

```

1  !MS1-Launch.f95
2  !2025.01.25.1420cst JMS- Use for Non-OpenGL algorithm testing
3  ! Computer- "T4"/HP-800-G4-Mini/i7-8700T/IntelUHD630/win10Pro-22H2
4  !           ^name ^Mfgr.Id ^chipset ^graphics ^OS
5  !           /AbsoftProFortran 21.0.2/OpenGL+Glut3.6
6  !           ^compiler           ^Fortran graphics
7  !Table of Contents - ToC:
8  !Module MS1Def
9  !Program MS1Launch           Opens cOutFile
10 !Subroutine SaveOutFile      Closes/Opens cOutFile
11 !Subroutine Jdate22(DaTime22L)
12 !Subroutine nRunSec(nClock,RunSecs,iP)
13 !Subroutine MyLenTrim(cLabel,nCu,iP)
14 !Subroutine Beamer(N,Nmax)
15 !-----7 9
16
17 Module MS1Def
18 !////////////////////////////////////
19 !use MS1Def                               JMS 2024.01.22
20
21 !-- Identifiers:
22 character*54::cProjectName = 'MS-RC-64' !a19
23 character*20::cProjDate    = '2025.01.25.1420cst'
24 character*54::cExeName     = 'MS-RC-64.exe' !"Min-Step Rubik`s Cube 64-bit"
25 character*5  ::cExeVsn     = 'v1.0'
26 character*3  ::cInitials   = 'JMS'
27
28 !-- i/o Files:
29 character*79::cNmIFile     = 'Nn-NmI.nmI' !hard-coded to 'MS4.ini'
30 character*79::cInitFile   = 'TBD-Init.txt'
31 character*79::cInFile     = 'TBD-In.txt'
32 character*79::cOutFile    = 'TBD-Out.txt'
33
34 !-- File access flags:
35 logical      ::LExists     !File existence flag
36 integer(4)::iAlloc        !Array allocation flag =0:success
37
38 !-- Read/write Voter#'s
39 integer(4)::Ur =11 !Voter#- to read various input files.
40 integer(4)::Us = 6 ! - write to- screen - on
41 integer(4)::Ut =13 ! = 13 all writes to ASCII file cOutFile, however
42 ! ...= Us - when cOutFile open fails
43 integer(4)::Up = 0 !Voter#= 0 - usually off. usage: if(Up>0) write(Up,...)
44 ! = Us- for one iteration- <- press key:"p"
45 ! = Ut- - <- : "P"
46
47 integer(4)::Uread1 =14! 14- for user use in apps
48 integer(4)::Uread2 =15! 15- for user use in apps
49 integer(4)::Uwrite1=16! = 16- (set =13 when not in use.)
50 integer(4)::Uwrite2=17! = 17- (set =13 when not in use.)
51 integer(4)::Urdwr =18! = 18- read & write
52
53 integer(4)::UseIfie=10 !For exporting selfies to .bmp files
54
55 !-- Millisec counter:
56 real(8) ::RunSecs !Elapsed seconds since program launch, msec resolution.
57 !-- Millisecond Millenia Differential Timers: (Good for > a thousand years!)
58 character ::DaTime22*22
59 integer(4)::JulianDay
60 real(8) ::DaySecond !...if & when it is debugged.
61 type :: TimeRec;sequence !2023.07.16
62 integer(4) ::nClock !Clock# (0:30)

