UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

GEOLAB: A COMPUTER PROGRAM FOR ANALYSIS OF LOW-FREQUENCY EARTHQUAKE PRECURSOR DATA

by

J. Herriot

Open-file Report
78-700

U. S. Geological Survey
OPEN FILE REPORT

Geolab is an interactive "geophysics laboratory" giving the user an arena in which to shape and test out ideas quickly and conveniently. It is a sort of software toolshop where the user is at the control panel and has immediate access to a large repetoir of existing tools or can add new ones instantly.

To access Geolab the first time (or to get the latest version):

```
ec >udd>Geolab>JHerriot>geolab>new_version
```

This adds the appropriate search rules and creates a segment called gl_state in the user's directory. Subsequently access can be facilitated by adding the following line to your start_up.ec:

```
ec >udd>Geolab>JHerriot>geolab>start_up.ec
```

Once in Geolab you will get a " ?" prompt. Type your command(s) or "operators" (80 char limit per line) and upon hitting carrage-return Geolab will execute that line left to right. When those commands are finished executing you will get another " ?" prompt, and so on. The exception to the rule is when parenthesis do not balance: in the case of missing right parenthesis, ")", a "1?" prompt will appear requesting that the command sequence be completed before execution can take place.

To exit from Geolab say "stop" or "s". Within this process "gl" will re-envoke Geolab. (if your start_up.ec contains the above ec then "gl" will always invoke Geolab.) Note that "gl" will cause "Geolab? -- <date>" to be announced prior to the first prompt; to suppress this use the "quiet" operator when in Geolab.

Geolab grew out of the prejudice that simple things should be simple to do. For example, if I have several numbers and want to square each of them, list their values, and plot them, this should be easily accomplished -- without need of a text editor and a compiler. The following is an example of the commands necessary to do the above assuming a set of ten integers exists in a segment called "mydata".

```
?iarr ten 10              % declare integer array (vector) of size 10
?file 'mydata'            % designate segment 'mydata' (path name)
?read ten '(10i6)'        % read into ten using (10i6) format
?ten is   ten**2 is       % display ten before and after computation
?plot ten                 % plot ten's new values on tektronix
?stop                     % return to Multics command level
```

Simple operations should be simple to carry out -- and more complex operations should be more complex, but still possible. To this end several hundred operators have been developed for general minipulation of numerical and character data. There is not room here to explicate them; however, by typing the operator "teach" the user may receive an interactive tutorial on some of the most useful operator-tools. We must apologetically add that although there are a couple of manuals written for Geolab1/LBL, Geolab2/Multics is too young to be so documented.

The following may give a flavor of what is available:

```
?var duck                 % declare a variable
?3*(2+100)=duck           % (left to right) assignment into duck
?rarr fred 5              % declare a real array (1 dimentional)
?fred count               % fill fred with the counting nos (1,2,3...)
?sin(fred/(pi+1))         % arithmetic on entire array
?5 do(fred[i] is)         % "do" loop causing display of "fred sub i"
?str message 60           % declare string of max size 60 characters
?'print my'=message       % string assignment -- left to right
?message cat 'data'       % concatenation onto message
?size message  is         % display the current size of message
?exec message             % send 'print mydata' to Multics as command
?e print mydata           % send rest of line to Multics (cf qedx, ted)
?op double(^ *2)          % make a new operator called "double"

?double (pi**2) is        % invoke op filling in "^" with parameter
?xop double               % "examine the definition of op" double
?change double * +        % change def of double to "^ + 2"
?rollin 'myops'           % read in op defs from text edited segment
```

Current developments of Geolab are in the direction of data base tools. Although general tools are not yet available, some more specialized operators for handling low frequency time series data are now working. The user may set up a directory to contain such data where each segment in that directory will contain data corresponding to one instrument. Assuming such a directory exists containing a null segment called "nutn", here are some examples of "bottle" operators:

```
?dir 'tiltdata'           % disignate directory
?'nutn'=sens              % disignate segment/intrument
?78=yr 365=jl 60=invl     % choose becin time and delta t in minutes
?ibottle                  % initialize segment to "missing data"
?ten =dat                 % write out array ten into 'nutn'
?ten  dat                 % read in -- using begin time
```

More detailed information is available by executing the following Multics cmd:

```
print >udd>Geolab>JHerriot>doc>dat
```

Also as documentation is written it will be located in the directory:

```
>udd>Geolab>JHerriot>doc
```

