176 2/13
SC-CR-68-3796
December 1968

POPULATION DENSITY STUDY

Prepared by

S. G. Vandevender

R. C. Lindberg

The Dikewood Corporation

657
MASTER

**AEROSPACE
NUCLEAR
SAFETY**

SANDIA LABORATORIES

OPERATED FOR THE U. S. ATOMIC ENERGY COMMISSION BY SANDIA CORPORATION

ALBUQUERQUE, NEW MEXICO; LIVERMORE, CALIFORNIA

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Issued by Sandia Corporation,
a prime contractor to the
United States Atomic Energy Commission

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

SC-CR-68-3796

POPULATION DENSITY STUDY

December 1968

Prepared by

S. G. Vandevender

R. C. Lindberg

The Dikewood Corporation

for

Sandia Corporation

under

Contract No. 48-6156

ABSTRACT

This report details a system of computer programs used to regiment world population statistics to a well-ordered form, compress the data to allow ready availability of any portion of it, and analyze these data according to population zones. The programs are divided into IBM 7090 routines, which accomplish the regimentation; and CDC 3600 routines, which compress and utilize the data.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

SUMMARY

The product of this study is a data base from which world population densities can be analyzed by computer with relative ease and to which other types of data can be added for analysis in conjunction with population statistics for use in hazard analyses of the use of nuclear power in space. The data base is stored in two distinct formats. The most usable format is contained on one reel of magnetic tape in binary mode. The other format is contained on many reels of tape in BCD mode.

The data base is the result of a system of computer programs which reformat and edit raw population data into a unified set of world population statistics containing both rural and urban data specifically located by latitude and longitude coordinates, and ordered by latitude from the South Pole and by longitude within each latitude band from Greenwich increasing eastward.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT.	i
SUMMARY.	ii
INTRODUCTION.	1
DATA STRUCTURE.	3
DATA EDITING SEQUENCE.	8
USING THE DATA.	10
UPDATING.	11
SPECIFIC PROGRAM SEQUENCE.	11
CONVTP ✓	
Introduction.	11
Main Program.	14
The Functions.	16
Subroutine CHECK.	17
Operation.	17
Flow Charts.	19
Listing.	24
CITSPL ✓	
Purpose.	28
Input and Output.	28
Program Description.	29
The Split Algorithm.	32
Error Analysis.	35
Flow Charts.	40
Listing.	47
CELGRP ✓	
Purpose.	60
Input and Output.	60
Program Description.	61
Flow Charts.	64
Listing.	67

	<u>Page</u>
DISCRD	
Purpose	72
Input and Output	72
Program Description	72
Messages Printed	73
Flow Charts	75
Listing	79
REFORM	
Purpose	82
Input and Output	82
Program Description	83
Printed Messages	84
Flow Charts	86
Listing	90
POPRET	
Purpose	94
Input	94
Calling Sequence	95
COMMON Storage	95
Printed Messages	96
Output	97
Flow Charts	99
Listing	104
CELLANAL	
Purpose	109
Input and Output	109
Program Description	110
Flow Charts	113
Listing	116
CELLGEN	
Purpose	120
Program Description	120
Card Input Format	121
Flow Charts	123
Listing	130

	<u>Page</u>
APPENDIX A--Deck Structures	141
APPENDIX B--MAPPRT.	145
Purpose	146
Program Description	146
Program Operation	147
Maintenance	148
Flow Chart.	149
Listing	155

INTRODUCTION

This report describes a set of programs designed to edit world-population statistical data, and to utilize the edited data in studies of population densities. The work was carried out under the direction of Mr. E. S. Phinney, Organization 9514, Sandia Corporation.

World population statistics are available for countries, in cells with dimensions of 20 min latitude by 30 min longitude, and for cities within countries, in circles with radii of 4 n mi or less (for cities of 25,000 or more population). These circles are more commonly referred to as "P95" data circles. Each city is covered by a number of nonoverlapping circles such that 95 percent of the population of that city is contained within these circles. These statistics include more data than are desired for the population density studies, and they are in an inconvenient format.

The planned population density studies require a base from which population densities can be analyzed by countries, by groups of latitude bands, or by any specified band around the globe. Nonpopulation statistics will be related to population density in the edited data base. The raw data are stored on many reels of magnetic tape in BCD records blocked 1 x 84 in one file per reel. Furthermore, this information is essentially random. Preliminary data-reduction programs will edit the presently available data, removing all nonpertinent information and reformatting the remaining data for cells and cities to make the two types compatible. Final data reduction programs will sort and merge the now-compatible data into one continuous

data base whose origin is the South Pole. The data are ordered by longitude increasing eastward from the Greenwich meridian within latitudes increasing northward. Each cell record is followed immediately by all city records belonging to that cell. The final step in the data reduction sequence is to compress the edited data base so that it will occupy only one reel of magnetic tape. This final tape will contain all the desired data in a highly compressed, well-ordered format, enabling rapid analysis by computer. Presently, two programs exist to analyze the compressed data base: (1) the Population Density Analysis program, and (2) the Mercator Plot program.

A standard subroutine will retrieve data from the compressed format for use in the Population Density Analysis program. This program will determine specific population density zones within countries within a given latitude band one cell wide around the globe.

The edited data base may be used to produce a Mercator plot of missing cells. Edited data for the missing cells will be approximated by smearing population data within a given latitude band for a given country across all the cells in the band that belong to the country.

The structure of the data and the formats of both edited and compressed data are detailed to simplify the descriptions of the data reduction sequence and of the utilization procedures. The individual programs are then described to provide coherence to the flow charts and program listings in the appendix.

DATA STRUCTURE

Because of the variability of raw data formats, a discussion of those formats in which the data has been received is of no value here; however, the formats are indicated in the program descriptions. Only the completed data base is referred to in the following discussion.

In the edited data base, each entry must have a key, which locates it in its geographic cell. One cell is assigned to only one country. Each cell is located by the coordinates of its center point, which will be at one of the following intersections: X+.250 or X+.750 deg longitude with Y+.167, Y+.500, or Y+.833 deg latitude. For example (see Fig. 1), the cell with center point 0.250 deg longitude, 90.167 deg latitude has for its southwest corner the intersection of the Greenwich meridian and the equator. Since the cell center-point coordinates key the data and are used in sorting, they are represented as integer numbers; hence, each coordinate is multiplied by 1000. Latitude of 90.167 deg becomes 90167, and the latitude range is 0167 to 179833 (South Pole to North Pole). The last three digits will be 167, 500, or 833. The longitude range (360.0 deg from Greenwich east to Greenwich) is then 0250 to 359750, and the last three digits will be 250 or 750.

Each cell is identified as belonging to either the Sino-Soviet bloc (S) or the Free-World (F). The population recorded for any cell is termed "residual," as it does not include population data for cities over 25,000. The data date for each cell is the date of the census of the

Fig. 1--Cell Structure. The diagram depicts 24 cells about the Equator at Greenwich.

residual population. Each cell is assigned to a single country by means of a 10-character code. The first four characters are always zero and the remaining characters are allocated two each to world division, subdivision, and country codes.

Statistics for each city over 25,000 within a cell remain separate from the cell statistics, but contain no identifier for the city. These P95 statistics include: population of the city within the cell, area of the city within the cell, and census date. Coordinates for each city are those of the center of the cell in which the city is located. Cities that lie in more than one cell are split between the cells, with population distributed in proportion to area. The data base is designed to allow one cell to have up to 1000 cities and city-segments (henceforth called cities) associated with it.

For efficiency in retrieving any data from the data base, data are ordered by latitude from 0167 to 179833, and within each latitude band by longitude from 0250 to 359750. The data entry for each cell precedes data for cities in the cell, and is tagged with an additional sort key, the digit 0, to differentiate it from city data. (City data contains a 1 as a sort key.)

The data base is in two forms. The first is written as BCD card images on several consecutively ordered magnetic tape reels, with one file per reel. The second, derived from the first, is written in buffered binary, to allow the content of the several BCD reels to be compressed into one reel of tape, with a tape file for each latitude band.

The BCD data base contains only cell and city records, one record per cell or city. The formats of a cell record and a city record follow:

Cell Record

<u>Card Image Columns</u>	<u>Data</u>	<u>Mode</u>
1-6	latitude	I6
7-12	longitude	I6
13	F or S	A1
14	blank or 0 (indicates cell data)	A1
15	blank	1X
16-20	residual population $\times 10^2$	I5
21-30	country code	A10
31-34	date of cell data	A4
35-84	blank	50X

City Record

<u>Card Image Columns</u>	<u>Data</u>	<u>Mode</u>
1-6	latitude	I6
7-12	longitude	I6
13	blank	1X
14	1 (indicates city data)	A1
15-35	blank	21X
36-40	population $\times 10^3$ of city	I5
41-44	area $\times 10^2$ of city	I4
45-48	date of city data	A4
49-84	blank	36X

The first 14 columns of any record are sort columns. Latitude and Longitude order the data, the F or S allow sorting between Sino-Soviet and Free-World data and the 0 or 1 facilitates positioning cities behind cells.

The buffered binary data are contained in 540 files on one tape. The record order is: (1) file label, (2) cell record, and (3) city record. The file label always occurs. It contains three words--latitude (a six-digit integer), length of cell record plus label record, and length of file, where length implies number of computer words. * If label-word two is greater than three, the band is nonempty. The number of cells is label-word two divided by five (integer division). If label-word two and the file length are the same, there are no cities in the current band. Otherwise the number of cities is the difference between label-word three and label-word two, divided by three. Since all cells for a band are given in longitude order in one record, the buffered binary cell data format of only one cell's data is given (where $1 \leq K \leq \text{total number of cells}$).

Record Word	Data and Type
K*5-4	longitude (integer)
K*5-3	cell population (integer)
K*5-2	first 8 characters of country code (alphanumeric)
K*5-1	(packed alphameric) <ul style="list-style-type: none"> a. last two characters of country code b. F or S c. blank character d. 4 character date

* The original buffered binary data base was written for use on a CDC 3600 computer. Word size for this machine is 48-bits, or 8-BCD (6-bit) characters.

<u>Record Word</u>	<u>Data and Type</u>
K*5	(packed integer)
	a. location in city record of this cell's first city word (left half of word)
	b. number of city words for this cell (right half of word)

Likewise, only one buffered binary city data format is given, since the cities are also given in longitude order in one record (where $1 \leq K \leq \text{total number of cities}$).

<u>Record Word</u>	<u>Data and Type</u>
K*3-2	city population $\times 10^3$ (integer)
K*3-1	city area $\times 10^2$ (integer)
K*3	date (alphanumeric, left half of word, 4 characters)

DATA EDITING SEQUENCE

Coordinates of cell data are usually given in degrees, minutes, seconds, with a directional indicator, so that a routine specific to the data must be developed to convert the coordinates to the convention set forth above. The cell coordinates from the raw data are expected to be the centers of cells, but are not required to be so for conversion. The nonrestriction allows the same routines, if not the same computer codes, to convert both city and cell coordinates, since city coordinates are only coincidentally centers of cells.

Reformatting of data merely requires reordering the pertinent data and omitting nonpertinent data. The only change made in any data at this stage should be the splitting of those cities contained in more than one cell.

The raw data form of a city is a circle (called a P95 circle) of given radius and population. To split or distribute a city among two or more cells, it is necessary to determine the area of the portion of the circle lying in each cell. Population is allocated to each cell in direct proportion to area.

Once the data have been edited and each card image is in the acceptable form, the image records must be sorted by latitude, longitude, cell, and city, and merged into one continuous data base. This sorting is usually carried out piecemeal due to limitations of computer memories, and each resulting sorted data set is merged into the whole.

After the data are sorted and merged, possibilities arise of duplicated cells and of cities for which no cell exists. Duplication of cells can take several forms. When duplications occur, as evidenced by identical coordinates, the cells are merged. If two (or more) records differ only in country code (whether due to different country codes where a cell boundary crosses a country boundary, or to the same code recorded in two differing orders), the populations are added and the remainder of the record with the larger population is preserved along with the new population. If no cell is found for a city, a cell with that city's coordinates is generated, with no population or date, and inserted ahead of the city record.

After the BCD data set is accepted as correct, it is compressed into the binary tape format for ease of handling.

USING THE DATA

The user may examine the BCD data set for missing data, termed discontinuities, by using the DIScontinuity RecorDer (DISCRD) program to produce a plot tape for use as input to a Mercator Plot program. The resulting plot shows bounds of discontinuity areas over a world map. Once the world data base is complete, the points plotted should sketch the coastlines of the world to indicate the only true discontinuities--the waters of the earth.

The CELL GEnerator (CELLGEN) program makes it possible to generate cells for which no data exist or for which new data have become available. The program smears the given population over the cells generated by distributing it evenly to the cells. This program uses and produces BCD tapes.

The POPulation REtriever (POPRET) subroutine allows the user to recover data from the binary tape for a particular cell or for a particular latitude band. The CELL ANALysis (CELLANAL) program, using the binary tape, summarizes areas and population densities of the cells in each band.

Each of these routines is described in detail in later subsections of this report.

UPDATING

The variable nature of the raw data made it extremely difficult to set up a simple updating procedure. Updating requires reiterating the entire data reduction sequence.

SPECIFIC PROGRAM SEQUENCE

The complete set of programs and system logical flow is given in Fig. 2. Each of the programs is described in detail in succeeding subsections of this report. The complete BCD editing sequence, containing a list of those BCD tapes used in the initial preparation of the final compressed data base, is shown in Fig. 3.

The specific programs are divided into IBM 7090 routines and CDC 3600 routines. For the 7090, the list includes CONVerTer Program (CONVTP), a coordinate converter program; CITy SPLitter (CITSPL) routine; CEl Generator Remover Program (CELGRP), a routine used to generate and remove cells; and DISCRD, a discontinuity recorder program. The 3600 list includes CELGEN, a cell generator routine (used in smearing population over a band of cells); binary reformat (REFORM), the program used to produce the compressed binary data base; CELLANAL, a cell analysis routine; and MAP on PRinTer (MAPPRT), a program used to produce printer plots of the cells contained on the binary tape.

CONVTP

Introduction

CONVTP is an IBM 7090 Fortran program that reformats BCD records. These records are input from a single input tape, and are output to

Fig. 2--System logical flow

Fig. 3--BCD editing sequence.

a single output tape. The program detects parity or other mechanical errors, and ignores the record in which those errors occur. Exit is accomplished by detecting an end-of-file mark. A final printout of the total number of records reformatted is made, and control is returned to the monitor.

Main Program

CONVTP accepts as input a tape blocked 1 x 84, in one file. Each record of 84 characters contains the following information in the designated positions:

<u>Position</u>	<u>Contents</u>	<u>Format</u>
1-2	latitude, degrees	I2
3-4	latitude, minutes	I2
5-6	latitude, seconds	I2
7	N or S	A1
8-10	longitude, degrees	I3
11-12	longitude, minutes	I2
13-14	longitude, seconds	I2
15	E or W	A1
16-20	population (x 100)	I5
21-30	country code	5A2
31-34	date	A4
35-84	blank	50X

N or S in position 7 determines whether the given latitude is north or south of the equator. E or W in position 15 determines east or west longitude from Greenwich.

The latitude and longitude are converted to new values in accordance with the previously described scheme, i. e., latitudes from 0.167 deg to 179.833 deg, south to north, and longitudes from 0.250 deg to 359.750 deg, east to west. This conversion is accomplished through use of the functions CENLON and CENLAT, and a subroutine called CHECK, described later in this subsection.

A new output tape is then written, blocked 1 x 84 in one file, and containing the following information in the designated positions:

<u>Position</u>	<u>Contents</u>	<u>Format</u>
1-6	latitude, in degrees x 1000	I5, I1
7-12	longitude, in degrees x 1000	I5, I1
13	F or S } 0	A2
14		
15	blank	1X
16-20	population	I5
21-30	country code	5A2
31-34	date	A4
35-84	blank	50X

The F or S in position 13 is an input variable, and distinguishes whether the cell is a Free World cell or a Sino-Soviet cell. The zero in position 14 is a sort key and also an input variable.

The IBM 7090 considers integers to be restricted to the range $|I| \leq 2^{17} - 1 = 131071$. Since the output longitude will range from 250 to 359750 and the output latitude will range from 167 to 179833 when expressed as integers, a subterfuge must be used to circumvent this

difficulty. The least significant digit of the integer is always isolated and written as I1 following the more significant digits, which are written as I5. The same subterfuge is employed in CITSP, the next program discussed.

The input tape is read in groups of 100 cell images for speed of processing. Processing of these 100 images then takes place as described. When the end-of-file is encountered, a flag (IEOFFG) is set and the last set of records is then processed. A final printout of the number of records that were reformatted is then made, and the program exits.

The Functions

As has been stated, the format of the cell records is such that the latitude and longitude are given in degrees, minutes, and seconds. The functions CENLAT and CENLON convert these values to degrees and thousandths of a degree. The argument list is the same for each:

$$A = \text{CENLXX} (\text{IDEG}, \text{MIN}, \text{ISEC}, \text{IDIR})$$

where the first three values are the degrees, minutes, and seconds of the coordinate, and the last value is the direction indicator (N or S for latitude, E or W for longitude). Both CENLAT and CENLON return a real number in degrees and thousandths of a degree. The accompanying flowcharts and listings should be self-explanatory.

Subroutine CHECK

Because of the discrete binary nature of digital computers, the conversion of a real number to an integer is in general not unique, with roundoff occurring. Since the reformatted latitude and longitude, when multiplied by 1000, are to be exact integers, subroutine CHECK is used to accomplish this.

Briefly, the subroutine splits the incoming coordinates into two integers--the integral portion and the fractional portion. The fractional portion is then bounded and transformed to the proper value, depending upon whether it is part of the latitude and longitude.. The code is straightforward, as evidenced by the flowcharts and listings.

Operation

The deck setup is described in Appendix A. Since this program is written for the IBM 7090, the deck structure for that machine should be used. The deck structure for the CDC 3600 is also provided in Appendix A for reference.

The format of the first six cards, the last and the (last - 2) cards, is fixed by the Sandia Data Center, and should be obtained exactly from them. The data card [the (last - 1) card] contains an F or S in column 3, a zero in column 4, an N in column 7, and an E in column 10. The latter two variables are necessary to detect the differences between north and south latitude and east and west longitude

As of this writing, the standard input unit is Logical Unit (LUN) 2, so the preceding data card is read by this unit. The first user input tape is LUN 5, which corresponds on the signup sheet to physical unit A5. The input tape containing the cell images must be specified on this unit. The first user output tape is LUN 16, which corresponds to physical unit B6. This place on the signup sheet must indicate "SAVE OUTPUT." The program printout comes automatically from A3. Further details are best obtained from the Sandia Data Center-7090 Operations.

