td>V</td>
<td>IHEIT</td>
<td>I</td>
<td>SCALR</td>
<td>001C</td>
<td>V</td>
</tr>
<tr>
<td>I/M</td>
<td>I</td>
<td>SCALR</td>
<td>0010</td>
<td>V</td>
<td>INX</td>
<td>I</td>
<td>SCALR</td>
<td>0011</td>
<td>V</td>
<td>IREF</td>
<td>I</td>
<td>SCALR</td>
<td>0021</td>
<td>V</td>
</tr>
<tr>
<td>I/TAPE</td>
<td>I</td>
<td>SCALR</td>
<td>0002</td>
<td>V</td>
<td>IYR</td>
<td>I</td>
<td>SCALR</td>
<td>0022</td>
<td>V</td>
<td>KEW</td>
<td>I</td>
<td>SCALR</td>
<td>0018</td>
<td>V</td>
</tr>
<tr>
<td>KGDA</td>
<td>I</td>
<td>SCALR</td>
<td>0036</td>
<td>V</td>
<td>KGB</td>
<td>I</td>
<td>SCALR</td>
<td>0025</td>
<td>V</td>
<td>KGMB</td>
<td>I</td>
<td>SCALR</td>
<td>0026</td>
<td>V</td>
</tr>
<tr>
<td>KGMB</td>
<td>I</td>
<td>SCALR</td>
<td>0033</td>
<td>V</td>
<td>KGMB</td>
<td>I</td>
<td>SCALR</td>
<td>0037</td>
<td>V</td>
<td>KGMB</td>
<td>I</td>
<td>SCALR</td>
<td>0039</td>
<td>V</td>
</tr>
<tr>
<td>KG/YR</td>
<td>I</td>
<td>SCALR</td>
<td>0038</td>
<td>DUMMY</td>
<td>KG/YR</td>
<td>I</td>
<td>SCALR</td>
<td>0027</td>
<td>V</td>
<td>KNS</td>
<td>I</td>
<td>SCALR</td>
<td>0017</td>
<td>V</td>
</tr>
<tr>
<td>L/N</td>
<td>I</td>
<td>SCALR</td>
<td>0014</td>
<td>V</td>
<td>L/BG</td>
<td>I</td>
<td>SCALR</td>
<td>0018</td>
<td>V</td>
<td>M</td>
<td>I</td>
<td>ARRAY</td>
<td>0009</td>
<td>V</td>
</tr>
<tr>
<td>N/A</td>
<td>I</td>
<td>SCALR</td>
<td>002D</td>
<td>V</td>
<td>NW</td>
<td>UNUSED</td>
<td>002F</td>
<td>V</td>
<td>NX</td>
<td>I</td>
<td>SCALR</td>
<td>002C</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>R/LAT</td>
<td>R</td>
<td>SCALR</td>
<td>0034</td>
<td>V</td>
<td>RLAT</td>
<td>R</td>
<td>SCALR</td>
<td>0029</td>
<td>V</td>
<td>PLT</td>
<td>R</td>
<td>ARRAY</td>
<td>0001</td>
<td>V</td>
</tr>
<tr>
<td>R/LONG</td>
<td>R</td>
<td>SCALR</td>
<td>0035</td>
<td>V</td>
<td>R/LONG</td>
<td>R</td>
<td>SCALR</td>
<td>0029</td>
<td>V</td>
<td>R/LONG</td>
<td>R</td>
<td>SCALR</td>
<td>002A</td>
<td>V</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (43 WORDS):**

<table>
<thead>
<tr>
<th>NAME</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000C3GETH</td>
<td>00001</td>
<td>PLT</td>
<td>0000C3GETH</td>
<td>00001</td>
<td>PLT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000012ANB</td>
<td>00013</td>
<td>STAD</td>
<td>000014LAT</td>
<td>000013</td>
<td>STAD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000018LBNG</td>
<td>00019</td>
<td>A2</td>
<td>00001AIBAD</td>
<td>00018</td>
<td>KEW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00001EBND</td>
<td>0001F</td>
<td>HF</td>
<td>000010IBAD</td>
<td>0001E</td>
<td>REX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000024IBAD</td>
<td>00025</td>
<td>KGDA</td>
<td>000027IKYR</td>
<td>000024</td>
<td>KGMB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ENTRY POINTS:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000C3GETH</td>
<td></td>
<td></td>
<td></td>
<td>0000C3GETH</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**EXTERNAL SUBPROGRAMS REQUIRED:**

<table>
<thead>
<tr>
<th>NAME</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>NAME</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVIL</td>
<td>N/A</td>
<td>STAT</td>
<td>F1101</td>
<td>EVIL</td>
<td>N/A</td>
<td>STAT</td>
<td>F1103</td>
<td>EVIL</td>
<td>N/A</td>
<td>STAT</td>
<td>F1105</td>
</tr>
<tr>
<td>99DLES</td>
<td>9ITOR</td>
<td>9SETUP</td>
<td>F1103</td>
<td>99DLES</td>
<td>9ITOR</td>
<td>9SETUP</td>
<td>F1103</td>
<td>99DLES</td>
<td>9ITOR</td>
<td>9SETUP</td>
<td></td>
</tr>
<tr>
<td>99DREAD</td>
<td>9168ATA</td>
<td></td>
<td></td>
<td>99DREAD</td>
<td>9168ATA</td>
<td></td>
<td></td>
<td>99DREAD</td>
<td>9168ATA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### HIGHEST ERROR SEVERITY:

- **0 (NO ERRORS)**

### DEC vs HEX

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WORDS</strong></td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>GENERATED CODE</td>
<td>189</td>
<td>CO08D</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>0</td>
<td>CO00C</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>43</td>
<td>CO02B</td>
</tr>
<tr>
<td>TEMPS</td>
<td>21</td>
<td>CO015</td>
</tr>
<tr>
<td><strong>TOTAL PROGRAM</strong></td>
<td>253</td>
<td>CO0FD</td>
</tr>
</tbody>
</table>
SUBROUTINE GETL(ITAPE,NX,NY,NZ,NX,NY,NZ,DATAx,DATAy,DATAz,DATAv)
1 1 REAL,RLONG,KGDA,KGMB,KGMR,KGHL,IE0D)
2 VERSION 16 APRIL 1975, TO CORRECT KEYPUNCH ERROR
3 VERSION OF 8 APRIL 1975, TO ADD SELECTION OF TOTAL
4 ACCELERATION WITH SIGN OF RACIAL COMPONENT
5 VERSION OF 23 MAR 1973, ADDING BERGGER CALCULATION
6 VERSION OF 28 FEB 1973, CHANGE FROM MSC FORMAT TO AN01 
7 VERSION OF 27 OCT 1972
8 SUBROUTINE GETL, READS LUNAR DATA FROM MSC SPSN PROGRAM
9
10 SSW(34) = 1, TO READ LUNAR DATA ON 2 CARDS
11 SSW(35) = 1, TO WRITE LUNAR DATA ON 2 CARDS
12
13 DIMENSION FLT(15)
14 DIMENSION IC(7)
15 DATA ISRT/C/
16 IF(ISRT NE C) GO TO 50
17
18 IE0D = 0
19 OUTPUT ' SUBROUTINE GETL, VERSION OF 16 APRIL 1975'
20 KGDA = 0
21 KGMB = 0
22 KGMR = 0
23 KGHL = 0
24 CEGRA(1) = 745329E+2
25
26 ISRT = 1
27
28 ***************
29
30 SETTING DENSITY VALUES FOR RIM AND CRUST
31
32 RIMD=2.50
33 CRUST = 2.73
34
35 ***************
36 END OF INITIALIZATION
37
38 IE0D = 0
39 IF(ISW(34) = 1) G0 TO 55
40 READ(ITAPE,100) IREC,ISORC,KDAsKGMB,KGMR,KGHL,SEC,DLAT,DLONG,SVEC, 
41 ALT, AZ, INC, STAC, SNAC, FA, THEOR, KSST, KSSN, KSSR, KSSA, ELEV, ELFL, 
42 LKEY, LGKEY
43 100 FORMAT (11I4,2F9.4,3F7.3,F6.2,F6.1)
44 GO TO 58
45 READ(ITAPE,102) IREC,ISORC,KDAsKGMB,KGMR,KGHL,SEC,DLAT,DLONG,SVEC, 
46 ALT, AZ, INC, STAC, SNAC, FA, THEOR, KSST, KSSN, KSSR, KSSA, ELEV, ELFL, 
47 LKEY, LGKEY
48 102 FORMAT (11I4,2F9.4,3F7.3,F6.2,F6.1)
49 CALL STAT(1)
50 CALL EVIL
("""
51 IF(ISW(34) = 1) G0 TO 55
52 READ(ITAPE,61) IREC,EG,1) GO TO 65
53 C FOR DATA READ FROM 2 CARDS
54 60 CONTINUE
55 IF(ISW(34) = 1) GO TO 55
56 BLTPUT 'IREC DOES NOT = 1'
57 GO TO 50
58 READ(ITAPE,62)
CONTINUE
KGDA = KDA
KGM8 = KMB
KGYR8 = Kyr
KGM = KHM
RLAT = DLAT*DEGRA
RLONG = DLONG*DEGRA
GBBS = FA+THE8R
PLT(1) = SVEC
PLT(2) = SVEC = 1738.0
IF (NX=NE=3.0 AND NX*NE=4) GO TO 80
IF (ALTL=LT=0.001) PLT(3)=0.1 PLT(4)=0.1 GO TO 50
PLT(3) = ALTL
PLT(4) = (SVEC=ALTL) = 1738.0
80 PLT(5) = AZ
PLT(6) = SINC
PLT(7) = STAC
PLT(8) = SNAAC
PLT(9) = FA
PLT(10) = THE8R
PLT(11) = GBBS
PLT(12) = ELEV
IF (NX=EQ=13.0 AND ALTL=LT=0.001) PLT(13)=0.1 GO TO 50
PLT(13) = ELFL
IF (ELFL=22.23) 23
22 SETTING DENSITY TO THAT OF CRATER RIM
C DENSC=CRIMC
89 GB TO 24
C SETTING DENSITY TO THAT OF CRUST
90 C DENSC=CRUSTD
91 23 CONTINUE
92 24 BG = FA - ((DENSC=ELFL)*0.04185)
93 PLT(14) = BG
94 C DETERMINE TOTAL ACCELERATION MAGNITUDE
95 CALL TOTAC(STAC,SNAAC,FA,TACEL)
96 PLT(15) = TACEL
97 IF (NX=90.55.9C
98 CATAX=PLT(NX)
99 9C CATAY=PLT(NY)
100 CATAZ = PLT(2)
101 RETURN
102 RETURN
103 IEOC = 1
104 RETURN
105 END
### Variables (64 Words):

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTL</td>
<td>R SCAL</td>
<td>00020</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CRSTC</td>
<td>R SCAL</td>
<td>00020</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATAV</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DENSC</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ELEV</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FA</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOBS</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I5</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISRT</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KAC</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KGMB</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KMB</td>
<td>I SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KSSA</td>
<td>I SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KVR</td>
<td>I SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NA</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZ</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINC</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAT</td>
<td>SPROG</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THEO3</td>
<td>R SCAL</td>
<td>00030</td>
<td>V 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Local Variables (64 Words):

- CCCCC GETL 00001 PLT 00001 ID 00017 ISTR 00018 I1N 00019 1:RUT
- CCCCC GETL 00018 KGMB 00018 KGM8 00018 DEGRA 00018 RGMD
- CCCCC GETL 00029 KGMB 00029 KGMB 00029 KGMB 00029 KGMB
- CCCCC GETL 00039 KGMB 00039 KGMB 00039 KGMB 00039 KGMB
- CCCCC GETL 00049 KGMB 00049 KGMB 00049 KGMB 00049 KGMB
- CCCCC GETL 00059 KGMB 00059 KGMB 00059 KGMB 00059 KGMB
- CCCCC GETL 00069 KGMB 00069 KGMB 00069 KGMB 00069 KGMB
- CCCCC GETL 00079 KGMB 00079 KGMB 00079 KGMB 00079 KGMB

### Blank Comments (6 Words):

- CCCCC GETL

### Entry Points:

- CCCCC GETL

### External SlepPrograms Required:

- EVIL
- ISR
- STAT
- TOTAC
- FI101
- FI103
- FI105
- F110R
<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generated Code</td>
<td>325</td>
<td>C0145</td>
</tr>
<tr>
<td>Constants</td>
<td>7</td>
<td>C0007</td>
</tr>
<tr>
<td>Local Variables</td>
<td>64</td>
<td>C0040</td>
</tr>
<tr>
<td>Temps</td>
<td>17</td>
<td>C0011</td>
</tr>
<tr>
<td>Total Program</td>
<td>413</td>
<td>C019D</td>
</tr>
</tbody>
</table>
SUBROUTINE GETM(ITAPE, NX, NY, NZ, NAX, DATAX, DATAY, DATAZ, DATAW, RLAT, RLON, KGDA, KGMB, KGYR, KGHM, IEBC)

SUBROUTINE GETM, FOR READING BATHYMETRY AT MBATR FORMAT

CHANGED 8 NOV. 1971 BY R. C. GRISHMAN TO UNIFY DEFINITION OF
FLT(1) * TIME IN ALL 'GET' SUBROUTINES

DIMENSION PLT(7)
IE6D = 0
I10UT = 108
CALL ENDIO
READ(ITAPE, 16) KGDA, KGMB, KGYR, KGHM, ITZ, DATA, MTAB, CDFM, CDM, DLAT, DLON
GDIS, CIR = SPC
FORMAT(312, 14, 1X, I3, 1X, F5*0, 1X, I2, 2X, F5*0, 1X, I2, 2X, F5*0, 1X, I2, 3, 1X, F5*3)
READ (ITAPE, 16) KGDA, KGMB, KGYR, KGHM, ITZ, DATA, MTAB, CDFM, CDM, DLAT, DLON
GDIS, CIR = SPC
FORMAT(312, 14, 1X, I3, 1X, F5*0, 1X, I2, 2X, F5*0, 1X, I2, 2X, F5*0, 1X, I2, 3, 1X, F5*3)
CALL STAT(1)
CALL EVIL(110UT, I, IBAD, KGDA, KGMB, KGYR, KGHM)
IF (IBAD) 14, 3C, 65
I10US = 1
RETURN
3C IF (NX) 110, 120, 110
CALL ENDC
RETURN
110 CATAX = PLT(NX)
120 CATAX = PLT(NY)
130 CATAX = PLT(NZ)
140 RETURN
END

SELECTING DATA POINT TO BE PLOTTED

IF (NX) 110, 120, 110
CALL ENDC
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX</th>
<th>DEC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CDF</td>
<td>R</td>
<td>SCALR</td>
<td>000C</td>
<td>V</td>
<td>1</td>
<td>CDF</td>
<td>R</td>
<td>SCALR</td>
<td>000C</td>
<td>V</td>
<td>1</td>
<td>CDF</td>
<td>R</td>
<td>SCALR</td>
<td>000A</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>DATA</td>
<td>R</td>
<td>SCALR</td>
<td>0021</td>
<td>V</td>
<td>DUMMY</td>
<td>DATA</td>
<td>R</td>
<td>SCALR</td>
<td>0001</td>
<td>F</td>
<td>V</td>
<td>DATA</td>
<td>R</td>
<td>SCALR</td>
<td>0001</td>
<td>F</td>
<td>V</td>
</tr>
<tr>
<td>DCLR</td>
<td>R</td>
<td>SCALR</td>
<td>0020</td>
<td>V</td>
<td>DUMMY</td>
<td>DCLR</td>
<td>R</td>
<td>SCALR</td>
<td>0011</td>
<td>V</td>
<td>1</td>
<td>DCLR</td>
<td>R</td>
<td>SCALR</td>
<td>0010</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>DLN</td>
<td>R</td>
<td>SCALR</td>
<td>00C3</td>
<td>V</td>
<td>DUMMY</td>
<td>DLN</td>
<td>R</td>
<td>SCALR</td>
<td>0015</td>
<td>V</td>
<td>1</td>
<td>DLN</td>
<td>R</td>
<td>SCALR</td>
<td>0015</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>EVIL</td>
<td>S</td>
<td>PRG</td>
<td>EXTRN</td>
<td></td>
<td></td>
<td>EVIL</td>
<td>S</td>
<td>PRG</td>
<td>EXTRN</td>
<td></td>
<td></td>
<td>EVIL</td>
<td>S</td>
<td>PRG</td>
<td>EXTRN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>0013</td>
<td>V</td>
<td>1</td>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>0019</td>
<td>V</td>
<td>DUMMY</td>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>0009</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>KGDA</td>
<td>R</td>
<td>SCALR</td>
<td>0024</td>
<td>V</td>
<td>DUMMY</td>
<td>KGDA</td>
<td>R</td>
<td>SCALR</td>
<td>0015</td>
<td>V</td>
<td>1</td>
<td>KGDA</td>
<td>R</td>
<td>SCALR</td>
<td>0007</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>KGMB</td>
<td>R</td>
<td>SCALR</td>
<td>0018</td>
<td>V</td>
<td></td>
<td>KGMB</td>
<td>R</td>
<td>SCALR</td>
<td>0015</td>
<td>V</td>
<td>KGMB</td>
<td>R</td>
<td>SCALR</td>
<td>0001</td>
<td>V</td>
<td>KGMB</td>
<td>R</td>
</tr>
<tr>
<td>KOYR</td>
<td>R</td>
<td>SCALR</td>
<td>0026</td>
<td>V</td>
<td>DUMMY</td>
<td>KOYR</td>
<td>R</td>
<td>SCALR</td>
<td>0017</td>
<td>V</td>
<td>1</td>
<td>KOYR</td>
<td>R</td>
<td>SCALR</td>
<td>0008</td>
<td>V</td>
<td>1</td>
</tr>
<tr>
<td>NW</td>
<td>I</td>
<td>SCALR</td>
<td>0010</td>
<td>V</td>
<td>DUMMY</td>
<td>NW</td>
<td>I</td>
<td>SCALR</td>
<td>0014</td>
<td>V</td>
<td>DUMMY</td>
<td>NW</td>
<td>I</td>
<td>SCALR</td>
<td>0018</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>NZ</td>
<td>I</td>
<td>SCALR</td>
<td>0011</td>
<td>V</td>
<td>DUMMY</td>
<td>NZ</td>
<td>I</td>
<td>SCALR</td>
<td>0011</td>
<td>V</td>
<td>DUMMY</td>
<td>NZ</td>
<td>I</td>
<td>SCALR</td>
<td>0018</td>
<td>V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>SLNG</td>
<td>R</td>
<td>SCALR</td>
<td>0023</td>
<td>V</td>
<td>DUMMY</td>
<td>SLNG</td>
<td>R</td>
<td>SCALR</td>
<td>0015</td>
<td>V</td>
<td>1</td>
<td>SLNG</td>
<td>R</td>
<td>SCALR</td>
<td>0007</td>
<td>V</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>CCC17</td>
<td>15</td>
<td>00019</td>
<td>16</td>
<td>CCO28</td>
<td>30</td>
<td>CCO31</td>
<td>65</td>
<td>00050</td>
</tr>
<tr>
<td>120</td>
<td>CCO37</td>
<td></td>
<td></td>
<td>110</td>
<td>CCO37</td>
<td></td>
<td></td>
<td>110</td>
<td>CCO37</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (25 WORDS):

<table>
<thead>
<tr>
<th>OOCOC</th>
<th>GETM</th>
<th>000C</th>
<th>PLT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOCOC</td>
<td>CDFH</td>
<td>000C</td>
<td>CDF</td>
</tr>
<tr>
<td>OOCOC</td>
<td>SCALR</td>
<td>0014</td>
<td>IBA</td>
</tr>
<tr>
<td>OOCOC</td>
<td>KGMB</td>
<td>0011</td>
<td>KGMB</td>
</tr>
</tbody>
</table>

BLANK COMMON (0 WORDS)

ENTRY POINTS:

OOCOC GETM

EXTERNAL SUBPROGRAMS REQUIRED:

ENCID SITBR STAT F1101 F1103 F1105 98CDREAD 918DATA

HIGHEST ERROR SEVERITY: C (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>125</td>
<td>CCC11</td>
</tr>
</tbody>
</table>

TOTAL PROGRAM: 173
SUBROUTINE GETP(ITAPE,AX,NX,NY,NZ,NW,CATAX,CATAY,CATAZ,DATAZ,DATAN, 
1 FLAT,RLANG,KGDA,KGMB,KGYR,KGHR,IE8C)
C
C SUBROUTINE GETP, FOR INPUT OF SEISMIC REFRACTION 
PROFILE DATA AT SFFMT FORMAT
C
C SSF(32) LP TO READ SFFMT DATA ON TWO CARDS
C SSF(33) LP TO WRITE SFFMT DATA ON TWO CARDS
C
C DIMENSION PLT(15),IDES(6)
C DATA KG,NEN,KS/19 'T','H','S'/
C
C USES SUBROUTINES EVIL, ISW, STAT ,DMT8R
C
C***************************************************************************
C***************************************************************************
C
C IIIN = 105
C IINUT = 108
C KGDA=C
C KGMB=C
C KGYR=0
C
C READING OF TORONTO WORLD SEISMIC REFRACTION Compilation
C 10 CONTINUE
C 12 READ(ITAPE,99C)REC,ISTA,KEY,LAT,LM,LNM,KMS,LSMG,LBM,
1 KEK,K1,K2,K3,K4,K5,K6,K7,K8,
2 IMANT,NELEV,N2,N3,N4,MET,ISYR,DESC,CLN,STHI,CRV,
3 WGT,AVTN,CRV,GTW,AVTH
C 99C FORMAT(I1,I4,A1,N1,A2,N1,A1,N1,A2,I8(I2,I3),I2,I4,A1, 
1 I1,I2,6A2,2F4.1,F3,1.2F6.0,1X,F3,1.2F6.0,5X)
G0 TO 18
C 15 READ(ITAPE,991)REC,ISTA,KEY,LAT,LM,LNM,KMS,LSMG,LBM,
1 KEK,K1,K2,K3,K4,K5,K6,K7,K8,
2 IMANT,NELEV,N2,N3,N4,MET,ISYR,DESC,CLN,STHI,CRV,
3 WGT,AVTN,CRV,GTW,AVTH
C 991 FORMAT(I1,I4,A1,N1,A2,N1,A1,N1,A2,I8(I2,I3),I2,I4,A1, 
1 I1,I2,6E2,2F4.1,F3,1.2F6.0,1X,F3,1.2F6.0,5X)
G0 TO 18
C CALL STAT(I)
C CALL EVIL(IINUT,I,IBAD,KDA,KMB,KGYR,ISTAB)
C IF (IBAD) 10 20, 900
C
C CHECKING IF KEY = 9
C 19 IF(KEY=9)20,10,20
C 20 ELEV=NELEV
C 46 ELEV=ELEV * 0.01
C 47 VMAT=IMANT
C 48 VMAT=VMAT * 0.1
C 49 KGHR=ISTA
C 50 IF(N1=2)70,60,70
C 51 C SEA SEISMIC PROFILE
C 52 60 VELH= 1.5
C 53 HEIGT=ELEV
C G0 TO 80
C 55 C LAND SEISMIC PROFILE
C 56 70 VELH= 0.0
C 57 HEIGT=ELEV
C 58 C MAIN PLOTTING LOOP
C 59 80 RLATM = LATM
60. \texttt{RLAT} = \texttt{LAT}
61. \texttt{RLAT} = \texttt{DMTB(LAT,RLAT)}
62. \texttt{RLONG} = \texttt{DMTB(LONG,RLAT)}
63. IF (\texttt{KNS} = \texttt{NNS}) 54, 52, 54
64. \texttt{RLAT} = \texttt{RLAT}
65. IF (\texttt{KEW} = \texttt{NEW}) 58, 56, 58
66. \texttt{RLONG} = \texttt{RLONG}
67. 58 \texttt{PLT(1)} = \texttt{PLT}
68. \texttt{PLT(2)} = \texttt{HEIGT}
69. \texttt{PLT(3)} = \texttt{VMAAT}
70. \texttt{PLT(4)} = \texttt{DINE}
71. \texttt{PLT(5)} = \texttt{THIK}
72. \texttt{PLT(6)} = \texttt{CRVH}
73. \texttt{PLT(7)} = \texttt{GTNH}
74. \texttt{PLT(8)} = \texttt{AVTH}
75. \texttt{PLT(9)} = \texttt{AVTH}
76. \texttt{PLT(10)} = \texttt{AVTH}
77. \texttt{PLT(11)} = \texttt{AVTH}
78. \texttt{C SELECTING DATA TO BE PLOTTED}
79. IF (\texttt{NX}) 110, 120, 110
80. \texttt{DATA} = \texttt{PLT(NX)}
81. \texttt{DATA} = \texttt{PLT(NY)}
82. \texttt{DATA} = \texttt{PLT(NZ)}
83. \texttt{DATA} = \texttt{PLT(NN)}
84. \texttt{IE80} = \texttt{C}
85. \texttt{RETURN}
86. \texttt{IE80} = \texttt{1}
87. \texttt{RETURN}
88. \texttt{END}
ENTRY POINTS:
COCCOC GETP

EXTERNAL SUBPROGRAMS REQUIRED:

EXTERNAL SUBPROGRAMS REQUIRED:

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS  HEX WORDS
GENERATED CODE: 298  CO12B
CONSTANTS: 4  CO004
LOCAL VARIABLES: 80  CO05C
TEMPs: 17  CO111
TOTAL PROGRAM: 400  CO19C
SUBROUTINE GETS

DIMENSION PLT(20)
DATA IFLAG/0/
IF (IFLAG NE 1) GO TO 30
IFLAG = 1

SSW(3) UP TO ADD CURRENT VELOCITIES TO SHIP'S VELOCITIES
SSW(12) UP TO LIST DATE IDENTIFICATION

LSES SUBROUTINES ENDIB(DUMMY), SHTV, EVIL, STAT
ASSUME STAT INITIALIZED IN MAIN PROGRAM

OUTPUT 'GETS VERSION 28 MAY 1974 FOR 67 GRAVITY FORMULA'

IFLAG = 0
CALL ENDIB
READ(ITAPE,12)IREC1,KGCA,KGMB,KGVR,KGHM,IFIP
IF (IREC1 NE 1) GO TO 30
CALL STAT(I)
CALL EVIL(I)
FORMAT(110,13)
IF (ITEST NE 3) GO TO 45
WRITE(JIOLT20)
FORMAT('EOR')
GOTO 50

CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF
THE 1967 INTERNATIONAL GRAVITY FORMULA
AND NEW GEODETIC REFERENCE SYSTEM

CONTINUE
IREC1 = IREC1 + 1
DG = 3.2E13*ABS(SIN(ABS(RLAT))**2)
IF (DKFA NE 9999) 57, 58, 59
DKFA = DKFA + DG
57
IF (DKVG NE 9999) 59, 70, 70
DKVG = DKVG + DG
59
GO TO 70
52
IF (IREC1 NE 9) 53, 62, 50
53
READ(ITAPE,64)IREC9,12,121,ITEST
54
FORMAT(110,13)
IF (ITEST NE 6560) 56, 57, 58
WRITE(JIOLT20)
FORMAT('EOR')
56
FORMAT(110,13)
57
GOTO 50
58
GOTO 50
59

