

G100296

SER FC
USGS
OFR
79-1521

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

UNIVERSITY OF UTAH
RESEARCH INSTITUTE
EARTH SCIENCE LAB.

Open-File Report 79-1521

1979

TABLE OF CONTENTS

| | <u>Page</u> |
|---------------------------------|-------------|
| Introduction | 1 |
| Acknowledgments | 1 |
| Input | 2 |
| Table 1 | 3 |
| Table 2 | 4 |
| Functions | 5 |
| Access and Operating Procedures | 6 |
| References | 8 |
| Appendix | 9 |

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 Multics¹ 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.

¹Use 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 | |

¹A/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 ₂ O 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 |

¹A/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.

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

```

1 *****
2 *****
3 *           SOURCE_RX PROGRAM
4 *           WRITTEN BY:
5 *           TOM J. CLISHAM,U. S. GEOLOGICAL SURVEY
6 *****
7 *****
8 c
9 c           DICTIONARY
10 c
11 *****
12 *****
13 c
14 c           Main arrays
15 c
16 *****
17 c
18 c   array (i,1-33)
19 c   input and output matrix
20 c
21 c           Following variables are locality related:
22 c
23 c   1 - locality      2 - state      3 - low tai      4 - high tai
24 c   5 - total thickness      6 - thickness uncertainty
25 c   7 - non-carbonates thickness      8 - carbonates thickness
26 c   9 - minimum overburden      10 - maximum overburden
27 c   11 - overburden age
28 c
29 c           Following variables are sample related:
30 c
31 c   12 - slope      13 - dip      14 - depth
32 c
33 c           no array values 15-16
34 c
35 c   17 - lithology      18 - CaCO3      19 - P2O5
36 c   20 - organic carbon      21 - organic carbon type
37 c   22 - hydrocarbon      23 - total pyrolytic hydrocarbon yield
38 c   24 - total hydrocarbon
39 c   25 - pyrolytic hydrocarbon/organic carbon
40 c   26 - volatile hydrocarbon/organic carbon
41 c   27 - temperature 1 uncertainty
42 c   28 - temperature 1      29 - temperature 2
43 c   30 - temperature 2 uncertainty      31 - year
44 c   32 - formation      33 - float
45 c
46 *****
47 c
48 c   count(1-13,1-78)
49 c   counts
50 c
51 c   1,1 - locality      2,1-8 - locality variables
52 c   3,1 - samples      3,2 - samples/locality
53 c   4,1-9 - lithologies/locality
54 c   5,1-9 - lithologies      6,1-6 - formations/locality

```

```

55 c 7,1-6 - formations          8,75-78 - years/locality
56 c 9,75-78 - years           10,1 - samples/locality <.5%oc
57 c 10,2 - samples <.5% organic carbon
58 c 11,12-29 - sample variables
59 c 12,12-29 - sample variables/locality
60 c 13,1-51 - states
61 c
62 *****
63 c
64 c ave (1-3,1-29)
65 c average value matrix
66 c
67 c 1,18-29 - sample variables/locality
68 c 2,1-8 - locality variables
69 c 3,12-29 - sample variables
70 c
71 *****
72 c
73 c ara (1-11,1-29)
74 c lithology matrix of array variables 18-29
75 c
76 c 1-9,1-12 - averages/lithology
77 c 1-9,18-29 - sums/lithology
78 c 10,18-29 - averages/carbonates
79 c 11,18-29 - averages/(claystones + mudstones + siltstones)
80 c
81 c cnt(1-9,18-29)
82 c count used for above averages
83 c
84 *****
85 c
86 c sum (1-3,1-29)
87 c sums
88 c
89 c 1,1-8 - locality variables
90 c 2,12-29 - sample variables
91 c 3,12-29 - sample variables/locality
92 c
93 *****
94 c
95 c bray (isct,1-7)
96 c matrix of locality values for IMSL tm
97 c
98 c bray columns relate to array columns in this manner:
99 c
100 c          bray          array
101 c
102 c          1-3          3-5
103 c          4-7          7-10
104 c
105 *****
106 c
107 c cray (iact,1-10)
108 c matrix of sample variables

```

```

109 c
110 c   cray columns relate to array values in this manner:
111 c
112 c           cray           array
113 c
114 c           1-3           18-20
115 c           4-8           22-26
116 c           9-10          28-29
117 c
118 c *****
119 c
120 c   secray (ict,1-10)
121 c   matrix of sample variables/locality
122 c
123 c   secray values relate to array the same as cray values
124 c
125 c *****
126 c *****
127 c
128 c           I/O VARIABLES AND USEAGE
129 c
130 c   iage           amod
131 c   age - age     agemod1 - age
132 c   ageout        agemod2   modifiers
133 c
134 c   cail           county - county
135 c   cai2 - tai
136 c   caiout
137 c
138 c   data - comments
139 c
140 c   eq - equivalence
141 c
142 c   fm
143 c   iform - formation
144 c   iiform
145 c
146 c   iapprox - approximation
147 c
148 c   ilith           imod1
149 c   lith - lithology   imod2 - lithology
150 c   litha           mod1   modifiers
151 c   lithb           mod2
152 c   lithc           mod
153 c
154 c   ioc - organic carbon <.5%
155 c
156 c   it1 - temperature 1 uncertainty
157 c   it2 - temperature 2 uncertainty
158 c
159 c   lat1           long1
160 c   lat2 - latitude   long2 - longitude
161 c   lat3           long3
162 c

```

```

163 c oage - overburden age
164 c
165 c octype - organic carbon type
166 c otype
167 c
168 c quad - quadrangle
169 c
170 c sampno - sample number
171 c loc - locality, Point Locality segment
172 c
173 c state
174 c ystate - state
175 c instate
176 c
177 c thick - total thickness uncertainty
178 c
179 c *****
180 c *****
181 c
182 c          FLAGS
183 c
184 c  idata - data file type
185 c  flag - record type          iend - end of input file
186 c  ipage - full page          iplot - plot option
187 c  irept - locality w/o samples zero - array of zero length
188 c  iskip - skip IMSL call
189 c
190 c *****
191 c *****
192 c
193 c          IMSL tm ARRAYS
194 c  see IMSL manual for explanation of use within library
195 c
196 c  s      st      sa      r      rt      ra
197 c  u      ut      ua      xm      xmt     xma
198 c
199 c *****
200 c *****
201 c
202 c          PLOT ARRAYS
203 c
204 c  DISSPLA tm plots require minimum and maximum
205 c  values of arrays
206 c
207 c  c      - CaCO3          h      - total
208 c  cmax          hmax      hydrocarbon
209 c  cmin          hmin
210 c
211 c  name - titles
212 c  namex - x axis title
213 c  namey - y axis title
214 c
215 c  Three organic carbon arrays are used, each one
216 c  relates to arrays with differing counts

