

```

1  !MS6-RC-Solve.f95
2  !2025.01.25.1445cst JMS- Rubik's Cubes - wisdom on Call
3  !
4  !           - For:2x2x2 & 3x3x3 Rubik's Cubes
5  !           Computer- "T4"/HP-800-G4-Mini/i7-8700T/IntelUHD630/win10Pro-22H2
6  !           ^name ^Mfgr.Id ^chipset ^graphics ^OS
7  !           /AbsoftProFortran 21.0.2/OpenGL+Glut3.6
8  !           ^compiler ^Fortran graphics
9  !
10 !Table of Contents - ToC:
11 !Subroutine GenScramble(iP) -opens Data.Emulator.txt
12 !Subroutine SolveScramble(iP) -opens Data.Emulator.txt
13 !Subroutine AtoD(VtotL,VAin,iDiscord,iP) -opens Discord.bim files
14 !-----7 9
15 Subroutine GenScramble(iP)
16 !2025.01.21.1830cst JMS- !Generates the scramble defined by Ein in a .nml file.
17 !           using: AECAV().
18 !           Processes an arbitrary voter subset.
19 !--Globals
20 use MS1Def ,only: Ur,Us,Ut
21 use MS2RCDef ,only: SRrec,resetSR,AECAV,Rtot8,C,MC &
22 ,Ein,RubSize,Sfound8,Vtot,V,Asymbol &
23 ,ALV &
24 ,SrTestIn,Mtot,MC,Ctot,C,Vtot,V
25 !--End Globals
26 implicit none
27 !--Arguments
28 integer(4)::iP !Write enable>5: write(iP,...)
29 !--Internals
30 integer(4)::Init
31 integer(4)::nV,nV2 !Voter- index
32 integer(1)::nVE ! = V(nV) is used within the model- cell#
33 integer(4)::nA,nA2 !Choice- index
34
35 type(SRrec)::SRin !The cube being scrambled
36 integer(4)::nM !Move counter
37 integer(1)::nCE ! = C(nC) is used within the model- rotation#
38 integer(1)::VAn(27)!now copy of SRn%VA()
39
40 !--EndDefs-----
41 if(iP>5) write(iP,"(/'GenScramble: MS6-RC-Solve @L41'/)")
42 if(Init==0) then
43 write(Us,"('GenScramble:Import Data-Emulator.txt:AECAV(0:19,24,27):')")
44
45 if(iP>5) write(iP,"(/,'GenScramble initializing @L45'/)")
46 !-- Import AECAV:
47 if(iP>5)write(iP,"('Import AECAV(0:19,24,27): 2025.01.05.1946:')")
48 open(11, file='Data-Emulator.txt', action='read' &
49 , access='sequential' , status='old' )
50 read(11,"(////)") !skips the first 5 lines
51 !write(13,"(/' | D | E | F | G | H | I | ' )")
52 !! write(13,"( ' 0 1 2 3 4 5 6 7 8 9 A B C',\ )" )
53 !! write(13,"(' | D E F G H I _' /)")
54 do nV = 1,27
55 do nA = 1,24
56 read(11,*) nV2,nA2,AECAV(0:19,nA2,nV2)
57 !write(13,"(2i4,':',20i3)") nV2,nA2,AECAV(0:19,nA2,nV2)
58 enddo!nA
59 enddo!nV
60 close(11)
61 Init = 1
62 endif!(Init==0)

```

```

63  !-----
64  !SRin = SrTestIn
65  !call PrintSRrec(SrTestIn,iP,'The input to GenScramble')
66  if(iP>5) write(iP,"('Mtot = ',i2,' Vtot = ',i2)") Mtot,Vtot
67  if(iP>5) write(iP,"('Voters: ',:,20(i2,','))") Ein%V(1:Vtot)
68
69  Ein%VA      = 1
70  !Cycle through the Moves
71  do nM = 1,Ein%Mtotvis
72  VAn(1:20) = Ein%VA(1:20)
73  nCe = Ein%MCvis(nM)
74  !cycle through all the active cells:
75  do nV = 1,Vtot; nVE = Ein%V(nV)
76  !Here's the beef: the face rotation changes one cell's attitude:
77  Ein%VA(nV) = AECAV(nCE,VAn(nV),nVE)
78  ! ... & the cell's attitude character symbol is updated:
79  Ein%cVAtest(nV:nV) = char(Asymbol(Ein%VA(nV)))
80  enddo!nV
81
82  if(iP>5) then
83  write(iP,"('Move Choice = ',i2,' ',:,20a1)") &
84  nCe,char(Asymbol(AECAV(nCE,1,Ein%V(1:Vtot))))
85  write(iP,"(8x, :,20(i2,','))") AECAV(nCE,1,Ein%V(1:Vtot))
86
87  write(iP,"('result#',i3,9x, :,20a1)") &
88  nM,char(Asymbol(Ein%VA(1:Vtot)))
89  write(iP,"(8x, :,20(i2,','))") Ein%VA(1:Vtot)
90  endif!(iP>5)
91  enddo!nM
92  !Modelling completed. ----- in four lines of Fortran
93  call PrintErec(Ein,iP,'GenScramble output @L93')
94
95  End Subroutine GenScramble
96  !-----7 9
97
98  Subroutine SolveScramble(iP)
99  !2025.01.25.1445cst JMS- !Exercises the scramble defined by Ein in a .nml file.