This document itself is entered as segment "intro" in the above directory. And if there are any questions or comments address them to my Multics mailbox:

```
JHerriot Geolab
```

```
cfrom vrea -- vrea sets y=y2 ie type=real -- branch acc. to asgn flag
  282 if(a.ne.f)goto 2182
ctake number from mem and push onto stack
 1182 z=z+1 ; s(z)=mm(n)-adr+1       ; ty(z)=y ; goto 190
cfrom typ,nam,dyn -- r/w into from/to stk to/from stk field
  182 n=n+z ; y=y1 ; if(a.eq.f)goto 1182
cwhen writing onto stk field (typ,nam,dyn) check for legit values
      n=n-1 ; if(s(z).lt.i.or.s(z).gt.j)goto 905
cput number into mem and turn off asgn flag
 2182 a=f   ; mm(n)=s(z)+adr-1                    ; goto 190
cset type to into=1, real=2, strg=3
  184 ty(z)=y1                                    ; goto 190
  284 ty(z)=y2                                    ; goto 190
  384 ty(z)=y3                                    ; goto 190
cpush real or intg onto scr stack
  191 z=z+1 ; s(z)=sz ; ty(z)=y1                  ; goto 190
  291 z=z+1 ; s(z)=sz ; ty(z)=y2                  ; goto 190
cpop one and replace new top of stack with real or intg (diadic arith)
  194 z=z-1 ; s(z)=sz ; ty(z)=y1                  ; goto 190
  294 z=z-1 ; s(z)=sz ; ty(z)=y2                  ; goto 190
cpop two or one elements off scratch stack
  198 z=z-1
  199 z=z-1
check for underflow or overflow of scratch stack
  190 if(z.lt.zz.or.z.ge.zr)goto 1902
c--cntl-stk-maint-------------------------------------
ccheck for auto-rtn (up arrow)
  200 if(pn(c).ne.xup)goto 300
cpop cntl-stk
  210 c=c-1 ; if(c.lt.1)goto 901 ; if(pn(c).eq.xup)goto 210
ccompute program counter
      d=c ; if(pa(c).lt.0)d=-pa(c)
c--get-token------------------------------------------
  300 p=pa(d)+1   ; pa(d)=p  ; ptr=mm(p)
      if(p.le.pz(d))goto 310 ; if(d.gt.1.and.pn(c).ne.xup)goto 210
      if(d.eq.1    )goto 900 ; goto 907
  310 if(d.le.dlv  )goto 920 ; ncyc=ncyc+1
cresolve ptr in nam-tab, if neg hard-op else soft-op
  400 if(ptr.lt.1.or.ptr.gt.zn)goto 903; adr=aa(ptr); if(adr)600,903,500
c--soft-op--------------------------------------------
cpush new soft-op on to cntl-stk
  500 c=c+1   ; if(c.gt.zc)goto 901 ; pa(c)=adr ; pz(c)=az(ptr)
      ca(c)=d ; d=c                 ; pn(c)=ptr ; goto 300
c--hard-op--------------------------------------------------
  600 ce=-adr
c--go-to-particular-ce--or--pre-ce calc---------------------
  620 if(ce.gt.100)goto 621
      goto(  1,  2,700,701,701,    6,  7,701,701,700,
           700,701, 13, 14, 15,   16,724,724,724,724,
            21, 22, 23,739, 25,   26, 27,701,739,723,
           721, 32,700,701,700,   36, 37, 38, 39, 40,
           701,701,739,722, 45,  739,701, 48, 49, 50,
           732,732,732,732,732,  732,732,732,732,737,
           737,737,737,737,737,  737,737,737,737,731,
           737,731,737,731,739,  732,732,732,732,732,
           732, 82,732,732,731,   86,739, 88,739, 90,
            91, 92, 93, 94, 95,  721, 97, 98, 99,100)ce
  621 goto(700,700,700,700,700,  700,701,739,739,739,
           111,112,113,114,739,  116,117,118,119,120,
           121,700,123,700,700,  700,127,128,129,130,
```

```
119               131,132,133,721,135,  136,137,730,139,140,
120               141,142,143,144,145,  146,147,148,149,150)ce-100
121     c--pre-ce--automatic-type-conversion-----------------------
122     cforce number to integer
123       739 y=ty(z); s1=s(z); if(y.eq.y1)goto 790; if(y.eq.y2)goto 1739; goto 90
124      1739 s1=r1  ; goto 790
125     cforce number to real
126       737 y=ty(z); s1=s(z); if(y.eq.y2)goto 790; if(y.eq.y1)goto 1737; goto 90
127      1737 r1=s1  ; goto 790
128     cdyadic arithmetic -- ii=i  ir=r  ri=r  rr=r
129       732 i=ty(z) ; j=ty(z-1) ; s1=s(z) ; s2=s(z-1) ; k=i+j-1+umod*3
130           if(i.lt.y3.and.j.lt.y3)goto 734
131     c     -check for arithmetic on scalar letters
132           if(i.eq.y3)i=y1 ; if(j.eq.y3)j=y1 ; k=i+j-1+umod*3
133           if(i.gt.y2.or. j.gt.y2)goto 905
134     c           -u sys off- -miss=unew- -1miss=num-
135     c           int mix rea int mix rea int mix rea
136       734 goto(790,733,730,742,733,743,742,733,743)k
137       733 if(ty(z).eq.y1)r1=s1 : if(ty(z-1).eq.y1)r2=s2 ; goto(730,743,743)umo
138     cdyadic integer arith with missing data
139       742 ksym=usym
140           uyes=uyes+1 ; if(s1.ne.ksym.and.s2.ne.ksym)goto 790 ; sz=unew
141           uno2=uno2+1 ; uyes=uyes-1 ; if(s1.eq.ksym.and.s2.eq.ksym)goto 194
142           uno1=uno1+1 ; uno2=uno2-1 ; if(umod.eq.1)goto 194
143           sz=s1       ; if(s1.eq.ksym)sz=s2 ; goto 194
144     cdyadic real arith with missing data
145       743 uyes=uyes+1 ; if(r1.ne.usym.and.r2.ne.usym)goto 790 ; rz=unew
146           uno2=uno2+1 ; uyes=uyes-1 ; if(r1.ne.usym.and.r2.ne.usym)goto 294
147           uno1=uno1+1 ; uno2=uno2-1 ; if(umod.eq.1)goto 294
148           sz=s1       ; if(r1.eq.usym)sz=s2 ; goto 294
149     cmonadic arithmetic
150       731 y=ty(z) ; s1=s(z) ; k=y+umod*2 ; if(y.lt.y3)goto 735
151     c     -check for arithmetic on scalar letters
152           if(y.eq.y3)y=y1 ; k=y+umod*2 ; if(y.gt.y2)goto 905
153     c           -umod=0 -umod=1 -umod=2
154     c           int rea int rea int rea
155       735 goto(790,730,741,740,741,740)k
156     cmonadic integer arithmetic with missing data
157       741 uyes=uyes+1 ; if(s1.ne.usym)goto 790 ; uyes=uyes-1
158           uno1=uno1+1 ; sz=unew     ; goto 194
159     cmonadic real arithmetic with missing data
160       740 uyes=uyes+1 ; if(r1.ne.usym)goto 730 ; uyes=uyes-1
161           uno1=uno1+1 ; rz=unew     ; goto 294
162       730 k=ce-50
163           goto(851,852,853,854,855,  856,857,858,859, 60,
164                 61, 62, 63, 64, 65,   66, 67, 68, 69,870,
165                 71,872, 73,874, 75,   76, 77,878,879,880,
166                881, 82, 83, 84,885)k
167     c--pre-ce--descriptor-resolution----------------------------
168     casign-724  nx.to.top-723  two-722  one-721
169       724 if(a    .ne.f )goto 723
170       721 y=ty(z)   ; n=dn(z)   ; if(y.ge.y4)goto 790 ; goto 905
171       722 if(ty(z).lt.y4)goto 905
172       723 y=ty(z-1) ; n=dn(z-1) ; if(y.ge.y4)goto 790 ; goto 905
173     c--pre-ce--relative-level-calculation-----------------------
174       711 l=d
175       712 if(u.le.0) goto 713 ; l=ca(l) ; u=u-1 ; goto 712
176       713 u=zu     ; goto 790
177       701 l=ca(d)  ; goto 709
178       700 l=d
```

```
179       700 if(u.ne.zu)goto 711
180     c--go-to-particular-ce----------------------------------------
181       700 if(ce.gt.100)goto 701
182           goto(  1,  2,  3,  4,  5,    6,  7,  8,  9, 10,
183                 11, 12, 13, 14, 15,   16, 17, 18, 19, 20,
184                 21, 22, 23, 24, 25,   26, 27, 28, 29, 30,
185                 31, 32, 33, 34, 35,   36, 37, 38, 39, 40,
186                 41, 42, 43, 44, 45,   46, 47, 48, 49, 50,
187                 51, 52, 53, 54, 55,   56, 57, 58, 59, 60,
188                 61, 62, 63, 64, 65,   66, 67, 68, 69, 70,
189                 71, 72, 73, 74, 75,   76, 77, 78, 79, 80,
190                 81, 82, 83, 84, 85,   86, 87, 88, 89, 90,
191                 91, 92, 93, 94, 95,   96, 97, 98, 99,100)ce
192       701 goto(101,102,103,104,105,  106,107,108,109,110,
193                111,112,113,114,115,  116,117,118,119,120,
194                121,122,123,124,125,  126,127,128,129,130,
195                131,132,133,134,135,  136,137,138,139,140,
196                141,142,143,144,145,  146,147,148,149,150)ce-100
197     c--code-escapes-----------------------------------------
198     cinit   -- initialize the system -- called upon entering gl
199         1 print, "geolab2 -- 25 apr 78"
200     crset   -- reset the system -- called by init and errors
201         2 c=1 ; d=1 ; z=zz ; a=f ; u=zu ; lur=5 ; p=1 ; ty(z)=y1
202           umod=udef ; mm(1)=xty  ; goto 900
203     c(      -- eval until explicit return -- rl=0  (skip over len-field)
204         3 pa(d)=pa(d)+1
205     ceval   -- eval until explicit return -- rl=1
206         4 continue
207     c'      -- eval one op (auto-return)  -- rl=1
208         5 c=c+1 ; if(c.gt.zc)goto 901 ; pa(c)=-l
209           d=l                         ; pn(c)=ptr ; goto 300
210     c) ]    -- return one level
211         6 goto 210
212         7 goto 210
213     cquox   -- quo (rl=1) with cntl of decr formation (xop-- true f f quox)
214         8 i=s(z-2) ; j=s(z-1) ; k=s(z) ; z=z-3 ; if(z.lt.zz)goto 902 ; goto 80
215     cquo    -- quote -- push descr of code vector onto stk -- rl=1
216     c     -note that qu will not continue to another tty line like '
217         9 i=t ; j=t ; k=t
218     c     -up arrow if multi-word op except lparen -- t.. quox
219       80 ptr=xup ; p=pa(l) ; n=p+1 ; g=p+2
220           if(mm(g).lt.0.and.mm(n).ne.xlp.and.n.lt.pz(l).and.i.ne.f)goto 5
221     c     -up arrow if exclam point except rl=0  -- .t. quox
222           p=n ; pa(l)=p ; n=mm(p)
223           if(n.eq.xcl.and.d.ne.l.and.j.ne.f)goto 5
224     c     -put descriptor on stack
225           z=z+1 ; if(n.lt.1.or.n.gt.zn)goto 903
226           ty(z)=y4 ; dn(z)=n ; da(z)=aa(n) ; dz(z)=az(n) ; dy(z)=0
227     c     -if lparen put descr of whole group on stack -- ..t quox
228           if(n.eq.xlp.and.k.ne.f)goto 1809  ; k=0
229     c     -increment the pc and exit
230           if(mm(g).lt.0)k=-mm(g) ; if(n.eq.xlp)k=1 ; pa(l)=p+k ; goto 100
231     c     -put descr of whole (...) group on stack, incr pc, and exit
232     c     -to make cnam work (.) 3 -3 137 6 will be -3 137 since pc gets +1ed
233      1809 dn(z)=pn(l) ; da(z)=g ; pa(l)=p-mm(g) ; dz(z)=pa(l)-1 ; goto 100
234     castr   -- string array -- same as single quote
235        10 continue
236     c'      -- single quote -- put string descr on stack
237        11 z=z+1 ; p=pa(l) ; ty(z)=y6 ; dn(z)=pn(l) ; k=(p+3)*4 ; da(z)=k
238           j=mm(p+1)       ; n=mm(p+2) ; i=(-j-2)*4 ; n=min(n,i)
```

```
239           if(j.gt.0)goto 908
240           dz(z)=k+n-1      ; pa(l)=pa(l)-j           ; dv(z)=0 ; goto 100
241     cskip   -- skip 1 or (...) of toks -- rl=1, k=num skipped, = means copy
242        12 p=pa(l)+1 ; k=0 ; if(mm(p+1).lt.0)k=-mm(p+1)
243           p=p+k ; pa(l)=p ; if(p.gt.pz(l))goto 210 ; hh(d)=k
244           if(a.eq.f)goto 200 ; a=f ; k=k-2
245           do 812 i=1,k ; j=m+i ; ij=p-k-1+i
246              mm(j)=mm(ij)
247         812   continue
248           goto 200
249     ctyp0   -- jump  s=1  v=?  d=3 -- next to top of stack
250        13 n=pa(d) ; pa(d)=n+3 ; j=z-1 ; goto 814
251     ctyp1   -- jump  s=0  v=1  d=?  -- top of stack
252        14 n=pa(d) ; pa(d)=n+3 ; j=z
253       814 y=ty(j) ; if(y.ge.y4)n=n+1 ; if(y.ge.y4.and.dy(j).ne.0)n=n+1
254           ptr=mm(n+1) ; goto 400
255     ctyp?   -- jump ss=0 sv=1 sd=2 vs=3 vv=4 vd=5 ds=6 dv=7 dd=8
256        15 n=pa(d) ; pa(d)=n+9 ; j=z ; k=z-1
257           y=ty(k) ; if(y.ge.y4)n=n+3 ; if(y.ge.y4.and.dy(k).ne.0)n=n+3
258           goto 814
259     ctyp-16  nam-17  beg-18  fin-19  dyn-20  --  fields of scr stack
260     ctyp    -- rw ty(z) of any descriptor
261        16 adr=1      ; n=000800 ; i=1    ; j=zy ; goto 183
262     cnam    -- rw dn(z) of vect descriptors
263        17 adr=1      ; n=000840 ; i=1    ; j=zn ; goto 183
264     cbeg    -- rw da(z) of vect descriptors
265        18 k=000880+z ; goto 819
266     cfin    -- rw dz(z) of vect descriptors
267        19 k=000920+z
268     c     -combined beg/fin -- i=density  g=absbeg
269       819 i=yy(y) ; g=aa(n)*i ; if(a.ne.f)goto 1019
270           sz=mm(k)-g+1  ; goto 191
271     c     -write in beg/fin -- first force to integer
272      1019 y=ty(z) ; s1=s(z) ; goto(3019,2019,905,905,905,905)y
273      2019 s1=r1
274     c     -now do the assignment into da(z) of dz(z)
275      3019 j=s1+g-1  ; if(j.lt.g)j=g ; n1=(az(n)+1)*i-1 ; if(j.ge.n1)j=n1
276           mm(k-1)=j ; a=f ; goto 200
277     cdyn    -- rw dy(z) of vect descriptors -- 0=stat  1,zn=dynamic
278        20 adr=1      ; n=000960 ; i=0    ; j=zn ; goto 183
279     c=      -- set asgn flag to true
280        21 a=t ; goto 200
281     caf     -- move asgn-flag to f-reg
282        22 ff(d)=a ; a=f ; goto 200
283     cfa     -- move f-reg to asgn-flag
284        23 a=ff(d)        ; goto 200
285     c $     -- put stk top into u ($)
286        24 u=s1 ; goto 199
287     cof     -- put 1        into u ($) -- shorthand for 1$
288        25 u=1     ; goto 200
289     cug     -- move u ($) to g register
290        26 gg(d)=u ; u=zu ; goto 200
291     cgu     -- move g register to u ($)
292        27 k=gg(d) ; if(u.ne.zu)goto 827 ; u=k    ; goto 200
293       827             if(k.eq.zu)k=0       ; u=k+u ; goto 200
294     ccalr   -- nt ptr of caller or =calr returns just abs level
295        28 sz=l ; if(a.eq.f)sz=pn(l) ; a=f ; goto 191
296     cifju   -- if  s(z-1)=true  then  jump s(z) toks ahead
297        29 if(s(z-1).ne.f)pa(d)=pa(d)+s1 ; goto 198
298     cdecl   -- declare(nam len decl), redecl(nam len =decl), erase(0=len)
```

```
299       30 len=s(z)-1        ; if(aa(n).eq.0)goto 830 ; if(a.eq.f)goto 904
300      830 aa(n)=m           ; if(len.eq. -1)aa(n)=0  ; a=f
301          mm(m)=n           ; m=m+len ; az(n)=m-1    ; if(m.lt.zm)goto 199
302          call garcol(t,e) ; if(e.ne.f)goto 908     ; goto 199
303     cslip   -- slip in an op between me and my caller (descr fr scr-stk)
304       31 c=c+1    ; pn(c)=pn(d) ; pa(c)=pa(d) ; pz(c)=pz(d)
305          ca(c)=d ; pn(d)=dn(z) ; pa(d)=da(z) ; pz(d)=dz(z)
306          d=c     ; if(pa(d).gt.0)goto 199 ; pa(d)=1 ; pz(d)=0 ; goto 199
307     ccnam   -- call by name (obtained from scr-stk) -- (sort of indirect)
308       32 k=z        ; z=z-1 ; if(z.lt.zz)goto 902      ; n=000840+k
309          if(ty(k).eq.y4)goto 1032
310          if(ty(k).ne.y1)goto 905  ; ptr=s(k) ; goto 400
311     1032 ptr=dn(k) ; adr=da(k)             ; if(adr)600,903,832
312      832 c=c+1     ; if(c.gt.zc)goto 901 ; pa(c)=adr ; pz(c)=dz(k)
313          ca(c)=d   ; d=c                 ; pn(c)=ptr ; goto 300
314     c    the following is a simpler cnam which cannot do part of an op
315     c   32 ptr=dn(z) ; z=z-1 ; if(z.lt.zz)goto 902 ; goto 400
316     cdrop   -- drop down levels -- rl=0
317       33 c=l ; d=l ; goto 200
318     ceot    -- end of text
319       34 sz=f ; if(pa(l).ge.pz(l))sz=t
320          if(a.eq.f)goto 191; pa(l)=pz(l); a=f; goto 200
321     cchpc   -- change pc -- rl=0  -- primitive goto within an op
322       35 y=ty(z) ; if(y.ne.y1)goto 905 ; pa(l)=pa(l)+s(z)
323          n=pn(l) ; if(pa(l).lt.aa(n))pa(l)=aa(n) ; goto 199
324     cup2    -- up arrow two levels down -- specially for vect arith
325       36 ce=5 ; ptr=xup ; l=ca(d) ; l=ca(l) ; goto 5
326     cdynx dyn0 -- dynamic exec of descr ty(z) or ty(z-1)
327       37 ptr=dy(z  ) ; if(ptr.ne.0)goto 400 ; goto 908
328       38 ptr=dy(z-1) ; if(ptr.ne.0)goto 400 ; goto 908
329     cmakd   -- make a code descr from any vector descr
330       39 n=dn(z)  ; if(ty(z).ge.v4)goto 839
331          n= s(z)  ; if(n.lt.1.or.n.gt.zn)goto 908        ; dn(z)=n
332      839 ty(z)=y4 ; da(z)=aa(n) ; dz(z)=az(n) ; dy(z)=0 ; goto 200
333     cact3   -- special op of active arrays -- op active(act3 [ : xvvm)
334     cif [or: then ^ else (if descr8= then xvvm else nop )
335       40 n=pa(d) ; pa(d)=n+3 ; ptr=xup ; ce=5
336          l=ca(d) ; l=ca(l)   ; p=pa(l) ; k=mm(p+1)
337          if(k.eq.mm(n+1).or.k.eq.mm(n+2))goto 5
338          if(a.eq.f.or.ty(z-1).lt.y4)goto 200; ptr=mm(n+3); a=f; goto 400
339     cdolu   -- special ce for fast implem of do loops -- rl=1
340       41 ii(l)=ii(l)+kk(d) ; n=pn(l) ; pa(l)=jj(d)+aa(n)-1
341          if(kk(d).lt.0)goto 1041     ; if(ii(l).gt.ii(d))goto 12
342      841 pa(d)=p-1 ; ptr=xup         ; goto 5
343     1041 if(ii(l).lt.ii(d))goto 12   ; goto 841
344     cwhiz   -- op while(0=of i, pc=i, skip skip pc=j, whiz) -- while()()
345       42 n=pn(l) ; j=jj(d)+aa(n)-1
346          ptr=xup ; pa(d)=p-1 ; if(pa(l).ne.j)goto 842 ; pa(l)=ii(d)+aa(n)-1
347          i=ii(l)+1 ; ii(l)=i     ; goto 5
348      842 z=z-1 ;if(z.lt.zz)goto 902 ; if(s(z+1).ne.f)goto 5 ; pa(l)=j ; goto
349     couts   -- descr of all mem -- 1,4=intg  2,5=real  3,6=strg
350       43 if(s1.lt.4)s1=s1+3 ; if(s1.lt.4.or.s1.gt.6)s1=4
351          ty(z)=s1 ; dn(z)=xqm ; da(z)=yy(s1) ; dz(z)=zm+(s1/6)*zm*3
352          dy(z)=0  ; goto 200
353     cvvm    -- vector vector move -- f a s t
354       44 call moreops(1,e) ; if(e.eq.1)goto 908 ; goto 199
355     cvsm    -- vector scalar move -- f a s t
356       45 if(ty(z).ge.y4.or.ty(z-1).lt.y4)goto 905
357          call morecps(2,e) ; if(e.eq.1)goto 908 ; goto 199
358     cmops   -- call moreops
```

```
359        46 z=z-1 ; call moreops(s1,e) ; if(e.eq.1)goto 908 ; goto 190
360     cptr   -- picks up nt ptr of next token (of ouo) -- rl=0
361        47 p=pa(l)+1 ; sz=mm(p) ; if(mm(p+1).lt.0)p=p-mm(p+1)
362           pa(l)=p   ; goto 191
363     cx048 x049 x050
364        48 goto 200
365        49 goto 200
366        50 goto 200
367     c--arithemec-----------------------------------
368     cplus
369        51 sz  =s2   +  s1  ; goto 194
370       851 rz  =r2   +  r1  ; goto 294
371     cminu
372        52 sz  =s2   -  s1  ; goto 194
373       852 rz  =r2   -  r1  ; goto 294
374     ctims
375        53 sz  =s2   *  s1  ; goto 194
376       853 rz  =r2   *  r1  ; goto 294
377     cdiv
378        54 sz  =s2   /  s1  ; goto 194
379       854 rz  =r2   /  r1  ; goto 294
380     cidiv
381        55 sz  =s2   /  s1  ; goto 194
382       855 k  =r2   /  r1  ; rz=k ; goto 294
383     cpow
384        56 sz  =s2  **  s1  ; goto 194
385       856 rz  =r2  **  r1  ; goto 294
386     cmod
387        57 sz = mod(s2,s1)  ; goto 194
388       857 rz =amod(r2,r1)  ; goto 294
389     cmin
390        58 sz=s1 ; if(s2.lt.s1)sz=s2 ; goto 194
391       858 rz =amin1(r2,r1) ; goto 294
392     cmax
393        59 sz=s1 ; if(s2.gt.s1)sz=s2 ; goto 194
394       859 rz =amax1(r2,r1) ; goto 294
395     crnd
396        60 k=r1      + 0.5      ; r(z)=k ; goto 284
397     csi co ta asi aco ata
398        61 r(z)= sin(r1)        ; goto 284
399        62 r(z)= cos(r1)        ; goto 284
400        63 r(z)= tan_(r1)        ; goto 284
401        64 r(z)=asin(r1)        ; goto 284
402        65 r(z)=acos(r1)        ; goto 284
403        66 r(z)=atan(r1)        ; goto 284
404     clogx l10x -- logarithms
405        67 if(r1.le.0.0)r1=1. ; r(z)=alog(r1)        ; goto 284
406        68 if(r1.le.0.0)r1=1. ; r(z)=alog10(r1)      ; goto 284
407        69 r(z)= exp(r1)      ; goto 284
408     cabs    -- absolute value
409        70 s(z)=iabs(s1)        ; goto 200
410       870 r(z)= abs(r1)        ; goto 200
411     csqrt   -- square root
412        71 if(r1.lt.0.0)r1=0.0 ; r(z)=sqrt(r1)        ; goto 284
413     cchs    -- change sign
414        72 s(z)=-s1             ; goto 200
415       872 r(z)=-r1             ; goto 200
416     cflot
417        73 r(z)=r1              ; goto 294
418     cfix
```

```
419        74 goto 200
420       874 s(z)=r1 ; ty(z)=y1    ; goto 200
421     cnotx
422        75 s(z)=f ; if(s1.eq.f)s(z)=t ; goto 194
423     ceq
424        76 sz=f ; if(s2.eq.s1)sz=t ; goto 194
425     cne
426        77 sz=f ; if(s2.ne.s1)sz=t ; goto 194
427     clt
428        78 sz=f ; if(s2.lt.s1)sz=t ; goto 194
429       878 sz=f ; if(r2.lt.r1)sz=t ; goto 194
430     cle
431        79 sz=f ; if(s2.le.s1)sz=t ; goto 194
432       879 sz=f ; if(r2.le.r1)sz=t ; goto 194
433     cgt
434        80 sz=f ; if(s2.gt.s1)sz=t ; goto 194
435       880 sz=f ; if(r2.gt.r1)sz=t ; goto 194
436     cge
437        81 sz=f ; if(s2.ge.s1)sz=t ; goto 194
438       881 sz=f ; if(r2.gt.r1)sz=t ; goto 194
439     cx082
440        82 goto 200
441     cand
442        83 sz=f ; if(s2.ne.f.and.s1.ne.f)sz=t ; goto 194
443     cor
444        84 sz=f ; if(s2.ne.f.or. s1.ne.f)sz=t ; goto 194
445     csign
446        85 sz=1 ; if(s1.lt.0)sz=-1  ; s(z)=sz ; goto 200
447       885 rz=1 ; if(r1.lt.0)rz=-1. ; r(z)=rz ; goto 200
448     ccmdp -- call command processor of a string
449        86 if(ty(z).ne.y6)goto 905 ; call justr(ju,mm,da(z),dz(z),t)
450           n=dz(z)-da(z)+1 ; call cmdproc(ju,n,s(z)) ; goto 194
451     cprom -- print "x?" prompt where x is ascii code param
452        87 call prompt(s1) ; goto 199
453     ccodr   -- code real -- num str fmt =codr, str fmt codr is,
454        88 if(ty(z).ne.y6.or.ty(z-1).ne.y6)goto 905
455           call justr(jv,mm,da(z),dz(z),t) ; z=z-2 ; if(a.ne.f)goto 888
456           call justr(ju,mm,da(z+1),dz(z+1),t); decode(ju,jv)sz; goto 291
457       888 encode(ju,jv)s(z) ; call justr(ju,mm,da(z+1),dz(z+1),f)
458           a=f ; goto 190
459     cio     -- array io -- <vect> <fmt> <lu> =io
460        89 y=ty(z-2) ; if(ty(z-1).ne.y6.or.y.lt.y4)goto 905
461           call justr(jv,mm,da(z-1),dz(z-1),t) ; sz=0
462           j=da(z-2) ; k=dz(z-2) ; n=(k-j)/4+1 ; if(j.gt.k)goto 194
463           if(a.eq.f)y=y-3 ; a=f ; goto(1089,1089,3089,4089,4089,6089)y
464      1089 read (s1,jv,end=8089,err=9089)(mm(i),i=j,k) ; goto 194
465      3089 read (s1,jv,end=8089,err=9089)(ju(i),i=1,n)
466           call justr(ju,mm,j,k,f)                     ; goto 194
467      4089 write(s1,jv          ,err=9089)(mm(i),i=j,k) ; goto 194
468      6089 call justr(ju,mm,j,k,t)
469           write(s1,jv          ,err=9089)(ju(i),i=1,n) ; goto 194
470      8089 sz= 1 ; goto 194
471      9089 sz=-1 ; goto 194
472     cx090
473        90 goto 200
474     cis8    -- is in octal
475        91 write(6,1091)s(z) ; goto 200
476     1091 format(1x,o12)
477     cis4    -- is in a4 format
478        92 write(6,1092)s(z) ; goto 200
```

```
479      1092 format(1x,a4)
480     crstr -- read a string
481        93 if(ty(z).ne.y6)goto 905 ; read(lur,893)(ju(i),i=1,20)
482           call justr(ju,mm,da(z),dz(z),f) ; goto 200
483       893 format(20a4)
484     cisd
485        94 y =ty(z) ; i="scal"; q="stat"; i2=" "; q2=i2; j=0; k=0; n=0
486           n1=dy(z) ; if(y.lt.y4)goto 894
487           n =dn(z) ; j=da(z)-aa(n)*yy(y)+1; k=dz(z)-aa(n)*yy(y)+1
488           i1=nm(n) ; i=mm(i1+2) ; if(mm(i1+1).gt.4)i2=mm(i1+3)
489           if(n1.lt.1.or.n1.gt.zn)goto 894
490           q1=nm(n1); q=mm(q1+2) ; if(mm(q1+1).gt.4)q2=mm(q1+3)
491       894             write(6,1094)i,i2,   j ,   k ,q,q2,ynam(y)
492           if(a.ne.f)write(6,2094)n   ,da(z),dz(z),  n1,y ; a=f ; goto 200
493       1094 format(1x,2a4,"[",i6,",",i6,"]",6x,2a4,1x,a4)
494       2094 format(1x, i8," ",i6,",",i6," ",6x, i8,1x,i4)
495     cis      -- output is scalers, arrays and strings
496        95 y=ty(z) ; n=dn(z) ; k=min(y,4) ; goto(1095,2095,3095,4095)k
497       1095 write(6,1795)s(z) ; goto 200
498       2095 write(6,1895)r(z) ; goto 200
499       3095 write(6,1995)s(z) ; goto 200
500       4095 j=da(z)/yy(y) ; n=dz(z)/yy(y) ; if(j.le.0.or.j.gt.n)goto 895
501           if(y.eq.y4)write(6,1495)(mm(i),i=j,n)
502           if(y.eq.y5)write(6,1595)(mm(i),i=j,n)
503           if(y.ne.y6)goto 200 ; call justr(ju,mm,da(z),dz(z),t)
504           n=(dz(z)-da(z))/4+1  ; write(6,1695)(ju(i),i=1,n) ; goto 200
505       1495 format(10(1x,i5))
506       1595 format(5(1x,f11.5))
507       1695 format(1x,20a4)
508       1795 format(1x,i10)
509       1895 format(1x,f16.5)
510       1995 format(1x,r1)
511        895 print, "<null>" ; goto 200
512     cx096 x097 x098 x099 x100 -- plt fare  rmm tv
513        96 j=da(z)/yy(y) ; n=dz(z)/yy(y)
514           call vvplat(mm(j),n-j+1,4hline)
515           call tvsend ; goto 200
516        97 call tvfare(i,xfare,yfare,j)
517           z=z+1 ; s(z)=xfare ; ty(z)=y2 ; sz=yfare ; goto 291
518        98 n=s(z) ; sz=mm(n) ; z=z-1 ; if(a.eq.f)goto 291
519           mm(n)=s(z) ; a=f ; goto 190
520        99 n=s(z) ; sz=tv(n) ; z=z-1 ; if(a.eq.f)goto 291
521           ipool(n)=s(z) ; a=f ; goto 190
522       100 goto 200
523     c--101---------------------------------------
524     cf g h i j k
525       101 n=000560+l     ; goto 185
526       102 n=000600+l     ; goto 185
527       103 n=000640+l     ; goto 185
528       104 n=000680+l     ; goto 185
529       105 n=000720+l     ; goto 185
530       106 n=000760+l     ; goto 185
531     cpc
532       107 n=000480+l ; k=pn(l) ; adr=aa(k) ; goto 182
533     cx108
534       108 goto 200
535     cmem
536       109 n=s(z)      ; goto 181
537     cstk
538       110 k=1 ; if(a.ne.f)k=2 ; n=000960+z-k-s(z) ; goto 181
```

```
c,
  111 goto 199
cxchg
  112 sz=ty(z) ; ty(z)=ty(z-1) ; ty(z-1)=sz
      sz=dn(z) ; dn(z)=dn(z-1) ; dn(z-1)=sz
      sz=da(z) ; da(z)=da(z-1) ; da(z-1)=sz
      sz=dz(z) ; dz(z)=dz(z-1) ; dz(z-1)=sz
      sz=dy(z) ; dy(z)=dy(z-1) ; dy(z-1)=sz ; goto 200
cpush
  113 ty(z+1)=ty(z) ; dn(z+1)=dn(z) ; dy(z+1)=dy(z)
      da(z+1)=da(z) ; dz(z+1)=dz(z) ; z=z+1 ; goto 199
clkup -- access to lookup subroutine
  114 if(ty(z).ne.y6)goto 905 ; call justr(ju,mm,da(z),dz(z),t)
      n=dz(z)-da(z)+1 ; s(z)=lookup(ju,n) ; goto 184
crwel    -- r/w an element of a descr-ed array
  115 z=z-1
 1115 k=z
 2115 y=ty(k)          ; if(y.lt.y4)goto 905
      r=da(k)+s1-1     ; if(s1.lt.1.or.r.gt.dz(k))goto 906
      if(a.ne.f)y=y-3 ; a=f ; goto(4116,5116,6116,4115,5115,6115)y
c     ----intg----
 4115 s (z)=mm(n)      ; goto 184
 4116 y=ty(z-1)        ; s1=s(z-1) ; goto(4119,4117,4118,905,905,905)y
 4117 s1=r1
 4118 mm(n)=s1         ; goto 199
c     ----real----
 5115 s (z)=mm(n)      ; goto 284
 5116 y=ty(z-1)        ; s1=s(z-1) ; goto(5117,5118,5117,905,905,905)y
 5117 r1=s1
 5118 mm(n)=s1         ; goto 199
c     ----strg----
 6115 j=n-3 ; call bits(mm,j,s(z  ),4,.false.) ; goto 384
 6116 y=ty(z-1)        ; s1=s(z-1) ; goto(6119,6117,6118,905,905,905)y
 6117 s1=r1
 6118 j=n-3 ; call bits(mm,j,s1,4,.true. ) ; goto 199
csubi subj
  116 s1=ii(d) ; goto 1115
  117 s1=jj(d) ; goto 1115
csui1 sui2 sui3
  118 s1=ii(d) ; k=z   ; if(a.ne.f)k=z-1 ; z=z+1 ; goto 2115
  119 s1=ii(d) ; k=z-1 ; if(a.ne.f)k=z-2 ; z=z+1 ; goto 2115
  120 s1=ii(d) ; k=z-2 ; if(a.ne.f)k=z-3 ; z=z+1 ; goto 2115
ckint    -- read an integer constant
  121 p=pa(d)+2 ; pa(d)=p ; sz=mm(p) ; goto 191
cvint    -- r/w variable -- rl=0 -- auto-rtn (affected by rl)
  122 n=pa(l)+2 ; pa(l)=n ; goto 185
ckrea    -- read a real constant onto stack
  123 p=pa(d)+2 ; pa(d)=p ; sz=mm(p) ; goto 291
cvrea    -- r/w a real variable -- rl=0 -- auto-rtn
  124 n=pa(l)+2 ; pa(l)=n ; y=y2 ; adr=1 ; goto 282
caint    -- int array descr -- rl=0 -- auto-rtn (affected by rl)
  125 y=y4
 1125 z=z+1 ; n=pn(l)    ; g=az(n)
      j=p-mm(p+1)        ; if(j.le.p.or.j.gt.g)goto 908  ; pa(l)=j
      k=p+mm(p+2)+2      ; if(k.lt.p+2)k=p+2   ; if(k.gt.j)k=j
      ty(z)=y ; dn(z)=n ; da(z)=p+3 ; dz(z)=k ; dy(z)=0 ; goto 190
carea    -- real array descr -- rl=0 -- auto-rtn
  126 y=y5 ; goto 1125
cvdes    -- r/w a descriptor variable -- auto-rtn
  127 n=pn(d) ; p=pa(d)    ; pa(d)=p+6 ; if((p+6).gt.pz(d))goto 908
```

```
599           if(a.ne.f)goto 3127
600       1127 z=z+1
601           y=mm(p+2) ; ty(z)=y  ; if(y.lt.1.or.y.gt.zy)goto 905
602           n=mm(p+3) ; dn(z)=n  ; if((n.lt.1.or.n.gt.zn).and.y.ge.y4)goto 905
603           j=1       ; k=1      ; if(y.lt.y4)goto 2127 ; j=aa(n) ; k=az(n)
604       2127 n=mm(p+4)+j-1;da(z)=n; if((n.lt.j.or.n.gt.k).and.y.ge.y4)k=k
605           n=mm(p+5)+j-1;dz(z)=n; if((n.lt.j.or.n.gt.k).and.y.ge.y4)k=k
606           dy(z)=mm(p+6) ; goto 100
607       3127 mm(p+2)=ty(z)          ; n=dn(z) ; mm(p+3)=n ; mm(p+6)=dy(z)
608           j=1                     ; if(ty(z).ge.y4)j=aa(n)
609           mm(p+4)=da(z)-j+1       ; mm(p+5)=dz(z)-j+1 ; a=f ; goto 200
610     cvdas -- r/w descr of array only
611        128 n=pn(d) ; p=pa(d) ; pa(d)=p+6 ; if((p+6).gt.pz(d))goto 908
612           if(a.ne.f.and.ty(z).ge.y4)goto 3127 ; goto 1127
613     ca    -- a register for whole descr -- vulnerable to garcol
614        129 if(a.ne.f)goto 1129 ; z=z+1
615           ty(z)=aty; dn(z)=adn; da(z)=ada; dz(z)=adz; dy(z)=ady; goto 100
616       1129 aty=ty(z); adn=dn(z); ada=da(z); adz=dz(z); ady=dy(z); a=f
617           goto 200
618     cb    -- b register for whole descr -- vulnerable to garcol
619        130 if(a.ne.f)goto 1130 ; z=z+1
620           ty(z)=bty; dn(z)=bdn; da(z)=bda; dz(z)=bdz; dy(z)=bdy; goto 100
621       1130 bty=ty(z); bdn=dn(z); bda=da(z); bdz=dz(z); bdy=dy(z); a=f
622           goto 200
623     cdump   -- (dump)all dumps, (n; dump)some dumps, (=dump)off
624        131 dlv=u ; u=zu ; if(a.ne.f)dlv=0 ; a=f ; goto 200
625     cgarc   -- garbage collect -- to get print-out use =garc
626        132 call garcol(a,e) ; a=f ; if(e.ne.f)goto 908 ; goto 200
627     cstat
628        133 if(a.eq.f)read(1)(mm(i),i=1,zm)
629           if(a.ne.f)write(1)(mm(i),i=1,zm) ; a=f ; goto 200
630     cgmem   -- pseudo op for all of memory
631        134 goto 200
632     clure
633        135 lur=lurd ; if(a.ne.f)lur=5 ; a=f ; goto 200
634     cpars   -- access to parse subroutine
635        136 if(ty(z-3).ne.y4.or.ty(z-2).ne.y4)goto 905
636           if(ty(z-1).ne.y1.or.ty(z  ).ne.y1)goto 905
637           j=da(z-3) ; k=da(z-2) ; call parse(mm(j),mm(k),s(z-1),s(z))
638           goto 200
639     c.      -- no operation
640        137 goto 200
641     cce     -- numerical access to any code escape
642        138 ce=s(z) ; z=z-1 ; if(ce.lt.1.or.ce.gt.zh)goto 908 ; goto 620
643     cstop   -- exit from geolab
644        139 return
645     ctrpt   -- tty (end of line) interrupt
646        140 pa(1)=1 ; pz(1)=1 ; aa(xty)=1 ; n=0 ; k=32 ; if(c.ne.1)k=94
647       1140 if(lur.eq.5)call prompt(k) ; read(lur,1540,end=1940)tbuf
648     c     -the following lines are for multics bksp and e commands
649           if(bksp.eq.1)call backsp(tbuf,80) ; if(c.eq.1)a=f
650           if(tbuf(1).ne.101.or.tbuf(2).ne.32)goto 1240
651           tbuf(2)=39 ; tbuf(80)=39
652     c     -call parser, if unfinished line loop back to read again
653       1240 call parse(tbuf,mm,pz(1),n) ; if(pz(1).gt.120)goto 1640
654           k=n+48  ; az(xty)=pz(1) ; if(n)1740,300,1140
655       1540 format(80a1)
656       1640 print, "parse error: overflow" ; goto 1840
657       1740 print, "parse error: ) or '  "
658       1840 if(lur.ne.5)write(6,1540)tbuf
```

```
 1940 if(lur.ne.5)rewind lur ; lur=5 ; goto 140
cectl estk enam edec etyp eidx exup emis
  141 emsg(1)="cntl" ; emsg(2)=" sta" ; emsg(3)="ck e" ; goto 1141
  142 emsg(1)="scra" ; emsg(2)=" sta" ; emsg(3)="ck e" ; goto 1141
  143 emsg(1)="unde" ; emsg(2)="fin " ; emsg(3)="op e" ; goto 1141
  144 emsg(1)="re-d" ; emsg(2)="ecla" ; emsg(3)="re e" ; goto 1141