```

```

217 c
218 c o,ol,o2 - organic carbon
219 c omax,olmax,o2max
220 c omin,olmin,o2min
221 c
222 c p - phosphate          t - temperature 2
223 c pmax                  tmax
224 c pmin                  tmin
225 c
226 c z - pyrolytic hydrocarbon/organic carbon
227 c zmax
228 c zmin
229 c
230 c following values are counters for above plot arrays
231 c
232 c nc          no          np          nz
233 c
234 c *****
235 c
236 c          Data File Notes
237 c
238 c *****
239 c
240 c Locality records must precede related sample records
241 c Comment records must follow related records
242 c
243 c *****
244 c *
245 c *****
246 c          MAIN PROGRAM
247 c *****
248 c *
249 c *****
250 c
251 c Set up arrays
252 c
253 c          dimension array(300,33),ave(3,33),count(13,78),sum(3,33),
254 c          &ara(11,33),cnt(33,33),z(300),t(300),o(300),h(300),
255 c          &ol(300),o2(300),p(300),c(300),state(51),amod(3),
256 c          &age(13),iage(3)
257 c
258 c IMSL tm routine arrays and vectors.
259 c
260 c          dimension xm(10),u(3,33),s(820),r(820),xmt(10),ut(3,10),
261 c          &st(820),rt(820),bray(300,33),secray(100,10),cray(300,33),
262 c          &xma(10),ua(3,10),sa(820),ra(820)
263 c
264 c Initialize necessary variables
265 c
266 c          data i,iskip,irept,nz,no,np,nc,
267 c          &zmin,tmin,hmin,omin,olmin,pmin,o2min,cmin,zmax,tmax,
268 c          &hmax,omax,olmax,pmax,o2max,cmax/7*0,8*1000.,8*0./
269 c
270 c Initialize Point Locality Data routine variables

```

```

271 c
272     data cai2out,ipage /2*0/
273     character * 10 loc
274 c
275 c Define length of alphanumeric variables
276 c
277     character*1 ioc,it1,it2,thick
278     character * 3 cai2
279     character * 6 agemod1,agemod2,amod
280     character*8 sampno,eq
281     character * 10 octype,otype(3)
282     character*11 lith(9),iform(6),ilith,iiform,litha
283     character*13 mod(30),imod1,imod2,oage,ageout,age
284     character * 14 state,istate      ; character * 15 county
285     character*20 lithb,lithc,fm,quad
286     character*50 name,namex,namey
287     character* 132 data
288 c
289 c Wait function used for the plotting routines.
290 c
291     external iowait(descriptors)
292 c
293 c Create repeatedly used formats
294 c
295 50   format("0")
296 270  format("0",123("*"),"Source_rx")
297 340  format (lx)
298 c
299 c Zero locality routine variables and blank alphabetic variables
300 c
301     istate = "          " ; ageout = "          "
302     agemod1 = "          " ; agemod2 = "          "
303     thick=(" ") ; ioc=(" ")
304     it1=(" ") ; it2=(" ")
305     oage=(" ") ; octype=(" ")
306     imod1=(" ") ; imod2=(" ")
307 c
308 c Zero arrays to prevent computer erratics from
309 c   interfering with calculations
310 c
311     do 2 id=1,33
312     do 242 ic=1,300
313     array(ic,id)=0.0 ; bray(ic,id)=0.0
314     cray(ic,id)=0.0
315     z(ic)=0.0 ; t(ic)=0.0
316     o(ic)=0.0 ; ol(ic)=0.0
317     h(ic)=0.0 ; o2(ic)=0.0
318     p(ic)=0.0 ; c(ic)=0.0
319 242  continue
320     do 203 ib=1,33
321     cnt(id,ib)=0.0
322 203  continue
323     do 204 ic=1,3
324     sum(ic,id)=0.0 ; ave(ic,id)=0.0

```



```

325      u(ic,id)=0.0
326  204  continue
327      do 205 ib=1,11
328      ara(ib,id)=0.0
329  205  continue
330  2    continue
331      do 211 ia = 1,10
332      xm(ia)=0.0
333      xmt(ia)=0.0 ; xma(ia)=0.0
334      do 292 ic=1,3
335      ua(ic,ia)=0.0 ; ut(ic,ia)=0.0
336  292  continue
337      do 207 ib=1,100
338      secray(ib,ia)=0.0
339  207  continue
340  211  continue
341      do 208 ia=1,13
342      do 209 ib=1,78
343      count(ia,ib)=0.0
344  209  continue
345  208  continue
346  c
347  c Define alphabetic variables.
348  c
349  c Lithologic modifiers
350  c
351      mod(1)="arenaceous" ; mod(14)="intraclastic"
352      mod(2)="argillaceous" ; mod(15)="laminated"
353      mod(3)="bioclastic" ; mod(16)="limonitic"
354      mod(4)="calcareous" ; mod(17)="massive"
355      mod(5)="carbonaceous" ; mod(18)="micritic"
356      mod(6)="cherty" ; mod(19)="nodular"
357      mod(7)="clastic" ; mod(20)="oolitic"
358      mod(8)="concretionary" ; mod(21)="pelletal"
359      mod(9)="crinoidal" ; mod(22)="petroliferous"
360      mod(10)="dolomitic" ; mod(23)="phosphatic"
361      mod(11)="fossiliferous" ; mod(24)="radiolarian"
362      mod(12)="fossil fragmental" ; mod(25)="siliceous"
363      mod(13)="glaucconitic" ; mod(26)="silty"
364  c
365  c States by API code
366  c
367      state(1)="ALABAMA" ; state(27)="NEVADA"
368      state(2)="ARIZONA" ; state(28)="NEW HAMPSHIRE"
369      state(3)="ARKANSAS" ; state(29)="NEW JERSEY"
370      state(4)="CALIFORNIA" ; state(30)="NEW MEXICO"
371      state(5)="COLORADO" ; state(31)="NEW YORK"
372      state(6)="CONNECTICUT" ; state(32)="NORTH CAROLINA"
373      state(7)="DELAWARE" ; state(33)="NORTH DAKOTA"
374      state(8)="DIST. COLUMBIA" ; state(34)="OHIO"
375      state(9)="FLORIDA" ; state(35)="OKLAHOMA"
376      state(10)="GEORGIA" ; state(36)="OREGON"
377      state(11)="IDAHO" ; state(37)="PENNSYLVANIA"
378      state(12)="ILLINOIS" ; state(38)="RHODE ISLAND"