100 !--Globals
101 use MS1Def ,only: Ur,Us,Ut,LExists,iAlloc
102 use MS2RCDef ,only: SRrec,resetSR,AECAV,Rtot8,C,MC &
103 ,Erec,Ein,EinW,RubSize,Sfound8,Vtot,V,Asymbol &
104 ,AlV,Csymbol,nCinv4 &
105 ,SrTestIn,Mtot,MC,Ctot,C,Vtot,V,Ztot,Z
106 !--End Globals
107 implicit none
108 !--Arguments
109 integer(4)::iP !write enable>5: write(iP,...)
110 !--Internals
111 integer(4)::Init
112 integer(4)::nV,nV2 !voter - index
113 integer(4)::nA,nA2 !choice- index
114
115 integer(1)::nC ,nC1 !the symbolic move
116 integer(4)::nC4,Nc4m
117
118 integer(1) ::VtotL
119 integer(4) ::VAin(20) ! The votes [1:VtotL]
120 character(20)::cVAin !Sortable character version of VAin(1:Vtot)
121 integer(4) ::VAout(20) ! The votes [1:VtotL]
122 character(20)::cVAOut !Sortable character version of VAiout(1:Vtot)
123
124 integer(4)::iDiscord(0:9)!Function results(i.e.:output:returned values)
125 ! (0) is the sum =-1:all invalid

```

```

125                                     ! (0) is the sum, =-1. all invalid
126                                     ! (1): R3-08161920-12-RrToDis.bim
127                                     ! (2): R2C-1234567-6-RrToDis.bim
128                                     ! (3): R3E-091011121314151718-6-RrToDis.bim
129 integer(4)::iDscd01(-18:18)
130 integer(4)::iDscd18(-18:18)
131 integer(4)::i,nOrder(0:9)
132 integer(4)::VAhell(20) = & !a scramble from hell:
133     (/ 7,10,23,13,23,15,22,14, 4,18,10,14,18, 2,11,22,19,22, 8,22/)
134 !     (/ 3,7,1,9,16,1,6,13,1,5,1,10,5,2,10,1,5,13,10,2,5,10,13/) move choices
135
136 integer(4)::iPrint,iP2
137
138 integer(4)::VA159(20) = & !Result of moves ->1->5->9 -> "tdtfbdkadtffbdkadbaa"
139     (/20, 4,20, 6, 2, 4,11, 1, 4,20, 6, 6, 2, 4,11, 1, 4, 2, 1, 1/)
140
141 integer(4)::VAprev(20) = & !Choice #2
142     (/ 3, 1, 3, 1, 3, 1, 3, 1, 1, 3, 1, 1, 3, 1, 3, 1, 1, 3, 1, 1/)
143 integer(4)::VAnew( 20)
144 integer(4)::VAnow( 20) = & !Choice #6
145     (/ 5, 5, 1, 1, 5, 5, 1, 1, 5, 1, 1, 1, 5, 5, 1, 1, 5, 1, 1, 1/)
146 type(Erec)::Euse,Euse2 !Accumulates moves.
147 character*2::cMC
148 !---EndDefs-----
149
150 if(iP>5) write(iP,"(/'SolveScramble: MS5-RC-Solve@L148'/)")
151 if(Init==0) then
152     write(us,"('SolveScramble:Import Data-Emulator.txt: AECAV(0:19,24,27):')")
153
154     if(iP>5) write(iP,"(/,'SolveScramble initializing @L152'/)")
155     !-- Import AECAV:
156     if(iP>5)write(iP,"('Import AECAV(0:19,24,27): 2025.01.05.1946:')")
157     open(11, file='Data-Emulator.txt', action='read' &
158         , access='sequential' , status='old' )
159     read(11,"(////)") !Skips the first 5 lines
160     do nv = 1,27
161         do nA = 1,24
162             read(11,*) nv2,nA2,AECAV(0:19,nA2,nv2)
163             !if(iP>5) &
164             !write(iP,"(2i4,':',20i3)") nv2,nA2,AECAV(0:19,nA2,nv2)
165         enddo!nA
166     enddo!nv
167     close(11)
168     Euse = Ein
169     Init = 1
170 endif!(Init==0)
171 !-----
172 !Ein% input:
173 VtotL = Ein%Vtot
174 VaIn = 0
175 VAin(1:VtotL) = Ein%VA(1:VtotL)
176 do i=1,20
177     cVAin(i:i) = char(Asymbol(VAin(i)))
178 enddo!i
179 ! VtotL = 20
180 call Concatenate(VtotL,VAnew,VAnow,VAprev, 0) !or iP
181
182 ! if(iP>5) write(iP, &
183 !     ("(/,'SolveScramble testing AllMoves @L181 - works!')")