Flow Chart for Converter--CONVTP

Converter (Continued)

Converter (Continued)

Converter (Continued)

Converter (Concluded)

CCONVTP

C
C

NUCLEAR SAFETY CONVERTER PROGRAM

```

COMMON P95,IE,IN
  DIMENSION LATDEG(100),LATMIN(100),NS(100),LONDEG(100),LONMIN(100)
  $,IEW(100),IPOP(100),ICODE(100,5),IDATE(100),LADEG(100),ADEG(100)
  $,LODEG(100),ODEG(100)
  DIMENSION LQDEG(100),LRDEG(100)
  DIMENSION IBUG(80)
7  READ INPUT TAPE 2,1,ITYPE,IN,IE
1  FORMAT (2X,A2,2(2X,A1))
  K=0
  IZERO=0
  IEOFFG=0
  RFWIND 5
  RFWIND 16
2  I=1
3  IF(NBUF(5)) 4,7,13
4  CALL REREAD
  READ INPUT TAPE 5,5,(IBUG(J),J=1,80)
5  FORMAT (80A1)
  WRITE OUTPUT TAPE 3,72,(IBUG(J),J=1,80),K,I
6  FORMAT (13HUPARITY ERROR/1X,80A1/13H RECORD GROUP,15,6H CARD,I4)
  GO TO 3
7  CALL REREAD
  READ INPUT TAPE 5,8,LATDEG(I),LATMIN(I),LATSEC,NS(I),LONDEG(I),
  $LONMIN(I),LONSEC,IEW(I),IPOP(I),(ICODE(I,J),J=1,5),IDATE(I)
8  FORMAT (3I2,A1,I3,2I2,A1,I5,5A2,A4)
  I=I+1
  IF(I-101) 3,9,3
9  IMAX=I-1
  DO 10 I=I,IMAX
    ADEG(I)=CENLAT(LATDEG(I),LATMIN(I),LATSEC,NS(I))
    ODEG(I)=CENLON(LONDEG(I),LONMIN(I),LONSEC,IEW(I))
10  CONTINUE
  DO 12 I=I,IMAX
    LADEG(I)=ADEG(I)*1000.
    LQDEG(I)=ODEG(I)*100.
    CALL CHECK (LADEG(I),LODEG(I),LQDEG(I),LRDEG(I))
    IF(NS(I)-IN) 101,102,101
C
C      ADD 90 TO LATITUDE.
C
101  N=LQDEG(I)+90000
    IDIGIT=N-(N/10)*10
    N=N/10
    GO TO 103
102  IDIGIT=LQDEG(I)-((LQDEG(I)/10)*10)
    N=(LQDEG(I)/10)+9000
103  WRITE OUTPUT TAPE 16,11,N,IDIGIT,LRDEG(I),IZERO,ITYPE,IPOP(I),
  $ (ICODE(I,J),J=1,5),IDATE(I)
11  FORMAT (2(I5,11),A2,1X,I5,5A2,A4,50X)
12  CONTINUE
  K=K+1
  IF(IEOFFG-1) 2,14,14

```

```

13      IEOFFG=1
        GO TO 9
14      END FILE 16
        CALL RUN (5)
        CALL RUN (16)
        K=K-1
        WRITE OUTPUT TAPE 3,15,IMAX,K
15      FORMAT ('1H0,11HA TOTAL OF ,I3,3H + ,I3,41H00 RECORDS WERE REFORMAT
        $TED ON THIS TAPE.')
        CALL EXIT
        END

```

```

      FUNCTION CENLAT (I,J,K,L)
      COMMON P95,IE,IN
      MINSEC=J*100+K
      XI=I
      XJ=J
      XK=K
      IF(XI) 1,3,1
1      IF(MINSEC) 3,2,3
2      CENLAT=(XI-I.0)+0.8333
      GO TO 8
3      IF(4000-MINSEC) 4,6,5
4      CENLAT=XI+0.8333
      GO TO 8
5      IF(2001-MINSEC) 6,6,7
6      CENLAT=XI+C.5000
      GO TO 8
7      CENLAT=XI+0.167
8      IF(L-IN) 9,10,9
9      CENLAT=-CENLAT
10     RETURN
      END

```

```

FUNCTION CENLON (I,J,K,L)
COMMON P95,IE,IN
IF(L-IE) 1,13,1
1  II=359-I
   JJ=59-J
   KK=60-K
   IF(KK-60) 3,2,3
2  KK=0
   JJ=JJ+1
3  IF(JJ-60) 5,4,5
4  JJ=0
   II=II+1
5  IF(II-360) 7,7,6
6  II=II-360
7  MINSEC=JJ*100+KK
   XI=II
   XJ=JJ
   XK=KK
   IF(MINSEC) 10,8,10
8  CENLON=(XI-1.0)+0.7500
   IF(XI) 9,12,9
9  RETURN
10 IF(3001-MINSEC) 11,11,12

11 CENLON=XI+0.7500
   RETURN
12 CENLON=XI+0.2500
   RETURN
13 II=I
   JJ=J
   KK=K
   GO TO 5
END

```

```

SUBROUTINE CHECK (LATG, LONG, LATR, LONGR)
  IF (LATG) 1,8,2
1  LATR=-LATG
  IFLG=1
  GO TO 3
2  LATR=LATG
  IFLG=0
3  NINT=LATR/1000*1000
  NFRACT=XABSF(LATR-NINT)
  IF(NFRACT-250) 4,4,5
4  LATR=NINT+167
  GO TO 9
5  IF(NFRACT-750) 6,6,7
6  LATR=NINT+500
  GO TO 9
7  LATR=NINT+833
  GO TO 9
8  LATR=0
  GO TO 11
9  IF(IFLG) 10,11,10
10 LATR=-LATR
11 IF(LONG) 12,17,13
12 LONGR=-LONG
  IFLG=1
  GO TO 14
13 LONGR=LONG
  IFLG=0
14 NINT=LONGR/100*100
  NFRACT=XABSF(LONGR-NINT)
  IF(NFRACT-50) 15,15,16
15 LONGR=NINT+25
  GO TO 18
16 LONGR=NINT+75
  GO TO 18
17 LONGR=0
  GO TO 20
18 IF(IFLG) 19,20,19
19 LONGR=-LONGR
20 CONTINUE
  RETURN
  END

```

CITY SPLITTER AND COORDINATE CONVERTER

Purpose

CITSPL is written in FORTRAN II for the IBM 7090. This sub-routine converts the coordinates of the P95 data to the same form as that of the reformatted cell data, i.e., it places all city data at the centers of the appropriate cells, with coordinates in the integer form $ddd.ddd \text{ deg} \times 10^3 = dddddd$. Cities that overlap cell boundaries must be divided among the cells involved, hence the mnemonic CITSPL (CITY SPLitter), and the data for the cities "placed" at the centers of cells.

Input and Output

The input is a BCD tape file containing unblocked P95 card images in the following format:

<u>Card Image Columns</u>	<u>Data</u>	<u>Mode</u>
1-40	considered to be blank	40X
41-50	codes	5A2
51-53	P95 (nautical miles x 10)	F3.1
54-57	date of data	A4
58-59	blank	2X
60-61	latitude (degrees)	I2
62-63	latitude (minutes)	I2
64-65	latitude (seconds)	I2
66	north or south designator (N or S)	A1
67-69	longitude (degrees)	I3
70-71	longitude (minutes)	I2
72-73	longitude (seconds)	I2
74	east or west designator (E or W)	A1
75-79	population ($\times 10^3$)	I5

The output is an unblocked BCD tape file containing the split-city-segment card images in the following format:

<u>Card Image Columns</u>	<u>Data</u>	<u>Mode</u>
1-6	latitude (degrees to nearest thousandths) ($\times 10^3$)	I6
7-12	longitude (degrees to nearest thousandths) $\times 10^3$)	I6
13	blank	1X
14	1	A1
15-35	blank	21X
36-40	population ($\times 10^3$)	I5
41-44	area ($\times 10^2$)	I4
45-48	date	A4

Program Description

CITSPL processes city data, P95 record by P95 record, until an end-of-file is sensed. Each record is read, counted, and tested for zero population. If a record has no population, it is again counted, and a new record is read. If any P95 radius is zero or less, it is set to 0.5 n mi. The area of the P95 circle is computed using $\text{Area} = \pi(\text{P95})^2$. (1)

The coordinates of the city center are converted by the function subprogram CONVRT. The coordinates of the center of the cell containing the city center are computed by the function subprograms CENLON and CENLAT (discussed in the subsection on CONVTP). To avoid common cell boundaries in assigning a city to a cell, cell boundaries are defined

in the following manner: The longitude boundary on the west shall be one second east of the minute line, at either $x^{\circ}00'01''$ or $x^{\circ}30'01''$; the longitude boundary on the east shall be the minute line, either $x^{\circ}00'00''$ or $x^{\circ}30'00''$.

In the northern hemisphere the latitude boundary to the south is one second north of the minute line, whether $x^{\circ}00'01''$ or $x^{\circ}20'01''$ or $x^{\circ}40'01''$, and the boundary to the north is the minute line, $x^{\circ}00'00''$ or $x^{\circ}20'00''$ or $x^{\circ}40'00''$. In the southern hemisphere the northern boundary is one second south of the minute line and the southern boundary is the minute line. This creates a "no-man's land" at the equator, which is arbitrarily assigned to the northern hemisphere. Thus, if the longitude of a city is 0 deg, it must be set to 360 deg, and if the latitude is 0 deg, it must be set to 1 deg/3600 (Fig. 1, p. 4).

Computations involving longitude must include a correction factor (K) for proximity to the poles:

$$K = \cos (|\text{latitude, in degrees}| \times 0.017452292) \quad (2)$$

This factor is computed once for any point and stored in the common block.

To determine whether a city is to be split, one compares the P95 radius with the distance to the nearest constant latitude and longitude cell boundaries. If a city is not to be split, its record is merely reformatted and written on the BCD tape.

When a city must be split, there are nine possible recipient cells. The quadrant of the cell containing the city center must be determined, (Fig. 4A)

Fig 4A - quadrants of the main cell

Fig. 4B - possible recipient cells and quadrants of main cell. (The main cell is cell 1)

since the quadrant serves as a pointer to the three recipient cells (Fig. 4B) After CIT SPL determines the quadrant, the subroutine SPLIT is called to determine how the city should be divided, and to split it. The area of each segment is computed and then the population, using the following formula:

$$\text{Segment Population} = \frac{\text{segment area}}{\text{total area}} \times \text{total population} \quad (3)$$

SPLIT also determines the center coordinates of the recipient cells, stores the coordinates, areas, and populations in the common output matrix, and returns control to CIT SPL.

The method of splitting a city is based on the assumption that a cell is rectangular. (The error analysis following this subsection shows that this assumption is sufficiently valid for the purposes of the population density study.) The chords formed by overlapped rectangular cell boundaries are perpendicular, greatly simplifying the computation of segment areas. A city can be split among four or fewer of the nine possible recipient cells, in any one of nine ways (Fig. 5). (The main cell is considered to be the fourth recipient.)

Fig. 5--Possible Split Cases

The Split Algorithm*

Fig. 6--A General Split**

* All formulas may be derived from those found in any geometry text. The particular reference is R. S. Burington, Handbook of Mathematical Tables and Formulas, Handbook Publishers, Inc., Sandusky, Ohio, 3rd edition, 1958.

** All other splits are special cases of this one.

Refer to Fig. 6 for illustration of the algorithm. Let R be the radius of the circle, and h_i be the height of a segment i of the circle; then the area of the segment in n miles sq. is

$$\text{Area (segment } i) = R^2 \cos^{-1} \frac{(R-h_i)}{R} - (R-h_i) \sqrt{2Rh_i - h_i^2}$$

$$i = A_1, C+B, \text{ or } D+B \quad (4)$$

where $h_i = R - d_i$. For segments $(C+B)$ and $(D+B)$, d_i is the distance from the city center to the cell boundary in the direction of the segment under consideration. Thus, d_{lat}^* is the perpendicular distance from the city center to the nearest constant latitude boundary, and d_{lon}^{**} is the perpendicular distance to the nearest longitude boundary (cell is assumed rectangular). If (X_0, Y_0) are the latitude and longitude of the city center, and (X_1, Y_1) are the coordinates of the cell center, then for the southern hemisphere

$$d_{lat} = |Y_0| - \left| Y_1 + \frac{1}{6} - \frac{1}{3600} \right|; \text{ or } d_{lat} = |Y_0| - \left| Y_1 - \frac{1}{6} \right| \times 60 \frac{n \text{ miles}}{\text{deg}} \quad (5)$$

and

$$d_{lon} = K \left| X_0 - X_1 + \frac{1}{4} \right|; \text{ or } d_{lon} = K \left| X_0 - \frac{1}{4} + \frac{1}{3600} \right| \times 60 \frac{n \text{ miles}}{\text{deg}} \quad (6)$$

where K (Eq. 2, p.31) is the pole proximity correction. The formula to be used depends on the cell quadrant in which the city is located. Since $h_i = P95 - d_i$, the variables necessary for computing the area of a circle segment are known. Thus one computes the areas of segments $(C+B)$ and $(D+B)$. Now, the area of the circle is the sum of the areas of the four sections, A, C, D, and B; therefore, the area of B is

*Where d_{lat} can be called d_{C+B}

**Where d_{lon} can be called d_{D+B}

$$B = -\pi R^2 + A + (D+B) + (C+B) \quad (7)$$

and one now needs only the area of A to complete the computation.

Section A can be divided into two parts--a segment A_1 , and a triangle A_2 .

Fig. 7--Triangular Portion of Section A

To find the area of the triangle, one must first find the lengths of the perpendicular legs, $\overline{P_1P_2}$ and $\overline{P_2P_3}$.

By inspection $\overline{P_1P_2}$ is the sum of d_{lon} and the leg of the triangle formed by d_{lat} and the radius vector R, or

$$\overline{P_1P_2} = d_{lon} + \sqrt{R^2 - d_{lat}^2} \quad (8)$$

and similarly,

$$\overline{P_2P_3} = d_{lat} + \sqrt{R^2 - d_{lon}^2} \quad (9)$$

so that the area of the triangle is

$$A_2 = \frac{1}{2} (\overline{P_1P_2}) (\overline{P_2P_3}), \quad (10)$$

and the length $\overline{P_1P_3}$ is

$$\overline{P_1P_3} = \sqrt{(\overline{P_1P_2})^2 + (\overline{P_2P_3})^2} \quad (11)$$

Again, using the formula for area of a segment and substituting

$R - \frac{1}{2} \sqrt{4R^2 - (\overline{P_1 P_3})^2}$ for h_A , one finds that the area of A is the area of the segment plus the area of the triangle, or

$$A = R^2 \cos^{-1} \left(\frac{R - h_A}{R} \right) - (R - h_A) \sqrt{2Rh_A - h_A^2} + \frac{1}{2} (\overline{P_1 P_2}) (\overline{P_2 P_3}). \quad (12)$$

It is necessary to determine the type of split before computing the areas, in order to avoid using the algorithm in its most general form (as presented here). The determination is made by comparing the radius to d_{lat} , d_{lon} , and to the diagonal distance from the cell corner to the city center. For example, if $d_{lat} \leq R$ and $d_{lon} > R$, then the city will split into two parts across a longitude boundary, and the subroutine SEGMNT, which computes the area of a circle segment, need be called only once.

Error Analysis

Figures 8-10 illustrate the error analysis for the assumption that cells are rectangular. The conclusions drawn from that analysis follow.

The total error in splitting any city is ≤ 0.278 percent. The maximum population allowed in a city is 99999×10^3 . Thus the maximum possible error due to assuming rectangular cells is 277977 people. These people are not lost by the program, but are assigned to the wrong cells.

Figures 8-10 show that for most splits, the errors tend to cancel, so that the true error is probably much less than one-half the maximum error, in all cases except a four-equal-part split. In that case the error is the maximum, but the case is the least probable one.

Fewer than 10 cities have populations as large as 10 million. If all the population of one of those cities is given in one circle and the city splits, the error for the city is approximately 28000 in the worst case but is more likely to be 1000 to 10000. Since most cities have populations of 1 million or less, the errors are on the order of 100 to 1000 per city. Due to the format of the P95 data, which records city population to the nearest thousand, these errors are not evident in the final city records. In fact, a city must have a population greater than 360×10^3 before an error as large as 0.278 percent is evident.

($c \equiv$ Cell Corner $a \equiv$ Apparent Corner R is Radius)
 (Based on area of sector of a circle: $A = \frac{1}{2} R^2 \theta$ (where $\theta = 0015'0''$)

I. This case, although a 4-part split, appears to the program to be a 2-part split. Error is negligible since shaded area is only $R^2 \times 10^{-6}$ sq n mi.

II. This case appears to be a 2-pair-of-equal-parts split. Hatched area is $2.18 R^2 \times 10^{-3} \text{ mi}^2$. Triangular area is $\approx 2.18 X^2 \times 10^{-3} \text{ mi}^2$, $X < R$. Shaded area $\approx 2.18 (R^2 - X^2) \times 10^{-3} \text{ mi}^2$, $X < R$. Total error is $4(2.18 R^2 \times 10^{-3} \text{ mi}^2)$.

III. This case appears to be a 4-equal-part split. The error in each segment is $2.18 R^2 \times 10^{-3} \text{ mi}^2$. Thus total city error is $4(2.18 R^2 \times 10^{-3} \text{ mi}^2)$. This is the worst case, since all errors add.

IV. This case is a 4-part split. The triangular area is $\approx 2.18 X^2 \times 10^{-3} \text{ mi}^2$, $X < R$. The hatched area H is $0 \leq H \leq 2.18 R^2 \times 10^{-3} \text{ mi}^2$. The shaded area is approximately $2.18 (R^2 - X^2) \times 10^{-3} \text{ mi}^2$, $X < R$. Thus the total error is again $\approx 4(2.18 R^2 \times 10^{-3} \text{ mi}^2)$.

Total city error for a four part split is $\leq 0.278\%$.

Fig. 8--Four Part Splits

I. This case appears to be a 2-part split. Shaded area is at most $R^2 \times 10^{-6} \text{ mi}^2$.

II. This case is a 3-part split. The shaded area H is $R^2 \times 10^{-6} \text{ mi}^2 \leq H < 2.18 X^2 \times 10^{-3} \text{ mi}^2$, $X < R$. The hatched area is $< H$.

III. This case $\equiv$ Case I.

Total city error for 3-part split is $\ll 4 (2.18 R^2 \times 10^{-3} \text{ mi}^2)$, $\ll 0.278\%$.

Fig. 9--Three Part Splits

I. This case is a 2-part split that appears to be no split. Shaded area is at most $R^2 \times 10^{-6} \text{ mi}^2$.

II. This is a 2-part split. The shaded area will always be less than or $\approx 2.18 R^2 \times 10^{-3} \text{ mi}^2$. The hatched area is always $< 2.18 R^2 \times 10^{-3} \text{ mi}^2$.

III. This is a half split. The errors cancel.

Total city error for a 2-part split is $\leq 4(2.18 R^2 \times 10^{-3} \text{ mi}^2) \leq 0.278\%$.

Fig. 10--Two Part Splits

Flow Chart for City Splitter--CITSPL

City Splitter (Continued)

City Splitter (Continued)

City Splitter (Continued)

City Splitter (Continued)

City Splitter (Continued)

City Splitter (Concluded)

```

CCITSPL
C      **** CITY SPLITTER AND COORDINATE CONVERTER ****
C      NUCLEAR SAFETY -- CHARGE 2041.9 -- VANDEVENDER
      DIMENSION NCELL(5,9),KOUNT(8)
      COMMON POP,AREA,P95,IW,IS,NCELL,KOUNT
      COMMON ONE3,ONE6,ONE360,FIVE6,TWO3,FUDGE
      DIMENSION ICODE(5)
      READ INPUT TAPE 2,1,IE,IW,IN,IS,ITYPE
1      FORMAT (5(2X,A1))
      NUMB=0
      DO 2 K=1,8
      KOUNT(K)=0
2      CONTINUE
      ONE6=1.0/6.0
      ONE3=1.0/3.0
      TWO3=2.0/3.0
      FIVE6=5.0/6.0
      ONE360=1.0/3600.0
      IZERO=0
      NOTPUT=C
      REWIND 5
      REWIND 16
3      IF(NBUF(5)) 4,6,46
C
C      PARITY CHECK ERROR
C
4      WRITE OUTPUT TAPE 3,5,NUMB
5      FORMAT (1H1,30H PARITY ERROR - RECORD NUMBER ,16)
      NUMB=NUMB+1
      GO TO 3
6      CALL RRFAD
      READ INPUT TAPE 5,7,(ICODE(J),J=1,5),P95,IDATE,LATDEG,LATMIN,
      $LATSEC,NS,LONDEG,LONMIN,LONSEC,IEW,IPOP
7      FORMAT (40X,5A2,F3.1,A4,2X,3I2,A1,I3,2I2,A1,I5,5X)
      NUMB=NUMB+1
C      P95 = RADIUS OF CELL
      IF(IPOP) 9,8,9
8      KOUNT(8)=KOUNT(8)+1
      GO TO 3
9      IF(P95) 10,10,11
10     P95=0.5
11     AREA=3.1416*P95*P95
      POP=IPOP
      ISFLAG=2
      ITEMP=NS-IS
      IF(NS-IS) 13,12,13
12     ISFLAG=1
13     CONTINUE
C
C      HEMISPHERE DETERMINATION
C
C      LOCATE CENTER OF CIRCLE IN A CELL WITH KNOWN CENTER
C
      XCELL=CENLON(LONDEG,LONMIN,LONSEC,IEW)
C

```



```

C      XCELL = LONGITUDE OF CELL CENTER IN DEGREES
C
C      YCELL=CENLAT(LATDEG,LATMIN,LATSEC,NS)
C
C      YCELL = LATITUDE OF CELL CENTER IN DEGREES
C
C      XCITY=CONVRT(LONDEG,LONMIN,LONSEC,IEW)
C
C      XCITY = LONGITUDE OF CITY CENTER IN DEGREES
C
C      YCITY=CONVRT(LATDEG,LATMIN,LATSEC,NS)
C
C      YCITY = LATITUDE OF CITY CENTER IN DEGREES
C
C
C      IQUAD=0
C      IFLG1=0
C
C      IF(XCITY) 15,14,15
14      XCITY=360.0000
15      IF(YCITY) 17,16,17
16      YCITY=YCITY+ONE360
17      CTEMP=XCITY-XCELL
C
C      FIND PROPER QUADRANT
C
C      FUDGE=COSE(ABSF(YCITY)*0.017452292)
C
C      FUDGE=CORRECTION FOR CURVATURE
C
C      IF((ABSF(CTEMP)*60.0-14.98)*FUDGE+P95) 19,19,18
18      IF(CTEMP) 21,20,22
19      DTEMP=ABSF(YCITY)-ABSF(YCELL)
C
C      LONGITUDE DISTANCE
C
C
C      IF(ABSF(DTEMP)*60.0+P95-9.98) 30,30,18
20      IFLG1=1
C      IQUAD=IQUAD+1
C      GO TO 23
21      IQUAD=1
C      GO TO 23
22      IQUAD=0
23      GO TO (29,24),ISFLAG
24      IF(YCITY-YCELL) 27,25,28
25      IFLG1=IFLG1+1
C
C      IF(IFLG1-2) 27,30,31
26      IFLG1=IFLG1+1
C      IF(IFLG1-2) 28,30,31
27      IQUAD=IQUAD+3
C      GO TO 31
28      IQUAD=IQUAD+1
C      GO TO 31
29      IF(YCITY-YCELL) 27,26,28