```

```

379     state(13)="INDIANA"           ;   state(39)="SOUTH CAROLINA"
380     state(14)="IOWA"             ;   state(40)="SOUTH DAKOTA"
381     state(15)="KANSAS"           ;   state(41)="TENNESSEE"
382     state(16)="KENTUCKY"         ;   state(42)="TEXAS"
383     state(17)="LOUISIANA"        ;   state(43)="UTAH"
384     state(18)="MAINE"            ;   state(44)="VERMONT"
385     state(19)="MARYLAND"         ;   state(45)="VIRGINIA"
386     state(20)="MASSACHUSETTS"    ;   state(46)="WASHINGTON"
387     state(21)="MICHIGAN"         ;   state(47)="WEST VIRGINIA"
388     state(22)="MINNESOTA"        ;   state(48)="WISCONSIN"
389     state(23)="MISSISSIPPI"     ;   state(49)="WYOMING"
390     state(24)="MISSOURI"         ;   state(50)="ALASKA"
391     state(25)="MONTANA"          ;   state(51)="HAWAII"
392     state(26)="NEBRASKA"
393 c
394 c Ages , modifiers, and lithologies
395 c
396     amod(1)="LOWER"               ;   age(1)="PRECAMBRIAN"
397     amod(2)="MIDDLE"             ;   age(2)="CAMBRIAN"
398     amod(3)="UPPER"              ;   age(3)="ORDOVICIAN"
399     lith(1)="claystone"          ;   age(4)="SILURIAN"
400     lith(2)="mudstone"           ;   age(5)="DEVONIAN"
401     lith(3)="shale"              ;   age(6)="MISSISSIPPIAN"
402     lith(4)="siltstone"          ;   age(7)="PENNSYLVANIAN"
403     lith(5)="limestone"         ;   age(8)="PERMIAN"
404     lith(6)="dolomite"           ;   age(9)="TRIASSIC"
405     lith(7)="phosphorite"        ;   age(10)="JURASSIC"
406     lith(8)="chert"              ;   age(11)="CRETACEOUS"
407     lith(9)="uncertain"          ;   age(12)="TERTIARY"
408     lithb="carbonates"           ;   age(13)="QUATERNARY"
409     lithc="clay,mud&siltstone"
410 c
411 c Formation codes and organic carbon types
412 c
413     iform(1)="Redwall"           ;   otype (1)="amorphous"
414     iform(2)="Lyons"             ;   otype (2)="humic"
415     iform(3)="Rogers Ls"        ;   otype (3)="sapropelic"
416 c
417 c Query user for type of data
418 c
419     write(42,610)
420 610 format (lx,"Enter data type code"/lx,"Mixed data = 0"/
421 &lx,"Point locality data only = 1"/lx,"For test enter 0")
422     read (41,620) idata
423 620 format (il)
424     if (idata.eq.1) go to 710
425 c
426 c If data is not point locality type, drops into main program
427 c
428     open(1,file="Source_rx.test",form="formatted")
429 c
430 c Output file will be named test.out
431 c
432     open(6,carriage = .true.,file="test.out",form="formatted")

```

```

433 c
434 c Initial counter and start of the program.
435 c
436 l   i=i+1
437 c
438 c Determine type of data on current record
439 c
440     read(1,10,end=3)flag
441 l10  format(fl.0)
442 c
443 c Send sample records to read
444 c
445     if(flag.eq.2.0) go to 15
446 c
447 c Send comments to Comment loop
448 c
449     if(flag.eq.3) go to 701
450 c
451 c Skip average loop if last locality had no samples
452 c
453     if(irept.eq.1) go to 5
454 c
455 c Go to average loop before starting on next locality
456 c
457     if(i.ne.1) go to 4
458 *****
459 c
460 c           Locality loop
461 c
462 *****
463 c
464 c Initial read determined which type of data record
465 c To read that record according to the proper format,
466 c a backspace must be performed to get CPU to read record
467 c a second time.
468 c
469 l5    backspace 1
470     read(1,20) array(i,1),array(i,2),lat1,lat2,lat3,
471     &long1,long2,long3,(array(i,j),j=3,11),county
472 l20   format(lx,a4,f2.0,2i2,a2,i3,i2,a2,2f4.1,f4.0,fl.0,
473     &2f4.0,2f5.0,f2.0,a15)
474 c
475 c Convert codes
476 c
477     if(array(i,6).eq.1) thick=("+")
478     if(array(i,6).eq.2) thick=("?")
479 c
480 c Find proper state and age codes, count states
481 c
482     do 802 j = 1,51
483     if (array(i,2).eq.j)  istate=state(j)
484     if (array(i,2).eq.j)  count(13,j) = count(13,j) + 1.0
485     if(array(i,11).eq.j)  oage = age(j)
486 l802  continue

