

```
1  //////////////////////////////////////
2  // FICHERO    : solver.tol
3  // PROPOSITO : 3D Square Solver.
4  //////////////////////////////////////
5
6
7  //////////////////////////////////////
8  // CONSTANTES
9  //////////////////////////////////////
10
11 Set  PieAll =
12 [[
13     ".r."+
14     "M1v"+
15     ".A.",
16
17     ".a."+
18     "A2v"+
19     ".R.",
20
21     ".R."+
22     "V3m"+
23     ".a.",
24
25     ".R."+
26     "M4v"+
27     ".A.",
28
29     ".R."+
30     "M5r"+
31     ".V.",
32
33     ".m."+
34     "A6R"+
35     ".V.",
36
37     ".a."+
38     "m7r"+
39     ".v.",
40
41     ".r."+
42     "v8M"+
43     ".A.",
44
45     ".v."+
46     "M9m"+
47     ".A."
48 ]];
49
50
51 //////////////////////////////////////
52 // FUNCIONES INTERNAS
53 //////////////////////////////////////
54
55 //////////////////////////////////////
56 Real EvalWhile(Set set, Code fun)
57 //////////////////////////////////////
58 {
59     Real num = 1;           // Contador de elementos
60     Real max = Card(set);  // Maximo del set
61
62     Real While(LE(num, max), // Hasta fin del set
63     {
64         Anything ele = set[num];
65         Real     res = fun(ele);
66         Real     (num:= Copy(num) + 1);
67
68         TRUE
```

```

69     });
70     max
71 };
72
73
74 ///////////////////////////////////////////////////////////////////
75 // PIE(za) FUNCIONES
76 ///////////////////////////////////////////////////////////////////
77
78 ///////////////////////////////////////////////////////////////////
79 Real PieVer(Text pie) // Pinta una pieza
80 ///////////////////////////////////////////////////////////////////
81 {
82     Text WriteLn(Sub(pie, 1, 3)+"\n"+
83                 Sub(pie, 4, 6)+"\n"+
84                 Sub(pie, 7, 9)+"\n"+
85                 "---");
86     TRUE
87 };
88
89
90 ///////////////////////////////////////////////////////////////////
91 Text PieRotDer(Text pie) // Rota a la derecha una pieza
92 ///////////////////////////////////////////////////////////////////
93 {
94     Text nor = Sub(pie, 2, 2);
95     Text oes = Sub(pie, 4, 4);
96     Text est = Sub(pie, 6, 6);
97     Text sur = Sub(pie, 8, 8);
98
99     Text cen = Sub(pie, 5, 5); // Conserva el numero de pieza
100
101     Text rot = "."+oes+"."+
102              sur+cen+nor+
103              "."+est+".";
104     rot
105 };
106
107
108 ///////////////////////////////////////////////////////////////////
109 Text PieRot180(Text pie) // Rota 180° una pieza
110 ///////////////////////////////////////////////////////////////////
111 { PieRotDer(PieRotDer(pie)) };
112
113
114 ///////////////////////////////////////////////////////////////////
115 Text PieRotIzq(Text pie) // Rota a la izquierda una pieza
116 ///////////////////////////////////////////////////////////////////
117 { PieRotDer(PieRotDer(PieRotDer(pie))) };
118
119
120 ///////////////////////////////////////////////////////////////////
121 Real PieCmbNor(Text uno, Text dos) // Si uno combina por el norte con dos
122 ///////////////////////////////////////////////////////////////////
123 {
124     Text nor = Sub(uno, 2, 2);
125     Text sur = Sub(dos, 8, 8);
126     EQ(32, Abs(ASCII(nor)-ASCII(sur))) // (a-A) = (m-M) = (r-R) = (v-V) = 32
127 };
128
129
130 ///////////////////////////////////////////////////////////////////
131 Real PieCmbEst(Text uno, Text dos) // Si uno combina por el este con dos
132 ///////////////////////////////////////////////////////////////////
133 {
134     Text est = Sub(uno, 6, 6);
135     Text oes = Sub(dos, 4, 4);
136     EQ(32, Abs(ASCII(est)-ASCII(oes))) // (a-A) = (m-M) = (r-R) = (v-V) = 32