```

```

30  NCELL(3,1)=POP
    NCELL(4,1)=AREA*100.
    KOUNT(1)=KOUNT(1)+1
    GO TO 36
31  IF(IQUAD) 32,32,34
32  WRITE OUTPUT TAPE 3,33,IQUAD,XCELL,YCELL,ISFLAG,XCITY,YCITY,NUMB
33  FORMAT (7H0IQUAD=,I4,6H CELL=,2F10.3,8H ISFLAG=,I4,13H COORDINATES
    $=,2F10.3,6H CARD=,I8)
    GO TO 3
34  IF(IQUAD-4) 35,35,32
35  CONTINUE
    CALL SPLIT (IQUAD,XCELL,YCELL,ISFLAG,XCITY,YCITY,ILAT,ILON,ICORN)
36  NCELL(2,1)=XCELL*100.
    NCELL(1,1)=YCELL*1000.
    JPOP=0
    DO 43 IK=2,9
    IF(IK-TCOR) 37,39,37
37  IF(IK-ILAT) 38,39,38
38  IF(IK-TCORN) 43,39,43
39  IF(NCELL(3,IK)) 41,40,41
40  NOTPUT=NOTPUT+1
41  NCELL(5,IK)=IDATE
    JPOP=JPOP+NCELL(3,IK)
    NCELL(1,IK)=MINCHK(1,IK)
    IF(ISFLAG=1) 411,412,411
C
C      ADD 90 TO LATITUDE.
C
411 IDIGIT=NCELL(1,IK)-(NCELL(1,IK)/10)*10
    NFIRST=(NCELL(1,IK)/10)+9000
    GO TO 415
412 NFIRST=NCELL(1,IK)+90000
    IDIGIT=NFIRST-(NFIRST/10)*10
    NFIRST=NFIRST/10
C
C      OUTPUT THE SPLIT CITIES
C
415 WRITE OUTPUT TAPE 16,42,NFIRST,IDIGIT,NCELL(2,IK),IZERO,ITYPE,
    $(NCELL(JK,IK),JK=3,5)
42  FORMAT (2(I5,I1),1X,A1,21X,I5,I4,A4,36X)
43  CONTINUE
    NCELL(1,1)=MINCHK(1,1)
    IF(ISFLAG=1) 432,431,432
431 NFIRST=NCELL(1,1)+90000
    IDIGIT=NFIRST-(NFIRST/10)*10
    NFIRST=NFIRST/10
    GO TO 433
432 IDIGIT=NCELL(1,1)-(NCELL(1,1)/10)*10
    NFIRST=(NCELL(1,1)/10)+9000
433 NCELL(3,1)=IPOP-JPOP
    NCELL(5,1)=IDATE
    WRITE OUTPUT TAPE 16,42,NFIRST,IDIGIT,NCELL(2,1),IZERO,ITYPE,
    $(NCELL(JK,1),JK=3,5)
    DO 45 IK=1,9
    DO 44 JK=1,5

```

```

      NCELL(JK,IK)=0
44      CONTINUE
45      CONTINUE
      ILON=0
      ILAT=0
      ICORN=0
      GO TO 3
C
C      END OF FILE WAS FOUND
C
46      END FILE 16
      CALL RUN (5)
      CALL RUN (16)
      WRITE OUTPUT TAPE 3,47,(KOUNT(I),I=1,7),NUMB,KOUNT(8),NOIPUI
47      FORMAT (1X,I4,16H WERE NOT SPLIT,71X,I4,22H WERE SPLIT IN HALVES
$,/1X,I4,23H WERE SPLIT IN FOURTHS,/1X,I4,23H WERE SPLIT IN 3 PAR
STS,/1X,I4,23H WERE SPLIT IN 2 PARTS,/1X,I4,37H WERE SPLIT IN 2 P
SAIR OF EQUAL PARTS,/1X,I4,23H WERE SPLIT IN 4 PARTS,/1X,I4,17H C
ARDS WERE READ,/1X,I4,53H CITIES HAD ZERO POPULATIONS AND WERE NO
ST PROCESSED.,/1X,I4,47H SPLIT SEGMENTS HAD LESS THAN 1000 POPULAT
SION.)
      CALL EXIT
      END

```

```

SUBROUTINE SPLIT (IQUAD,XCELL,YCELL,ISFLAG,XCITY,YCITY,ILAT,ILON
$,ICORN)
  DIMENSION NCELL(5,9),KOUNT(8)
  COMMON POP,AREA,P95,IW,IS,NCELL,KOUNT
  COMMON ONE3,ONE6,ONE360,FIVE6,TWO3,FUDGE
  DIMENSION CELL(4)
  C   ADJACENT CELLS RECEIVING PART OF CITY POPULATION DEPEND
  C   UPON QUADRANT OF CELL CONTAINING CITY CENTER.
  C
  IF(IQUAD=2) 1,2,4
1    ICORN=2
    ILAT=9
    ILON=2
    NCELL(2,ICORN)=(XCELL+.50)*100.
    NCELL(2,ILON)=(XCELL+.50)*100.
    GO TO 3
2    ICORN=4
    ILAT=9
    ILON=5
    NCELL(2,ICORN)=(XCELL-.50)*100.
    NCELL(2,ILON)=(XCELL-.50)*100.
3    NCELL(1,ILAT)=(YCELL+ONE3)*1000.
    GO TO 8
4    IF(IQUAD=4) 5,6,6
5    ICORN=6
    ILAT=7
    ILON=3
    NCELL(2,ICORN)=(XCELL+.50)*100.
    NCELL(2,ILON)=(XCELL+.50)*100.
    GO TO 7
6    ICORN=8
    ILAT=7
    ILON=5
    NCELL(2,ICORN)=(XCELL-.50)*100.
    NCELL(2,ILON)=(XCELL-.50)*100.
7    NCELL(1,ILAT)=(YCELL-ONE3)*1000.
8    NCELL(2,ILAT)=XCELL*100.
    NCELL(1,ILON)=YCELL*1000.
    IF(ICORN=4) 9,9,10
9    NCELL(1,ICORN)=(YCELL+ONE3)*1000.
    GO TO 11
10   NCELL(1,ICORN)=(YCELL-ONE3)*1000.
11   CONTINUE
  C
  C   FIND LATITUDE DISTANCE AND LONGITUDE DISTANCE IN NAUTICAL
  C   MILES FROM CENTER OF CIRCLE TO NEAREST CELL BOUNDARY
  C   AND CORNER OF CELL.
  C   XBOUND = LONGITUDE OF NEAREST BOUNDARY TO EAST OR WEST
  C   YBOUND = LATITUDE OF NEAREST BOUNDARY TO NORTH OR SOUTH
  GO TO (12,13,12,13),IQUAD
12   XBOUND=XCELL+.2500
    GO TO 14
13   XBOUND=XCELL-(0.25-ONE360)
14   GO TO (15,18),ISFLAG
  C   SOUTHERN HEMISPHERE
15   GO TO (16,16,17,17),IQUAD
16   YBOUND=YCELL+(ONE6-ONE360)
    GO TO 21
17   YBOUND=YCELL-ONE6
    GO TO 21

```

C	NORTHERN HEMISPHERE	6
18	GO TO (19,19,20,20),IQUAD	6
19	YBOUND=YCELL+ONE6	6
	GO TO 21	6
20	YBOUND=YCELL-(ONE6-ONE360)	6
C	DLAT = DISTANCE FROM YCITY TO YBOUND	6
C	DLON = DISTANCE FROM XCITY TO XBOUND	6
C	DIAG = DISTANCE FROM CITY CENTER TO CORNER OF CELL	6
21	DLAT=ABSF(YCITY)-ABSF(YBOUND)	6
	DLAT=ABSF(DLAT)*60.0	7
	DLON=ABSF(XCITY-XBOUND)*60.0*FUDGE)	7
	DIAG=SQRTF(DLAT*DLAT+DLON*DLON)	7
	IFLGD=0	7
	IF(DLAT-ONE6/10.) 22,22,23	7
22	IFLGD=1	7
	DLAT=0.0	7
23	IF(DLON-ONE6/10.*FUDGE) 24,24,25	7
24	IFLGD=IFLGD+2	7
	DLON=0.0	7
	IF(IFLGD-3) 25,49,25	8
25	CONTINUE	8
C		8
C		8
C	ALL FLAGS HAVE BEEN SET FOR PROPER CHOICE	8
C	OF CELL SPLIT	8
C		8
C	DETERMINE TYPE OF CELL SPLIT	8
	IFLAGR=0	8
	IF(P95-DLAT) 26,27,28	8
26	IFLAGR=1	9
	GO TO 29	9
27	IFLAGR=2	9
	GO TO 29	9
28	IFLAGR=3	9
29	IF(P95-DLON) 32,30,31	9
30	IFLAGR=IFLAGR+3	9
	GO TO 32	9
31	IFLAGR=IFLAGR+6	9
C	DETERMINE METHOD OF CALCULATING AREA	9
32	CONTINUE	10
	GO TO (44,44,33,44,44,33,34,34,38),IFLAGR	10
33	IFLAG2=1	10
	GO TO 35	10
34	IFLAG2=2	10
35	IF(IFLGD-1) 37,36,36	10
36	IFLG24=1	10
	GO TO 42	10
37	IFLG24=2	10
	GO TO 42	10
38	CONTINUE	11
	IF(P95-DIAG) 50,50,39	11
39	IF(IFLGD-1) 41,40,40	11
40	IFLG24=3	11
	GO TO 42	11
41	IFLG24=4	11
42	CONTINUE	11
	GO TO 43	11
43	CONTINUE	11
	GO TO (45,51,55,59),IFLG24	11

```

C      NO SPLIT
44     KOUNT(1)=KOUNT(1)+1
      NCELL(4,1)=AREA*100.
      ILAT=0
      ILON=0
      ICORN=0
      RETURN
C      SPLIT IS 2 EQUAL PARTS
45     KOUNT(2)=KOUNT(2)+1
      GO TO (46,47),IFLAG2
46     NCELL(3,ILAT)=POP/2.0
      NCELL(4,ILAT)=AREA*50.
      ILON=0
      GO TO 48
47     NCELL(3,ILON)=POP/2.0
      NCELL(4,ILON)=AREA*50.
      ILAT=0
48     NCELL(4,1)=AREA*50.
      ICORN=0
      RETURN
C      SPLIT IS 4 EQUAL PARTS
49     TPOP=POP/4.0
      AREA=AREA*25.0
      NCELL(4,1)=AREA
      NCELL(3,ILAT)=TPOP
      NCELL(4,ILAT)=AREA
      NCELL(3,ILON)=TPOP
      NCELL(4,ILON)=AREA
      NCELL(3,ICORN)=TPOP
      NCELL(4,ICORN)=AREA
      KOUNT(3)=KOUNT(3)+1
      RETURN
C      SPLIT IS 3 PARTS
50     KOUNT(4)=KOUNT(4)+1
      CELL(1)=SEGMNT(DLAT)
      CELL(4)=POP*CELL(1)/AREA
      CELL(2)=SEGMNT(DLON)
      CELL(3)=POP*CELL(2)/AREA
      NCELL(3,ILAT)=CELL(4)
      NCELL(4,ILAT)=CELL(1)*100.
      NCELL(4,ILON)=CELL(2)*100.
      NCELL(3,ILON)=CELL(3)
      NCELL(4,1)=(AREA-CELL(1)-CELL(2))*100.
      ICORN=0
      RETURN
C      SPLIT IS 2 PARTS
51     KOUNT(5)=KOUNT(5)+1
      GO TO (52,53),IFLAG2
52     CELL(1)=SEGMNT(DLAT)
      CELL(2)=POP*CELL(1)/AREA
      NCELL(4,ILAT)=CELL(1)*100.0
      NCELL(3,ILAT)=CELL(2)
      ILON=0
      GO TO 54
53     CELL(1)=SEGMNT(DLON)
      CELL(2)=POP*CELL(1)/AREA
      NCELL(3,ILON)=CELL(2)
      NCELL(4,ILON)=CELL(1)*100.0
      ILAT=0

```

```

54 NCELL(4,1)=(AREA-CELL(1))*100.0
   ICORN=0
   RETURN
C   SPLIT IS 2 PAIR OF EQUAL PARTS
55 KOUNT(6)=KOUNT(6)+1
   GO TO (56,57),IFLGD
56 CELL(1)=SEGMNT(DLON)
   CELL(2)=POP*CELL(1)/AREA
   NCELL(4,ILON)=CELL(1)*100.
   NCELL(3,ILON)=CELL(2)
   NCELL(4,ICORN)=NCELL(4,ILON)
   NCELL(3,ICORN)=NCELL(3,ILON)
   NCELL(3,ILAT)=POP/2.-CELL(2)
   NCELL(4,ILAT)=(AREA/2.-CELL(1))*100.0
   GO TO 58
57 CELL(1)=SEGMNT(DLAT)
   CELL(2)=POP*CELL(1)/AREA
   NCELL(3,ICORN)=CELL(2)
   NCELL(4,ICORN)=CELL(1)*100.0

   NCELL(3,ILAT)=NCELL(3,ICORN)
   NCELL(4,ILAT)=NCELL(4,ICORN)
   NCELL(3,ILON)=POP/2.-CELL(2)
   NCELL(4,ILON)=(AREA/2.-CELL(1))*100.0
58 NCELL(4,1)=(AREA/2.-CELL(1))*100.0
   RETURN
C   SPLIT IS 4 UNEQUAL PARTS
59 KOUNT(7)=KOUNT(7)+1
   HSEG=SEGMNT(DLAT)
   TEMP=DLAT-DLON
   IF(DLAT-DLON) 61,60,61
60 VSEG=HSEG
   GO TO 62
61 VSEG=SEGMNT(DLON)
62 RSQR=P95*P95
   DTEMP=SQRTF(RSQR-DLON*DLON)+DLAT
   CTEMP=SQRTF(RSQR-DLAT*DLAT)+DLON
   ASEG=CTEMP*CTEMP+DTEMP*DTEMP
   D3=SQRTF(4.0*RSQR-ASEG)/2.0
   ASEG=SEGMNT(D3)+CTEMP*DTEMP/2.0
   BSEG=ASEG+HSEG+VSEG-AREA
   NCELL(4,ILAT)=(HSEG-BSEG)*100.0
   NCELL(4,ILON)=(VSEG-BSEG)*100.0
   NCELL(4,ICORN)=BSEG*100.0
   NCELL(4,1)=ASEG*100.0
   NCELL(3,ILAT)=POP*(HSEG-BSEG)/AREA
   NCELL(3,ILON)=POP*(VSEG-BSEG)/AREA
   NCELL(3,ICORN)=POP*BSEG/AREA
   RETURN
END

```

```

FUNCTION CONVRT (I,J,K,L)
  DIMENSION NCELL(5,9),KOUNT(8)
  COMMON POP,AREA,P95,TW,IS,NCELL,KOUNT
  COMMON ONE3,CNE6,ONE360,FIVE6,TWO3
  IF(L-TW) 10,1,10
1  II=359-I
  JJ=59-J
  KK=60-K
  IF(KK-60) 3,2,3
2  KK=0
  JJ=JJ+1
3  IF(JJ-60) 5,4,5
4  JJ=0
  II=II+1
5  IF(II-360) 7,6,6
6  II=II-360
7  XI=II
  XJ=JJ
  XK=KK
  CONVRT=XI+XJ*ONE6/10.+XK*ONE360
  IF(L-IS) 9,8,9
8  CONVRT=-CONVRT
9  RETURN
10 II=I
  JJ=J

  KK=K
  GO TO 5
END

```



```

FUNCTION SEGMENT (D)
  DIMENSION NCELL(5,9),KOUNT(8)
  COMMON POP,AREA,P95,IW,IS,NCELL,KOUNT
  COMMON ONE3,ONE6,ONE360,FIVE6,TWO3
  H=P95-D
  X=D/P95
  SEGMENT=P95*P95*ATANF(SQRTF(1.-X*X)/X)-D*SQRTF(2.0*P95*H-H*H).
  RETURN
END

```

```

      FUNCTION CENLON (I,J,K,L)
      DIMENSION NCELL(5,9),KOUNT(8)
      COMMON POP,AREA,P95,IW,IS,NCELL,KOUNT
      COMMON ONE3,ONE6,ONE360,FIVE6,TWO3
      IF(L-IW) 10,1,10
1      II=359-I
      JJ=59-J
      KK=60-K
      IF(KK-60) 3,2,3
2      KK=0
      JJ=JJ+1
3      IF(JJ-60) 5,4,5
4      JJ=0
      II=II+1
5      IF(II-360) 7,6,6
6      II=II-360
7      MINSEC=JJ*100+KK
      XI=II
      XJ=JJ
      XK=KK
      IF(MINSEC) 11,8,11
8      CENLON=(XI-1.)*.7500
      IF(XI) 9,14,9
9      RETURN
10     II=I
      JJ=J
      KK=K
      GO TO 5
11     IF(3001-MINSEC) 12,12,13
12     CENLON=XI+.7500
      RETURN
13     CENLON=XI+.2500
      RETURN
14     CENLON=359.7500
      RETURN
      END

```

```

FUNCTION CENLAT (I,J,K,L)
DIMENSION NCELL(5,9),KOUNT(8)
COMMON POP,AREA,P95,TW,IS,NCELL,KOUNT
COMMON ONE3,ONE6,ONE360,FIVE6,TWO3

```

```

MINSEC=J*100+K

```

```

XI=I

```

```

XJ=J

```

```

XK=K

```

```

IF(XI) 1,3,1

```

```

1 IF(MINSEC) 3,2,3

```

```

2 CENLAT=(XI-1.)+FIVE6

```

```

GO TO 8

```

```

3 IF(4000-MINSEC) 4,6,5

```

```

4 CENLAT=XI+FIVE6

```

```

GO TO 8

```

```

5 IF(2001-MINSEC) 6,6,7

```

```

6 CENLAT=XI+.5000000000

```

```

GO TO 8

```

```

7 CENLAT=XI+ONE6

```

```

8 IF(L-IS) 10,9,10

```

```

9 CENLAT=-CENLAT

```

```

10 RETURN

```

```

END

```

```

FUNCTION MINCHK (JK,IK)
  DIMENSION NCELL(5,9),KOUNT(8)
  COMMON POP,AREA,P95,IW,IS,NCELL,KOUNT
  COMMON ONE3,ONE6,ONE360,FIVE6,TWO3,FUDGE
  IF(NCELL(1,IK)) 1,10,2
1  NEGF=1
  NCELL(1,IK)=-NCELL(1,IK)
  GO TO 3
2  NEGF=0
3  NTEMP=(NCELL(1,IK)/1000)*1000
  ITEMP=XABS(NCELL(1,IK)-NTEMP)
  IF(ITEMP-250) 4,4,5
4  NCELL(1,IK)=NTEMP+167
  GO TO 8
5  IF(ITEMP-750) 6,6,7
6  NCELL(1,IK)=NTEMP+500
  GO TO 8
7  NCELL(1,IK)=NTEMP+833
8  IF(NEGF) 9,10,9
9  NCELL(1,IK)=-NCELL(1,IK)
10 MINCHK=NCELL(1,IK)
  RETURN
  END

```

CELGRP

Purpose

CELGRP is written in Fortran II for the IBM 7090. This program is the last executed program in the sequence of routines used to generate complete, sorted records of world population data.

Input and Output

The program accepts, as input, tapes containing both rural and urban cell records, in a specified, sorted format. It compares records for rural cell omissions and duplications, generating cells in the former case, removing cells in the latter.

Output consists of a set of one or more tapes containing both rural and urban cell records (in the same specified, sorted format as the input tapes), and a summary page containing the number of cells checked, the number generated, and the number removed. If errors occurred, an error printout for each error is made, containing the record image. That record is omitted from the output tapes.

Program Description

The process of data collection up to this point has resulted in tapes containing cell images and city images, with city images immediately following cell images. It is possible, in this sequence, to have cities with no associated cells, and duplicate cells. CELGRP provides cells in the former case, and combines cells in the latter.

The program first reads a card containing the following setup information:

<u>Variable</u>	<u>Contents</u>	<u>Format</u>
ICLK	1 $\emptyset$	A2
IRPLSE	F or S	A1
IZERO	0 $\emptyset$	A2
IT	Starting LUN	I2
LASTIT	Last LUN	I2
ISPACE	$\emptyset$	

$\emptyset$ = blank; LUN = Logical Unit Number.

The variable IRPLSE is used to identify the resulting cell from either a generation or a removal process, and contains the identifier "F" initially. IT identifies the location of the first input tape, while LASTIT identifies the location of the last. All input tapes must be sequentially mounted.

After initializing various counters and reading this setup information, the program enters the main loop, reading records in groups of 100 (or less). During this read-in process, the records are checked for cell

omissions; if any occur, new cells are generated and added to the input stream. When 100 records have been read or generated, they are then checked for duplicate entries. Duplicate records are combined into one record by Subroutine CELRMV. After the records are checked they are written onto the output tape in the same format as the input tapes. When an end-of-file (EOF) is encountered, those records read in are processed, an EOF is written on the output tape, and the program checks to see whether more input tapes exist. If so, processing continues, if not, the program exits.