```

```

487 c
488 c To save reentering data, single tai and overburden figures
489 c were placed in their 1st respective column, duplicated below
490 c
491     if(array(i,4).eq.0.0) array(i,4)=array(i,3)
492     if(array(i,10).eq.0.0) array(i,10)=array(i,9)
493 c
494 c Locality counters
495 c
496     count(1,1)=count(1,1)+1.0
497     isct=count(1,1)
498 c
499 c Output locality information
500 c
501     if(i.ne.1) go to 104
502     write(6,30) ; write(6,340) ; write(6,350) ; write(6,340)
503     write(6,360) ; write(6,370) ; write(6,380)
504 30  format("0","THIS PROGRAM WAS DESIGNED FOR STORAGE, RETRIEVAL
505     & AND SIMPLE STATISTICS ON STRATIGRAPHIC DATA.")
506 350 format(" ","SAMPLES ARE ARRANGED STRATIGRAPHICALLY, FLOAT
507     & SAMPLES ARE PLACED AT THE BOTTOM OF THE LOCALITY.")
508 360 format(" ","SAMPLES WITH LESS THAN .5% ORGANIC CARBON PRODUCE
509     & ANOMALOUS VALUES OF Z AND TEMPERATURE PEAKS I AND II.")
510 370 format(" ","THESE SAMPLES ARE DESIGNATED WITH AN ASTERISK
511     & BEFORE THE ORGANIC CARBON VALUE.")
512 380 format(" ","Z AND TEMPERATURE PEAK VALUES FOR THESE SAMPLES
513     & ARE LEFT OUT OF ALL STATISTICS")
514 104 write(6,280) array(i,1),istate,lat1,lat2,lat3,long1,
515     &long2,long3,county
516 280 format("1"/" ",27x,"LOCALITY",11x,"STATE",12x,"LATITUDE
517     &LONGITUDE",7x,"COUNTY"/"0",29x,a4,11x,a14,5x,2(i2,1x),a2,5x,
518     & i3,1x,i2,1x,a2,7x,a15)
519     write(6,270)
520     write(6,40) (array(i,j),j=3,5),thick,
521     &(array(i,ja),ja=7,10),oage
522 40  format("0",16x,"TOTAL",7x,"NON"/1x,"TAI VALUES THICKNESS",
523     & 5x,2("CARBONATES "),"MIN. & MAX. OVERBURDEN OIL GENERATION
524     &AGE "/1x,"LOW HIGH",3(6x,"METERS"),14x,"METERS"/
525     &"0",2(f4.1,2x),4x,f5.0,a1,2(7x,f5.0),12x,2(f6.0,5x),a13)
526 c
527 c Zero alphabetic locality variables.
528 c
529     thick=" " ; oage=" "
530     istate = " "
531 c
532 c Locality sums & counts.
533 c
534     do 12 j=3,10
535     if(array(i,j).eq.0.0) go to 12
536     ja=j-2
537     sum(1,ja)=sum(1,ja)+array(i,j)
538     count(2,ja)=count(2,ja)+1.0
539 12  continue
540 c

```

```

541 c Create an array for IMSL tm statistics use
542 c
543     do 512 j=1,3
544         ja=j+2
545         if (array(i,ja).ne.0.0) go to 501
546 c
547 c IMSL tm uses this figure to designate missing values
548 c
549     bray(isct,j) = -9999999.
550     go to 512
551 501 bray(isct,j)=array(i,ja)
552 512 continue
553     do 52 j=4,7
554         ja=j+3
555         if(array(i,ja).ne.0.0) go to 502
556         bray(isct,j) = -9999999.
557         go to 52
558 502 bray(isct,j)=array(i,ja)
559 52 continue
560 c
561 c Flag to check localities without samples
562 c
563     irept=1
564 c
565 c Locality loop completed,return to program start
566 c
567     go to 1
568 *****
569 c
570 c             Comment loop
571 c
572 *****
573 701  backspace 1
574     read(1,702) data
575 702  format(lx,a132)
576     write(6,702) data
577 c
578 c Return to start of program
579 c
580     go to 1
581 *****
582 c
583 c             Sample loop
584 c
585 *****
586 15  backspace 1
587     read(1,60) sampno,(array(i,j),j=12,14),mod1,mod2,
588     &(array(i,jb),jb=17,23),array(i,25),(array(i,ja),
589     &ja=27,33),eq
590 60  format(lx,a8,5i2,f2.0,f4.0,f5.0,f4.0,f1.0,f5.0,
591     &2f4.0,f1.0,2f3.0,f1.0,f2.0,2f1.0,a8)
592 c
593 c Count samples.
594 c

```

```

595     count(3,1)=count(3,1)+1.0
596     count(3,2)=count(3,2)+1.0
597     ict=count(3,2)
598     iact=count(3,1)
599 c
600 c Count lithologies and formations
601 c
602     do 32 j=1,9
603     if(array(i,17).eq.j) count(4,j)=count(4,j)+1.0
604     if (array(i,17).eq.j) count(5,j)=count(5,j)+1.0
605     if (array(i,32).eq.j) count(6,j)=count(6,j)+1.0
606     if (array(i,32).eq.j) count(7,j)=count(7,j)+1.0
607 c
608 c Find proper lithologies, formations and modifiers
609 c
610     if (array(i,17).eq.j) ilith=lith(j)
611     if (array(i,32).eq.j) iiform=iiform(j)
612 32 continue
613     do 212 j=1,30
614     if(mod1.eq.j) imod1=mod(j)
615     if(mod2.eq.j) imod2=mod(j)
616     if(array(i,21).eq.j) octype = otype(j)
617 212 continue
618 c
619 c Count years here.
620 c
621     if (array(i,31).eq.75) count(8,75)=count(8,75)+1.0
622     if (array(i,31).eq.75) count(9,75)=count(9,75)+1.0
623     if (array(i,31).eq.76) count(8,76)=count(8,76)+1.0
624     if (array(i,31).eq.76) count(9,76)=count(9,76)+1.0
625     if (array(i,31).eq.77) count(8,77)=count(8,77)+1.0
626     if (array(i,31).eq.77) count(9,77)=count(9,77)+1.0
627     if (array(i,31).eq.78) count(8,78)=count(8,78)+1.0
628     if (array(i,31).eq.78) count(9,78)=count(9,78)+1.0
629 c
630 c Define output symbols
631 c
632     if (array(i,20).lt.0.5.and.array(i,20).gt.0.0) ioc=("*")
633     if (array(i,27).eq.1.0) it1=("?")
634     if (array(i,30).eq.1.0) it2=("?")
635 c
636 c Calculate dependent values
637 c
638 c
639 c Total hydrocarbon calculation
640 c
641     array(i,24)=array(i,22)+(array(i,23)*10**4)
642 c
643 c Volatile HC/OC calculated in following steps
644 c
645     if(array(i,20).eq.0.0.or.array(i,22).eq.0) go to 11
646     pchc=array(i,22)/10**2
647     array(i,26)=pchc/array(i,20)
648 c