```

```

137 };
138
139
140 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
141 Real PieCmbSur(Text uno, Text dos) // Si uno combina por el sur con dos
142 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
143 {
144   Text sur = Sub(uno, 8, 8);
145   Text nor = Sub(dos, 2, 2);
146   EQ(32, Abs(ASCII(sur)-ASCII(nor))) // (a-A) = (m-M) = (r-R) = (v-V) = 32
147 };
148
149
150 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
151 Real PieCmbOes(Text uno, Text dos) // Si uno combina por el oeste con dos
152 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
153 {
154   Text oes = Sub(uno, 4, 4);
155   Text est = Sub(dos, 6, 6);
156   EQ(32, Abs(ASCII(oes)-ASCII(est))) // (a-A) = (m-M) = (r-R) = (v-V) = 32
157 };
158
159
160 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
161 // VAR(iaciones) FUNCIONES
162 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
163
164 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
165 Set VarAll(Text pie) // Retorna la pieza original y sus 3 giros
166 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
167 {
168   SetOfText(pie, PieRotDer(pie), PieRot180(pie), PieRotIzq(pie))
169 };
170
171
172 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
173 // SOL(uciones) FUNCIONES
174 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
175
176 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
177 Real SolVer(Set sol) // Ver una solucion completa
178 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
179 {
180   Text Write("\n\nSOLUCION:\n\n"+
181     " "+"----"           "+" "+"----"           "+" "+"----"           "+" \n");
182   Set cic = For(0, 2, Real(Real lin))
183   {
184     Real li3 = lin * 3;
185     Text Write(
186       " |+Sub(sol[li3+1],1,3)+"|"+Sub(sol[li3+2],1,3)+"|"+Sub(sol[li3+3],1,3)+"|\n"+
187       " |+Sub(sol[li3+1],4,6)+"|"+Sub(sol[li3+2],4,6)+"|"+Sub(sol[li3+3],4,6)+"|\n"+
188       " |+Sub(sol[li3+1],7,9)+"|"+Sub(sol[li3+2],7,9)+"|"+Sub(sol[li3+3],7,9)+"|\n"+
189       " "+"----"           "+" "+"----"           "+" "+"----"           "+" \n");
190     li3
191   });
192   Text WriteLn("");
193   Card(cic)
194 };
195
196 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
197 Real SolBue(Set sol) // Mira si una solucion es buena
198 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
199 {
200   Real crd = Card(sol);
201
202   If( LE(crd, 1),      TRUE,      // Con 0 o 1 piezas siempre esta bien
203     If( LE(crd, 3),    PieCmbEst(sol[crd-1], sol[crd]), // El 2 y el 3
204     If(Or(EQ(crd, 4),

```

```
205      EQ(crd, 7)),      PieCmbSur(sol[crd-3], sol[crd]), // El 4 y 7
206      And(PieCmbSur(sol[crd-3], sol[crd]), // El 5, 6, 8 y 9
207      PieCmbEst(sol[crd-1], sol[crd])))
208 };
209
210
211 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
212 Real SolCua(Set entSol, Set entPie) // Resuelve
213 ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
214 {
215     Real crdSol = Card(entSol);
216     Text Write(FormatReal(crdSol,"%0.0lf"));
217
218     If(EQ(crdSol, 9), SolVer(entSol), // Ha encontrado una solucion
219     { // Busca soluciones
220         EvalWhile(entPie, Real(Text unoPie)
221         {
222             Set salPie = entPie - [[unoPie]];
223             Set varPie = VarAll(unoPie);
224             EvalWhile(varPie, Real(Text unoVar)
225             {
226                 Set salSol = entSol << [[ unoVar ]];
227                 If(SolBue(salSol), SolCua(salSol, salPie), FALSE)
228             })
229         })
230     })
231 };
232
233 Text WriteLn("\nINICIO BÚSQUEDA:\n");
234 Real SolCua(Empty, PieAll);
235 Text WriteLn("\nFIN BÚSQUEDA:\n");
236
237
238 /*
239     Text WriteLn(PieAll[1]);
240
241     Text PieVer(PieAll[1]);
242     Text PieVer(PieRotDer(PieAll[1]));
243     Text PieVer(PieRot180(PieAll[1]));
244     Text PieVer(PieRotIzq(PieAll[1]));
245     Real tst001 = PieCmbOes(PieAll[1], PieAll[3]);
246
247 */
248
249
250
251
252
253
254
255
256
257
258
```

```

1
2 C:\asc\App\3d.sqr>"C:\asc\Bin\Tolbase-1.1.5\bin\tol.exe" -i -v sqrsolver.tol
3
4 Initializing TOL OIS: versión implementada: 02.03
5 INICIO BÚSQUEDA:
6
7 012345453456774567345677674534545234564534545423445674534454344456734456745123454456773454345
445345445677234434456773434453445677234443443444234434434434423454343454123444567345445344
445345444345444523444344434444344443444423444567774344456777434544543454443456774442344456787
743454443456777444567877412344443444344442345677745543454344567754234543455454345541234567745
456345456774545634567456777453456786745677567784567877677734545675674542345456767734564567673
4545234567767744344567767744344434443444567674412344345445344534542345445344534567744534544567
72345434567756777345434554234434443443442344345434567787412344454344444344445434444234444543
444434444541234345234345674345345423453453455345675677345523453412344544453445674456234544445
345674445623444544344454544344454442345674445677454563444423445443454443456774454434544541234444
344442344454344443444523444453454445345674445345444567876773454442345456744345454412345443454
534544523454554523445674434445434445442345453444534445453445434453445452344434434442344544344412
344454345445434444543454444234545454453454545454523454454344445674344564543445442345454545434
5456564545677745674345454543456774554567745423444434444344444344444234454443445644412345453
456777456734567745345452345454345678453454542344567453445434445673445677451234443443444234534
344344523444567434434445674344423456774453443454451234567874434456774434456743444344454423444567
853444456785344456344434444567567234443444344344423444434444344412344443444434443444423454543
456777455434543445677542345434554543456777541234544567887434544567887434567456787674443454544
34567784567789