CELGRP accepts as tape inputs up to 10 BCD tapes containing cell and city data. The records on these tapes should be blocked 1 x 84 in one file. The records must be sorted by latitude, by longitude within a latitude band, and by the rural-urban key, with rural cells preceding city cells. CELGRP reads these records into the arrays in the format specified below:

<u>Variable</u>	<u>Contents</u>	<u>Format</u>
LAT(I)	Latitude	I5
IDIGIT(I)	Last digit of latitude	I1
LONG(I)	Longitude	I5
(skipped)	Last digit of longitude	1X
ICODE(I, 1)	F or S	A1
ICODE(I, 2)	Rural-Urban key	A2
IPOP(I)	Population	I5
IBUG(I, J)	Remainder of record	32A2

The main output is at least one BCD output tape per BCD input tape, each tape containing records blocked 1 x 84 in one file. These output tapes constitute the final product of the series of 7090 programs, and contain the complete set of rural and urban cells in sorted format.

Other output can include a diagnostic error message and a statement of the number of input reels read, the number of cells removed, and the number of cells generated.

The diagnostic printout has the following format:

TAPE READ ERROR ON INPUT

147167 34750 1 38 4520266

THE INPUT REEL WAS 6, THE OUTPUT REEL WAS 16

THE RECORD GROUP WAS 588, AND THE RECORD WAS 14.

THE RECORD WILL BE SKIPPED.

The second line contains the record image; for this example, it was a city record. The third line states the LUNs of the input and output reels. The fourth line states which record in the number sequence 1-N was rejected; in this case,

Record rejected = (record group) 100 + record = 58800 + 14
= 58814

The count message has the format:

FOR THE 3 INPUT REELS 3965 CELLS WERE REMOVED,
AND 36 CELLS WERE GENERATED (ADDED).

For three reels of input information, containing approximately 90000 records, the CELGRP run time was approximately 50 minutes.

1

CELGRP (Continued)

CELGRP (Concluded)

```

CCELGRP
  DIMENSION LAT(100),LONG(100),IBUG(100,32),IPOP(100),ICODE(100,2)
  DIMENSION IDUMP(84),IDIGIT(100)
  COMMON LAT, LONG, IBUG, IPOP, ICODE, ICHK, IRPLSE, IZERO, IDIGIT
  COMMON KOUNT2, KOUNT3
  COMMON ISPACE

C
C      REDUNDANT CELL REMOVER PROGRAM.
C
C
C      7090 IMPLEMENTATION.
C      ICHK=1, IRPLSE=F, IZERO=0, IT=FIRST INPUT TAPE
C      NUMBER (I2), LASTIT=LAST INPUT TAPE NUMBER (I2).
C
  READ INPUT TAPE 2, I, ICHK, IRPLSE, IZERO, IT, LASTIT, ISPACE
1  FORMAT (A2, A1, A2, 2I2, A1)
  KOUNT1=1
  KOUNT2=0
  KOUNT3=0
  IDIFR=LASTIT-IT+1
2  REWIND IT
  IO=IT+10
  REWIND IO

C
C      READ THE FIRST RECORD.
C
  IF(NBUF(IT)) 38,3,38
3  CALL REREAD
  READ INPUT TAPE IT, 4, LAT(1), IDIGIT(1), LONG(1), (ICODE(1,I), I=1,2),
  $IPOP(1), (IBUG(1,I), I=1,32)
4  FORMAT (I5, I1, I5, IX, A1, A2, I5, 32A2)

C
C      IF FIRST CELL IS A CITY, GENERATE ANOTHER.
C
  IF(ICODE(1,2)-ICLK) 6,5,6
5  CALL CELGEN (1,L)
  GO TO 7
6  L=2
7  IFLAG=0
  ICELL=0
8  I=L
9  IF(NBUF(IT)) 38,10,17
10 CALL REREAD
  READ INPUT TAPE IT, 4, LAT(I), IDIGIT(I), LONG(I), (ICODE(I,J), J=1,2),
  $IPOP(I), (IBUG(I,J), J=1,32)
  IF(ICODE(I,2)-ICLK) 11,12,11
11 ICELL=I
  GO TO 15
12 IF(LAT(I)-LAT(ICELL)) 14,13,14
13 IF(LONG(I)-LONG(ICELL)) 14,15,14

C
C      A CITY CELL HAS BEEN FOUND WHICH HAS NO RURAL CELL.
C
14 CALL CELGEN (1,NEXT)
  ICELL=I

```

```

      I=NEXT
      GO TO 16
15     I=I+1
16     IF(I-99) 9,9,18
17     IFLAG=1
18     IMAX=I-1
      KOUNT1=KOUNT1+1
C
C      INPUT OF RECORDS IS COMPLETED. COMPARE ADJACENT
C      RECORDS FOR DUPLICATE ENTRIES.
C
      I=2
      LAST=I
19     IF(ICODE(I,2)-ICLK) 20,24,20
20     IF(LAT(I)=LAT(LAST)) 23,21,23
21     IF(LONG(I)-LONG(LAST)) 23,22,23
C
C      CELLS ARE DUPLICATES. REMOVE ONE.
C
22     CALL CELRMV (LAST,I,JUMP)
      GO TO (23,24),JUMP
23     LAST=I
24     I=I+1
      IF(I-IMAX) 19,19,25
C
C      RECORDS HAVE BEEN CHECKED. WRITE A NEW TAPE, BUT DO
C      NOT WRITE THE LAST RECORD, WHICH STILL HAS TO BE
C      CHECKED FOR DUPLICATION.
C
25     I=2
      J=1
26     IF(LAT(I)-99999) 27,31,27
27     IF(LAT(J)-99999) 29,28,29
28     J=J+1
      GO TO 31
29     WRITE OUTPUT TAPE 10,30,LAT(J),IDIGIT(J),LONG(J),IZERO,ICODE(J,1),
      SICODE(J,2),IPOP(J),(IBUG(J,K),K=1,32)
30     FORMAT (I5,I1,I5,2A1,A2,I5,32A2)
      J=I
31     I=I+1
      IF(I-IMAX) 26,26,32
C
C      NOW TRANSFER THE LAST RECORD TO THE FIRST POSITION.
C
32     IF(IFLAG=1) 33,37,33
33     LAT(1)=LAT(J)
      IDIGIT(1)=IDIGIT(J)
      LONG(1)=LONG(J)
      ICODE(1,1)=ICODE(J,1)
      ICODE(1,2)=ICODE(J,2)
      DO 34 K=1,32
      IBUG(1,K)=IBUG(J,K)
34     CONTINUE
      IPOP(1)=IPOP(J)
C

```

```

C      RETURN FOR MORE.
C
      L=2
      IF(ICODE(1,2)-ICLK) 35,36,35
35     ICELL=1
      GO TO 8
36     ICELL=0
      GO TO 8
C
C      THE EOF WAS DETECTED, SO WRITE THE LAST RECORD
C      AND REWIND THE TAPE.
C
37     WRITE OUTPUT TAPE 10,12,LAT(J),IDIGIT(J),LONG(J),IZERO,ICODE(J,1),
      $ICODE(J,2),IPOP(J),(IEUG(J,K),K=1,32)
      END FILE IO
      CALL RUN (IO)
      IT=IT+1
      IF(IT-IASTIT) 2,2,42
38     CALL REREAD
      READ INPUT TAPE IT,39,(IDUMP(K),K=1,84)
39     FORMAT (84A1)
      WRITE OUTPUT TAPE 3,40,(IDUMP(K),K=1,84)
40     FORMAT (25H TAPE READ ERROR ON INPUT,7IX,84A1)
      WRITE OUTPUT TAPE 3,41,IT,IO,KOUNT1,I
41     FORMAT (19H THE INPUT REEL WAS,I3,21H, THE OUTPUT REEL WAS,I3,21H
      $THE RECORD GROUP WAS,I6,20H, AND THE RECORD WAS,I6,1H,728H THE REC
      $ORD WILL BE SKIPPED.)
      GO TO 16
42     WRITE OUTPUT TAPE 3,43,IDIFR,KOUNT2,KOUNT3
43     FORMAT (8HOFOR THE,I4,12H INPUT REELS,I6,20H CELLS WERE REMOVED,
      $74H AND,I6,30H CELLS WERE GENERATED (ADDED).71H1)
      CALL EXIT
      END

```

SUBROUTINE CELRMV (IFST,ISCND,JUMP)

```
C
C      CELL REMOVER - THIS SUBROUTINE COMBINES TWO CELLS
C      AND DELETES THE ONE WITH THE SMALLER POPULATION.
C
C      DIMENSION LAT(100),LONG(100),IBUG(100,32),IPOP(100),ICODE(100,2)
C      DIMENSION IDIGIT(100)
C      COMMON LAT, LONG, IBUG, IPOP, ICODE, ICHK, IRPLSE, IZERO, IDIGIT
C      COMMON KOUNT2, KOUNT3
C
C      KOUNT2=KOUNT2+1
C      IF(IPOP(IFST)-IPOP(ISCND)) 2,1,1
C
C      FIRST CELL IS DOMINANT.
C
C      1      IPOP(IFST)=IPOP(IFST)+IPOP(ISCND)
C      LAT(ISCND)=99999
C      JUMP=2
C      RETURN
C
C      SECOND CELL IS DOMINANT.
C
C      2      IPOP(ISCND)=IPOP(ISCND)+IPOP(IFST)
C      LAT(IFST)=99999
C      JUMP=1
C      RETURN
C      END
```

```

SUBROUTINE CELGEN (LOC,NEW)
C
C           CELL GENERATOR - THIS SUBROUTINE WILL GENERATE A CELL
C           FROM CITY DATA AND PLACE THAT CITY DATA IN THE
C           SUCCEEDING POSITION.
C
      DIMENSION LAT(100),LONG(100),IBUG(100,32),IPOP(100),ICODE(100,2)
      DIMENSION IDIGIT(100)
      COMMON LAT, LONG, IBUG, IPOP, ICODE, ICHK, IRPLSE, IZERO, IDIGIT
      COMMON KOUNT2, KOUNT3
      COMMON ISPACE
C
      KOUNT3=KOUNT3+1
      NEW=LOC+1
      LAT(NEW)=LAT(LOC)
      IDIGIT(NEW)=IDIGIT(LOC)
      LONG(NEW)=LONG(LOC)
      ICODE(NEW,1)=ICODE(LOC,1)
      ICODE(NEW,2)=ICODE(LOC,2)
      ICODE(LOC,1)=IRPLSE
      ICODE(LOC,2)=IZERO
      IPOP(NEW)=IPOP(LOC)
      IPOP(LOC)=0
      DO 1 J=1,32
        IBUG(NEW,J)=IBUG(LOC,J)
        IBUG(LOC,J)=ISPACE
1      CONTINUE
      NEW=NEW+1
      RETURN
      END

```


DISCRD

Purpose

The DISCRD program produces a plot tape of discontinuities in the existing data base. The plot tape is input to a Mercator plotting routine, developed by Sandia Corporation, which plots the points over a world map. DISCRD is written in FORTRAN II for the IBM 7090.

Input and Output

Input must be the BCD tapes containing the sorted cell and city data, and one card containing, in Col. 1, the number of input tapes.

The output consists of a BCD tape containing in each record the latitude and longitude in degrees of a point to be plotted. The record format is 2F7.3, and the last record contains 14-9s to signify the end of data.

Program Description

Refer to the flowchart for procedure. DISCRD searches each band of records on the tapes, in the order in which they occur, for discontinuities in cell data. A discontinuity occurs if two consecutive data cells in a band are not geographically adjacent to each other. The discontinuity is the gap between the bounding cells, whose center points are recorded to indicate that there are no existing data cells between them.

Discontinuities may take many forms. The longest possible is an empty band, which cannot be recorded since it has no bounding cells. A band containing only one data cell will have only one discontinuity with a single bound. If a cell bounds two discontinuities, that cell is recorded only once. If a discontinuity exists between the last data cell and the first

data cell in a band, the eastern bound will be the first cell in the bounds list, and the western bound of the discontinuity will be the last cell in the list. All other discontinuities will lie between those two on the list, since all bounds are listed in order of increasing longitude.

Any city records on the BCD tapes are counted as records read, but are ignored in the discontinuity search. Two counters are necessary to count the total records read, due to the maximum integer limit of 131071 in the IBM 7090. Only one counter is maintained for the discontinuity bounds since the probability of exceeding 131071 points is low.

After completing the plot tape, DISCRD lists each of the bounds to provide a cross reference for the final plot.

Messages Printed

When a parity error occurs, DISCRD prints the bad record along with the following message:

PARITY ERROR, RECORD IS $131071 \times (X) + \gamma$

where X = the number of times 131071 records have been read and γ = the excess records over 131071.

When there is only one record in a band the following message is printed:

CELL X, γ IS THE ONLY CELL IN ITS BAND

where X = latitude and γ = the longitude.

When the plot tape is completed, the following message is printed:

THERE ARE X POINTS TO BE PLOTTED.
FOLLOWING ARE DISCONTINUITY BOUNDS
XXX.XXX YYY.YYY

where XXX.XXX = latitude, and YYY.YYY = longitude of each plot point.

Flow Chart for Discontinuity Recorder--DISRCD

Discontinuity Recorder (Continued)

Discontinuity Recorder (Continued)

Discontinuity Recorder (Concluded)

C ***** DISCONTINUITY RECORDER *****

C
C

DIMENSION XLONG(720),BUF(1440),ICATCH(84)

IT=5

KIT=1

INDEX=1

KOUNT=3

NKOUNT=0

NUMB=0

MNUMB=0

LATFLG=0

LASFLG=0

READ INPUT TAPE 2,1, LAST

FORMAT(I1)

1

C

IT REPRESENTS TAPE UNIT FOR READ, 13 FOR WRITE.

C

C

GO TO 10

2

NUMB=NUMB+1

IF(NUMB-131071) 4,3,3

3

MNUMB=MNUMB+1

NUMB=0

4

IF(NBUF(IT))5,11,8

C

C

PARITY ERROR

C

C

CALL REREAD

5

READ INPUT TAPE IT, 6, (ICATCH(I),I=1,84)

FORMAT(84A1)

6

WRITE OUTPUT TAPE 3,7,(ICATCH(I),I=1,84),MNUMB,NUMB

FORMAT(1H0,84A1/34H PARITY ERROR, RECORD IS 131071 X ,I4,3H + ,I6)

7

GO TO 2

C

C

END OF FILE IT. MOUNT NEXT TAPE FOR READ, UNTIL LAST TAPE IS READ.

C

C

CALL RUN(IT)

8

LASFLG=1

IF(KIT-LAST)9,17,9

C

C

IT=IT+1

9

KIT=KIT+1

REWIND IT

10

GO TO 2

CALL REREAD

11

READ INPUT TAPE IT,12,XLAT,XLON,NTYPE

FORMAT(2F6.3,1X,I1)

12

C

IF FIRST CELL THIS LATITUDE BELT, SET FLAGS

C

C

IF(LATFLG)14,13,14

ALAT=XLAT

LATFLG=1

13

C

IF CELL IMAGE, CHECK LATITUDE. OTHERWISE GET ANOTHER IMAGE.

C

C

C


```

14  IF(NTYPE)2,15,2
C
C  IF SAME LATITUDE, MOVE LONGITUDE TO XLONG(INDEX).
C  OTHERWISE, PROCESS THIS LATITUDE BELT.
C
15  IF(ALAT-XLAT)17,16,17
16  XLONG(INDEX)=XLON
    INDEX=INDEX+1
    GO TO 2
C
C  PROCESS THIS LATITUDE BELT.
C
17  NDEX=INDEX
    XLONG(NDEX)=XLONG(1)+360.000000
    LONFLG=0
    INDEX=1
    IF=NDEX-2)18,26,18
18  IF(NDEX-INDEX)19,27,19
C
C  CHECK FOR DISCONTINUITY
C
19  IF(XLONG(INDEX)+.50-XLONG(INDEX+1))20,25,20
C
C  THERE IS A DISCONTINUITY.  SAVE XLONG(INDEX), AND XLONG(INDEX+1).
C
C  IS THIS THE FIRST CELL THIS BELT.
C
20  IF(LONFLG)23,21,23
C
C  IT IS.
C
21  LONFLG=1
22  BUF(KOUNT)=ALAT
    BUF(KOUNT+1)=XLONG(INDEX)
    BUF(KOUNT+2)=ALAT
    BUF(KOUNT+3)=XLONG(INDEX+1)
    KOUNT=KOUNT+4
    GO TO 25
C
C  IS THIS DISCONTINUITY ADJACENT TO THE LAST ONE.
C
23  IF(BUF(KOUNT-1)-XLONG(INDEX))22,24,22
C
C  IT IS.
C
24  BUF(KOUNT)=ALAT
    BUF(KOUNT+1)=XLONG(INDEX+1)
    KOUNT=KOUNT+2
25  INDEX=INDEX+1
    GO TO 18
26  BUF(KOUNT)=ALAT
    BUF(KOUNT+1)=XLONG(1)
    KOUNT=KOUNT+2
    WRITE OUTPUT TAPE 3,39,ALAT,XLONG

```

```

C
C   BUF CONTAINS ALL DISCONTINUITIES OF THIS BELT.
C
27   KOUNT=KOUNT-1
      TEMP=BUF(KOUNT)=360.0000000
      IF(TEMP)31,28,28
28   BUF(KOUNT)=TEMP
      IF(BUF(KOUNT)-BUF(4))29,30,30
29   BUF(1)=ALAT
      BUF(2)=BUF(KOUNT)
      BUF(2)=BUF(KOUNT)
      NN=1
      NKOUNT=KOUNT/2+NKOUNT-1
      KOUNT=KOUNT-2
      GO TO 33
30   NKOUNT=KOUNT/2+NKOUNT-2
      KOUNT=KOUNT-2
      GO TO 32
31   NKOUNT=KOUNT/2+NKOUNT-1
32   NN=3
33   DO 35 N=NN,KOUNT,2
      WRITE OUTPUT TAPE 13,34,BUF(N),BUF(N+1)
34   FORMAT(2F7.3)
35   CONTINUE
      LATFLG=1
      KOUNT=3
      INDEX=1
      ALAT=XLAT
      IF(LASFLG)16,16,36
C
C
C   LAST TAPE HAS BEEN READ.
C
36   MAGIC=99
C
      WRITE OUTPUT TAPE 13,37,MAGIC,MAGIC,MAGIC,MAGIC,MAGIC,MAGIC,MAGIC
37   FORMAT(7I2)
      END FILE 13
      WRITE OUTPUT TAPE 3,38,NKOUNT
38   FORMAT(11H0THERE ARE ,I6,23H POINTS TO BE PLOTTED.)
39   FORMAT(5H CELL ,F6.3,1X,F6.3,30H IS THE ONLY CELL IN ITS BAND.)
      REWIND IT
      REWIND 13
      WRITE OUTPUT TAPE 3,43
      DO 41 I=1,NKOUNT
      READ INPUT TAPE 13,40,DLAT,DLON
40   FORMAT(2F7.3)
C
      WRITE OUTPUT TAPE 3,42,DLAT,DLON
41   CONTINUE
42   FORMAT(11H ,2F7.3)
43   FORMAT(11H1,35HFOLLOWING ARE DISCONTINUITY BOUNDS.)
      CALL RUN(13)
      CALL EXIT
      END

```

REFORM

Purpose

This program reformats the population studies cell and city data contained in sorted card-image form on several BCD tapes into one file per band (3 buffered binary records per file) on a binary tape.

Input and Output

The input consists of: One card containing in Col. 1 the number of input reels (number = NUMB), and NUMB BCD tapes containing sorted cell and city data in card-image records.

The output consists of a binary tape containing 540 files, one latitude band per file, three buffered binary records per latitude band. Record 1 of each file is a file label containing three words: latitude of center of band, number of words in the cell record, and number of words in the city record. Record 2 is the cell record, five words per cell. Word 5 contains two keys to the location in the city record of the cities for that cell. Record 3 is the city record, three words per city. (See the section Data Structure, p. 3).

Program Description

Flowcharts following this subsection of text illustrate the following program description. REFORM determines the number of tapes to be read by reading a data card. The first device to read from is 5, and the last is (NUMB + 5 - 1). The BCD records are read and processed in groups of 1000. The program processes latitude bands in order from 0167 to 179833. If no BCD record exists for a particular band, a dummy file is written, containing only a file record--word 1 is the band label, and words 2 and 3 are zero.