```

```

649 c Create lithology array & count
650 c
651     do 302 j=1,9
652         if (array(i,17).ne.j) go to 302
653         do 332 ja=18,29
654             if((array(i,20).lt.0.5).and.(ja.eq.25.or.ja.eq.28.
655             & or.ja.eq.29)) go to 332
656             if(array(i,ja).ne.0.0) cnt(j,ja)=cnt(j,ja)+1.0
657             ara(j,ja)=array(i,ja)+ara(j,ja)
658     332     continue
659     302     continue
660 c
661 c Less than .5%oc routine,creating bray and secray arrays
662 c
663     11     do 42 j=18,29
664         if (array(i,j).eq.0.0) go to 500
665         if((j.eq.20).and.(array(i,20).lt.0.5))
666         &count(10,1)=count(10,1)+1.0
667         if((j.eq.20).and.(array(i,20).lt.0.5))
668         &count(10,2)=count(10,2)+1.0
669         if((j.eq.25.or.j.eq.28.or.j.eq.29).and.
670         &(array(i,20).lt.0.5)) go to 513
671         sum(2,j)=sum(2,j)+array(i,j)
672         count(11,j)=count(11,j)+1.0
673         sum(3,j)=sum(3,j)+array(i,j)
674         count(12,j)=count(12,j)+1.0
675     500     if (j.eq.21.or.j.eq.27) go to 42
676 c
677 c Adjust array subscripts
678 c
679     513     if (j.le.20) ja=j-17
680             if (j.ge.22.and.j.le.26) ja=j-18
681             if (j.eq.28.or.j.eq.29) ja=j-19
682             if((array(i,20).lt.0.5).and.(j.eq.25.or.j.eq.28.or.j.eq.29))
683             & go to 511
684             if(array(i,j).ne.0.0) go to 503
685     511     cray(iact,ja) = -9999999.
686             secray(ict,ja) = -9999999.
687             go to 42
688     503     cray(iact,ja)=array(i,j)
689             secray(ict,ja)=array(i,j)
690     42     continue
691 c
692 c Output sample information
693 c
694         if (count(3,2).ne.1.0) go to 14
695         write(6,90)
696     90     format("0"/"0",55x,"TOTAL"/" ", "SAMPLE DEGREE OF ...",5x,
697     & "LITHOLOGY",4x,"CaCO3",4x,"P205 ORGANIC VOLATILE",5x,
698     & "TOTAL",7x,"TOTAL PYROLYTIC VOLATILE TEMPERATURE"/" ",
699     & " &",5x,"SLOPE,DIP,DEPTH",7x,"&",10x,"%",8x
700     & ",%",4x,"CARBON HYDROCARBON PYROLYTIC HYDROCARBON
701     &HC/OC",6x,"HC PEAK (CELSIUS)/" ", " YEAR",3x,"ANGLE",3x,
702     & "METERS",4x, "FORMATION",20x,"TOC %",8x,"PPM",6x,"HC YIELD",

```

```

703      & 7x,"PPM",8x,"Z",8x,"/OC",7x,"I",5x,"II"/1x,55x,"& TYPE",17x,
704      & "H %")
705 14   write(6,50)
706     write (6,100) imod1,imod2
707 100  format (" ",18x,2(a13,2x))
708     write(6,110) sampno,(array(i,j),j=12,14),ilith,(array
709     &(i,ja),ja=18,19),ioc,array(i,20),(array(i,jb),jb=22,26),it1,
710     &array(i,28),array(i,29),it2,array(i,31),iform,octype
711 110  format(" ",a8,2x,3(i2,3x),all,3x,f5.1,3x,f6.2,
712     & 3x,a1,f5.2,2x,f7.0,7x,f5.3,4x,f6.0,5x,f6.2,3x,f7.3,
713     & 2x,a1,1x,f4.0,2x,f4.0,1x,a1,/1x," 19",f3.0,19x,all,
714     &3x,a10)
715     if(array(i,33).ne.0.0) write(6,130)
716 130  format(" ", "FLOAT")
717     if(eq.eq."          ") go to 44
718     write(6,120) eq
719 120  format(" ","THIS SAMPLE STRATIGRAPHICALLY EQUIVALENT TO:
720     & ",a8)
721 c
722 c Check data for possible lithologic description errors.
723 c
724 44   if ((array(i,17).eq.5).and.(array(i,18).lt.50.))
725     &write(6,70) sampno
726 70   format(1x,"limestone < 50% caco3-", a8)
727     if ((array(i,17).eq.7).and.(array(i,19).lt.5.))
728     & write (6,85) sampno
729 85   format(1x,"phosphorite < 5% P205 - ",a8)
730 c
731 c Create arrays for plots
732 c
733     if (array(i,25).eq.0.0.or.array(i,29).eq.0.0) go to 342
734     nz = nz + 1
735     z(nz) = array(i,25)
736     t(nz) = array(i,29)
737 342  if (array(i,20).eq.0.0.or.array(i,24).eq.0.0) go to 352
738     no = no + 1
739     o(no)=array(i,20)
740     h(no)=array(i,24)
741 352  if (array(i,20).eq.0.0.or.array(i,19).eq.0.0) go to 362
742     np = np + 1
743     ol(np)=array(i,20)
744     p(np)=array(i,19)
745 362  if(array(i,18).eq.0.0.or.array(i,20).eq.0.0) go to 372
746     nc = nc + 1
747     o2(nc)=array(i,20)
748     c(nc)=array(i,18)
749 c
750 c Zero necessary sample variables
751 c
752 372  ilith=(" ")      ; iform=(" ")
753     ioc=(" ")        ;  octype=(" ")
754     it1=(" ")        ;  it2=(" ")
755     imod1="          " ; imod2="          "
756 c

