UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

SOURCE_RX: A MULTICS-FORTRAN PROGRAM FOR STRATIGRAPHIC AND PETROLEUM GEOCHEMICAL DATA

Tom J. Clisham

and

Dan R. Grogan

Denver, Colorado

July, 1979

Open-File Report 79-1521

1979

TABLE OF CONTENTS

INTRODUCTION

A quantitative analysis of sedimentary rocks with possible present or past petroleum-generating capability is based on the collection of large numbers of rock samples. The need to manage a high volume of data generated thereby was the impetus for the Source_rx computer program. This program provides a format for data storage, a retrieval routine with listings that reflect stratigraphic ordering, basic statistics, and plots of selected variables. It is currently available on the U.S. Geological Survey Honeywell Multics[1] system at Denver in the FORTRAN V language.

This paper includes a discussion of sample data input and output, a summary of program functions, instructions for program execution, and a copy of the documented program. It is directed towards two interests: first, toward the person who uses data similar to Source_rx data; and second, toward the person who is looking for an example of this type of Multics utilization.

ACKNOWLEDGMENTS

Many people helped make this program possible. Special acknowledgments are due to the following USGS colleagues: Thaddeus S. Dyman for project and moral support; Robert J. Cassidy for his Multics programming expertise and review of this manuscript; and William W. Mallory and Ellen W. Sharp for reviewing and editing.

---

[1]Use of brand or manufacture names in this report is for descriptive purposes only and do not constitute endorsement by the USGS.

INPUT

Most Source_rx data are numeric, but special codes are employed for non-numeric data such as formation name and lithology. The American Petroleum Institute (1970) state codes are used. The data format was designed to be compact and easy for both the programmer and the computer to utilize. A data file can be created in a reasonable time frame. For example, inputting 200 samples with 18 variables each requires approximately 16 man-hours.

Data are divided into three types:

(1) Locality

(2) Sample

(3) Comment

Tables 1 and 2 list the variables related to localities and samples, respectively. As used in the Source_rx data file each locality record is allotted a maximum of 100 sample records. A locality record with no sample records can also be entered. The third type of record affords the user the capability of entering comments. By placing the number "3" in column one, the remaining 132 columns can be used for alpha-numeric input.

Table 1.--LOCALITY DATA EXPLANATION
[N/A indicates not applicable]

Input record column	Variable as noted in output heading	Input characteristics[1]	Input example	Output example	Comments
1	N/A	flag	1	N/A	"1" designates locality record
2-5	LOCALITY	A/N	HYPE	HYPE	
6-7	STATE	code	41	TENNESSEE	
8-13	LATITUDE	integer	362115	36 21 15	
14-20	LONGITUDE	integer	8355	83 55	
21-24	LOW TAI	decimal number	2.5	2.5	
25-28	HIGH TAI	decimal number	3.0	3.0	
29-32	TOTAL THICKNESS	integer	2750	2750	
33	N/A	code	1 2	+ ?	optional uncertainty notations for total thickness
34-37	NON-CARBONATES METERS	integer	1250	1250	
38-41	CARBONATES METERS	integer	1500	1500	
42-46	MIN. OVERBURDEN METERS	integer	11000	11000	
47-51	MAX. OVERBURDEN METERS	integer	20500	20500	
52-53	OIL GENERATION AGE	code	9	TRIASSIC	
54-68	COUNTY	A/N	KNOX	KNOX	

[1] A/N = alpha-numeric; code = codes indexed in program; for decimal numbers, decimal point is included and uses one place.

Table 2.--SAMPLE DATA EXPLANATION
[N/A indicates not applicable]

Input record column	Variable as noted in output heading	Input characteristics[1]	Input example	Output example	Comments
1	N/A	flag	2	N/A	"2" designates sample record
2-9	SAMPLE	A/N	HYPE-250	HYPE-250	
10-11	SLOPE ANGLE	integer	15	15	slope angle of outcrop
12-13	DIP	integer	35	35	dip angle of strata
14-15	DEPTH METERS	integer	2	2	depth of sample collection
16-17	N/A	secondary lithologic modifier code	26	silty	
18-19	N/A	primary lithologic modifier code	22	petroliferous	
20-21	LITHOLOGY	code	5	limestone	
22-25	$CaCO_3$ %	decimal number	56.3	56.3	
26-30	P_2O_5 %	decimal number	21.36	21.36	
31-34	TOTAL ORGANIC CARBON TOC %	decimal number	1.89	1.89	samples with less than 0.5% TOC are noted by asterisk
35	TYPE	type of organic matter code	3	sapropelic	options: sapropelic, humic, amorphous
36-40	VOLATILE HYDROCARBON PPM	integer	15000	15000	
41-44	TOTAL PYROLYTIC HC YIELD %	decimal number	.253	.253	
N/A	TOTAL HYDROCARBON PPM	N/A	N/A	17530	Calculated in program--Volatile HYDROCARBON AND (TOTAL PYROLYTIC HC YIELD x 10^4)
45-48	PYROLYTIC HC/OC	decimal number	10.3	10.3	
N/A	VOLATILE HC/OC	N/A	N/A	79.37	Calculated in program--(VOLATILE HYDROCARBON/10^2)/TOC
49	N/A	code	1	?	optional uncertainty notation for Peak I temperature
50-52	TEMPERATURE PEAK (CELCIUS) I	integer	308	308	
53-55	TEMPERATURE PEAK (CELCIUS) II	integer	550	550	
56	N/A	code	1	?	optional uncertainty notation for Peak II temperature
57-58	YEAR	integer	78	1978	year of collection
59	FORMATION	code	3	Rogers Ls.	
60	N/A	code	1	FLOAT	optional float sample designation
61-68	N/A	A/N	HYPE-4	THIS SAMPLE EQUIVALENT TO HYPE-4	optional feature to note identical samples with different labels

[1] A/N = alpha-numeric; code = codes indexed in program; for decimal numbers, decimal point is included and uses one place.

FUNCTIONS

Initially the program reads the data set, does the necessary code conversions, and then writes the data in the specified format. Counts, averages, and standard deviations are performed on selected values (defined in the program) for each locality and sample. An International Mathematical and Statistical Library (IMSL) routine, currently available on Multics, is used to calculate standard deviations.

If the user elects, the program proceeds into the plotting routines. Plots are produced through calls to the Display Integrated Software System and Plotting Language (DISSPLA) library (also available on Multics) on a Tektronix terminal. DISSPLA requires relatively simple input and does most of the graph set-up internally. Two types of plots are produced, arithmetic and semilogarithmic. The arithmetic plots are:

Temperature of Peak II versus Pyrolytic Hydrocarbon/Organic Carbon

Total Organic Carbon versus $CaCO_3$

Total Organic Carbon versus P_2O_5

Only one semilogarithmic plot is produced:

Total Organic Carbon versus log of Total Hydrocarbons

Two special purpose features have been written into Source_rx.