  145 emsg(1)="stac" ; emsg(2)="k ty" ; emsg(3)="pe e" ; goto 1141
  146 emsg(1)="subs" ; emsg(2)="cr i" ; emsg(3)="dx e" ; goto 1141
  147 emsg(1)="^ of" ; emsg(2)="f op" ; emsg(3)="   e" ; goto 1141
  148 emsg(1)="misc" ; emsg(2)="ella" ; emsg(3)="ny e" ; goto 1141
 1141 emsg(4)="rror" ; emsg(5)=" on " ; write(6,1241)emsg ; goto 2
 1241 format(1x,10a4)
c! tty  -- pseudo ops
  149 goto 200
  150 goto 200
c--error-recovery-----------------------------------------
  900 ce=140   ; goto 990
  901 ce=141   ; goto 910
  902 ce=142   ; goto 910
 1902 if(z.lt.zz)goto 902 ; n=z-(z-zz)/2+1
      do 2902 i=n,z        ; j=zz+i-n
         ty(j)=ty(i); dn(j)=dn(i); da(j)=da(i); dz(j)=dz(i); dy(j)=dy(i)
 2902    continue
      z=j                  ; goto 200
  903 ce=143   ; goto 910
  904 ce=144   ; z=z-1 ; goto 910
  905 ce=145   ; goto 910
  906 ce=146   ; goto 910
  907 ce=147   ; goto 910
  908 ce=148   ; goto 910
  910 e=f      ; goto 930
  920 e=t
  930 n =mm(p) ; nmmp=2h??     ; if(n.ge.1.and.n.le.zn)nmmp=nm(n)
      i =pn(d) ; npnd=2h??     ; if(i.ge.1.and.i.le.zn)npnd=nm(i)
      n =dn(z) ; ndnz=2h??     ; if(n.ge.1.and.n.le.zn)ndnz=nm(n)
      nmmp=mm(nmmp+2) ; npnd=mm(npnd+2) ; ndnz=mm(ndnz+2)
      if(dlv.lt.0)goto 940
      y =ty(z) ; ntyz=2h??     ; if(y.ge.1.and.y.le.zy)ntyz=vnam(y)
      ipc=0 ; j=0 ; k=0        ; if(npnd.ne.2h??)ipc=p-aa(i)+1
      if(ndnz.eq.2h??)goto 932 ; j=da(z)-aa(n)+1 ; k=dz(z)-aa(n)+1
  932 sz= s(z)                 ; if(y.eq.y2)sz=r(z)
      if(y.le.y3)write(6,934)nmmp,npnd,ipc,c,d,sz,z,ntyz
      if(y.ge.y4)write(6,934)nmmp,npnd,ipc,c,d,sz,z,ntyz,ndnz,j,k
      if(e.ne.f)goto 400
  940 s(1)=c   ; s(2)=d  ; s(3)=p  ; s(4)=z ; s(5)=s(6)
      ecis=c   ; edis=d  ; epis=pa(d)  ; ezis=z ; eptr=mm(epis)
      enon=xup ; if(eptr.ge.1.and.eptr.le.zn)enon=nm(eptr)
      epnd=pn(d) ; enin=nm(epnd)
      if(ce.eq.141)c=1 ; if(ce.eq.141)d=1 ; if(ce.eq.142)z=zz
      emsg(6)=nmmp    ; emsg( 7)="    "  ; emsg(8)=" in "
      emsg(9)=npnd    ; emsg(10)="    "
  990 ptr=cenu(ce)         ; goto 400
  934 format(1x,a4," in " ,a4, "[" ,i3, "]  cntl(" ,i2, "," ,i2, ")" ,
       i9, "=stk[" ,i2, "]  " ,a4,3x,a5, "[" ,i3, "," ,i3, "]" )
      end
```

```
.....glin.....May 17, 1978....15:57.....

  1    skip('------f i l e 0 2------g l i n------16 may 78------')
  2    =skip      (af  quo  =a  =skip  h        fa decl)  quo op h decl
  3    op op2     (af 2$quo =a =skip  h^plus   fa decl)
  4    op op3     (af 2$quo =a =skip m=k+h-2+^=m,=skip m-k+h k=m,fa decl)
  5    =op !(quo!)(^     typ 6 eq ifthen lkup   cnam)
  6    op bang    (0$ quo !)
  7    op erase   (nuo 0 =decl)
  8    op note    (skip)
  9    op qu      (quo)
 10    note('------looping and conditionals------')
 11    op do      (of i =q, 1=k, =i, 0      =of i, pc=j,  dolu  q=of i,)
 12    op do      (of i =g, 1=k, =i, 1 minu=of i, pc=j,  dolu  g=of i,)
 13    op do2     (of i =g, 1=k, =i, 1 minu=of i, pc=j,  dolu  g=of i,)
 14    op do3     (of i =g,  =k, =i, k minu=of i, pc=j,  dolu  g=of i,)
 15    op dohak   (of i=g, -1=k,1=i, k minu=of i, pc=j,  dolu  g=of i,)
 16    op while   (of i =k,      0      =of i, pc=i, skip skip pc=j, whiz k=of i,)
 17    op if      (3 ifju  skip ^ 1   ^ skip)
 18    op ifthen  (2 ifju  skip   1   ^      )
 19    op ifelse  (2 ifju       ^ 1    skip)
 20    op if3     (push 0ge 4ifju ^skip skip1 4ifju skip^skip1 skip skip^)
 21    note('------descr ops------')
 22    op part    (af  beg ^  plus          1minu=beg   ^ plus 1minu=fin, fa)
 23    op blo     (af  beg ^ 1minu ^=k tims plus=beg   k plus 1minu=fin, fa)
 24    op lenx    (beg=j, fin j minu 1 plus)
 25    op size    (^ lenx only)
 26    op word    (af ^ makd ^ fa rwel)
 27    op w3      (af ^ makd 3 fa rwel)
 28    op len     (af ^ makd 4 fa rwel)
 29    op w5      (af ^ makd 5 fa rwel)
 30    op lim     (af w3 (^ dens=k,) chs 2minu k tims)
 31    op makall  (all=a  lenx=len a,)
 32    op lenrow  (af ^ makd slip skip  of f=f,  fa ^  =eot)
 33    op lencol  (len(^=a) lenrow a  div)
 34    op dens    (typ 6eq  3tims 1plus)
 35    op abeg    (typ 6eq 12tims 4plus)
 36    op rest    (af  =a  fin    1plus=beg,  abeg  lim a  plus=fin,  fa)
 37    op fir     (af                         beg   1minu ^plus=fin,  fa)
 38    op aft     (af                         beg         ^plus=beg,  fa)
 39    op las     (af                         fin   ^minu 1plus=beg,  fa)
 40    op all     (af  =a  abeg=j 1plus=beg,  j     lim a  plus=fin,  fa)
 41    op none    (fir 0)
 42    op stat    (0=dyn,)
 43    op int     (1=typ,)
 44    op rea     (2=typ,)
 45    op ckln    (^ =a,  len a  lim a  minx=len a,)
 46    op nptr0   (nptr)
 47    op namo    (nam xchg,)
 48    op substr  (af ^ part ^ ^ fa active2)
 49    op piece   (af ^ part ^ ^ fa active2)
 50    op first   (af ^ fir  ^   fa active2)
 51    op last    (af ^ las  ^   fa active2)
 52    op firline(find 10 1minu=k, fir k)
 53    note('------dynamic descr ops------')
 54    op dynm    (af  nptr=dyn,  fa)
 55    op twod    (,dynm dim2)
 56    op dim2    (af  =k, stat=a  blo k (lenrow a)  fa)
 57    note('------variables and arrays------')
 58    op intg    (op !quo(vint -2.. 0..))
```

```
op real    (op !quo(vrea -2.. 0..))
op var     (op !quo(vdes -6.. 1.. 0.. 0.. 0.. 0..))
op desc    (op !quo(vdas -6.. 1.. 0.. 0.. 0.. 0..))
op iarri   (op2(aint 0.. 0..)(^=k)          -2k zdec k)
op rarri   (op2(area 0.. 0..)(^=k)          -2k zdec k)
op iarr    (op3(aint 0.. 0..)(^=k)(active) -2k zdec k)
op rarr    (op3(area 0.. 0..)(^=k)(active) -2k zdec k)
op iar2    (op3(aint 0.. 0..)(par2)(kint -2.. 0.. twod) -2j zde2 j)
op rar2    (op3(area 0.. 0..)(par2)(kint -2.. 0.. twod) -2j zde2 j)
op strni   (op2(' ')(^    1minu 4div=k)           -3k zdec 0)
op stri    (op2(' ')(^ =j 1minu 4div=k)           -3k zdec j)
op strn    (op3(' ')(^    1minu 4div=k)(active)  -3k zdec 0)
op str     (op3(' ')(^ =j 1minu 4div=k)(active)  -3k zdec j)
op str2    (op3(' ')(par2 1minu 4div=k)(kint -2.. 0.. twod) -3k zde2 j)
op zdec    (minu=w3 a, ^=len a,  xega)
op zde2    (of i=mm[m-2], zdec ^)
op xega    (a makd cnam)
op par2    (up2 =of k up2 =of i tims =of j)
note('------two dim var length arrays------')
op stra    (op3(' ')(par2 +k-1  4div=k)(kint -2.. 0.. twov) -3k zde2 j)
op twov    (af, dynm dim2v fa)
op dim2v   (af =k dim2 fir(a all[len a k plus=j]) fa activev)
op activev(act3 [ : xvvmv)
op xvvmv   (fir(lenrow a) xchg vvm k a all =[?^j],)
note('------dyadic arithmetic------')
op quo2    (3$ quo)
op upty    (3$ ^ typ 4 lt)
op asvx    (xchg=b, lenx quo2=a, do(b    sui2  a cnam  =sui1,) )
op avsx    (    =b, lenx quo2=a, do(sui1 b     a cnam  =sui1,) )
op avvx    (        lenx quo2=a, do(sui2 sui2  a cnam  =sui2,),)
op apo2    (typ 4 lt  if(upty if ^ asvx)(upty if avsx avvx))
op apl2    (up2 typ2  ^ asvx asvx   avsx avvx avvx   avsx avvx avvx)
op +       (apl2 plus)
op -       (apl2 minu)
op *       (apl2 tims)
op /       (apl2 div )
op **      (apl2 pow )
op ==      (apl2 eq  )
op /=      (apl2 ne  )
op <       (apl2 lt  )
op <=      (apl2 le  )
op >       (apl2 gt  )
op >=      (apl2 ge  )
op and     (apl2 andx)
op or      (apl2 orx )
op min     (apl2 minx)
op max     (apl2 maxx)
op mod     (apl2 modx)
op dyav    (apl2 (plus 2.0 div) )
op rms     (apl2 (=v, push tims v v tims plus sqr))
note('------monadic arithmetic------')
op avx     (lenx quo2=a, do(sui1  a cnam  =sui1,))
op apo1    (upty if ^ avx)
op apl1    (up2 typ1  ^ avx avx)
op sin     (apl1 si  )
op cos     (apl1 co  )
op tan     (apl1 ta  )
op asin    (apl1 asi )
op acos    (apl1 aco )
op atan    (apl1 ata )
```

```
119     op ln      (apl1 logx)
120     op log     (apl1 l10x)
121     op exp     (apl1 ex  )
122     op abs     (apl1 ab  )
123     op sqrt    (apl1 sqr )
124     op --      (apl1 chs )
125     op crop    (apl1 fix )
126     op round   (apl1 rnd )
127     op not     (apl1 notx)
128     op sign    (apl1 sig )
129     op incr    (pc=k, ^ 1plus k=pc, =^,)
130     op decr    (pc=k, ^ 1minu k=pc, =^,)
131     op flip    (pc=k, ^ ^ k=pc, =^, =^,)
132     note('------arithmetic constants------')
133     op pi      (3.1415926)
134     op E       (2.7182818)
135     op true    (1)
136     op fals    (0)
137     note('------missing data ops------')
138     op neix    (umod=k, 0=uyes=uno1=uno2, ^=umod, ^=unew, 2$eval k=umod,)
139     op nei0    (neix 1    0)
140     op nei9    (neix 1 usym)
141     op nein    (neix 2 usym)
142     op shou    ('  yes   no1   no2' is, mr part 3914 3 is,)
143     note('------move ops------')
144     op avsm    ( =b, lenx b  xchg                    do(    =sui1 ),)
145     op avvm    (xchg lenx=k, xchg lenx k minx=of b do(sui1=sui2,),)
146     op mov     (^       typ 4 lt  if vsm vvm)
147     op move    (^ xchg typ 4 lt  if vsm vvm)
148     op active (act3 [ : xvvm)
149     op xvvm    (all xchg vvm =a  h=len a,  makd cnam)
150     op active2(act3 [ : xvvm2)
151     op xvvm2   (xchg vvm)
152     op zero    (mov  0)
153     op blank   (mov 32)
154     op cat     (ckcat fin=k, rest ^cat2 vvm  makd=a   k h plus j minu=len a, cr
155     op !       (ckcat fin=k, rest ^cat2 vvm  makd=a   k h plus j minu=len a, cr
156     op catsp   (cat ' ' cat ^)
157     op cat1    (typ1 cat1x  . .)  op cat1x(strsca=scrstr)
158     op cat2    (typ1 strsca . .)
159     op ckcat   (cat1  abeg=of j,  nam=k 2$calr eq k xqm eq orx ifthen (=scrstr)
160     op insertn(^ push=scrstr3, fir (^=k) cat ^ cat (scrstr3 aft k))
161     op insert (insertn (^ findn (^:1) ^ =k,)  k  ^)
162     note('------misc array ops------')
163     op iota    (lenx do(i=sui1,))
164     op iotn    (lenx ^ =a, do(a i tims =sui1,))
165     op alph    (lenx do(i 1minu 26modx 97plus=sui1,))
166     op across  (lenx=j, push:1 quo=a,? j do(sui? a cnam .) only)
167     op sum     (across plus)
168     op deriv   (push:1=a, lenx ? xchg do(a sui?=a minu chs=sui1,))
169     op integ   (     0=a, lenx         do(a sui?   plus =a =sui1,))
170     op reverse(lenx+1=k/2do(sui1 k i=j minu=i,sui? xchg=sui2, j=i,=sui1,))
171     op rotate (^ - 1=j, lenx=k do(sui2 i=h j plus k modx 1plus=i, =sui1, h=i,)
172     op random (ranseed 734593modx 1777tims 524287plus=ranseed 100modx)
173     op random1(ranseed 734593modx 1777tims 524287plus=ranseed 10000modx/10000.
174     op rand    (lenx do(random=sui1,))  intg ranseed  1=ranseed,
175     op ranlett(lenx do(random1 26tims 97plus fix =sui1,))
176     op find    (^=a,       lenx       do(       sui1 a eq 5ifju          )0] drc
177     op find2   (^=a, ^=b, lenx 1xchg do( 7ifju sui1 b eq 8ifju sui1 a ne)0] drc
178     op finx2   (^=a, ^=b, lenx 1xchg do( 7ifju sui1 b eq 8ifju sui1 a eq)0] drc
```

```
179     op findn  (^=a, ^=k, lenx 1xchg do(15ifju k-1=k 0eq 8ifju sui1 a ne)01 dro
180     op label  (skip)
181     op goto   (calr makd  find2 (nptr0 label) nptr  ckgoto  =pc,)
182     op ckgoto (nush 0eq ifthen("goto!' error in '!(23calr nmstr) is, rset))
183     op ~      (ampflag if(0=ampflag, scriar)
184                          (1=ampflag, while(^ ampflag)(i=len scriar, =scriar:i,
185     op &      (ampf1 if(0=ampf1, ampf2-1 if scrrar (scrrar=scriar))
186                        (1=ampf1=ampf2, while(^ ampf1)
187                        (i=len scrrar, typ ampf2 maxx=ampf2, =scrrar:i,) ))
188               intg ampflag  0=ampflag,     intg ampf1  intg ampf2  0=ampf1,
189     note('------input/output------')
190     op read   (^     ^ 1  io checkio)
191     op readv  (^ vfmt 1  io checkio)
192     op write  (^     ^ 1 =io checkio)
193     op readt  (^     ^ 5  io checkio)
194     op readtv (^ vfmt 5  io checkio)
195     op writet (^     ^ 6 =io checkio)
196     op rewind (^ 13 mops)
197     op rew    (rewind  1)
198     op endfile(^ 14 mops)
199     op enf    (endfile 1)
200     op checkio(=eof  if3 ('io error'is,) . ('end of file'is,))   intg eof
201     op rline  (read (scrstr fir 80) a4  is)
202     op open   (exec 'io open  file01 si')
203     op open2  (exec 'io open  file01 soi')
204     op close  (exec 'if [opened   file01] -then "io close  file01"')
205     op detach (exec 'if [attached file01] -then "io detach file01"')
206     op filex  (exec(null ! 'io attach file01 vfile_ ' ! ^))
207     op file1  ('please use ''file'' operator now' is,)
208     op file   (rew  close  detach  filex ^  open)
209     op filc   (filex (of"))
210     op trash  (exec 'fo gl_trash')
211     op untrash(exec 'co')
212     op a4     ('(20a4)')
213     op vfmt   ('(v)')
214     note('------subscripting------')
215     op [      (af eval fa  typ0 rwel rwel dyn0)
216     op :      (af ^    fa  typ0 rwel rwel dyn0)
217     op star   (dynm stardyn  0)
218     op stardyn(, af 0=dyn, f ifthen(xchg=a vsm, a))
219     note('------window into system------')
220     op sysc   (361 mem 1 minu)
221     op sysd   (=calr)
222     op me     ( calr)
223     op myname (calr nmstr)
224     op xqm    (334 mem)
225     op xty    (350 mem)
226     op zm     (353 mem)
227     op zn     (354 mem)
228     op z      (365 mem)
229     op mptr   (367 mem)
230     op m      (367 mem)
231     op lur    (368 mem)
232     op ncyc   (374 mem)
233     op ndec   (385 mem)
234     op iter   (386 mem)
235     op nloc   (387 mem)
236     op gbeg   (388 mem)
237     op lurd   (389 mem)
238     op umod   (3911 mem)
```

```
239     op usym    (rmm:3912)
240     op usymi   (0=umod, usym fix udef=umod,)
241     op ?       (usym)
242     op unew    (rmm:3913)
243     op uyes    (3914 mem)
244     op uno1    (3915 mem)
245     op uno2    (3916 mem)
246     op udef    (3917 mem)
247     op ecis    (3931 mem)
248     op edis    (3932 mem)
249     op enis    (3933 mem)
250     op ezis    (3934 mem)
251     op entr    (3935 mem)
252     op enon    (3936 mem)
253     op enin    (3938 mem)
254     op nc      (ncyc ncy2 minu 8 minu is, ncyc=ncy2,)  intg ncy2
255     op ncx     (ncyc=ncy2,)
256     op rstk    (6=z)
257     op inst    (qu inst:12, 'xxxx')
258     op fourh   (" move '1234', qu fourh:7)
259     op sai     (af fin+3/4=k, makd 5=bec, k=fin, fa active2)
260     note('------arrays of the system------')
261     op mm      (1 guts)
262     op rmm     (2 guts)
263     op lmm     (3 guts)
264     op mtty    (af  mm  part    1 120  fa)
265     op tbuf    (af  mm  part  121  80  fa)  str sbuf 80
266     op cenu    (af  mm  part  201 150  fa)
267     op zlim    (af  mm  part  351  10  fa)
268     op cetc    (af  mm  part  361  40  fa)
269     op ca      (af  mm  part  401  40  fa)  iarr ca2 40
270     op nn      (af  mm  part  441  40  fa)  iarr nn2 40
271     op na      (af  mm  part  481  40  fa)  iarr na2 40
272     op nz      (af  mm  part  521  40  fa)  iarr nz2 40
273     op ff      (af  mm  part  561  40  fa)
274     op gg      (af  mm  part  601  40  fa)
275     op hh      (af  mm  part  641  40  fa)
276     op ii      (af  mm  part  681  40  fa)
277     op jj      (af  mm  part  721  40  fa)
278     op kk      (af  mm  part  761  40  fa)
279     op ty      (af  mm  part  801  40  fa)
280     op dn      (af  mm  part  841  40  fa)
281     op da      (af  mm  part  881  40  fa)
282     op dz      (af  mm  part  921  40  fa)
283     op dy      (af  mm  part  961  40  fa)
284     op s       (dy)
285     op R       (dy 5=typ,)
286     op nm      (af  nm  part 1001 900  fa)
287     op name    (af  nm  dynm nam2  fa)
288     op nam2    (af =k, stat:k=j,  j mmstr  fa)
289     op mmstr   (=j, af lmm part(j 2plus 4tims 3minu)(mm[j 1plus]) fa)
290     op nmstr   (=j, nm:j mmstr)
291     =op mmstr  (4mops)
292     =op nmstr  (3mops)
293     op "       (ug  of gu nptr  nmstr)
294     op aa      (af  mm  part 1901 900  fa)
295     op az      (af  mm  part 2801 900  fa)
296     op emsg    (af  lmm part 15681 40  fa)
297     op emsi    (af  mm  part 3921  10  fa)
298     op rmem    (af  mm  part mptr (zm m minu 1 plus)  fa)
```

```
note('------graphics------')
op plt     (1 xchg 1  9 mops  only)
op tic     (0 xchg 4  9 mops  only)
op ticn    (^ xchg 4  9 mops  only)
op bar     (0 xchg 5  9 mops  only)
op barn    (^ xchg 5  9 mops  only)
op plot    (^ plm plt framn)
op tplot   (^ plm plt frame)
op tick    (^ plm tic framn)
op ttick   (^ plm tic frame)
op barr    (^ plm bar framn)
op tbarr   (^ plm bar frame)
op plotxy  (^ plmx  ^ plmy  pltxy  framn)
op pltn    (     lenx do(  i  sui2  i strnum  tvltr))
op pltc    (^=a, lenx do(  i  sui2  a         tvltr))
op pltxyn  (     lenx do(sui2 sui2  i strnum  tvltr))
op pltxyc  (^=a, lenx do(sui2 sui2  a         tvltr))
op pt      (1 xchg 2 ^mops xchg)
op pltxy   (       1 ^mops      )
op ptxy    (       2 ^mops      )
op page    (.5 .5 827 128 move 'ep' tvltr dly)
op repro   (.5 .5 '' tvltr  3 do dly)
op dly     (' ' 60 do is,)
op ddd     (' '  ^ do is,)
op esc     (.5 .5 '' cat ^ tvltr)
op escolon(esc ':')
op tvltr   (7mops)
op fare    (8mops)
op hilo    (5mops)
op plmx    (hilo =xz, =xa,)
op plmy    (hilo =yz, =ya,)
op plm0    (1=xa, lenx=xz, plmy)
op plm     (plm0  yz-ya*.1=a  +yz=yz,  ya-a=ya,)
op vert    (ya+yz/2. push ^/2.=a plus=yz, a minu=ya,)
op horz    (xa+xz/2. push ^/2.=a plus=xz, a minu=xa,)
op square  (yz-ya xz-xa maxx=b,  vert b  horz b)
op tvpool  (af  rmm part 3953   8  fa)
op itune   (af  mm  part 3961  30  fa)
op tune    (af  rmm part 3961  30  fa)
op tvse    (3952 mem)
op west    (tvpool:1)   op xa(tvpool:1)
op east    (tvpool:2)   op xz(tvpool:2)
op south   (tvpool:3)   op ya(tvpool:3)
op north   (tvpool:4)   op yz(tvpool:4)
op panel   (^ + .05 =tvpool:7,   ^ - .05 =tvpool:8,)
op panup   (^ =tvpool:5, ^ =tvpool:6,)
op bot     (panel 0 .49)                              op top (panel .51 1)
op bot3    (panel 0 .30)  op mid3(panel .35 .45)  op top3(panel .70 1)
op norm    (panel 0   1)
op frame   (af f if framn framt)  op fr(framn)
op framt   (0 mor2 1)   op framn(1 mor2 1)
op genline(flot=b, flot=a, ^ lenx do(a i tims b plus=sui1,))
op pltltsq(ltsq  genline ^  xa a*xa+b xz a*xz+b line)
op play    (0=theta=xloc=yloc, & -100 100 -100 100 &=first tvpool 4,)
op line    (=yvec:2, =xvec:2, =yvec:1, =xvec:1, xvec yvec pltxy)
op fly     (xloc yloc  sin theta*(^=b)+xloc=xloc
                       cos theta*  b  +yloc=yloc)
op go      (fly ^ line)
op turn    (^ * pi / 180 + theta =theta,)
           var xloc  var yloc  var theta  rarr xvec 2  rarr yvec 2
```

```
359     op STAR   (108 do(go 16 turn(5*i) ) )
360     op geostar(3 do(play turn(60*i) star))
361     note('------printer plots------')
362     str2 ppspace 10 30   str ppchar 2  '.*' =ppchar,
363     intg ppxz  intg ppyz  intg ppch  desc ppar  real nsdif
364     op pperase(ppchar:2=ppch,  lenrow ppspace=ppxz,  lencol ppspace=ppyz,
365                ppchar:1 move ppspace,  north-south=nsdif,)
366     op ppput  (=ppar lenx ppxz minx do(ppch=ppspace:ppy:i,))
367     op ppy    (ppar[of i]-south/nsdif*ppyz  +.99999 fix  1maxx  ppyz minx)
368     op ppshow (ppspace iz9,)
369     op ppplot (^ plm  pperase  ppput  ppshow)
370     op pp     (^ plm  pperase  ppput  ppshow)
371     op ppsize (=str2 ppspace ^ ^)
372     note('------op examination------')
373     op ck     (aa:nptr if3 'hard op' 'undefined' 'soft op' is,)
374     op adr    (quo nam=j, aa:j is,)
375     op sho    (^ nmstr lenx 20minx=j, fir j  is,)
376     op isn    (nam=j, sho j)
377     op nslo   (0=j, nm lenx do(sui1 0 ne i tims j maxx=j,), j)
378     op nops   (aa  zn do(sui1 0eq ifth(nm subi push 0ne ifth is4,)))
379     op wipe   (aa  zn do(sui1 0eq ifth(-1=nm subi,)))
380     op mine2  (qbeg=k, aa       zn do(sui1 k ge ifthen(i nmstr is,)))
381     op mine   (qbeg=k, aa null zn do(sui2 k ge ifthen(|(i nmstr)|' '))only is`
382     op xop    (xopnois quo  is)
383     op xopnois(aa[^=a namo=k] if3 'hard op' 'undefined' (strop !a xopx) )
384     op xopx   (scrstr3 none | 'op ' | (of k nmstr) | ' ( ' | scrstr | ')' )
385     op strop  (quo slip null while(eot notx)(cat(of strquo)cat' '))
386     op strquo (uq 1 0 0 of qu quox  typ1 strsca strvec strvec)
387     op strsca (typ 3lt if strnum strlet)
388     op strnum (typ=j, flot=b numscr numfmt=codr,  numscr 31 b 0lt minu
389                b ab 1maxx l10x fix minu=beq, j 2eq precision tims 31plus=fin,)
390     op strlet (=numscr:1, numscr fir 1)
391     op strvec (typ 6lt if strdes strstr)
392     op strdes (nam=j,, j nmstr)
393     op strstr (=b, scrstr2 none cat '''' cat b cat '''')
394     op numfmt ('(f26.10)')  str numscr 26   intg precision  6=precision,
395     note('------scratch space------')
396     iarr scriar 50   str scrstr  200   str scrstr2 200   str scrstr3 200
397     rarr scrrar 50   str scrstr4 200   str scrstr5 200
398     op null   (scrstr none)
399     iarr onei 1   rarr oner 1
400     note('------error recovery------')
401     note( =op ectl  (esav  141ce)  )
402     =op estk  (edsp  'scratch stack underflow')
403     =op enam  (edsp  'undefined operator')
404     =op enam  (af f  if(var !(eptr makd)  eptr makd =cnam)
405                      (edsp 'undefined operator'        )  )
406     =op edec  (a=b, edecead edsp2 're-declare' (b namo  nmstr) )
407     op edecead(curm if(curm=m, 0=curm,)(m=oldm 2$h=oldh plus=m,))
408     op instead(af m=curm, oldm=m, quo=a oldh fa decl  curm=m, 0=curm)
409     =op etyp  (edsp  'stack type')    into oldh  into oldm  into curm
410     =op eidx  (edsp2 'subscript index' (dn:ezis  nmstr) )
411     =op exup  (edsp  '^ off the end of an operator')
412     =op emis  (edsp  'misc')
413      op edyn  (esav 'dynamic descr error' is rset)
414      op ecat  (esav 'cat to a constant error' is rset)
415     -1$ dump
416     op esav   (af ug  tbuf=sbuf, ca=ca2, pn=pn2, pa=pa2,)
417     op edsp   (edsp2 ^ onop)
418     op edsp2  (esav null |^| error on '|^|' in '|inop is, rset)
```

```
419     op onop    (enon mmstr)
420     op inop    (enin mmstr)
421     op what    (ecis do(sho(pn? subi)) who)
422     op whata   (null ecis do(! (pn?:i nmstr) ! ' ') ! (eptr nmstr) is)
423     op whatn   (null     ^ do(! (pn:i  nmstr) ! ' ') is)
424     op who     (sho eptr)
425     op how     (sbuf is,)
426     op fixup   (nptr =mm:epis,)
427     op change  (quo finx2 -2 nptr =i>1 if(nptr=sui1)(skip 'not found'is),,)
428     op changen(quo find2 -2 ^    =i>1 if(^   =sui1)(skip 'not found'is),,)
429     op swop    (af nptr=j, nptr=k, aa:j=h, aa:k=i,
430                 flip(aa:j)(aa:k)  flip(az:i)(az:k)
431                 h 0qt ifthen(k=mm:h,)  i 0qt ifthen(j=mm:i,) )
432     op sysnum2(nm[nptr is=k]'nm'is,is,'aa'is,aa:k=j is>0ifthen(mm:j is,))
433     op sysnums('nm[' ! (nptr=k) ! ']=' ! (nm:k) ! '   ' !
434                'aa[' !       k  ! ']=' ! (aa:k)   is)
435     note('------misc ops------')
436     op let     (af pc=k, ^,skip eval pc=j,k=pc, =^, j=pc,)
437     op only    (xchg,)
438     op o       (stop)
439     op quiet   (=op init(rset))
440     op mark    (mptr=gbeg,)
441     op purge   (gbeg=k,  aa zn do(sui1 k ge ifthen(0=sui1,)))
442     op force   (=op op(af  1=f,  quo=a  =skip  h fa decl))
443     op unforce(=op op(af        quo=a  =skip  h fa decl))
444     op ioa     (exec(null ! 'fo ' !^! ' ; ioa_ "' !^! '" ; co') )
445     op ioaxop  (xopnois ^ =scrstr5,, ioa ^ scrstr5)
446     op rollo   (^=rofo, gbeg=k, aa zn
447                 do(sui1 k ge ifthen(ioaxop (i makd) rofo)) )  str rofo 80
448     op rollin  (file ^ lure)
449     op lookup  (^ lkup)
450     op lookup?(nm[^=b lkup=il=j, nm[nptr0 lookup?]=nm:i, b lkup j=nm:i,)
451     op parsego(3? move LINE, ^=LINE makall  scriar 1 0 pars
452                if('parse error' is)(=k, fir k only cnam))
453     op stan    (^ 15 mops)
454     op decim   (^ stan 2 only)
455     op avg     (^ stan 3 only)
456     op polate (999999 stan 4 ,)
457     op polaten( ^      stan 4 ,)
458     op mor2    (^ 16 mops) op mor3(^ 17 mops) op mor4(^ 18 mops) op mor5(^ 19
459     op dait    (mor3 1)
460     op daitsho(dait =inst=scrstr catsp xchg catsp (3stk) catsp (?stk) is)
461     op ltsq    (1 xchg 2 mor3 2)
462     op ltsqxy (       2 mor3 2)
463     op bits    (mor3 3)
464     op land    (mor3 4)
465     op lor     (mor3 5)
466     op lnot    (mor3 6)
467     note('------special ops------')
468     op e       (^ cmdp,)
469     op exec    (^ cmdp,)
470     op rold    (while(114prom  sbuf rstr:1 47ne)(sbuf cmdp =of h,))
471     op bksp    (384mem)  1=bksp,
472     op dl      (exec('if [exists segment ' ! (^=scrstr2) !
473                 '] -then "delete ' ! scrstr? ! '"'))
474     op fo_on   (dl 'gl_temp'  exec 'fo gl_temp')
475     op fo_off (exec 'co'  dl 'gl_temp')
476     op getmul (fo_on    ^   pounce 'gl_temp'
477                qu scrstr3  part 5 40  1mmm,  scrstr3 all firline   dl 'gl_tem
478     op actfun (getmul (exec('ioa_ [' !^! '] ; co')) )
```

```
479     op cetab  (cetmul ( exec('.l ' |^) exec'co' ) )
480     op f      (exec ('f glin ' | ^))
481     op glin   (exec 'ted -pn >udd>Geolab>JHerriot>glin')
482     note('------data-base-ops------')
483     op dbops  (11 mops)
484     op mmm    (20 mops)
485     op grab   (af ^ ^ fa mmm)
486     op iseg   (af onei dynm isegdyn fa)
487     op rseg   (af oner dynm rsegdyn fa)
488     op isegdyn(af =k,, f if(=onei:1 onei k =mmm,)(onei k mmm, onei:1))
489     op rsegdyn(af =k,, f if(=oner:1 oner k =mmm,)(oner k mmm, oner:1))
490     op ranlook('sensor name:  ' cat sennam is, 21 mops)
491     op dat    (12 mops)
492     op yr     (3091 mem)     78=yr
493     op jl     (3992 mem)    001=jl
494     op iv     (3993 mem)   1440=iv
495     op invl   (3993 mem)
496     op blog   (af   fa first scriar 10   1 fa mmm)
497     op xbottle(ckbottle % yr jl iv  0 100000  1 usymi  0 0 05)
498     op ckbottle(iseg:1 one ifthen
499                 ('error: bottle already initialized' is rset))
500     op ibottle(xbottle 1 =mmm,)
501     op rbottle(xbottle  2=scriar:6,  usym int=scriar:7,  1 =mmm,)
502     op bounce (exec(null | 'in ' | ^ | ' gl_dbop -force'))
503     op sens   (af f if(=sennam bounce(rfile none|dirnam|'>'|sennam))
504                      (sennam))   str sennam 20
505     op sen    (^ =sens,  )
506     op dir    (^ =dirnam,)   str dirnam 80   str rfile 80
507     op lookat (^=sens, ^=yr, ^=jl, ^=iv,)
508     dir '>udd>Geolab>JHerriot>tiltdata'
509     note('------teaching-ops------')
510     op course (file 'course' course2 48  'ok' is, mark)
511     op course2(str2 ctext (^=clen) 160 getcour)   intg clen  intg lesson
512     op getcour(clen*2 do(read LINE '(80r1)' move (ctext stat blo i 80)))
513     op teacher(' 'is, ^=lesson, =op trpt(xteach 140ce))  iarr LINE 80
514     op resume (teacher lesson)
515     op teach  ('geolab course' is,
516                'you will get a 2 line instruction then a "^2" to practice'is,
517                'say "graduate" to exit, "resume" to resume'is, teacher 1)
518     op xteach (lesson clen le ifthen(ctext:lesson is,) ckgrad)
519     op ckgrad (lesson+1=lesson>clen ifthen regtrpt)
520     op graduate('you graduate' is, regtrpt)
521     op regtrpt(-140=aa[nptr0 trpt],)
522     note('------lbl compatable ops------')
523     op isx    (flot is)
524     op is2    (=a is,is a)
525     op iz     (lenx  null xchg  do(cat ' ' cat sui2) is,)
526     op iz2    (=b,  lencol b  do   (b:i is,))
527     op iz9    (=b,  lencol b  dobak(b:i is,))
528     op istk   (' '  z-1-^  z-3  do(cat (s:i) cat ' ') is)
529     op arr    (^ =j,  op2(aint 0.. 0..) j   -2 j minu=w3 a,  j=len a,)
530     op count  (iota)
531     op trade  (xchg)
532     op memsize(353 mem)
533     op garcol (garc)
534     op vacuum (1 io checkio)
535     note('------temp ops------')
536     op fact   (1 ^ do(i tims))
537     op primes (2=pr:1, 1=k, 3^do(i pcheck ifelse(i=pr[k 1plus=k],))pr fir k)
538     op pcheck (=j, 0  of k do(j  pr subi  modx 0eq orx))
```

```
539     op isprime(=j, 0 2 j sqr fix dc(j i mod x 0eq orx)if'no' 'prime'is,)
540     op msg    ('this is a message')
541     op pcur   (page bot plot(sin(rarr cur 100 iotn .5)))
542     iarr n 5  iota*10      rarr x(5)  rarr y 5  var v  desc w
543     note('------go geolab------')
544     op jwh    (quiet  3=precision,  prsize 4 20  mark)
545     =op init  ('geolab2 -- 16 may 78' is rset)  mark  1=lurd, norm  init
```

```
.....geolab>parse.fortran.....May 17, 1978....15:57.....

  1     c-----p a r s e-----01 may 78-----
  2           subroutine parse(l,w,c,pp)
  3     c     parameters are  l=from-80r1  w=to  c=ptr-into-w  pp=paren-ptr
  4           integer l(1),c(8),q,t,p,c,a,ty,s,par(10),pp,zw,w(1),ww
  5           equivalence (vv,ww)
  6           p=0 ; a=1 ; zw=40
  7         1 q=tv(p,l)
  8        11 i=0 ; k=0 ; s=1 ; c=1h ; lastp=p ; goto(4,2,6,1,3,5,8)q
  9         2 i=i*10+l(p)-48 ; q=ty(p,l) ; goto(12,2,12,12,3,12,12)q
 10        12 q=q+1 ; w(q)=lookup(4hkint,4) ; q=q+1 ; w(q)=-2
 11           q=q+1 ; w(q)=i*s ; goto 11
 12         3 r=i ; i=0 ; f=1
 13        13 q=ty(p,l) ; goto(33,23,33,33,43,33,37)q
 14        23 i=i*10+l(p)-48 ; f=f*10 ; goto 13
 15        33 q=q+1 ; w(q)=lookup(4hkrea,4) ; q=q+1 ; w(q)=-2; vv=(r+i/f)*s
 16           q=q+1 ; w(q)=ww ; if(lastp+1.ne.p)goto 11
 17           q=q-2 ; c=1h.   ; k=1           ; goto 17
 18        43 q=q+1 ; w(q)=(r+i/f)*s ; goto 1
 19         4 j=(k+7)/4;c(j)=1h ;k=k+1;if(k.gt.zw)k=zw;call bits(c,k,l(p),4,.true.
 20           q=ty(p,l);goto(4,4,7,7,4,7,7)q
 21         5 j=(k+7)/4;c(j)=1h ;k=k+1;if(k.gt.zw)k=zw;call bits(c,k,l(p),4,.true.
 22           q=ty(p,l) ; goto(9,9,7,7,9,5,7)q
 23         6 if(l(p).eq.37)goto 200
 24           k=1       ; call bits(c,k,l(p),4,.true.) ; q=ty(p,l)
 25         7 q=q+1
 26        17 w(q)=lookup(c,k)
 27        37 if(c.ne.1h()goto 47 ; pp=pp+1 ; q=q+1 ; par(pp)=q ; goto 11
 28        47 if(c.ne.1h))goto 57 ; if(pp.le.0)goto 99
 29           i=par(pp)           ; pp=pp-1 ; w(i)=i-q-1        ; goto 11
 30    c      --strings--  'abcd' becomes  ' -3 4 abcd
 31        57 if(c.ne."'   ")goto 11            ; q=q+3 ; k=0
 32        67 j=(k-1)/4 ; i=q+(k+3)/4      ; w(i)="    "
 33           if(q.eq.7)goto 99 ; if(l(p).ne.39)goto 87
 34           if(l(p+1).ne.39)goto 77 ; p=p+1
 35        87 k=k+1 ; call bits(w(c),k,l(p),4,.true.)  ; q=ty(p,l) ; goto 67
 36        77 w(q-2)=-(j+3) ; w(q-1)=k       ; c=q+j     ; goto 1
 37    c      --end--
 38        99 pp=-1
 39         8 return
 40    c      --minus--(must be <blank>-<number>)--
 41         9 if(k.ne.1)goto 7 ; if(p.lt.3)goto 19
 42           m1=p ; p=p-3 ; m2=ty(p,l) ; p=m1 ; if(m2.le.2)goto 7
 43        19 i=0; s=-1; if(c(1).eq.1h-)goto(7,2,7,7,3,7,7)q
 44           goto 7
 45       200 q=ty(p,l) ; if(q.eq.7)goto 8 ; if(l(p).eq.37)goto 1
 46           goto 200
```

```
 47          end
```