58C IF (ITEST=6665) 68, 65, 68
61C IEXD=1
62C RETURN
63C WRITE(I16LT, 69)
64C FORMAT('IREC1=91')
65C G8 TO 50
66C IF(ISW(12)) 73, 73, 71
67C WRITE(I16LT, 72) KGDA, KGMB, KGYR, KGHM
68C FORMAT('DATE=1,313,15')
69C CONTINUE
70C USE CURRENT VELOCITIES
71C IF(ISW(3)) 80, 80, 75
72C VV=VV+(FLBAT(KCVN)*O*1)
73C VE=VE+(FLBAT(KCVE)*O*1)
74C KK=0
75C CALL SHTV(VN,VE,SPEED,XHEAD,KK)
76C E8TV=FLBAT(KETV)*O*1
77C TMAG=(MAG1*10000)+MAG2
78C KMAG2=(MAG2/1000)*1000
79C XMAG=MAG2*KMAG2
80C XREG=0*0
81C XKRES=0*0
82C XLDM=KCDM-MTDC
83C XLDF=XLDM*C+54681
84C PLT(1)=KGHM
85C PLT(2)=KKCDM
86C PLT(3)=XKFA
87C PLT(4)=XKBG
88C PLT(5)=SPEED
89C PLT(6)=XHEAD
90C PLT(7)=E8TV
91C PLT(8)=MT
92C PLT(9)=XMAG
93C PLT(10)=XREG
94C PLT(11)=XKRES
95C PLT(12)=SPEED
96C PLT(13)=KKCDM
97C PLT(14)=E8TV
98C PLT(15)=XKFA
99C PLT(16)=XHEAD
100C PLT(17)=TMAG
101C PLT(18)=XLDM
102C PLT(19)=XLDF
103C KGDA=KGDA
104C KGMB=KGMB
105C KGYR=KGYR
106C KGHM=KGHM
107C CHECKING FOR INVALID VALUES
108C IF(NX=2)1105,607,606
109C IF(NX=13)605,607,605
110C IF(KCDM)115,1C,105
111C IF(NX=2)115,610,615
112C IF(KFA=998)105,10,10
113C IF(NX=4)115,620,640
114C IF(KBG=998)105,10,10
115C IF(NX=17)105,650,105
116C IF(MAG1=10,10,105
117C SELECTING DATA TO BE PLOTTED
118C IF(NX)110,120,110
119C CATAX=PLT(NX)
120: DATAFLT(NY)
121: DATAZFLT(NZ)
122: DATAWFLT(NW)
123: X BLTFUT DATAx
124: RETURN
125: END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LBC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LBC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LBC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>SRCB</td>
<td>INTR</td>
<td>0</td>
<td>0</td>
<td>DATA</td>
<td>SCR</td>
<td>CALR</td>
<td>00047</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DAY</td>
<td>SRCB</td>
<td>INTR</td>
<td>0026</td>
<td>0</td>
<td>DATA</td>
<td>SCR</td>
<td>CALR</td>
<td>00032</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENDB</td>
<td>SRCB</td>
<td>EXTR</td>
<td>0</td>
<td>0</td>
<td>EBDTV</td>
<td>SCR</td>
<td>CALR</td>
<td>00034</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOW</td>
<td>SRCB</td>
<td>INTR</td>
<td>0</td>
<td>0</td>
<td>GETS</td>
<td>SRCB</td>
<td>CALR</td>
<td>00036</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IER</td>
<td>SCR</td>
<td>CALR</td>
<td>00037</td>
<td>V</td>
<td>IFLAG</td>
<td>SCR</td>
<td>CALR</td>
<td>00039</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I0R</td>
<td>SCR</td>
<td>CALR</td>
<td>00040</td>
<td>V</td>
<td>IRECI</td>
<td>SCR</td>
<td>CALR</td>
<td>00044</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I1H</td>
<td>SCR</td>
<td>CALR</td>
<td>00045</td>
<td>V</td>
<td>ITAPE</td>
<td>SCR</td>
<td>CALR</td>
<td>00047</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I2I</td>
<td>SCR</td>
<td>CALR</td>
<td>00048</td>
<td>V</td>
<td>IZ1</td>
<td>SCR</td>
<td>CALR</td>
<td>00050</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KCCM</td>
<td>SCR</td>
<td>CALR</td>
<td>00051</td>
<td>V</td>
<td>KCVE</td>
<td>SCR</td>
<td>CALR</td>
<td>00053</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KETV</td>
<td>SCR</td>
<td>CALR</td>
<td>00054</td>
<td>V</td>
<td>KFA</td>
<td>SCR</td>
<td>CALR</td>
<td>00056</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KDAB</td>
<td>SCR</td>
<td>CALR</td>
<td>00057</td>
<td>V</td>
<td>KGMB</td>
<td>SCR</td>
<td>CALR</td>
<td>00059</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KGYR</td>
<td>SCR</td>
<td>CALR</td>
<td>00060</td>
<td>V</td>
<td>KGMB</td>
<td>SCR</td>
<td>CALR</td>
<td>00062</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KY</td>
<td>SCR</td>
<td>CALR</td>
<td>00063</td>
<td>V</td>
<td>KV</td>
<td>SCR</td>
<td>CALR</td>
<td>00065</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M1Cu</td>
<td>SCR</td>
<td>CALR</td>
<td>00066</td>
<td>V</td>
<td>M1Cu2</td>
<td>SCR</td>
<td>CALR</td>
<td>00068</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDC</td>
<td>SCR</td>
<td>CALR</td>
<td>00069</td>
<td>V</td>
<td>MW</td>
<td>SCR</td>
<td>CALR</td>
<td>00071</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NY</td>
<td>SCR</td>
<td>CALR</td>
<td>00072</td>
<td>V</td>
<td>NZ</td>
<td>SCR</td>
<td>CALR</td>
<td>00074</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLAT</td>
<td>SCR</td>
<td>CALR</td>
<td>00075</td>
<td>V</td>
<td>MALT</td>
<td>SCR</td>
<td>CALR</td>
<td>00077</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIN</td>
<td>SCR</td>
<td>CALR</td>
<td>00078</td>
<td>V</td>
<td>SPEED</td>
<td>SCR</td>
<td>CALR</td>
<td>00080</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMAG</td>
<td>SCR</td>
<td>CALR</td>
<td>00081</td>
<td>V</td>
<td>VE</td>
<td>SCR</td>
<td>CALR</td>
<td>00083</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XHEAD</td>
<td>SCR</td>
<td>CALR</td>
<td>00084</td>
<td>V</td>
<td>XB7G3</td>
<td>SCR</td>
<td>CALR</td>
<td>00086</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XFA</td>
<td>SCR</td>
<td>CALR</td>
<td>00087</td>
<td>V</td>
<td>XRES</td>
<td>SCR</td>
<td>CALR</td>
<td>00089</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XREG</td>
<td>SCR</td>
<td>CALR</td>
<td>00090</td>
<td>V</td>
<td>XUDF</td>
<td>SCR</td>
<td>CALR</td>
<td>00092</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
<th>LABEL</th>
<th>LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000C</td>
<td>0</td>
<td>00001C</td>
<td>0</td>
<td>00002C</td>
<td>0</td>
<td>00003C</td>
<td>0</td>
<td>00004C</td>
<td>0</td>
<td>00005C</td>
<td>0</td>
</tr>
<tr>
<td>00006C</td>
<td>0</td>
<td>00007C</td>
<td>0</td>
<td>00008C</td>
<td>0</td>
<td>00009C</td>
<td>0</td>
<td>00010C</td>
<td>0</td>
<td>00011C</td>
<td>0</td>
</tr>
<tr>
<td>00012C</td>
<td>0</td>
<td>00013C</td>
<td>0</td>
<td>00014C</td>
<td>0</td>
<td>00015C</td>
<td>0</td>
<td>00016C</td>
<td>0</td>
<td>00017C</td>
<td>0</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (66 ADDRESS):

- CCCCC GETS
- COCC15 KVN
- COCC1F KVE
- COCC25 K2G
- COCC2E KGYR
- COCC31 XE
- COCC32 XEED
- COCC33 XEED
- COCC34 XEAD
- COCC35 XEAG
- COCC36 XREG
- COCC37 KK
- COCC38 XPAG

FLACK COPPON (CC ADDRESS)

ENTRY POINTS:

CCCCC GETS
INTRINSIC SUBPROGRAMS USED:
ABS  FLOAT  SIN

EXTERNAL SUBPROGRAMS REQUIRED:
ENCID  EVIL  ISH  SHTY  STAT  F1101  F1102
F1104  F1105  F1106  F1108  9BCDREAD  9BCDWRIT  9ENDBL  918DATA
9SITBR  9PRINT  9RTSI  9SETLPN  9SIN

NUMBER OF X CARDS IGNORED: 1
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>GENERATED CODE</td>
<td>363</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>10</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>66</td>
</tr>
<tr>
<td>TEMPS</td>
<td>17</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>456</td>
</tr>
</tbody>
</table>
SUBROUTINE GETST(ITAPE, AX, NY, NZ, NH, DATAX, DATAY, DATAZ, DATAK,
1   FLAT, RLONG, DA, JM, JYR, JHM, IEDD)

VERSION OF 30 JUNE 1971
CUMOY ROUTINE WHILE AWAITING A WORKING VERSION

RETURN
END
SUBROUTINE GETV(ITAPE, NX, NY, Z, NW, DATA, VDATA, WDATA, VDATA, WDATA,
       LAT, RLAT, KGD, KGMB, KGMB, KGMB, KGMB, IE8D)

  VERSION OF 29 JUNE 1971

  SUBROUTINE GETV, READS WORLD VOLCANOE CATALOGUE COMPIALATION

  DIMENSION PLT(3)
  DIMENSION ID(3)
  IC = 105
  IC8 = IC
  IE8D = C
  KGDA = 0

  CONTINUE
  READ(ITAPE, 74) IAREA, IGBA, ISUBA, IDASH, ING, LAT, RLAT, KNS,
  1, LON, RLON, KEW, IHEJ, IPT, IPAGE, ITYPE, ICHEM, ID
  74 FORMAT(I3, A1, I2, A1, I2, I1, X, I2, A1, I1, X, I3, F9.2, A1, I1, X,
  1, I5, 1X, I2, 1X, I3, 1X, A1, 1X, A1, 15X, 5A4)

  CALL STAT(I)
  CALL EVIL(I8UT, 1, IBAC, KGDA, KGMB, KGMB, KGMB, KGMB, KGMB, KGMB, KGMB) 
  20 IF (IBAC) = 10, 53, 900

  CONTINUE
  KGMB = IAREA
  KGMB = ISUBA
  KGMB = ING
  CALL NAVIN(LAT, RLAT, KNS, LON, RLON, KEW, RLAT, RLON)
  KGDA = KGDA
  KGMB = KGMB
  KGMB = KGMB
  KGYR = KGYR
  KGMB = KGMB
  PLT(1) = ING
  PLT(2) = IHEI
  PLT(3) = (IPT + 100) + IPAGE
  IF (NX) = 80, 85, 80
  DATA = PLT(NX)
  DATA = PLT(NY)
  DATA = IHEI
  DATA = INC
  RETURN
  900 IE8D = 1
  900 RETURN
  END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATEX</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC28</td>
<td>V DLMPY</td>
<td>DATEX</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC28</td>
<td>V DLMPY</td>
<td>DATEX</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC28</td>
<td>V DLMPY</td>
</tr>
<tr>
<td>CATAZ</td>
<td>R</td>
<td>SCALR</td>
<td>*CCC2A</td>
<td>V DLMPY</td>
<td>EVIL</td>
<td>SPRQG</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td>GETV</td>
<td>R</td>
<td>SCALR</td>
<td>*00029</td>
<td>V DLMPY</td>
</tr>
<tr>
<td>GETV</td>
<td>R</td>
<td>SPRQG</td>
<td>CCC0C</td>
<td>F</td>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>CCC1D</td>
<td>V</td>
<td>IAREA</td>
<td>I</td>
<td>SCALR</td>
<td>0000D</td>
<td>V</td>
</tr>
<tr>
<td>ISB1E</td>
<td>I</td>
<td>ARRAY</td>
<td>CCC1E</td>
<td>V</td>
<td>1</td>
<td>I</td>
<td>SCALR</td>
<td>CCC1C</td>
<td>V</td>
<td>ICBM</td>
<td>I</td>
<td>SCALR</td>
<td>0000E</td>
<td>V</td>
</tr>
<tr>
<td>IC</td>
<td>I</td>
<td>SCALR</td>
<td>CCC0C</td>
<td>V</td>
<td>1</td>
<td>I</td>
<td>SCALR</td>
<td>CCC1C</td>
<td>V</td>
<td>IEED</td>
<td>I</td>
<td>SCALR</td>
<td>*00032</td>
<td>V DLMPY</td>
</tr>
<tr>
<td>ITF</td>
<td>I</td>
<td>SCALR</td>
<td>CCC18</td>
<td>V</td>
<td>1</td>
<td>I</td>
<td>SCALR</td>
<td>CCC1B</td>
<td>V</td>
<td>IOUT</td>
<td>I</td>
<td>SCALR</td>
<td>0000C</td>
<td>V</td>
</tr>
<tr>
<td>IND</td>
<td>I</td>
<td>SCALR</td>
<td>CCC11</td>
<td>V</td>
<td>1</td>
<td>I</td>
<td>SCALR</td>
<td>CCC1A</td>
<td>V</td>
<td>IPT</td>
<td>I</td>
<td>SCALR</td>
<td>00019</td>
<td>V</td>
</tr>
<tr>
<td>ITT1A</td>
<td>I</td>
<td>SCALR</td>
<td>CCC0F</td>
<td>V</td>
<td>1</td>
<td>I</td>
<td>SCALR</td>
<td>CCC23</td>
<td>V</td>
<td>ITYPE</td>
<td>I</td>
<td>SCALR</td>
<td>00018</td>
<td>V</td>
</tr>
<tr>
<td>KE1</td>
<td>I</td>
<td>SCALR</td>
<td>CCC17</td>
<td>V</td>
<td>1</td>
<td>K</td>
<td>SCALR</td>
<td>CCC2E</td>
<td>V</td>
<td>KGDA1</td>
<td>I</td>
<td>SCALR</td>
<td>0001F</td>
<td>V</td>
</tr>
<tr>
<td>KEP81</td>
<td>I</td>
<td>SCALR</td>
<td>CCC31</td>
<td>V</td>
<td>1</td>
<td>K</td>
<td>SCALR</td>
<td>CCC2V</td>
<td>V</td>
<td>KGDA1</td>
<td>I</td>
<td>SCALR</td>
<td>*0002F</td>
<td>V DLMPY</td>
</tr>
<tr>
<td>KF81</td>
<td>I</td>
<td>SCALR</td>
<td>CCC2C</td>
<td>V</td>
<td>1</td>
<td>K</td>
<td>SCALR</td>
<td>CCC30</td>
<td>V</td>
<td>KGDA1</td>
<td>I</td>
<td>SCALR</td>
<td>00021</td>
<td>V</td>
</tr>
<tr>
<td>KNS</td>
<td>I</td>
<td>SCALR</td>
<td>CCC14</td>
<td>V</td>
<td>1</td>
<td>K</td>
<td>SCALR</td>
<td>CCC12</td>
<td>V</td>
<td>LBG1</td>
<td>I</td>
<td>SCALR</td>
<td>00015</td>
<td>V</td>
</tr>
<tr>
<td>NAV1</td>
<td>SPRQG</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td>NAV1</td>
<td>SCALR</td>
<td>*00026</td>
<td>V DLMPY</td>
<td>NAV1</td>
<td>SCALR</td>
<td>000024</td>
<td>V DLMPY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAV2</td>
<td>SCALR</td>
<td>*00027</td>
<td>V DLMPY</td>
<td></td>
<td>NAV2</td>
<td>SCALR</td>
<td>*00027</td>
<td>V DLMPY</td>
<td>NAV2</td>
<td>SCALR</td>
<td>000019</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>R1AT1</td>
<td>R</td>
<td>SCALR</td>
<td>*00025</td>
<td>V DLMPY</td>
<td>R1AT1</td>
<td>R</td>
<td>SCALR</td>
<td>CCC13</td>
<td>V</td>
<td>R1BM</td>
<td>R</td>
<td>SCALR</td>
<td>00016</td>
<td>V</td>
</tr>
<tr>
<td>R1AG1</td>
<td>R</td>
<td>SCALR</td>
<td>*0002D</td>
<td>V DLMPY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HEX LBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HEX LBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HEX LBC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL VARIABLES (35 WORDS):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCC0C GETV</td>
<td>CCC01 PLT</td>
<td>000C6 ID</td>
<td>000C6 IIN</td>
<td>000C6 IIN</td>
<td>000C6 IIN</td>
<td>000C6 IIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCC0E IC8M</td>
<td>CCC0F ISAE</td>
<td>000C10 IDASH</td>
<td>000C11 IKB</td>
<td>000C12 LAT</td>
<td>000C12 LAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCC14 KNS</td>
<td>CCC15 LBG</td>
<td>000C16 R1BM</td>
<td>000C17 KEW</td>
<td>000C18 IHEIT</td>
<td>000C19 IPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCC1A IPAGE</td>
<td>CCC1B ITYPE</td>
<td>000C1C ICHEM</td>
<td>000C1C I</td>
<td>000C1C I</td>
<td>000C1C I</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCC20 KGPHB</td>
<td>CCC21 KGPHB</td>
<td>000C22 KGPHB</td>
<td>000C22 KGPHB</td>
<td>000C22 KGPHB</td>
<td>000C22 KGPHB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLANK COMMON (6 CHARACTERS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENTRY POINTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCC0C GETV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTERNAL SUBPROGRAMS REQUIRED:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EVIL</td>
<td>NAV1</td>
<td>STAT</td>
<td>F:1101</td>
<td>F:1103</td>
<td>F:1105</td>
<td>SBCREAD</td>
<td>910DA A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIBULLSA</td>
<td>SIBUR</td>
<td>SIBUPA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**HIGHEST ERROR SEVERITY:** 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GENERATED CODE:</strong></td>
<td>140</td>
<td>CCCBC</td>
</tr>
<tr>
<td><strong>CONSTANTS:</strong></td>
<td>0</td>
<td>CCCCC</td>
</tr>
<tr>
<td><strong>LOCAL VARIABLES:</strong></td>
<td>35</td>
<td>CCC23</td>
</tr>
<tr>
<td><strong>TEMPS:</strong></td>
<td>17</td>
<td>CCC11</td>
</tr>
<tr>
<td><strong>TOTAL PROGRAM:</strong></td>
<td>192</td>
<td>CCCCC</td>
</tr>
</tbody>
</table>
SUBROUTINE GETX(ITAPE, NX, NY, NZ, NW, DATAX, DATAY, DATAZ, DATAW,
        1 RLAT, RLONG, KGDA, KGM8, KGYR, KGMY, IE8D)

VERSION OF 26 SEPT 1972, DUMMY ROUTINE
SUBROUTINE GETX, FOR READING VARIABLE DATA INPUT, WRITE YOUR
OWN GETX ROUTINE

DIMENSION PLT(10)
IIN = 105
IISLT = 108
IE8D = C
KGDA = 0
KGM8 = C
BLTPLT ' SUBROUTINE GETX IS A DUMMY ROUTINE, CALL EXIT'
CALL EXIT
RETURN
END
### Local Variables (13 Words):

- CCCC GETX
- CCCC PLT
- CCCCB IIIN
- CCCCC IIBUT

### Blank Common (0 Words)

### Entry Points:

- CCCC GETX

### External Subprograms Required:

- EXIT
- F:108
- SENDCO
- 9PRINT
- 9SETUP

### Highest Error Severity: G (No Errors)

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
</tr>
</tbody>
</table>

| GENERATED CODE: | 49 | CCCC31 |
| CONSTANTS: | 0 | CCCC00 |
| LOCAL VARIABLES: | 13 | CCCC00 |
| TEMPS: | 17 | CCCC11 |

### Total Program: 79 CCCC9F
SUBROUTINE GETY OF MARCH 19, 1972

SUBROUTINE GETY (ITAPE, NX, NY, KDA, KMB, KMR, KHM, IEOD)

INTEGER SN, KDA, KMB, KMR, KHM, IEOD

DIMENSION PLT (5)

IF (IFLAG == 1) 400, 10, 100

ILN = 105
IOUTPUT = 108
IFLAG = 1

CONTINUE

READ (ITAPE, 65)

CALL STAT(I)

CALL EVIL (J, 18AC, KDA, KMB, KMR, KHM)

IF (IBAD) 411, 53, 96

CONTINUE

FORMAT (6X, 12, 12, 14, 3X, F5.3, A1, F6.3, A1, F3.0, 3F3.2)

CALL CNAV (OLAT, KEK, COLON, KHE, RLAT, RLONG, KL)

PLT(1) = KDA * 10000 + KMB * 100 + KMR
PLT(2) = DEPT
PLT(3) = AMAG

IF (NX > 80) 80, 80, 80

DATA = PLT (NX)

DATA = PLT (NX)

DATA = DEPT

DATA = AMAG

RETURN

IEOD = 1

RETURN

END
SUBROUTINE GSIN(ITAPE, JTAPE, DERFA, IREC, IFFC, IA, IFBC)

VERSION 8 MAY 1975, ADD INITIALIZATION ZEROS AND OUTPUT COMMENTS ON GRAVITY FORMULA

VERSION OF 11 DEC 1974, TO ADD HANDLING OF PROJ OUTPUT

VERSION OF 1 OCT 1974, FOR 1967 GRAVITY FORMULA

VERSION OF 26 JULY 1973, START CONVERSION TO 1967 G FORMULA

VERSION OF 26 APR 72, TEMPORARY MODIF DUE TO BACKWARD CODE / MANAGE

VERSION OF 27 DECEMBER 71

VERSION OF 22 DECEMBER 1971

GINIT VERSION NOV 12/1971

INPUT + NEW GSUM FORMAT + PREVIOUS GSUM FORMAT

OUTPUT + NEW GSUM FORMAT

MODIF ON NOV 12/1971 BY MONET J.M. TO INCLUDE:

- USE OF DATA LOCATION TABLE
- USE OF LABEL TAPES

MODIFIED AUG 16 1971 BY FELINSBEE TO CORRECT ERROR ON CALLING ARGS OF GBLKI

VERSION OF JUNE 1971 DOES NOT WRITE EOF ON TO OUTPUT DEVICE

VERSION OF JUNE 29/71 MODIFIED JUNE 28 TO READ OR WRITE BLOCKED DATA BY A FELINSBEE

VERSION OF APRIL 1971 TO OPTIONALLY SUPPRESSREWIND OF ITAPE AND JTAPE

SUBROUTINE GSIN, FOR GSUM FORMATTED DATA

VERSION WITH DESIGNATION OF INPUT AND OUTPUT MAGNETIC TAPES
BY USE OF SUBROUTINE MOUNT

SSW(12) UP TO LIST DATE IDENTIFICATION

SSW(20) UP TO OUTPUT ON HIGH SPEED PRINTER ONLY

SSW(27) UP TO SUPPRESSREWIND OF TAPE AT START OF JOB

*1 SUPPRESSREWIND OF ITAPE

*2 SUPPRESSREWIND OF ITAPE

*4 SUPPRESSREWIND OF BOTH ITAPE AND JTAPE

SSW(29) = 1 = TO READ AND TEST FOR SELECTED SOURCE CODE NUMBERS TO BE PROCESSED

*2 = TO READ AND TEST FOR SELECTED SOURCE CODES

NUMBERS TO BE SKIPPED

SSW(30) UP FOR INPUT DATA ON CARDS

SSW(31) UP TO OUTPUT DATA ON CARDS

SSW(40) UP TO PROCESS WITH BOUNDS USING CLT

*4 = PROCESS WITHOUT BOUNDS

*1 = PROCESS WITH BOUNDS USING THE DATA LOCATION TABLE

SSW(60) UP TO PROCESS ONLY DATA WITH IFFC = ABSTRACTOR OUTPUT

SSW(61) UP TO REPLACE FRA, ELEV, LAT, LONG WITH AVERAGE VALUES

USES ENDMIDUMMY, EVID, STAT, ISM

ASSUME ISM AND STAT INITIALIZED IN MAIN PROGRAM

DIMENSION IDBIN(20), IBK(20), ITRK(20), IDENS(20), IDESC(17,20)

DIMENSION IDTGT(20), JBK(20), JTRK(20), JDENS(20), JDESC(17,20)

DIMENSION PLT(20), IA(35), IAP(9), IASM(35), ISRC(16)

DATA ITER, ITERP/IATIP, IEPTP/
411

IF(KK)420,440,410

C ------ G6UM INITIAlISATION LOGIC

C

400 IN = 105
401 IBLT = 108
402 IFLA=106
403 ICS = 100

C

DEGRA=1.7*329862
RDEG=57.25578

C

NEF = NO. OF FILE NOW BEING PROCESSED
NEF = 1
FILE = 1
NO. OF RECORDS NOW WRITTEN ON PRESENT OUTPUT TAPE
NRECT = NZERO

C

OUTPUT 'GINAT BF 8 MAY 1975'
MAXT = 125000
ILI = ICL / IDL = 100
IF(ISW(40)=NE+1) CALL ENCLT(J,CLAT8,CLAB8,CLBE,DLOR,IDL,ILL,0)
NEF = NEF + I / IFILE = J - 1
NOUT = NEF / IFILE
NZERO = 0
KBDA$ = NZERO
KGB$= NZERO
KGB$ = NZERO
IREC1 = 1
IREC2 = 2
IRECIN = 0

C

C ------ CHECK SSW(29) TO SEE IF SOURCE CODE NUMBERS

C ARE TO BE READ FOR DATA SELECTION

C

C

407 IF(ISW(29)=EGC) GO TO 1405
408 READ(IN,9010)SRC
900 FORMAT(1615)
1000 IF(ISW(29)=EGC) WRITE(11BLT,912)ISRC GO TO 1405
1010 WRITE(11BLT,913)ISRC
1020 913 FORMAT(161,1X,SKIPPED SOURCE CODES = '1615)
1030 912 FORMAT(161,1X,SELECTED SOURCE CODES = '1615)

C

C ------

1405 IF(ISW(40)=NE+1) GO TO 72C
1408 IF(ISW(30)=404*404*728
1054 404 J = 1
1104 READ(IN,406) ICLT(J),1BK(J),ITK(J),IDEAS(J)
1114 (DESC(K,J),K=1,17)
1124 406 FORMAT(A4,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1,1X,A1)
1134 IF(IDIN(J)+NE+ITER) J+1,10 GO TO 405
1144 NEF = 1
1154 IFILE = 1
1164 OUTPUT NEF / IFILE
1174 72C IF(ISW(31)=408*408*1410
1184 408 J = 1
1194 405 READ(IN,406) ICLT(J),1BK(J),ITK(J),IDEAS(J)
120 1 (IDESC(K),K=1,17)
121 IF(IDTOT(J)=E*ITER0) J=J+1 GO TO 409
122  NECD=1
123 JFILE* = 1
124 OUTPUT NECD,JFILE
125 1410 IF(ISW(30))1412-1412=1414
126 1412 CONTINUE
127 IF(ISW(40)=E*G+G070810
128 READ(IDESC,G070810) IDTOT(J),IBK(J),ITK(J),IDENS(J),
129 (IDESC(K),K=1,17)
130 810 IF(IDTOT(J)=E*ITER0) GO TO 1414
131 CALL MOUNT(I,TAPE,IDTOT(J))
132 WRITE (I,1413) IDTOT(J),IBK(J),ITK(J),IDENS(J),
133 (IDESC(K),K=1,17)
134 1413 FORMAT (1X,A4,1X,A1,1X,J,1X,J,1X,J)
135 IF(ISW(27)=E*G+G070890
136 CONTINUE
137 IF(ISW(27)=E*G+G070890
138 CONTINUE
139 IF(ISW(31)=E*G+G070890
140 CALL MOUNT(I,TAPE,I)
141 WRITE (I,1413) IDTOT(I),IBK(I),ITK(I),IDENS(I),
142 (IDESC(K),K=1,17)
143 IF(ISW(27)=E*G+G070890
144 CONTINUE
145 IF(ISW(27)=E*G+G070890
146 CALL MOUNT(I,TAPE,I)
147 CONTINUE
148 RETURN
149 C----- GSUM INPUT LOGIC
150 C
151 410 CONTINUE
152 5C CALL ENDB
153 IF(ISW(30)=E*G+G070810
154 READ(I,1413) IDTOT(J),IBK(J),ITK(J),IDENS(J),
155 (IDESC(K),K=1,17)
156 1413 FORMAT (1X,A4,1X,A1,1X,J,1X,J,1X,J)
157 IF(ISW(31)=E*G+G070890
158 CONTINUE
159 IF(ISW(31)=E*G+G070890
160 CALL STAT(I)
161 CALL EVIL(I,OUT,1,IBAD,KGDA,KGMB,KGYK,KGHM,
162 IF(IBAD)=50,53,579
163 C CHECKING IF IREC = 2 OR 1
164 C
165 IF(REC=2 OR 1) GO TO 600
166 IF(REC=2 OR 1) GO TO 600
167 GO TO 70
168 C
169 C
170 C
171 IF(REC=2 OR 1) GO TO 600
172 IF(REC=2 OR 1) GO TO 600
173 GO TO 70
174 GO TO 70
175 GO TO 70
176 GO TO 70
177 GO TO 70
178 GO TO 50
179 C
CONVERSION OF 1930 INTERNATIONAL GRAVITY FORMULA TO THAT OF
THE 1967 INTERNATIONAL GRAVITY FORMULA
AND NEW GEODETIC REFERENCE SYSTEM