If the latitude of the current card image is the same as that of the next band to be written, the program tests the cell-city key--an integer 0 indicates a cell image and a 1 indicates a city image. A cell image is stored in the cell-output area in the following format:

word	1:	longitude (integer)
	2:	cell population (integer)
	3:	first 8 characters of country code (alphameric)
	4:	(packed alphameric 8 characters, listed left to right)
		last 2 characters of country code
		F or S
		blank
		date
	5:	(packed integer)
		location in city record of first city word (left half of word)
		number of city words for this cell (right half of word)

A city image is stored in the city-output area in the following format:

word	1:	city population (integer)
	2:	city area (integer)
	3:	date (alphameric, left half of word)

When all records for a particular band have been moved to the output area, the output subroutine NOUT is called. NOUT writes the file for the band and returns control to the main program. REFORM computes the next latitude label and continues processing card images. If all card images are processed and 540 files have not been written, enough dummy files are written to complete the tape.

Printed Messages

REFORM

1. NOW READING ON DEVICE _____
Prints device number to aid in finding source of any program failure. The device number less 4 is the sequence order of the tape in process.
2. PARITY ERROR IN IMAGE SUCCEEDING _____
Prints card image preceding the one in which a parity error was detected. The bad image is ignored.
3. PROCESSING IMAGE GROUP _____
Prints number of the image group just read. Images are read in groups of 1000.
4. ERROR IN BAND ORDER. BAND IN PROCESS IS
_____ CELL OR CITY LATITUDE IS _____
RECORD IS _____
Prints latitude in process, then the out-of-order latitude and the remainder of the image. The image is ignored. Note that this error check requires the BCD tapes to be processed in ascending latitude order.
5. BINARY TAPE WRITTEN
This is the end-of-job message.

NOUT

1. NO CELLS WERE FOUND IN LATITUDE _____
FOR THESE CITY DATA _____

This condition should not occur, but if it does, the latitude and the city words are printed. The city words are also written on the tape but cannot be recovered meaningfully.

Flow Chart for Reformat Binary Tape--REFORM

REFORM (Continued)

REFORM (Continued)

REFORM (Concluded)

```

C *****REFORMAT DATA AND WRITE BINARY TAPE
C
C THE FOLLOWING ASSUMPTIONS ARE MADE--
C 1. LATITUDE BANDS ARE ORDERED FROM SOUTH TO NORTH,
C 0167 TO 179833.
C 2. EVERY CITY IS PRECEDED BY A CORRESPONDING CELL.
C 3. NO BAND HAS MORE THAN 3000 CITIES
C
C COMMON /CELL/LONTAB(720,5)/CITY/ICITTAB(3000,3)
C DIMENSION NB(1000,11)
C
C INITIALIZE COUNTERS
C NLAT IS NUMBER OF BANDS
C NLON IS NUMBER OF CELLS
C NCIT IS NUMBER OF CITIES
C LATLAB IS THE BAND LABEL
C
C NLAT=1$NLON=0$NCIT=0$LATLAB=0167$TEND=0
C
C -IT- IS THE TAPE UNIT TO READ, I3 TO WRITE.
C START WITH IT=5. -NUMB- IS THE NUMBER OF TAPES TO BE READ
C
C REWIND 13
C READ 1,NUMB
1 FORMAT (I1)
C IT=5
C REWIND IT
C KOUNT=0
C
C READ 1000 IMAGES FROM BCD TAPE, AND CHECK FOR EOF OR ERROR
C
C I=1
2 READ (IT,3) (NB(I,J),J=1,11)
3 FORMAT (2I6,A1,I1,1X,I5,A8,A2,A4,1X,I5,I4,A4)
C IF(EOF,IT) 4,8
4 IF(IT-NUMB-4) 6,5,5
5 IEND=1
C REWIND IT
C GO TO 12
6 REWIND IT
C IT=IT+1$I=I+1
C PRINT 7,IT
7 FORMAT (*NOWREADINGONDEVICE*,I3)
C GO TO 11
8 IF(I0CHECK,IT) 9,11
9 I=I+1
C PRINT 10,(NB(I,J),J=1,11)
10 FORMAT (34HOPARITY ERROR IN IMAGE SUCCEEDING ,2I6,A1,I1,1X,I5,A8
C $,A2,A4,1X,I5,I4,A4)
11 I=I+1$IF(I-1000)2,2,11
C
C PROCESS ALL COMPLETE BANDS IN THIS 1000 IMAGES.
C
C
12 NN=I-1

```

```

      KOUNT=KOUNT+1
      PRINT 13,KOUNT
13     FORMAT(*PROCESSING IMAGE GROUP*,I6)
C-----
C           FIND THE LATITUDE BAND, WRITING A DUMMY BAND FOR EACH
C           MISSING LATITUDE.
C-----
14     IF(NBCD(N,1)-LATELAB) 15,17,24
C-----
C           THIS IMAGE IS OUT OF ORDER. IT WILL NOT APPEAR ON THE BINA
C-----
15     PRINT 16,LATELAB,NBCD(N,1),(NBCD(N,J),J=2,11)
16     FORMAT (41H0ERROR IN BAND ORDER. BAND IN PROCESS IS ,I6,26H CELL O
$R CITY LATITUDE IS ,I6,11H RECORD IS ,I6,A1,I1,I6,A8,A2,A4,I6,I5
$,A4)
      GO TO 22
C-----
C           IS THIS A CELL IMAGE OR A CITY IMAGE
C-----
17     IF(NBCD(N,4)) 18,18,23
C-----
C           PROCESS A CELL IMAGE
C           LONTAB CONTAINS CELL (LONGITUDE) INFORMATION
C-----
18     NLON=NLON+1
      LONTAB(NLON,1)=NBCD(N,2)
      LONTAB(NLON,2)=NBCD(N,5)
      LONTAB(NLON,3)=NBCD(N,6)
      ENCODE(8,16,LONTAB(NLON,4),NBCD(N,7),NBCD(N,3),NBCD(N,8))
C-----
19     FORMAT (A2,A2,A4)
      LONTAB(NLON,5)=NCIT
      ASSIGN 22 TO IEXIT
      IF(NLON-1) 22,22,20
C-----
20     NEMP=(NCIT-LONTAB(NLON-1,5))*3
      IEMP=LONTAB(NLON-1,5)*3+1
      ENCODE(8,18,LONTAB(NLON-1,5))TEMP,NEMP
C-----
21     FORMAT (I4,I4)
      GO TO IEXIT
22     N=N+1
      IF(N-NN) 14,14,26
C-----
C           PROCESS A CITY IMAGE
C           ICITTAB CONTAINS CITY INFORMATION
C-----
23     NCIT=NCIT+1
      ICITTAB(NCIT,1)=NBCD(N,9)
      ICITTAB(NCIT,2)=NBCD(N,10)
      ICITTAB(NCIT,3)=NBCD(N,11)
      GO TO 22
C-----
C           OUTPUT THIS BAND
C-----

```

```

C
24  NLON=NLON+1
    ASSIGN 25 TO IEXIT
    IF (NLON-1) 25,25,20
25  CALL NOUT (LATLAB,NLON,NCIT)
C
C      ZERO CELL AND CITY COUNTERS
C      ADD ONE TO BAND COUNTER
C
    NLON=0$NCIT=0$NLAT=NLAT+1
C
C      CALCULATE NEXT BAND LABEL
C
    LATLAB=(NLAT+2)/3+NLAT*333-167
    GO TO 14
26  IF (IEND) 27,27,28
27  I=1
    GO TO 2
28  ASSIGN 29 TO IEXIT
    NLON=NLON+1
    IF (NLON=1.GT. 0) GO TO 20
    CALL NOUT (LATLAB,NLON,NCIT)
29  IF (NLAT .GE. 540) GO TO 30
    NLAT=NLAT+1
    LATLAB=(NLAT+2)/3+NLAT*333-167
    NLON=1
    NCIT=0
    GO TO 29
30  END FILE 13
    REWIND 13
    PRINT 31
31  FORMAT(*ORDINARY TAPE WRITTEN*)
    STOP
    END

```

```

C
C      SUBROUTINE NCUT (LATLAB,NLON,NCIT)
C
C      LATLAB IS THE BAND LABEL
C      NLON IS THE NUMBER OF CELLS
C      NCIT IS THE NUMBER OF CITIES
C
C
C      WRITES 1, 2, OR 3 BUFFERED BINARY RECORDS, AND AN
C      END FILE MARK ON TAPE UNIT 13.
C
C      COMMON /CELL/LONTAB(720,5)/CITY/ICITTAB(3000,3)
C      COMMON /LABEL/IOUT(12604)
C
C      LENGTHCL IS THE NUMBER OF WORDS IN THE CELL RECORD SECTION
C      LENGTHSG IS THE NUMBER OF WORDS IN THE CITY RECORD SECTION
C
C      NLON=NLON-1
C      LENGTHCL=NLON*5
C      LENGTHSG=NCIT*3
C
C      IOUT(1)=LATLAB
C      IOUT(2)=LENGTHCL
C      IOUT(3)=LENGTHSG
1      IF(UNIT,13) 1,2
C
2      BUFFER OUT(13,1)(ICUT(1),IOUT(3))
      IF(LENGTHCL) 3,3,6
3      IF(LENGTHSG) 15,15,4
4      LAT=LATLAB-90000
C
      PRINT 5,LAT,((ICITTAB(I,J),J=1,3),I=1,NCIT)
5      FORMAT (1H0,32H NO CELLS WERE FOUND IN LATITUDE,17,20H FOR THESE C
      CITY DATA,2X,2I6,X,A47(62X,2I6,X,A4))
      GO TO 10
C
6      DO 7 I=1,NLON
      DO 7 J=1,5
      N=(I-1)*5+J
      IOUT(N)=LONTAB(I,J)
7      CONTINUE
8      IF(UNIT,13) 8,9
C
9      BUFFER OUT(13,1)(IOUT(1),IOUT(LENGTHCL))
10     IF(LENGTHSG) 15,15,11
11     DO 12 I=1,NCIT
      DO 12 J=1,3
      N=(I-1)*3+J
      IOUT(N)=ICITTAB(I,J)
12     CONTINUE
13     IF(UNIT,13) 13,14
C
14     BUFFER OUT(13,1)(IOUT(1),IOUT(LENGTHSG))
15     IF(UNIT,13) 15,16
16     END FILE 13
      RETURN
      END

```

POPRET

Purpose

The POPRET subroutine is used to retrieve the population data stored in binary form on a single tape containing 540 files--one file per latitude band. The data may be retrieved one cell at a time or one band at a time. POPRET is written in FORTRAN IV for the CDC 3600.

Input

A buffered binary tape on LUN 49 is the only input required. The format of the buffered binary tape is designed for efficient handling of variable-length records. One latitude band may require three, two, or one record. Therefore, each band is contained in a separate file to allow faster tape searching.

The first record in each file is a label record containing three items (words) of information: The first word is the latitude of the centers of the cells in the band, which varies from 0167 ($\times 10^3$) deg at the South Pole (beginning of the tape) to 179833 ($\times 10^{-3}$) deg at the North Pole (end of the tape). The second word is the number of words in the second record, which contains all of the band's cell data, five words per cell. The third word of the label record is the number of words in the last record of the file, which contains all the band's city data, three words per city.

The cell data record and the city data record are in longitude order from 0250 ($\times 10^{-3}$) deg at the first cell east of Greenwich to

359750 ($\times 10^{-3}$) deg at the first cell west of Greenwich. The data packed for each cell are, in order: longitude, residual population, country code, F or S, date, location in city record of first city for the cell relative to first city of the band, and number of city words for the cell. The data for each city are, in order: population, area, date. A "city" is that part of any city that lies within the boundaries of the cell in question.

Calling Sequence

CALL RETCELL (LAT, LON) or CALL RETBAN (LAT, LON).

RETCELL is the entry point for retrieving one cell and its associated cities, and RETBAN is the entry point for retrieving the contents of an entire band which is received as a BCD tape file. LAT is the latitude of the band requested; LON is the longitude of the specific cell required, or of the first cell in the band. For convenience to the user, LAT must be specified in the range -89833 (S) to +89833 (N) and LON in the range 0250 to 359750. Each time POPRET is executed the tape is searched and all data for the requested band is stored in COMMON in the / LABEL/ block.

COMMON Storage (labeled)

Blocks

```

/CELL/ILAT, ILON, IRPOP, IRCODE(2), IFORS, IRDATE
/CITY/LENGTH, ICPOP(1000), ICAREA(1000), ICDATE(1000)
/LABEL/INFILE(12604)

```


Variables

ILAT - latitude of band, S to N, -89833 to +89833

ILON - longitude of requested cell, E to W, 0250 to 359750

IRPOP - population of requested cell in hundreds

IRCODE(2) - country code of requested cell; word 1 contains the first eight characters, word 2 contains the remaining two characters in the leftmost twelve bits

IFORS - an F or S in the leftmost six bits

IRDATE - month and year of cell data in the leftmost 24-bits (always alphameric)

LENGTH - number of cities associated with this cell

ICPOP(I) - populations ($\times 10^3$) of I [I/I = 1, 2, ... LENGTH] cities

ICAREA(I) - areas ($\times 10^2$) of I cities

ICDATE(I) - month and year of I cities data (always alphameric)

INFILE - complete latitude band record: word 1 contains the latitude (0167-179833); word 2 contains 3+N where N=number of cell words; word 3 contains 3+N+M where M=number of city words (5 words per cell and 3 words per city). That is, INFILE(2) contains the address relative to INFILE(1) of the last cell word, and INFILE(3) contains the address relative to INFILE(1) of the last city word. Thus it is possible to retrieve an entire band and store it in the COMMON block to be manipulated by the main program with only one call to POPRET.

Printed Messages

If an end-of-file mark is read during an attempt to read the second or third record of a file, the following message is printed, the file is dumped, variables in the common block are set, and control returns to the main program:

END-OF-FILE FOUND INSTEAD OF EXPECTED XXXX
RECORD FOR YYYYYY LAT, OCTAL DUMP OF THIS
FILE FOLLOWS.

xxxx = 'CELL' or 'CITY' for an EOF in the second or the third
record respectively.

yyyyyy = the latitude of the requested band.

If the error occurred in the cell record, the variables in the /CELL/ block and the /CITY/ block are each set to zero, with the exception of IRDATE, which will contain, as an alphanumeric date, 9999.

If the error occurred in the city record, all the variables in the /CITY/ block are set to zero, with the exception of LENGTH, which will contain -2.

If a parity error is sensed, LENGTH is set to -1; all remaining /CELL/ and /CITY/ variables are set to zero and the following message is printed:

PARITY ERROR SENSED IN XXXX RECORD FOR BAND
AT YYYYYY LAT, SUGGEST YOU REWRITE TAPE

control returns to the main program.

If no data are found for a requested cell, IRDATE will contain, as an alphanumeric date, 8888* and all remaining /CELL/ and /CITY/ variables are set to 0.

If no city data are found for a requested cell, LENGTH is set to zero along with all other /CITY/ variables.

Output

The BCD tape written on LUN 50 by RETBAN contains card images of all the cells and cities for the requested band. Each cell is followed by its associated cities, and the cells are in order of increasing longitude. An end-of-file mark follows the last cell, and RETBAN rewinds tape 50 before giving control over to the RETCELL entry point. RETCELL then retrieves the first cell in the file as if it alone were requested.

*To check IRDATE for error conditions, the user's calling program will compare IRDATE to 11111111B or 10101010B. Refer to a CDC FORTRAN manual for clarification.

The card image formats are:

Cell - (2I6, A1, I1, I6, A8, A2, A4)
latitude, longitude, F or S, 0, residual population ($\times 10^2$), country codes, date.

City - (35X, I5, I4, A4)
blanks in portion corresponding to cell usage,
city population ($\times 10^3$), city area ($\times 10^2$), date

The user may copy each band from the BCD tape to a second tape before requesting the next band if he wishes to save the card images, or he may remove the REWIND and END FILE statements from RETBAN and produce his own set of BCD tapes.

To produce a printer listing of the data in card image form, the user may convert RETBAN to a lister by replacing the tape device number (50) with a printer device number, and removing all tape handling statements (perhaps by converting the statements to comments). A call to RETBAN will then produce the listing exactly as it would have produced a BCD tape.

Flow Chart for Population Retriever Subroutine--POPRET

POPRET Subroutine (Continued)

POPRET Subroutine (Continued)

POPRET Subroutine (Continued)

POPRET Subroutine (Concluded)

SUBROUTINE POPRET (LAT, LON)

AS MANY AS 1000 CITIES PER CELL OR 3000 CITIES PER BAND
MAY BE RETRIEVED.
LATITUDE BELTS ON THE TAPE ARE LABELED 167 TO 179833
AND ARE RETRIEVED AS -89833 TO +89833.
REQUIRES A BINARY TAPE ON UNIT 49.

UPON RETURN OF
CONTROL TO THE
CALLING PROGRAM
THE COMMON BLOCK
LABELED /CELL/
CONTAINS EITHER THE
REQUESTED CELL'S
DATA, OR DIAGNOSTICS,
OR (FOR RETBELT) THE
DATA OR DIAGNOSTICS
OF THE FIRST CELL
CONTAINING DATA.

THE /CITY/ COMMON
BLOCK CONTAINS
ALL CITY DATA FOR
THE CELL IN /CELL/
OR A DIAGNOSTIC FOR
THAT CELL.

THE /LABEL/ BLOCK
CONTAINS THE ENTIRE
FILE FOR THE BAND
REQUESTED.

COMMON /CELL/ ILAT, ILON, IRPOP, IRCODE(2), IFORS, IRDATE
COMMON /CITY/ LENGTH, ICPOP(1000), ICAREA(1000), ICDATE(1000)
COMMON /LABEL/ INFILE(12604)
EQUIVALENCE (INFILE(1), LATLAB), (INFILE(2), LENGTHCL), (INFILE(3),
LENGTHSG)

THE VARIABLES
IRCODE, IFORS,
IRDATE, ICDATE
CONTAIN
ALPHAMERIC DATA
AT ALL TIMES.

DATA (NTAPE=0)

FIND THE LATITUDE FILE AND READ THE FIRST RECORD.

ENTRY RETCELL

```

1      IBELTFLG=0
      IF (NTAPE .NE. 0) GO TO 2
      REWIND 49
      NTAPE=1
      IOCT=0
2      NLAT=LAT
      LOOP=0
      LAT=LAT+90000
      IF (LAT .LT. 0167 .OR. LAT .GT. 179833) GO TO 30
      INDEX=(LAT+167)/333
      CALL SKIPFILE (49)
      BUFFER IN(49,1)(INFILE(1),INFILE(3))
3      CONTINUE
      LOOP=LOOP+1
      IF (LOOP .GT. 10) GO TO 30
4      IF (UNIT,49) 4,5,6,37
5      NDEX=(LATLAB+167)/333;N=IABS(INDEX-NDEX);IF (LAT-LATLAB) 61,101,91
C      POSITION TAPE AT BEGIN-
C      NING OF FILE FOR WHICH
C      EOF WAS FOUND
6      CALL BACKFILE (49)
      CALL BACKFILE (49)
      CALL SKIPFILE (49)
      GO TO 3
61     IF (INDEX=NDEX/4) 7,8,8
7      REWIND 49
      GO TO 3
C      BACK TAPE TO REQUESTED FILE
8      DO 9 J=1,N
      CALL BACKFILE (49)
9      CONTINUE
      CALL BACKFILE (49)
      IF (INDEX .GT. 1) CALL SKIPFILE (49)
      GO TO 3
C      SKIP TAPE FORWARD TO REQUESTED FILE
91     DO 10 J=1,N
      CALL SKIPFILE (49)
10     CONTINUE
      GO TO 3
C
C      IF LENGTHCL IS NOT GREATER THAN 0, THEN THIS BAND HAS NO
C      CELLS. OTHERWISE, READ THE CELL RECORD
C      FOR THIS BAND. IF LON IS LESS THAN THE LONGITUDE OF A CELL
C      IT IS COMPARED TO, OR IF IT IS GREATER THAN THAT OF THE
C      LAST CELL, THEN THERE IS NO CELL FOR LON.
C
101    ILAT=NLAT
      ILON=LON
      IF (LENGTHCL) 19,19,11
11     LENGTHCL=LENGTHCL+3
      NCELL=4HCELL
      BUFFER IN(49,1)(INFILE(4),INFILE(LENGTHCL))
      NCEL=1
12     IF (UNIT,49) 12,16,13,37
13     PRINT 14,NCELL,ILAT