.....geolab>ty.fortran.....May 17, 1978....15:57.....

```
  1     c-----t y-----type function called by parse-----17 dec 77-----
  2     c  1=let  2=num  3=noble  4=blank  5=.  6=bonding  7=fin
  3        integer function ty(p,l)
  4        integer p,l(1),t(96)
  5     c  ---------------------------------
  6     c  b ! " # $ % & '   ( ) * + , - . /   0 1 2 3 4 5 6 7   8 9 : ; < = > ?
  7        data t/
  8        4,3,3,3,3,3,3,3,  3,3,6,3,3,6,5,6,  2,2,2,2,2,2,2,2,  2,2,3,3,6,6,6,3,
  9        3,1,1,1,1,1,1,1,  1,1,1,1,1,1,1,1,  1,1,1,1,1,1,1,1,  1,1,1,3,3,3,3,1,
 10        3,1,1,1,1,1,1,1,  1,1,1,1,1,1,1,1,  1,1,1,1,1,1,1,1,  1,1,1,3,3,3,3,3/
 11     c    A B C D E F G   H I J K L M N O   P Q R S T U V W   X Y Z [ \ ] ^ _
 12     c  ` a b c d e f g   h i j k l m n o   p q r s t u v w   x y z { | } ~ bad
 13     c  ---------------------------------
 14        p=p+1       ; if(p.gt.80)goto 7
 15        k=l(p)      ; if(k.lt.32.or.k.gt.127)goto 3
 16        ty=t(k-31)  ; return
 17      3 ty=3        ; return
 18      7 ty=7        ; return
 19        end
```