```

```

63  integer(4)  ::V(8)  !Output of: date_and_Time(values=V)
64                !V( ) = 1:iYear, 2:iMonth, 3:iDay, 4:Utc time offset(if any)
65                !      ,5:iHour, 6:iMin  , 7:iSec, 8:imSec
66                !
67  character(22)::cDat22  !Character string:          24->22   2023.08.22
68  integer(4)  ::JDay    !Julian Day
69  real(8)     ::Sec     !Day second... millisecond resolution
70  end type TimeRec
71  type(TimeRec)::DaT    !Current Date & Time- computed by: Jdate20(DaTime)
72  type(TimeRec)::DaTo(0:30)!30 separate clocks -   nRunSec(nClock,RunSecs,iP)
73
74  integer(4)::NowPrint!=0 usually; during one iteration...
75                !=1 by pressing key: 'p' :fp=fpt    ->.txt
76                !=2                : 'P' :fp=stdout ->screen
77  !Contains
78  !-----7 9
79  !////////////////////////////////////
80  End Module MS1Def
81  !-----7 9
82
83  Program MS1Launch
84  !2025.01.25.1420cdt JMS- Non-OpenGL algorithm testing
85  !--Globals
86  use MS1Def ,only:                                     &
87      cProjectName,cProjDate,cExeName,cExeVsn         &
88      ,cNm1File,cInitFile,cInFile,cOutFile,LExists,iAlloc &
89      ,Ur,Us,Ut,Up,Uread1,Uwrite1,DaTime22,DaT,DaTo,RunSecs
90  !--End Globals
91  implicit none                                       !internals
92  !--Arguments
93  !--Internals
94  character*22 ::cToday  =('2025.01.22.2045cst JMS')
95  character*9  ::cThisApp=('MS-RC')
96  character*5  ::cThisVsn =('v1.0')
97  integer(4)  ::i
98  !--EndDefs-----
99  Ut          = 6
100
101  call nRunSec( 0,RunSecs,6)
102  write(Ut,"(a22,' Begin: ',a22,1x,a5)") &
103      DaT%cDat22      ,cExeName(1:22),cExeVsn
104  write(uT,"('      Use this free software at your own risk:'))"
105  write(uT,"('      Press enter to continue      -or-'))"
106  pause              "      Close this command prompt window."
107
108  !write(Ut,"(22x,'      ',a9,14x,a5)") &
109  !      cThisApp,cThisVsn
110  write(Us,"('/NoGL: reading your NameList file...')")
111
112  !-- Application specific:
113  call ReadYourNm1(cNm1File,Ur,Us,Ut,cOutFile) !'MS4.ini designates *.nm1'
114
115  !-- Setting up the output file:
116  if(Ut<10) goto 19
117  write(Us,"('Ut      = '      ,i2)") Ut
118  write(Us,"('cOutFile = ',:,79a1)") &
119      (cOutFile(i:i),i=1,len_trim(cOutFile))
120  !write(Us,"('CGeomTest = '      ,a8)") CGeomTest
121
122  open(unit=Ut,file=cOutFile,action='write',status='replace',err=19); goto 20
123  19 Ut=Us
124  write(Us,"('Opening the output text file failed or was disabled.))")
125  write(Us,"('      Ut=Us & the output text will go to the DOS screen.))")

```

```

125 write(6, ( ut=ut, & the output text will go to the DOS screen. ) )
126 20 call JDate22(DaTime22)
127 write(ut,"(a22,' Begin: ',a22,1x,a22)") DaTime22,cExeName,cExeVsn
128 call nRunSec( 0,RunSecs,ut)
129 call SaveOutFile
130
131 !-- Call your app(s) here, with a Ut argument: -----
132 call ImportEin(ut) !This branches to Ein%iType
133
134 !-- Application-specific calls done. Normal exit. -----
135 call nRunSec( 0,RunSecs,6)
136 if(ut>6) write(ut,"(/a22,' End: ',:,54a1)") &
137 DaT%cDaT22,(cExeName(i:i),i=1,len_trim(cExeName))
138 call SaveOutFile
139 if(ut>6) close(ut)
140
141 write( 6,"(a22,' End: ',:,54a1)") &
142 DaT%cDaT22,(cExeName(i:i),i=1,len_trim(cExeName))
143 write(6,"(a1)" char(7) !Beep
144 End Program MS1Launch
145 !-----7-9
146
147 Subroutine SaveOutFile !2023.10.21.1740
148 !2023.10.21.1740cdt JMS
149 !--Globals
150 use MS1Def, only: ut,cOutFile
151 !--End Globals
152 implicit none
153 !--Arguments
154 !--Internals
155 !--EndDefs-----
156 if(ut<12) return
157 close(ut)
158 open(ut,file=cOutFile,action='write',position='Append') ;return
159 End Subroutine SaveOutFile
160 !-----7-9
161
162 Subroutine Jdate22(DaTime22L) !2023.08.20.1210
163 !2023.08.11.0600cdt JMS- Reads a snapshot of the computers current date & time.
164 use MS1Def, only: DaT
165 implicit none
166 !--Arguments
167 character::DaTime22L*22
168 !--Internals
169 integer(4)::v(8)
170 integer(4)::iYear,iMonth,iDay,iHour,iMin,iSec,imSec
171 !integer(4)::JulianDay
172 !real(8) ::DaySecond
173 !--EndDefs-----
174 call date_and_Time(VALUE=v) !V(4) =Time offset w.r.t. UTC
175 iYear = v(1); iHour = v(5)
176 iMonth = v(2); iMin = v(6)
177 iDay = v(3); iSec = v(7)
178 imSec = v(8)
179 write(DaTime22L & !includes: milliseconds
180 ,(i4,'.',i2.2,'.',i2.2,'.',i2.2,i2.2,'.',i2.2,'.',i3.3)") &
181 iYear,iMonth,iDay,iHour,iMin,iSec,imSec
182 DaT%cDaT22 = DaTime22L
183 !Gregorian Calendar -> JulianDay (author unknown)
184 DaT%jDay = 367* iyear &
185 -7* (iyear+(imonth+9)/12) /4 &
186 -3*((iyear+(imonth-9)/ 7)/100+1)/4 &