```



```

757 c Flag indicates locality with sample data
758 c
759     irept=0
760 c
761 c Sample loop complete, return to start of program
762 c
763     go to 1
764 c
765 c End flag
766 c
767 3     iend=1
768 *****
769 c
770 c             Averaging loop for each locality
771 c
772 *****
773 c
774 c Skip average routine if locality has too few values.
775 c
776 4     if (count(3,2).le.1.0) go to 92
777     do 62 j=18,29
778     if(count(12,j).eq.0.0) go to 22
779     ave(1,j)=sum(3,j)/count(12,j)
780 c
781 c Write averages.
782 c
783 22     if (j.ne.21.and.j.ne.27.and.count(12,j).eq.0.0) zero=1
784 62     continue
785     write(6,270)      ; write(6,50)
786     write(6,150)
787     write(6,160)(ave(1,ja),ja=18,20),(ave(1,jb),jb=22,26),
788     &(ave(1,jc),jc=28,29)
789 150    format("0",lx,"averages with <.5%oc figures left
790     & out of z average and temperature peak averages")
791 160    format("0",39x,f5.1,3x,f6.2,4x,f5.2,2x,f7.0,7x,f5.3,4x,
792     &f6.0,5x,f6.2,3x,f7.3,4x,f4.0,2x,f4.0)
793     if (zero.eq.1) write (6,24)
794 24     format(lx," Standard Deviations skipped
795     & because of <.5% OC samples or too few values")
796     if(count(10,1).ne.0.0) write(6,24)
797     if(count(10,1).ne.0.0) iskip=1
798     if (zero.eq.1.or.iskip.eq.1) go to 54
799 c
800 c Standard deviations for current locality.
801 c calculated through an IMSL tm routine
802 c
803     call bemmi(secray,ict,10,100,xm,u,s,r,ier)
804     write(6,400)
805     write(6,160) (u(1,j),j=1,10)
806 400    format("0","standard deviations:")
807 c
808 c write counts for current locality.
809 c
810 54     write(6,170) count(3,2),count(10,1)

```

```

811 170 format("0","number of samples in this locality:",f5.0,
812      &5x,"with",f4.0,2x,"samples under .5% organic carbon")
813      do 72 j=1,9
814          if(count(4,j).eq.0.0) go to 72
815          write(6,180) lith(j),count(4,j)
816 180 format(9x,all,"=",f4.0)
817 72 continue
818      do 82 j=1,6
819          if(count(6,j).eq.0.0) go to 82
820          write(6,190)iform(j),count(6,j)
821 190 format(20x,all,"=",f4.0)
822 82 continue
823      do 92 j=75,78
824          if (count(8,j).eq.0.0) go to 92
825          write(6,200) j,count(8,j)
826 200 format (1x,"19",i2," samples = ",f3.0)
827 92 continue
828 c
829 c Zero all counters.
830 c
831      count(10,1)=0.0          ; iskip=0
832      count(3,2)=0.0          ; zero=0.0
833      do 232 j=12,29
834          count(12,j)=0.0
835          ave(1,j)=0.0
836          sum(3,j)=0.0
837 232 continue
838      do 222 j=75,78
839          count(8,j)=0.0
840 222 continue
841      do 102 j=1,9
842          count(4,j)=0.0
843          count(6,j)=0.0
844 102 continue
845      do 312 j=1,100
846          do 322 ja=1,10
847              secray(j,ja)=0.0
848 322 continue
849 312 continue
850 c
851 c Go to final calculations or to read another locality record
852 c
853      if (iend.eq.1) go to 13
854      go to 5
855 c
856 *****
857 c          Calculate final averages
858 c
859 *****
860 13 do 112 j=1,8
861      if((sum(1,j).eq.0.0).or.(count(2,j).eq.0.0)) go to 112
862      ave(2,j)=sum(1,j)/count(2,j)
863 112 continue
864      do 122 j=18,29

```

```

865         if(sum(2,j).eq.0.0.or.count(11,j).eq.0.0) go to 122
866         ave(3,j)=sum(2,j)/count(11,j)
867 122    continue
868 c
869 c IMSL tm routine call
870 c
871         call bemmi(bray,isct,7,300,xmt,ut,st,rt,ier)
872
873         call bemmi(cray,iact,10,300,xma,ua,sa,ra,ier)
874 c
875 c Print counts of total localities,total samples.
876 c
877         write(6,210)count(1,1),count(3,1)
878 210    format("1/" " ",50x,"Total localities...",f4.0/51x,
879         &"Total samples...",f5.0/51x,"FINAL AVERAGES...")
880         write(6,270) ; write(6,270) ; write(6,150)
881 c
882 c Write final locality values
883 c
884         write(6,80) (ave(2,j),j=1,3),(ave(2,ja),ja=5,8)
885 80    format("0",16x,"TOTAL",7x,"NON-"/,1x,"TAI VALUES THICKNESS
886 & CARBONATES & CARBONATES MIN. & MAX. OVERBURDEN "/
887 &1x,"LO HI ",6x,"METERS",8x,"METERS",6x,"METERS",
888 &14x,"METERS"/"0",2(f4.1,2x),4x,f5.0,2x,2(7x,f5.0),7x,
889 &2(f6.0,5x))
890         write(6,400)
891         write(6,510) (ut(1,j),j=1,7)
892 510    format("0",2(f4.1,2x),4x,f5.0,2x,2(7x,f5.0),7x,2(f6.0,5x))
893         write(6,220)(count(2,j),j=1,3),(count(2,ja),ja=5,8)
894 220    format("0","with these counts:"/"0",2(1x,f4.0),5x,
895         &f4.0,9x,f4.0,3(8x,f4.0))
896 c
897 c Write final sample values
898 c
899         write (6,90)
900         write(6,160) (ave(3,ja),ja=18,20),(ave(3,jb),jb=22,26),
901         &(ave(3,jc),jc=28,29)
902         write(6,400)
903         write(6,160) (ua(1,j),j=1,10)
904 c
905 c Write total number of samples of less than .5% organic carbon.
906 c
907         write(6,600) count(10,2)
908 600    format(1x,"number of samples < .5% organic carbon =" f4.0)
909         write(6,50)
910 c
911 c Write total number of samples of each lithology.
912 c
913         do 132 j=1,9
914         if(count(5,j).eq.0) go to 132
915         write(6,180) lith(j),count(5,j)
916 132    continue
917 c
918 c Write total number of samples of each formation.