.....geolab>bits.fortran.....May 17, 1978....15:57.....

```
  1        subroutine bits(w,n,b,ty,rw)
  2        integer w(1),n,b,ty,p,j,k,bas
  3        logical rw
  4        if(ty.ne.4)goto 3 ; bas=512
  5        m=(n-1)/ty+1 ; p=bas**(m*ty-n)
  6        j=w(m)/p     ; k=mod(j,bas)  ; if(rw)goto 2
  7      1 b=k                          ; return
  8      2 w(m)=w(m)+(b-k)*p            ; return
  9      3 if(rw)goto 4 ; b=w(n)        ; return
 10      4 w(n)=b                       ; return
 11        end
```

.....geolab>lookup.fortran.....May 17, 1978....15:57.....

```
  1     c-----l o o k u p-----arbitrary len names-----18 dec 77-----
  2           function lookup(name,nchar)
  3           integer name(1),nchar,nc,nw,n,hash,h
  4           integer mm,nm,aa,m,zn,ndec,iter,nloo
  5           common /gl_state$/mm(350),z1(3),zn,z2(6),
  6                             m1(6),m,m2(12),m3(5),ndec,iter,nloo,m4(13),
  7                             stacks(600),nm(900),aa(900)
  8     ccompute numwords=nw and hash constant
```

```
 9           nc=nchar ; nw=(nc+3)/4 ; n1=name(1) ; nloo=nloo+1
10           hash=(n1/256)*83       ; if(nc.lt.0)goto 3
11     csearch for match -- mem is   <bakptr> <len> <name14> <name58> etc
12           do 2 i=1,zn ; h=mod(hash+i,zn)+1 ; iter=iter+1
13              n=nm(h) ; if(n.eq.0)goto 4
14              if(mm(n+2).ne.n1)goto 2 ; if(mm(n+1).ne.nc)goto 2
15     c        -if nw=1 then we have a match otherwise check further
16              lookup=h ; if(nw.eq.1)return
17              do 1 j=2,nw ; k=n+j+1 ; if(mm(k).ne.name(j))goto 2
18         1       continue
19     c        -name matches completely, incl length so  lookup
20              return
21         2    continue
22     cno match found -- enter name into name table
23           goto 4
24     cif initialization by setmem nchar was negative
25         3 nc=-nchar ; nw=(nc+3)/4
26     clook for an empty slot in name table (nm(h)=0)
27         4 do 5 i=1,zn ; h=mod(hash+i,zn)+1 ; iter=iter+1
28              if(nm(h).le.0)goto 6
29         5    continue
30     cno empty slots available -- name table full
31           print, "name table full -- abort" ; stop
32     cempty slot found -- enter name into name table there
33         6 nm(h)=m ; mm(m)=h ; mm(m+1)=nc ; ndec=ndec+1
34           do 7 j=1,nw ; k=m+j+1
35         7    mm(k)=name(j)
36           m=k+1 ; lookup=h ; return
37           end
```