```

```

14  FORMAT(*1END=OF-FILE FOUND INSTEAD OF EXPECTED*A4* RECORD FOR,*,
      $I8,1X,*LAT*,4X,*OCTAL DUMP OF THIS FILE FOLLOWS.*)
      NUMBWRD=LENGTHH(49)
      NUMBWRD=NUMBWRD+3
      PRINT 15,(INFILE(I),I=1,NUMBWRD)
15  FORMAT (4X,6(016,4X))
      IOCT=1
      GO TO (30,33),NCEL
16  IF(LENGTHSG) 19,19,17
17  L=LENGTHCL+1
      NCELL=4HCITY
      LENGTHSG=LENGTHSG+LENGTHCL
      NCEL=2
      BUFFER IN(49,1)(INFILE(L),INFILE(LENGTHSG))
18  IF(UNIT,49) 18,19,13,37
19  NCEL=0
      IF (IBELTFLG .EQ. 1) GO TO 40
      IF (LENGTHCL .LE. 0) GO TO 31
20  DO 23 I=4,LENGTHCL,5
      IF(LON-INFILE(I)) 31,21,23
21  IRPOP=INFILE(I+1)
      IRCODF(1)=INFILE(I+2)
      DECODE(8,22,INFILE(I+3))IRCODE(2),IFORS,IRDATE
22  FORMAT (A2,A2,A4)
      LTEMP=INFILE(I+4)
      GO TO 24
23  CONTINUE
      GO TO 31

C
C      IF LENGTHSG IS NOT GREATER THAN ZERO, THERE ARE NO CITIES
C      IN THIS BAND. OTHERWISE LTEMP CONTAINS THE LOCATION
C      (RELATIVE) OF THE FIRST CITY WORD (=KK), AND THE
C      NUMBER OF CITY WORDS FOR THIS CELL (=LENGTH).
C      IF LENGTH IS GREATER THAN ZERO, READ THE CITY RECORD
C      FOR THIS CELL.
C
24  IF(LENGTHSG) 33,33,25
C
C      DECODE LTEMP. RIGHTMOST 24 BITS = NUMBER OF CITY WORDS IN CELL
C      (=LENGTH.) LEFTMOST 24 BITS = LOCATION OF FIRST CITY WORD
C      OF THIS CELL RELATIVE TO FIRST CITY WORD OF THIS LATITUDE(=KK.)
C
25  DECODE(8,26,LTEMP)KK,LENGTH
C
26  FORMAT (I4,I4)
      IF(LENGTH) 33,33,27
27  KK=LENGTHCL+KK
      KL=LENGTH+KK-1
      L=0
      DO 28 K=KK,KL,3
      L=L+1
      ICPOP(L)=INFILE(K)
      ICAREA(L)=INFILE(K+1)
      ICDATF(L)=INFILE(K+2)
28  CONTINUE

```

```

      LENGTH=LENGTH/3
      LAT=NLAT
      L=L+1
      DO 29 M=L,1000
      ICPOP(M)=0
      ICAREA(M)=0
      ICDATE(M)=0
29    CONTINUE
      GO TO 39
C
C      LATITUDE BAND IS MISSING. SET CELL DATE=9999(ALPHAMERIC),
C      CODES = 0, CITY LENGTH = 0.
C
30    IRDATE=11111111B
      GO TO 32
C
C      CELL IS MISSING. SET CELL POP, CODES, CITY LENGTH = 0,
C      CELL DATE = 8888(ALPHAMERIC).
C
31    IRDATE=10101010B
32    IRPOP=0
      IRCODE(1)=0
      IRCODE(2)=0
      IFORS=0
C
C      CITIES ARE MISSING. SET CITY LENGTH = 0.
C
33    LENGTH=0
      DO 34 I=1,1000
      ICPOP(I)=0
      ICAREA(I)=0
      ICDATE(I)=0
34    CONTINUE
      IF(NCEL-2) 36,35,36
C      CITIES INCOMPLETE DUE TO RECORD ERROR
C      SET LENGTH = -2
35    LENGTH=-2
36    LAT=NLAT
      GO TO 39
C
C      PARITY ERROR. SET LENGTH=-1.
C
37    LENGTH=-1
      LAT=NLAT
      PRINT 38,NCELL,ILAT
38    FORMAT(1H1,24H PARITY ERROR SENSED IN A4* RECORD FOR BAND AT*
      $,I6,2X,*LAT SUGGEST YOU REWRITE BINARY TAPE.*)
39    CALL BACKFILE (49)
      RETURN
      ENTRY RETBELT
      REWIND 50
      IBELTFLG=1
      GO TO 1
40    IF (LENGTHCL .LE. 0) GO TO 45
      DO 44 I=4,LENGTHCL,5

```

```

-----
DECODE(8,22,INFILE(I+3))NCONTRY2,NFORS,NRDATE
C
WRITE (50,41) NLAT,INFILE(I),NFORS,C,INFILE(I+1),INFILE(I+2)
S,NCONTRY2,NRDATE
41  FORMAT (2I6,A1,I1,I6,A8,A2,A4)
DECCDE(8,26,INFILE(I+4))NK,NLENGTH
C
IF (NLENGTH .LE. 0) GO TO 44
NK=LENGTHCL+NK
NL=NLENGTH+NK-1
DO 43 K=NK,NL,3
WRITE (50,42) INFILE(K),INFILE(K+1),INFILE(K+2)
42  FORMAT (35X,I5,I4,A4)
43  CONTINUE
44  CONTINUE
END FILE 50
LON=INFILE(4)
GO TO 20
45  END FILE 50
GO TO 31
END
-----

```

CELLANAL

Purpose

The CELLANAL program analyzes population densities in a given latitude band by cells for each country in the band according to eight density zones, and produces a printed report of the analysis. The population of a cell is considered to be the total cell population and the area is the total cell area. CELLANAL uses the binary-data retrieval subroutine POPRET to retrieve the band data from the binary tape described under Data Structure. CELLANAL is written in FORTRAN IV for the CDC 3600.

Input and Output

The user may specify any number of consecutive latitude bands as a group, defined by the latitudes of the first and last bands in the group. As many as 50 such groups may be analyzed in one pass of the program. The limit pairs are specified on punched cards, one pair per card. The beginning latitude is punched in Cols. 1-10 (right justified) and the ending latitude in Cols. 11-20 (right justified). The latitude limits are specified in south to north range -89833 to +89833. The deck should be followed by a single card containing only the number 99999 in Cols. 1-5 and the date (XX-XX-XX) in Cols. 30-37.

The output is a printed report (see Fig. 11, p. 113) for each latitude band, in matrix form, of:

total cells and percentage of total country area per density
zone per country crossed by the latitude band;

average population density per country;

number of cells per country;

total area per country;

total cells and percentage of total band area per zone, and
total cells and total area per band;

cells and percent of total country area per density zone per
country crossed by the latitude band, average population
density per country, number of cells per country, and total
area per country; as well as total cells and percent total
band area per zone, and total cells and total area per band.

Program Description

Refer to the CELLANAL flowchart (p. 114) for the program logic in
the country, zone, and band analyses.

The program processes the group specified by each pair of lim-
its before proceeding to the next group. The area of any cell in a band
is the same as the area of every other cell in the band. It is computed as
the area of a trapezoid where the height is taken to be 20 min of latitude
(20 n mi) and the base is taken as 30 min of longitude (30 n mi times the
cosine of the latitude of the base). Thus, the formula for the area of a
trapezoidal cell reduces to

$$(300) (\cos \alpha + \cos \beta) \text{ n mi}^2$$

where α is the latitude of base a, and β is the latitude of base b.

The population of a cell is the sum of all city populations and the
residual population. The population density is population divided by area,
and is assigned to one of eight density zones.

People/n mile²

0-1	101-500
2-5	501-1000
6-25	1001-5000
26-100	5001-

The program computes density and zone for each cell in a band and maintains a continuing sum of cells per zone per country, by comparing the country code to each previously discovered country code in the band and adding the current population to the existing total for the country to which the cell belongs, retaining the relative location of that sum as the row index to a matrix of zones per country. The column index is the population density zone number assigned. The data stored in the matrix are the number of cells in each zone for each country.

Using these country totals (after processing every recorded cell in a band), the program computes the total cells in the band for each country, and the total area in the band for each country. The average density for a country is the country's population divided by the total cells in the country. The percentage of total cells in each country assigned to each zone is computed to complete the country analysis. The zone analysis consists of computing total cells in each zone for a band and the percentage of total cells in the band assigned to each zone. The band analysis is merely to compute the total recorded area in the band.

POPULATION ANALYSIS FOR LATITUDE BELT 548J3, ACCORDING TO ZONES OF POPULATION PER SQUARE MILE, 04-21-68

COUNTRY CODE	ZONE 1 0-1		ZONE 2 2-5		ZONE 3 6-25		ZONE 4 26-100		ZONE 5 101-500		ZONE 6 501-1000		ZONE 7 1001-5000		ZONE 8 5001+		AVERAGE POPULATION DENSITY	TOTAL COUNTRY CELLS AREA	
	CELLS	%AREA	CELLS	%AREA	CELLS	%AREA	CELLS	%AREA	CELLS	%AREA	CELLS	%AREA	CELLS	%AREA	CELLS	%AREA		CELLS	AREA
GYM3EU	5	62.50	0	0.00	0	0.00	1	12.50	2	25.00	0	0.00	0	0.00	0	0.00	79.03	8	2764.60
DEN3EU	0	0.00	0	0.00	1	16.67	2	33.33	3	50.00	0	0.00	0	0.00	0	0.00	128.05	6	2073.45
POR6SX	9	100.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0.00	9	3110.18
URR6SX	113	40.50	12	4.30	39	13.98	73	26.16	33	11.83	7	2.51	2	0.72	0	0.00	70.95	279	96415.50
USA4NA	18	100.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0	0.00	0.00	18	6220.36
CNA4NA	127	94.07	5	3.70	2	1.48	1	0.74	0	0.00	0	0.00	0	0.00	0	0.00	0.64	135	46652.66
EIM3EU	7	77.78	0	0.00	0	0.00	1	11.11	1	11.11	0	0.00	0	0.00	0	0.00	18.52	9	3110.18
UKW3EU	3	20.00	0	0.00	0	0.00	3	20.00	7	46.67	0	0.00	2	13.33	0	0.00	375.97	15	5183.63
TOTALS	282	58.87	17	3.55	42	8.77	81	16.91	46	9.60	7	1.46	4	0.84	0	0.00		479	165530.56

Fig. 11--Sample Output from CELLANAL

Flow Chart for Cell Analysis--CELLANAL

CELLANAL (Continued)

CELLANAL (Concluded)

```

      DIMENSION LIMBEG(50),LIMEND(50),CONTRY(3,50),ZONE(50,8)
      S,TOTZON(50),TOTAREA(50),AVGDEN(50),ZONTCT(8),ZONPER(8),FZONE(50,8)
      COMMON /CELL/IR(6),IRDATE
      COMMON /CITY/LENGTH,IC(3000)
      COMMON /LABEL/INFILE(12604)
      EQUIVALENCE (INFILE(1),LATLAB),(INFILE(2),LENCL),(INFILE(3),LENSG)
      INTEGER CONTRY,ZONE,TOTZON,ZONTOT
C      READ PAIRS OF LIMITS, 1 PAIR PER CARD, UNTIL END CARD
C      IS RECOGNIZED. END CARD MUST HAVE 99999 IN COLS. 1-5,
C      AND BLANKS IN COLS. 6-10, DATE XX-XX-XX IN COLS. 30-37
      NUMLIM=0
      DO 1 I=1,50
      READ 37,LIMBEG(I),LIMEND(I),IRDATE
      IF (LIMBEG(I).EQ. 9999900000) GO TO 2
1      NUMLIM=NUMLIM+1
C      ADD 90000 TO LIMITS TO CONVERT LATITUDES TO
C      POSITIVE ANGLES
2      DO 36 I=1,NUMLIM
      LIMBEG(I)=LIMBEG(I)+90000
      LIMEND(I)=LIMEND(I)+90000
      IF (LIMBEG(I).LE. LIMEND(I)) GO TO 3
      LTEMP=LIMBEG(I)
      LIMBEG(I)=LIMEND(I)
      LIMEND(I)=LTEMP
3      CONTINUE
      LAT=LIMBEG(I)
C      CALCULATE NUMBER OF BANDS TO PROCESS FOR THIS PAIR OF
C      LIMITS
      NDEX=(LIMEND(I)+167)/333-(LAT+167)/333+1
C      PROCESS REQUIRED NUMBER OF LATITUDE BANDS
      DO 35 II=1,NDEX
      NLAT=(LAT+167)/333+1
C      CALCULATE LATITUDE OF NEXT BAND TO PROCESS
      NCXTLAT=(NLAT+2)/3+NLAT*333-167
C      ZERO STORAGE AREAS FOR THIS BAND
      DO 5 J=1,50
      CONTRY(I,J)=0
      CONTRY(3,J)=0
      CONTRY(2,J)=0
      DO 4 JJ=1,8
      ZONE(J,JJ)=0
4      FZONE(J,JJ)=0.
5      CONTINUE
      DO 6 JJ=1,8
      ZONTOT(JJ)=0
6      CONTINUE
C      CALCULATE AREA FOR CELLS OF THIS BAND
      LAT=LAT-90000
      ALAT=IABS(LAT)
      AREA=300.0*(COS((ALAT-167.)*1.74533E-5)+COS((ALAT+167.)
      *1.74533E-5))
C      READ FILE FOR THIS BAND
      CALL RETCELL (LAT,250)
      IF (LENGTH) 34,7,7
7      IF (IRDATE-11111111B) 8,34,8

```

```

8   IF (LENCL .LE. 3) GO TO 29
    NUMCON=0
    DO 23 N=4,LENCL,5
C     FIND POPULATION FOR THIS CELL
C     DECODE(8,1003,INFILE(N+4))KK,KLEN
    PRINT 38
    IPOP=0
    IF (KLEN .LE. 0) GO TO 10
    KK=KK+LENCL
    KL=KK+KLEN-1
    DO 9 NN=KK,KL,3
C     FIND POPULATIONS FOR CITIES IN THIS CELL
    IPOP=IPOP+INFILE(NN)*1000
9     CONTINUE
10    IPOP=IPOP+INFILE(N+1)*100
C     CALCULATE POPULATION DENSITY
    IDENS=IPOP/AREA
C     DECODE(2,1004,INFILE(N+3))ICON
    PRINT 39
C     DETERMINE COUNTRY TO WHICH THIS CELL BELONGS
    DO 13 J=1,50
    IF (CONTRY(1,J) .NE. INFILE(N+2)) GO TO 11
    IF (CONTRY(2,J) .EQ. ICON) GO TO 12
11    IF (CONTRY(1,J) .NE. 0) GO TO 13
    NUMCON=NUMCON+1
    CONTRY(1,J)=INFILE(N+2)
    CONTRY(2,J)=ICON
12    MC=J
    CONTRY(3,J)=CONTRY(3,J)+IPOP
    GO TO 14
13    CONTINUE
C     DETERMINE DENSITY RANGE TO WHICH THIS POPULATION DENSITY BELONGS.
14    IF (IDENS .GT. 1) GO TO 15
    MZ=1$GOTO96
15    IF (IDENS .GT. 5) GO TO 16
    MZ=2$GOTO96
16    IF (IDENS .GT. 25) GO TO 17
    MZ=3$GOTO96
17    IF (IDENS .GT. 100) GO TO 18
    MZ=4$GOTO96
18    IF (IDENS .GT. 500) GO TO 19
    MZ=5$GOTO96
19    IF (IDENS .GT. 1000) GO TO 20
    MZ=6$GOTO96
20    IF (IDENS .GT. 5000) GO TO 21
    MZ=7
    GO TO 22
21    MZ=8
22    ZONE(MC,MZ)=ZONE(MC,MZ)+1
23    CONTINUE
C     FIND TOTALS FOR THIS BAND
    DO 26 J=1,NUMCON
    TOTZON(J)=0
C     COMPUTE TOT CELLS FOR THIS COUNTRY
    DO 24 JJ=1,8

```

```

24  TOTZON(J)=TOTZON(J)+ZONE(J,JJ)
C      COMPUTE TOTAL AREA FOR THIS COUNTRY
    TOTAREA(J)=AREA*TOTZON(J)
C      COMPUTE AVERAGE DENSITY FOR THIS COUNTRY
    AVGDEN(J)=CONTRY(3,J)/TOTAREA(J)
C      COMPUTE PER CENT TOTAL CELLS OF THIS COUNTRY WHICH
C      ARE IN THIS ZONE
    DO 25 JJ=1,8
25  FZONE(J,JJ)=ZONE(J,JJ)*100.0/TOTZON(J)
26  CONTINUE
C      COMPUTE ZONE TOTALS FOR THIS BELT
C      COMPUTE TOTAL CELLS IN THIS ZONE
    DO 27 JJ=1,8
    DO 27 J=1,NUMCON
    ZONTOT(JJ)=ZONTOT(JJ)+ZONE(J,JJ)
27  CONTINUE
C      COMPUTE PER CENT TOTAL CELLS IN THIS ZONE
    ITOTCEL=(LENCL-3)/5
    DO 28 J=1,8
28  ZONPER(J)=ZONTOT(J)*100.0/ITOTCEL
C      COMPUTE TOTAL AREA IN THIS BELT
    AREATOT=AREA*ITOTCEL
    IND=1$GOTO450
29  IND=0
C      PRINT THE REPORT FOR THIS BELT
    PRINT 40,LAT,IDATE
    IF (IND.EQ.0) GO TO 32
    NN=NUMCON
    J=0
    IF (NUMCON.GT.24) NN=25
    N=1
30  PRINT 41,1,2,3,4,5,6,7,8
    PRINT 42
    DO 31 J=N,NN
    PRINT 43,(CONTRY(K,J),K=1,2),(ZONE(J,JJ),FZONE(J,JJ),JJ=1,8)
    $,AVGDEN(J),TOTZON(J),TOTAREA(J)
31  CONTINUE
    N=N+25
    NN=NUMCON
    IF (NN.LT.N) GO TO 33
    PRINT 44
    GO TO 30
32  PRINT 46
    GO TO 34
33  PRINT 42
    PRINT 45,(ZONTOT(J),ZONPER(J),J=1,8),ITOTCEL,AREATOT
34  LAT=NEXTLAT
35  CONTINUE
36  CONTINUE
    REWIND 49
    CALL EXIT
C      FORMAT STATEMENTS
37  FORMAT (2I10,9X,A8)
38  FORMAT (2I4)
39  FORMAT (A2)

```

```

40  FORMAT(1H1,14X,*POPULATION ANALYSTS FOR LATITUDE BELT*,16,*
    $,ACCORDING TO ZONES OF POPULATION PER SQUARE MILE*,2X,A8)
41  FORMAT(// * COUNTRY*,3X,7(*ZONE*,12,7X),*ZONE*,12,5X,*AVERAGE*,7X
    $,*TOTAL*/3X,*CODE*,5X,*0-1*,10X,*2-5*,10X,*6-25*,8X,*26-100*,6X
    $,*101-500*,6X,*501-1000*,4X,*1001-5000*,4X,*5001-*,7X
    $,*POPULATION COUNTRY*/8X,8(*CELLS = AREA*,2X),1X,*DENSITY CEL
    $LS AREA*)
42  FORMAT (1X,135(*-*))
43  FORMAT (/1X,R4,A2,I5,F7.2,7(I6,F7.2),1X,F9.2,16,F10.2)
44  FORMAT (1H1,/)
45  FORMAT (71X,*TOTALS*,15,F7.2,7(I6,F7.2),10X,16,F10.2)
46  FORMAT(//1X,*THIS BELT HAS NO RECORDED DATA.*/)
    END

```


CELLGEN

Purpose

This program is written for the CDC 3600. CELLGEN generates cells for a given latitude band between given longitude limits for one country, smearing a given population over the generated cells. Zero populations will be generated only if the input population is zero. An input population of one ($\times 10^2$) results in each generated population being set to one.

Program Description

Three options are available to the user of CELLGEN: (1) full run, (2) generate and list, or (3) merge only. A full run generates new cells and writes them onto a "scratch" tape, then merges that tape with the existing BCD tapes. A generate-and-list run stops after listing the generated cells. A merge-only run accepts the tape written during a generate-and-list run and merges new cells with old cells.

Input tapes are placed on units 1-9, starting with unit 1 and using as many units as necessary. Output tapes are placed on units 11-19, starting with unit 11. The scratch unit is 49.

Unlimited requests are allowed for one job, one request per card. However, the program cannot generate more cell records than can be written on one tape. If requests overlap in a latitude band, the easternmost country receives the overlapping cells; this assignment effectively reduces the ending longitude limit for the westernmost of the two countries. One request may generate cells in only one latitude band. Requests are processed in batches of 500. Since the generated cell tape must be in both

ascending latitude order, and in ascending longitude order within a latitude, no group of 500 may overlap any other latitude group. The simplest way to ensure that the input deck is in ascending order is to run the deck through a card sorter, sorting on Cols. 11-16, in ascending order of field content, then sorting on Cols 1-6, in ascending order, before submitting the job for a generation run.

During the merge run, any duplicate cells are compared, and the cell with the greatest population is retained; the discarded cell with either the old or new cell designation is listed on the printer. If a tape is not used in the merge, a printer message informs the user that tape x was not merged; the tape is not recopied. Thus, a merge run could produce only one tape or several tapes.