C 61C CONTINUE
C IF (ITRECIN+EGO) OUTPUT (CONVERTING TO 1967 GRAV FORMULA NOW)
C IRECIN = 1
C KK = 1
C CALL BBS (K977,GBSG,G6BS,KK)
C GBSKGBBS+140
C KK = 2
C CALL BBS (K977,GBSG,G6BS,KK)
C RLAT = DLAT*DEGRA
C DG=3+2*(13+6*(SIN(ABS(RLAT)*)**2))
C IF(FA=990.0) 611 612 612
C 611 FAFA+DG
C 612 IF(BG=990.0) 613 70 70
C 613 BG+BG+DG
C 615 G8 TO 7C
C 575 IF (NEF = IFILE) 576 577 577
C 576 NENEF = 1
C 201 IF(ISW(6)+1) 202 202
C 202 REAC(IDISC,IG60,IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
C 203 * (IDESC(K,NEF),K=1,17)
C 204 820 CALL MOUNT(ITAPE,IDTIN(NEF))
C 205 WRITE (ITBNOL,1413) IDTIN(NEF),IBK(NEF),ITK(NEF),IDENS(NEF),
C 206 1 (IDESC(K,NEF),K=1,17)
C 207 REWIND ITAPE
C 208 G8 TO 50
C 209 C END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
C 777 KK=9 1 RETURN
C 77 IC IF(ISW(12),13,73,71
C 7IC 71 WRITE(IOUT,1739,KGDA,KGMD,KGYR,KGWM
C 72 72 FORMAT('DATE=',13,15)
C 73 73 IF(ISW(29)+1) 74 74
C 74 IF(ISW(29)+1) 75 75
C 75 CALL MAUNT(ITAPE,IO)
C 209 WRITE (IO,1443) ITBNOL(NEF),TBK(NEF),ITK(NEF),IDENS(NEF),
C 210 (IDESC(K,NEF),K=1,17)
C 211 REWIND ITAPE
C 212 G8 TO 50
C 213 C PROCESS ONLY SELECTED SOURCE CODES
C 214 C
C 215 C 081650081216
C 216 IF(ISRC(J)+1) 217 217
C 217 IF(ISRC=ISRC(J)) 1650 650
C 218 CONTINUE
C 219 G8TO50
C 220 C IGNOR SELECTED SOURCE CODES
C 221 170C C01530=1,16
C 222 IF(ISRC(J)+1) 223 223
C 223 IF(ISRC=ISRC(J)) 1710 1710
C 224 CONTINUE
C 225 C
C 226 C USE DATA ONLY FOR IFCC = 4 (IE ABSTRACTER OUTPUT)
C 227 8CC IF(IFCC=4) 228 228
C 228 8CC IF(IFCC=4) 801 801
C 229 CONTINUE
C SET FA, BG, AND ELEV = AVERAGED VALUES FROM ABSTRACTER OUTPUT

C SET LLAT AND LLONG TO VALUES AT CENTER OF Grid AREA

105 CONTINUE

IF(ISW(61))10921092105 ¥

3 240° IF (1SW(61))10921092105 ¥

esi C SET FA, BG, ANC ELEV = AVERAGED VALUES FROM ABSTRACTER BUTINPUT GIN@14480 g

246 C Ser LaT ANC LONG Te ValVES aT CENTER OF GRIC AREA G1NO1490 4

243° 108 C@NTINLE GIN@1500 4

Z44e D6 8oe WK91s35 GI1N61510 . i

2456 80E IASH(UKISISL( IAC UK), #24) G1N01520 ; 2

z46e CALL BEOe ESM: LACT RS) GIN81530

247s DECADE (35.8¢3,1AFMT) CLAT»CLONG, AKEIGT, KAFA,KABG GIN®81540

248° "  -RLATSCLAT#OEGRA 3

250° RLONGsCLENG*DEGRA ;

2516 APFA®FLATI(KAFA) #001 G1N61580 ;

2526 ABGSFLBAT(KABG) #001 GIN81590 i

; 2530 FASAFA GIN61600 i

2 2546 BG=ABC GIN61610 ,

255° ELEVsAREIGT GIN@1620 4

2560 NUM@sYFBC ° GIN@1630 4

257° 10S CONTINLE GIN61640 3

258+ 418 CONTINVE G1N61650 3

2596 KGDABsKGCA GIN61660

260¢ KGMOA62kGMS GIN@1670

2616 KGYR@=KGYR : GIN61680

2626 KGHM6 =KGHM GIN61690

263° RETURN GIN@1700 4

264° C -------------- GSUM OUTPUT LOGIC

265° C

420 CALL ENDID

268° CLAT=RLAT*RADEG

269° CLONG=RLONG*RADEG

270° PLAT=CLAT+5C. 1 LTKEY=PLAT

271° PLONG=CLONG+180. 1 LGKEY=PLONG

272° CALL AREAK(DLAT,DLONG,IAKEY)

273° IF(ISW(31)+EQ,1) G0 TO 110

274° IF(ISW(26)+EQ,1) G0 TO 118

275° IF(ISW(31)+EQ,1) G0 TO 118

276° WRITE (TAPE,108) IREC2,C

277° CLAT,CLONG,ELEV,K977,BBSG,ICEP,FA,BC,TC,IELC,IGC,

278° RFA,IREGC,IPFC,IA,FBC,LTKEY,LYKEY,IAKEY

279° 11 FORMAT(11i14,31i14,29i4,6f7.2,13f6.1,215r,214f6.1,i4,1,4)

280° 1 212,6+1,12,35A,12,2F,11,213,12)

281° NRECT = NRECT + 1

282° IF (NRECT = MAXCT) G0 TO 85, 50, 90

283° 85 RETURN

284° 110 CONTINUE

285° WRITE (PLN,1469) IREC2,ISBC,KGDA,KGMG,KGYR,KGHM,

286° CLAT,CLONG,ELEV,K977,BBSG,ICEP,FA,BC,TC,IELC,IGC,

287° RFA,IREGC,IPFC,IA,FBC,LTKEY,LYKEY,IAKEY

288° 1469 FORMAT(11i14,31i14,29i4,6f7.2,13f6.1,215r,214f6.1,i4,1,4)

289° 1 212,6+1,10,12,35A,12,2F,11,213,12)

290° RETURN

5C NRECT = NZERO

291° WRITE (IOUT, 91) GIN@1870

292° 91 FORMAT (IPAXCT OUTPUT!)

293° END FILE TAPE

294° ENDING TAPE

295° NEND=KEND+1

296° IF(NEND=FILE)990,990,995

297° CALL MOUNT(,TAPE,IDOT(NEND))

298° 99C CALL MOUNT(TAPE,IDOT(NEND),BK(NEND),WK(NEND),JDS(NEND),)

299° WRITE (IOUT,1413) IDOT(NEND),BK(NEND),WK(NEND),JDS(NEND)
360  KK=9  RETURN
361  WRITE(I18UT,345)
362  345  FORMAT(1HO, "INCORRECT DLT TABLE = FOUND EBF WHILE PROCESSING"
363  "RECRODS")
364  KK=9  RETURN
365  WRITE(I18UT,346)
366  346  FORMAT(1HO, "INCORRECT DL T TABLE = FOUND END OF TAPE WHILE"
367  "SKIPPING RECORDS",/)
368  KK=9  RETURN
369  WRITE(I18UT,347)
370  347  FORMAT(1HO, "ERROR CONDITION WHILE SKIPPING RECORDS",/)
371  KK=9  RETURN
372  END
FUNCTION GINTF(RLAT)

GINTF CALCULATES THEORETICAL GRAVITY FROM INTERNATIONAL GRAVITY FORMULA 'EARTH AND ITS GRAVITY FIELD' HEISKANEN AND VENING MEINESZ 1958 PAGE 74 EXPANDED BY A FOLINSBEE USING FORMULA (SIN(X))**2 = (1-COS(2X))/2

VALLE RETURNS A G = 977CCC MGALS

10. A = ABS(RLAT)
11. GINTF = 3632*272*2586*157*COS(2**A)+2*885*COS(4**A)
12. RETURN
13. END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALR</td>
<td>CCCC01 V</td>
<td>1</td>
<td>ABS</td>
<td>R</td>
<td>SPROG</td>
<td>INTRIN</td>
<td></td>
<td>CBS</td>
<td>R</td>
<td>SPROG</td>
<td>INTRIN</td>
<td></td>
</tr>
<tr>
<td>GINTF</td>
<td>R</td>
<td>SCALR</td>
<td>CCCC00 V</td>
<td>1</td>
<td>GINTF</td>
<td>R</td>
<td>SPROG</td>
<td>CCCC00 P</td>
<td></td>
<td>RLAT</td>
<td>R</td>
<td>SCALR</td>
<td>#00002 V DUMMY</td>
<td></td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES** (2 WORDS):
- GCCC COMMON GINTF CCCC01 A

**BLANK COMMON** (0 WORDS)

**ENTRY POINTS:**
- CCCC COMMON GINTF

**INTRINSIC SUBPROGRAMS USED:**
- ABS
- CBS

**EXTERNAL SUBPROGRAMS REQUIRED:**
- 9COS 9SETUP1

**HIGHEST ERROR SEVERITY:** C (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>CCCC14</td>
</tr>
<tr>
<td>15</td>
<td>CCCC05</td>
</tr>
<tr>
<td>20</td>
<td>CCCC02</td>
</tr>
<tr>
<td>3</td>
<td>CCCC03</td>
</tr>
<tr>
<td>30</td>
<td>CCCC1E</td>
</tr>
</tbody>
</table>

**TOTAL PROGRAM:** 30 CCCC1E
FUNCTION GI67F(FLAT)
VERSION OF 25 APR 75 TO REDUCE FORMULA BY USING
EXPRESSION \( \sin(x)^2 \times \frac{1 - \cos(2x)}{2} \)
CALCULATES THEORETICAL GRAVITY FROM THE INTERNATIONAL FORMULA
ACCORDING TO RESOLUTION NO. 2 OF THE XIV TH GENERAL ASSEMBLY OF
THE I.E.S. 1967
FOR DETAILS OF THE FORMULA SEE PAGE 74 OF
GEODETIC REFERENCE SYSTEM 1967
VALLE RETURNED AS G=977000 MGALs
A=ABS(FLAT)
GI67F = 3621*9455 - 2592*9639*\cos(2*A) + 2*8683*\cos(4*A)
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R SCALR</td>
<td></td>
<td>CO001</td>
<td>V</td>
<td>1</td>
<td>G167F</td>
<td>R SPROG</td>
<td>INTRIN</td>
<td></td>
<td>CBS</td>
<td>R SPROG</td>
<td>INTRIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL VARIABLES (2 WORDS):</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CO000G</td>
<td>G167F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLANK COMMON (0 WORDS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENTRY POINTS:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CO00CC</td>
<td>G167F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTRINSIC SUBPROGRAMS USED:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABS</td>
<td></td>
<td></td>
<td></td>
<td>CBS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTERNAL SUBPROGRAMS REQUIRED:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9CBS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9SETUP1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HIGHEST ERROR SEVERITY:</td>
<td>C (NO ERRORS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DEC WORDS</td>
<td>20</td>
<td>CO014</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEX WORDS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENERATED CODE:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CONSTANTS:</td>
<td>5C</td>
<td>CO005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOCAL VARIABLES:</td>
<td>24</td>
<td>CO002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TEMPS:</td>
<td>3</td>
<td>CO003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOTAL PROGRAM:</td>
<td>30</td>
<td>CO01E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE GRICG(ZZ,HGT,XFAC,YFAC,TOP,BOT,DLEFT,RIGHT)

C SUBROUTINE GRICG, FOR GRAF3

C ANNOTATES X AND Y SCALES

C SSW=1 LP TO SUPPRESS ANNOTATION OF X AND Y SCALES

C

C USES SUBROUTINES FOR CALCOMP AND ISM

C ASSUME ISM INITIALIZED IN MAIN PROGRAM

C

C

C IBLT=2

C IBLT = 1C8

C WRITE(IIBLT=1C8)

C FORMAT(BSET PEN TO ORIGIN OF PLOT IN BOTH X AND Y)

C

C FALSE 10

C

C CALL WHERE(XORG,YORG)

C CALL WHERE (XORG, YORG, RFAC)

C CALL PLOT(XORG,YORG,-3)

C CALL SYMB(0*0,0*0,0*14,3*0,0*0,0*1)

C CALL SYMBL(0*0,0*0,0*14,3*0,0*0,0*1)

C IF(ISM(8)=300,200,300)

C PLOTTING X AND Y SCALES EVERY INCH

C

C ANGC=0*0

C JDEC=1

C ANGD=0*0

C KDEC=-1

C TDFY=TDFP/YFAC

C BBTY=BBTP/YFAC

C DLEFX=DLEFT/XFAC

C RIGTX=RIGT/XFAC

C AK=0*2*ZZ

C A3=0*3*ZZ

C X=DLEFX

C Y=BBTY

C ANT=DLEFT

C CALL PLOT(XX,YY,3)

C PLOTTING TIC

C YY+YY+A3

C CALL PLOT(XX,YY,2)

C XT=XX+AK

C YT=YY+AK

C CALL NUMB(XT,YT,HGT,ANT,ANGC,JDEC)

C CALL NUMBER (XT, YT, HGT, ANT, ANGC, JDEC)

C CALL PLOT(XX,YY,3)

C IF(XX=RIGTX)100,200,20C

C 10C XX*XX+(1*0*ZZ)

C CALL PLOT(xx,yy,2)

C ANT=ANT+(XFAC*ZZ)

C GO TO 50

C PLOTTING BORDER OF GRID LIMITS

C XX=RIGTX

C YY=BBTY

C CALL PLOT(XX,YY,3)

C YY=TDFY

C CALL PLOT(XX,YY,2)

C XX=DLEFX

C CALL PLOT(XX,YY,2)

C ANT=TDFP

C AX=(0*5*ZZ)
AY=(C-03*ZZ)

C PLOTTING TIC
220 XT=XX+AX
CALL FL0T(XT,YY,2)
XT=XX+AX
Y=YY+AY
CALL NLMB(XT,YT,HG,T,ANT,ANGD,KDEC)
CALL NUMBER (XT, YT, HGT, ANT, ANGD, KDEC)
CALL FLOT(XX,YY,3)
IF(YY=YY)300,300,250
YY=YY-(1+C*ZZ)
CALL FLOT(XX,YY,2)
ANT=ANT+(YFAC*ZZ)
GO TO 220
RETURN
END
### Local Variables (22 Words):

<table>
<thead>
<tr>
<th>Label</th>
<th>Hex</th>
<th>Label</th>
<th>Hex</th>
<th>Label</th>
<th>Hex</th>
<th>Label</th>
<th>Hex</th>
<th>Label</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>C0024</td>
<td>50</td>
<td>C0C49</td>
<td>100</td>
<td>C0067</td>
<td>200</td>
<td>C0075</td>
<td>220</td>
<td>00096</td>
</tr>
<tr>
<td>300</td>
<td>C00C4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>250</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (22 WORDS):

- C0C00 GRIDG
- C0006 JDEC
- C0GCC RIGTX
- C0C12 YT
- C0001 IIBUT
- C0002 XORG
- C0003 YORG
- C0004 RFCT
- C0005 ANGC
- C0007 ANGD
- C0008 KDEC
- C0009 YPFX
- C0010 YY
- C0000 A3
- C000F XX
- C0014 AX
- C0015 AY

### Blank Common (C Words):

ENTRY POINTS:

- C0C00 GRIDG

### External Subprograms Required:

- ISH
- NUMBER
- PLBT
- SYMBOL
- WHERE
- SSETUPN

### Highest Error Severity: C (No Errors)

<table>
<thead>
<tr>
<th>Dec</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>137</td>
<td>C0C0C</td>
</tr>
</tbody>
</table>

### Generated Code: 137 C0C0C

### Constants:

- 12  C0C0C

### Local Variables:

- 22  C0C1C

### Total Program: 240 C0C0C
SUBROUTINE GRID(ZZ,ZHT,NUMPL,DEGRA,FDEG2,RDEG2,RTOP,ITOP,RTOP,BOTMP)

C PLOTS AND ANNOTATES MERCATOR CHART GRID

C 20 OCTOBER 1972

C

C SSX(1) LP TO DRAW PERIMETER OF GRID ONLY
C SSX(5) LP TO MAKE DEGREE ANNOTATIONS INSIDE GRID (CHARACTER SIZE 0.07 INCH)
C SSX(8) LP TO MAKE DEGREE ANNOTATIONS OUTSIDE GRID (CHARACTER SIZE 0.21 INCH)
C SSX(1) LP TO SUPPRESS PLOTTING OF GRID

C USES SUBROUTINES ISW, ISX, AND CALCOMP ROUTINES
ASSUMES ISX INITIALIZED IN MAIN PROGRAM

C INITIALIZING DISTANCE AND CHARACTER HEIGHT CONSTANTS

SA=0.02*ZZ
SB=0.05*ZZ
SC=0.15*ZZ
SD=0.22*ZZ
SE=0.18*ZZ
SF=0.25*ZZ
SG=0.26*ZZ
SFAC=0 IF(ISW(5)=0) G0 TO 18
SFAC=1 IF(ISW(5)=1) G0 TO 18
SFAC=2 IF(ISW(5)=2) G0 TO 18
TA=(0.05+(ZFA F=0.05))*ZZ
TB=(0.07+(ZFACT=0.07))*ZZ
TC=(ZFACT=0.24)*ZZ
TD=0.26*ZZ
HGT=ZFACT=0.24*ZHT
C SET ORIGIN FOR CHART
CALL WHERE (XX,YY,FACT)
CALL PLOT (XX,YY,3)
C ANNOTATING PLOT NUMBER IN LOWER LEFT HAND CORNER
IF(ISW(S)=2) G0 TO 26
SPP=ZFACT=0.24*ZHT
CALL NUMBER (TA,TB,HGT,FLEFT,FAC=0)
CALL PLOT (XX,YY,3)
C START PLOTTING GRID
SLAT=SLAM
SLON=SLONG
CALL WRH(DEGRA,FDEG2,RDEG2,DEG2,SLON,SLAT,SLAM,SLONG,BOTMP,XX,YY)
CALL PLOT (XX,YY,2)
IF(ISW(5)=1) G0 TO 28
XT=XX+TA
YT=YY+TB
C START PLOTTING GRID
C
```fortran
60 IF(ISW(5)) 71,71,72
61 72 XT=XX+TA
62 YT=TD
63 GO TO 73
64 71 YT=SC
65 73 CALL NUMBER(XT,YT,HGT,FBBT,0,0,1)
66 C DRAW LEFT AND TOP SIDES OF FIDUCIAL HALF-INC SQUARE
67 XFID=XX+1-C
68 YFID=YY+0+C
69 CALL PLOT(XFID,YFID,3)
70 XFID=YY+C+5
71 CALL PLOT(XFID,YFID,2)
72 XFID=YY+C+5
73 CALL PLOT(XFID,YFID,2)
74 CALL PLOT(XX,YY+3)
75 SLAT=RT8P
76 CALL WR(DEGR,DFEG2,RDEG2,RLEFT,SINCH,SM,SLAT,SLONG,NOTMP,XX,YY)
77 CALL PLOT(XX,YY,2)
78 IF(ISW(5)) 36,36,34
79 34 XT=TC
80 YT=YY+TD
81 GO TO 77
82 YT=YY+SC
83 77 CALL NUMBER(XT,YT,HGT,FBBT,0,0,1)
84 SLONG=RLEFT
85 CALL WR(DEGR,DFEG2,RDEG2,RLEFT,SINCH,SM,SLAT,SLONG,NOTMP,XX,YY)
86 CALL PLOT(XX,YY,2)
87 IF(ISW(5)) 36,36,34
88 34 XT=TC
89 YT=YY+TD
90 GO TO 38
91 XT=XX+SC
92 38 CALL NUMBER(XT,YT,HGT,FBBT,0,0,1)
93 CALL PLOT(XX,YY,3)
94 CALL PLOT(0+0,0+0,2)
95 IF(ISW(5)) 42,42,40
96 40 XT=TC
97 YT=TC
98 GO TO 44
99 42 XT=SC
100 YT=SC
101 44 CALL NUMBER(XT,YT,HGT,FBBT,0,0,1)
102 C FINISHED DRAWING AND ANNOTATING PERIMETER OF GRID
103 C CHECKING IF NDEG GRID LINES WANTED
104 IF(ISW(1)) 80,50,80
105 50 NND=((IRIGT=1LEFT)/NDEG)*NDEG
106 NBOT=1
107 GO 60 N=NDEG,NND=NDEG
108 J=IRIGT-(1LEFT+N)
109 IF(J) 51,61,51
110 51 AAA=N
111 REM=DEGRA*AAA
112 SLONG=RLEFT*REM
113 FLONG=SLONG=57,29578
114 IF(NBOT=1) 54,52,54
115 52 SLAT=R8BT
116 CALL WR(DEGR,DFEG2,RDEG2,RLEFT,SINCH,SM,SLAT,SLONG,NOTMP,XX,YY)
117 CALL PLOT(XX,YY,3)
118 IF(ISW(5)) 55,55,53
119 53 XT=XX+TA
```
120. \( \text{CALL NUMBER (XT, YT, HGT, FLAT, O, O, -1)} \)
121. \( \text{CALL PLOT(XX, YY, 3)} \)
122. \( \text{SLAT=RTOP} \)
123. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
124. \( \text{CALL PLOT(XX, YY, 2)} \)
125. \( \text{NBT=2} \)
126. \( \text{GO TO 60} \)
127. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
128. \( \text{CALL PLOT(XX, YY, 2)} \)
129. \( \text{IF (ISH(5)) 56, 58} \)
130. \( \text{57 XT=XX+TA} \)
131. \( \text{YT=YY+TB} \)
132. \( \text{GO TO 59} \)
133. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
134. \( \text{CALL PLOT(XX, YY, 3)} \)
135. \( \text{NBT=2} \)
136. \( \text{C FINISHED DRAWING AND ANNOTATING THIS LONGITUDE LINE} \)
137. \( \text{60 CONTINUE} \)
138. \( \text{61 ND=(NBDP-I887)/NDEG*NDEG} \)
139. \( \text{NLEFT=2} \)
140. \( \text{GO TO 70} \)
141. \( \text{GO TO 59} \)
142. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
143. \( \text{CALL PLOT(XX, YY, 3)} \)
144. \( \text{NBT=2} \)
145. \( \text{C COMPLETED ALL LONGITUDE GRID LINES} \)
146. \( \text{COMPLETED ALL LONGITUDE GRID LINES} \)
147. \( \text{NLEFT=2} \)
148. \( \text{NBT=2} \)
149. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
150. \( \text{CALL PLOT(XX, YY, 3)} \)
151. \( \text{NBT=2} \)
152. \( \text{GO TO 70} \)
153. \( \text{GO TO 60} \)
154. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
155. \( \text{CALL PLOT(XX, YY, 3)} \)
156. \( \text{NBT=2} \)
157. \( \text{GO TO 70} \)
158. \( \text{GO TO 60} \)
159. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
160. \( \text{CALL PLOT(XX, YY, 3)} \)
161. \( \text{NBT=2} \)
162. \( \text{GO TO 70} \)
163. \( \text{GO TO 60} \)
164. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
165. \( \text{CALL PLOT(XX, YY, 3)} \)
166. \( \text{NBT=2} \)
167. \( \text{GO TO 70} \)
168. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
169. \( \text{CALL PLOT(XX, YY, 3)} \)
170. \( \text{SLONG=RRIGHT} \)
171. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
172. \( \text{CALL PLOT(XX, YY, 2)} \)
173. \( \text{NLEFT=2} \)
174. \( \text{GO TO 70} \)
175. \( \text{SLONG=RRIGHT} \)
176. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
177. \( \text{CALL PLOT(XX, YY, 3)} \)
178. \( \text{SLONG=LEFT} \)
179. \( \text{CALL WRH(CEGRA, FDEG2, RDEG2, RLEFT, SINCH, SMP, SLAT, SLONG, BMTMP, XX, YY)} \)
CALL FLAT(XX, YY, 2)
IF(ISW(5))68, 68, 67
XT=XX-TC
YT=YY-TD
G8 TR 69
XT=XX+SC
YT=YY+SB
CALL NUMBER (XT, YT, HGT, FLAT, 0, 0, -1)
CALL PLOT(XX, YY, 3)
C FINISHED ANNOTATING AND DRAWING THIS LATITUDE LINE
C COMPLETED ALL GRID LINES AND ANNOTATIONS
RETURN
END
HIGHEST ERROR SEVERITY: C (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>GENERATED CODE:</td>
<td>755</td>
<td>C02F3</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>26</td>
<td>C0C1A</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>31</td>
<td>C0C1F</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>27</td>
<td>C0C1B</td>
</tr>
<tr>
<td>TOTAL PROGRAM:</td>
<td>839</td>
<td>C0347</td>
</tr>
</tbody>
</table>
SUBROUTINE INCEP(SL, SH, CX, CY, CXP, CYP, BX, BY, A1, A2, B1, B2)

SUBROUTINE INCEP, DETERMINES INTERCEPT POINTS OF LINE BETWEEN ANY 2 POINTS AND GIVEN BOUNDARIES

USES SUBROUTINE EXTD

\[ h = 0 \]

\[ s_x = (CY - CYP)/(CX - CXP) \]

\[ B_x = ((CYP - CX) - (CY - CXP))/(CX - CXP) \]

IF \((|ABS(CXP)|) = SL\), \(20522052300\)

IF \((|ABS(CYP)|) = SW\), \(20622062300\)

\[ A_1 = CXP \]

\[ B_1 = CYP \]

\[ h = h + 1 \]

IF \((h = 2)\), \(8002700800\)

\[ B_x = SL \]

\[ B_y = SM = BX*SB \]

IF \((|ABS(BY)|) = SW\), \(31C231C07330\)

CALL EXTD(CX, CXP, CY, CYP, BX, BY, IND)

IF \((INC)\), \(3202320,330\)

\[ A_1 = BX \]

\[ B_1 = BY \]

\[ h = h + 1 \]

\[ B_x = SH \]

\[ B_y = SB \]

IF \((|ABS(BX)|) = SL\), \(34C234C2365\)

CALL EXTD(CX, CXP, CY, CYP, BX, BY, IND)

IF \((INC)\), \(3502350,365\)

\[ h = h + 1 \]

IF \((h = 2)\), \(3652365,387,800\)

\[ A_2 = BX \]

\[ B_2 = BY \]

GO TO 700

\[ B_x = EX \]

\[ B_y = EY \]