(1) A comparison test checks lithologic description against the appropriate analytical data and prints a warning message when the two conflict. For example, if a sample is described as a limestone, but actually contains only 45 percent calcium carbonate, the warning is printed:

limestone <50% $CaCO_3$ - [sample number]

(2) The program also withholds data relating to samples with less than 0.5 percent total organic carbon from certain statistical calculations. Source_rx utilizes data derived from a thermal analysis technique discussed by Claypool and Reed (1976). According to Claypool (oral commun., 1977) those samples deficient in organic carbon can yield erroneous results.

For a more detailed documentation of the program, see the Appendix.

ACCESS AND OPERATING PROCEDURES

To transfer the program into the user's area, type:

```
cp >udd>Ends>TDyman>Source_rx.fortran
```

A documented program (as shown in the Appendix) can be obtained by following the same procedure, but substituting "Source_rx.doc" for "Source_rx.fortran".

The test data file can be obtained with:

```
cp >udd>Ends>TDyman>Source_rx.test
```

The program must be compiled by the user:

```
fortran Source_rx
```

After the compiling, a few commands are necessary (if they are not contained in the user's start_up.ec)

```
line_length 132
asr >iml>imsl -after working_dir
asr >iml>disspla -after working_dir
```

The first command sets the line length to 132 characters for the output listing. The last two commands add search rules which enable the program to call the IMSL and DISSPLA libraries. If the user is logged in on a Tektronix terminal, plot routines can be accessed by using the command:

```
setup_tektronix_tcs
```

To run the program on the test data, type:

```
Source_rx
```

The program will then prompt the user to state his choice of options. To obtain a line printer listing of the test run, type:

```
dp test.out
```

If the user is on a 132 character printer, he may obtain a terminal listing by typing:

```
pr test.out
```

To clean up the user's directory of files created by this program run, type:

```
dl Source_rx.test test.out
```

This test run exemplifies the current operating procedure. The Point Data program segment (see the Appendix) was not utilized in this test. It is presented in the documentation as an example of program modification, written to display a different data file.

REFERENCES

American Petroleum Institute, 1970, The API well number and standard state and county codes: American Petroleum Institute Bulletin, D12A, 81 p.

Claypool, G. E., and Reed, R. R., 1976, Thermal-analysis technique for source-rock evaluation; quantitative estimate of organic richness and effects of lithologic variation: American Association of Petroleum Geologists Bulletin, v. 60, no. 4, p. 608-612.