CELLGEN Card Input Format

Block 1 - one card

col. 5 number of input tapes
 15 option desired
 1 = full run
 2 = generate and list
 3 = merge only

Block 2 - number of cards unlimited

col. 1-6 latitude band
 11-16 beginning longitude
 21-26 ending longitude
 31-36 characters 5 through 10 of country code

41-46 population to be smeared

51 F or S

57-60 date of data

Block 3 - one card

col. 1-6 999999

Flow Chart for Cell Generator--CELLGEN

Cell Generator (Continued)

Cell Generator (Continued)

Cell Generator (Continued)

Cell Generator (Continued)

Cell Generator (Continued)

Cell Generator (Concluded)

```

C          CELLGEN GENERATES CELLS IN A GIVEN LATITUDE BAND BETWEEN
C          GIVEN LONGITUDE LIMITS, INCLUSIVE, FOR A GIVEN COUNTRY AND
C          SMEARS A GIVEN POPULATION OVER THE GENERATED CELLS.
C          TAPE UNITS ARE 1,2,3,4,5,6,7,8,9 (MAX 9 INPUT REELS),
C          OUTPUT UNITS ARE 11, ..., 19, GENERATED CELL UNIT IS 49.
C          ALL TAPES ARE NOT NECESSARILY REWRITTEN
C
C          CAUTION, INPUT DECK MUST BE SORTED ON FIELD 1 (COL. 1-6)
C          AND ON FIELD 2 (COL. 11-16)
C          IN ASCENDING ORDER, WITH FIELD 2 IN ORDER WITHIN
C          FIELD 1.
C
C          DIMENSION LAT(500),LONB(500),LONE(500),CC(2,500),POP(500)
C          $,NEWCEL(8,720),FORS(500),IDATE(500),DUMMY(10)
C          DIMENSION OLDCEL(10240),OUTARE(12000)
C          INTEGER CC,POP,BLANK,ZERO,FORS
C          COMMON /A/BLANK,ISLAVE,NTAPE,JUNIT
C          COMMON /B/DUMMY
C          COMMON /C/NEWCEL,OLDCEL,OUTARE
C          DATA(BLANK=1H),(ZERO=4H0000),(IDUM=1H0),(ISLAVE=2H1)
C          REWIND 49
C          IOPT = 1 IF FULL RUN SELECTED
C          IOPT = 2 IF GENERATE AND LIST SELECTED
C          IOPT = 3 IF MERGE ONLY SELECTED
C          READ 1,NTAPE,IOPT
C          FORMAT(4X,11,9X,11)
C          IEOF=0
C          LASBAN=0
C          LASCEL=0
C
C          OPTION CHECK
C          IF(IOPT-3) 2,2,2
C
C          READ DATA CARDS. EACH CARD CONTAINS IN ORDER LATITUDE,
C          STARTING LONGITUDE, ENDING LONGITUDE, 6-CHARACTER COUNTRY
C          CODE, POPULATION IN HUNDREDS, F OR S, DATE.
C          ANY NUMBER OF CARDS ARE ALLOWED. THE LAST CARD MUST
C          HAVE 999999 IN COLUMNS 1-6.
C          DO 7 I=1,500
C          READ 3,LAT(I),LONB(I),LONE(I),CC(1,I),CC(2,I),POP(I),FORS(I)
C          $,IDATE(I)
C          3  FORMAT(3(I6,4X),A4,A2,4X,I6,4X,A1,5X,A4)
C          IF(LAT(I)-999999) 5,4,5
C          4  NUMB=I-1
C          IEOF=1
C          GO TO 8
C          5  IF(LONE(I)-LONB(I)) 6,7,7
C          6  IEMP1=LONB(I)
C          LONB(I)=LONE(I)
C          LONE(I)=IEMP1
C          7  CONTINUE
C          NUMB=500

```

```

C      SORT ARRAYS IN ORDER OF ASCENDING LATITUDE
8      CALL SORTR (LAT,LONB,LONE,CC,POP,FORS,IDATE,1,NUMB)
C
C      SORT BANDS BY STARTING LONGITUDE, IN ASCENDING ORDER
      II=0
      N=NUMB-1
      DO 17 I=1,N
        J=I+1
        IF (LAT(I)-LAT(J)) 12,9,12
9       IF (II) 11,10,11
10      II=I
11      JJ=J
        GO TO 17
12      IF (II) 13,17,13
13      CALL SORTR (LONB,LAT,LONE,CC,POP,FORS,IDATE,II,JJ)
        MM=JJ-1
        DO 16 M=II,MM
          IF (LONE(M)-LONB(M+1)) 16,14,14
14      LONE(M)=LONB(M+1)+500
          PRINT 15,LAT(II),CC(1,M),CC(2,M),CC(1,M+1),CC(2,M+1),LONB(M+1)
          $,CC(1,M+1),CC(2,M+1)
15      FORMAT(1X,*IN LATITUDE BAND*,I7,*,COUNTRY*,A4,A2
          $,*OVERLAPPED COUNTRY*,A4,A2,*AT LONGITUDE*,I7,*,COUNTRY*,A4,A2
          $,*RECEIVED THE OVERLAPPED CELLS.*)
16      CONTINUE
        II=0
17      CONTINUE
C
C      CHECK FOR OVERLAPPING BAND REQUESTS
C
      IF (LASBAN=LAT(I)) 21,18,19
18      IF (LASCEL=LONB(1)) 21,19,19
19      PRINT 20
20      FORMAT (123H1YOUR INPUT DECK IS NOT IN ASCENDING ORDER. EITHER RES
      $ORT THE DECK AND RESUBMIT THIS JOE, OR HAVE THE GENERATED TAPE SOR
      $TED.)
      IOPT=2
C
C      GENERATE NEW CELL RECORDS
21      DO 31 I=1,NUMB
        LEN=((LONE(I)-LONB(I)+500)*2)/1000
        IPOP=POP(I)/LEN
        IF (IPOP-1) 22,23,23
22      IPOP=1
23      IF (POP(I)) 25,24,25
24      IPOP=0
25      IF (POP(I)-1) 27,26,27
26      IPOP=1
27      DO 30 J=1,LEN
        NEWCEL(1,J)=LAT(I)
        NEWCEL(2,J)=LONB(I)+(J-1)*500
        NEWCEL(3,J)=FORS(I)/4398046511104*4398046511104+IDUM/64
        NEWCEL(4,J)=IPOP
        NEWCEL(5,J)=ZERO
        NEWCEL(6,J)=CC(1,I)

```

```

NEWCEL(7,J)=CC(2,I)
NEWCEL(8,J)=IDATE(I)
C      WRITE GENERATED CELL TAPE ON 49
      WRITE (49,28) (NEWCEL(JJ,J),JJ=1,8)
28     FORMAT (2I6,A3,I5,A4,A4,A2,A4)
      PRINT 29,BLANK,(NEWCEL(JJ,J),JJ=1,8)
29     FORMAT ('A1,2I6,A3,I5,A4,A4,A2,A4')
30     CONTINUE
31     CONTINUE
      LASBAN=NEWCEL(1,LEN)
      LASCEL=NEWCEL(2,LEN)
      IF (IEOF .EQ. 0) GO TO 2
      END FILE 49
      REWIND 49
      IF(TIOPT=1) 33,32,33
32     CALL MERGE
33     CALL EXIT
      END

```

```

SUBROUTINE MERGE
  DIMENSION OLDCEL(12,1000),OUTARE(12,1000),NEWCEL(8,500)
  COMMON /A/BLANK,ISLAVE,NTAPE,IUNIT
  COMMON /B/OCEL,NCEL,NEW,IOLD,LASUNI,NUNIT,NREC,LASNEW,NOUT,LASOLD
  COMMON /C7NEWCEL,OLDCEL,OUTARE
  INTEGER OLDCEL,OUTARE,BLANK,OCEL
  LASNEW=0
  LASOLD=0
  NOUT=0
  NFLAG=0
  LASUNI=0
  OCFL=0
  NCFL=0
  NREC=0
  IOLD=1
  NEW=1
  NUNIT=1
C
C      READ INPUT TAPES AND MERGE TO WRITE NEW TAPE
C
C      FIND WHICH TAPE NEW CELLS START ON
  IF(NTAPE-1) 9,9,2
1  FORMAT (2I6,A3,I5,A4,A4,A2,A4)
2  READ (49,1) (NEWCEL(I,1),I=1,8)
  DO 8 IUNIT=2,NTAPE
  READ (IUNIT,3) (OLDCEL(I,1),I=1,12)
  REWIND IUNIT
3  FORMAT (2I6,A3,I5,A4,A4,A2,A4,1X,3A4,A2,35X)
  IF(NEWCEL(1,1)-OLDCEL(1,1)) 9,4,4
4  IF(IUNIT-NTAPE) 5,7,5
5  PRINT 6,NUNIT
6  FORMAT (*TAPE*,I4,*WASNOTMERGED*)
7  NUNIT=IUNIT
8  CONTINUE
  REWIND NTAPE
9  REWIND 49
  CALL READNE (NGO)
  CALL READOL (IGO)
  JUNIT=NUNIT+10
C
C      MERGE NEW AND OLD CELLS
C
10 CONTINUE
  IF(NEWCEL(1,1)-OLDCEL(1,250)) 15,15,11
11 IF(NOUT) 13,13,12

```

```

12  NN=NOUT
    CALL WRITEN (NN)
13  WRITE (JUNIT,3) ((OLDCEL(I,J),I=1,12),J=1,1000)
    IOLD=2000
    CALL READOL (IGO)
    IF(IGO) 10,10,30
14  CALL READNE (NGO)
    IF(NGO-1) 15,39,10
15  CALL READOL (IGO)
    IF(IGO) 16,16,30
16  IF(NEWCEL(1,NEW)=OLDCEL(1,IOLD)) 26,19,17
17  CALL WRITEN (1000)
C    MOVE OLD RECORD TO OUTPUT AREA
    DO 18 M=1,12
18  OUTARE(M,NOUT)=OLDCEL(M,IOLD)
C    IF NFLAG.NE.0 IGNORE CURRENT NEW CELL
    IF(NFLAG) 29,15,29
C    IF THIS OLD RECORD IS CITY, MOVE IT TO OUTPUT AREA
19  IF(OLDCEL(3,IOLD)=ISLAVE) 20,17,20
C    IF NEW LONGITUDE .GT. OLD, MOVE OLD CELL TO OUTPUT AREA
20  IF(NEWCEL(2,NEW)=OLDCEL(2,IOLD)) 26,21,17
C    IF NEW LONGITUDE .LT. OLD, MOVE NEW CELL TO OUTPUT AREA
C    IF NEW CELL IS A DUPLICATE, SAVE RECORD WITH GREATEST POPULA
21  NFLAG=1
    IF(NEWCEL(4,NEW)=OLDCEL(4,IOLD)) 23,24,24
22  FORMAT (*NEWCELL IGNORED*,5X,I6,1X,I6,A3,I5,A4,A4,A2,A4)
23  PRINT 22,(NEWCEL(IJK,NEW),IJK=1,8)
    GO TO 17
24  PRINT 25,(OLDCEL(IJK,IOLD),IJK=1,12),(NEWCEL(IJK,NEW),IJK=1,8)
25  FORMAT (*OLDCELL*,1X,I6,1X,I6,A3,I5,A4,A4,A2,A4,1X,3A4,A2
    $,*WASREPLACEDBYNEWCELL*,1X,I6,1X,I6,A3,I5,A4,A4,A2,A4)
    CALL READOL (IGO)
    IF(IGO) 26,26,30
26  CALL WRITEN (1000)
    NFLAG=0
C    MOVE NEW CELL TO OUTPUT AREA
    DO 27 M=1,8
27  OUTARE(M,NOUT)=NEWCEL(M,NEW)
    DO 28 M=9,12
28  OUTARE(M,NOUT)=BLANK
    CALL READNE (NGO)
    IF(NGO-1) 16,39,10
29  NFLAG=0
    GO TO 14
C
C    ALL OLD CELLS HAVE BEEN PROCESSED
C    FINISH NFW CELLS

```

```

30  CONTINUE
    IF (NEW) 32,31,32
31  IF (NCEL) 32,36,32
32  DO 35 MM=NEW,NCEL
    CALL WRITEN (1000)
    DO 33 M=1,8
33  OUTARE(M,NOUT)=NEWCEL(M,MM)
    DO 34 M=9,12
34  OUTARE(M,NOUT)=BLANK
35  CONTINUE
    NN=NOUT
    CALL WRITEN (NN)
36  IF (LASNEW=1) 37,38,37
37  NEW=NCEL+1
    CALL READNE (NGO)
    GO TO 30
38  END FILE JUNIT
    REWIND 49
    REWIND JUNIT
    CALL EXIT
C      ALL NEW CELLS HAVE BEEN PROCESSED
C      FINISH OLD CELLS
39  CONTINUE
    IF (IOLD) 41,40,41
40  IF (OCEL) 41,44,41
41  DO 43 MM=IOLD,OCEL
    CALL WRITEN (1000)
    DO 42 M=1,12
42  OUTARE(M,NOUT)=OLDCEL(M,MM)
43  CONTINUE
    NN=NOUT
    CALL WRITEN (NN)
44  IF (LASUNI=1) 45,38,45
45  IF (LASOLD=2) 46,47,46
46  LASOLD=1
    IOLD=OCEL+1
    CALL READOL (IGO)
    GO TO 39
47  WRITE (61,48) NUNIT
48  FORMAT(1X,I6,2X,*WAS THE LAST REEL TO BE REWRITTEN.*)
    GO TO 38
END

```



```

SUBROUTINE SORTR (KEY,ARRAY1,ARRAY2,ARRAY3,ARRAY4,ARRAY5,ARRAY6,I
$ ,J)
DIMENSION KEY(500),ARRAY1(500),ARRAY2(500),ARRAY3(2,500)
$ ,ARRAY4(500)
DIMENSION ARRAY5(500),TEMP3(2),ARRAY6(500)
INTEGER ARRAY1,ARRAY2,ARRAY3,ARRAY4,ARRAY5,TEMP1,TEMP2,TEMP3
$ ,TEMP4,TEMP5,TEMP6,ARRAY6
NI=J-1
DO 3 N=I,NI
NJ=N+1
DO 2 NN=NJ,J
IF (KEY(N)-KEY(NN)) 2,2,1
1 ITEM=KEY(N)
TEMP1=ARRAY1(N)
TEMP2=ARRAY2(N)
TEMP3(1)=ARRAY3(1,N)
TEMP3(2)=ARRAY3(2,N)
TEMP4=ARRAY4(N)
TEMP5=ARRAY5(N)
TEMP6=ARRAY6(N)
KEY(N)=KEY(NN)
ARRAY1(N)=ARRAY1(NN)
ARRAY2(N)=ARRAY2(NN)
ARRAY3(1,N)=ARRAY3(1,NN)
ARRAY3(2,N)=ARRAY3(2,NN)
ARRAY4(N)=ARRAY4(NN)
ARRAY5(N)=ARRAY5(NN)
ARRAY6(N)=ARRAY6(NN)
ARRAY1(NN)=TEMP1
ARRAY2(NN)=TEMP2
ARRAY3(1,NN)=TEMP3(1)
ARRAY3(2,NN)=TEMP3(2)
ARRAY4(NN)=TEMP4
ARRAY5(NN)=TEMP5
ARRAY6(NN)=TEMP6
KEY(NN)=ITEM
2 CONTINUE
3 CONTINUE
RETURN
END

```

```

SUBROUTINE READOL (IGO)
  DIMENSION OLDCEL(12,1000),OUTARE(12,1000),NEWCEL(8,500)
  COMMON /A/BLANK,ISLAVE,NTAPE,JUNIT
  COMMON /B/OCEL,NCEL,NEW,IOLD,LASUNI,NUNIT,NREC,LASNEW,NOUT,LASOLD
  COMMON /C/NEWCEL,OLDCEL,OUTARE
  INTEGER OLDCEL,OUTARE,OCEL,BLANK
C      READ 500 OLD CELL AND CITY RECORDS
      IGO=0
      IOLD=IOLD+1
      IF(IOLD-OCEL) 16,16,1
1      IOLD=1
      OCEL=0
      IF(LASUNI-1) 2,15,2
2      I=1
3      READ (NUNIT,4) (OLDCEL(J,I),J=1,12)
4      FORMAT (2I6,A3,I5,A4,A4,A2,A4,1X,3A4,A2,35X)
      IF(EOF,NUNIT) 5,10
5      IF(NUNIT-NTAPE) 7,6,17
6      LASUNI=1
7      REWIND NUNIT
      NUNIT=NUNIT+1
      IF(LASOLD-1) 14,8,9
8      LASOLD=2
      GO TO 14
9      LASUNI=1
      GO TO 14
10     IF(IOCHECK,NUNIT) 11,13
11     I=I-1
      PRINT 12,NUNIT,(OLDCEL(J,I),J=1,12)
12     FORMAT(*PARITY ERROR ON TAPE*,I6,*FOLLOWING THIS RECORD*/1X,2I6,A3
      $,I5,A4,A4,A2,A4,1X,3A4,A2,35X)
13     I=I+1
      IF(I-1000) 3,3,14
14     OCEL=I-1
      IF(OCEL) 15,15,16
15     IGO=1
16     JUNIT=NUNIT+10
      RETURN
17     PRINT 18,NUNIT,NTAPE
18     FORMAT(*COUNTER ERROR. PRESENT UNIT IS*,I6,2X,*LAST UNIT SHOULD BE
      $*,I6)
      CALL EXIT
      END

```

```

SUBROUTINE READNE (NGO)
DIMENSION OLDCEL(12,1000),OUTARE(12,1000),NEWCEL(8,500)
COMMON /A/BLANK,ISLAVE,NTAPE,JUNIT
COMMON /B/OCEL,NCEL,NEW,IOLD,LASUNI,NUNIT,NREC,LASNEW,NOUT,LASOLD
COMMON /C/NEWCEL,OLDCEL,OUTARE
INTEGER OLDCEL,OUTARE,OCEL,BLANK
NGO=0
NEW=NEW+1
C      READ 250 NEW CELL RECORDS
      IF (NEW-NCEL) 15,15,1
1      NFW=1
      NCEL=0
      IF (LASNEW-1) 2,14,2
2      I=1
3      READ (49,4) (NEWCEL(J,I),J=1,8)
4      FORMAT (2I6,A3,I5,A4,A4,A2,A4)
      IF (EOF,49) 5,6
5      NCEL=I-1
      LASNEW=1
      REWIND 49
      GO TO 11
6      IF (IOCHECK,49) 7,9
7      I=I-1
      PRINT 8,(NEWCEL(J,I),J=1,8)
8      FORMAT(*THE GENERATED CELL FOLLOWING THIS GENERATED CELL COULD NOT
      $ BE RETRIEVED*/IX,2I6,A3,I5,A4,A4,A2,A4)
9      I=I+1
      IF (I-500) 3,3,10
10     NCEL=I-1
11     IF (NCEL) 14,14,12
12     IF (NEW-IOLD) 15,13,15
13     IF (IOLD) 15,16,15
14     NGO=1
15     RETURN
16     NGO=2
      RETURN
END

```

```

SUBROUTINE WRITEN (NN)
  DIMENSION OLDCEL(12,1000),OUTARE(12,1000),NEWCEL(8,500)
  COMMON /A/BLANK,ISLAVE,NTAPE,JUNIT
  COMMON /B/OCEL,NCEL,NEW,TOLD,LASUNT,NUNIT,NREC,LASNEW,NOUT,LASOLD
  COMMON /C/NEWCEL,OLDCEL,OUTARE
  INTEGER OLDCEL,OUTARE,OCEL,BLANK
  C      WRITE NEW TAPE
      JUNIT=NUNIT+10
      NOUT=NOUT+1
      IF(NOUT>NN) 5,5,1
1      IF(NN) 2,2,3
2      NOUT=NOUT-1
      GO TO 5
3      WRITE (JUNIT,4) ((OUTARE(I,J),I=1,12),J=1,NN)
4      FORMAT (2I6,A3,I5,A4,A4,A2,A4,1X,3A4,A2,35X)
      NOUT=1
5      RETURN
      END

```

APPENDIX A

IBM 7090

DECK STRUCTURE FOR PROCESSING WITH A FORTRAN SOURCE DECK, AND OBTAINING A BINARY DECK.

(Ref. IBM manual entitled IBM 7090/7094 Operating Systems Basic Monitor (IBSYS), No. C28-6248.)

IBM 7090

DECK STRUCTURE FOR PROCESSING WITH A BINARY PROGRAM DECK.

(Ref. IBM Manual entitled IBM 7090/7094 Operating Systems Basic Monitor (IBSYS) No. C28-6248.)

CDC 3600

DECK STRUCTURE FOR PROCESSING WITH A FORTRAN SOURCE DECK.

(Ref. Computer Systems Fortran Reference Manual, Sec. 11.)

APPENDIX B

MAPPRT

Purpose

This program was developed to facilitate examination of the binary data base for errors in location of country codes. MAP on PRinTer (MAPPRT) uses the printer to display graphically all cells for which data are recorded. Unfortunately, a final map was not completely produced before the contract terminated. The program is finished; but errors on the data tape prohibited completion of a world map.

Program Description

Each cell is represented on the map by the two characters of the country code assigned to that cell. If no population data exist for a particular cell (e.g., in an ocean), then two dots or periods are printed in place of a country code. A user option allows two zeros to be printed for any cell without population data.

Since two printer characters are required for one cell, and since the maximum printer capability is 120 characters per line, 60 cells may be represented on one line of one printer page. One band has 720 cells, so (one line across) 13 printer pages are required to complete one band of the map. There are 540 bands from the South Pole to the North Pole, which require 9 printer pages at 60 bands per page to print 540 bands. Thus the map requires $13 \times 9 = 117$ pages.