184 !     VtotL=20; VAprev=1; nC4=1 !aaaaaaaaaaaaaaaaaaaaaa
185 call AllMoves(VtotL,VAnew,VAprev, 0) !aaaaaaaaaaaaaaaaaaaaaa
186

```

```

187   if(0>1) then !This bypasses a deep scramble printout of non-quad edge cells
188   !%Mused = 18 %MC(1:18) = 1 1 5 5 9 9 1 6 A 2 5 5 2 5 9 9 1 6
189       VAprev = VAnew
190   call AllMoves(VtotL,VAnew,VAprev, 1 ,iP) ! 1  0 0  0 0 0 0
191       VAprev = VAnew
192   call AllMoves(VtotL,VAnew,VAprev, 1 ,iP) ! 2
193       VAprev = VAnew
194   call AllMoves(VtotL,VAnew,VAprev, 5 ,iP) !
195       VAprev = VAnew
196   call AllMoves(VtotL,VAnew,VAprev, 5 ,iP) !
197       VAprev = VAnew
198   call AllMoves(VtotL,VAnew,VAprev, 9 ,iP) !
199       VAprev = VAnew
200   call AllMoves(VtotL,VAnew,VAprev, 9 ,iP) !
201       VAprev = VAnew
202   call AllMoves(VtotL,VAnew,VAprev, 1 ,iP) !
203       VAprev = VAnew
204   call AllMoves(VtotL,VAnew,VAprev, 6 ,iP) !
205       VAprev = VAnew
206   call AllMoves(VtotL,VAnew,VAprev,10 ,iP) !
207       VAprev = VAnew
208   call AllMoves(VtotL,VAnew,VAprev, 2 ,iP) !10
209       VAprev = VAnew
210   call AllMoves(VtotL,VAnew,VAprev, 5 ,iP) !
211       VAprev = VAnew
212   call AllMoves(VtotL,VAnew,VAprev, 5 ,iP) !
213       VAprev = VAnew
214   call AllMoves(VtotL,VAnew,VAprev, 2 ,iP) !
215       VAprev = VAnew
216   call AllMoves(VtotL,VAnew,VAprev, 5 ,iP) !
217       VAprev = VAnew
218   call AllMoves(VtotL,VAnew,VAprev, 9 ,iP) !
219       VAprev = VAnew
220   call AllMoves(VtotL,VAnew,VAprev, 9 ,iP) !
221       VAprev = VAnew
222   call AllMoves(VtotL,VAnew,VAprev, 1 ,iP) !
223       VAprev = VAnew
224   call AllMoves(VtotL,VAnew,VAprev, 6 ,iP) !18
225       VAprev = VAnew
226   endif!(0>1) bypass.
227
228   call AtoD(VtotL,VAout,iDiscord,iP) !initializes AtoD Zones & returns
229
230   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
231   do iPrint = 1,2 !Prints to Screen & then iP >=10:
232       select case(iPrint)
233           case(1); iP2 = 6
234           case(2); iP2 = iP; if(iP2<10) exit
235       end select!(iPrint)
236   call PrintErec(Ein,iP2,'SolveScramble input @L236')
237   write(iP2, &
238       "(/,50('|') ,//, 'Move = 0 closes this program.')")