```

```

187          +275*          imonth          /9+iday+1721029
188      !DaySecond
189      DaT%Sec = (iHour*60.d0+iMin)*60.d0+iSec*1.d0+imSec/1000.d0
190      !Use DaySecond (& JulianDay) for differential timing to 1 millisc:
191      return
192 End Subroutine Jdate22
193 !-----7 9
194
195 Subroutine nRunSec(nClock,RunSecs,iP)
196 !2023.10.28.1400cdt JMS- Run seconds for (0:30) separate clocks Not verified
197 !2024.08.20.1220cdt JMS- Error: seconds do not transit midnight correctly.
198 !--Globals          ... 1. millisecond resolution
199 use MS1Def, only: DaT, DaTo
200 !--End Globals
201 implicit none
202 !--Arguments
203 integer(4) ::nClock !<0:start&reset; >0:read; =0:sec since first call
204 real(8)    ::RunSecs !nClock`s elapsed seconds ^to this subroutine
205 integer(4) ::iP
206 !--Internals
207 integer(4) ::Init = 0
208 integer(4) ::nClockL
209 character  ::DaTime22L*22
210 !--EndDefs-----
211 call Jdate22(DaTime22L) !Computes DaT
212 nClockL = nClock
213 if(Init==0) then
214     DaTo(0) = DaT
215     Init = 1
216 endif!(Init==0)
217 select case(nClock)
218 case( :-31); Stop 'nRunSec(nClock <-30... undefined. Halt@L214'
219 case(-30: -1); nClockL = -nClockL; DaTo(nClockL) = DaT !Resets the clock
220 case(+31: ); Stop 'nRunSec(nClock >+30... undefined. Halt@L216'
221 end select!(nClock)
222 RunSecs = (DaT%Sec -DaTo(nClockL)%Sec ) &
223           +(DaT%JDay-DaTo(nClockL)%JDay)*86400.d0
224 if(iP>5) write(iP, "('Clock',SP,i3.2,S, ':',f31.3,' sec')") nClock,RunSecs
225 !Debug:          2024.08.20.1220
226 if((abs(RunSecs)>80000.d0).and.(iP>5)) then
227     write(iP, "('NoGL.f95\nRunSec()@L223: possible seconds timing error:')")
228     write(iP, "('TimeRec records follow:')")
229     write(iP,*) 'Dat :',DaT
230     write(iP,*) 'Dato:',DaTo
231     call SaveOutFile
232     pause 'Press enter to terminate the run...'
233     stop '@L229'
234 endif!((abs(RunSecs)>43200.d0).and.(iP>5))
235
236 End Subroutine nRunSec
237 !-----7 9
238
239 Subroutine MyLenTrim(cLabel,nCu,iP) !2013.11.07.0510
240 !2013.11.07.0510cst JMS- My version of len_trim.
241 ! Trailing spaces, C/R's, & LF's are char(0)'d.
242 ! Interior char(0)'s are converted to spaces.
243 implicit none
244 character*(*)::cLabel !arguments
245 integer(4) ::nCu
246 integer(4) ::iP
247
248 integer(4) ::nSize !internals

```

```

249 integer(4)    ::i
250 character    ::A0*1,A32*1
251 !-----
252 nSize=len(cLabel)
253 if(iP.gt.5) then; write(iP,*) 'cLabel=',cLabel
254                 write(iP,*) 'cLabel=',(iChar(cLabel(i:i)),i=1,80)
255                 write(iP,*) 'nSize =',nSize
256 endif !iP>5
257
258 A0 =char(0) ;A32=char(32) ;nCU=0
259 do i=nSize,1,-1
260   select case(iChar(cLabel(i:i)))
261     case( 0) ;goto 10
262     case(10) ;goto 10
263     case(13) ;goto 10
264     case(32) ;goto 10
265     case default; if(nCU.gt.0) cycle ;nCU=i ;cycle
266   end select !iChar(cLabel(i:i))
267 10  if(nCU.eq.0) cLabel(i:i)=A0
268     if(nCU.gt.0) cLabel(i:i)=A32
269   enddo!i
270
271   if(iP.gt.5) then
272     write(iP,"('nCU=',i6)") nCU
273     do i=1,nSize ;write(iP,"(i4,\)") iChar(cLabel(i:i));enddo
274                 write(iP,"(' ')")
275   endif!iP>5
276   return
277 End Subroutine MyLenTrim
278 !-----7 9
279
280 Subroutine Beamer(N,Nmax)
281 implicit none
282 integer(8) :: kStat,n,nMax,nStat
283 !-----
284 if(Nmax .lt. 1) stop 'BeamerL: Nmax < 1'
285 if(N .le. 1) then; Kstat=0 ;write(6,"(1x,\)") ;endif
286                 Nstat=(N*78)/Nmax
287 10 if(Nstat.gt.Kstat) then; Kstat=Kstat+1 ;write(6,"('*',\)") ;goto 10 ;endif
288    if(N .ge.Nmax ) write(6,"('*')")
289    return
290 End Subroutine Beamer
291 !-----7 9
292

```