APPENDIX

```
*************************************************************
*************************************************************
*                   SOURCE_RX PROGRAM
*                   WRITTEN BY:
*         TOM J. CLISHAM,U. S. GEOLOGICAL SURVEY
*************************************************************
*************************************************************
c
c                   DICTIONARY
c
*************************************************************
*************************************************************
c
c                   Main arrays
c
*************************************************************
c
c  array (i,1-33)
c  input and output matrix
c
c         Following variables are locality related:
c
c  1 - locality     2 - state     3 - low tai     4 - high tai
c  5 - total thickness        6 - thickness uncertainty
c  7 - non-carbonates thickness         8 - carbonates thickness
c  9 - minimum overburden     10 - maximum overburden
c  11 - overburden age
c
c         Following variables are sample related:
c
c  12 - slope        13 - dip    14 - depth
c
c                   no array values 15-16
c
c  17 - lithology      18 - CaCO3         19 - P2O5
c  20 - organic carbon         21 - organic carbon type
c  22 - hydrocarbon    23 - total pyrolytic hydrocarbon yield
c  24 - total hydrocarbon
c  25 - pyrolytic hydrocarbon/organic carbon
c  26 - volatile hydrocarbon/organic carbon
c  27 - temperature 1 uncertainty
c  28 - temperature 1         29 - temperature 2
c  30 - temperature 2 uncertainty        31 - year
c  32 - formation      33 - float
c
*************************************************************
c
c  count(1-13,1-78)
c  counts
c
c  1,1 - locality   2,1-8 - locality variables
c  3,1 - samples   3,2 - samples/locality
c  4,1-9 - lithologies/locality
c  5,1-9 - lithologies        6,1-6 - formations/locality
```

```
c  7,1-6 - formations          8,75-78 - years/locality
c  9,75-78 - years             10,1 - samples/locality <.5%oc
c  10,2 - samples <.5% organic carbon
c  11,12-29 - sample variables
c  12,12-29 - sample variables/locality
c  13,1-51 - states
c
************************************************************
c
c  ave (1-3,1-29)
c  average value matrix
c
c  1,18-29 - sample variables/locality
c  2,1-8 - locality variables
c  3,12-29 - sample variables
c
************************************************************
c
c  ara (1-11,1-29)
c  lithology matrix of array variables 18-29
c
c  1-9,1-12 -  averages/lithology
c  1-9,18-29 -  sums/lithology
c  10,18-29 - averages/carbonates
c  11,18-29 - averages/(claystones + mudstones + siltstones)
c
c  cnt(1-9,18-29)
c  count used for above averages
c
************************************************************
c
c  sum (1-3,1-29)
c  sums
c
c  1,1-8 - locality variables
c  2,12-29 - sample variables
c  3,12-29 - sample variables/locatity
c
************************************************************
c
c  bray (isct,1-7)
c  matrix of locality values for IMSL tm
c
c  bray columns relate to array columns in this manner:
c
c                  bray                  array
c
c                  1-3                   3-5
c                  4-7                   7-10
c
************************************************************
c
c  cray (iact,1-10)
c  matrix of sample variables
```

```
c
c  cray columns relate to array values in this manner:
c
c                      cray                 array
c
c                      1-3                  18-20
c                      4-8                  22-26
c                      9-10                 28-29
c
**********************************************************
c
c  secray (ict,1-10)
c  matrix of sample variables/locality
c
c  secray values relate to array the same as cray values
c
**********************************************************
**********************************************************
c
c                      I/O VARIABLES AND USEAGE
c
c  iage                          amod
c  age     - age                 agemod1     - age
c  ageout                        agemod2       modifiers
c
c  cai1                          county  - county
c  cai2    - tai
c  caiout
c
c  data   - comments
c
c  eq - equivalence
c
c  fm
c  iform   - formation
c  iiform
c
c  iapprox   - approximation
c
c  ilith                         imod1
c  lith    - lithology           imod2    - lithology
c  litha                         mod1       modifiers
c  lithb                         mod2
c  lithc                         mod
c
c  ioc   - organic carbon <.5%
c
c  it1   - temperature 1 uncertainty
c  it2   - temperature 2 uncertainty
c
c  lat1                                   long1
c  lat2    - latitude                     long2    - longitude
c  lat3                                   long3
c
```

```
c   oage    - overburden age
c
c   octype    - organic carbon type
c   otype
c
c   quad    - quadrangle
c
c   sampno    - sample number
c   loc  - locality, Point Locality segment
c
c   state
c   istate    - state
c   instate
c
c   thick    - total thickness uncertainty
c
************************************************************
************************************************************
c
c                   FLAGS
c
c   idata - data file type
c   flag - record type                      iend - end of input file
c   ipage - full page                       iplot - plot option
c   irept - locality w/o samples            zero - array of zero length
c   iskip - skip IMSL call
c
************************************************************
************************************************************
c
c                             IMSL tm ARRAYS
c   see IMSL manual for explanation of use within library
c
c   s      st        sa        r         rt        ra
c   u      ut        ua        xm        xmt       xma
c
************************************************************
************************************************************
c
c                   PLOT ARRAYS
c
c   DISSPLA tm plots require minimum and maximum
c   values of arrays
c
c   c   - CaCO3               h      - total
c   cmax                      hmax     hydrocarbon
c   cmin                      hmin