239   write(iP2, &
240       "(/, 'Negatives of Move Sequence values will deconstruct the scramble'")
241   write(iP2, &
242       "( 'to solved by post-multiplication.',/ )")
243   write(iP2, &
244       "( 'Otherwise, pick any move that reduces the Discord of a Zone, &' )")
245   write(iP2, &
246       "( 'thus the 2x2x2 RC will solve in a mimimum number of moves.',/ )")
247   write(iP2, &
248       "( 'Move#0 shows the individual zone Discord(s) and order(s).' )")
249   write(iP2, &

```

```

249 write(ip2, &
250 "( 'Order: the number of self-concatenations to solved.',/))"
251 write(ip2, &
252 "( 'For 2x2x2`s: now = ___ is only Zone:[1,2,3,4,5,6,7]' )")
253 write(ip2, &
254 "( 'For 3x3x3`s: now = ___ is Zone#1 followed by' )")
255 write(ip2, &
256 "( 'now = ___ the sum of Zone#2 & #3',/ )")
257 write(ip2, &
258 "(50('|'),/, 'Previous move [-18:18] = 0' )")
259 enddo!i
260 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
261
262 10 continue
263 if(ip>5) write(ip, "(/, 'cVAin = ',12x,a20))" cVAin
264
265 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
266 do iPrint = 1,2 !Prints to Screen & then iP >=10:
267 select case(iPrint)
268 case(1); iP2 = 6
269 case(2); iP2 = iP; if(ip2<10) exit
270 end select!(iPrint)
271 write(ip2, &
272 "( 'The current scramble, pre-mult`s on left, post-mults on right:')"
273 write(ip2, "( 'Moves:(',i2,':1) = ',\ )" Euse%Mtotvis
274 do i = Euse%Mtotvis,1,-1
275 if(Euse%MCvis(i)>0) write(cMC, "( '+' ,a1 )" Csymbol( Euse%MCvis(i) )
276 if(Euse%MCvis(i)<0) write(cMC, "( '-' ,a1 )" Csymbol( abs(Euse%MCvis(i)) )
277 if(mod(i,20)/=0) then
278 write(ip2, "(a2, ',',\ )" cMC
279 else
280 write(ip2, "(a2,/,17x,',',\ )" cMC
281 endif!(mod(i,20)/=0)
282 enddo!i
283 write(ip2,*)
284 select case(RubSize)
285 case(2); write(ip2, "( 'Ztot = ',i1,13x, 'Discord: Order:' )") Ztot
286 write(ip2, "( 'Move# Attitudes: Zone#: ',\ )" )
287 case(3); write(ip2, "( 'Ztot = ',i1,21x, 'Discord: Order:' )") Ztot
288 write(ip2, "( 'Move# Attitudes: Zone#: ',\ )" )
289 end select!(RubSize)
290
291 do i=1,Ztot;write(ip2, "(i4,\ )" ) Z(i) ;enddo!i
292 write(ip2, "( ' :',\ )" )
293 do i=1,Ztot;write(ip2, "(i5,\ )" ) Z(i) ;enddo!i
294 write(ip2,*)
295 enddo!i
296 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
297
298 !Cycle through all the Move Choices:
299 do nC = -18,18; Nc4 = Nc; Nc4m = nC4
300 ! do nC = 0, 0; Nc4 = Nc; Nc4m = nC4
301 if(RubSize==2) then
302 select case(abs(nC))
303 case(3,4,7,8,11,12,14,16,18); cycle
304 end select!(abs(nC))
305 endif!RubSize==2)
306 VtotL = Vtot
307 VAout = VAin
308 cVAout = cVAin
309 if(nC==0) goto 20 !...report the input case
310 call AllMoves(VtotL,VAout, VAout,nC4m, 0) !iP

```

```

311 do nV = 1,VtotL
312   cVAout(nV:nV) = char(Asymbol(VAout(nV)))
313 enddo!nV
314 20 continue
315 nOrder = 1
316 !if(iP>5) write(iP,"('nC = ',i2,' cVAout = ',a20,' @L297')") nC,cVAout
317 !Evaluate the resulting Discords:
318 call AtoD(VtotL,VAout,iDiscord, 0) !iP <- Discords computed here
319 call ComputeOrder(VtotL,VAout,nOrder, 0)
320 nC4 = nC; iDscd18(nC4) = iDiscord(0)
321     iDscd01(nC4) = iDiscord(1)
322 !if(RubSize==2) iDscd18(nC4) = iDiscord(2)
323
324 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
325 do iPrint = 1,2 !Prints to Screen & then iP >=10:
326   select case(iPrint)
327     case(1); iP2 = 6
328     case(2); iP2 = iP; if(iP2<10) exit
329   end select!(iPrint)
330   if(nC==0) write(iP2,*)
331   if(nC <0) write(iP2,"(i3,1x,'-',a1,1x,a20,\)") &
332     nC,char(Csymbol(abs(nC4))),cVAout
333   if(nC==0) write(iP2,"(i3,1x,' ',a1,1x,a20,\)") &
334     nC,char(Csymbol(abs(nC4))),cVAout
335   if(nC >0) write(iP2,"(i3,1x,'+',a1,1x,a20,\)") &