CALL EXTD(CX, CXP, CY, CYP, BX, BY, IND)

IF \((INC)\), \(3752375,390\)

\[ h = h + 1 \]

IF \((h = 2)\), \(3852385,390\)

\[ A_3 = BX \]

\[ B_3 = BY \]

GO TO 700

\[ B_x = (BY - SB)/SM \]

\[ B_y = SW \]

IF \((|ABS(BX)|) = SL\), \(40524052400\)

CALL EXTD(CX, CXP, CY, CYP, BX, BY, IND)
6C 40C IF(INC)4CC,410,400
61 42C IF(N=1)90C,420,800
62 A2=BX
63 B2=BY
64 G6 TO 700
65 41C N=N+1
66 411 A1=BX
67 B1=BY
68 A2=BX
69 B2=BY
70 G6 TO 700
71 412 A2=BX
72 B2=BY
73 IF(N=2)8OC,70C,800
74 5OC BX=SL
75 BY=SM*BX+B
76 505 IF((ABS(BY))=SW)515,515,525
77 515 CALL EXT(D CX,CXP,CY,CYP,BX,BY,IND)
78 IF(INC)525,520,525
79 52C N=N+1
80 525 A2=BX
81 B2=BY
82 IF(N=2)80C,70C,800
83 525 BY=SW
84 BX=(BY-B)/SM
85 53C IF((ABS(BX))=SL)545,545,555
86 545 CALL EXT(D CX,CXP,CY,CYP,BX,BY,IND)
87 IF(INC)555,550,555
88 55C N=N+1
89 55C A2=BX
90 B2=BY
91 IF(N=2)80C,70C,800
92 555 BX=SL
93 BY=SM*BX+B
94 56C IF((ABS(BY))=SW)58C,58C,590
95 58C CALL EXT(D CX,CXP,CY,CYP,BX,BY,IND)
96 IF(INC)590,585,59C
97 585 N=N+1
98 585 A2=BX
99 B2=BY
100 IF(N=2)8OC,70C,800
101 59C BY=SH
102 BX=(BY-B)/SM
103 59S IF((ABS(BX))=SL)60C,60C,800
104 60C N=N+1
105 60C A2=BX
106 B2=BY
107 IF(N=2)8OC,70C,800
108 7CC RETURN
109 HCC WRITF (108,802)
110 802 FROAT ("INCEP BAD BRANCH")
111 9CC A1 = 9999*C
112 A2 = 9999*C
113 B1 = 9999*C
114 B2 = 9999*C
115 RETURN
116 END
Here is the text transcribed from the document:

### Local Variables (5 Words):
- 00000 INCEP 00001 N 00002 SM 00003 5 00004 INO

### Blank Common (0 Words)

### Entry Points:
- COOC00 INCEP

### Intrinsic Subprograms Used:
- ABS

### External Subprograms Required:
- EXTD F1103 SPRINT SETUP

### Highest Error Severity: 0 (No Errors)

<table>
<thead>
<tr>
<th>LOCAL VARIABLES</th>
<th>HEX WORDS</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>TYPE</td>
<td>CLASS</td>
</tr>
<tr>
<td>ABS</td>
<td>R</td>
<td>SPRG</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>SCALR</td>
</tr>
<tr>
<td>C2</td>
<td>R</td>
<td>SCALR</td>
</tr>
<tr>
<td>EXTC</td>
<td>SPRG</td>
<td>EXTERN</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>SCALR</td>
</tr>
<tr>
<td>SH</td>
<td>R</td>
<td>SCALR</td>
</tr>
</tbody>
</table>

### Local Variables (5 Words):
- 00000 INCEP 00001 N 00002 SM 00003 5 00004 INO

### Blank Common (0 Words)

### Entry Points:
- COOC00 INCEP

### Intrinsic Subprograms Used:
- ABS

### External Subprograms Required:
- EXTD F1103 SPRINT SETUP

### Highest Error Severity: 0 (No Errors)
FUNCTION ISW(I)
VERSION OF 27 JULY 1973, OUTPUT SSW CHANGE IF MADE

C VERSION OF 25 JULY 1973, TO ADD ENTRY ICWH
C THIS FUNCTION READS A CARD WHEN I<0, SETTING VALUES OF ISW(I) IN COLUMNS 1 TO
C 79, THE VALUE OF ISW(0) IS SET EQUAL TO ISW(80). WITH THIS EXCEPTION
C THE VALUE OF ISW(I) CORRESPONDS TO THE COLUMN NUMBER.
C FOR I<0, RETURNS THE VALUE OF JSW(I)
D DIMENSION JS(0:80)
I IN=105
I OUT=108
IF (I=LYC) Go T8 50
JSW=JSW(I)
RETURN

C ENTRY POINT TO ALLOW CHANGING OF SSW VALUE
ENTRY ICWH(I,IVAL)
JSW(I)=IVAL
WRITE(I OUT,42),IVAL
F85 FORMAT(I,SENSE SWITCH I41 SET TO 112)
RETURN
READ(I IN,60)(JSW(K),K=1:80)
F86 FORMAT(801I)
JSW(K)=JSW(80)
WRITE(I OUT,75)(JSW(K),K=1:79)
F76 FORMAT('SENSE SWITCH OPTIONS (ISW)'S,8(1Cl11,1%,4X,'(0-79)'))
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>00055</td>
<td>DUMMY</td>
<td>ICW</td>
<td>I</td>
<td>SPRB</td>
<td>00067</td>
<td>P</td>
<td>IIN</td>
<td>I</td>
<td>SCALR</td>
<td>00052</td>
<td>V</td>
</tr>
<tr>
<td>IOUT</td>
<td>I</td>
<td>SCALR</td>
<td>00053</td>
<td>V</td>
<td>1</td>
<td>ISM</td>
<td>SPRB</td>
<td>00060</td>
<td>P</td>
<td>ISW</td>
<td>I</td>
<td>SCALR</td>
<td>00000</td>
<td>V</td>
</tr>
<tr>
<td>IVAL</td>
<td>I</td>
<td>SCALR</td>
<td>00055</td>
<td>V DUMMY</td>
<td>JSW</td>
<td>I</td>
<td>ARRAY</td>
<td>00001</td>
<td>V</td>
<td>81</td>
<td>K</td>
<td>SCALR</td>
<td>00054</td>
<td>V</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>0001C</td>
<td>50</td>
<td>00029</td>
<td>60</td>
<td>00036</td>
<td>70</td>
<td>00057</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (85 WORDS):
- 00000 ISH
- 00001 JSW
- 00055 IIN
- 00053 IOUT
- 00054 K

BLANK COMMON (0 WORDS)
ENTRY POINTS:
- 00000 ISH
- 0000F ICW

EXTERNAL SUBPROGRAMS REQUIRED:
- F1101
- F1102
- F1103
- F1104
- F1105
- F1106
- F1107
- 98CDREAD

HIGHEST ERROR SEVERITY: 1 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED Code: 104</td>
<td>00068</td>
</tr>
<tr>
<td>Constants: 0</td>
<td>00000</td>
</tr>
<tr>
<td>Local Variables: 85</td>
<td>00055</td>
</tr>
<tr>
<td>Temp: 3</td>
<td>00003</td>
</tr>
</tbody>
</table>

TOTAL PROGRAM: 192 00000
SUBROUTINE M2DY(IY,MN,DAY,ID)
C*** M2DY CHANGES MONDAY TO THE NUMBER OF DAYS FOR THAT YEAR
C*** IY=YEAR (CONVERTS FOR LEAP YEAR
C*** MN=MONTH
C*** DAY=DAY
C*** IO=OUTFLT WHICH IS THE PROGRESSIVE DAY NUMBER
C
DIMENSION MYDAY(12)
DATA MYDAY/1,32,60,91,122,153,182,213,244,274,305,335/
C*** DETERMINE IF LEAP YEAR
10 A=IY
11 B=IY/4
12 A=A/4+C
13 IF (A-B) 12,10,12
14 LEAF=1
15 G8 TO 13
16 LEAF=C
17 CONTINUE
18 IF(MN=2) 21,22
19 IF(DAY=28) 20,20
20 LEAF=0
21 I0=DAY+MYDAY(MN)-1+LEAF
22 RETURN
23 END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALR</td>
<td>00000</td>
<td>4</td>
<td>B</td>
<td>R</td>
<td>SCALR</td>
<td>00000</td>
<td>4</td>
<td>C</td>
<td>R</td>
<td>SCALR</td>
<td>00000</td>
<td>4</td>
</tr>
<tr>
<td>IDAY</td>
<td>I</td>
<td>SCALR</td>
<td>00000</td>
<td>4</td>
<td>MYDAY</td>
<td>I</td>
<td>ARRAY</td>
<td>00000</td>
<td>4</td>
<td>LEAP</td>
<td>I</td>
<td>SCALR</td>
<td>00000</td>
<td>4</td>
</tr>
<tr>
<td>MB</td>
<td>I</td>
<td>SCALR</td>
<td>00000</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>M2DY</td>
<td>I</td>
<td>SCALR</td>
<td>00000</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 C001</td>
<td>12 00017</td>
<td>13 C001</td>
<td>20 C0020</td>
<td>21 C001D</td>
<td>22 C0022</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (16 WORDS):
C0000 M2DY C0001 MYDAY C0002 A C0003 B C0004 LEAP

BLANK COMMON (0 WORDS)

ENTRY POINTS:
C0000 M2DY

EXTERNAL SUBPROGRAMS REQUIRED:
9108R SSETPNP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GENERATED CODE: 41 C0029
CONSTANTS: 2 C0002
LOCAL VARIABLES: 16 C001C
TEMPs: 5 C0005

TOTAL PROGRAM: 64 C004C
SUBROUTINE NAVIN(LAT,RLATM,KNS,LONG,RLAM,KEW,RLAT,RLONG)

SUBROUTINE NAVIN, CONVERTS ANOTATED DEGREES AND MINUTES
TO SIGNED RADIANS LATITUDE AND
LONGITUDE

USES SUBROUTINE DMTOR

10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.

KNS = 1HS
KEW = 1HW
CS
CS
6C
6C
9C
9C
11C
11C
13C
13C
15C
15C
17C
17C
19C
19C
21C
21C
23C
23C

NNS = 1238
NEW = 1278
RLAT = DMTOR(LAT,RLATM)
RLONG = DMTOR(LONG,RLAM)
IF (KNS=NNS)75,70,75
SOUTH LATITUDE
7C
7E
WEST LONGITUDE
8C
8E
CONTINUE
RETURN
END
### LOCAL VARIABLES (3 WORDS):

- CO000: NAVIN
- CO001: NNS
- CO002: NEW

### BLANK COMMON (0 WORDS)

### ENTRY POINTS:

- CO000: NAVIN

### EXTERNAL SUBPROGRAMS REQUIRED:

- DMT8R
  - 95ETLPN

### HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

### GENERATED CODE:

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>38</td>
<td>CO026</td>
</tr>
<tr>
<td>00</td>
<td>CO002</td>
</tr>
<tr>
<td>00</td>
<td>CO003</td>
</tr>
<tr>
<td>00</td>
<td>CO009</td>
</tr>
<tr>
<td>52</td>
<td>CO034</td>
</tr>
</tbody>
</table>

### LOCAL VARIABLES:

<table>
<thead>
<tr>
<th>LBC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO000</td>
<td>NAVIN</td>
</tr>
<tr>
<td>CO001</td>
<td>NNS</td>
</tr>
<tr>
<td>CO002</td>
<td>NEW</td>
</tr>
</tbody>
</table>

### CONSTANTS:

<table>
<thead>
<tr>
<th>LBC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO000</td>
<td>00</td>
</tr>
<tr>
<td>CO001</td>
<td>00</td>
</tr>
<tr>
<td>CO002</td>
<td>00</td>
</tr>
<tr>
<td>CO003</td>
<td>00</td>
</tr>
<tr>
<td>CO009</td>
<td>00</td>
</tr>
</tbody>
</table>

### TEMPS:

<table>
<thead>
<tr>
<th>LBC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO000</td>
<td>00</td>
</tr>
<tr>
<td>CO001</td>
<td>00</td>
</tr>
<tr>
<td>CO002</td>
<td>00</td>
</tr>
<tr>
<td>CO003</td>
<td>00</td>
</tr>
<tr>
<td>CO009</td>
<td>00</td>
</tr>
</tbody>
</table>
SUBROUTINE NAVRT(RLAT,RLONG,LAT,RLATM,KNS,
                1LONG,RLONM,KEN,KDEC)
C
C SUBROUTINE NAVRT CONVERTS RACIANS TO DEG,MIN AND LETTER FOR HEMISPHERE
C
USES SUBROUTINES RTDMG, RTDM1, RTDM2

KDEC=KDEC+1
CS     JN=116B
CS     JS=123B
CS     JF=105B
CS     JH=127B
14*    JN = 1-H
15*    JS = 1-H
16*    JF = 1-H
17*    JH = 1-H
18*    ALAT=ABS(RLAT)
19*    41C GO TO(420,421,422),KDEC
20*    42C CALL RTDMG(ALAT,LAT,RLATM)
21*    GO TO 429
22*    421 CALL RTDM1(ALAT,LAT,RLATM)
23*    GO TO 429
24*    422 CALL RTDM2(ALAT,LAT,RLATM)
25*    429 IF(RLAT).GT.430,432,432
26*    43C KNS*JE
27*    GO TO 435
28*    432 KNS=JN
29*    43E ALONG=ABS(RLONG)
30*    43E GO TO(435,437,43E),KDEC
31*    43E CALL RTDMG(ALONG,RLONG,RL0M)
32*    GO TO 439
33*    437 CALL RTDM1(ALONG,RLONG,RL0M)
34*    GO TO 439
35*    438 CALL RTDM2(ALONG,RLONG,RL0M)
36*    435 IF(RLONG).GT.440,442,442
37*    44C KEK=JH
38*    GO TO 445
39*    442 KEK=JE
40*    445 RETURN
41*    END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>R</td>
<td>SFRPG</td>
<td></td>
<td></td>
<td></td>
<td>ALAT</td>
<td>R</td>
<td>SCALR</td>
<td>00009</td>
<td></td>
<td>1</td>
<td>ALBNG</td>
<td>R</td>
<td>SCALR</td>
<td>00006</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WE</td>
<td>1</td>
<td>SCALR</td>
<td>00003</td>
<td>V</td>
<td>1</td>
<td>JN</td>
<td>1</td>
<td>SCALR</td>
<td>00001</td>
<td>V</td>
<td>1</td>
<td>JS</td>
<td>1</td>
<td>SCALR</td>
<td>00002</td>
<td>V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAVT</td>
<td>1</td>
<td>SCALR</td>
<td>00004</td>
<td>V</td>
<td>DUMMY</td>
<td>PDEC</td>
<td>1</td>
<td>SCALR</td>
<td>0000F</td>
<td>V</td>
<td>DUMMY</td>
<td>KEK</td>
<td>1</td>
<td>SCALR</td>
<td>0000E</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTD1</td>
<td>1</td>
<td>SCALR</td>
<td>00009</td>
<td>V</td>
<td>DUMMY</td>
<td>LAT</td>
<td>1</td>
<td>SCALR</td>
<td>00009</td>
<td>V</td>
<td>DUMMY</td>
<td>LONG</td>
<td>1</td>
<td>SCALR</td>
<td>0000C</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTD2</td>
<td>1</td>
<td>SCALR</td>
<td>00007</td>
<td>V</td>
<td>DUMMY</td>
<td>NAVT</td>
<td>1</td>
<td>SCALR</td>
<td>00007</td>
<td>V</td>
<td>DUMMY</td>
<td>RLAT</td>
<td>1</td>
<td>SCALR</td>
<td>00007</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTD3</td>
<td>1</td>
<td>SCALR</td>
<td>0000D</td>
<td>V</td>
<td>DUMMY</td>
<td>RL87</td>
<td>1</td>
<td>SCALR</td>
<td>0000D</td>
<td>V</td>
<td>DUMMY</td>
<td>RLOM1</td>
<td>1</td>
<td>SCALR</td>
<td>00008</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTD0</td>
<td>1</td>
<td>SCALR</td>
<td>0000F</td>
<td>V</td>
<td>DUMMY</td>
<td>RLOM2</td>
<td>1</td>
<td>SCALR</td>
<td>0000F</td>
<td>V</td>
<td>DUMMY</td>
<td>RTDH1</td>
<td>1</td>
<td>SCALR</td>
<td>00009</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Local Variables (7 Words)**:
- 00000 NAVT
- 00001 WA
- 00002 JS
- 00003 JE
- 00004 JW
- 00005 ALAT

**Blank Common (0 Words)**

**Entry Points**:
- CCC6C NAVT

**Intrinsic Subroutines Used**:
- ABS

**External Subroutines Required**:
- RTM1
- RTD1
- RTD2
- RTDH1

**Highest Error Severity**: C (No Errors)

**Generated Code**: 90
**Constants**: 4
**Local Variables**: 7
**Temp**: 10
**Total Program**: 111

**HEX**

<table>
<thead>
<tr>
<th>Label</th>
<th>HEX LOC</th>
<th>Label</th>
<th>HEX LOC</th>
<th>Label</th>
<th>HEX LOC</th>
<th>Label</th>
<th>HEX LOC</th>
<th>Label</th>
<th>HEX LOC</th>
<th>Label</th>
<th>HEX LOC</th>
<th>Label</th>
<th>HEX LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>00017</td>
<td>20</td>
<td>0001F</td>
<td>21</td>
<td>00025</td>
<td>22</td>
<td>00026</td>
<td>23</td>
<td>00030</td>
<td>24</td>
<td>00032</td>
<td>25</td>
<td>00034</td>
</tr>
<tr>
<td>32</td>
<td>00035</td>
<td>35</td>
<td>00037</td>
<td>36</td>
<td>00041</td>
<td>37</td>
<td>00047</td>
<td>38</td>
<td>0004D</td>
<td>39</td>
<td>00052</td>
<td>40</td>
<td>00054</td>
</tr>
</tbody>
</table>

**HEX**

- GENERATED CODE: 90
- CONSTANTS: 4
- LOCAL VARIABLES: 7
- TEMPS: 10
- TOTAL PROGRAM: 111
SUBROUTINE 88G(K977, G8CR, G8BS, KK)
C SUBROUTINE 88G, CHANGE OBSERVED GRAVITY IN TWO
C WORDS TO GRAVITY LESS 97700C.0 IF
C KK=1; OR VISA VERSA FOR KK=2

6  C IF(KK)200*100*100
7  C TWO WORDS TO ONE
8  100 A = K977=977
9  G8BS=(A * 1000+C)+G8GR
10  RETURN
11  C ONE WORD TO TWO
12  200 A=G8BS*C+01
13  II=A
14  B=II
15  G8GR=(A-B)*1000*0
16  K977=II+577
17  RETURN
18  END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALR</td>
<td>00005</td>
<td>V DUMMY</td>
<td>B</td>
<td>R</td>
<td>SCALR</td>
<td>00003</td>
<td>V</td>
<td>G98G</td>
<td>I</td>
<td>SCALR</td>
<td>00004</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>K977</td>
<td>I</td>
<td>SCALR</td>
<td>00006</td>
<td>V DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
<th>LABEL</th>
<th>HEX</th>
<th>LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0009</td>
<td>20C</td>
<td>00012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (4 WORDS):**

0000C 0BG  00001 A  0000C 11  0000C B

**BLANK COMMON (0 WORDS)**

**ENTRY POINTS:**

0000C 0BG

**EXTERNAL SUBPROGRAMS REQUIRED:**

9170R  9RT8I  9SETUPN

**HIGHEST ERROR SEVERITY: 0 (NO ERRORS)**

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>00023</td>
</tr>
<tr>
<td>2</td>
<td>00002</td>
</tr>
<tr>
<td>4</td>
<td>00004</td>
</tr>
<tr>
<td>5</td>
<td>00005</td>
</tr>
<tr>
<td>46</td>
<td>0002E</td>
</tr>
</tbody>
</table>

**TOTAL PROGRAM:**

46  0002E
SUBROUTINE BLINE(ZZ,ZHT,
A ICATA, IE8D, IIN, IEHUT, ITAPE, NPL, IAT, RLAT, RLANG, KAGH, IAGAP, LCN
B, RST, RAGEO, DRAK, RDEO, IDEG2, DDEG2, RDEG2, RTOP, ITOP, RBOT, IBOT, LFRIGHT,
C ILEFT, IRIGHT, SLIGHT, SLIGHT, SINK, SPP, FBOT, FTOP, FLEFT, FRIGHT, NDEG,
D SLAT, SLONG, BBTMP, XX, YY, INIT, XBLC, Y8LD)

C VERSION 11 JULY 73 TO INCLUDE FIDUCIAL SQUARE
C SUBROUTINE BLINE(ZZ,ZHT), PLOTS GRID FOR MERIDIAN
C CHART HAVING NON-INTEGRAL BOUNDARIES

C SSW$(8) UF TO SUPPRESS PLOTTING OF GRID
C
C CALLS SUBROUTINES RTDCM, WHR, ISW, AND CALLCOMP ROUTINES

C INITIALIZING DISTANCE AND CHARACTER HEIGHT CONSTANTS

C HGT = 0.072 * H
SA = 0.072 * ZZ
SB = 0.072 * ZZ
SC = 0.072 * ZZ
SD = 0.072 * ZZ
SE = 0.072 * ZZ
SF = 0.072 * ZZ
SG = 0.072 * ZZ
CALL RTDCM(RTOP, LAT, RLAT)
CALL RTDCM(RLEFT, LLANG, RLBM)
FRHT = LAT
FBRM = RLATM
FLEFT = LLANG
CALL RTDCM(RBRM, LAT, RLATM)
CALL RTDCM(RHRT, LLANG, RLM)
FBOT = LAT
FBOT = RLATM
FRIGHT = LLANG
FRIGHT = RLAM
C SET ORIGIN FOR CHART
CALL WHERE (XORG, YORG, RFCT)
CALL WHERE (XORG, YORG)
CALL PLOT (XORG, YORG, 3)
IF (IISW$(8)) BQ 20*80
C ANNOTATING CONSECUTIVE PLOT NUMBER
PNUML = PLNUM
CALL NUMBER (SF, SA, HGT, PNUM, 9C0, 1)
PNUML = PLNUM + 1
C ANNOTATING GRID
CALL NUMBER (SB, SA, HGT, FLEFT, 0G0, 1)
CALL NUMBER (SB, SA, HGT, FLEFT, 0G0, 1)
XT = SE + SD
YT = SA
CALL NUMBER (XT, YT, HGT, FLTM, 0G0, 1)
CALL NUMBER (XT, YT, HGT, FLTM, 0G0, 1)
CALL PLOT (0C0, 0C0, 0G0, 3)
C START PLOTTING GRID
SLAT=88T
SLANG=FRIGT
CALL MH(DGRA,FAEG2,DEEG2,RLF1T,SINH,FMF,SLAT,SLONG,B86MP,XX,YY)
CALL PLAT (XX,YY,2)
C DRAG LEFT AND TOP SIDES OF FIDUCIAL SQUARE
XX=XX+1+C
YY=YY+0.5
CALL PLAT(XX,YY,3)
YY=YY+C-5
CALL PLAT(XX,YY,2)
XX=XX+C-5
CALL PLAT(XX,YY,2)
CALL PLAT(XX,YY,3)
XT=XX-(2*C*SG)-SB
YT=YY+SB
CS CALL NUMBER (XT,YT,HGT,FRIPT,0*C,-1)
CALL NUMBER (XT, YT, HGT, FRIGT, C*C, -1)
XT=XX+SG
YT=YY+SB
CALL NUMBER (XT, YT, HGT, FRIPT, 0*C, -1)
CS CALL NUMBER (XT, YT, HGT, FRIPT, 0*C, -1)
CALL PLAT (XX,YY,2)
SLAT=86F
SLANG=FRIGT
CALL MH(DGRA,FAEG2,DEEG2,RLF1T,SINH,FMF,SLAT,SLONG,B86MP,XX,YY)
CALL PLAT (XX,YY,2)
SLAT=86F
CALL PLAT (XX,YY,2)
CALL NUMBER (XT, YT, HGT, FRIPT, 0*C, -1)
CALL NUMBER (XT, YT, HGT, FRIPT, 0*C, -1)
XT=XT+10C
YT=YT+5
CALL NUMBER (XT, YT, HGT, FRIPT, 0*C, -1)
CALL NUMBER (XT, YT, HGT, FRIPT, 0*C, -1)
CALL NUMBER (XT, YT, HGT, F86M, 0*C, -1)
CALL NUMBER (XT, YT, HGT, F86M, 0*C, -1)
OUTPUT 'SUBROUTINE ALINE ON JULY 73'
C FINISHED PLAT AND ANST BASIC GRID
RETURN
END
<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE:</td>
<td>375</td>
<td>C0177</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>19</td>
<td>C0013</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>25</td>
<td>C0019</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>46</td>
<td>C002E</td>
</tr>
<tr>
<td>TOTAL PROGRAM:</td>
<td>465</td>
<td>C01D1</td>
</tr>
</tbody>
</table>
SUBROUTINE PINOT (ITAPE, JTAPE, KK, ISTA, KEY, LAT, LATM, KNS, N2, N3, N4, MET, IYR, IDESC, THICK)

DIMENSION IDESC(6), VEL(8), THICK(8)

USES SUBROUTINE ISWNAV

C
IF (KK) 420 ?40 410

C ITAPE = URN FOR SEISMIC DATA INPLT
C JTAPE = URN FOR DATA OUTPLT

C

C C READING U OF TORONTO WORLD SEISMIC REFRACTION COMPILATION
C

C CONTINUE

C

C 12 READ (ITAPE, 990, END=300, ERR=10)

C

C

C

C

C 15 READ (ITAPE, 991, END=900, ERR=11)