c
c   name  - titles
c   namex  - x axis title
c   namey  - y axis title
c
c   Three organic carbon arrays are used, each one
c   relates to arrays with differing counts
```

```
c
c  o,o1,o2 - organic carbon
c  omax,o1max,o2max
c  omin,o1min,o2min
c
c  p  - phosphate                t  - temperature 2
c  pmax                          tmax
c  pmin                          tmin
c
c  z   - pyrolytic hydrocarbon/organic carbon
c  zmax
c  zmin
c
c  following values are counters for above plot arrays
c
c  nc                 no          np         nz
c
************************************************************
c
c                     Data File Notes
c
************************************************************
c
c  Locality records must precede related sample records
c  Comment records must follow related records
c
************************************************************
*
************************************************************
c                     MAIN PROGRAM
************************************************************
*
************************************************************
c
c Set up arrays
c
      dimension array(300,33),ave(3,33),count(13,78),sum(3,33),
     &ara(11,33),cnt(33,33),z(300),t(300),o(300),h(300),
     &o1(300),o2(300),p(300),c(300),state(51),amod(3),
     &age(13),iage(3)
c
c IMSL tm routine arrays and vectors.
c
      dimension xm(10),u(3,33),s(820),r(820),xmt(10),ut(3,10),
     &st(820),rt(820),bray(300,33),secray(100,10),cray(300,33),
     & xma(10),ua(3,10),sa(820),ra(820)
c
c Initialize necessary variables
c
      data i,iskip,irept,nz,no,np,nc,
     &zmin,tmin,hmin,omin,o1min,pmin,o2min,cmin,zmax,tmax,
     &hmax,omax,o1max,pmax,o2max,cmax/7*0,8*1000.,8*0./
c
c Initialize Point Locality Data routine variables
```

```
c
      data cai2out,ipage /2*0/
      character * 10 loc
c
c Define length of alphanumeric variables
c
      character*1 ioc,it1,it2,thick
      character * 3 cai2
      character * 6 agemod1,agemod2,amod
      character*8 sampno,eq
      character * 10 octype,otype(3)
      character*11 lith(9),iform(6),ilith,iiform,litha
      character*13 mod(30),imod1,imod2,oage,ageout,age
      character * 14 state,istate       ;  character * 15 county
      character*20 lithb,lithc,fm,quad
      character*50 name,namex,namey
      character* 132 data
c
c Wait function used for the plotting routines.
c
      external iowait(descriptors)
c
c Create repeatedly used formats
c
 50   format("0")
 270  format("0",123("*"),"Source_rx")
 340  format (1x)
c
c Zero locality routine variables and blank alphabetic variables
c
      istate = "              "  ;  ageout = "             "
      agemod1 = "      "    ;  agemod2 = "      "
      thick=(" ")   ;  ioc=(" ")
      it1=(" ")     ;  it2=(" ")
      oage=("             ")  ;  octype=("          ")
      imod1=("             ")  ;  imod2=("             ")
c
c Zero arrays to prevent computer erratics from
c  interferring with calculations
c
      do 2 id=1,33
      do 242 ic=1,300
      array(ic,id)=0.0        ;  bray(ic,id)=0.0
      cray(ic,id)=0.0
      z(ic)=0.0      ;  t(ic)=0.0
      o(ic)=0.0      ;  o1(ic)=0.0
      h(ic)=0.0      ;  o2(ic)=0.0
      p(ic)=0.0      ;  c(ic)=0.0
 242  continue
      do 203 ib=1,33
      cnt(id,ib)=0.0
 203  continue
      do 204 ic=1,3
      sum(ic,id)=0.0  ;  ave(ic,id)=0.0
```

```
      u(ic,id)=0.0
 204  continue
      do 205 ib=1,11
      ara(ib,id)=0.0
 205  continue
 2    continue
      do 211 ia = 1,10
      xn(ia)=0.0
      xmt(ia)=0.0    ;  xma(ia)=0.0
      do 292 ic=1,3
      ua(ic,ia)=0.0    ;  ut(ic,ia)=0.0
 292  continue
      do 207 ib=1,100
      secray(ib,ia)=0.0
 207  continue
 211  continue
      do 208 ia=1,13
      do 209 ib=1,78
      count(ia,ib)=0.0
 209  continue
 208  continue
c
c Define alphabetic variables.
c
c Lithologic modifiers
c
      mod(1)="arenaceous"             ;  mod(14)="intraclastic"
      mod(2)="argillaceous"           ;  mod(15)="laminated"
      mod(3)="bioclastic"             ;  mod(16)="limonitic"
      mod(4)="calcareous"             ;  mod(17)="massive"
      mod(5)="carbonaceous"           ;  mod(18)="micritic"
      mod(6)="cherty"                 ;  mod(19)="nodular"
      mod(7)="clastic"                ;  mod(20)="oolitic"
      mod(8)="concretionary"          ;  mod(21)="pelletal"
      mod(9)="crinoidal"              ;  mod(22)="petroliferous"
      mod(10)="dolomitic"             ;  mod(23)="phosphatic"
      mod(11)="fossiliferous"         ;  mod(24)="radiolarian"
      mod(12)="fossil fragmental"     ;  mod(25)="siliceous"
      mod(13)="glauconitic"           ;  mod(26)="silty"
c
c States by API code
c
      state(1)="ALABAMA"         ;  state(27)="NEVADA"
      state(2)="ARIZONA"         ;  state(28)="NEW HAMPSHIRE"
      state(3)="ARKANSAS"        ;  state(29)="NEW JERSEY"
      state(4)="CALIFORNIA"      ;  state(30)="NEW MEXICO"
      state(5)="COLORADO"        ;  state(31)="NEW YORK"
      state(6)="CONNECTICUT"     ;  state(32)="NORTH CAROLINA"
      state(7)="DELAWARE"        ;  state(33)="NORTH DAKOTA"
      state(8)="DIST. COLUMBIA";  state(34)="OHIO"
      state(9)="FLORIDA"         ;  state(35)="OKLAHOMA"
      state(10)="GEORGIA"        ;  state(36)="OREGON"
      state(11)="IDAHO"          ;  state(37)="PENNSYLVANIA"
      state(12)="ILLINOIS"       ;  state(38)="RHODE ISLAND"
```

```
      state(13)="INDIANA"      ;  state(39)="SOUTH CAROLINA"
      state(14)="IOWA"         ;  state(40)="SOUTH DAKOTA"
      state(15)="KANSAS"       ;  state(41)="TENESSEE"
      state(16)="KENTUCKY"     ;  state(42)="TEXAS"
      state(17)="LOUISIANA"    ;  state(43)="UTAH"
      state(18)="MAINE"        ;  state(44)="VERMONT"
      state(19)="MARYLAND"     ;  state(45)="VIRGINIA"
      state(20)="MASSACHUSETTS"     ;  state(46)="WASHINGTON"
      state(21)="MICHIGAN"     ;  state(47)="WEST VIRGINIA"
      state(22)="MINNESOTA"    ;  state(48)="WISCONSIN"
      state(23)="MISSISSIPPI"  ;  state(49)="WYOMING"
      state(24)="MISSOURI"     ;  state(50)="ALASKA"
      state(25)="MONTANA"      ;  state(51)="HAWAII"
      state(26)="NEBRASKA"
c