336     nC,char(Csymbol(abs(nC4))),cVAout
337   write(iP2,"(' ',\)")
338   do i=1,Ztot; write(iP2,"(i4,\)") iDiscord(i) ;enddo!i
339   write(iP2,"(' ',\)")
340   do i=1,Ztot; write(iP2,"(i5,\)") nOrder(i) ;enddo!i
341   write(iP2,*)
342   if(nC==0) write(iP2,*)
343   if((nC<18).and.(RubSize==3))cycle
344   if((nC<17).and.(RubSize==2))cycle
345   write(iP2,"(24x,'max: ',:,10i4)") &
346     EinW(Z(1:Ztot))%mTot
347   write(iP2,"(8x,'Pre-multiplies:',15x,'Post-multiplies:')")
348   write(iP2,"('Move# s:', ' 1 2 13D 3 4 14E',\ )")
349   write(iP2,"( 4x,' -1 -2 -13 -3 -4 -14' )")
350   write(iP2,"( 8x,' 5 6 15F 7 8 16G',\ )")
351   write(iP2,"( 4x,' -5 -6 -14 -7 -8 -16' )")
352   write(iP2,"( 8x,' 9 10A 17H 11B 12C 18I',\ )")
353   write(iP2,"( 4x,' -9 -10 -17 -11 -12 -18' )")
354   write(iP2,"(' --: -90 +90 180 +: -90 +90 180',\ )")
355   write(iP2,"(' --: -90 +90 180 +: -90 +90 180' )")
356 !The build environment hung on this line of code:
357 ! write(iP2,"('Z#2+Z#3 =',i3,',35x,' <-for all move#`s:')") iDscd18(0)
358 write(iP2,"(' X: ',3i4,3x,3i4,4x,3i4,3x,3i4,' :Z#1')") &
359 iDscd01( 1),iDscd01( 2),iDscd01( 13),iDscd01( 3),iDscd01( 4),iDscd01( 14)&
360 ,iDscd01(-1),iDscd01(-2),iDscd01(-13),iDscd01(-3),iDscd01(-4),iDscd01(-14)
361 write(iP2,"(' Y: ',3i4,3x,3i4,4x,3i4,3x,3i4,' now =',i5)") &
362 iDscd01( 5),iDscd01( 6),iDscd01( 15),iDscd01( 7),iDscd01( 8),iDscd01( 16)&
363 ,iDscd01(-5),iDscd01(-6),iDscd01(-15),iDscd01(-7),iDscd01(-8),iDscd01(-16)&
364 ,iDscd01( 0)
365 write(iP2,"(' Z: ',3i4,3x,3i4,4x,3i4,3x,3i4)") &
366 iDscd01( 9),iDscd01( 10),iDscd01( 17),iDscd01( 11),iDscd01( 12),iDscd01( 18)&
367 ,iDscd01(-9),iDscd01(-10),iDscd01(-17),iDscd01(-11),iDscd01(-12),iDscd01(-18)
368 if(RubSize==2) cycle
369 write(iP2,"( 3x,2(5x,11(' -'),4x,11(' -')) )") !dividing line
370 write(iP2,"(' X: ',3i4,3x,3i4,4x,3i4,3x,3i4,' :Z#2+Z#3')") &
371 iDscd18( 1),iDscd18( 2),iDscd18( 13),iDscd18( 3),iDscd18( 4),iDscd18( 14)&
372 ,iDscd18(-1),iDscd18(-2),iDscd18(-13),iDscd18(-3),iDscd18(-4),iDscd18(-14)
373 write(iP2,"(' Y: ',3i4,3x,3i4,4x,3i4,3x,3i4,' now =',i5)") &

```

```

373       write(ip2, '( z: ',3i4,3x,3i4,4x,3i4,3x,3i4, ' )' )
374       iDscd18( 5),iDscd18( 6),iDscd18( 15),iDscd18( 7),iDscd18( 8),iDscd18( 16)&
375       ,iDscd18(-5),iDscd18( -6),iDscd18(-15),iDscd18( -7),iDscd18( -8),iDscd18(-16)&
376       ,iDscd18( 0)
377       write(ip2,"( ' z: ',3i4,3x,3i4,4x,3i4,3x,3i4)")
378       iDscd18( 9),iDscd18( 10),iDscd18( 17),iDscd18( 11),iDscd18( 12),iDscd18( 18)&
379       ,iDscd18(-9),iDscd18(-10),iDscd18(-17),iDscd18(-11),iDscd18(-12),iDscd18(-18)
380       enddo!iPrint
381       !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
382
383       enddo!nC
384       if(ip<6)
385 30 write( 6,"(50('|'),/'input next move [-18:18] = ',\)")
386       read(5,*,err=30) nC
387       if(ip>6) &
388       write(ip,"(50('|'),/'Previous move [-18:18] = ',i3)") nC
389       call SaveOutFile
390       nC4 = nC
391
392       if(RubSize==2) then
393         select case(abs(nC))
394           case(3,4,7,8,11,12,14,16,18); goto 30
395         end select!(abs(nC))
396       endif!RubSize==2)
397
398
399       select case(nC)
400       case(19)
401         write( 6,"('cVAin = ',a20,' solved:')") cVAin
402         write( 6,"('          _____a_____a_____aa = quad')")
403         write( 6,"('          aaaaaaaaa_____a_____aa = & corners')")
404         write( 6,"('          _____aaaaaaaaaaaaaaaaaa = & edges')")
405         write( 6,"('          aaaaaaaaaaaaaaaaaaaaaaaaaaa = all')")
406         if(ip>6) then
407           write(ip,"('cVAin = ',a20,' solved:')") cVAin
408           write(ip,"('          _____a_____a_____aa = quad')")