```

```

8
9 SOLUCION:
10
11 --- --- ---
12 |.R|.A|.M.|
13 |M5r|R2a|A9v|
14 |.V|.v|.m.|
15 --- --- ---
16 |.v|.V|.M.|
17 |R4A|a3R|r8A|
18 |.M|.m|.v.|
19 --- --- ---
20 |.m|.M|.V.|
21 |v7a|A1r|R6A|
22 |.r|.v|.m.|
23 --- --- ---
24
25 445234545454543444567774567743454554544345674556456776745456774345455454543456774554567744523
44454344444344456778678423444443445444434454441234445534445676753444452344454545673444545673444
54553444545456756723444344434544412344344434443443444234456753444567675344455344456344344456
756723443443443442344434443441234443444344434442344434442344453444534443444234443444344434434
4434423445434545434544345445234544534456773456787878412344454344444344456789

```

```

26
27 SOLUCION:
28
29 --- --- ---
30 |.m|.v|.r.|
31 |A6R|r1A|a7v|
32 |.V|.M|.m.|
33 --- --- ---
34 |.v|.m|.M.|
35 |A8r|R3a|A4R|
36 |.M|.V|.v.|
37 --- --- ---
38 |.m|.v|.V.|
39 |v9A|a2R|r5M|
40 |.M|.A|.R.|
41 --- --- ---
42
43 64234544545445344444534544545234444545344445453444445453444434444523454544344454443454545123444
563445675234444563444567523444344434442344543456778456787234445534444553444556773444123444434
444344444344442345454453445644344454445344445234454544534454412344345453454542343455234456344
534454523454534345453454534545234434434442345453443441234544567877345445677734543454453454456
777234456777344344567773434453445677723444344234434434434434423454341234445677743444567774345

```

445434544434567744423444563454453454423454454345443456777445677742344544534456777434454456777  
234445456344445634445678787453444453444456778767878767878782344544344443444541234444434456744  
45673445674456723444443456789

44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86

SOLUCION:

--- --- ---  
.v	.R	.M
r7m	M4v	V5R
.a	.A	.r
--- --- ---  
.A	.a	.R
v1M	m3V	v2A
.r	.R	.a
--- --- ---  
.R	.r	.A
m6V	v8M	m9M
.A	.A	.v
--- --- ---

444456734567744456723444544344567445443445674544344544423456774456745567456767345677445674567  
456734444234454434544434544544345677744567741234444534544453456744453454445678767734544423444  
534445234444345444344443456744434544412344567345676734453434452344434434443444234343443443  
442344534443441234454456778344544567783444567863445423454456773456775677443455445677344567634  
554234544543445434544234544543454454345445434544345445234555455543455567743455567774123445454  
567344545673445677455344545456756723445454534456777454523454443454543454543454544123445667564  
344344566742344343442344344344123443444344423443444344423445673445673444344452344345678784543  
456789

SOLUCION:

--- --- ---  
.v	.A	.A
M9m	M8v	V6m
.A	.r	.R
--- --- ---  
.a	.R	.r
A2v	V3m	M1v
.R	.a	.A
--- --- ---  
.r	.A	.a
R5V	v4M	m7r
.M	.R	.v
--- --- ---

74567788412344434544345674434544234434443442344453454453444534567441234453445344534443444523445  
677534456776777534456775344567734455234544567787344345445344534541234443444344344423454534567  
77455345344567775677234534554534567775

FIN BÚSQUEDA:

Ha sido incluido el fichero C:/asc/App/3d.sqr/sqrsolver.tol [24.937 sec.]

C:\asc\App\3d.sqr>copy "C:\Documents and Settings\Usuario\Datos de programa\tol\syslog\start.log" start.log

1 archivos copiados.