c Ages , modifiers, and lithologies
c
      amod(1)="LOWER"             ;  age(1)="PRECAMBRIAN"
      amod(2)="MIDDLE"            ;  age(2)="CAMBRIAN"
      amod(3)="UPPER"             ;  age(3)="ORDOVICIAN"
      lith(1)="claystone"         ;  age(4)="SILURIAN"
      lith(2)="mudstone"          ;  age(5)="DEVONIAN"
      lith(3)="shale"             ;  age(6)="MISSISSIPPIAN"
      lith(4)="siltstone"         ;  age(7)="PENNSYLVANIAN"
      lith(5)="limestone"         ;  age(8)="PERMIAN"
      lith(6)="dolomite"          ;  age(9)="TRIASSIC"
      lith(7)="phosphorite"       ;  age(10)="JURASSIC"
      lith(8)="chert"             ;  age(11)="CRETACEOUS"
      lith(9)="uncertain"         ;  age(12)="TERTIARY"
      lithb="carbonates"          ;  age(13)="QUATERNARY"
      lithc="clay,mud&siltstone"
c
c Formation codes and organic carbon types
c
      iform(1)="Redwall"        ;  otype (1)="amorphous"
      iform(2)="Lyons"        ;  otype (2)="humic"
      iform(3)="Rogers Ls"    ;  otype (3)="sapropelic"
c
c Query user for type of data
c
     write(42,610)
 610  format (1x,"Enter data type code"/1x,"Mixed data = 0"/
     &1x,"Point locality data only = 1"/1x,"For test enter 0")
      read (41,620) idata
 620  format (i1)
      if (idata.eq.1) go to 710
c
c If data is not point locality type, drops into main program
c
      open(1,file="Source_rx.test",form="formatted")
c
c Output file will be named test.out
c
      open(6,carriage = .true.,file="test.out",form="formatted")
```

```
c
c Initial counter and start of the program.
c
 1     i=i+1
c
c Determine type of data on current record
c
       read(1,10,end=3)flag
 10    format(f1.0)
c
c Send sample records to read
c
       if(flag.eq.2.0) go to 15
c
c Send comments to Comment loop
c
       if(flag.eq.3) go to 701
c
c Skip average loop if last locality had no samples
c
       if(irept.eq.1) go to 5
c
c Go to average loop before starting on next locality
c
       if(i.ne.1) go to 4
************************************************************
c
c                    Locality loop
c
************************************************************
c
c Initial read determined which type of data record
c To read that record according to the proper format,
c a backspace must be performed to get CPU to read record
c a second time.
c
 5     backspace 1
       read(1,20)  array(i,1),array(i,2),lat1,lat2,lat3,
       &long1,long2,long3,(array(i,j),j=3,11),county
 20    format(1x,a4,f2.0,2i2,a2,i3,i2,a2,2f4.1,f4.0,f1.0,
      &2f4.0,2f5.0,f2.0,a15)
c
c Convert codes
c
       if(array(i,6).eq.1) thick=("+")
       if(array(i,6).eq.2) thick=("?")
c
c Find proper state and age codes, count states
c
       do 802 j = 1,51
       if (array(i,2).eq.j) istate=state(j)
       if (array(i,2).eq.j) count(13,j) = count(13,j) + 1.0
       if(array(i,11).eq.j) oage = age(j)
 802   continue
```

```
c
c To save reentering data,single tai and overburden figures
c  were placed in their 1st respective column,duplicated below
c
      if(array(i,4).eq.0.0) array(i,4)=array(i,3)
      if(array(i,10).eq.0.0) array(i,10)=array(i,9)
c
c Locality counters
c
      count(1,1)=count(1,1)+1.0
      isct=count(1,1)
c
c Output locality information
c
      if(i.ne.1) go to 104
      write(6,30)  ;  write(6,340) ; write(6,350) ;  write(6,340)
      write(6,360)  ;  write(6,370) ; write(6,380)
 30   format("0","THIS PROGRAM WAS DESIGNED FOR STORAGE, RETRIEVAL
     & AND SIMPLE STATISTICS ON STRATIGRAPHIC DATA.")
 350  format(" ","SAMPLES ARE ARRANGED STRATIGRAPHICALLY, FLOAT
     & SAMPLES ARE PLACED AT THE BOTTOM OF THE LOCALITY.")
 360  format(" ","SAMPLES WITH LESS THAN .5% ORGANIC CARBON PRODUCE
     & ANOMALOUS VALUES OF Z AND TEMPERATURE PEAKS I AND II.")
 370  format(" ","THESE SAMPLES ARE DESIGNATED WITH AN ASTERISK
     & BEFORE THE ORGANIC CARBON VALUE.")
 380  format(" ","Z AND TEMPERATURE PEAK VALUES FOR THESE SAMPLES
     & ARE LEFT OUT OF ALL STATISTICS")
 104  write(6,280) array(i,1),istate,lat1,lat2,lat3,long1,
     &long2,long3,county
 280  format("1"/" ",27x,"LOCALITY",11x,"STATE",12x,"LATITUDE
     &LONGITUDE",7x,"COUNTY"/"0",29x,a4,11x,a14,5x,2(i2,1x),a2,5x,
     & i3,1x,i2,1x,a2,7x,a15)
      write(6,270)
      write(6,40) (array(i,j),j=3,5),thick,
     &(array(i,ja),ja=7,10),oage
 40   format("0",16x,"TOTAL",7x,"NON"/1x,"TAI VALUES    THICKNESS",
     & 5x,2("CARBONATES   "),"MIN. & MAX. OVERBURDEN  OIL GENERATION
     &AGE  "/1x,"LOW   HIGH",3(6x,"METERS"),14x,"METERS"/
     &"0",2(f4.1,2x),4x,f5.0,a1,2(7x,f5.0),12x,2(f6.0,5x),a13)
c
c Zero alphabetic locality variables.
c
      thick=" "  ; oage="              "
      istate = "              "
c
c Locality sums & counts.
c
      do 12 j=3,10
      if(array(i,j).eq.0.0) go to 12
      ja=j-2
      sum(1,ja)=sum(1,ja)+array(i,j)
      count(2,ja)=count(2,ja)+1.0
 12   continue
c
```

```
c Create an array for IMSL tm statistics use
c
      do 512 j=1,3
      ja=j+2
      if (array(i,ja).ne.0.0) go to 501
c
c IMSL tm uses this figure to designate missing values
c
      bray(isct,j) = -9999999.
      go to 512
 501  bray(isct,j)=array(i,ja)
 512  continue
      do 52 j=4,7
      ja=j+3
      if(array(i,ja).ne.0.0) go to 502
      bray(isct,j) = -9999999.
      go to 52
 502  bray(isct,j)=array(i,ja)
 52   continue
c
c Flag to check localities without samples
c
      irept=1
c
c Locality loop completed,return to program start
c
      go to 1
***********************************************************
c
c                    Comment loop
c
***********************************************************
 701  backspace 1
      read(1,702) data
 702  format(1x,a132)
      write(6,702) data
c
c Return to start of program
c
      go to 1
***********************************************************
c
c                    Sample loop
c
***********************************************************
 15   backspace 1
      read(1,60) sampno,(array(i,j),j=12,14),mod1,mod2,
     &(array(i,jb),jb=17,23),array(i,25),(array(i,ja),
     &ja=27,33),eq
 60   format(1x,a8,5i2,f2.0,f4.0,f5.0,f4.0,f1.0,f5.0,
     &2f4.0,f1.0,2f3.0,f1.0,f2.0,2f1.0,a8)
c
c Count samples.
c
```

```
      count(3,1)=count(3,1)+1.0