409           write(ip,"('          aaaaaaaaa_____a_____aa = & corners')")
410           write(ip,"('          _____aaaaaaaaaaaaaaaaaa = & edges')")
411           write(ip,"('          aaaaaaaaaaaaaaaaaaaaaaaaaaa = all')")
412           write(ip,"(' = (/,' 8(i2,' ',')', '/')'") VAin(1: 8)
413           write(ip,"(' = (/,' 12(i2,' ',')', '/')'") VAin(9:20)
414         endif!(ip>6)
415
416       case( :-19,0,20:);
417       case(-18:-1,1:18)
418       !The concatenation-based emulator:
419         nC4m = nC4
420         call AllMoves(VtotL,VAout, VAin,nC4m, 0) !or ip
421
422       !The moves-based emulator:
423       !do nV = 1,VtotL
424       ! !Here's the beef: how a face rotation choice changes each cell's attitude:
425       !   VAout(nV) = AECAV(nC4,VAin(nV),nV)
426       ! ! ... & the cell's attitude character symbol is updated:
427       !   cVAout(nV:nV) = char(Asymbol(VAout(nV)))
428       !enddo!nV
429
430       Vain = VAout
431       do nV = 1,VtotL
432         cVAout(nV:nV) = char(Asymbol(VAin(nV)))
433       enddo!nV
434       cVAin = cVAout;

```

return

goto 30
return

goto 40 !goto 10

```

435   end select!(nC)
436
437   !Parse pre- & post- multiplication:
438 40 Euse2      = Euse
439   Euse2%Mtotvis = Euse%Mtotvis+1
440   select case(nC)
441     case(-18:-1) !Pre-multiplication - inserted before the first:
442       Euse2%MCvis(2:Euse2%Mtotvis) = Euse%MCvis(1:Euse%Mtotvis)
443       Euse2%MCvis(1) = nC4
444     case( 1: 18) !Post-multiplication - appended to the last:
445       Euse2%MCvis( Euse2%Mtotvis) = nC4
446   end select!(nC)
447   Euse = Euse2
448 !
449 End Subroutine solveScramble
450 !-----7 9
451
452 Subroutine AtoD(VtotL,VAin,iDiscord2,iP)
453 !2025.01.22.1920cst JMS- Zones evaluated are defined by Ein in a .nml file.
454 !   - Rubik's Cube 2x2x2 & 3x3x3 Discord evaluation.
455 !   The Output is iDiscord2(0:9)
456
457 !wisdom Zone#1 generation: forward-right-lower quad- cells
458 !01/21/2025  01:49 PM  select      6,359 MS-RC-R3Q-08161920-12.nml
459 !
460 !01/21/2025  01:56 PM  ZoneGen    25,935      R3Q-08161920-12-Summary.txt
461 !01/21/2025  01:56 PM  init        2,696      R3Q-08161920-12-Ein.bim
462 !01/21/2025  01:56 PM  Discord    253,440     R3Q-08161920-12-RrToDis.bim
463
464 !wisdom Zone#2 generation: 7 non=quad corners - also the 2x2x2 solution
465 !
466 !01/16/2025  04:08 PM  select      6,331 MS-RC-R2C-1234567-6.nml
467 !01/16/2025  05:31 PM  ZoneGen    33,180      R2C-1234567-6-Summary.txt
468 !01/16/2025  05:24 PM  init        2,696      R2C-1234567-6-Ein.bim
469 !01/16/2025  05:31 PM  Disc.    11,022,480     R2C-1234567-6-RrToDis.bim
470
471 !wisdom Zone#3 generation: 9 non=quad edges of a 3x3x3:
472 !
473 !01/17/2025  07:45 PM  select      7,968 MS-RC-R3E-091011121314151718-6.nml
474 !01/18/2025  02:46 AM  ZoneGen    43,300      R3E-091011121314151718-6-Summary.txt
475 !01/18/2025  12:33 AM  init        2,696      R3E-091011121314151718-6-Ein.bim
476 !01/18/2025  02:46 AM  Disc.    185,794,560     R3E-091011121314151718-6-RrToDis.bim
477
478 !This only operates on Vtot=8 or Vtot=20 all-vote strings.
479 !--Globals
480 use MS1Def ,only: Ur,Us,Ut,LExists,iAlloc
481 use MS2RCDf ,only: Vgzero,resetVg,Vg,Csymbol &
482 ,resetSR,SRrec,Erec,Ein,Einreset,RubSize,Ctot,Asymbol
483
484 use MS2RCDf ,only: Ztot,Z,EinW ,cZrFilename,nOrder &
485 ,Zone1,Zone2,Zone3,Zone4 &
486 ,Zone5,Zone6,Zone7,Zone8,Zone9,ZonePnt
487 !--End Globals
488 implicit none
489 !--Arguments
490 integer(1)::VtotL
491 integer(4)::VAin(20) !The votes [1:VtotL]
492 !4-groups
493 integer(4)::iDiscord2(0:9)!Function results (i.e.:output:returned values)
494 ! (0) is the sum, =-1:all invalid
495
496 integer(4)::iP !write enable>5: write(iP,...)