C

C

C

C

C
CONTINUE
40 IF(ISH(32))22,30,22
41 CHECK FOR CARDS BUT IF ORDER
42 IF(IREC1=1)24,30,24
43 OUTPUT 'IREC1 = NE* 1'
44 OUTTUT ISTAT
45 READ(ITAPE,25)
46 FORMAT(1X)
47 GO TO 15
48 3C VEL(1)=(FLAT(J1)) *0.1
49 VEL(2)=(FLAT(J2)) *0.1
50 VEL(3)=(FLAT(J3)) *0.1
51 VEL(4)=(FLAT(J4)) *0.1
52 VEL(5)=(FLAT(J5)) *0.1
53 VEL(6)=(FLAT(J6)) *0.1
54 VEL(7)=(FLAT(J7)) *0.1
55 VEL(8)=(FLAT(J8)) *0.1
56 THICK(1)=(FLAT(K1)) *0.1
57 THICK(2)=(FLAT(K2)) *0.1
58 THICK(3)=(FLAT(K3)) *0.1
59 THICK(4)=(FLAT(K4)) *0.1
60 THICK(5)=(FLAT(K5)) *0.1
61 THICK(6)=(FLAT(K6)) *0.1
62 THICK(7)=(FLAT(K7)) *0.1
63 THICK(8)=(FLAT(K8)) *0.1
64 LSTAT=1STA
65 RETURN
66 42C CONTINUE
67 LTKEY=LAT+90
68 RTHM=LAM
69 RLM=LM
70 CALL NAVIN(LAT,RLAT,KNS,RLONG,LOM,KEW,RLAT,RLONG)
71 C calculating sorting keys
72 PLAT(RLAT,RADEG)+90
73 PLNG=(RLNG,RADEG)+180
74 LTKEY=FLAT
75 LGKEY=FLAT
76 IF(ISH(26))420,430,435,430
77 JTAPE=990
78 IRECl=I40
79 43C WRITE(ITAPE,990)IREC1,ISTA,KEY,LAT,RLAT,KNS,RLNG,LM,
80 1 KEW,N1,K2,N3,K4,J5,N5,K6,J7,K7,J8,KB,
81 2 IMANT,NELEV,N1,N2,N3,N4,MT,RY,DESC,DINE,STH,CT,CRN,
82 3 GHTN,AVWTH,CRVW,GHTG,AVWTH
83 4,LTKEY,LGKEY,IAKEY
84 RETURN
85 44C WRITE(ITAPE,991)IREC1,ISTA,KEY,LAT,RLAT,KNS,RLNG,LM,
86 1 KEW,K1,K2,K3,K4,J5,K5,N6,J6,K6,J7,K7,J8,KB,
87 2 IMANT,NELEV,N1,N2,N3,N4,MT,RY,DESC,DINE,STH,CT,CRN,
88 3 GHTN,AVWTH,CRVW,GHTG,AVWTH
89 4,LTKEY,LGKEY,IAKEY
90 RETURN
91 C OUTPUT RECORD
92 42C CONTINUE
93 43C WRITE(ITAPE,990)IREC1,ISTA,KEY,LAT,RLAT,KNS,RLNG,LM,
94 1 KEW,K1,K2,K3,J4,K5,N6,J6,K6,J7,K7,8,KB,
95 2 IMANT,NELEV,N1,N2,N3,N4,MT,RY,DESC,DINE,STH,CT,CRN,
96 3 GHTN,AVWTH,CRVW,GHTG,AVWTH
97 4,LTKEY,LGKEY,IAKEY
98 RETURN
99 C OUTPUT RECORD
100 42C CONTINUE
101 C OUTPUT RECORD
102 43C WRITE(ITAPE,991)IREC1,ISTA,KEY,LAT,RLAT,KNS,RLNG,LM,
103 1 KEW,K1,K2,K3,K4,J5,K5,N6,J6,K6,J7,K7,8,KB,
104 2 IMANT,NELEV,N1,N2,N3,N4,MT,RY,DESC,DINE,STH,CT,CRN,
105 3 GHTN,AVWTH,CRVW,GHTG,AVWTH
106 4,LTKEY,LGKEY,IAKEY
107 RETURN
108 435 IF(ISH(33))430,440,460
109 44C WRITE(ITAPE,990)IREC1,ISTA,KEY,LAT,RLAT,KNS,RLNG,LM,
110 1 KEW,K1,K2,K3,K4,J5,K5,N6,J6,K6,J7,K7,8,KB,
111 2 IMANT,NELEV,N1,N2,N3,N4,MT,RY,DESC,DINE,STH,CT,CRN,
112 3 GHTN,AVWTH,CRVW,GHTG,AVWTH
113 4,LTKEY,LGKEY,IAKEY
114 RETURN
115 46C WRITE(ITAPE,991)IREC1,ISTA,KEY,LAT,RLAT,KNS,RLNG,LM,
116 1 KEW,K1,K2,K3,K4,J5,K5,N6,J6,K6,J7,K7,8,KB,
117 2 IMANT,NELEV,N1,N2,N3,N4,MT,RY,DESC,DINE,STH,CT,CRN,
118 3 GHTN,AVWTH,CRVW,GHTG,AVWTH
119 4,LTKEY,LGKEY,IAKEY
120*  RETURN
121*  900  CONTINUE
122*  WRITE(IIBLT,902)
123*  902  FORMAT('EOF FOUND, PROCESSING COMPLETED')
124*  KK=9
125*  RETURN
126*  END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVWTH</td>
<td>R SCALR</td>
<td>00039</td>
<td>V DUMMY</td>
<td>AVWTH</td>
<td>R SCALR</td>
<td>0003C</td>
<td>V DUMMY</td>
<td>AVWTH</td>
<td>R SCALR</td>
<td>0003D</td>
<td>V DUMMY</td>
<td>AVWTH</td>
<td>R SCALR</td>
<td>0003E</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>CRWTH</td>
<td>R SCALR</td>
<td>0003A</td>
<td>V DUMMY</td>
<td>CRWTH</td>
<td>R SCALR</td>
<td>0003B</td>
<td>V DUMMY</td>
<td>CRWTH</td>
<td>R SCALR</td>
<td>0003C</td>
<td>V DUMMY</td>
<td>CRWTH</td>
<td>R SCALR</td>
<td>0003D</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>IAKEY</td>
<td>I SCALR</td>
<td>00004</td>
<td>1</td>
<td>IAKEY</td>
<td>I SCALR</td>
<td>00005</td>
<td>1</td>
<td>IAKEY</td>
<td>I SCALR</td>
<td>00006</td>
<td>1</td>
<td>IAKEY</td>
<td>I SCALR</td>
<td>00007</td>
<td>1</td>
</tr>
<tr>
<td>I10BT</td>
<td>I SCALR</td>
<td>00002</td>
<td>1</td>
<td>I10BT</td>
<td>I SCALR</td>
<td>00003</td>
<td>1</td>
<td>I10BT</td>
<td>I SCALR</td>
<td>00004</td>
<td>1</td>
<td>I10BT</td>
<td>I SCALR</td>
<td>00005</td>
<td>1</td>
</tr>
<tr>
<td>ISTA</td>
<td>I SCALR</td>
<td>00022</td>
<td>1</td>
<td>ISTA</td>
<td>I SCALR</td>
<td>00023</td>
<td>1</td>
<td>ISTA</td>
<td>I SCALR</td>
<td>00024</td>
<td>1</td>
<td>ISTA</td>
<td>I SCALR</td>
<td>00025</td>
<td>1</td>
</tr>
<tr>
<td>ITAPE</td>
<td>I SCALR</td>
<td>0001F</td>
<td>V DUMMY</td>
<td>ITAPE</td>
<td>I SCALR</td>
<td>0002D</td>
<td>V DUMMY</td>
<td>ITAPE</td>
<td>I SCALR</td>
<td>00030</td>
<td>V DUMMY</td>
<td>ITAPE</td>
<td>I SCALR</td>
<td>00031</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>J1</td>
<td>I SCALR</td>
<td>00007</td>
<td>1</td>
<td>J1</td>
<td>I SCALR</td>
<td>00008</td>
<td>1</td>
<td>J1</td>
<td>I SCALR</td>
<td>00009</td>
<td>1</td>
<td>J1</td>
<td>I SCALR</td>
<td>00010</td>
<td>1</td>
</tr>
<tr>
<td>J9</td>
<td>I SCALR</td>
<td>00000</td>
<td>1</td>
<td>J9</td>
<td>I SCALR</td>
<td>00001</td>
<td>1</td>
<td>J9</td>
<td>I SCALR</td>
<td>00002</td>
<td>1</td>
<td>J9</td>
<td>I SCALR</td>
<td>00003</td>
<td>1</td>
</tr>
<tr>
<td>J7</td>
<td>I SCALR</td>
<td>00013</td>
<td>1</td>
<td>J7</td>
<td>I SCALR</td>
<td>00014</td>
<td>1</td>
<td>J7</td>
<td>I SCALR</td>
<td>00015</td>
<td>1</td>
<td>J7</td>
<td>I SCALR</td>
<td>00016</td>
<td>1</td>
</tr>
<tr>
<td>KEY</td>
<td>I SCALR</td>
<td>00023</td>
<td>V DUMMY</td>
<td>KEY</td>
<td>I SCALR</td>
<td>00024</td>
<td>V DUMMY</td>
<td>KEY</td>
<td>I SCALR</td>
<td>00025</td>
<td>V DUMMY</td>
<td>KEY</td>
<td>I SCALR</td>
<td>00026</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>K1</td>
<td>I SCALR</td>
<td>00008</td>
<td>1</td>
<td>K1</td>
<td>I SCALR</td>
<td>00009</td>
<td>1</td>
<td>K1</td>
<td>I SCALR</td>
<td>00010</td>
<td>1</td>
<td>K1</td>
<td>I SCALR</td>
<td>00011</td>
<td>1</td>
</tr>
<tr>
<td>K4</td>
<td>I SCALR</td>
<td>0000E</td>
<td>1</td>
<td>K4</td>
<td>I SCALR</td>
<td>0000F</td>
<td>1</td>
<td>K4</td>
<td>I SCALR</td>
<td>00010</td>
<td>1</td>
<td>K4</td>
<td>I SCALR</td>
<td>00011</td>
<td>1</td>
</tr>
<tr>
<td>K7</td>
<td>I SCALR</td>
<td>00014</td>
<td>1</td>
<td>K7</td>
<td>I SCALR</td>
<td>00015</td>
<td>1</td>
<td>K7</td>
<td>I SCALR</td>
<td>00016</td>
<td>1</td>
<td>K7</td>
<td>I SCALR</td>
<td>00017</td>
<td>1</td>
</tr>
<tr>
<td>LATH</td>
<td>I SCALR</td>
<td>00025</td>
<td>V DUMMY</td>
<td>LATH</td>
<td>I SCALR</td>
<td>00026</td>
<td>V DUMMY</td>
<td>LATH</td>
<td>I SCALR</td>
<td>00027</td>
<td>V DUMMY</td>
<td>LATH</td>
<td>I SCALR</td>
<td>00028</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>LINDEX</td>
<td>I SCALR</td>
<td>00027</td>
<td>V DUMMY</td>
<td>LINDEX</td>
<td>I SCALR</td>
<td>00028</td>
<td>V DUMMY</td>
<td>LINDEX</td>
<td>I SCALR</td>
<td>00029</td>
<td>V DUMMY</td>
<td>LINDEX</td>
<td>I SCALR</td>
<td>00030</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>NAVIN</td>
<td>SPRBG</td>
<td>EXTERN</td>
<td>70</td>
<td>NAVIN</td>
<td>SPRBG</td>
<td>EXTERN</td>
<td>70</td>
<td>NAVIN</td>
<td>SPRBG</td>
<td>EXTERN</td>
<td>70</td>
<td>NAVIN</td>
<td>SPRBG</td>
<td>EXTERN</td>
<td>70</td>
</tr>
<tr>
<td>N2</td>
<td>I SCALR</td>
<td>0002F</td>
<td>V DUMMY</td>
<td>N2</td>
<td>I SCALR</td>
<td>00030</td>
<td>V DUMMY</td>
<td>N2</td>
<td>I SCALR</td>
<td>00031</td>
<td>V DUMMY</td>
<td>N2</td>
<td>I SCALR</td>
<td>00032</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>PINT</td>
<td>R SCALR</td>
<td>00000</td>
<td>1</td>
<td>PINT</td>
<td>R SCALR</td>
<td>00001</td>
<td>1</td>
<td>PINT</td>
<td>R SCALR</td>
<td>00002</td>
<td>1</td>
<td>PINT</td>
<td>R SCALR</td>
<td>00003</td>
<td>1</td>
</tr>
<tr>
<td>PLONG</td>
<td>R SCALR</td>
<td>00010</td>
<td>1</td>
<td>PLONG</td>
<td>R SCALR</td>
<td>00011</td>
<td>1</td>
<td>PLONG</td>
<td>R SCALR</td>
<td>00012</td>
<td>1</td>
<td>PLONG</td>
<td>R SCALR</td>
<td>00013</td>
<td>1</td>
</tr>
<tr>
<td>RLLAT</td>
<td>R SCALR</td>
<td>00018</td>
<td>1</td>
<td>RLLAT</td>
<td>R SCALR</td>
<td>00019</td>
<td>1</td>
<td>RLLAT</td>
<td>R SCALR</td>
<td>00020</td>
<td>1</td>
<td>RLLAT</td>
<td>R SCALR</td>
<td>00021</td>
<td>1</td>
</tr>
<tr>
<td>STMK</td>
<td>R SCALR</td>
<td>00036</td>
<td>V DUMMY</td>
<td>STMK</td>
<td>R SCALR</td>
<td>00037</td>
<td>V DUMMY</td>
<td>STMK</td>
<td>R SCALR</td>
<td>00038</td>
<td>V DUMMY</td>
<td>STMK</td>
<td>R SCALR</td>
<td>00039</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>WGTN</td>
<td>R SCALR</td>
<td>00038</td>
<td>V DUMMY</td>
<td>WGTN</td>
<td>R SCALR</td>
<td>00039</td>
<td>V DUMMY</td>
<td>WGTN</td>
<td>R SCALR</td>
<td>00040</td>
<td>V DUMMY</td>
<td>WGTN</td>
<td>R SCALR</td>
<td>00041</td>
<td>V DUMMY</td>
</tr>
</tbody>
</table>

**Local Variables (31 Words):**

- 00000 PINBT
- 00006 IREC1
- 00007 K3
- 00008 KEY
- 00009 J7
- 00010 K8
- 00011 K9
- 00012 K10
- 00013 J11
- 00014 K12
- 00015 J13
- 00016 K14
- 00017 J15
- 00018 K16
- 00019 J17
- 00020 K18
- 00021 J19
- 00022 K20
- 00023 J21
- 00024 K22
- 00025 J23
- 00026 K24
- 00027 J25
- 00028 K26
- 00029 J27
- 00030 K28
- 00031 J29
- 00032 K30
- 00033 J31
- 00034 K32
- 00035 J33

**Blank Common (0 Words):**

**Entry Points:**

- 00000 PINBT

**Intrinsic Subprograms Used:**

- FLBAT
SUBROUTINE PLANET(KK,RRM,PMSS,GM,PCENS)
VERSION OF 23 MAR 1973, CHANGING SENSE SWITCH NUMBER
VERSION OF 27 FEB 1973, INITIAL VERSION
SUBROUTINE PLANET, RETURNS PARAMETER VALUES FOR
PLANET SELECTED BY SSW(38 & 39)
SSW(38) = 0 FOR SPHERICAL EARTH
= 1 FOR SPHERICAL MOON
= 2 FOR SPHERICAL MARS
RRM = RADIALS IN KILOMETERS
PMSS = PLANET MASS IN KILOGRAMS
GM = GM IN DYNES/GRAMS SQUARED
PCENS = MEAN DENSITY IN GRAMS/CC

DATA ISRT/0/
IF(ISRT)1C5*10
E OUTPUT 'SUBROUTINE PLANET, VERSION OF 23 MAR, 1973'
1C IF(ISRT)
2C IF(IP)2C20*20*30
Spherical Earth
2C RRM= 6371.2213
2C PMSS= 5.983E+24
2C GM= 3.940661E+20
2C PCENS= 5.51
2C GB TO 900
3C CONTINUE
28C GB TO(4C,50)IP
Spherical Moon
4C RRM=1738.*0
2C PMSS= 7.3554E+22
2C GM= 4.90605E+18
2C PCENS= 3.34
2C REF = FIELDER, 1961, P. 245
3C GB TO 900
Spherical Mars
5C RRM=3394.*0
2C PMSS= 0.*
2C GM= 0.*
2C PCENS= 3.56
2C GB TO 900
3C CONTINUE
50C CONTINUE
4C KK = IP
46C RETURN
E ND
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>R</td>
<td>SCARP</td>
<td>00006 V</td>
<td>DUMMY</td>
<td>ISRT</td>
<td>I</td>
<td>SCARP</td>
<td>00001 V</td>
<td>1</td>
<td>PDENS</td>
<td>R</td>
<td>SCARP</td>
<td>00007 V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>ISW</td>
<td>I</td>
<td>SCARP</td>
<td>00000 P</td>
<td></td>
<td>KK</td>
<td>I</td>
<td>SCARP</td>
<td>00002 V</td>
<td>1</td>
<td>PHASS</td>
<td>R</td>
<td>SCARP</td>
<td>00005 V</td>
<td>DUMMY</td>
</tr>
<tr>
<td>PLANET</td>
<td>R</td>
<td>SCARP</td>
<td>00000 V</td>
<td>DUMMY</td>
<td>PLANET</td>
<td>R</td>
<td>SCARP</td>
<td>00003 V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
<th>HEX</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>C000A</td>
<td>0000A</td>
<td>00000</td>
<td>0001A</td>
<td>20</td>
<td>00020</td>
<td>30</td>
<td>00029</td>
<td>40</td>
<td>00030</td>
<td>50</td>
<td>00039</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (3 WORDS):

C000C PLANET  C0001 ISRT  00002 IP

BLANK COMMON (0 WORDS)

ENTRY POINTS:

C000C PLANET

EXTERNAL SUBPROGRAMS REQUIRED:

ISW  F1108  SENDCOL  9PRINT  9SETUP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC WORDS | HEX WORDS
----------|-----------

GENERATED CODE: 69  C0045
CONSTANTS: 11  C000B
LOCAL VARIABLES: 3  C0003
TEMPs: 6  C0006

TOTAL PROGRAM: 89  C0083
SUBROUTINE PLOATA(NS, A(1:NL), NL, NS, NSCALE, IZERO, TMIN, TMAX)

CATALOG NUMBER 890004 PLOT

NOTE THAT THE NAME HAS BEEN CHANGED FROM PLOT TO PLOATA

TO AVOID CONFLICT WITH THE CALCOMP SUBROUTINES

A ----- THE ARRAY IN WHICH THE INDEPENDENT VARIABLE AND THE
DEPENDANT VARIABLES ARE STORED COLUMN WISE. IF THE ARRAY IS DI

E dimensioned A(1,M) IN THE CALLING PROGRAM THE INDEPENDANT VARIAB

LE IS IN A(1,1) THRU A(1,N), THE FIRST VARIABLE TO BE PLOTTED IS

STORED IN A(1,2) THRU A(N,2) AND SO ON.

NL ----- THE NUMBER OF POINTS IN EACH COLUMN OF THE ARRAY THAT

WE WISH TO PLOT.

NS = 1 TO REORDER THE ARRAY SO THAT THE VALUES OF THE INDEPENDANT

VARIABLE INCREASED INCREASING ORDER.

* THE INDEPENDANT VARIABLE IS ALREADY STORED IN INCREASING ORDER.

ISCALE = 0 SCALE THE ARRAY

ISCALE = 1 DO NOT SCALE THE ARRAY, USE THE LIMITS TMAX, TMIN

AND IF IT IS GREATER THAN TMAX OR TMIN PUT THE VARIABLE AT THE

EDGE.

IZERO = 0 DO NOT PUT IN ZERO LINE, = 1 PUT IN ZERO LINE

TMAX, TMIN MAXIMUM AND MINIMUM VALUES PLOTTED IF ISCALE=1

DIMENSION BUT(103), YPR(11), ANG(9), A(1)

DATA PLAN=XANG/*1, 11, 12, 13, 14, 15, 16, 17, 18, 19*/

REAL LINE ,

DATA LINE/*11*/

1 FORMAT(1H16X28#1910 (tower name))

2 FORMAT(1H16X28#1910 (tower name))

3 FORMAT(1H16X28#1910 (tower name)))

4 FORMAT(1H16X28#1910 (tower name)))

5 FORMAT(1H16X28#1910 (tower name)))

6 FORMAT(1H16X28#1910 (tower name)))

7 FORMAT(1H16X28#1910 (tower name)))

8 FORMAT(1H16X28#1910 (tower name)))

9 FORMAT(1H16X28#1910 (tower name)))

10 FORMAT(1H16X28#1910 (tower name)))

11 FORMAT(1H16X28#1910 (tower name)))

12 FORMAT(1H16X28#1910 (tower name)))

13 FORMAT(1H16X28#1910 (tower name)))

14 FORMAT(1H16X28#1910 (tower name)))

15 FORMAT(1H16X28#1910 (tower name)))

16 IF(ALL) 20, 18, 20

17 ALL=50
60* 20 WRITE(10T,1)N0
61* WRITE(107,7)
62* XSCAL=(A(NLL)-A(1))/(FLBAT(NLL-1))
63* IF(ISCALE*EG+1) GO TO 42
64* YMIN=1.0E75
65* YMAX=1.0E75
66* DB 40 MC=M+1
67* M1=M+1
68* M2=M+1
69* DB 40 W=M+1
70* IF(A(J)=YMIN) 28,26,26
71* 26 IF(A(J)=YMAX) 40,40,30
72* 22 YMIN=A(J)
73* GS TO 40
74* 3C YMAX=A(J)
75* 4C CONTINUE
76* 42 CONTINUE
77* YSCAL=(YMAX-YMIN)/100.0
78* XB=A(1)
79* L=1
80* M=1
81* DB 80 I=1\#NLL
82* F=1=1
83* XFR=A(L)
84* 5C DB 55 IX=1=102
85* 5B GUT(IX)=BLANK
86* GUT(1)=LINE
87* GUT(103)=LINE
88* IF(IZER=EQ=0) GO TO 59
89* W=YMIN/YSCAL+1.0
90* IF(JZ>1 AND JZ<103) GUT(JZ)=LINE
91* 59 CONTINUE
92* DB 60 W=1=1
93* 5L LL=L+1\#N
94* JP=((A(LL)-YMIN)/YSCAL)+2.0
95* IF(JP>103) GUT(103)=ANG(J) GO TO 60
96* IF(JP<1) GUT(JP)=ANG(J) GO TO 60
97* 6C GUT(LP)=ANG(J)
98* CONTINUE
99* WRITE(10T,2)XPHe(88)GUT(IZ),IZ=1=103)
100* L=L+1
101* GO TO 80
102* 7C WRITE(10T,3)
103* 3C CONTINUE
104* WRITE(10T,7)
105* YPR(1)=YMIN
106* DB 90 KN=1=9
107* 9C YPR(KN+1)=YPR(KN)+YSCAL*10.0
108* 108 YPR(11)=YMAX
109* WRITE(10T,8)(YPR(IP),IP=1=11)
110* 78 FORMAT(10E11.6,1X)
111* 78 RETURN
112* ENC
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WBRDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WBRDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LBC</th>
<th>DEC WBRDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>ARRAY</td>
<td>000007</td>
<td>V DUMMY</td>
<td>ANG</td>
<td>R</td>
<td>ARRAY</td>
<td>000073</td>
<td>V</td>
<td>9</td>
<td>BLANK</td>
<td>R</td>
<td>SCALR</td>
<td>00007C</td>
</tr>
<tr>
<td>F</td>
<td>R</td>
<td>SCALR</td>
<td>000067</td>
<td>V</td>
<td>IP</td>
<td>I</td>
<td>SCALR</td>
<td>000020</td>
<td>V</td>
<td>1</td>
<td>ISCALE</td>
<td>I</td>
<td>SCALR</td>
<td>00009C</td>
</tr>
<tr>
<td>10T</td>
<td>I</td>
<td>SCALR</td>
<td>000068</td>
<td>V</td>
<td>JZ</td>
<td>I</td>
<td>SCALR</td>
<td>000093</td>
<td>V</td>
<td>1</td>
<td>IZER0</td>
<td>I</td>
<td>SCALR</td>
<td>000090</td>
</tr>
<tr>
<td>IX</td>
<td>I</td>
<td>SCALR</td>
<td>000069</td>
<td>V</td>
<td>JP</td>
<td>I</td>
<td>SCALR</td>
<td>000082</td>
<td>V</td>
<td>1</td>
<td>JZ</td>
<td>I</td>
<td>SCALR</td>
<td>000081</td>
</tr>
<tr>
<td>J</td>
<td>I</td>
<td>SCALR</td>
<td>000083</td>
<td>V</td>
<td>KN</td>
<td>I</td>
<td>SCALR</td>
<td>000094</td>
<td>V</td>
<td>1</td>
<td>L</td>
<td>I</td>
<td>SCALR</td>
<td>000084</td>
</tr>
<tr>
<td>LINE</td>
<td>R</td>
<td>SCALR</td>
<td>000086</td>
<td>V</td>
<td>MY</td>
<td>I</td>
<td>SCALR</td>
<td>000085</td>
<td>V</td>
<td>1</td>
<td>M</td>
<td>I</td>
<td>SCALR</td>
<td>000099</td>
</tr>
<tr>
<td>K</td>
<td>I</td>
<td>SCALR</td>
<td>000087</td>
<td>V</td>
<td>NL</td>
<td>I</td>
<td>SCALR</td>
<td>000098</td>
<td>V DUMMY</td>
<td>1</td>
<td>NL</td>
<td>I</td>
<td>SCALR</td>
<td>00009A</td>
</tr>
<tr>
<td>H2</td>
<td>I</td>
<td>SCALR</td>
<td>000089</td>
<td>V</td>
<td>NS</td>
<td>I</td>
<td>SCALR</td>
<td>000096</td>
<td>V DUMMY</td>
<td>1</td>
<td>NS</td>
<td>I</td>
<td>SCALR</td>
<td>000098</td>
</tr>
<tr>
<td>ALL</td>
<td>I</td>
<td>SCALR</td>
<td>000081</td>
<td>V</td>
<td>PB</td>
<td>I</td>
<td>SCALR</td>
<td>000001</td>
<td>V</td>
<td>1</td>
<td>PB</td>
<td>I</td>
<td>SCALR</td>
<td>000000</td>
</tr>
<tr>
<td>BUT</td>
<td>R</td>
<td>ARRAY</td>
<td>000001</td>
<td>V 1C3</td>
<td>PEX</td>
<td>R</td>
<td>SCALR</td>
<td>00000F</td>
<td>V DUMMY</td>
<td>1</td>
<td>PEX</td>
<td>R</td>
<td>SCALR</td>
<td>000039</td>
</tr>
<tr>
<td>TFIN</td>
<td>R</td>
<td>SCALR</td>
<td>00000E</td>
<td>V DUMMY</td>
<td>XR</td>
<td>R</td>
<td>SCALR</td>
<td>00002E</td>
<td>V</td>
<td>1</td>
<td>XR</td>
<td>R</td>
<td>SCALR</td>
<td>00008D</td>
</tr>
<tr>
<td>XPR</td>
<td>R</td>
<td>SCALR</td>
<td>00002F</td>
<td>V</td>
<td>YMAX</td>
<td>R</td>
<td>SCALR</td>
<td>000088</td>
<td>V</td>
<td>1</td>
<td>YMAX</td>
<td>R</td>
<td>SCALR</td>
<td>00007E</td>
</tr>
<tr>
<td>YMAX</td>
<td>R</td>
<td>SCALR</td>
<td>00007F</td>
<td>V</td>
<td>YSCAL</td>
<td>R</td>
<td>SCALR</td>
<td>000068</td>
<td>V</td>
<td>11</td>
<td>YSCAL</td>
<td>R</td>
<td>SCALR</td>
<td>00006C</td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (150 WBRDS):

<table>
<thead>
<tr>
<th>LABEL LBC</th>
<th>LABEL LBC</th>
<th>LABEL LBC</th>
<th>LABEL LBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000 F8TA</td>
<td>00000 C1</td>
<td>0000068 YPR</td>
<td>000073 ANG</td>
</tr>
<tr>
<td>00007E YMAX</td>
<td>000081 NLL</td>
<td>000082 I</td>
<td>000083 J</td>
</tr>
<tr>
<td>000084 L</td>
<td>000086 F</td>
<td>000088 XSCAL</td>
<td>000089 MB</td>
</tr>
<tr>
<td>000085 P1</td>
<td>000088 P2</td>
<td>00008C YSCL</td>
<td>00008E MY</td>
</tr>
<tr>
<td>000089 IX</td>
<td>000091 JZ</td>
<td>000092 JP</td>
<td>000093 IZ</td>
</tr>
</tbody>
</table>

BLANK COMMON (G WORDS):

ENTRY POINTS:

000000 F8TA

INTRINSIC SUBPROGRAMS USED:

FLOAT

EXTERNAL SUBPROGRAMS REQUIRED:

F:102 S:104 F:106 F:108 98CDWRT 9ENDIBL 910DATA 91TBR
**HIGHEST ERROR SEVERITY: 0 (NO ERRORS)**

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODES</td>
<td>320</td>
<td>00140</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>5</td>
<td>00005</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>150</td>
<td>00096</td>
</tr>
<tr>
<td>TEMPS</td>
<td>13</td>
<td>0000D</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>488</td>
<td>001E8</td>
</tr>
</tbody>
</table>
SUBROUTINE RETBY

A, IDATA, IEBC, IIN, IISUT, ITAPE, NUMPL, DATA, RLAT, RLONG, K60HM, IAGAP, LCN

BT, RACED, DEGRA, KCEG2, IDEG2, FDEG2, RCEG2, RTOP, ITOP, RBOT, IBOT, RLEFT,

C ILEFT, IRIGT, IRIGT, SLTK, SLOK, SINC, SHP, FTOP, FTOP, FLEFT, FRIGT, NDEG.