      count(3,2)=count(3,2)+1.0
      ict=count(3,2)
       iact=count(3,1)
c
c Count lithologies and formations
c
      do 32 j=1,9
      if(array(i,17).eq.j) count(4,j)=count(4,j)+1.0
      if (array(i,17).eq.j) count(5,j)=count(5,j)+1.0
      if (array(i,32).eq.j) count(6,j)=count(6,j)+1.0
      if (array(i,32).eq.j) count(7,j)=count(7,j)+1.0
c
c Find proper lithologies, formations and modifiers
c
      if (array(i,17).eq.j) ilith=lith(j)
      if (array(i,32).eq.j) iiform=iform(j)
 32   continue
      do 212 j=1,30
      if(mod1.eq.j) imod1=mod(j)
      if(mod2.eq.j) imod2=mod(j)
      if(array(i,21).eq.j) octype = otype(j)
 212  continue
c
c Count years here.
c
      if (array(i,31).eq.75) count(8,75)=count(8,75)+1.0
      if (array(i,31).eq.75) count(9,75)=count(9,75)+1.0
      if (array(i,31).eq.76) count(8,76)=count(8,76)+1.0
      if (array(i,31).eq.76) count(9,76)=count(9,76)+1.0
      if (array(i,31).eq.77) count(8,77)=count(8,77)+1.0
      if (array(i,31).eq.77) count(9,77)=count(9,77)+1.0
      if (array(i,31).eq.78) count(8,78)=count(8,78)+1.0
      if (array(i,31).eq.78) count(9,78)=count(9,78)+1.0
c
c Define output symbols
c
      if (array(i,20).lt.0.5.and.array(i,20).gt.0.0) ioc=("*")
      if (array(i,27).eq.1.0) it1=("?")
      if (array(i,30).eq.1.0) it2=("?")
c
c Calculate dependent values
c
c
c Total hydrocarbon calculation
c
      array(i,24)=array(i,22)+(array(i,23)*10**4)
c
c Volatile HC/OC calculated in following steps
c
      if(array(i,20).eq.0.0.or.array(i,22).eq.0) go to 11
      pchc=array(i,22)/10**2
      array(i,26)=pchc/array(i,20)
c
```

```
c Create lithology array & count
c
      do 302 j=1,9
       if (array(i,17).ne.j) go to 302
       do 332 ja=18,29
      if((array(i,20).lt.0.5).and.(ja.eq.25.or.ja.eq.28.
     & or.ja.eq.29))  go to 332
       if(array(i,ja).ne.0.0) cnt(j,ja)=cnt(j,ja)+1.0
       ara(j,ja)=array(i,ja)+ara(j,ja)
 332   continue
 302   continue
c
c Less than .5%oc routine,creating bray and secray arrays
c
 11    do 42 j=18,29
       if (array(i,j).eq.0.0) go to 500
       if((j.eq.20).and.(array(i,20).lt.0.5))
     &count(10,1)=count(10,1)+1.0
       if((j.eq.20).and.(array(i,20).lt.0.5))
     &count(10,2)=count(10,2)+1.0
       if((j.eq.25.or.j.eq.28.or.j.eq.29).and.
     &(array(i,20).lt.0.5)) go to 513
       sum(2,j)=sum(2,j)+array(i,j)
       count(11,j)=count(11,j)+1.0
       sum(3,j)=sum(3,j)+array(i,j)
       count(12,j)=count(12,j)+1.0
 500   if (j.eq.21.or.j.eq.27) go to 42
c
c Adjust array subscripts
c
 513   if (j.le.20) ja=j-17
       if (j.ge.22.and.j.le.26) ja=j-18
       if (j.eq.28.or.j.eq.29) ja=j-19
       if((array(i,20).lt.0.5).and.(j.eq.25.or.j.eq.28.or.j.eq.29))
      & go to 511
       if(array(i,j).ne.0.0) go to 503
 511   cray(iact,ja) = -9999999.
       secray(ict,ja) = -9999999.
       go to 42
 503   cray(iact,ja)=array(i,j)
       secray(ict,ja)=array(i,j)
 42    continue
c
c Output sample information
c
       if (count(3,2).ne.1.0) go to 14
       write(6,90)
  90    format("0"/"0",55x,"TOTAL"/" ","SAMPLE  DEGREE OF ...",5x,
      & "LITHOLOGY",4x,"CaCO3",4x,"P2O5   ORGANIC   VOLATILE",5x,
      & "TOTAL",7x,"TOTAL   PYROLYTIC  VOLATILE  TEMPERATURE"/" ",
      & "  &",5x,"SLOPE,DIP,DEPTH",7x,"&",10x,"%",8x
      & ,"%",4x,"CARBON  HYDROCARBON  PYROLYTIC  HYDROCARBON
      &HC/OC",6x,"HC    PEAK (CELSIUS)"/" ", " YEAR",3x,"ANGLE",3x,
      & "METERS",4x, "FORMATION",20x,"TOC %",8x,"PPM",6x,"HC YIELD",
```

```
     & 7x,"PPM",8x,"Z",8x,"/OC",7x,"I",5x,"II"/1x,55x,"& TYPE",17x,
     & "H %")
 14   write(6,50)
      write (6,100) imod1,imod2
 100  format (" ",18x,2(a13,2x))
      write(6,110) sampno,(array(i,j),j=12,14),ilith,(array
     &(i,ja),ja=18,19),ioc,array(i,20),(array(i,jb),jb=22,26),it1,
     &array(i,28),array(i,29),it2,array(i,31),iiform,octype
 110  format(" ",a8,2x,3(i2,3x),a11,3x,f5.1,3x,f6.2,
     & 3x,a1,f5.2,2x,f7.0,7x,f5.3,4x,f6.0,5x,f6.2,3x,f7.3,
     & 2x,a1,1x,f4.0,2x,f4.0,1x,a1,/1x," 19",f3.0,19x,a11,
     &3x,a10)
      if(array(i,33).ne.0.0) write(6,130)
 130  format(" ", "FLOAT")
      if(eq.eq."          ") go to 44
      write(6,120) eq
 120  format(" ","THIS SAMPLE STRATIGRAPHICALLY EQUIVALENT TO:
     & ",a8)
c
c Check data for possible lithologic description errors.
c
 44   if ((array(i,17).eq.5).and.(array(i,18).lt.50.))
     &write(6,70) sampno
 70   format(1x,"limestone < 50% caco3-", a8)
      if ((array(i,17).eq.7).and.(array(i,19).lt.5.))
     &  write (6,85) sampno
 85   format(1x,"phosphorite < 5% P2O5 - ",a8)
c
c  Create arrays for plots
c
     if (array(i,25).eq.0.0.or.array(i,29).eq.0.0) go to 342
      nz = nz + 1
      z(nz) = array(i,25)
      t(nz) = array(i,29)
 342  if (array(i,20).eq.0.0.or.array(i,24).eq.0.0) go to 352
      no = no + 1
      o(no)=array(i,20)
      h(no)=array(i,24)
 352  if (array(i,20).eq.0.0.or.array(i,19).eq.0.0) go to 362
      np = np + 1
      o1(np)=array(i,20)
      p(np)=array(i,19)
 362  if(array(i,18).eq.0.0.or.array(i,20).eq.0.0) go to 372
      nc = nc + 1
      o2(nc)=array(i,20)
      c(nc)=array(i,18)
c
c Zero necessary sample variables
c
 372  ilith=(" ")      ;  iiform=(" ")
      ioc=(" ")       ;  octype=("          ")
      it1=(" ")       ;  it2=(" ")
      imod1="             " ;  imod2="             "
c
```

```
c Flag indicates locality with sample data
c
      irept=0
c
c Sample loop complete, return to start of program
c
      go to 1
c
c End flag
c
 3    iend=1
*************************************************************
c
c                     Averaging loop for each locality
c
*************************************************************
c
c Skip average routine if locality has too few values.
c
 4    if (count(3,2).le.1.0) go to 92
      do 62 j=18,29
      if(count(12,j).eq.0.0) go to 22
      ave(1,j)=sum(3,j)/count(12,j)
c
c Write averages.
c
 22   if (j.ne.21.and.j.ne.27.and.count(12,j).eq.0.0) zero=1
 62   continue
      write(6,270)    ;  write(6,50)
      write(6,150)
      write(6,160)(ave(1,ja),ja=18,20),(ave(1,jb),jb=22,26),
     &(ave(1,jc),jc=28,29)
 150  format("0",1x,"averages with <.5%oc figures left
     & out of z average and temperature peak averages")