```

```

919 c
920     do 142 j=1,6
921     if(count(7,j).eq.0) go to 142
922     write(6,190) iform(j),count(7,j)
923 142 continue
924 c
925 c Write total number of samples collected in each year.
926 c
927     do 152 j=75,78
928     if(count(9,j).eq.0.0) go to 152
929     write(6,200)j,count(9,j)
930 152 continue
931 c
932 c Print averages for all lithologies and selected lithologic
933 c groups.
934 c
935     write(6,240)
936     write(6,90)
937 240 format("0",27x,"averages per lithologies:")
938     do 172 j=1,9
939     do 162 ja=18,29
940     jc=ja-17
941     if(ara(j,ja).eq.0.0.or.cnt(j,ja).eq.0.0) go to 162
942     ara(j,jc)=ara(j,ja)/cnt(j,ja)
943 162 continue
944 172 continue
945     do 182 j=1,9
946     if (count(5,j).eq.0.0) go to 182
947     litha=lith(j)
948     write(6,250) litha,(ara(j,ja),ja=1,3),(ara(j,jb),jb=5,9),
949     &(ara(j,jc),jc=11,12)
950     litha=" "
951 250 format("0",9x,a20,10x,f5.1,3x,f6.2,4x,f5.2,2x,f7.0,
952     &7x,f5.3,4x,f7.0,4x,f6.2,4x,f7.3,4x,f4.0,2x,f4.0)
953 182 continue
954     do 192 j=18,29
955     if ((cnt(5,j)+cnt(6,j)).eq.0.0) go to 192
956     ara(10,j)=((ara(5,j)+ara(6,j))/(cnt(5,j)+cnt(6,j)))
957 192 continue
958     write(6,250) lithb,(ara(10,ja),ja=18,20),
959     &(ara(10,jb),jb=22,26),(ara(10,jc),jc=28,29)
960     do 202 j=18,29
961     if ((cnt(1,j)+cnt(2,j)+cnt(4,j)).eq.0.0) go to 202
962     ara(11,j)=((ara(1,j)+ara(2,j)+ara(4,j))/(cnt(1,j)+
963     &cnt(2,j)+cnt(4,j)))
964 202 continue
965     write(6,250) lithc,(ara(11,j),j=18,20),
966     &(ara(11,ja),ja=22,26),(ara(11,jb),jb=28,29)
967 c
968 c Last line of hard copy output.
969 c
970     write(6,270)
971 *****
972 c

```

```

973 c                               Plot loop
974 c
975 *****
976 c
977 c Check for plot option
978 c
979     write(42,470)
980 470 format(lx,"Do you want plots, please answer yes or no")
981     read(41,480) iplot
982 480 format(a2)
983     if (iplot.eq."no") go to 1999
984     do 420 l1 = 1,nz
985     if (z(l1).ge.zmax) zmax=z(l1)
986     if (z(l1).le.zmin) zmin=z(l1)
987     if (t(l1).ge.tmax) tmax=t(l1)
988     if (t(l1).le.tmin) tmin=t(l1)
989 420 continue
990 c
991 c Peak II versus Z plot call
992 c
993     namex="TEMPERATURE MAXIMUM PEAK II DEGREES CELCIUS$"
994     name="TEST DATA$"
995     namey="PYROLYTIC HC/OC, or Z$"
996     call plot(tmin,zmin,tmax,zmax,nz,t,z,name,namex,namey)
997 c
998 c Hydrocarbon versus organic carbon plot setup
999 c
1000     namex="ORGANIC CARBON % $"
1001     namey="LOG OF TOTAL HYDROCARBON PPM $"
1002     do 430 l = 1,no
1003     omin=aminl(o(l),omin)
1004     omax=amaxl(o(l),omax)
1005     hmin=aminl(h(l),hmin)
1006     hmax=amaxl(h(l),hmax)
1007 430 continue
1008     call slplot(omin,hmin,omax,hmax,no,o,h,name,namex,namey)
1009 c
1010 c Phosphate versus organic carbon plot setup
1011 c
1012     do 440 l2 = 1,np
1013     olmin=aminl(ol(l2),olmin)
1014     olmax=amaxl(ol(l2),olmax)
1015     pmin=aminl(p(l2),pmin)
1016     pmax=amaxl(p(l2),pmax)
1017 440 continue
1018     namey="P205 % $"
1019     call plot(olmin,pmin,olmax,pmax,np,ol,p,name,namex,namey)
1020     do 460 l3 = 1,nc
1021     o2min=aminl(o2(l3),o2min)
1022     o2max=amaxl(o2(l3),o2max)
1023     cmin=aminl(c(l3),cmin)
1024     cmax=amaxl(c(l3),cmax)
1025 460 continue
1026 c

```

```

1027 c Calcium carbonate versus organic carbon plot call
1028 c
1029     namey="CaCO3 % $"
1030     call plot(o2min,cmin,o2max,cmax,nc,o2,c,name,namex,namey)
1031 c
1032 c End of plot loop, skip Point locality loop
1033 c and go to end
1034 c
1035     go to 1999
1036 *****
1037 c
1038 c             Point Locality Data Type Routine
1039 c
1040 *****
1041 710  open(1,file="_____.data",form="formatted")
1042     open(6,file="_____.out",carriage=.true.,
1043         &form="formatted",defer=.true.)
1044 711  i = i+1
1045     read(1,750,end=721) loc,lat1,lat2,lat3,long1,long2,
1046         &long3,iapprox,instate,(iage(j),j=1,3),cail,cai2,fm,
1047         &quad,county
1048 750  format(a10,2i2,a2,i3,i2,a2,i1,i2,3i1,f3.1,a3,2a20,a15)
1049 c
1050 c Sum and count and define output variables
1051 c
1052     do 720 j=1,51
1053         if(instate.eq.j) count(1,j) = count(1,j)+1
1054         if(instate.eq.j) istate = state(j)
1055 720  continue
1056         if(cai2.ne." ") decode(cai2,712) cai2out
1057 712  format(f3.1)
1058         if(cail.ne.0.) count(2,1) = count(2,1)+1
1059         if(cai2.ne." ") count(2,2) = count(2,2)+1
1060         sum(1,1) = sum(1,1)+cail
1061         sum(1,2) = sum(1,2)+cai2out
1062         do 730 j = 1,9
1063             if(iage(3).eq.j) count(3,j)=count(3,j)+1
1064             if(iage(3).eq.j) ageout = age(j)
1065 730  continue
1066 c
1067 c Age modifiers loop
1068 c
1069     do 722 j=1,3
1070         if(iage(1).eq.j) agemod1 = amod(j)
1071         if(iage(2).eq.j) agemod2 = amod(j)
1072 722  continue
1073 c
1074 c Output
1075 c
1076     if(i.ne.1) go to 723
1077     write(6,751)          ; write(6,752)          ; write(6,753)
1078     write(6,754)          ; write(6,760)
1079 751  format(5("0"/))
1080 752  format("0",40x,"DATA FROM _____")