C SLAT, RLONG, BBTMP, XX, YY, INIT, X0LD, Y0LD)

SUBROUTINE RETBY, TO SET CHART BOUNDARIES

WRITE (IIBT, 10)

CS FORMAT ('E BOUNDS 1 LINE!')

CS READ(IIN, *) ITOP, IBOT, ILEFT, IRIGT

CS READ(IIN, *) ISFT, ISFOR, IBOT, ILEFT, IRIGT

FORMAT (4I5)

5 OUTFLX ITOP, IBOT, ILEFT, IRIGT

FTOP = ITOP

FBOT = IBOT

FLEFT = ILEFT

FRIGT = IRIGT

RTOP = FTOP * DEGRA

RBOT = FBOT * DEGRA

RLEFT = FLEFT * DEGRA

RIRIGT = IRIGT * DEGRA

KDEG2 = (IRIGT - ILEFT)

IDEG2 = KDEG2

FDEG2 = IDEG2

RDEG2 = FDEG2 * DEGRA

SMP = SINC / 60 * C

RETURN

ENC
SUBROUTINE RTDM2(RAD, IDEG, AMIN)
C SUBROUTINE RTDM TO CONVERT RADIANS (RAD) TO
C DEGREES (IDEG) AND MINUTES (AMIN)
C FOR AMIN WITH 2 DIGITS TO RIGHT OF DECIMAL

B = RAD * 57.29578
A = ABS(B)
A = A + 0.00005
A = SIGN(A, B)
IDEG = A
A = IDEG
AMIN = (B - A) * 60.0
AMIN = SIGN(AMIN, RAD)
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALAR</td>
<td>COCO02</td>
<td>V 1</td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>SCALAR</td>
<td>COCO01</td>
<td>V 1</td>
</tr>
<tr>
<td>RTDM2</td>
<td></td>
<td></td>
<td>CCCC00</td>
<td>P</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (3 WORDS):**
- COCO00 RTDM2
- COCO01 B
- COCO02 A

**BLANK COMMON (0 WORDS)**

**ENTRY POINTS:**
- COCO00 RTDM2

**INTRINSIC SUBPROGRAMS USED:**
- ABS
- SIGN

**EXTERNAL SUBPROGRAMS REQUIRED:**
- $TBR
- $RTBI
- $SETUP

**HIGHEST ERROR SEVERITY:** 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>COCO03</td>
</tr>
</tbody>
</table>

**GENERATED CODE:** 35
**CONSTANTS:** 3
**LOCAL VARIABLES:** 3
**TEMPs:** 4

**TOTAL PROGRAM:** 45 COCO20
SUBROUTINE RT0DM(RAD, IDEG, AMIN)
C SUBROUTINE RT0DM TO CONVERT RADIANS (RAD) TO
C DEGREES (IDEG) AND MINUTES (AMIN)
C FOR AMIN WITH 0 DIGITS TO RIGHT OF DECIMAL
R = RAD*57.29578
A = ABS(B)
A = A + 0.005
A = SIGN(A, B)
IDEG = A
A = IDEG
AMIN = (R-A)*60.0
AMIN = SIGN(AMIN, RAD)
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>SCALR</td>
<td>00002 V</td>
<td>1</td>
<td>ABS</td>
<td>R</td>
<td>SPROG</td>
<td>INTRIN</td>
<td>00005 V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>R</td>
<td>SCALR</td>
<td>00001 V</td>
<td>1</td>
<td>1DEG</td>
<td>I</td>
<td>SCALR</td>
<td>00004 V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RT8CH</td>
<td>SPROG</td>
<td>00000 P</td>
<td></td>
<td></td>
<td>RT8CH</td>
<td>R</td>
<td>SCALR</td>
<td>00000 V</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (3 WORDS):
- 00000 RT8CH
- 00001 B
- 00002 A

BLANK COMMON (0 WORDS)

ENTRY POINTS:
- 00000 RT8CH

INTRINSIC SUBPROGRAMS USED:
- ABS
- SIGN

EXTERNAL SUBPROGRAMS REQUIRED:
- 9ITOR
- 9RT81
- 9SETUP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>00023</td>
</tr>
<tr>
<td>3</td>
<td>00003</td>
</tr>
<tr>
<td>3</td>
<td>00003</td>
</tr>
<tr>
<td>4</td>
<td>00004</td>
</tr>
</tbody>
</table>

TOTAL PROGRAM: 45 0002D
SUBROUTINE SIMUL(N, A, X, EPS, INDIC, NRC, DETER)


REFERENCE: CARNABAH, LUTHER AND MILKES (1959) APPLIED NUMERICAL METHODS. WILEY, NEW YORK.

CONVERTED TO XDS FORTRAN IV+ BY H. FERKINS APRIL 1970.

DIMENSION IRW(50), JCOL(50), JORD(50), Y(50), A(NRC,NRC), X(N)

MAX = N
IF ( INDIC GE 0 ) MAX = N + 1

I IS N LARGER THAN 5c
IF ( A.LE.5e ) GO TO 5
WRITE(10B20C)
SIMUL = 0
RETURN

BEGIN ELIMINATION PROCEDURE
DETER = 1.

CB 18 K = 1, N
KM1 = K = 1

SEARCH FOR THE PIVOT ELEMENT
PIVOT = 0.

SCAN IRW AND JCOL ARRAYS FOR INVALID PIVOT SUBSCRIPTS
IF ( K.EQ.1 ) GO TO 9

SCAN = 1, KM1
GO TO 11

IF ( I.EQ.IROW(ISCAN) ) GO TO 11

GO TO 5
**C**

8 IF (J<>E+J*JCBL(JSCAN)) GO TO 11

9 IF (CABS(A(I,J))+LE*CABS(PIVOT)) GO TO 11

11 CONTINUE

C

... INSURE THAT SELECTED PIVOT IS LARGER THAN EPS ...

13 IF (CABS(PIVOT)+GT+EPS) GO TO 13

16 SIMUL = 0

17 RETURN

C

... UPDATE THE DETERMINANT VALUE ...

19 IRB(K) = IRB(K)

20 JCOL(K) = JCOL(K)

21 DETER = DETER*PIVOT

25 CONTINUE

C

... NORMALIZE PIVOT ROW ELEMENTS ...

27 CB 14 J = 1, MAX

28 AI(ROB,K,J) = A(ROB,K,J)/PIVOT

29 C

... CARRY OUT ELIMINATION AND DEVELOP INVERSE ...

31 AI(ROB,K,J) = 1/PIVOT

32 CB 18 I = 1, MAX

33 AI(J) = AI(J)/PIVOT

34 CB 17 J = 1, MAX

35 IF (J<>E+J*JCBL(K)) GO TO 18

36 CONTINUE

C

... ORDER SOLUTION VALUES (IF ANY) AND CREATE JORD ARRAY ...

38 CB 22 I = 1, MAX

39 IRB(I) = IRB(I)

40 JCOL(I) = JCOL(I)

41 JORD(I) = JORD(I)

49 IF (INDIC=EX+G) X(JCOL(I)) = A(IRB(I),MAX)

48 CONTINUE

C

... ADJUST SIGN OF DETERMINANT ...

51 IF (EX+G) GO TO 10

52 INTCH = 0

53 NMP = 1

54 CB 22 I = 1, MAX

55 IF (IP1 = I + 1)

56 IF (JORD(I)+GE+JORD(1)) GO TO 22

57 TEMP = JORD(I)

58 JORD(I) = TEMP

59 INTCH = INTCH + 1

60 CONTINUE

61 IF (INTCH/2*1+INTCH) DETER = - DETER

C

... IF INDIC IS POSITIVE RETURN WITH RESULTS ...

64 IF (INDIC=EX+G) GO TO 24

66 SIMUL = DETER

67 RETURN

C

... IF INDIC IS NEGATIVE OR ZERO, UNSCRAMBLE THE INVERSE ...

69 FIRST BY REVNS ...

70 GO TO 11

71 GO TO 0

72 GO TO 1

73 GO TO 1

74 GO TO 1

75 GO TO 1

76 GO TO 1

77 GO TO 1

78 GO TO 1

79 GO TO 1

80 GO TO 1

81 GO TO 1

82 GO TO 1

83 GO TO 1

84 GO TO 1

85 GO TO 1

86 GO TO 1

87 GO TO 1

88 GO TO 1

89 GO TO 1

90 GO TO 1

91 GO TO 1

92 GO TO 1

93 GO TO 1

94 GO TO 1

95 GO TO 1

96 GO TO 1

97 GO TO 1

98 GO TO 1

99 GO TO 1

100 GO TO 1

101 GO TO 1

102 GO TO 1

103 GO TO 1

104 GO TO 1

105 GO TO 1

106 GO TO 1

107 GO TO 1

108 GO TO 1

109 GO TO 1

110 GO TO 1

111 GO TO 1

112 GO TO 1

113 GO TO 1

114 GO TO 1

115 GO TO 1

116 GO TO 1

117 GO TO 1

118 GO TO 1

119 GO TO 1
120* 26 CP 28  J = 1, N
121* 27 CB 27  I = 1, N
122* IR8 = IR8[I]
123* LCBL = LCBL[I]
124* Y(CBL) = A(IR8[I,J])
125* CB 28  I = 1, N
126* 28 A(I,J) = Y(I)
127* C ****** THEN BY COLUMNS ******
128* CB 30  I = 1, N
129* CB 29  J = 1, N
130* IR8 = IR8(J)
131* LCBL = LCBL(J)
132* 29 Y(IR8,J) = A(I,JCOL(J))
133* CB 30  J = 1, N
134* 30 A(I,J) = Y(J)
135* C ****** RETURN FOR INDIC NEGATIVE OR ZERO ******
136* SIMUL = DETER
137* RETURN
138* C ****** FORMAT FOR OUTPUT STATEMENT ******
139* 2DC FORMAT (1CFON 10B BIG )
140* C
141* 142* END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>R</td>
<td>ARRAY</td>
<td>CCC00</td>
<td>DUMMY</td>
<td>A12C</td>
<td>R</td>
<td>SCALAR</td>
<td>CCC03</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC0D</td>
<td>V</td>
</tr>
<tr>
<td>CTER</td>
<td>R</td>
<td>SCALAR</td>
<td>CCC02</td>
<td>DUMMY</td>
<td>EPS</td>
<td>R</td>
<td>SCALAR</td>
<td>CCC06</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC08</td>
<td>DUMMY</td>
</tr>
<tr>
<td>ISDC</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC00</td>
<td>DUMMY</td>
<td>INTCH</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC04</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC0E</td>
<td>V</td>
</tr>
<tr>
<td>IRW</td>
<td>I</td>
<td>ARRAY</td>
<td>CCC01</td>
<td>50</td>
<td>IR0W1</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC05</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC0A</td>
<td>V</td>
</tr>
<tr>
<td>IRWX</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC01</td>
<td>1</td>
<td>IR0WJ</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC06</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC0B</td>
<td>V</td>
</tr>
<tr>
<td>JGRL</td>
<td>I</td>
<td>ARRAY</td>
<td>CCC03</td>
<td>50</td>
<td>JGRL1</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC05</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC0D</td>
<td>V</td>
</tr>
<tr>
<td>JBLK</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC02</td>
<td>1</td>
<td>JB0D1</td>
<td>I</td>
<td>ARRAY</td>
<td>CCC07</td>
<td>V</td>
<td>50</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC0F</td>
<td>DUMMY</td>
</tr>
<tr>
<td>JTEMP</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC05</td>
<td>1</td>
<td>J</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC06</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC10</td>
<td>V</td>
</tr>
<tr>
<td>MAX</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC09</td>
<td>1</td>
<td>NM1</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC10</td>
<td>V</td>
<td>I</td>
<td>J</td>
<td>SCALAR</td>
<td>CCC14</td>
<td>V</td>
</tr>
<tr>
<td>NRC</td>
<td>I</td>
<td>SCALAR</td>
<td>CCC0E</td>
<td>V DUMMY</td>
<td>X</td>
<td>R</td>
<td>ARRAY</td>
<td>CCC16</td>
<td>V DUMMY</td>
<td>Y</td>
<td>R ARRAY</td>
<td>CCC097</td>
<td>V</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCAL VARIABLES (220 WORDS):</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCC00 SIMUL</td>
</tr>
<tr>
<td>CCC01 IRW</td>
</tr>
<tr>
<td>CCC02 K</td>
</tr>
<tr>
<td>CCC03 JGRL</td>
</tr>
<tr>
<td>CCC04 JBLK</td>
</tr>
<tr>
<td>CCC05 JTEMP</td>
</tr>
<tr>
<td>CCC06 INTCH</td>
</tr>
<tr>
<td>CCC07 NM1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BLANK COMMON (10 WORDS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTRY POINTS:</td>
</tr>
<tr>
<td>CCC00 SIMUL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INTRINSIC SUBPROGRAMS USED:</th>
</tr>
</thead>
<tbody>
<tr>
<td>DABS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EXTERNAL SUBPROGRAMS REQUIRED:</th>
</tr>
</thead>
<tbody>
<tr>
<td>F11C0</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td>GENERATED CODE</td>
</tr>
<tr>
<td>CONSTANTS</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
</tr>
<tr>
<td>TEMPS1</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
</tr>
</tbody>
</table>
SUBROUTINE SPLT(IS,RLAT,RLONG,VEL,THICK,WATTK,VMANT,XX,YY,
1 YFAC,ZHT,HGT,AC,ANG,BDIST)
C SUBROUTINE SPLT, PLOTS SEISMIC REFRACTION PROFILE DATA IN COLUMN FORM
C
DIMENSION VEL(8),THICK(8)
CATA RADEG/57.29578/
ACT=8
C
C LABLING COLUMN
AISTA = ISTA
XT = C*X(1) + Y(1)
YT = C*Y + X
XF = (X*AC) + (Y*AS)
YP = -1*C*(X*AS) + (Y*AC)
XT = XX + XP
YT = YY*YP
CALL NUMBER(XT,YT,HGT,AISTA,ANG,-1)
CALL PLT(XX,YY,3)
IIFS(ISH(5))=50,465,490
46E XN = C*1C
YN = C*4C+(0*12*2HT)
XP = (X*AC) + (Y*AS)
YP = -1*C*(X*AS) + (Y*AC)
XT = XX + XP
YT = YY*YP
DGRAM = RLONG*RADEG
CALL NUMBER(XT,YT,HGT,DGRAM,ANGB,2)
27 XT = 0*1C
28 YT = C*4C+(0*24*2HT)
29 XF = (X*AC) + (Y*AS)
30 YP = -1*C*(X*AS) + (Y*AC)
31 XT = XX + XP
32 YT = YY*YP
33 DLAT = RLAT*RADEG
CALL NUMBER(XT,YT,HGT,DLAT,ANGB,2)
34 CALL PLT(XX,YY,3)
35 C BEGIN PLATTING COLUMN
LIND = 0
MAX = NCT+1
IIFS(VELW=C*001)510*510*5CC
50 CVEL = VELW
51 DBWK = WATTK/YFAC
52 G8 TB 100
53 LIND = LIND + 1
54 IIFS(LIND=MAX)512*530*53C
55 IIFS(VEL(LIND)=C*001)510*520
56 CVEL = VEL(LIND)
57 IIFS(THICK(LIND)=0*001)522*522*524
58 CVEL = CVEL+7*524*523*523
59 DBWK = BDIST + 2*0
G8 TB 100
60 DBWK = THICK(LIND)/YFAC
61 G8 TB 100
62 DBWK = (BDIST + 2*0)
G8 TB 100
63 DBWK = VMANT/C*001
55C G8 TB 75C
CC
CC
CC
C WRITING TICK LINE AT ANGLE (ANGB)
10C
   XP = (BDIST*AC)
   YP = -1.0*(BDIST * AS)
   XT = XX+(ABS(XP))
   YT = YY+YP
   CALL PLAT(XT, YT, 2)
   CALL PLAT(xx, yy, 3)

C WRITING VELOCITY VALUE
11C
   XN = -0.25 * ZHT
   YN = -0.085 * ZHT
   XF = (XN*AC)+(YN*AS)
   YP = -1.0*(XN*AS)+(YN*AC)
   XT=XX+XP
   YT=YY+YP
   CALL NUMBER(XT, YT, HGT, CVEL, ANGB, 1)
   CALL PLAT(xx, yy, 3)

C CONTINUE PLOTTING DOWNWARD LINE
12C
   XF = DWNKN *AS
   YF = DWNKN *AC
   XX = XX+XP
   YY = YY+YP
   CALL PLAT(XX, YY, 2)
   GB TO 51C
75C RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>R</td>
<td>SPROG</td>
<td>INTRIN</td>
<td>0000F</td>
<td>AC</td>
<td>R</td>
<td>SCALR</td>
<td>0001D</td>
<td>V DUMMY</td>
<td>AISTA</td>
<td>R</td>
<td>SCALR</td>
<td>00003</td>
<td>V</td>
</tr>
<tr>
<td>ANGB</td>
<td>R</td>
<td>SCALR</td>
<td>0000F</td>
<td>V DUMMY</td>
<td>AS</td>
<td>R</td>
<td>SCALR</td>
<td>0001E</td>
<td>V DUMMY</td>
<td>BDIST</td>
<td>R</td>
<td>SCALR</td>
<td>00020</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>CVEL</td>
<td>R</td>
<td>SCALR</td>
<td>000CE</td>
<td>1</td>
<td>DLAT</td>
<td>R</td>
<td>SCALR</td>
<td>000CB</td>
<td>1</td>
<td>DLONG</td>
<td>R</td>
<td>SCALR</td>
<td>0000A</td>
<td>V</td>
</tr>
<tr>
<td>DMB</td>
<td>R</td>
<td>SCALR</td>
<td>000CE</td>
<td>1</td>
<td>HGT</td>
<td>R</td>
<td>SCALR</td>
<td>0001C</td>
<td>V DUMMY</td>
<td>ISTA</td>
<td>I</td>
<td>SCALR</td>
<td>00001</td>
<td>00010</td>
</tr>
<tr>
<td>ISx</td>
<td>I</td>
<td>SCALR</td>
<td>EXTERN</td>
<td>00000</td>
<td>IIND</td>
<td>I</td>
<td>SCALR</td>
<td>0000C</td>
<td>1</td>
<td>MAX</td>
<td>I</td>
<td>SCALR</td>
<td>00000</td>
<td>D</td>
</tr>
<tr>
<td>NCT</td>
<td>I</td>
<td>SCALR</td>
<td>00002</td>
<td>1</td>
<td>NUMBER</td>
<td>SPROG</td>
<td>EXTERN</td>
<td>00000</td>
<td>1</td>
<td>PLOT</td>
<td>SPROG</td>
<td>EXTERN</td>
<td>00000</td>
<td>PLOT</td>
</tr>
<tr>
<td>RADES</td>
<td>R</td>
<td>SCALR</td>
<td>00001</td>
<td>1</td>
<td>R LAT</td>
<td>R</td>
<td>SCALR</td>
<td>00011</td>
<td>V DUMMY</td>
<td>R BNG</td>
<td>R</td>
<td>SCALR</td>
<td>00002</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>SPL0T</td>
<td>R</td>
<td>SCALR</td>
<td>00000</td>
<td>1</td>
<td>SPLOT</td>
<td>SPROG</td>
<td>00000</td>
<td>P</td>
<td>VMANT</td>
<td>R</td>
<td>SCALR</td>
<td>00017</td>
<td>V DUMMY</td>
<td>1</td>
</tr>
<tr>
<td>WATK</td>
<td>R</td>
<td>SCALR</td>
<td>00016</td>
<td>V DUMMY</td>
<td>XN</td>
<td>R</td>
<td>SCALR</td>
<td>0000X</td>
<td>1</td>
<td>XP</td>
<td>R</td>
<td>SCALR</td>
<td>00006</td>
<td>V</td>
</tr>
<tr>
<td>XT</td>
<td>R</td>
<td>SCALR</td>
<td>00008</td>
<td>1</td>
<td>XX</td>
<td>R</td>
<td>SCALR</td>
<td>00018</td>
<td>V DUMMY</td>
<td>YFA C</td>
<td>R</td>
<td>SCALR</td>
<td>0001A</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>YN</td>
<td>R</td>
<td>SCALR</td>
<td>00005</td>
<td>1</td>
<td>YP</td>
<td>R</td>
<td>SCALR</td>
<td>00007</td>
<td>1</td>
<td>YT</td>
<td>R</td>
<td>SCALR</td>
<td>00009</td>
<td>V</td>
</tr>
<tr>
<td>YY</td>
<td>R</td>
<td>SCALR</td>
<td>00019</td>
<td>DUMMY</td>
<td>ZMT</td>
<td>R</td>
<td>SCALR</td>
<td>0001B</td>
<td>V DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Local Variables (16 WORDS):**

- COCCO SPLIT: 00001 RACES
- COCC0 XP: 00007 VP
- COCC0 LIND: 00007 VP
- COCC0 MAX: 00007 VP

**Blank Common (0 Words):**

**Entry Points:**

- COCC0 SPLIT

**Intrinsic Subprograms Used:**

- ABS

**External Subprograms Required:**

- ISA
- NUMBER
- FLOAT
- SITOR
- SETUFA

**Hex Locations:**

```
<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>00005</td>
<td>110</td>
<td>000ED</td>
<td>120</td>
<td>CC18</td>
<td>465</td>
<td>C0049</td>
<td>450</td>
<td>0009C</td>
<td>500</td>
<td>000A4</td>
</tr>
<tr>
<td>510</td>
<td>COCA1</td>
<td>512</td>
<td>COAFC</td>
<td>520</td>
<td>COB8</td>
<td>522</td>
<td>C008A</td>
<td>523</td>
<td>000BD</td>
<td>524</td>
<td>000C2</td>
</tr>
</tbody>
</table>
```

**320 00068 540 000CB 550 000D4 750 0012A**
**HIGHEST ERROR SEVERITY: 0 (NO ERRORS)**

<table>
<thead>
<tr>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>GENERATED CODE: 299</td>
<td>C0128</td>
</tr>
<tr>
<td>CONSTANTS: 16</td>
<td>C0C1C</td>
</tr>
<tr>
<td>LOCAL VARIABLES: 16</td>
<td>C0C1C</td>
</tr>
<tr>
<td>TEMPS: 19</td>
<td>C0G13</td>
</tr>
<tr>
<td>TOTAL PROGRAM: 350</td>
<td>C015E</td>
</tr>
</tbody>
</table>
SUBROUTINE SPOT(XX, YY)

C SUBROUTINE SPOT PLATS A CIRCLE AT DATA POINT

C USES CALCOMP SUBROUTINES

ST = XX + C*0.2
TT = YY - 0*01
CALL PLOT(ST,TT,3)
YT = YY + 0*01
CALL PLOT(ST,YT,2)
XT = XX + 0*01
YT = YY + 0*02
CALL PLOT(XT,YT,2)
XT = XX - 0*01
CALL PLOT(XT,YT,2)
XT = XX - 0*02
YT = YY + 0*01
CALL PLOT(XT,YT,2)
YT = YY - 0*01
CALL PLOT(XT,YT,2)
XT = XX - 0*01
YT = YY - 0*02
CALL PLOT(XT,YT,2)
XT = XX + 0*01
CALL PLOT(XT,YT,2)
CALL PLOT(XX,YY,3)
RETURN
END
### Local Variables (5 Words):
- COGCG 0000 SP0T 00001 ST
- 00002 TT 00003 YT 00004 XT

### Blank Common (0 Words)
- COGCG SP0T

### Entry Points:
- COGCG SP0T

### External Subprograms Required:
- PLOT 9SETUP2

### Highest Error Severity: C (No Errors)

<table>
<thead>
<tr>
<th>Dec Words</th>
<th>Hex Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE:</td>
<td>91</td>
</tr>
<tr>
<td>CONSTANTS:</td>
<td>6</td>
</tr>
<tr>
<td>LOCAL VARIABLES:</td>
<td>5</td>
</tr>
<tr>
<td>TEMPS:</td>
<td>3</td>
</tr>
<tr>
<td>TOTAL PROGRAM:</td>
<td>105</td>
</tr>
</tbody>
</table>
SUBROUTINE SPOT2(XX,YY)
USES CALCOMP SUBROUTINES

MODIFIED TO MAKE SSW(4) DEFAULT NO MARKING 3 OCT 72
VERSION OF 20 AUGUST 1971 CHECKS ISM(4) TO DEFINE SYMBOL ANNOTATION

SSW(4) = 0 FOR SUPPRESSING PLOTTING OF ANY SPOT
1 FOR PLOTTING A CIRCLE AT DATA POINT
2 FOR PLOTTING A DOT AT DATA POINT

INCREMENTS INDEX BY ONE TO PERMIT USE OF GO TO STATEMENT

GO TO (999,100,200,300,400,500,600,700,800,900)

TO PLOT A CIRCLE AROUND DATA POINT

ST = XX + C*02
TT = YY - C*01
CALL FLT(ST,TT,3)

YT = YY + C*01
CALL FLT(ST,YT,2)

XT = XX + C*01
CALL FLT(XT,YT,2)

XT = XX + C*02
CALL FLT(XT,YT,2)

YT = YY - C*01
CALL FLT(XT,YT,2)

CALL FLT(ST,TT,2)

CALL FLT(XX,YY,3)
RETURN

TO PLOT ONLY A DOT BY LOWERING AND RAISING PEN

CALL FLT(XX,YY,2)
CALL FLT(XX,YY,3)

OTHER OPTIONS TO BE IMPLEMENTED

RETURN
RETURN
RETURN
RETURN
RETURN
RETURN
RETURN
RETURN
RETURN
RETURN
RETURN
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>TYPE CLASS</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
<th>NAME</th>
<th>LBC</th>
<th>MBCS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE TIDAL(RLAT,RLONG,IY,IM,CLS,HONK,DTD)

* WRITTEN BY C. GANTAF ACCORDING TO LONGMAN AND USCGS FORMULAE
ADAPTED FOR SIGMA-7 AUG. 69 BY C. KOLFE
* THE OUT PUT OF THE PROGRAM HAS BEEN CHECKED WITH THE G* PROG.
* BBD FOR THE YEAR 69. WEIRD RESULTS BUT ARE IN GENERAL AGREEMENT

* RLAT(+ IP ABRTH),RLONG(+ IF EAST)=GEOGR CBBRD.IN RADIANS
* IY, YEAR=19CC, ID=DAY, PROGRESSIVE OF THE YEAR
* IM=HOUR, MINUTES (GMT)
* CLS=TIDAL CORRECTION (MGAL)
* HONK+HONKASALB TERM (MGAL) TO BE ADDED TO CLS

* DOUBLE PRECISION D1Y,DIC,DTD,BIS,CENT,DS,S,SH,DP,DN,DP1,PI,P1
* MEGA,SP,SH,SP,SH,SH,S1,SENM,CS8M,SENNE,SENSU,SENL,CBSAL,
* DE,LL,CH,CI,CSZ,CT,DMUN,RH8,PC1,PC2,PC3,DUN,AN,SENI,FNU

* DOUBLE PRECISION N
* DOUBLE PRECISION RLAT,RLONG,CBSAL

* IMPLICIT REAL (*,*)
* X=FLOAT(IY)/4.**1
* X1=X=FLOAT(1X))