 160  format("0",39x,f5.1,3x,f6.2,4x,f5.2,2x,f7.0,7x,f5.3,4x,
     &f6.0,5x,f6.2,3x,f7.3,4x,f4.0,2x,f4.0)
      if (zero.eq.1) write (6,24)
 24   format(1x," Standard Deviations skipped
     & because of <.5% OC samples or too few values")
      if(count(10,1).ne.0.0) write(6,24)
      if(count(10,1).ne.0.0) iskip=1
      if (zero.eq.1.or.iskip.eq.1) go to 54
c
c Standard deviations for current locality.
c calculated through an IMSL tm routine
c
      call bemmi(secray,ict,10,100,xm,u,s,r,ier)
      write(6,400)
      write(6,160) (u(1,j),j=1,10)
 400  format("0","standard deviations:")
c
c write counts for current locality.
c
 54   write(6,170) count(3,2),count(10,1)
```

```
 170  format("0","number of samples in this locality:",f5.0,
     &5x,"with",f4.0,2x,"samples under .5% organic carbon")
      do 72 j=1,9
      if(count(4,j).eq.0.0) go to 72
      write(6,180) lith(j),count(4,j)
 180   format(9x,a11,"=",f4.0)
 72   continue
      do 82 j=1,6
      if(count(6,j).eq.0.0)  go to 82
      write(6,190)iform(j),count(6,j)
 190   format(20x,a11,"=",f4.0)
 82   continue
      do 92 j=75,78
      if (count(8,j).eq.0.0) go to 92
      write(6,200) j,count(8,j)
 200  format (1x,"19",i2,"  samples = ",f3.0)
 92   continue
c
c Zero all counters.
c
      count(10,1)=0.0           ;  iskip=0
      count(3,2)=0.0            ;  zero=0.0
      do 232 j=12,29
      count(12,j)=0.0
      ave(1,j)=0.0
      sum(3,j)=0.0
 232  continue
      do 222 j=75,78
      count(8,j)=0.0
 222  continue
      do 102 j=1,9
      count(4,j)=0.0
      count(6,j)=0.0
 102  continue
      do 312 j=1,100
      do 322 ja=1,10
      secray(j,ja)=0.0
 322  continue
 312  continue
c
c Go to final calculations or to read another locality record
c
      if (iend.eq.1) go to 13
      go to 5
c
**************************************************************
c                     Calculate final averages
c
**************************************************************
 13   do 112 j=1,8
      if((sum(1,j).eq.0.0).or.(count(2,j).eq.0.0)) go to 112
      ave(2,j)=sum(1,j)/count(2,j)
 112  continue
      do 122 j=18,29
```

```
      if(sum(2,j).eq.0.0.or.count(11,j).eq.0.0) go to 122
      ave(3,j)=sum(2,j)/count(11,j)
 122  continue
c
c IMSL tm routine call
c
      call bemmi(bray,isct,7,300,xmt,ut,st,rt,ier)

      call bemmi(cray,iact,10,300,xma,ua,sa,ra,ier)
c
c Print counts of total localities,total samples.
c
      write(6,210)count(1,1),count(3,1)
 210  format("1"/" ",50x,"Total localities...",f4.0/51x,
     &"Total samples...",f5.0/51x,"FINAL AVERAGES...")
      write(6,270)  ;  write(6,270)  ;  write(6,150)
c
c Write final locality values
c
      write(6,80) (ave(2,j),j=1,3),(ave(2,ja),ja=5,8)
 80   format("0",16x,"TOTAL",7x,"NON-"/,1x,"TAI VALUES  THICKNESS
     & CARBONATES & CARBONATES  MIN. & MAX. OVERBURDEN "/
     &1x,"LO    HI  ",6x,"METERS",8x,"METERS",6x,"METERS",
     &14x,"METERS"/"0",2(f4.1,2x),4x,f5.0,2x,2(7x,f5.0),7x,
     &2(f6.0,5x))
      write(6,400)
      write(6,510) (ut(1,j),j=1,7)
 510  format("0",2(f4.1,2x),4x,f5.0,2x,2(7x,f5.0),7x,2(f6.0,5x))
      write(6,220)(count(2,j),j=1,3),(count(2,ja),ja=5,8)
 220  format("0","with these counts:"/"0",2(1x,f4.0),5x,
     &f4.0,9x,f4.0,3(8x,f4.0))
c
c Write final sample values
c
      write (6,90)
      write(6,160) (ave(3,ja),ja=18,20),(ave(3,jb),jb=22,26),
     &(ave(3,jc),jc=28,29)
      write(6,400)
      write(6,160) (ua(1,j),j=1,10)
c
c Write total number of samples of less than .5% organic carbon.
c
      write(6,600) count(10,2)
 600  format(1x,"number of samples < .5% organic carbon =" f4.0)
      write(6,50)
c
c Write total number of samples of each lithology.
c
      do 132 j=1,9
      if(count(5,j).eq.0) go to 132
      write(6,180) lith(j),count(5,j)
 132  continue
c
c Write total number of samples of each formation.
```

```
c
      do 142 j=1,6
      if(count(7,j).eq.0) go to 142
      write(6,190) iform(j),count(7,j)
  142 continue
c
c Write total number of samples collected in each year.
c
      do 152 j=75,78
      if(count(9,j).eq.0.0) go to 152
      write(6,200)j,count(9,j)
  152 continue
c
c Print averages for all lithologies and selected lithologic
c groups.
c
      write(6,240)
      write(6,90)
  240 format("0",27x,"averages per lithologies:")
      do 172 j=1,9
      do 162 ja=18,29
      jc=ja-17
      if(ara(j,ja).eq.0.0.or.cnt(j,ja).eq.0.0) go to 162
      ara(j,jc)=ara(j,ja)/cnt(j,ja)
  162 continue
  172 continue
      do 182 j=1,9
      if (count(5,j).eq.0.0) go to 182
      litha=lith(j)
      write(6,250) litha,(ara(j,ja),ja=1,3),(ara(j,jb),jb=5,9),
     &(ara(j,jc),jc=11,12)
      litha=" "
  250 format("0",9x,a20,10x,f5.1,3x,f6.2,4x,f5.2,2x,f7.0,
     &7x,f5.3,4x,f7.0,4x,f6.2,4x,f7.3,4x,f4.0,2x,f4.0)
  182 continue
      do 192 j=18,29
      if ((cnt(5,j)+cnt(6,j)).eq.0.0) go to 192
      ara(10,j)=((ara(5,j)+ara(6,j))/(cnt(5,j)+cnt(6,j)))
  192 continue
      write(6,250) lithb,(ara(10,ja),ja=18,20),
     &(ara(10,jb),jb=22,26),(ara(10,jc),jc=28,29)
      do 202 j=18,29
      if ((cnt(1,j)+cnt(2,j)+cnt(4,j)).eq.0.0) go to 202
      ara(11,j)=((ara(1,j)+ara(2,j)+ara(4,j))/(cnt(1,j)+
     &cnt(2,j)+cnt(4,j)))
  202 continue
      write(6,250) lithc,(ara(11,j),j=18,20),
     &(ara(11,ja),ja=22,26),(ara(11,jb),jb=28,29)
c
c Last line of hard copy output.
c
      write(6,270)
************************************************************
c
```

```
c                       Plot loop
c
*************************************************************
c
c Check for plot option
c
      write(42,470)
 470  format(1x,"Do you want plots, please answer yes or no")
      read(41,480) iplot
 480  format(a2)
      if (iplot.eq."no") go to 1999
      do 420 11 = 1,nz
      if (z(11).ge.zmax) zmax=z(11)
      if (z(11).le.zmin) zmin=z(11)
      if (t(11).ge.tmax) tmax=t(11)
      if (t(11).le.tmin) tmin=t(11)
 420  continue
c
c Peak II versus Z plot call
c
      namex="TEMPERATURE MAXIMUM PEAK II DEGREES CELCIUS$"
      name="TEST DATA$"
      namey="PYROLYTIC HC/OC, or Z$"
      call plot(tmin,zmin,tmax,zmax,nz,t,z,name,namex,namey)
c
c Hydrocarbon versus organic carbon plot setup
c
      namex="ORGANIC CARBON % $"