```



```

497  !--Internals
498  integer(4)::Init
499  integer(8)::n8,H8a,Indexer
500  type(Erec) ::EinSave
501  type(SRrec)::Srin
502  type(SRrec)::Sr1      !Indexer`s Sr argument
503  character(60)::FileIn
504  integer(4)::i,n,nin,nU,nZi,nZ,Vtw
505  integer(4)::iDiscord(0:9)
506  !--EndDefs-----
507  EinSave = Ein
508  Ein     = EinReset
509  SRin    = resetSR
510  if(Init==0) then
511     resetVg = Vgzero
512     write(Us,"('/AtoD():  initialization & data import:')")
513     if(iP>5) &
514     write(iP,"('/AtoD():  initialization & data import:  @L484'")
515     call PrintSRrec(Srin,iP,'Input to AtoD')
516
517     EinW          = EinReset
518     Ztot          = EinSave%Ztot
519     Z(1:Ztot)    = EinSave%Z(1:Ztot)
520     cZrFilename(Z(1:Ztot)) = EinSave%cZrFilename(Z(1:Ztot))
521     iDiscord     = 0
522     nOrder      = 0
523
524     if(iP>5) write(iP,"('Ztot=',i2,2x,9i2)") Ztot,Z
525
526     do nZi = 1,Ztot; nZ = Z(nZi)
527        if(iP>5) write(iP,"('nZ=',i2,2x,a60)") nZ,EinSave%cZrFilename(nZ)
528        inquire(file = EinSave%cZrFilename(nZ), exist = LExists)
529        if(LExists)then
530           open(Ur, file = EinSave%cZrFilename(nZ) , action='read' &
531              , access='direct', recl= 2696      , status='old',err = 4)
532           read(Ur, rec = 1, err = 4) EinW(nZ)
533           close(Ur);
534           call PrintErec(EinW(nZ),iP,'AtoD: EinW import @L504')      ;goto 5
535  4         pause "Error reading the file. Press enter to continue."
536           close(Ur)
537        else
538           write(Us,"('Did not find: ',a60)") EinSave%cZrFilename(nZ)
539           cycle
540        endif!LExists
541  5         continue
542        if(iP>5) write(iP,"('Allocating ',i12,' bytes for Discord data:')") &
543           EinW(nZ)%Zonebimsize
544        resetVg%CVgFileName(nZ) = EinW(nZ)%cRrtoDisBinary
545        select case(nZ)
546        case(1);allocate(Zone1(EinW(nZ)%Zonebimsize),stat=iAlloc)
547           if(iAlloc==0)then; Zone1=-2; ZonePnt=>Zone1; endif
548        case(2);allocate(Zone2(EinW(nZ)%Zonebimsize),stat=iAlloc)
549           if(iAlloc==0)then; Zone2=-2; ZonePnt=>Zone2; endif
550        case(3);allocate(Zone3(EinW(nZ)%Zonebimsize),stat=iAlloc)
551           if(iAlloc==0)then; Zone3=-2; ZonePnt=>Zone3; endif
552        case(4);allocate(Zone4(EinW(nZ)%Zonebimsize),stat=iAlloc)
553           if(iAlloc==0)then; Zone4=-2; ZonePnt=>Zone4; endif
554        case(5);allocate(Zone5(EinW(nZ)%Zonebimsize),stat=iAlloc)
555           if(iAlloc==0)then; Zone5=-2; ZonePnt=>Zone5; endif
556        case(6);allocate(Zone6(EinW(nZ)%Zonebimsize),stat=iAlloc)
557           if(iAlloc==0)then; Zone6=-2; ZonePnt=>Zone6; endif
558        case(7);allocate(Zone7(EinW(nZ)%Zonebimsize),stat=iAlloc)

```



```

621   do nZi = 1,Ztot; nZ = Z(nZi)
622   !!!!!if((VtotL==8).and.(nV/=7)) cycle           !This may introduce a bug.
623   if(iP>5) &
624   write(iP,"(/,'>>>> Processing Discord mask#',i2,' >>>> @L594',/)" nZ
625   Ein      = EinReset
626   Vg%iDiscord(nZ) = -1
627
628   if(iP>5) then
629     write(iP,"('      '      These cells are being evaluated...')")