.....geolab>moreops.fortran.....May 17, 1978....15:57.....

```
 1     c-----m o r e o p s-----09 may 78-----
 2           subroutine moreops(cmd,ierr)
 3           integer cmd
 4     c     -declarations for vvm
 5           integer beg0,beg1,den0,den1,ty0,ty1,iword,iget,iput
 6           real rword
 7     c     -declarations for plotting routines
 8           real xxx(201),yyy(201),xxt(801),yyt(801)
 9           logical xlet,xint,xnum,ylet,yint,ynum,xylet,xynum,xy,yx,bool
10           integer dyr,djl,div,dfn,dlm,dty,duu,dmem,dbmem(1),dbeg,dfin
11     c --al-identifiers---------
12           integer
13             mm,thuf,cena,cenu,
14             zc,zh,zm,zn,zr,zs,zu,zy,zz,
15             c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,q,i,j,k,n,
16             bksp,nptr,iter,nloo,gbeg,lurd,
17             y,y1,y2,y3,y4,y5,y6,yy,ynam,s1,s2,
18             aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
19             umod,uyes,uno1,uno2,emsg,
20             ca,pn,pa,pz,ff,gg,hh,ii,jj,kk,
21             ty,dn,da,dz,dy,s(40),
22             nm,aa,az,
23             rmem
24           real r(40),rz,r1,r2,h,usym,unew
```

```
25          integer a,e,t,f
26          integer xup,xlp,xgm,xcl,xty,ju(200),jv(40)
27          integer g1,g2,i1,i2,j1,j2,k1,k2,n1,n2
28          integer gpool,gtune,tvse,yr,jl,iv
29      c--common--gl-mem-----------------------------------------------
30      c  mm   zc    c   ca   ff   ty   nm   aa   az cena    u rmem   end
31      c0001 0351 0361 0401 0561 0801 1001 1901 2801 3701 3911 4001 10000
32          common /gl_state$/
33            mm(120),tbuf(80),cenu(150),
34            zc,zb,zm,zn,zr,zs,zu,zy,zz,znad,
35            c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,b,i,j,k,n,
36            a,e,t,f,bksp,nptr,iter,nloop,gbeg,lurd,mrad(11),
37            ca(40),pn(40),pa(40),pz(40),
38            ff(40),gg(40),hh(40),ii(40),jj(40),kk(40),
39            ty(40),dn(40),da(40),dz(40),dv(40),
40            nm(900),aa(900),az(900),
41            cena(150),yy(6),ynam(6),y1,y2,y3,y4,y5,y6,ypad(32),
42            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
43            umod,usym,unew,uyes,uno1,uno2,upad(4),
44            emsg(10),epad(20),
45            gpad(1),tvse,gpool(8),gtune(30),yr,jl,iv,xpad(7),
46            rmem(6000)
47      c--common--tvpool-&-tvtune--------------------------------
48          common /tvpool/ ipool(8)
49          common /tvtune/ itune(30)
50      c--common/equiv--dbop-------------------------------------
51          common /gl_dbop$/ dyr,djl,div,dfn,dlm,dty,duu,dpad(3),dmem(100)
52          equivalence (dyr,dbmem)
53      c--equivalence--------------------------------------------
54          equivalence (xup,cenu(  5)),(xlp,cenu(  3)),(r ,s ),(s1,r1),
55               (dy,s),(xcl,cenu(149)),(xty,cenu(150)),(rz,sz),(s2,r2),
56                    (xgm,cenu(134))
57      c--end-of-gl-declarations---------------------------------
58          equivalence (iword,rword)
59      c--branch to particular ce--------------------------
60          ierr=0
61          goto(998, 1, 2, 3, 4, 5, 6, 7, 8, 9,10,
62                   11,12,13,14,15,16,17,18,19,20,
63                   21)cmd+1
64        999 ierr=1 ; return
65        998 print, "moreops -- 09 may 78" ; return
66      c--vvm----------------------------------------
67      ccalc min number of elements to move
68          1 beg0=da(z) ; beg1=da(z-1)
69          len0=dz(z)-beg0+1 ; len1=dz(z-1)-beg1+1 ; n=min(len0,len1)
70      cmove either whole or quarter words, den0,den1=1,4
71          ty0=ty(z) ; den0=yy(ty0) ; ty1=ty(z-1) ; den1=yy(ty1)
72      cadjust beg0,beg1 for counting thru memory
73          beg0=beg0-den0 ; beg1=beg1-den1
74      cdo the move
75          hh(d)=n ; if(n.lt.1)return
76          do 101 i=1,n ; iget=beg0+i ; iput=beg1+i
77             if(den0.eq.1)iword=mm(iget)
78             if(ty0 .eq.5.and.ty1.ne.5)iword=rword
79             if(den0.ne.1)call bits(mm,iget,iword,den0,.false.)
80             if(den1.ne.1)call bits(mm,iput,iword,den1,.true. )
81             if(ty0 .ne.5.and.ty1.eq.5)rword=iword
82             if(den1.eq.1)mm(iput)=iword
83        101    continue
84          return
```

```
c--vsm-----------------------------------------
     2 beg1=da(z-1) ; n=dz(z-1)-beg1+1
       ty0=ty(z)      ; ty1=ty(z-1) ; den1=vv(ty1)
       beg1=beg1-den1 ; if(n.lt.1)return
       iword=s(z)
       if(ty0.eq.2.and.ty1.ne.5)iword=rword
       if(ty0.ne.2.and.ty1.eq.5)rword=iword
       do 2002 i=1,n ; iput=beg1+i
          if(den1.ne.1)call bits(mm,iput,iword,den1,.true.)
          if(den1.eq.1)mm(iput)=iword
  2002    continue
       return
c--nmstr,mmstr---------------------------------
     3 continue
     4 iword=s(z) ; if(ty(z).eq.2)iword=rword
       n=iword    ; if(ty(z).gt.3)return
       if(cmd.eq.4)goto 41
       if(n.lt.1.or.n.gt.zn)return ; n=nm(n)
    41 ty(z)=y6 ; dn(z)=xqm ; da(z)=(n+2)*4 ; dz(z)=da(z)+mm(n+1)-1
       dy(z)=0 ; k=mm(n) ; if(k.lt.1.or.k.gt.zn)goto 49
       if(nm(k).eq.n)return
    49 dz(z)=da(z) ; return
c--hilo----------------------------------
     5 if(ty(z).lt.y4)goto 999
       beg0=da(z) ; len0=dz(z)-beg0+1
       x0=usym ; y0=usym ; j=beg0-1 ; bool=.false.
       xlet=ty(z).eq.y6 ; xnum=ty(z).ne.y6 ; xint=ty(z).ne.y5
       do 53 i=1,len0 ; j=j+1
          if(xlet)call bits(mm,j-3,iword,4,.false.)
          if(xnum)iword=mm(j)  ; if(xint)rword=iword
          if(umod.eq.0)goto 51 ; if(rword.eq.usym)goto 53
    51    if(bool)goto 52 ; x0=rword ; y0=x0 ; bool=.true. ; goto 53
    52    if(rword.lt.x0)x0=rword ; if(rword.gt.y0)y0=rword
    53    continue
       if(x0.lt.y0)goto 59 ; x0=x0-.5 ; y0=x0+1
    59 r(z+1)=x0 ; r(z+2)=y0 ; ty(z+1)=y2 ; ty(z+2)=y2 ; z=z+2 ; return
c--page----------------------------------
     6 call tvnext ; return
c--tvltr---------------------------------
     7 do 1007 i=1,8
  1007 ipool(i)=gpool(i)
       do 2007 i=1,30
  2007 itune(i)=gtune(i)
       if(a.ne.f)itune(15)="page" ; a=f
cplot a character string -- form:  x y string tvltr
       if(ty(z-1).gt.y3.or.ty(z-2).gt.y3)goto 999
       iword=s(z-2) ; if(ty(z-2).ne.y2)rword=iword ; x0=rword
       iword=s(z-1) ; if(ty(z-1).ne.y2)rword=iword ; y0=rword
       if(ty(z).eq.y6)goto 71 ; if(ty(z).gt.y3)goto 999
cplot a scalar as a one letter string
       iword=s(z) ; if(ty(z).eq.y2)iword=rword ; ju(1)="    "
       if(iword.lt.0.or.iword.gt.128)return
       call bits(ju,1,iword,4,.true.) ; n=1 ; goto 72
    71 call justr(ju,mm,da(z),dz(z),1) ; n=dz(z)-da(z)+1
    72 call tvltr(x0,y0,ju,n)
       if(tvse.ne.1)call tvsend ; z=z-3 ; return
c--fare----------------------------------------
     8 call tvfare(i,r(z+1),r(z+2),j)
       ty(z+1)=y2 ; ty(z+2)=y2 ; z=z+2 ; return
c--tvplot--------------------------------------
```

```
145     cmove gpool and gtune into tvpool and tvtune
146           do 1009 i=1,8
147      1009 ipool(i)=gpool(i)
148           do 2009 i=1,30
149      2009 itune(i)=gtune(i)
150     cget plot-type switch from stack -- 1=join  2=point  3=segment
151           iword=s(z) ; if(ty(z).eq.y2)iword=rword ; s1=iword
152     cget facts about y array -- must exist
153           if(ty(z-1).lt.y4)goto 999 ; yint=ty(z-1).ne.y5
154           beg0=da(z-1) ; len0=dz(z-1)-beg0+1 ; iy=beg0-1
155           ylet=ty(z-1).eq.y6 ; if(ty(z-2).ge.y4)goto 97
156     cx is a scalar use as subscript base -- will plot y vs sub
157           iword=s(z-2) ; if(ty(z-2).ne.y2)rword=iword ; sub=rword-1
158           ticbase=sub+1; if(s1.ge.4)sub=0
159           len1=9999999 ; xy=.false. ; xlet=xy ; xint=xy
160           ix=0         ; goto 98
161     cx is an array -- will plot y vs x
162        97 beg1=da(z-2) ; len1=dz(z-2)-beg1+1 ; ix=beg1-1
163           xy=.true.    ; xint=ty(z-2).ne.y5 ; xlet=ty(z-2).eq.y6
164     cget ready to loop thru arrays
165        98 ilim=200     ; n=min(len0,len1) ; iylim=iy+n
166           mode=0 ; nplots=0 ; xnum=.not.xlet ; ynum=.not.ylet
167           xylet=xy.and.xlet ; xynum=xy.and.xnum
168     cinitialize i and loop
169        91 i=0 ; j=0 ; x0l=-9999.
170     cmain loop -- once around per element of y array
171        92 npts=i ; i=i+1 ; ix=ix+1 ; iy=iy+1 ; sub=sub+1.0 ; rword=sub
172           if(xylet)call bits(mm,ix-3,iword,4,.false.)
173           if(xynum)iword=mm(ix) ; if(xint)rword=iword ; x0=rword
174           if( ylet)call bits(mm,iy-3,iword,4,.false.)
175           if( ynum)iword=mm(iy) ; if(yint)rword=iword ; y0=rword
176     cbranch depending on reg,mis-data,end-buf,end-plot
177           if(umod.eq.0)goto 93  ; if(x0.eq.usym.or.y0.eq.usym)goto 94
178        93 if(i.le.ilim.and.iy.le.iylim)goto 96 ; mode=1
179     cend of buffer or end of plot of emiss data so plot upto i-1
180        94 if(npts.lt.1)goto 95 ; nplots=nplots+1
181           if(s1.le.1)call tvplot(xxx,yyy,npts,4hjoin)
182           if(s1.eq.2)call tvplot(xxx,yyy,npts,5hpoint)
183           if(s1.eq.3)call tvplot(xxx,yyy,npts,7hsegment)
184           if(s1.ge.4)xxx(2)=xxx(npts)
185           if(s1.ge.4)call tvplot(xxx,yyy,   2,4hjoin)
186           if(s1.ge.4)call tvplot(xxt,yyt,   j,7hsegment)
187     cif missing data skip x0,y0
188        95 if(mode.eq.0)goto 91
189     cif end of plot return else is end of buff so repeat xxx,yyy(ilim)
190           if(iy.gt.iylim)goto 99 ; xxx(1)=xxx(ilim)
191           mode=0 ; i=2 ; j=0     ; yyy(1)=yyy(ilim)
192     cregular trip around the loop -- put x0,y0 into buffers xxx,yyy
193        96 xxx(i)=x0      ; yyy(i)=y0  ; if(s1.lt.4)goto 92
194           yyy(i)=ticbase             ; if(s1.eq.4.and.y0.eq.ticbase)goto 92
195           j=j+1          ; xxt(j)=x0  ; yyt(j)=ticbase
196           j=j+1          ; xxt(j)=x0  ; yyt(j)=y0
197     c     j5=j-1 ; write(6,342)j5,j,xxt(j5),xxt(j),yyt(j5),yyt(j)
198     c 342 format(2i5,2f10.2/10x,2f10.2/)
199           if(s1.ne.5)goto 92         ; if(x0l.eq.-9999.)goto 196
200           ymax=amax1(y0,y0l)         ; yyt(j  )=amax1(ymax,ticbase)
201           ymin=amin1(y0,y0l)         ; yyt(j-1)=amin1(ymin,ticbase)
202           j=j+1          ; xxt(j)=x0l ; yyt(j)=y0l
203           j=j+1          ; xxt(j)=x0  ; yyt(j)=y0l
204     c     j7=j-3 ; write(6,343)j7,j,(xxt(j6),j6=j7,j),(yyt(j6),j6=j7,j)
```

```
205     c 343 format(2i5,4f10.2/10x,4f10.2/)
206       196 x01=x0          ; y01=y0      ; goto 99
207     cend of plot -- call tvsend
208        99 if(nplots.gt.0.and.tvse.ne.1)call tvsend ; z=z-1 ; jsav=j ; return
209     c--tvsend-----------------------------------------
210        10 call tvsend ; return
211     c--dbops------------------------------------------
212        11 if(ty(z-2).ne.y4.and.ty(z-2).ne.y5)goto 999
213           if(ty(z-1).ne.y1.or. ty(z  ).ne.y1)goto 999
214           j=da(z-2) ; n=dz(z-2)-j+1 ; k=s(z) ; z=z-1
215           call dbops(mm(j),n,a,s(z),k) ; a=f ; return
216     c--dat-------------------------------------------------
217        12 y=ty(z) ; if(y.ne.y4.and.y.ne.y5)goto 999
218           k=da(z) ; n=dz(z)-k+1
219           dbeg=((yr-dyr)*365 + (yr-1)/4-(dyr-1)/4 + jl-djl)*1440/iv +1
220           dfin=dbeg+n-1
221     cprepare for possible type conversion between data base and ql array
222           xy=.false. ; if(dty.eq.1.and.y.eq.y5)xy=.true.
223           yx=.false. ; if(dty.eq.2.and.y.eq.y4)yx=.true.
224     cdo necessary type conversion for missing data symbol
225           rword=usym ; if(y.eq.y4)iword=rword ; msym=iword
226     cbranch on read or write
227           if(a.ne.f)goto 212
228     cread from data base into internal ql memory
229           do 112 i=1,n      ; j=dbeg+i-1 ; g=k+i-1
230              iword=msym     ; if(j.lt.1.or.j.gt.dfn)goto 112
231              iget =dmem(j) ; if(iget.eq.duu)goto 112
232              iword=iget     ; if(xy)rword=iword ; if(yx)iword=rword
233        112   mm(g)=iword
234           return
235     cwrite from internam ql memory into data base
236        212 do 612 i=1,n ; j=dbeg+i-1 ; g=k+i-1
237     c        -if j is outside limits of db dont write (skip it)
238              if(j.lt.1.or.j.gt.dlm)goto 612
239     c        -if j is outside devel portion of db extend it
240              if(j.le.dfn)goto 512 ; if(j.eq.dfn)goto 412
241     c        -the moving of the frontier may leave a gap; fill it
242              ibeg=dfn+1 ; ifin=j-1
243              do 312 i2=ibeg,ifin
244        312      dmem(i2)=duu
245        412   dfn=j
246     c        -write one word into db, watch for missing data
247        512   iword=mm(g) ; if(xy)iword=rword ; if(yx)rword=iword
248              dmem(j)=iword ; if(mm(g).eq.msym)dmem(j)=duu
249        612   continue
250           a=f ; return
251     c--rewind----------------------------------------
252        13 if(ty(z).ne.y1)goto 999
253           k=s(z) ; z=z-1 ; rewind k ; return
254     c--endfile---------------------------------------
255        14 if(ty(z).ne.y1)goto 999
256           k=s(z) ; z=z-1 ; endfile k ; return
257     c--stanops---------------------------------------
258        15 if(ty(z).ne.y1)goto 999 ; s1=s(z) ; z=z-1
259           call stanops(s1,ierr)   ; return
260     c--mor2ops---------------------------------------
261        16 if(ty(z).ne.y1)goto 999 ; s1=s(z) ; z=z-1
262           do 1016 i=1,8
263      1016 ipool(i)=gpool(i)
264           do 2016 i=1,30
```

```
265      2016 itune(i)=gtune(i)
266           call mor2ops(s1,ierr)   ; return
267      c--mor3ops-----------------------------------------
268         17 if(ty(z).ne.y1)goto 999 ; s1=s(z) ; z=z-1
269           call mor3ops(s1,ierr)   ; return
270      c--mor4ops-----------------------------------------
271         18 if(ty(z).ne.y1)goto 999 ; s1=s(z) ; z=z-1
272           call mor4ops(s1,ierr)   ; return
273      c--mor5ops-----------------------------------------
274         19 if(ty(z).ne.y1)goto 999 ; s1=s(z) ; z=z-1
275           call mor5ops(s1,ierr)   ; return
276      c--mmm---------------------------------
277         20 if(ty(z).ge.y4)goto 999    ; iword=s(z)
278           if(ty(z).eq.y2)iword=rword ; z=z-1
279           if(ty(z).le.y3)goto 999
280           beg0=da(z) ; n=dz(z)-beg0+1
281           if(ty(z).ne.y6)goto 1020 ; beg0=beg0/4 ; n=(n+3)/4
282       1020 if(a    .ne. f)goto 1120
283           do 2020 i=1,n ; j=beg0+i-1 ; k=iword+i-1
284       2020 mm(j)=dbmem(k)
285           return
286       1120 do 2120 i=1,n ; j=beg0+i-1 ; k=iword+i-1
287       2120 dbmem(k)=mm(j)
288           a=f ; return
289      c--ranlook-----------------------------------
290         21 k=(dfn-1)/(1440/iv) ; itype="intg" ; if(dty.eq.2)itype="real"
291           iword=duu ; if(dty.eq.2)iword=rword ; iduu=iword
292           if(dyr.gt.0.and.dyr.lt.100)goto 5021
293           print, "invalid bottle" ; return
294       5021 call dait(dyr,djl,0,iy1,ij1,im1,id1,in1)
295           call dait(dyr,djl,k,iy2,ij2,im2,id2,in2)
296           call dait( yr, jl,0,iy3,ij3,im3,id3,in3)
297           write(6,1021)in1,id1,iy1,ij1
298           write(6,2021)in2,id2,iy2,ij2
299           write(6,4021)itype,div,dfn,iduu
300           write(6,3021)in3,id3,iy3,ij3,iv
301       1021 format(" begin  time:  ",a3,2i3,i4)
302       2021 format(" end    time:  ",a3,2i3,i4)
303       3021 format(" lookat time:  ",a3,2i3,i4,i8,"=iv")
304       4021 format(" data   form:  ",a4,    9x,i8,"=iv",i7,"=nsam",i10,"=usym")
305           end
```

```
.......geolab>mor2ops.fortran.........May 17, 1978....15:57.........
   1     c-----m o r 2 o p s-----01 apr 78-----
   2           subroutine mor2ops(cmd,ierr)
   3           integer cmd
   4     c     -declarations for vvm
   5           integer beg0,fin0,beg1,den0,den1,ty0,ty1,iword,iget,iput
   6           real rword
   7     c     -declarations for plotting routines
   8           real xxx(501),yyy(501)
   9           logical xlet,xint,xnum,ylet,yint,ynum,xylet,xynum,xy,yx,bool
  10           integer dyr,djl,div,dfn,dlm,dty,duu,dmem,dbmem(1),dbeg,dfin
  11     c --gl-identifiers---------
  12           integer
```

```
13            mm,tbuf,cena,cenu,
14            zc,zh,zm,zn,zr,zs,zu,zy,zz,
15            c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,q,i,j,k,n,
16            bksp,nptr,iter,nloc,cbeg,lurd,
17            y,y1,y2,y3,y4,y5,y6,yy,ynam,s1,s2,
18            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
19            umod,uyes,uno1,uno2,emsg,
20            ca,pn,pa,pz,ff,gg,hh,ii,jj,kk,
21            ty,dn,da,dz,dy,s(40),
22            nm,aa,az,
23            rmem
24          real r(40),rz,r1,r2,t,usym,unew
25          integer a,e,t,f
26          integer xup,xlp,xgm,xcl,xty,ju(200),jv(40)
27          integer g1,g2,i1,i2,j1,j2,k1,k2,n1,n2
28          integer gpool,gtune,tvse,yr,jl,iv
29    c--common--gl-mem-----------------------------------------------
30    c  mm   zc    c   ca   ff   ty   nm   aa   az cena    u rmem   end
31    c0001 0351 0361 0401 0561 0801 1001 1901 2801 3701 3911 4001 10000
32          common /gl_state$/
33            mm(120),tbuf(80),cenu(150),
34            zc,zh,zm,zn,zr,zs,zu,zy,zz,zpad,
35            c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,h,i,j,k,n,
36            a,e,t,f,bksp,nptr,iter,nloc,cbeg,lurd,mpad(11),
37            ca(40),pn(40),pa(40),pz(40),
38            ff(40),gg(40),hh(40),ii(40),jj(40),kk(40),
39            ty(40),dn(40),da(40),dz(40),dy(40),
40            nm(900),aa(900),az(900),
41            cena(150),yy(6),ynam(6),y1,y2,y3,y4,y5,y6,ypad(32),
42            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
43            umod,usym,unew,uyes,uno1,uno2,upad(4),
44            emsg(10),epad(20),
45            gpad(1),tvse,gpool(8),gtune(30),yr,jl,iv,xpad(7),
46            rmem(6000)
47    c--common--tvpool-&-tvtune--------------------------------
48          common /tvpool/ ipool(8)
49          common /tvtune/ itune(30)
50    c--equivalence--------------------------------------------
51          equivalence (xup,cenu(  5)),(xlp,cenu(  3)),(r ,s ),(s1,r1),
52               (dy,s),(xcl,cenu(149)),(xty,cenu(150)),(rz,sz),(s2,r2),
53                    (xgm,cenu(134))
54    c--end-of-gl-declarations---------------------------------
55          equivalence (iword,rword)
56    c--branch to particular ce--------------------------
57          ierr=0
58          goto(1,2,3,4,5)cmd
59      999 ierr=1 ; return
60    c-------------------------------------------------------
61    c---m o r 2 o p s    m o d u l e s----------------------
62    c-------------------------------------------------------
63    c--module number 1----------------------------------------
64        1 if(ty(z).ne.y1)goto 999 ; s1=s(z) ; z=z-1
65          call frame(yr,jl,iv,s1,ju)    ; return
66    c--module number 2----------------------------------------
67        2 print, "mor2ops #", cmd ; return
68    c--module number 3----------------------------------------
69        3 print, "mor2ops #", cmd ; return
70    c--module number 4----------------------------------------
71        4 print, "mor2ops #", cmd ; return
72    c--module number 5----------------------------------------
```

```
 73         5 print, "mor2ops #", cmd ; return
 74           end
```

```
.....geolab>mor3ops.fortran.....May 17, 1978....15:57.....

  1     c-----m o r 3 o p s-----12 may 78-----
  2           subroutine mor3ops(cmd,ierr)
  3           integer cmd
  4     c     -declarations for vvm
  5           integer beg0,fin0,beg1,den0,den1,ty0,ty1,iget,iput
  6           integer iword,iword1,iword2
  7           real    rword,rword1,rword2
  8           logical lword,lword1,lword2
  9     c     -declarations for plotting routines
 10           real xxx(501),yyy(501)
 11           logical xlet,xint,xnum,ylet,yint,ynum,xylet,xynum,xy,yx,bool
 12           integer dyr,djl,div,dfn,dlm,dty,duu,dmem,dbmem(1),dbeg,dfin
 13     c --cl-identifiers---------
 14           integer
 15             mm,tbuf,cena,cenu,
 16             zc,zh,zm,zn,zr,zs,zu,zy,zz,
 17             c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,g,i,j,k,n,
 18             bksp,nptr,iter,nloo,gbeg,lurd,
 19             y,y1,y2,y3,y4,y5,y6,yy,ynam,s1,s2,
 20             aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
 21             umod,uyes,uno1,uno2,emsg,
 22             ca,pn,pa,pz,ff,gg,hh,ii,jj,kk,
 23             ty,dn,da,dz,dy,s(40),
 24             nm,aa,az,
 25             rmem
 26           real r(40),rz,r1,r2,h,usym,unew
 27           integer a,e,t,f
 28           integer xup,xlp,xgm,xcl,xty,ju(200),jv(40)
 29           integer g1,g2,i1,i2,j1,j2,k1,k2,n1,n2
 30           integer gpool,gtune,tvse,vr,jl,iv
 31     c--common--gl-mem-----------------------------------------------
 32     c  mm   zc    c   ca   ff   ty   nm   aa   az cena    u rmem   end
 33     c0001 0351 0361 0401 0561 0801 1001 1901 2801 3701 3911 4001 10000
 34           common /gl_state$/
 35             mm(120),tbuf(80),cenu(150),
 36             zc,zh,zm,zn,zr,zs,zu,zy,zz,zpad,
 37             c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,h,i,j,k,n,
 38             a,e,t,f,bksp,nptr,iter,nloo,gbeg,lurd,mpad(11),
 39             ca(40),pn(40),pa(40),pz(40),
 40             ff(40),gg(40),hh(40),ii(40),jj(40),kk(40),
 41             ty(40),dn(40),da(40),dz(40),dy(40),
 42             nm(900),aa(900),az(900),
 43             cena(150),yy(6),ynam(6),y1,y2,y3,y4,y5,y6,ypad(32),
 44             aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
 45             umod,usym,unew,uyes,uno1,uno2,upad(4),
 46             emsg(10),epad(20),
 47             gpad(1),tvse,gpool(8),gtune(30),vr,jl,iv,xpad(7),
 48             rmem(6000)
 49     c--common--tvpool-&-tvtune--------------------------------
 50           common /tvpool/ ipool(8)
 51           common /tvtune/ itune(30)
```

```
52      c--equivalence-------------------------------------------
53            equivalence (xup,cenu(  5)),(xlp,cenu(  3)),(r ,s ),(s1,r1),
54                 (dy,s),(xcl,cenu(149)),(xty,cenu(150)),(rz,sz),(s2,r2),
55                      (xgm,cenu(134))
56      c--end-of-gl-declarations--------------------------------
57            equivalence (iword,rword,lword),(iword1,lword1),(iword2,lword2)
58      c--branch to particular ce-------------------------
59            ierr=0
60            goto(1,2,3,4,5,6)cmd
61          999 ierr=1 ; return
62      c-------------------------------------------------------
63      c---m o r 3 o p s    m o d u l e s-------------------------
64      c-------------------------------------------------------
65      c--module number 1---------------------------------------
66            1 call dait(s(z-2),s(z-1),s(z),k1,k2,k3,s(z+1),s(z+2))
67              s(z-2)=k1 ; s(z-1)=k2 ; s(z)=k3 ; z=z+2 ; return
68      c--ltsq etc----------------------------------------------
69            2 sxx=0 ; sxy=0 ; sx=0 ; sy=0 ; x0sum=0 ; y0sum=0 ; cow=0
70      cget cmd from stack -- 1=sum 2=ltsq
71            iword=s(z) ; if(ty(z).eq.y2)iword=rword ; s1=iword
72      cget facts about y array -- must exist
73            if(ty(z-1).lt.y4)goto 999 ; yint=ty(z-1).ne.y5
74            beg0=da(z-1) ; len0=dz(z-1)-beg0+1 ; iy=beg0-1
75            ylet=ty(z-1).eq.y6 ; if(ty(z-2).ge.y4)goto 97
76      cx is a scalar use as subscript base -- will plot y vs sub
77            iword=s(z-2) ; if(ty(z-2).ne.y2)rword=iword ; sub=rword-1
78            len1=9999999 ; xy=.false. : xlet=xy ; xint=xy ; sub=0
79            ix=0         ; goto 98
80      cx is an array -- will plot y vs x
81           97 beg1=da(z-2) ; len1=dz(z-2)-beg1+1 ; ix=beg1-1
82              xy=.true.    ; xint=ty(z-2).ne.y5  ; xlet=ty(z-2).eq.y6
83      cget ready to loop thru arrays
84           98 ilim=200     ; n=min(len0,len1) ; iylim=iy+n
85              mode=0 ; nplots=0 ; xnum=.not.xlet ; ynum=.not.ylet
86              xylet=xy.and.xlet ; xynum=xy.and.xnum
87      cinitialize i and loop
88           91 i=0 ; j=0
89      cmain loop -- once around per element of y array
90            do 92 i=1,n ; ix=ix+1 ; iy=iy+1 ; sub=sub+1.0 ; rword=sub
91            if(xylet)call bits(mm,ix-3,iword,4,.false.)
92            if(xynum)iword=mm(ix) ; if(xint)rword=iword ; x0=rword
93            if( ylet)call bits(mm,iy-3,iword,4,.false.)
94            if( ynum)iword=mm(iy) ; if(yint)rword=iword ; y0=rword
95      cbranch depending on reg,mis-data,end-buf,end-plot
96            if(umod.eq.0)goto 93  ; if(x0.eq.usym.or.y0.eq.usym)goto 92
97           93 cow=cow+1 ; goto(801,802,803)s1
98          801 x0sum=x0sum+x0 ; y0sum=y0sum+y0 ; goto 92
99          802 sxx=sxx+x0*x0 ; sx=sx+x0
100             sxy=sxy+x0*y0 ; sy=sy+y0 ; goto 92
101         803 print 1803, cow,x0,y0     ; goto 92
102        1803 format(f6.0,2x,f10.3,"=x0  ",f10.3,"=y0")
103          92 continue
104     cend of loop -- now do epilogue
105           goto(901,902,903)s1
106         901 r(z  )=x0sum ; ty(z  )=y2
107             r(z+1)=y0sum ; ty(z+1)=y2 ; z=z+1 ; return
108         902 temp=cow*sxx-sx*sx
109             r(z  )=(cow*sxy-sx*sy)/temp ; ty(z  )=y2
110             r(z+1)=(sy-r(z  )*sx )/cow  ; ty(z+1)=y2 ; z=z+1 ; return
111         903 print, "end of task three"                        ; return
```

```
c--bits-------------------------------------------------
    3 k=a ; a=f ; if(k.ne.f)goto 1003
      call bits36(s(z-1),s(z  ),s1  ,36,.false.) ; s(z)=s1 ; return
 1003 call bits36(s(z-2),s(z-1),s(z),36,.true. ) ; z=z-2  ; return
c--land -- logical and ----------------------------------------
    4 iword1=s(z-1) ; iword2=s(z)
      lword=lword1.and.lword2 ; z=z-1 ; s(z)=iword ; return
c--lor  -- logical or  ----------------------------------------
    5 iword1=s(z-1) ; iword2=s(z)
      lword=lword1.or. lword2 ; z=z-1 ; s(z)=iword ; return
c--lnot -- logical not ----------------------------------------
    6 iword1=s(z) ; lword=.not.lword1 ; s(z)=iword ; return
      end
```