* IBIS=INT(X)
* IF(X1=LT*C+2)IBIS=IBIS+1
* CiY=IB
* CiC=ID+1
* CiH=14
* CyP=1'M

* CIPS=IBIS
* CTY=CTY+365*`CIBIS+0.`ECC+DIC+*C016666670+DIH++00069444D0*DYM

* DS=7.200005D+*C399+7.9275D+CENT+000035D+CENT+2
* S=CBD(DS+2831853C718C)
* C=4.886280D+*235355D+CENT+00005D+CENT+2
* P=CBN(DS+2831853C718C)
* CN=5.23603D+3375746D+CENT+00006D+CENT+2
* P1=CMDC(DP+2831853C718C)
* S=MEGA+4C0932D+0+000227D+CENT
* S=SP
* CIE=2.CC*SP
* SPY=5.2*DI+C+H+P
* CIE=2.CC+1(D+H)
* S1=7000.8DSIN(3P)+337675D*DSIN(DS)+0154001D0*DSIN(SHP)+C

* TIDAL(90.95973600000800000)*SIN(SENM)+CBSNE

* SENM+DSIN(QM)

* CBSNE+CCB(N)

* AI=AARC8(DARAT+99957036*CBSAN+8000308*SENME+CBSNE))

* SENI=DSIN(AI)

* SENL=CCB(N)+SENNE/SENI

* FAL=ARSIN(SENL)

* CBSAL+CBSNE+CBSF(SN)+SENN+SBN+CBSM

* CASAL+1.DC+CBSAL

* L=S1+4.DC+CATAN2(SENL,CBSAL)
DE1=033502080D0*=0000080C0*CENT
L1=H*DE1=DSIN(+P1)
G=57*296700W
GLONG=57*295780*RLONG
TH=FLBAT(1)*FLBAT(2)/60*
CHI=17453290=TH=15*L=180*D=GLONG*GH
CHI=CHI=FNL
CSZ=DSIN(RLAT)*SENI*DSIN(L)*DOC8(SRLAT)*((DOC8(AI/2+DI)*2+DOC8(L+C
1)*DSIN(AI/2+DI)*2+DOC8(L+C))
1=DSIN(RLAT)*SENI*DSIN(L)*DOC8(RLAT)*4*958781D*DOC8(L1=CHI1
1=DSIN(RLAT)*SENI*DSIN(L)*DOC8(RLAT)*4*958781D=DOC8(L1=CHI1
1=DSIN(RLAT)*SENI*DSIN(L)*DOC8(RLAT)*4*958781D=DOC8(L1=CHI1
1=DSIN(RLAT)*SENI*DSIN(L)*DOC8(RLAT)*4*958781D=DOC8(L1=CHI1
1=DSIN(RLAT)*SENI*DSIN(L)*DOC8(RLAT)*4*958781D=DOC8(L1=CHI1
1=DSIN(RLAT)*SENI*DSIN(L)*DOC8(RLAT)*4*958781D=DOC8(L1=CHI1
RETURN
END
### LOCAL VARIABLES (59 WORDS):

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Class</th>
<th>Hex LOC</th>
<th>Dec Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>R</td>
<td>SCALR</td>
<td>0005D</td>
<td>1</td>
</tr>
<tr>
<td>TIDAL</td>
<td>R</td>
<td>SCALR</td>
<td>00000</td>
<td>1</td>
</tr>
<tr>
<td>DLY</td>
<td>R</td>
<td>SCALR</td>
<td>00048</td>
<td>1</td>
</tr>
<tr>
<td>P1</td>
<td>D</td>
<td>SCALR</td>
<td>00016</td>
<td>1</td>
</tr>
<tr>
<td>P3</td>
<td>D</td>
<td>SCALR</td>
<td>00044</td>
<td>2</td>
</tr>
<tr>
<td>S</td>
<td>D</td>
<td>SCALR</td>
<td>00360</td>
<td>2</td>
</tr>
<tr>
<td>ENNU</td>
<td>D</td>
<td>SCALR</td>
<td>00020</td>
<td>2</td>
</tr>
<tr>
<td>SF</td>
<td>D</td>
<td>SCALR</td>
<td>00012</td>
<td>2</td>
</tr>
<tr>
<td>R</td>
<td>D</td>
<td>SCALR</td>
<td>00326</td>
<td>3</td>
</tr>
</tbody>
</table>

### BLANK CODE (10 WORDS)

- 00000 TIDAL
- 00000 DH
- 00000 DC
- 00000 S1
- 00000 SENAL
- 00000 PC2
- 00000 A
- 00000 IBIS

### ENTRY POINTS:

- 00000 TIDAL

### INTRINSIC SUBPROGRAMS USED:

- DBAS
- CATA2
- DCBS
- DMUL
- CSIN
- FLOAT
- INT

### EXTERNAL SUBPROGRAMS REQUIRED:

- ARCO
- ARSIN
- 9CATAN2
- 9COS
- 9SIN
- 9DMUL
- 9TAN
- 9SETH
- 9ITAN
- 9ITAN
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>DEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE</td>
<td>450</td>
<td>CC1C2</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>116</td>
<td>00074</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>99</td>
<td>00063</td>
</tr>
<tr>
<td>TEMPS</td>
<td>22</td>
<td>00016</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>687</td>
<td>002AF</td>
</tr>
</tbody>
</table>
SUBROUTINE VETBY

TO SET CHART BOUNDARIES BY OPERATOR ENTRY ON CONSOLE TTY
FOR NON INTEGER CHART BOUNDARIES

CALLS SUBROUTINE ARLIM

CALL ARLIM(INN,INOUT,RTOP,RTOT,LLEFT,RRIGHT)
SMP=SINCH/60*C
RDEG2=(RRIGHT-LLEFT)
FDEG2=RDEG2*RADC
RETURN
END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARLIM</td>
<td>SPROG</td>
<td>EXTERN</td>
<td></td>
<td></td>
<td>BBMP</td>
<td>UNUSED</td>
<td>00016</td>
<td>V DUMMY</td>
<td></td>
<td>DATA</td>
<td>UNUSED</td>
<td>00007</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>DEPRA</td>
<td>UNUSED</td>
<td>0001F</td>
<td>V DUMMY</td>
<td></td>
<td>FBOT</td>
<td>UNUSED</td>
<td>00022</td>
<td>V DUMMY</td>
<td></td>
<td>FDEG2</td>
<td>R SCALR</td>
<td>00011</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>FLEFT</td>
<td>UNUSED</td>
<td>00008</td>
<td>V DUMMY</td>
<td></td>
<td>FRIGT</td>
<td>UNUSED</td>
<td>00016</td>
<td>V DUMMY</td>
<td></td>
<td>FTBP</td>
<td>UNUSED</td>
<td>00020</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>IAGF</td>
<td>UNUSED</td>
<td>00010</td>
<td>V DUMMY</td>
<td></td>
<td>IBOT</td>
<td>UNUSED</td>
<td>00016</td>
<td>V DUMMY</td>
<td></td>
<td>IDATA</td>
<td>UNUSED</td>
<td>00001</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>IDEG2</td>
<td>I SCALR</td>
<td>00004</td>
<td>V DUMMY</td>
<td></td>
<td>ILEFT</td>
<td>UNUSED</td>
<td>00018</td>
<td>V DUMMY</td>
<td></td>
<td>INIT</td>
<td>UNUSED</td>
<td>00029</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>IRIGT</td>
<td>UNUSED</td>
<td>0001A</td>
<td>V DUMMY</td>
<td></td>
<td>ITAPE</td>
<td>UNUSED</td>
<td>00019</td>
<td>V DUMMY</td>
<td></td>
<td>ITBP</td>
<td>UNUSED</td>
<td>00014</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>KEG2</td>
<td>UNUSED</td>
<td>00008</td>
<td>V DUMMY</td>
<td></td>
<td>KGMP</td>
<td>UNUSED</td>
<td>00019</td>
<td>V DUMMY</td>
<td></td>
<td>LANG</td>
<td>UNUSED</td>
<td>00001</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>NDEG</td>
<td>UNUSED</td>
<td>00023</td>
<td>V DUMMY</td>
<td></td>
<td>NMP</td>
<td>UNUSED</td>
<td>00026</td>
<td>V DUMMY</td>
<td></td>
<td>RADEG</td>
<td>R SCALR</td>
<td>00000</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>BGT</td>
<td>R SCALR</td>
<td>00015</td>
<td>V DUMMY</td>
<td></td>
<td>RDEG2</td>
<td>R SCALR</td>
<td>00012</td>
<td>V DUMMY</td>
<td></td>
<td>RLAT</td>
<td>UNUSED</td>
<td>00008</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>RLEFT</td>
<td>R SCALR</td>
<td>00017</td>
<td>V DUMMY</td>
<td></td>
<td>RLONG</td>
<td>UNUSED</td>
<td>00009</td>
<td>V DUMMY</td>
<td></td>
<td>RRIOT</td>
<td>R SCALR</td>
<td>00019</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>RTBP</td>
<td>R SCALR</td>
<td>00013</td>
<td>V DUMMY</td>
<td></td>
<td>RNC</td>
<td>UNUSED</td>
<td>00010</td>
<td>V DUMMY</td>
<td></td>
<td>SLAT</td>
<td>UNUSED</td>
<td>00024</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>SLOK</td>
<td>UNUSED</td>
<td>0001C</td>
<td>V DUMMY</td>
<td></td>
<td>SLONG</td>
<td>R SCALR</td>
<td>00025</td>
<td>V DUMMY</td>
<td></td>
<td>SLTK</td>
<td>UNUSED</td>
<td>00018</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>SMP</td>
<td>R SCALR</td>
<td>0001E</td>
<td>V DUMMY</td>
<td></td>
<td>SPROG</td>
<td>UNUSED</td>
<td>0001F</td>
<td>V DUMMY</td>
<td></td>
<td>VETBY</td>
<td>R SCALR</td>
<td>00000</td>
<td>V DUMMY</td>
<td></td>
</tr>
<tr>
<td>XBLD</td>
<td>UNUSED</td>
<td>0002A</td>
<td>V DUMMY</td>
<td></td>
<td>XX</td>
<td>UNUSED</td>
<td>00027</td>
<td>V DUMMY</td>
<td></td>
<td>YBLD</td>
<td>UNUSED</td>
<td>00028</td>
<td>V DUMMY</td>
<td></td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (1 WORD):

COCOC VETBY

BLANK COMMON (0 WORDS)

ENTRY POINTS:

COCOC VETBY

EXTERNAL SUBPROGRAMS REQUIRED:

ARLIM  9SETUP

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

DEC | HEX
----|----
64  | 00040
1   | 00001
44  | 00020
110 | 0006E

GENERATED CODE: 64

CONSTANTS: 1

LOCAL VARIABLES: 1

TEMP: 44

TOTAL PROGRAM: 110
SUBROUTINE WEIG2(XPOL, ZPOL, NVERT, X, NPTS, SL, RHO, TEST, DSU)

THIS IS VERSION 2 WHICH ALSO DOES WEIGHTTEST

THIS SUBROUTINE IS TO BE USED WITH TALPLOT. IT COMPUTES
THE DENSITY CONTRIBUTION OF A POLYGON OF DENSITY RHO
AND ADDS THE CONTRIBUTION TO THE SL.

XPOL, ZPOL ARE THE COORDINATES OF THE VERTICES OF THE POLYGON
NVERT IS THE # OF VERTICES IN A POLYGON
X IS THE COORDINATE AT WHICH WE WISH THE SUM CALCULATED.
NPTS IS THE # OF POINTS AT WHICH WE WISH THE SUM CALCULATED
SL IS THE ACCUMULATED DENSITY CONTRIBUTION
A RESTRICTION IS THAT THE FIRST THREE (3) POINTS OF A POLYNOMIAL
MAY NOT HAVE THE SAME X COORDINATE. THE FIRST TWO (2) MAY
BE THE SAME, AND AFTER THE FIRST VERTEX ANY NUMBER MAY
THE DIMENSION OF XPOL, ZPOL, MUST BE 3 GREATER IN THE MAIN
PROGRAM THAN THE ACTUAL N# OF VERTICES (NVERT)
DIMENSION XPOL(1), ZPOL(1), X(1), Z(1), MT(1), SUM(1), NFLAG(1),
ICIS(1C), SORT(1C), KFLAG(1C), TEST(1), DSl(1)

12
ZPOL(NVERT+1) = ZPOL(2)
ZPOL(NVERT+2) = ZPOL(3)
ZPOL(NVERT+3) = ZPOL(4)
XPOL(NVERT+1) = XPOL(2)
XPOL(NVERT+2) = XPOL(3)
XPOL(NVERT+3) = XPOL(4)
C0 3CC I=1, NPTS
SL = 0.0
INTER1
I10LT=108
NDLM=NVERT+2
UW=3
XX=X(1)
C0 9 IC=1, IC
DIS(1G)=0
KFLAG(1G)=1
IF (XX=XPOL(3T)) 11, 15, 80
15
UW=2
KDL=NVERT
IF (XX=XPOL(2)) 11, 15, 80
14
UW=1
NDLM=NVERT
13
IF (XX=XPOL(1)) 11, 17, 80
17
CONTINUE
18
WRITE(I10LT,18)
FORMAT('*********** FIRST 3 VERTICES HAVE *EG* X COORD. ! )
10
CONTINUE
11
CONTINUE
12
UW=UW+1
14
IF (UW.GT.NDLM ) GO TO 1CC
13
IF (XX=XPOL(JJ)) 11, 26, 21
13
AC=UW+1
20
IF (XX.NE.XPOL(JJ+1)) GO TO 24
24
GO TO 22
21
DIS(INTER)=(XPOL(JJ).XX.ZPOL(JJ+1)+(XX-XPOL(JJ+1)) ZPOL(JJ))
C((XPOL(JJ)-XPOL(JJ+1)))
INTER=INTER+1
26
GO TO 80
C THIS SECTION HANDLES INTERSECTION WITH A VERTICAL
C LINE OR INTERSECTION THRU ONE OF THE VERTICES OF THE POLYGON
24
IF (XPOL(JJ+1).GT.XX) GO TO 26
IF (WW*GE*NLUM ) G0 T0 100
61*   DIS(INTER)*=(2PB(L(WW)+ZPB(L(WAC))/2.
62*   INTER=INTER+1
63*   G0 TO 60
64*   64 IF (WAC*EG*WW ) G0 TO 11
65*   DIS(INTER)*=ZPB(L(WAC)
66*   NFLAG(INTER)=INTER
67*   INTER=INTER+1
68*   DIS(INTER)*=ZPB(L(WW)
69*   NFLAG(INTER)=INTER+1
70*   INTER=INTER+1
71*   G0 TO 11
72*   80 CONTINUE
73*   WW*WW+1
74*   IF (WW*GT*SNLUM ) G0 TO 100
75*   IF (XPBL(WW)=$XX ) 8C/9C+91
76*   WAC=SAC
77*   52 IF (XX*NE*XPBL(WW+1) ) G0 TO 54
78*   WW+1
79*   G0 TO 80
80*   51 DIS(INTER)*=((XX*XPBL(WW)+ZPB(L(WW-1)*(XPBL(WW-1)-XX)*ZPB(L(WW))
81*   1/(XPBL(WW-1)*XPBL(WW))
82*   INTER=INTER+1
83*   G0 TO 11
84*   54 IF ( XPBL(WW+1)*LT*XX ) G0 TO 56
85*   DIS(INTER)*=(ZPB(L(WW)+ZPB(L(WAC))/2.
86*   INTER=INTER+1
87*   G0 TO 11
88*   96 IF (WAC*EG*WW ) G0 TO 80
89*   DIS(INTER)*=ZPB(L(WAC)
90*   NFLAG(INTER)=INTER
91*   INTER=INTER+1
92*   DIS(INTER)*=ZPB(L(WW)
93*   NFLAG(INTER)=INTER+1
94*   INTER=INTER+1
95*   G0 TO 80
96*   100 CONTINUE
97*   C WRITE (108,517)
98*   517 FORMAT(DIS (NFLAG))
99*   C
100*   C
101*   C WE HAVE NOW LACZTED ALL THE INTERSECTIONS WHICH RUN DOWN THE
102*   BODY OF A POLYGON AND NEVER CROSSES IN OR OUT
103*   C THE INTERSECTION WILL NOW BE SORTED FROM SMALLEST TO LARGEST
104*   C
105*   C
106*   C THIS CHANGES INTER SO THAT IT NOW = THE # OF INTERSECTIONS
107*   C
108*   C IF THERE ARE NO INTERSECTINS WE BYPASS THE COMPUTATION
109*   C
110*   C
111*   C
112*   C
113*   C
114*   KFLAG(IL)=NFLAG(1)
115*   SORT(IL)=DIS(1)
116*   G0 110 WW=2*INTER
117*   IF(SORT(IL)+LE*DIS(WW) ) G0 TO 110
118*   SORT(IL)+DIS(WW)
119*   KFLAG(1U)=KFLAG(WW)
120*   WW=WU
120 CONTINUE
121 DIS(LJL)=1.*E0
122 CONTINUE
123 SBT=0.*
124 IF(SORT(1)) 2201,2202,2203
125 2201 CONTINUE
126 CSBT=0.*
127 IF(SORT(2).LT.0) DSBR=SORT(2)
128 SBT=SORT(1)-DSBR
129 2202 CONTINUE
130 C WRITE (108,52) INTER
131 C WRITE (108,518),(SORT(I),KFLAG(I),I=1,4)
132 52 FORMAT (1X,I3)
133 518 FORMAT (1X,F6.3,I3)
134 C THE NO ARE ALL SORTED NOW
135 C WE ARE NOW GOING TO COMPLETE THE SI DISTANCE
136 C
137 MDIC=C
138 201 IF(INTER-MDIC) 999,999,820
139 820 MDIC=MDIC+1
140 IF(KFLAG(MDIC)) 203,203,221
141 203 SL =SU +(SORT(MDIC+1)-SORT(MDIC))
142 IF (KFLAG(MDIC+1)) 204,204,245
143 204 MDIC=MDIC+1
144 205 GO TO 201
145 C THIS HAS NOW HANDLED THE NORMAN SECTION
146 221 IF (KFLAG(MDIC)*AE*KFLAG(MDIC+1)) GO TO 224
147 SL =SU +(SORT(MDIC+1)-SORT(MDIC))/2
148 MDIC=MDIC+1
149 GO TO 201
150 224 SL =SU +(SORT(MDIC+3)+SORT(MDIC+2)+SORT(MDIC+1)-SORT(MDIC))
151 C/2
152 MDIC=MDIC+3
153 GO TO 201
154 245 IF (KFLAG(MDIC+1)*AE*KFLAG(MDIC+2)) GO TO 248
155 SL =SU +(SORT(MDIC+2)-SORT(MDIC+1))/2
156 MDIC=MDIC+2
157 GO TO 203
158 248 SL =SU +(SORT(MDIC+4)+SORT(MDIC+3)+SORT(MDIC+1)+SORT(MDIC+3))
159 B/2
160 MDIC=MDIC+4
161 GO TO 203
162 999 SUM(I)=SUM(I)+RH0*SU "*100.*
163 TEST(I)*TEST(I)+(SU+SUBT)*267.*
164 CSL(I)=SU*RH0*100.*
165 300 CONTINUE
166 RETURN
167 END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC</th>
<th>HEX</th>
<th>LOC</th>
<th>WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>DEC</th>
<th>HEX</th>
<th>LOC</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIS</td>
<td>R</td>
<td>ARRAY</td>
<td>0000 V</td>
<td>10</td>
<td>0008</td>
<td>DSOR</td>
<td>R</td>
<td>SCALR</td>
<td>0008 V</td>
<td>1</td>
<td>0009</td>
<td>DSOR</td>
<td>R</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>SCALR</td>
<td>0028 V</td>
<td>1</td>
<td>0038</td>
<td>IOUT</td>
<td>I</td>
<td>SCALR</td>
<td>0028 E</td>
<td>1</td>
<td>0039</td>
<td>IOUT</td>
<td>I</td>
</tr>
<tr>
<td>IG</td>
<td>I</td>
<td>SCALR</td>
<td>0032 V</td>
<td>1</td>
<td>0042</td>
<td>IU</td>
<td>I</td>
<td>SCALR</td>
<td>0032 A</td>
<td>1</td>
<td>0043</td>
<td>IU</td>
<td>I</td>
</tr>
<tr>
<td>JU</td>
<td>I</td>
<td>SCALR</td>
<td>0036 V</td>
<td>1</td>
<td>0046</td>
<td>JU</td>
<td>I</td>
<td>SCALR</td>
<td>0036 W</td>
<td>1</td>
<td>0047</td>
<td>JU</td>
<td>I</td>
</tr>
<tr>
<td>KFLAG</td>
<td>I</td>
<td>ARRAY</td>
<td>0040 V</td>
<td>10</td>
<td>0050</td>
<td>KFLAG</td>
<td>I</td>
<td>SCALR</td>
<td>0040 A</td>
<td>1</td>
<td>0051</td>
<td>KFLAG</td>
<td>I</td>
</tr>
<tr>
<td>NRTS</td>
<td>R</td>
<td>SCALR</td>
<td>0042 V</td>
<td>10</td>
<td>0052</td>
<td>NRTS</td>
<td>R</td>
<td>SCALR</td>
<td>0042 W</td>
<td>1</td>
<td>0053</td>
<td>NRTS</td>
<td>R</td>
</tr>
<tr>
<td>SUBT</td>
<td>R</td>
<td>SCALR</td>
<td>0046 V</td>
<td>1</td>
<td>0056</td>
<td>SUBT</td>
<td>R</td>
<td>ARRAY</td>
<td>0046 V</td>
<td>1</td>
<td>0057</td>
<td>SUBT</td>
<td>R</td>
</tr>
<tr>
<td>X</td>
<td>R</td>
<td>ARRAY</td>
<td>0060 V</td>
<td>1</td>
<td>0070</td>
<td>X</td>
<td>R</td>
<td>ARRAY</td>
<td>0060 V</td>
<td>1</td>
<td>0071</td>
<td>X</td>
<td>R</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (56 WORDS):**

- 000000 HEIG2
- 000001 Z
- 000002 KFLAG
- 000003 JJ
- 000004 SUBT

**ENTRY POINTS:**

- 000000 HEIG2

**EXTERNAL SUBPROGRAMS REQUIRED:**

- F102
- F104
- F106
- F108
- 9BCDWRIT
- SENDIOL
- 9SETURN

**HIGHEST ERROR SEVERITY:**

- DEC | HEX | WORDS | WORDS

- GENERATED CODE: 409
- CONSTANTS: 1
- LOCAL VARIABLES: 92
- TEMPS: 12

**TOTAL PROGRAM:** 433
SUBROUTINE YBLKI  
BLOKKED TAPE INPLT SEIS FORMAT  
MODIFIED BY REMOVING RECORD SKIP AFTER BAD READ (FOR  
NEW MONITOR)  
MODIFIED TO YBLKI FROM GBLKI MARCH 25, 1971 BY FOELINGSEE  
FOR C BOWIN  
NOV 1970  

YBLKI HAS TWO ENTRY POINTS: YSETI, AND YBLKI  
YSETI IS THE INITIALIZING ENTRY; YBLKI IS THE NORMAL ENTRY  
SUBROUTINE READS BLOCKED RECORDS FROM A MAG TAPE,  
AND RETURNS DATA FROM ONE LOGICAL RECORD,  
CONVERTED ACCORDING TO A SPECIFIED FORMAT  

YBLKI(  
1 IS1, IS2, ICA, KMO, KPR, KHR, ISEC, ILAT, ILEN, KWE, IHEP, IMAG, IMB,  
2 ISOS, ITS, IAS, ITEL, IVOLC, ION, IHG, IMS, IASP, IZH,  
3 ICE, ICG, IAUTH, IGHY, NPF, 1G, ILS,  
4  )  

ALL YBLKI ARGUMENTS ARE VARIABLE NAMES FOR DATA  
TO BE READ FROM 1 LOGICAL RECORD  
GO TO 100  

ENTRY YSETI(ITAPE, IFMT, INDI, IBLF, IRLEN, IBUF)  
DIMENSION IBUF(1)  
THIS IS THE INITIALIZING ENTRY  
ITAPE IS LOGICAL UNIT NUMBER FOR INPLT  
IFMT IS STATEMENT NO. OF FORMAT  
INDI IS INDICATOR OF INPLT STATUS  
1=READ OKAY  
2=END OF FILE  
3=PARITY ERROR  
4=FORMAT ERROR  
5= BOTH 4 AND 5 TYPE ERRORS FOUND  

IBLF IS BLOCKING FACTOR (NO. OF LOGICAL RECORDS PER BLOCK)  
IRLEN IS LOGICAL RECORD LENGTH (MUST BE MULTIPLE OF 4)  
IBUF IS INPUT BUFFER  
IF IBLF AND IRLEN ARE CHANGED, THE SIZE OF ARRAY IBUF  
MUST ALSO BE CHANGED TO IBLF=IRLEN/4  
INBD=IRLEN/4  
IBLSZ=IBF*INBD  
ICTWO  
RETURN  

ENC OF INITIALIZING PART OF SUBROUTINE  
SET UP RUN-TIME ABORT FOR FORMAT ERRORS  
CALL ABORTSET(226826)  
IF INDIC=4 THEN INDIC=4  
IF INDIC=5 THEN INDIC=1  
TEST WHETHER BLOCK IS TO BE READ FROM TAPE  
IF (ICH+NE=0) GO TO 125
489

60* C YES READ PHYSICAL RECORD
61* C CALL BUFFER(ITAPE,0,IBLUF,IBLSZ,IND,NW)
62* 110 G0 TO (110,120,260,210), IND
63* 120 INDIC=1
64* 125 IF(NW*E*:IBLSZ) ITRSTRM/N/IRLEN
65* 501 WRITE(108,501), NW
66* 510 FORMAT('YBLKIII ABNORMAL REC LENGTH; NW=',NW)
67* 520 CONTINUE
68* 526 CONTINUE
69* 530 WRITE('YBLKII ABNORMAL REC LENGTH; NW=',NW)
70* 540 CONTINUE
71* C DECODE LOGICAL RECORD ACCORDING TO FORMAT STATEMENT
72* C DECODE(IRLEN,IFMT,IBUF(4))
73* 1 ISR1,ISR2,KDA,KKB,KYF,KHF,ISEC1,LAT,KSN,ILN,KWE,IDEPT,IBM,IMB,
74* 2 ISDS,INT,DIAS,ITSU,ISEICH,IVBLC,INBT,IMG,FEG,IMG,IASP,IZH,
75* 3 ICE,IMG,IALTH,IGHY,APD,IBL,ILM,IS1,IS2
76* C KEEP TRACK OF NO. OF LOGICAL RECORDS
77* 130 CONTINUE
78* 136 CONTINUE
79* ICAT=ICAT+1
80* IF(INDC+5*ITSTR) ICAT=C
81* CALL ABORTSET(0)
82* RETURN
83* C CONTROL TRANSFERS HERE FOR EOF
84* 200 INDIC=2
85* G0 TO 130
86* C CONTROL COMES HERE FOR READ ERROR
87* 210 INDIC=4
88* OUTPUT 'YBLKII READ ERROR'
89* C RESET NW (COMPENSATING FOR BUFFIN ERROR)
90* NW=IBLSZ
91* C GO TO 126
92* C CONTROL COMES HERE FOR FORMAT ERROR
93* 220 CONTINUE
94* IF(INDIC+4*G0 TO 130
95* INDIC=5
96* G0 TO 130
97* END
HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th></th>
<th>CEC WORDS</th>
<th>HEX WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERATED CODE</td>
<td>204</td>
<td>CO0CC</td>
</tr>
<tr>
<td>CONSTANTS</td>
<td>3</td>
<td>CO0C3</td>
</tr>
<tr>
<td>LOCAL VARIABLES</td>
<td>9</td>
<td>CO009</td>
</tr>
<tr>
<td>TEMPS</td>
<td>43</td>
<td>CO028</td>
</tr>
<tr>
<td>TOTAL PROGRAM</td>
<td>259</td>
<td>CO103</td>
</tr>
</tbody>
</table>
SUBROUTINE YBLK8 - BLOCKED TAPE OUTPUT
VERSION OF APRIL 11, 72 TO OUTPUT LAT AND LONG KEY AT END