630     write(iP,"('nz#',i1,' Vg%Maskneed =', 8i3,'|',i2,11i3)" &
631           nZ,          resetVg%Maskneed(1:20,nZ)
632   endif!(iP>5)
633
634   do nU = 1,VtotL
635     if( resetVg%Maskneed(nU,nZ)==0) cycle
636     if((resetVg%Maskneed(nU,nZ)>0).and.(Vg%VA(nU)>0)) cycle
637     goto 100
638   enddo!nV
639
640   if(iP>5) then
641     write(iP,"('      '      These cells must already be solved...')")
642     write(iP,"('      '      Vg%Maskfor1 =', 8i3,'|',i2,11i3)" &
643           resetVg%Maskfor1(1:20,nZ)
644   endif!(iP>5)
645
646   do nU = 1,VtotL
647     if( resetVg%Maskfor1(nU,nZ)==0) cycle
648     if((resetVg%Maskfor1(nU,nZ)>0).and.(Vg%VA(nU)==1)) cycle
649     goto 100
650   enddo!nV
651
652   Sr1      = resetSR
653   Sr1%Vtot = VtotL
654
655   Ein%nZ      = nZ
656   Ein%Vtot    = EinW(nZ)%Vtot
657   Vtw        = Ein%Vtot
658   Ein%V(1:Vtw) = EinW(nZ)%V(1:Vtw)
659   Ein%Ctot    = EinW(nZ)%Ctot
660   Sr1%Vtot    = Vtw
661   !Sr1%VA(1:Vtw) = Ein%VA(1:Vtw)
662   Sr1%VA(1:Vtw) = VAin(Ein%V(1:Vtw))
663
664   RubSize    = EinW(nZ)%RubSize
665   Ctot       = Ein%Ctot
666
667   !Compute the index holding nV`s vote:
668   do i = 1,Sr1%Vtot
669     Sr1%cVAsymb(i:i) = char(Asymbol(Sr1%VA(i)))
670   enddo!i
671   if(iP>5) write(iP,*)
672   call PrintSRrec(Sr1,iP,'Indexer input @L642')
673
674   nin = 0
675   H8a = Indexer(nin,Sr1, 0) !<- The Lookup table address,
676   ! a recursive function.
677
678   if(iP>5) write(iP,"('H8a = ',i12,5x,40('|'))") H8a
679   !call PrintSRrec(Sr1,iP,'Indexer result @L649')
680
681   if(H8a>0_8) then
682     !Record the Discord as valid:

```

```

683     select case(nZ)
684         case(1); Vg%iDiscord(1) = Zone1(H8a)
685         case(2); Vg%iDiscord(2) = Zone2(H8a)
686         case(3); Vg%iDiscord(3) = Zone3(H8a)
687         case(4); Vg%iDiscord(4) = Zone4(H8a)
688         case(5); Vg%iDiscord(5) = Zone5(H8a)
689         case(6); Vg%iDiscord(6) = Zone6(H8a)
690         case(7); Vg%iDiscord(7) = Zone7(H8a)
691         case(8); Vg%iDiscord(8) = Zone8(H8a)
692         case(9); Vg%iDiscord(9) = Zone9(H8a)
693     end select!(nV)
694     iDiscord2(nZ) = Vg%iDiscord(nZ)
695     if(iP>5) write(iP,"('Dis = ',i12,5x,40('|'))") iDiscord2(nZ)
696     else
697         !Record the Discord as invalid:
698         Vg%iDiscord(nZ) = -1; iDiscord2(nZ) = Vg%iDiscord(nZ)
699     endif!(H8a>0_8)
700
701     call PrintSRrec(Sr1,iP,'Indexer result @L671')
702
703     !Report results:
704     if(iP>5) then
705         write(iP,"('nZ      =',i3,' Utot= ',i2,' U( 1:Utot) =',:,9i3)") &
706             nZ,          Sr1%Vtot,      Ein%V(1:Sr1%Vtot)
707         write(iP,"('***iDiscord(nZ) =',i3,' UA(1:Utot) =',:,9i3)") &
708             Vg%iDiscord(nZ),Sr1%VA(1:Sr1%Vtot)
709         write(iP,"('  H8a =',i22)") H8a
710     endif!(iP>5)
711 100    continue
712     enddo!nzi
713
714     !Compute iDiscord(0)
715     if(VtotL==20) then
716         iDiscord2(0) = - 1
717         if(iDiscord2(2)>=0) iDiscord2(0) = iDiscord2(2)
718         if(iDiscord2(3)>=0) iDiscord2(0) = iDiscord2(0) + iDiscord2(3)
719     else
720         iDiscord2(0) = iDiscord2(1)
721     endif!(VtotL==20)
722     iDiscord = iDiscord2
723     if(iP>5) &
724     write(iP,"('/AtoD argument iDiscord(0:3) = ',10i3,' @L694'))" iDiscord2
725     if(iP>5) &
726     write(iP,"('>>>> Discord mask computations done >>>> @L696',/)")
727
728     Ein = EinSave; return
729 End Subroutine AtoD
730 !-----7 9
731
732

```