```

```

1081 753 format("0",20x,"EXAMPLE OF THE SOURCE_RX PROGRAM FOR
1082      & RETRIEVAL OF POINT LOCALITY DATA")
1083 c
1084 c New page
1085 c
1086 754 format("1")
1087 723 ipage = ipage + 1
1088      write(6,756) ; write(6,757) ; write(6,758) ; write(6,759)
1089      write(6,770) loc,istate,county,quad,lat1,lat2,lat3,
1090      &long1,long2,long3
1091      if(i.eq.1) write (6,771)
1092      write(6,760) ; write(6,761) ; write(6,762) ; write(6,763)
1093 771 format(" ",76x,"approximately")
1094 756 format(" "," SAMPLE")
1095 757 format(" "," NAME")
1096 758 format("+",18x,"STATE")
1097 759 format("+",34x,"COUNTY";12x,"QUADRANGLE",9x,"LATITUDE
1098      & LONGITUDE")
1099 760 format("0")
1100 761 format("0",49x,"AGE")
1101 762 format("+",72x,"FORMATION")
1102 763 format("+",93x,"CAI RANGE")
1103 770 format("0",a10,4x,a14,1x,a12,4x,a20,2x,2(i2,2x),a2,
1104      &3x,i3,2x,i2,2x,a2)
1105      write(6,764) agemod1,agemod2,ageout,fm
1106 764 format("0",35x,2(a6,2x),a13,4x,a20)
1107      if(cai1.eq.0) write(6,755)
1108 755 format("+",92x,"not determined")
1109      if(cai1.ne.0) write(6,774) cai1
1110 774 format("+",84x,f3.1)
1111      if(cai2.ne." ") write(6,765) cai2out
1112 765 format("+",97x," - ",f3.1)
1113      cai2out = 0.
1114      agemod1 = "      "
1115      agemod2 = "      "
1116      ageout = "      "
1117      write(6,270) ; write(6,760)
1118      if (ipage.eq.4) write(6,754)
1119      if (ipage.eq.4) ipage = 0
1120      go to 711
1121 c
1122 c File read, now average and end.
1123 c
1124 721 ave(1,1) = sum(1,1)/count(2,1)
1125      ave(1,2) = sum(1,2)/count(2,2)
1126 c
1127 c Output averages and counts
1128 c
1129      write(6,754) ; write(6,751) ; write(6,766)
1130 766 format("0",57x,"COUNTS"/"0")
1131      write(6,758) ; write(6,760)
1132      do 725 j=1,51
1133      if (count(1,j).ne.0.) write(6,767) state(j),count(1,j)
1134 767 format(" ",14x,a14," = ",f4.0)

```

```

1135 725 continue
1136 write(6,760) ; write(6,761) ; write(6,760)
1137 do 726 j=1,9
1138 if(count(3,j).ne.0.) write(6,768) age(j),count(3,j)
1139 768 format(44x,a14," = ",f4.0)
1140 726 continue
1141 write(6,773) count(2,1),count(2,2)
1142 773 format ("0",80x,"CAI COUNTS"/"0",80x,"LOW HIGH"/
1143 &"0",80x,f3.1,2x,f3.1)
1144 c
1145 c Output cai averages
1146 c
1147 write(6,760) ; write(6,769)
1148 write(6,772) (ave(1,j),j=1,2)
1149 769 format("0",75x,"AVERAGE CAI RANGE")
1150 772 format("0",80x,2(f3.1,2x))
1151 write(6,751) ; write(6,270)
1152 c
1153 c End of Point Locality type loop
1154 c
1155 *****
1156 c
1157 c Close all files
1158 c
1159 *****
1160 1999 close(6)
1161 close(1)
1162 close(41)
1163 close(42)
1164 stop
1165 end
1166 *****
1167 c
1168 c Plot routine using DISSPLA tm
1169 c
1170 *****
1171 subroutine plot(xmin,ymin,xmax,ymax,ncnt,x,y,label,
1172 &labelx,labely)
1173 dimension x(300),y(300)
1174 call tk120
1175 call marker (4)
1176 call axsplt(xmin,xmax,6.,xor,xstep,axis)
1177 call axsplt(ymin,ymax,9.,yor,ystep,axis)
1178 call title(label,100,labelx,100,labely,100,6.,9.)
1179 call graph (xor,xstep,yor,ystep)
1180 call grid (1,1)
1181 call curve (x,y,ncnt,-1)
1182 call iowait(100)
1183 call endpl (-1)
1184 return
1185 end
1186 *****
1187 c
1188 c Semilog plot routine using DISSPLA tm

```



```
1189 c
1190 *****
1191     subroutine splot (xmin,ymin,xmax,ymax,ncnt,x,y,label,
1192     &labelx,labely)
1193     dimension x(300),y(300)
1194     call marker (4)
1195     call axsplt(xmin,xmax,6.,xor,xstep,axis)
1196     call algplt(ymin,ymax,9.,yor,cycle)
1197     call title (label,100,labelx,100,labely,100,6.,9.)
1198     call ylog(xor,xstep,yor,cycle)
1199     call grid (1,1)
1200     call curve(x,y,ncnt,-1)
1201     call iowait (100)
1202     call endpl (-2)
1203     return
1204     end
```