.....genlab>mor4ops.fortran.....May 17, 1978....15:57.....

```
c-----m o r 4 o p s-----01 apr 78-----
      subroutine mor4ops(cmd,ierr)
      integer cmd
c     -declarations for vvm
      integer beg0,fin0,beg1,den0,den1,ty0,ty1,iword,icet,inut
      real rword
c     -declarations for plotting routines
      real xxx(501),yyy(501)
      logical xlet,xint,xnum,ylet,yint,ynum,xvlet,xynum,xy,yx,bool
      integer dyr,djl,div,dfn,dlm,dty,duu,dmem,dbmem(1),dbeg,dfin
c --gl-identifiers---------
      integer
        mm,tbuf,cena,cenu,
        zc,zb,zm,zn,zr,zs,zu,zy,zz,
        c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,q,i,j,k,n,
        bksp,nptr,iter,nloc,gbeg,lurd,
        y,y1,y2,y3,y4,y5,y6,yy,ynam,s1,s2,
        aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
        umod,uyes,uno1,uno2,emsg,
        ca,pn,pa,pz,ff,gg,hh,ii,jj,kk,
        ty,dn,da,dz,dy,s(40),
        nm,aa,az,
        rmem
      real r(40),rz,r1,r2,h,usym,unew
      integer a,e,t,f
      integer xup,xlp,xom,xcl,xty,ju(200),jv(40)
      integer g1,g2,i1,i2,j1,j2,k1,k2,n1,n2
      integer gpool,gtune,tvse,vr,jl,iv
c--common--gl-mem---------------------------------------------
c  mm   zc    c   ca   ff   ty   nr   aa   az cena    u rmem   end
c0001 0351 0361 0401 0561 0801 1001 1901 2801 3701 3911 4001 10000
      common /gl_state$/
        mm(120),tbuf(80),cenu(150),
        zc,zb,zm,zn,zr,zs,zu,zy,zz,zpad,
        c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,h,i,j,k,n,
        a,e,t,f,bksp,nptr,iter,nloc,gbeg,lurd,mpad(11),
        ca(40),pn(40),pa(40),pz(40),
        ff(40),gg(40),hh(40),ii(40),jj(40),kk(40),
        ty(40),dn(40),da(40),dz(40),dv(40),
        nm(900),aa(900),az(900),
```

```
41            cena(150),yy(6),ynam(6),y1,y2,y3,y4,y5,y6,ypad(32),
42            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
43            umod,usym,unew,uyes,uno1,uno2,upad(4),
44            emsg(10),epad(20),
45            qpad(1),tvse,qpool(8),gtune(30),yr,jl,iv,xpad(7),
46            rmem(6000)
47      c--common--tvpool-8-tvtune---------------------------------
48            common /tvpool/ ipool(8)
49            common /tvtune/ itune(30)
50      c--equivalence---------------------------------------------
51            equivalence (xup,cenu(  5)),(xlp,cenu(  3)),(r ,s ),(s1,r1),
52                 (dy,s),(xcl,cenu(149)),(xtv,cenu(150)),(rz,sz),(s2,r2),
53                      (xgm,cenu(134))
54      c--end-of-gl-declarations----------------------------------
55            equivalence (iword,rword)
56      c--branch to particular ce---------------------------
57            ierr=0
58            goto(1,2,3,4,5)cmd
59          999 ierr=1 ; return
60      c-----------------------------------------------------------
61      c---m o r 4 o p s    m o d u l e s-----------------------
62      c-----------------------------------------------------------
63      c--module number 1-----------------------------------------
64            1 print, "---mor4ops module number 1---"
65              if(ty(z).lt.y4)goto 999 ; beg0=da(z) ; fin0=dz(z) ; k=usym
66              print 11,beg0,fin0,umod,k,ty(z)
67             11 format("   beg0  fin0  umod  usym tv(z)"/1x,5i6)
68              print 12, (mm(i),i=beg0,fin0)
69             12 format(1x,10i4)
70              print, "--ok--" ; return
71      c--module number 2-----------------------------------------
72            2 print, "mor4ops #", cmd ; return
73      c--module number 3-----------------------------------------
74            3 print, "mor4ops #", cmd ; return
75      c--module number 4-----------------------------------------
76            4 print, "mor4ops #", cmd ; return
77      c--module number 5-----------------------------------------
78            5 print, "mor4ops #", cmd ; return
79              end
```

.....geolab>mor5ops.fortran.....May 17, 1978....15:57.....

```
 1      c-----m o r 5 o p s-----01 apr 78-----
 2            subroutine mor5ops(cmd,ierr)
 3            integer cmd
 4      c     -declarations for vvm
 5            integer beg0,fin0,beg1,den0,den1,ty0,ty1,iword,iget,iput
 6            real rword
 7      c     -declarations for plotting routines
 8            real xxx(501),yyy(501)
 9            logical xlet,xint,xnum,vlet,vint,vnum,xvlet,xvnum,xy,vx,bool
10            integer dyr,djl,div,dfn,dlm,dty,duu,dmem,dbmem(1),dbeg,dfin
11      c --gl-identifiers---------
12            integer
13              mm,tbuf,cena,cenu,
14              zc,zh,zm,zn,zr,zs,zu,zy,zz,
```

```
15            c,d,n,l,z,sz,m,lur,u,ntr,adr,ce,dlv,ncyc,g,i,j,k,n,
16            bksp,nptr,iter,nloo,obeg,lurd,
17            y,y1,y2,y3,y4,y5,y6,yy,ynam,s1,s2,
18            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
19            umod,uyes,uno1,uno2,emsg,
20            ca,pn,pa,pz,ff,gg,hh,ii,jj,kk,
21            ty,dn,da,dz,dy,s(40),
22            nm,aa,az,
23            rmem
24          real r(40),rz,r1,r2,',usym,unew
25          integer a,e,t,f
26          integer xup,xlp,xgm,xcl,xty,ju(200),jv(40)
27          integer g1,g2,i1,i2,j1,j2,k1,k2,n1,n2
28          integer gpool,gtune,tvse,vr,jl,iv
29      c--common--gl-mem-----------------------------------------------
30      c  mm   zc    c   ca   ff   ty   nm   aa   az cena    u rmem   end
31      c0001 0351 0361 0401 0561 0801 1001 1901 2801 3701 3911 4001 10000
32          common /gl_state$/
33            mm(120),tbuf(80),cenu(150),
34            zc,zh,zm,zn,zr,zs,zu,zy,zz,zpad,
35            c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,h,i,j,k,n,
36            a,e,t,f,bksp,nptr,iter,nloo,obeg,lurd,mpad(11),
37            ca(40),pn(40),pa(40),pz(40),
38            ff(40),gg(40),hh(40),ii(40),jj(40),kk(40),
39            ty(40),dn(40),da(40),dz(40),dy(40),
40            nm(900),aa(900),az(900),
41            cena(150),yy(6),ynam(6),y1,y2,y3,y4,y5,y6,ypad(32),
42            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
43            umod,usym,unew,uyes,uno1,uno2,upad(4),
44            emsg(10),epad(20),
45            gpad(1),tvse,gpool(8),gtune(30),vr,jl,iv,xpad(7),
46            rmem(6000)
47      c--common--tvpool-&-tvtune--------------------------------
48          common /tvpool/ ipool(8)
49          common /tvtune/ itune(30)
50      c--equivalence--------------------------------------------
51          equivalence (xup,cenu(  5)),(xlp,cenu(  3)),(r ,s ),(s1,r1),
52               (dy,s),(xcl,cenu(149)),(xty,cenu(150)),(rz,sz),(s2,r2),
53                    (xgm,cenu(134))
54      c--end-of-gl-declarations---------------------------------
55          equivalence (iword,rword)
56      c--branch to particular ce--------------------------
57          ierr=0
58          goto(1,2,3,4,5)cmd
59        999 ierr=1 ; return
60      c-------------------------------------------------------------
61      c---m o r 5 o p s    m o d u l e s------------------------
62      c-------------------------------------------------------------
63      c--module number 1----------------------------------------
64          1 print, "---mor5ops module number 1---"
65            if(ty(z).lt.y4)goto 999 ; beg0=da(z) ; fin0=dz(z) ; k=usym
66            print 11,beg0,fin0,umod,k,ty(z)
67         11 format("   beg0  fin0  umod  usym ty(z)"/1x,5i6)
68            print 12, (mm(i),i=beg0,fin0)
69         12 format(1x,10i4)
70            print, "--ok--" ; return
71      c--module number 2----------------------------------------
72          2 print, "mor5ops #", cmd ; return
73      c--module number 3----------------------------------------
74          3 print, "mor5ops #", cmd ; return
```

```
75      c--module number 4-----------------------------------------
76          4 print, "mor5ops #", cmd ; return
77      c--module number 5-----------------------------------------
78          5 print, "mor5ops #", cmd ; return
79            end
```