To make reading the map simpler, two pointers are given: The first is the latitude and longitude of the center of the upper left cell on each page. The second pointer is given in every band for each country crossed

by the band and is the number of consecutive cells that have the same country code, providing the string contains three or more cells. For example, a country containing four cells in a string in one band and having the country code, CC would appear as CC04CCCC, i. e., the number of cells in the string replaces the second pair of characters. The maximum number printed is 59 since each page is only 59 cells wide.

The program requires 27154 words of storage in the IBM System 360 Model 44 computer, for which it is written. On that computer the time required to produce a complete world map is approximately 30 minutes if the tape drive is a medium speed unit and the printer is a high speed unit.

Program Operation

Only one data card is required; it contains three items:

<u>Card Image Columns</u>	<u>Data</u>	<u>Mode</u>
1-10	latitude of band to start map on	I
20	number of 60-band belts required	I
30	zero population option	I
	1 no	
	0 yes	

MAPPRT maps the world in 60-band belts so that it reads 60 bands, stores the keys for those bands, then prints them before starting the next 60 bands. To process one band, the program reads the data for that band;

if no data exist, print keys are stored to indicate that 720 pairs of dots (· ·) should be printed for the band. When data exist for the band, the longitude, country code, and population of each cell recorded is stored. After investigating those data, the program stores print keys that account for each cell, indicating position and print characters--the country code, dots, or zeros. After repeating this procedure for 60 bands, the program prints 13 pages and starts the next 60 bands. Refer to the flowchart for detailed operation of MAPPRT.

Maintenance

MAPPRT was written specifically for the IBM System 360 Model 44 operating under programming system 44, using the full capabilities of Fortran IV. Much of the data storage is half-word integers, which would make conversion of the program for a small machine difficult.

Subroutine ITOBCD, used to convert the integer number of cells in a string to 2 BCD characters, was written with routines specific to the Dikewood facility. The routine must be approximated in Fortran for use at any other computer facility.

FLOW CHART FOR MAP ON PRINTER

MAPPRT (continued)

MAPPRT (continued)

MAPPRT (continued)

MAPPRT (continued)

MAPPRT (concluded)

```

C *****
C *                               MAP ON PRINTER                               *
C *
C * PROGRAM TO MAP THE WORLD AS IT EXISTS ON NUCLEAR SAFETY *
C * DATA TAPE BY PRINTING THE TWO-CHARACTER COUNTRY CODE FOR *
C * EACH CELL CONTAINING DATA AND A NONZERO POPULATION, OR *
C * TWO ZEROES FOR ZERO POPULATION, OR TWO DOTS FOR NO DATA *
C *
C *****
C
C THE LATITUDES CORRESPONDING TO THE BAND LIMITS OF EACH BELT
C
C BELT BANDS LATITUDES
C 1 1 - 60 000167 - 019833
C 2 61 - 120 020167 - 039833
C 3 121 - 180 040167 - 059833
C 4 181 - 240 060167 - 079833
C 5 241 - 300 080167 - 099833
C 6 301 - 360 100167 - 119833
C 7 361 - 420 120167 - 139833
C 8 421 - 480 140167 - 159833
C 9 481 - 540 160167 - 179833
C
C *****
C
C INTEGER*2 IDCN(60,600*),IALPH(706*),ICCEL(630*),NHAF,IDUM(2*
COMMON IALPH,IDCN,IDUM,NDUMMY(1000*),TWO5,IPOPCL,LONCEL,ICCEL
DIMENSION LONCEL(630*),IPOPCL(630*)
DIMENSION INFILE(5043*
EQUIVALENCE (INFILE)1*,IDCN)1,516**, (INFILE)2*,LENCL*
EQUIVALENCE (IALPH)27*,NHAF*, (INT,IDUM)1**
EQUIVALENCE (INFILE)3*,LENSG*
NAMLIST/NAM/IBR
TWO5=32
IN=12
REWIND IN
C.....INITIALIZE ALPHABET
IALPH)1*=193
DO 1 I=1,8
1 IALPH)I.1*=IALPH)I*.1
IALPH)10*=209
DO 2 I=10,17
2 IALPH)I.1*=IALPH)I*.1
IALPH)19*=226
DO 3 I=19,25
3 IALPH)I.1*=IALPH)I*.1
IALPH)28*=240*256.240
IALPH)30*=64*256.64
IALPH)29*=75*256.75
IDOTS=29
K=30
DO 5 I=1,26
DO 4 J=1,26
K=K.1
INT=IALPH)I*
INT=INT*256.IALPH)J*

```

```

4     IALPH)K*=IDUM)2*
5     CONTINUE
C.....PROGRAM MAY BE STARTED AT ANY ONE OF THE 9 BELTS OF BANDS AND WILL
C.....PROCESS THE SPECIFIED NUMBER OF BELTS
C.....LATST IS THE SOUTHERN-MOST LATITUDE USER WISHES TO MAP
C.....IBELT IS THE NUMBER OF 60 BAND BELTS USER WISHES TO MAP ABOVE LATS
6     READ )5,7,END=46* LATST,IBELT,IZERO
7     FORMAT )3)10*
      IF)IZERO* 9,9,8
8     NZERO=2
      GO TO 10
9     NZERO=1
C.....COMPUTE BAND NUMBER OF SOUTHERN-MOST BAND OF STARTING BELT
10    NFILE=)LATST.90)67*7333
      N=NFILE-1)NFILE/60**60
      I=0
      IF)N* 11,12,11
11    NFILE=)NFILE/60*.1**60
12    IFILE=NFILE-60
      IF)IFILE* 15,15,13
13    DO 14 I=1,IFILE
14    READ )IN* )INFILE)J*,J=1,3*,)INFILE)J*,J=4,LENSG*
15    DO 45 MM=1,IBELT
C.....IDCN ARRAY WILL CONTAIN AT PRINT TIME KEYS TO PRINTING 60 BANDS
C.....ON 13 PRINTER PAGES, 60 LINES PER PAGE.
C.....I=COLUMN INDEX--1,720
C.....J=ROW INDEX--1,60
C.....READ 60 BANDS AND PROCESS EACH
      DO 44 JK=1,60
      J=61-JK
      I=1
      READ )IN,END=47* )INFILE)K*,K=1,3*,)INFILE)K*,K=4,LENSG*
      LEN=)LENCL-3*/8
      KK=0
      IF)LEN* 18,18,16
16    DO 17 K=4,LENCL,8
      KK=KK.1
      LONCEL)KK*=INFILE)K*
      IPOPC)KK*=INFILE)K.1*
      INT=INFILE)K.3*
      ICCCEL)KK*=IDUM)1*
17    CONTINUE
18    PRINT 7,INFILE)1*,LEN
      ICON=0
      IF)LEN-1* 19,20,20
C.....ACCOUNT FOR EACH CELL IN CURRENT BAND
C.....NO DATA. STORE DOTS FOR EACH CELL.
19    N=720
      CALL STCON )N,IDOTS,J,I*
      N=0
      GO TO 44
C.....DATA EXISTS. STORE PROPER CODE FOR EACH CELL.
20    LON=250
      N=0
      DO 38 K=1,LEN

```

```

      IF(LONCEL)K*-LON* 21,27,24
C.....ERROR IF LON .GT. LONCEL
21  PRINT 22,K,LONCEL)K*,LON,NFILE,ILAT,LONCEL)K-1*
22  FORMAT('PROGRAM ERROR ONE. LONCEL)'I3'*= 'I7',LON='I7',NFILE='
      $I4',LAT='I7',PRECEDING LONCEL='I7*
23  CONTINUE
      CALL MAP )NFILE,J,1*
      CALL EXIT
C.....NO DATA FOR CELLS FROM LON TO LONCEL.
24  IF)N* 26,26,25
25  CALL STCON )N,ICON,J,I*
C.....STORE DOTS FOR MISSING COUNTRY CODES.
26  N=LONCEL)K*-LON*/500
      CALL STCON )N,IDOTS,J,I*
      N=0
C.....FIND NUMERIC KEY FOR LONCEL'S COUNTRY CODE
27  DO 28 L=30,706
      IF)ALPH)C*=LONCEL)K** 28,30,28
28  CONTINUE
      PRINT 29,LONCEL)K*
29  FORMAT )IX,'COUNTRY CODE UNRECOGNIZED)'A2'*,REPLACED BY DOTS)....'
      $*
      LM=IDOTS
      GO TO 31
30  LM=L
C.....DETERMINE IF POPULATION IS GT 0.
31  IDIR=1
      IF )N.LE. 0* IDIR=2
      IF)IPOPCL)K** 32,32,36
32  GO TO )33,34*,NZFRO
33  LM=30
34  GO TO )35,37*,IDIR
35  CALL STCON )N,ICON,J,I*
      N=0
      GO TO 37
C.....DETERMINE IF CURRENT C. CODE IS NEW
36  IF)LM-ICON* 34,37,34
37  N=N.1
      ICON=LM
      LON=LONCEL)K*.500
38  CONTINUE
C.....FINISH CURRENT RECORD
      IF)N* 40,40,39
39  CALL STCON )N,ICON,J,I*
40  IF)LON-359750* 41,41,44
41  N=)359750-LONCEL)K**/500
      IF)N* 42,42,43
42  N=1
43  CALL STCON )N,IDOTS,J,I*
44  CONTINUE
C.....PRINT NEXT 13 PAGES )60 BANDS*
      CALL MAP )NFILE,1*
      NFILE=NFILE.60
      IF)NFILE-540* 45,45,6
45  CONTINUE
46  CALL EXIT
47  PRINT 7,ILAT
      GO TO 23
      END

```

```

SUBROUTINE STCON (N,NN,J,I*
C *****
C *
C *      STCON PACKS AND STORES THE RECODED COUNTRY CODE
C *      FOR EFFICIENT ACCESS AT PRINT TIME.
C *
C *****
INTEGER*2 IDCN(60,600*,M)2*,IALPH(706*
COMMON IALPH,IDCN
EQUIVALENCE (MM,M)1**
IF(N-59* 3,3,1
1  N1=N/59
  N=N-N1*59
  DO 2 K=1,M1
    MM=59*1024.NN
    IDCN(J,I*=M)2*
    I=I.1
2  CONTINUE
3  IF(N* 5,5,4
4  MM=N*1024.NN
  IDCN(J,I*=M)2*
  I=I.1
5  RETURN
  END

```

```

      FUNCTION ITOBCD )N*
      INTEGER*4 ZERO(4)0000/
C.....CONVERT INTEGER)2 OR LESS DIGITS* TO BCD HALFWORD
      CALL INITL
      CALL AMODE )ZERO,4,1*
      CALL TMODE )3*
      CALL IMODE )N,2*
      CALL TMODE )1*
      CALL AMODEI )N1,4,1*
      ITOBCD=N1
      RETURN
      END

```

```

SUBROUTINE MAP (NFILE,NJ*)
*****
C  *      MAP PRINTS THE WORLD MAP IN 60 BANDS PER EXECUTION,      *
C  *      PRINTING THE NORTHERN-MOST LATITUDE AND THE STARTING      *
C  *      LONGITUDE IN THE UPPER LEFT CORNER OF EACH PAGE--        *
C  *      13 PAGES ARE REQUIRED TO PRINT 720 CELLS X 60 BANDS.      *
C  *      *
C  *****
INTEGER*2 IDCN(60,600*,IALPH)706*,NCEL(60*,IOUT)60,59*,IDUM)2*
COMMON IALPH,IDCN,IDUM,NDUMMY)1000*,TWO5,IOUT
DIMENSION IB)60*
INTEGER*4 ZERO/4H0000/
EQUIVALENCE (IDUM)1*,INT*
NAMelist/NAM/IBR
PRINT 1
1  FORMAT (1' ENTERING MAP'*)
   IF(NJ-61* 4,2,2
2  PRINT 3
3  FORMAT (1H1,'ERROR, NO BANDS TO PRINT'*)
   RETURN
4  IF(NJ-1* 5,6,5
5  NFILE=NFILE-NJ.1
6  NLAT=)NFILE.2*73.NFILE*333-167
   DO 7 J=1,60
7  IB)J*=0
   NLON=250
   DO 31 KK=1,13
C.....PRINT LATITUDE AND LONGITUDE OF FIRST CELL TO BE MAPPED IN UPPER
C.....LEFT CORNER OF PAGE.
   PRINT 8,NLAT,NLON
8  FORMAT (1H1,16,17/*
   DO 25 J=NJ,60
C.....STORE IN OUTPUT ARRAY CELLS (KK-1**59 THRU KK*59 FOR EACH OF THE
C.....60 BANDS
   NCEL)J*=0
   N2=0
9  IB)J*=IB)J*.1
   IDUM)1*=0
   IDUM)2*=IDCN)J,IB)J**
   N1=INT/1024
   LM=INT-N1*1024
   INT=0
   IDUM)2*=NCEL)J*
   INT=INT.N1
   NCEL)J*=IDUM)2*
   IF)KK-13* 10,11,11
10  N3=59
   GO TO 12
11  N3=12
12  IDUM)2*=NCEL)J*
   IDUM)1*=0
   IF)INT-N3* 13,15,14
13  LAST=0
   GO TO 16
14  N=INT-N3
   INT=INT-N
   NCEL)J*=IDUM)2*
   N1=N1-N

```



```

      INT=N*1024.LM
      IDCN)J,IB)J**=IDUM)2*
      IB)J*=IB)J*-1
15     LAST=1
16     INT=0
      IDUM)2*=NCEL)J*
      N=INT
      NN=N-N1.1
      DO 17 KJ=NN,N
17     IOUT)J,KJ*=IALPH)LM*
      IDUM)2*=NCEL)J*
      IDUM)1*=0
      IF)INT-N1* 22,22,18
18     IF)IOUT)J,N*=IOUT)J,NN=1** 19,22,19
C.....CONVERT NUMBER OF CONSECUTIVE LIKE CODES TO BCD AND REPLACE 2ND
C.....CELL CODE WITH BCD NUMBER IF MORE THAN 2 LIKE CODES.
19     IF)N2-2* 21,21,20
20     IBR=24
      INT=ITOBCD)N2*
      IOUT)J,NN=N2.1*=IDUM)2*
21     N2=0
22     N2=N2.N1
      IF)LAST-1* 9,23,23
23     IF)N2-2* 25,25,24
24     INT=ITOBCD)N2*
      IOUT)J,NCEL)J*=N2.2*=IDUM)2*
25     CONTINUE
C.....PRINT ONE PAGE OF MAP
      IF)N3-59* 27,26,27
26     PRINT 29,))IOUT)J,K*,K=1,N3*,J=NJ,60*
      GO TO 30
27     PRINT 28,))IOUT)J,K*,K=1,N3*,J=NJ,60*
28     FORMAT )1X,12A2*
29     FORMAT )1X,59A2*
30     NLON=NLON.29500
31     CONTINUE
      RETURN
      END

```

DISTRIBUTION:

Std. Dist. UC-36, Aerospace Safety (176)
TID-4500 (52nd Ed.)

U. S. Atomic Energy Commission
Division of Space Nuclear Systems
Space Electric Power Office
Washington, D. C. 20545

Attn: G. A. Newby, Assistant Director
G. P. Dix, Chief, Safety Branch
R. T. Carpenter, Chief, Isotope
Power Systems Branch
J. A. Powers, Chief, Isotopes
Fuels and Materials Branch
C. E. Johnson, Chief, Reactor
Power Systems Branch

U. S. Atomic Energy Commission
Space Nuclear Propulsion Office
Washington, D. C. 20545

Attn: R. S. Decker, Jr., Chief,
Safety Branch

U. S. Atomic Energy Commission
Division of Isotope Development
Washington, D. C. 20545

U. S. Atomic Energy Commission
Director of Regulation
Washington, D. C. 20545

Attn: C. K. Beck, Deputy Director
R. W. Klecker, Division of
Reactor Licensing

U. S. Atomic Energy Commission
Division of Biology and Medicine
Washington, D. C. 20545

Attn: J. Z. Holland, Fallout Studies Br.
H. D. Bruner, Asst. Director
Medical and Health Research

U. S. Atomic Energy Commission
Space Nuclear Propulsion Office
Albuquerque Extension
Albuquerque Operations Office
P. O. Box 5400
Albuquerque, New Mexico 87115
Attn: H. P. Smith

U. S. Atomic Energy Commission
Albuquerque Operations Office
P. O. Box 5400

Albuquerque, New Mexico 87115

Attn: B. W. Colston, Director
Non Nuclear Activities Div.
For: J. Nicks
J. F. Burke, Director
Operational Safety Div.

AEC Site Representative
National Aeronautics and Space Adm.
Manned Space Center
Houston, Texas 77058
Attn: W. C. Remini
Bldg. 16, Code ZS-5

Donald W. Douglas Laboratories
P. O. Box 310
Richland, Washington 99352
Attn: S. P. Gydesen

Douglas Aircraft Company
Missile & Space Systems Division
3000 Ocean Park Blvd.
Santa Monica, California
Attn: Sig Gronich, Advanced Space
Technology
Attn: Charles Willis, A-830, BCFD,
Unit 24

General Electric Company
Valley Forge Space Technology Center
P. O. Box 8555
Philadelphia, Pennsylvania 19101
Attn: S. M. Scala, Room M9539,
Space Sciences Lab
G. W. Rivenbark, Isotope Power
Systems Operations

Deputy I. G. for Insp. & Safety, USAF
Kirtland Air Force Base, New Mexico
87117

Attn: Col. D. C. Jameson (AFINSR)

Isotopes
Nuclear Systems Division
P. O. Box 4937
Middle River, Md. 21220

The Dikewood Corporation (10)
1009 Bradbury Drive, SE
Albuquerque, New Mexico 87106

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, California 91103
Attn: A. L. Klascius, Radiation Health
and Safety

Dick Shulherr (2)
Codo MOP
National Aeronautics and Space Adm.
Washington, DC 20545

DISTRIBUTION (cont):

David J. Hyams
Local Area Data Branch
Foreign Demographic Analysis Div.
Bureau of Census
2001 Wisconsin Avenue, NW
Room 410
Washington, DC 20007

R. F. Hausman
Lockheed Missile and Space Co.
Department 5150
Building 538
P. O. Box 504
Sunnyvale, California 94088

Chief, National Military Command
System Support Center (3)
The Pentagon, Room BE685
Washington, D. C. 20301
Attn: Paul T. Strickler, Jr.

Elder P. DeTurk
Aerospace Corporation
2350 E. El Segundo Blvd.
El Segundo, California 90245

Los Alamos Scientific Laboratory
P. O. Box 1663
Los Alamos, New Mexico 87544
Attn: L. D. P. King
Wright Langham
C. F. Metz, CMB-1
F. W. Schonfeld, CMF-5
J. A. Leary, CMB-11

Monsanto Research Corporation
Mound Laboratory
P. O. Box 32
Miamisburg, Ohio 45342
Attn: G. R. Grove

Thomas B. Kerr
Code RNS
National Aeronautics and Space Adm.
Washington, D. C. 20545

National Aeronautics and Space Adm.
Ames Research Center
Moffett Field, California
Attn: Glenn Goodwin

U. S. Naval Radiological Defense Lab.
San Francisco, California 94135
Attn: S. Z. Mikhail, Nuclear Systems Gp.

National Aeronautics and Space Adm.
Goddard Space Flight Center
Glenn Dale Road
Greenbelt, Maryland 20771
Attn: A. W. Fihelly, Nimbus Project

Naval Facilities Engineering Command
Department of the Navy, Code 042
Washington, D. C. 20390
Attn: Graham Hagey

Space Nuclear Propulsion Office
Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
Attn: L. Nichols

TRW Systems
P. O. Box 287
Redondo Beach, California 90278
Attn: Donald Jortner

Union Carbide Corporation
Nuclear Division
P. O. Box X
Oak Ridge, Tennessee 37831
Attn: R. A. Robinson,
Isotope Development Center
B. R. Fish, Health Physics Div.

U. S. Public Health Service
Nat. CTR for Radiological Health
1901 Chapman Avenue
Rockville, Maryland 20852
Attn: Nuclear Facilities Section

D. W. Sherwood
National Aeronautics and Space Adm.
Manned Spacecraft Center (MSC-TD)
Houston, Texas 77058

D. B. Shuster, 1200
J. R. Banister, 5270
J. D. Shreve, 5271
L. E. Lamkin, 7300
G. A. Fowler, 9000
H. E. Viney, 9130
R. C. Maydew, 9320
L. A. Hopkins, Jr., 9500
V. E. Blake, 9510
H. E. Hansen, 9511
S. L. Jeffers, 9512
S. McAlees, Jr., 9513
J. D. Appel, 9514
J. D. Appel, 9514 (ARPIC) (2)

DISTRIBUTION (cont):

R. J. Everett, 9515
A. J. Clark, 9520
J. W. McKiernan, 9521
J. Jacobs, 9522
B. F. Hefley, 8232 (5)
W. J. Wagoner, 3413
 For: DTIE (3)
 M. S. Goldstein
B. R. Allen, 3421
C. H. Sproul, 3428-2 (10)