OF RECORD
MOD OF AUG 16, 71 TO OUTPUT SHORT RECORD WHEN YEND CALLED
MODIFIED FROM GBLK8 ON MAR 25, 71 BY ALEXIS BEE TO OUTPUT SEIS DATA
PROGRAMMER J. WEBSTER
FOR C BSWIN
NOV 1970

SUBROUTINE HAS 3 ENTRY POINTS: YSET6, YBLK8, YEND
YSET6 IS THE INITIALIZING ENTRY
YBLK8 IS THE NORMAL ENTRY
YEND IS THE TERMINATING ENTRY

SUBROUTINE YBLK8:
1. ISR1, ISR2, KDA, KPB, KYR, KHY, ISEC, ILAT, KSN, ILYN, KKE, IDEPT, IAMAG, IMB,
2. IS68, INT8, IDIAS, ITSU, ISELICH, IVBLC, IN8NT, IKG, IFEG, IMS, IASP, IZH,
3. ICE, IPG, IALT, ICHY, NPP, I8G, ILM, IS1, IS2

ALL YBLK8 ARGUMENTS ARE VARIABLE NAMES FOR DATA
TO BE WRITTEN IN 1 LOGICAL RECORD
DATA ...1N/1N/
DATA ...E/E/1E/
GO TO 300

ENTRY YSET6(VTAPEx, JFMT, ACPS, JFULL, JBLF, JRE, MAXBL, JBUF)

THIS IS THE INITIALIZING ENTRY
VTAPE IS THE LOGICAL UNIT NUMBER FOR OUTPUT
JFMT IS STATEMENT NUMBER OF FORMAT
ACPS IS THE NO. OF DATA POINTS CURRENTLY WRITTEN ON A TAPE
JFULL IS STATEMENT NO. TO WHICH CONTROL IS TRANSFERRED
WHEN OUTPUT TAPE IS FULL

JBLF IS BLOCKING FACTOR (NO. OF LOGICAL RECORDS PER BLOCK)
JRE IS LOGICAL RECORD LENGTH (MUST BE MULTIPLE OF 4)
JBLF IS OUTPUT BUFFER
IF JBLF AND JRE ARE CHANGED, THE SIZE OF ARRAY JBUF
MUST BE CHANGED TO JBLF*JRE/4

I8BLT+108
MAXBL IS MAXIMUM NO. OF BLOCKS PER TAPE
JBLK COUNTS BLOCKS
JBLK0

ZERO OUT DATA POINT COUNTER
ACPS0

COUNTS LOGICAL RECORDS WITHIN A BLOCK
CMT0
CWORD=JRE/4
JBLSZ=JBLF*CWORD
RETURN

END OF INITIALIZATION

DIMENSION JBUF(1)
60 C THIS IS THE TERMINATING ENTRY POINT
61 ENTRY YEND
62 C
63 IF (CNT=EG+C) GO TO 400
64 C SETTING UP TO BLT PUT SHORT RECORD
65 LBSZ=ARJ=CNT
66 GO TO 310
67 C
68 C CONVOR T LACICAL RECORD TO BCD
69 300 I=CN T=WORD+1
70 TKEY=ILAT+1000
71 GKEY=ILBN+1000
72 IF (KEE=NE=JAN) TKEY=TKEY
73 IF (KEE=NE=JEE) GKEY=GKEY
74 LTKEY=TKEY+90
75 LGKEY=GKEY=180
76 ENCODE (LREK, FMT, BUF(I))
77 1 ISR1, ISR2, KCA, KMB, KYP, KHP, ISEC, ILAT, KSN, ILBN, KWE, IDEPT, IAMAG, IMB,
78 2 IRS1, IRS2, IDA, IDB, ISE, IVEC, IMIN, IMAX, ITH, IFEG, IMS, IASP, IZH,
79 3 ICE, IMG, IAUTH, IGHY, KPP, ILG, ILN, IS1, IS2, LTKEY, LGKEY
80 C
81 C TEST IF READY TO WRITE BLOCK TO TAPE
82 IF (CNT=LT=JBLFC) GO TO 400
83 C
84 C WRITE BLOCK ONTO TAPE
85 CALL BLFFER BLT (JTAPE,JBUFF, JBLSZ, IND)
86 LBSZ=BLFC=WORD
87 GO TO (320+35C+330+330, IND
88 C WRITE (1100, 340, JBLK)
89 C FORMAT ("ERROR IN WRITING TAPE AFTER", I6, 1 BLOCKS")
90 C CALL EXIT
91 C
92 C INCREMENT AND RESET COUNTERS
93 CONTINUE
94 NDPS=NDPS+CNT
95 CNT=C
96 JBLK=JBLK+1
97 C
98 C TEST IF TAPE IS FULL
99 IF (JBLK=LE=MAXBL) GO TO 400
100 JBLK=0
101 NDPS=0
102 C WRITE (1100, 36C)
103 C FORMAT ("MAXBL OUTPUT")
104 C RETURN JFULL
105 C RETURN
106 C
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUFFERL</td>
<td>SPRG</td>
<td>EXTERN</td>
<td>EXIT</td>
<td>SPRG</td>
<td>EXTERN</td>
<td>GKEY</td>
<td>R SCALR</td>
<td>0000A</td>
<td>V</td>
<td>1</td>
<td>IAS</td>
<td>I SCALR</td>
<td>000026</td>
<td>V</td>
</tr>
<tr>
<td>I</td>
<td>SPRG</td>
<td>CCC0</td>
<td>Y</td>
<td>1</td>
<td>IAMAG</td>
<td>I SCALR</td>
<td>000028</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IAUTH</td>
<td>SPRG</td>
<td>CCC0</td>
<td>A</td>
<td>V</td>
<td>DUMMY</td>
<td>I SCALR</td>
<td>000029</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDIAS</td>
<td>SPRG</td>
<td>CCC0</td>
<td>E</td>
<td>V</td>
<td>DUMMY</td>
<td>I SCALR</td>
<td>000030</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ILAT</td>
<td>SPRG</td>
<td>CCC0</td>
<td>E</td>
<td>V</td>
<td>DUMMY</td>
<td>I SCALR</td>
<td>000031</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMB</td>
<td>SPRG</td>
<td>CCC0</td>
<td>C</td>
<td>V</td>
<td>DUMMY</td>
<td>I SCALR</td>
<td>000032</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISH</td>
<td>SPRG</td>
<td>CCC0</td>
<td>B</td>
<td>V</td>
<td>DUMMY</td>
<td>I SCALR</td>
<td>000033</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISEICH</td>
<td>SPRG</td>
<td>CCC0</td>
<td>C</td>
<td>V</td>
<td>DUMMY</td>
<td>I SEIC</td>
<td>000034</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISR2</td>
<td>SPRG</td>
<td>CCC0</td>
<td>F</td>
<td>V</td>
<td>DUMMY</td>
<td>I SCR2</td>
<td>000035</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITSL</td>
<td>SPRG</td>
<td>CCC0</td>
<td>F</td>
<td>V</td>
<td>DUMMY</td>
<td>I SCR1</td>
<td>000036</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J2H</td>
<td>SPRG</td>
<td>CCC0</td>
<td>G</td>
<td>V</td>
<td>DUMMY</td>
<td>J SCR1</td>
<td>000037</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JBLNZ</td>
<td>SPRG</td>
<td>CCC0</td>
<td>E</td>
<td>V</td>
<td>DUMMY</td>
<td>J BLNZ</td>
<td>000038</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VEE</td>
<td>SPRG</td>
<td>CCC0</td>
<td>V</td>
<td>1</td>
<td>UFM</td>
<td>J ARRAY</td>
<td>000039</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JNRD</td>
<td>SPRG</td>
<td>CCC0</td>
<td>V</td>
<td>1</td>
<td>UFM</td>
<td>J SCR2</td>
<td>000040</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JBLK</td>
<td>SPRG</td>
<td>CCC0</td>
<td>V</td>
<td>1</td>
<td>UFM</td>
<td>J BLK</td>
<td>000041</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KEY</td>
<td>SPRG</td>
<td>CCC0</td>
<td>V</td>
<td>1</td>
<td>UFM</td>
<td>J KYR</td>
<td>000042</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TKEY</td>
<td>SPRG</td>
<td>CCC0</td>
<td>V</td>
<td>1</td>
<td>UFM</td>
<td>J TKEY</td>
<td>000043</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YENB</td>
<td>SPRG</td>
<td>CCC0</td>
<td>V</td>
<td>1</td>
<td>UFM</td>
<td>J YENB</td>
<td>000044</td>
<td>V</td>
<td>DUMMY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NAME</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>CCC4</td>
<td>310</td>
<td>CCC5</td>
<td>320</td>
<td>CCCA9</td>
<td>330</td>
<td>CCCB2</td>
<td>340</td>
<td>CCCB7</td>
</tr>
</tbody>
</table>

**LOCAL VARIABLES (14 WORDS):**

- 000C C YBLKG
- 000C A JAN
- 000C I WER
- 000C J GH
- 000C P LKEY

**BLANK COMMON (0 WORDS)**

**ENTRY POINTS:**

- 000C C YBLKG

**EXTERNAL SUBPROGRAMS REQUIRED:**

- BUFFERL
- EXIT
- F11G2
- F11G4
- F11G6
- G7T9
- G7T8
- G8T9
- G8T8
- G8T7
- G8T6
- G8T5
- G8T4
- G8T3
- G8T2
- G8T1
- G8T0
- H7T9
- H7T8
- H7T7
- H7T6
- H7T5
- H7T4
- H7T3
- H7T2
- H7T1
- H7T0
- H6T9
- H6T8
- H6T7
- H6T6
- H6T5
- H6T4
- H6T3
- H6T2
- H6T1
- H6T0

**EXTERNAL SUBPROGRAMS REQUIRED:**

- BUFFERL
- EXIT
- F11G2
- F11G4
- F11G6
- G7T9
- G7T8
- G8T9
- G8T8
- G8T7
- G8T6
- G8T5
- G8T4
- G8T3
- G8T2
- G8T1
- G8T0
- H7T9
- H7T8
- H7T7
- H7T6
- H7T5
- H7T4
- H7T3
- H7T2
- H7T1
- H7T0
- H6T9
- H6T8
- H6T7
- H6T6
- H6T5
- H6T4
- H6T3
- H6T2
- H6T1
- H6T0
<table>
<thead>
<tr>
<th></th>
<th>DEC Words</th>
<th>HEX Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generated Code</td>
<td>223</td>
<td>CCGDF</td>
</tr>
<tr>
<td>Constants</td>
<td>5</td>
<td>CCG05</td>
</tr>
<tr>
<td>Local Variables</td>
<td>14</td>
<td>CCG0E</td>
</tr>
<tr>
<td>Temp</td>
<td>45</td>
<td>CCG2C</td>
</tr>
<tr>
<td><strong>Total Program</strong></td>
<td><strong>287</strong></td>
<td><strong>CC11F</strong></td>
</tr>
</tbody>
</table>
SUBROUTINE YINGST(ITAPE, JTAPE, KK)
1
1 ISR1, ISR2, KDA, KMB, KYR, KPY, SEC, CLAT, KSN, CLON, KHE, DEPT, IAMAG, ID,
2 ISR3, ISR4, IDIAS, ITSU, ISEICH, IVBL, INHT, IMG, IFEG, IMS, IASP, IZH,
3 ICE, IMG, IALTH, IGHY, NPP, JBLF, JEN, JLEN, IS1, IS2
4)
5 FOR INPUT AND OUTPUT OF SEISMICITY DATA USCSGS
6
7 VERSION 24 FEB 1975 TO ADD HANDLING OF PROJ4 OUTPUT
8 VERSION OF 25 AUGUST 1972, TO UPDATE DECK TO BE LIKE
9 VERSION OF 8 MARCH 1972 COMPILED BY FOLINSBEE
10
11 VERSION OF APRIL 2 71 TO CORRECT FORMAT ERRORS
12 VERSION OF MARCH 25 1971 BY A FOLINSBEE TO INCORPORATE
13 FACILITY TO READ AND WRITE BLOCKED DATA
14
15 SSW(44) = 1 FOR BLOCKED INPUT
16 SN AND WE WERE MASDE INTO INTEGER S FOR COMPATIBILITY WITH THE SI7-7
17 USAGE OF ALPHA NUMERIC
18 A FILE OPTION NOT IMPLEMENTED FOR THIS PROGRAM
19 DIMENSION IBUF(400), JBLF(200)
20 DATA IFLAG/C/
21 IF (KK) 420*400*410
22 I1N = 105
23 I17LT = 108
24 IFLA = 106
25 OUTPUT 'SLERBLTINE YINGST, VERSION OF 24 FEB 1975'
26 NZERO=0
27 KGDAB=NZERO
28 KGMAB=NZERO
29 KGAB=NZERO
30 KGHMAB=NZERO
31 JLEN=ILEN=88
32 JBLF=JBLF=10
33 NREC=0
34 MXCT=34000
35 IF (ISR1*45) NE 0) MXCT=15000
36 IF (ISR1*44) NE 0) CALL YSET(ITAPE, ISER, INDIC, IBLF, ILEN, IBUF)
37 IF (ISR1*45) NE 0) CALL YSETB(ITAPE, 85S, ACPS, 88S, JBLF, JLEN
38 #MXCT
39 #JBLF
40 GOTO 500
41 CONTINUE
42 IF (ISR1*44) NE 0) G0 TO 600
43 READ(ITAPE, 65)
44 1 ISR1, ISR2, KDA, KMB, KYR, KPY, ISEC, ILAT, KSN, CLON, KHE, DEPT, IAMAG, IMB,
45 ISR3, ISR4, IDIAS, ITSU, ISEICH, IVBL, INHT, IMG, IFEG, IMS, IASP, IZH,
46 ICE, IMG, IALTH, IGHY, NPP, I6Q, ILM, IS1, IS2
47 CALL STAT(I)
48 CONTINUE
49 IF (ISR1*45*800) KK = 8 RETURN
50 SEC=ISEC/10.
51 DLT=ILAT/1000.
52 CLON=ILON/1000.
53 DEPT=IECT.
54 IAMAG=IAMAG/100.
55 CALL EVIL(IIUT, I, IBAD, KGDAB, KGMAB, KGAB, KGHMAB)
56 IF (IBAD) 410, 53, 74
57 C END OF INPUT DATA, REQUIRED NO. OF FILES NOW PROCESSED
58 74 CONTINUE
59 577 CONTINUE
CONTINUE

IFLAG=1

ISEC*SEC+IC* ++49

ILAT*CLAT*1000 * ++49

ILAN*CLAN*1000 * ++49

IDEPTE*DEPT * ++49

IAMAG*AMAG*1000 * ++49

IF(ISL(45)<NE<45) GO TO 700

IF(ISL(26)<EC<26) UTAPE=10B

CALCULATING LATITUDE AND LONGITUDE KEY

TKEY=CLAT

GKEY=CLBN

IF(KSN<NE<A4) TKEY=TKEY

IF(KWE<NE<EF) GKEY=GKEY

LTKEY=TKEY+5C

GKEY=GKEY+18C

WRITE(TAPE,65)

1 ISRI, ISR2, ICA, KMB, KYP, ISEC, ILAT, KSN, ILAN, KWE, IDPT, IAMAG, IMB,

2 ISOS, INTS, ITIAS, ITLS, ISEICH, TVOLC, INBNT, IMG, IFEG, IMS, IASP, IZH,

3 ICE, IFG, ILAT, IGHY, NPP, 18G, ILM, IS1, IS2, LTKEY, LGKEY

4 FORMAT(2A3,312,14,13,15,16,17) A1,16, A1,213, A2, A3,71,13,12, A2, A1,13,

5 A3, A2,13, A1,13, A1, A4, 12, A1, A4, 213, 13)

6 IF(RECGE>MXCT) AREC=GE TO 88

AREC=AREC+1

RETURN

ENCFILE WTAPE

OUTFLT "REQUESTING NEW OUTFLT TAPE FOR YINBT"

CALL MVOL (UTAPE)

CONTINUE

GO TO 50C

CONTINUE

CALL YELKI

1 ISRI, ISR2, ICA, KMB, KYP, ISEC, ILAT, KSN, ILAN, KWE, IDPT, IAMAG, IMB,

2 ISOS, INTS, ITIAS, ITLS, ISEICH, TVOLC, INBNT, IMG, IFEG, IMS, IASP, IZH,

3 ICE, IFG, ILAT, IGHY, NPP, 18G, ILM, IS1, IS2

4 )

1=INDIC

CONTINUE

GO TO 411

CONTINUE

CALL YELKB

1 ISRI, ISR2, ICA, KMB, KYP, ISEC, ILAT, KSN, ILAN, KWE, IDPT, IAMAG, IMB,

2 ISOS, INTS, ITIAS, ITLS, ISEICH, TVOLC, INBNT, IMG, IFEG, IMS, IASP, IZH,

3 ICE, IFG, ILAT, IGHY, NPP, 18G, ILM, IS1, IS2

4 )

GO TO 50C

CALL YENB

ENCFILE WTAPE

GO TO 50C

END
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
<th>NAME</th>
<th>TYPE</th>
<th>CLASS</th>
<th>HEX LOC</th>
<th>DEC WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>APAG</td>
<td>R</td>
<td>CALR</td>
<td>005684</td>
<td>V DUMMY</td>
<td>DLAT</td>
<td>R</td>
<td>CALR</td>
<td>002281</td>
<td>V DUMMY</td>
<td>DLAT</td>
<td>R</td>
<td>CALR</td>
<td>002281</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>DLEN</td>
<td>I</td>
<td>CALR</td>
<td>000263</td>
<td>V DUMMY</td>
<td>EVIL</td>
<td>I</td>
<td>CALR</td>
<td>002284</td>
<td>V DUMMY</td>
<td>EVIL</td>
<td>I</td>
<td>CALR</td>
<td>002284</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>I</td>
<td>I</td>
<td>CALR</td>
<td>00026F</td>
<td></td>
<td>IAMAG</td>
<td>I</td>
<td>CALR</td>
<td>00228E</td>
<td></td>
<td>IAMAG</td>
<td>I</td>
<td>CALR</td>
<td>00228E</td>
<td></td>
</tr>
<tr>
<td>IAUTH</td>
<td>I</td>
<td>CALR</td>
<td>002396</td>
<td>V DUMMY</td>
<td>IBAD</td>
<td>I</td>
<td>CALR</td>
<td>002270</td>
<td></td>
<td>IBAD</td>
<td>I</td>
<td>CALR</td>
<td>002270</td>
<td></td>
</tr>
<tr>
<td>IBUF</td>
<td>I</td>
<td>ARRT</td>
<td>000201</td>
<td>400</td>
<td>ICE</td>
<td>I</td>
<td>CALR</td>
<td>002284</td>
<td>V DUMMY</td>
<td>ICE</td>
<td>I</td>
<td>CALR</td>
<td>002284</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>IDIAS</td>
<td>I</td>
<td>CALR</td>
<td>00238A</td>
<td>V DUMMY</td>
<td>IFEQ</td>
<td>I</td>
<td>CALR</td>
<td>002290</td>
<td>V DUMMY</td>
<td>IFEQ</td>
<td>I</td>
<td>CALR</td>
<td>002290</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>IMA</td>
<td>I</td>
<td>CALR</td>
<td>00238A</td>
<td>V DUMMY</td>
<td>IIBM</td>
<td>I</td>
<td>CALR</td>
<td>002286</td>
<td>V DUMMY</td>
<td>IIBM</td>
<td>I</td>
<td>CALR</td>
<td>002286</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>IM</td>
<td>I</td>
<td>CALR</td>
<td>00238A</td>
<td>V DUMMY</td>
<td>IMS</td>
<td>I</td>
<td>CALR</td>
<td>002291</td>
<td>V DUMMY</td>
<td>IMS</td>
<td>I</td>
<td>CALR</td>
<td>002291</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>INBNT</td>
<td>I</td>
<td>CALR</td>
<td>00238E</td>
<td>V DUMMY</td>
<td>INTS</td>
<td>I</td>
<td>CALR</td>
<td>002289</td>
<td>V DUMMY</td>
<td>INTS</td>
<td>I</td>
<td>CALR</td>
<td>002289</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>IPUN</td>
<td>I</td>
<td>CALR</td>
<td>00238C</td>
<td>V</td>
<td>IGYH</td>
<td>I</td>
<td>CALR</td>
<td>002297</td>
<td>V DUMMY</td>
<td>IGYH</td>
<td>I</td>
<td>CALR</td>
<td>002297</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>ISEC</td>
<td>I</td>
<td>CALR</td>
<td>00236A</td>
<td>V DUMMY</td>
<td>ISEC</td>
<td>I</td>
<td>CALR</td>
<td>002281</td>
<td>V DUMMY</td>
<td>ISEC</td>
<td>I</td>
<td>CALR</td>
<td>002281</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>ISRI</td>
<td>I</td>
<td>CALR</td>
<td>00232A</td>
<td>V DUMMY</td>
<td>IS2</td>
<td>I</td>
<td>CALR</td>
<td>002289</td>
<td>V DUMMY</td>
<td>IS2</td>
<td>I</td>
<td>CALR</td>
<td>002289</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>ITSL</td>
<td>I</td>
<td>CALR</td>
<td>00232B</td>
<td>V DUMMY</td>
<td>JBLFC</td>
<td>I</td>
<td>CALR</td>
<td>002264</td>
<td>V</td>
<td>JBLFC</td>
<td>I</td>
<td>CALR</td>
<td>002264</td>
<td>V</td>
</tr>
<tr>
<td>J2H</td>
<td>I</td>
<td>CALR</td>
<td>002329</td>
<td>V DUMMY</td>
<td>JHA</td>
<td>I</td>
<td>CALR</td>
<td>002273</td>
<td>V</td>
<td>JHA</td>
<td>I</td>
<td>CALR</td>
<td>002273</td>
<td>V</td>
</tr>
<tr>
<td>JCE</td>
<td>I</td>
<td>CALR</td>
<td>002274</td>
<td>V</td>
<td>KDA</td>
<td>I</td>
<td>CALR</td>
<td>002278</td>
<td>V DUMMY</td>
<td>KDA</td>
<td>I</td>
<td>CALR</td>
<td>002278</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>KHM</td>
<td>I</td>
<td>CALR</td>
<td>00232B</td>
<td>V DUMMY</td>
<td>KK</td>
<td>I</td>
<td>CALR</td>
<td>002279</td>
<td>V DUMMY</td>
<td>KK</td>
<td>I</td>
<td>CALR</td>
<td>002279</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>KEN</td>
<td>I</td>
<td>CALR</td>
<td>002323</td>
<td>V DUMMY</td>
<td>KWE</td>
<td>I</td>
<td>CALR</td>
<td>002284</td>
<td>V DUMMY</td>
<td>KWE</td>
<td>I</td>
<td>CALR</td>
<td>002284</td>
<td>V DUMMY</td>
</tr>
<tr>
<td>LGKEY</td>
<td>I</td>
<td>CALR</td>
<td>002276</td>
<td>V</td>
<td>LTRK</td>
<td>I</td>
<td>CALR</td>
<td>002275</td>
<td>V</td>
<td>LTRK</td>
<td>I</td>
<td>CALR</td>
<td>002275</td>
<td>V</td>
</tr>
<tr>
<td>MCT</td>
<td>I</td>
<td>CALR</td>
<td>002377</td>
<td>V</td>
<td>KDFS</td>
<td>I</td>
<td>CALR</td>
<td>002269</td>
<td>V</td>
<td>KDFS</td>
<td>I</td>
<td>CALR</td>
<td>002269</td>
<td>V</td>
</tr>
<tr>
<td>NAP</td>
<td>I</td>
<td>CALR</td>
<td>002367</td>
<td>V</td>
<td>NZER0</td>
<td>I</td>
<td>CALR</td>
<td>002260</td>
<td>V</td>
<td>NZER0</td>
<td>I</td>
<td>CALR</td>
<td>002260</td>
<td>V</td>
</tr>
<tr>
<td>STAT</td>
<td>I</td>
<td>SPRG</td>
<td>002327</td>
<td>EXTERN</td>
<td>TKY</td>
<td>I</td>
<td>CALR</td>
<td>002271</td>
<td>V</td>
<td>TKY</td>
<td>I</td>
<td>CALR</td>
<td>002271</td>
<td>V</td>
</tr>
<tr>
<td>YBLKG</td>
<td>I</td>
<td>SPRG</td>
<td>00232B</td>
<td>EXTERN</td>
<td>VENDO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YENDO</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>YINOT</td>
<td></td>
<td>SPRG</td>
<td>000000</td>
<td>P</td>
<td>YSETI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>YSETI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
<th>LABEL</th>
<th>HEX LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>0040A</td>
<td>65</td>
<td>01148</td>
<td>74</td>
<td>0040E</td>
<td>88</td>
<td>0022F</td>
<td>400</td>
<td>0022C</td>
<td>410</td>
<td>0022A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>411</td>
<td>0040B</td>
<td>420</td>
<td>000E1</td>
<td>500</td>
<td>0040F</td>
<td>577</td>
<td>0040C</td>
<td>600</td>
<td>0017C</td>
<td>700</td>
<td>001A4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOCAL VARIABLES (631 WORDS):

C00CC YINOT
C0025C ILEA
C00262 ILEN
C00268 INDIC
C00266 IAMAG
C00274 JEE

BLANK COMMAND (63 WORDS)

ENTRY POINTS:

C00000 YINOT
EXTERNAL SUBPROGRAMS REQUIRED:

<table>
<thead>
<tr>
<th></th>
<th>DEC</th>
<th>HEX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WORDS</td>
<td>WORDS</td>
</tr>
<tr>
<td>EVIL ISM</td>
<td>MCVAL</td>
<td>STAT</td>
</tr>
<tr>
<td>YSET6 F101</td>
<td>FILE2</td>
<td>F103</td>
</tr>
<tr>
<td>YSEToyota</td>
<td>F104</td>
<td>F105</td>
</tr>
<tr>
<td>SBCDREAD</td>
<td>SBCWRIT</td>
<td>SENCFILE</td>
</tr>
<tr>
<td>YSET6</td>
<td>YSEToyota</td>
<td>SIBDATAL</td>
</tr>
<tr>
<td>SBCRDFN</td>
<td>SBCWRIT</td>
<td>SENCFILE</td>
</tr>
</tbody>
</table>

HIGHEST ERROR SEVERITY: 0 (NO ERRORS)

<table>
<thead>
<tr>
<th>GENERATED CODE</th>
<th>LOCAL VARIABLES</th>
<th>LOCAL VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>465</td>
<td>631</td>
<td>1146</td>
</tr>
<tr>
<td>CO101</td>
<td>CO277</td>
<td>CO47A</td>
</tr>
<tr>
<td>COCOA</td>
<td>C0028</td>
<td></td>
</tr>
</tbody>
</table>

TOTAL PROGRAM: 1146
A summary and documentation of a family of computer programs that have been developed by the gravity group at the Woods Hole Oceanographic Institution is presented. The programs provide for format conversion, computation of the regional gravity field from spherical harmonic coefficients, selective data retrieval, graphic display, and construction of two- and three-dimensional structure models and the computation of the gravitational attraction of those models.
GRAVITY DATA PROCESSING PROGRAMS by Carl Bowin.  pages.
February 1977. Prepared for the Office of Naval Research under
Contract N00014-74-C-0262; NR 083-004.

A summary and documentation of a family of computer programs
that have been developed by the gravity group at the Woods Hole
Oceanographic Institution is presented. The programs provide for
format conversion, computation of the regional gravity field from
spherical harmonic coefficients, selective data retrieval, graphic
display, and construction of two- and three-dimensional structure
models and the computation of the gravitational attraction of those
models.