      namey="LOG OF TOTAL HYDROCARBON PPM $"
      do 430 1 = 1,no
      omin=amin1(o(1),omin)
      omax=amax1(o(1),omax)
      hmin=amin1(h(1),hmin)
      hmax=amax1(h(1),hmax)
 430  continue
      call slplot(omin,hmin,omax,hmax,no,o,h,name,namex,namey)
c
c Phosphate versus organic carbon plot setup
c
      do 440 12 = 1,np
      o1min=amin1(o1(12),o1min)
      o1max=amax1(o1(12),o1max)
      pmin=amin1(p(12),pmin)
      pmax=amax1(p(12),pmax)
 440  continue
      namey="P205 % $"
      call plot(o1min,pmin,o1max,pmax,np,o1,p,name,namex,namey)
      do 460 13 = 1,nc
      o2min=amin1(o2(13),o2min)
      o2max=amax1(o2(13),o2max)
      cmin=amin1(c(13),cmin)
      cmax=amax1(c(13),cmax)
 460  continue
c
```

```
c Calcium carbonate versus organic carbon plot call
c
      namey="CaCO3 % $"
      call plot(o2min,cmin,o2max,cmax,nc,o2,c,name,namex,namey)
c
c End of plot loop, skip Point locality loop
c and go to end
c
      go to 1999
*******************************************************
c
c                    Point Locality Data Type Routine
c
*******************************************************
 710  open(1,file="_________.data",form="formatted")
      open(6,file="__________.out",carriage=.true.,
     &form="formatted",defer=.true.)
 711  i = i+1
      read(1,750,end=721) loc,lat1,lat2,lat3,long1,long2,
     &long3,iapprox,instate,(iage(j),j=1,3),cai1,cai2,fm,
     &quad,county
 750  format(a10,2i2,a2,i3,i2,a2,i1,i2,3i1,f3.1,a3,2a20,a15)
c
c Sum and count and define output variables
c
      do 720 j=1,51
      if(instate.eq.j) count(1,j) = count(1,j)+1
      if(instate.eq.j) istate = state(j)
 720  continue
      if(cai2.ne."   ") decode(cai2,712) cai2out
 712  format(f3.1)
      if(cai1.ne.0.) count(2,1) = count(2,1)+1.
      if(cai2.ne."   ") count(2,2) = count(2,2)+1.
      sum(1,1) = sum(1,1)+cai1
      sum(1,2) = sum(1,2)+cai2out
      do 730 j = 1,9
      if(iage(3).eq.j) count(3,j)=count(3,j)+1
      if(iage(3).eq.j) ageout = age(j)
 730  continue
c
c Age modifiers loop
c
      do 722 j=1,3
      if(iage(1).eq.j) agemod1 = amod(j)
      if(iage(2).eq.j) agemod2 = amod(j)
 722  continue
c
c Output
c
      if(i.ne.1) go to 723
      write(6,751)     ;  write(6,752)     ;  write(6,753)
      write(6,754)    ;  write(6,760)
 751  format(5("0"/))
 752  format("0",40x,"DATA FROM ________ _________")
```

```
753  format("0",20x,"EXAMPLE OF THE SOURCE_RX PROGRAM FOR
    & RETRIEVAL OF POINT LOCALITY DATA")
c
c New page
c
754  format("1")
723  ipage = ipage + 1
     write(6,756) ; write(6,757) ; write(6,758) ; write(6,759)
     write(6,770) loc,istate,county,quad,lat1,lat2,lat3,
    &long1,long2,long3
    if(i.eq.1) write (6,771)
     write(6,760) ; write(6,761) ; write(6,762) ; write(6,763)
771  format(" ",76x,"approximately")
756  format(" ","  SAMPLE")
757  format(" ","   NAME")
758  format("+",18x,"STATE")
759  format("+",34x,"COUNTY",12x,"QUADRANGLE",9x,"LATITUDE
    &  LONGITUDE")
760  format("0")
761  format("0",49x,"AGE")
762  format("+",72x,"FORMATION")
763  format("+",93x,"CAI RANGE")
770  format("0",a10,4x,a14,1x,a12,4x,a20,2x,2(i2,2x),a2,
    &3x,i3,2x,i2,2x,a2)
     write(6,764) agemod1,agemod2,ageout,fm
764  format("0",35x,2(a6,2x),a13,4x,a20)
     if(cai1.eq.0) write(6,755)
755  format("+",92x,"not determined")
     if(cai1.ne.0) write(6,774) cai1
774  format("+",84x,f3.1)
     if(cai2.ne."    ") write(6,765) cai2out
765  format("+",97x," - ",f3.1)
     cai2out = 0.
     agemod1 = "      "
     agemod2 = "      "
     ageout = "             "
     write(6,270)    ;  write(6,760)
     if (ipage.eq.4) write(6,754)
     if (ipage.eq.4) ipage = 0
     go to 711
c
c File read, now average and end.
c
721  ave(1,1) = sum(1,1)/count(2,1)
     ave(1,2) = sum(1,2)/count(2,2)
c
c Output averages and counts
c
     write(6,754)     ;  write(6,751)     ;  write(6,766)
766  format("0",57x,"COUNTS"/"0")
     write(6,758)     ;  write(6,760)
     do 725 j=1,51
     if (count(1,j).ne.0.) write(6,767) state(j),count(1,j)
767  format(" ",14x,a14," = ",f4.0)
```

```
 725  continue
      write(6,760)     ;  write(6,761)     ;  write(6,760)
      do 726 j=1,9
      if(count(3,j).ne.0.) write(6,768) age(j),count(3,j)
 768  format(44x,a14," = ",f4.0)
 726  continue
      write(6,773) count(2,1),count(2,2)
 773  format ("0",80x,"CAI COUNTS"/"0",80x,"LOW  HIGH"/
     &"0",80x,f3.1,2x,f3.1)
c
c Output cai averages
c
      write(6,760)     ;  write(6,769)
      write(6,772) (ave(1,j),j=1,2)
 769  format("0",75x,"AVERAGE CAI RANGE")
 772  format("0",80x,2(f3.1,2x))
      write(6,751)     ;  write(6,270)
c
c End of Point Locality type loop
c
 ***********************************************************
c
c                   Close all files
c
***********************************************************
 1999 close(6)
      close(1)
      close(41)
      close(42)
      stop
      end
***********************************************************
c
c                   Plot routine using DISSPLA tm
c
***********************************************************
      subroutine plot(xmin,ymin,xmax,ymax,ncnt,x,y,label,
     &labelx,labely)
      dimension x(300),y(300)
      call tk120
      call marker (4)
      call axsplt(xmin,xmax,6.,xor,xstep,xaxis)
      call axsplt(ymin,ymax,9.,yor,ystep,yaxis)
      call title(label,100,labelx,100,labely,100,6.,9.)
      call graph (xor,xstep,yor,ystep)
      call grid (1,1)
      call curve (x,y,ncnt,-1)
      call iowait(100)
      call endpl (-1)
      return
      end
***********************************************************
c
c               Semilog plot routine using DISSPLA tm
```

```
c
 *****************************************************************
       subroutine slplot (xmin,ymin,xmax,ymax,ncnt,x,y,label,
      &labelx,labely)
       dimension x(300),y(300)
       call marker (4)
       call axsplt(xmin,xmax,6.,xor,xstep,xaxis)
       call algplt(ymin,ymax,9.,yor,cycle)
       call title (label,100,labelx,100,labely,100,6.,9.)
       call ylog(xor,xstep,yor,cycle)
       call grid (1,1)
       call curve(x,y,ncnt,-1)
       call iowait (100)
       call endpl (-2)
       return
       end
```