.....geolab>frame.fortran.....May 17, 1978....15:57.....

```
 1      c-----f r a m e------20 apr 78-------------
 2            subroutine frame(yr,jl,iv,cmd,ju)
 3            integer yr,jl,iv,cmd,ju(200)
 4            integer str(10),beg,fin,len
 5            integer y2,j2,m2,d2,mnm
 6            real w,e,n,s,ww,ee,nn,ss,w2,e2,n2,s2,dx,dy,a(200),b(200)
 7            common /tvpool/ tv(8)
 8            common /tvtune/ itune(30)
 9            data  ndy, mdy, n10, m10,kmon,  kr,mmon, m6m, nyr, myr
10               /  41, 101, 151, 201, 301, 601,1001,2001,9999,9999/
11      c-plot box
12            dx=tv(2)-tv(1) ; dy=tv(4)-tv(3) ; itune(15)=4hdata
13            w=tv(1) ; ww=w+dx*.09 ; a(1)=w ; a(4)=w ; a(5)=w ; w2=w+dx*.10
14            e=tv(2) ; ee=e-dx*.09 ; a(2)=e ; a(3)=e          ; e2=e-dx*.00
15            s=tv(3) ; ss=s+dy*.10 ; b(1)=s ; b(2)=s ; b(5)=s ; s2=s+dy*.06
16            n=tv(4) ; nn=n-dy*.10 ; b(3)=n ; b(4)=n          ; n2=n-dy*.08
17            call tvplot(a,b,5,4hjoin)
18            call tvplot(a,b,5,4hjoin)
19      c--x--axis-------------------------------------------
20            if(cmd.eq.0)goto 31
21      c--x--numbers----------------------------------------
22      c-label x axis -- drop page space down .05 units
23            tv(7)=tv(7)-.05 ; ss=s+.05/(tv(8)-tv(7))*dy ; s2=s+.5*(ss-s)
24            idxlog=alog10(dx)+20 ; dxlog=idxlog-20 ; dxdist=10.**dxlog
25            ntics=dx/dxdist ; k8=(4-ntics)*(4-ntics)+1 ; dx=dxdist ; n8=ntics
26            if(ntics.le.3)dxdist=dxdist/k8
27            ntics=dx/dxdist+9 ; j=0
28            k9=10**5 ; k1=w*k9+.5 ; k2=dxdist*k9+.5 ; k=k1/k2*k2
29            x=(k-k2*2.)/k9          ; xprobe=x
30      c-cycle once per tic constructing that tic mark
31            do 10 i=1,ntics
32            x=x+dxdist        ; if(x.lt.(w-.001).or.x.gt.(e2+.001))goto 10
33            x2=x              ; if(x2.lt.w)x2=w
34            j=j+1             ; a(j)=x2 ; b(j)=s2
35            j=j+1             ; a(j)=x2 ; b(j)=ss
36            x0=x+.001         ; if(x.lt.0)x0=x-.001    ; encode(str,11)x0
37            absx=abs(x)       ; if(absx.eq. 0)absx=1   ; logx=alog10(absx+.001)
38            beg=10-logx       ; if(beg .gt.10)beg =10 ; beg=beg+3
39            absx=abs(dxdist) ; if(absx.eq. 0)absx=1   ; logx=alog10(absx)-.9
40            fin=11-logx       ; if(logx.ge. 0)fin =10 ; fin=fin+3
41            if(x.lt.0)beg=beg-1                       ; len=fin-beg+1
42            call justr (ju,str,beg,fin,1)
43            call tvltr (x,s,ju,len)
44         10 continue
45            call tvplot(a,b,j,7hsegment)
46            tv(7)=tv(7)+.05
47         11 format(f21.10)
48            goto 41
```

```
c--x--time-----------------------------------------------
c-label x axis with time marks
   31 tv(7)=tv(7)-.03 ; ss=s+.03/(tv(8)-tv(7))*dy ; s2=s+.8*(ss-s)
      nd=(tv(2)-tv(1))*(iv/1440.) ; days=nd ; i=0 ; j=0
   30 siz=0 ; num=0 ; len=2 ; if(nd.ge.1)goto 41
      call gait(yr,jl,i,y2,j2,m2,d2,mnm)
      if(nd.lt. ndy                                        )num=d2
      if(nd.lt. mdy                                        )siz=.2
      if(nd.lt. n10.and.(d2.eq.11.or.d2.eq.21)             )num=d2
      if(nd.lt. m10.and.(d2.eq.11.or.d2.eq.21)             )siz=.4
      if(nd.lt.km  .and.d2.eq.1                            )len=1
      if(nd.lt.kmon.and.d2.eq.1                            )len=3
      if(nd.lt.mmon.and.d2.eq.1                            )siz=.6
      if(nd.lt. m6m.and.m2-m2/06*06.eq.1.and.d2.eq.1)siz=.8
      if(nd.lt. nyr.and.j2.eq.1                            )num=y2
      if(nd.lt. nyr.and.j2.eq.1                            )len=2
      if(nd.lt. myr.and.j2.eq.1                            )siz=1.0
c-make the tic mark and label it if necessary
      if(siz.eq.0)goto 39 ; xloc=w+i/days*dx
      sizsav=siz
c-label the tic mark if necessary
      if(num.eq.0.and.len.eq.2)goto 32
      str(1)=mnm ; if(len.eq.2)encode(str,38)num
      call tvltr(xloc,s2,str,len)
c-plot the tic mark
   32 j=j+1 ; a(j)=xloc ; b(j)=s+(1.-siz)*(ss-s)
      j=j+1 ; a(j)=xloc ; b(j)=ss
c-kick i ahead if scale is suitably course
      if(nd.ge.mmon)i=i+25
c-call tvplot if a,b buffer is full
      if(j.lt.190)goto 39
      call tvplot(a,b,j,7hsegment) ; j=0
c-end of loop
   39 i=i+1 ; if(i.le.nd)goto 30
c-after looping...
      if(j.gt.0)call tvplot(a,b,j,7hsegment)
      jsav=j
      tv(7)=tv(7)+.03
   38 format(i2)
c--y--numbers--------------------------------------------
c-label y axis -- move page wider to the left by .09 units
   41 tv(5)=tv(5)-.09 ; ww=w+.09/(tv(6)-tv(5))*dx ; w2=w+.8*(ww-w)
      idylog=alog10(dy)+20 ; dylog=idylog-20 ; dydist=10.**dylog
      ntics=dy/dydist ; k8=(4-ntics)*(4-ntics)+1 ; d9=dydist ; n9=ntics
      if(ntics.le.3)dydist=dydist/k8
      ntics=dy/dydist+9 ; j=0
      k9=10**5 ; k1=s*k9+.5 ; k2=dydist*k9+.5 ; k=k1/k2*k2
      y=(k-k2*2.)/k9          ; yprobe=y
c-cycle once per tic construction each y axis tic mark
      do 20 i=1,ntics
      y=y+dydist        ; if(y.lt.(s-.001).or.y.gt.(n+.001))goto 20
      y2=y              ; if(y2.lt.s)y2=s
      j=j+1             ; b(j)=y2 ; a(j)=w2
      j=j+1             ; b(j)=y2 ; a(j)=ww
      y0=y+.001         ; if(y.lt.0)y0=y-.001    ; encode(str,11)y0
      absy=abs(y)       ; if(absy.eq. 0)absy=1   ; logy=alog10(absy+.001)
      beg=10-logy       ; if(beg .gt.10)beg =10 ; beg=beg+3
      absy=abs(dydist) ; if(absy.eq. 0)absy=1   ; logy=alog10(absy)-.9
      fin=11-logy       ; if(logy.ge. 0)fin =10 ; fin=fin+3
      beg=beg-2                                  ; len=fin-beg+1
```

```
109           call justr (ju,str,beg,fin,1)
110           call tvltr (s,y,ju,len)
111        20 continue
112           call tvplot(a,b,j,7hsegment)
113           tv(5)=tv(5)+.09
114           call tvsend ; return
115           end
```

.....geolab>garcol.fortran.....May 17, 1978....15:57.....

```
  1     c-----g a r c o l-----19 dec 77-----
  2           subroutine garcol(msg,e)
  3           integer msg,e,copy,bak,oldm,newm,savm,zw,dir
  4           integer mm,pn,pa,pz,dn,da,dz,nm,aa,az,
  5                   zn,zm,c,d,m,gbeg,adn,ada,adz,bdn,bda,bdz
  6     c--common-gl-mem----------------
  7           common /gl_state$/
  8             mm(120),tbuf(80),cenu(150),
  9             zc,zh,zm,zn,zr,zs,zu,zy,zz,zpad,
 10             c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,h,i,j,k,n,
 11             a,e,t,f,bksp,nptr,iter,nloo,gbeg,mpad(12),
 12             ca(40),pn(40),pa(40),pz(40),
 13             ff(40),gg(40),hh(40),ii(40),jj(40),kk(40),
 14             ty(40),dn(40),da(40),dz(40),dy(40),
 15             nm(900),aa(900),az(900),
 16             cena(150),yy(6),ynam(6),y1,y2,y3,y4,y5,y6,ypad(32),
 17             aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy
 18     cchange cntl-stk,scr-stk,a,b from abs to rel ptrs
 19           dir=-1 ; zw=4
 20        10 do 11 i=1,c ; n=pn(i)
 21              pa(i)=pa(i)+aa(n)*dir
 22        11    pz(i)=pz(i)+aa(n)*dir
 23           do 12 i=1,z ; n=dn(z) ; if(n.lt.1.or.n.gt.zn)goto 12
 24              da(i)=da(i)+aa(n)*dir
 25              dz(i)=dz(i)+aa(n)*dir
 26        12    continue
 27           if(adn.lt.1.or.adn.gt.zn)goto 13
 28           ada=ada+aa(adn)*dir ; adz=adz+aa(adn)*dir
 29        13 if(bdn.lt.1.or.bdn.gt.zn)goto 14
 30           bda=bda+aa(bdn)*dir ; bdz=bdz+aa(bdn)*dir
 31        14 if(dir.ne.-1)return
 32     cleft shift memory -- copy=-1=init  copy>0=copy til there
 33           newm=gbeg ; savm=gbeg-1 ; copy=-1
 34           do 6 oldm=gbeg,zm ; if(oldm.le.copy)goto 4 ; if(oldm.ge.m)goto 5
 35     c       -look for two way ptr loop with nm or aa
 36             bak=mm(oldm) ; if(bak.lt.1.or.bak.gt.zn)goto 3
 37             if(nm(bak).eq.oldm)goto 1; if(aa(bak).eq.oldm)goto 2; goto 3
 38     c       -ptr loop found with nm
 39           1 copy=oldm+(mm(oldm+1)+3)/zw+1 ; nm(bak)=newm ; goto 4
 40     c       -ptr loop found with aa
 41           2 copy=az(bak); az(bak)=copy-(oldm-newm); aa(bak)=newm; goto 4
 42     c       -if copy=-1=init then copy otherwise delete
 43           3 if(copy.ge.0)goto 6
 44           4 savm=newm
 45           5 mm(newm)=mm(oldm) ; newm=newm+1
 46           6 continue
```

```
47      cif msg=1 print out, assign anew to m
48            e=0 ; if(savm.gt.zm-100)e=1 ; savm=savm+1
49            if(e  .eq.1)print, "memory is full -- garbage col impossible"
50            if(msg.eq.0)write(6,9)savm,m ; m=savm
51          9 format(" garcol:"i7,"=new mptr",i7,"=old mptr")
52      cchange cntl-stk,scr-stk,a,b back to abs ptrs from rel ptrs
53            dir=1 ; goto 10
54            end
```

.....geolab>justr.fortran.....May 17, 1978....15:57.....

```
 1      c-----j u s t r-----left justify strings (rw=1), or undo it (rw=0)
 2            subroutine justr(ju,mm,da,dz,rw)
 3            integer ju(1),mm(1),da,dz,rw,temp,da4
 4            n=dz-da+1  ; da4=da-4 ; if(rw.eq.0)goto 111
 5            do 1 i=1,n ; j=i+da4
 6               call bits(mm,j,temp,4,.false.)
 7          1    call bits(ju,i,temp,4,.true. )
 8            do 2 i=1,4 ; j=n+i ; temp=32
 9          2    call bits(ju,j,temp,4,.true. )
10            return
11        111 do 3 i=1,n ; j=i+da4
12               call bits(ju,i,temp,4,.false.)
13          3    call bits(mm,j,temp,4,.true. )
14            return
15            end
```

.....geolab>prompt.fortran.....May 17, 1978....15:57.....

```
 1      c-----p r o m p t-----25 jan 78-----
 2            subroutine prompt(n)
 3            character*2 ask
 4            equivalence (ask,k)
 5            k=2hx? ; call bits(k,1,n,4,.true.)
 6            call io("put_chars","user_output",ask,"-nnl") ; return
 7            end
```

.....geolab>backsp.fortran.....May 17, 1978....15:57.....

```
 1      c-----backsp-----make backsp an erase char-----15 dec 77-----
 2            subroutine backsp(l,len)
 3            integer l(1),len
 4            j=1
 5            do 1 i=1,len ; k=l(i)
 6               n=1 ; if(i.gt.  1)n=l(i-1)
 7               m=1 ; if(i.lt.len)m=l(i+1)
 8               if(k.eq.8.or.n.eq.8.or.m.eq.8)goto 1
 9               l(j)=k ; j=j+1
```

```
      1     continue
        if(j.gt.len)return
        do 2 i=j,len
      2     l(i)=32
        return
        end
```

.....geolab>cmdproc.pl1.....May 17, 1978....15:57.....

```
    /*----c m d p r o c-----07 dec 77----*/
          cmdproc: proc(strg,len,code);
          dcl strg char(80);
          dcl (len,code) fixed bin;
          dcl c fixed bin(35);
          dcl p pointer;
          dcl cu_$cp entry(ptr,fixed bin,fixed bin(35));
          p=addr(strg) ; call cu_$cp(p,len,c) ; code=c;
          end;
```

.....setmem.fortran.....May 17, 1978....15:57.....

```
    c-----s e t m e m-----20 apr 78-----
    c--setmem-identifiers------
          integer beg,fin,nnn,rw,cenx(150),ynamx(6)
    c --gl-identifiers---------
          integer
            mm,tbuf,cena,cenu,
            zc,zh,zm,zn,zr,zs,zu,zy,zz,
            c,d,p,l,z,sz,m,lur,u,ptr,adr,ce,dlv,ncyc,g,i,j,k,n,
            bksp,nptr,iter,nloo,gbeg,lurd,
            y,y1,y2,y3,y4,y5,y6,yy,ynam,s1,s2,
            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
            umod,uyes,uno1,uno2,emsg,udef,
            ca,pn,pa,pz,ff,gg,hh,ii,jj,kk,
            ty,dn,da,dz,dy,s(40),
            nm,aa,az,
            rmem
          real r(40),rz,r1,r2,h,usym,unew
          integer a,e,t,f
          integer xup,xlp,xgm,xcl,xty,ju(200),jv(40)
          integer g1,g2,i1,i2,j1,j2,k1,k2,n1,n2
          integer gpool,gtune,tvse
    c--common--gl-mem----------------------------------------------
    c  mm    zc     c    ca    ff    ty    nm    aa    az cena     u rmem    end
    c0001 0351 0361 0401 0561 0801 1001 1901 2801 3701 3911 4001 10000
          common /gl_state$/
            mm(120),tbuf(80),cenu(150),
            zc,zh,zm,zn,zr,zs,zu,zy,zz,zpad,
            c,d,p,l,z,sz,m,lur,u,otr,adr,ce,dlv,ncyc,h,i,j,k,n,
            a,e,t,f,bksp,nptr,iter,nloo,gbeg,lurd,mpad(11),
            ca(40),pn(40),pa(40),pz(40),
            ff(40),gg(40),hh(40),ii(40),ji(40),kk(40),
```

```
32            ty(40),dn(40),da(40),dz(40),dv(40),
33            nm(900),aa(900),az(900),
34            cena(150),yy(6),ynam(6),y1,y2,y3,y4,y5,y6,ypad(32),
35            aty,adn,ada,adz,ady,bty,bdn,bda,bdz,bdy,
36            umod,usym,unew,uyes,uno1,uno2,udef,upad(3),
37            emso(10),epad(20),
38            gpad(1),tvse,gpool(8),gtune(30),xpad(10),
39            rmem(6000)
40      c--common--tvpool-8-tvtune-----------------------------------
41            common /tvpool/ ipool(8)
42            common /tvtune/ itune(30)
43      c--equivalence-----------------------------------------------
44            equivalence (xup,cenu(  5)),(xlp,cenu(  3)),(r ,s ),(s1,r1),
45                 (dv,s),(xcl,cenu(149)),(xtv,cenu(150)),(rz,sz),(s2,r2),
46                      (xgm,cenu(134))
47      c--end-of-gl-declarations------------------------------------
48      c--ce-names--------------------------------------------------
49      c          1111    2222    3333    4444    5555    6666    7777    8888
50            data cenx/
51             4hinit,4hrset,4h(    ,4heval,4h"    ,4h)    ,4h]    ,4hquox,
52             4hquo ,4hastr,4h'    ,4hskip,4htyp0,4htyp1,4htyp2,4htyp ,
53             4hnam ,4hbeg ,4hfin ,4hdyn ,4h=     ,4haf   ,4hfa   ,4h$    ,
54             4hof  ,4hug  ,4hgu  ,4hcalr,4hifju,4hdecl,4hslip,4hcnam,
55             4hdrop,4heot ,4hchpc,4hup2 ,4hdynx,4hdyn0,4hmakd,4hact3,
56             4hdolu,4hwhiz,4hguts,4hvvm ,4hvsm ,4hmops,4hnptr,4hx048,
57             4hx049,4hx050,4hplus,4hminu,4htims,4hdiv ,4hidiv,4hpow ,
58             4hmodx,4hminx,4hmaxx,4hrnd ,4hsi   ,4hco   ,4hta   ,4hasi ,
59             4haco ,4hata ,4hlogx,4hl10x,4hex   ,4hab   ,4hsqr ,4hchs ,
60             4hflot,4hfix ,4hnotx,4heq   ,4hne   ,4hlt   ,4hle   ,4hgt  ,
61             4hge   ,4hx082,4handx,4horx ,4hsia ,4hcmdp,4hprom,4hcodr,
62             4hio   ,4hx090,4his8 ,4his4 ,4hrstr,4hisd ,4his  ,4hx096,
63             4hx097,4hx098,4hx099,4hx100,4hf    ,4hg    ,4hh    ,4hi    ,
64             4hj    ,4hk    ,4hpc   ,4hx108,4hmem ,4hstk ,4h,    ,4hxchg,
65             4hpush,4hlkup,4hrwel,4hsubi,4hsubj,4hsui1,4hsui2,4hx120,
66             4hkint,4hvint,4hkrea,4hvrea,4haint,4harea,4hvdes,4hvdas,
67             4ha    ,4hb    ,4hdump,4hgarc,4hstae,4hcmem,4hlure,4hpars,
68             4h.    ,4hce   ,4hstop,4htrpt,4hectl,4hestk,4henam,4hedec,
69             4hetyp,4heidx,4hexup,4hemis,4h!    ,4htty /
70      c--initialization--------------------------------------------
71         data ynamx/4hintg,4hreal,4hlett,4hiarr,4hrarr,4hstrg/
72         y1=1 ; y2=2 ; y3=3 ; y4=4 ; y5=5 ; y6=6
73         yy(1)=1 ; yy(2)=1 ; yy(3)=1 ; yy(4)=1 ; yy(5)=1 ; yy(6)=4
74         zc=39   ; zh=150  ; zm=50000 ; zn=900  ; zr=30   ; zs=35   ; zu=999
75         zy=6    ; zz=5    ; m=4001 ; lurd=2
76         umod=0  ; usym=99999. ; unew=usym ; udef=1
77         t=1     ; f=0     ; a=f     ; lur=5   ; dlv=0   ; ncvc=0
78         gpad(3)=   0 ; gpad(4)=100 ; gpad(5)=0 ; gpad(6)=100
79         gpad(7)=0  ; gpad(8)=1  ;  gpad(9)=0  ; gpad(10)= 1
80         gtune(2)=80  ; gtune(3)=0  ; gtune(4)=1; gtune(5)=0 ; gtune(15)="data"
81         gtune(6)=0   ; gtune(7)=0  ; gpad(18)=.5
82         do 80 i=1,zh
83         cena(i)=cenx(i)
84      80 cenu(i)=cena(i)
85         do 81 i=1,zy
86      81 ynam(i)=ynamx(i)
87         do 82 i=1,zn
88         nm(i)=0 ; aa(i)=0 ; az(i)=-1
89      82 continue
90         do 83 ce=1,zh
91         name=cena(ce) ; nchar=1
```