

MEMORIAL DAY PROGRAMMING RACE

The puzzle shown in the figure consists of nine blocks which are to be arranged in a three by three square array so that the line segments crossing a block meet the line segments on adjacent blocks. A solution is shown. Write a program to print the number of solutions not counting as different solutions obtainable from each other by rotating the whole puzzle,

The winner will be the person who completes the problem with the lowest composite time score determined as follows:

$$\begin{aligned} \text{composite-time} = & (\text{time from receipt of problem to time of} \\ & \text{start of use of the console}) + \\ & 2.0 * (\text{time from start of use of console to time of} \\ & \text{completion of creation of first version of program - this time ends} \\ & \text{with completion of first edit}) + \\ & 4.0 * (\text{time from completion of first edit to successful} \\ & \text{completion of the problem}) + \\ & 20.0 * (\text{total computer time charged by the system to} \\ & \text{the user}) \end{aligned}$$

Any user whom the system charges more than 15 minutes will be disqualified.

SAIL, i.e., ALGOL, is an appropriate language for the problem.

The problem has already been done twice. In one case, the elapsed time was eight hours and the run time 20 minutes. In the other case, the elapsed time was 4 hours and the run time 27 seconds. We expect the winning elapsed time to be less than 2 hours, perhaps much less.

In case of a system crash or other emergency, the Judges will make appropriate ad hoc rules.

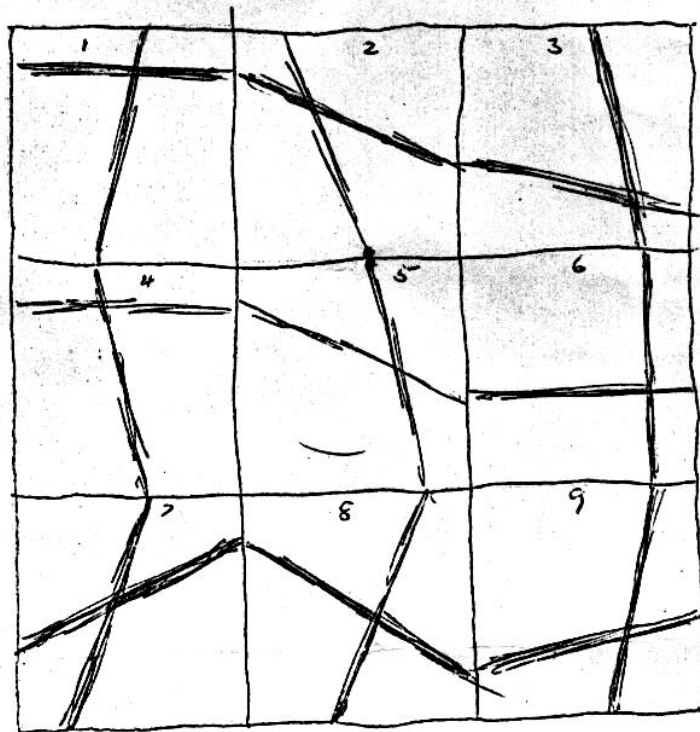
YOU MUST KEEP TRACK OF YOUR OWN TIMES IN EACH OF THE FOUR CATEGORIES.

When you have an answer call J. McCarthy at 4430 or 4971. If the answer is wrong, you may continue, but a penalty of one hour composite time will be charged for each wrong answer.

For the purposes of competition among machine language programmers, the last term in the composite time is changed to

$$50.0 * (\text{computer time charged for the best of three runs of the final problem})$$

Comparison of the computer time charged by the 360/67 with that charged by the PDP-10 will be made according to the results of a calibration run with the same program written in SAIL and ALGOL W, running time only.



There are just four levels of line on each block. Apologies for xerox troubles.

STANFORD 4 05/31/71 14:17:36

NAME? knuth

RECORD? ****

COMMAND ? set terse

? show time

05/31/71 14:17:57

ELAPSED TIME = 00:00:05

EDITING TIME = 0.00 SECONDS

COMPUTE TIME = 0.01 SECONDS

MEMORY USAGE = 0.01 PAGE-SECONDS

I/O ACTIVITY = 0 UNITS

? use env

VOLUME? sys14

? set vol sys14

? list

0.001 //DEKJOB JOB (K601,512,0.3,3),'KNUTH'

0.002 //JOB LIB DD DSN=SYS2.PROGLIB,DISP=OLD

0.003 // EXEC XALGOLW

0.004 //ALGOLW.SYSIN DD *

0.005 %ALGOL 0:15,3000

0.006 \$DEBUG,4

999.9 %EOF

999.901 /*

? collect 1

1. ? begin comment race problem;

2. ? integer array p(0::999),a(0::10),tally(0::10);

3. ? integer array c1,c2,c3,c4(1::8);

4. ? logical array easy(1::8);

5. ? procedure find(integer value k);

6. ? begin integer t1,t2;

7. ? t1:=5-p(a(c1(k))+c2(k));

8. ? t2:=5-p(a(c3(k))+c4(k));

9. ? for j:=100 step 100 until 900 do

10. ? begin if p(j)=0 then

11. ? begin for jj:=j+1 until j+4 do

12. ? if (p(jj)=t1) and (easy(k) or (p(jj+1)=t2))

13. ? then begin tally(k):=tally(k)+1;

14. ? if k<8 then begin p(j):=1;a(k):=jj; find(k+1); p(j):=0 end;

15. ? else if tally(8)<4 then write(a(0),a(1),a(2),a(3),a(4),a(5),a(6),a(7),a(8));

LINE NO. 15. CONTAINS 78 CHARACTERS.

16. ? a(6),a(7),a(8));

17. ? end end loopjj;

STANFORD - X ALGOL W - 20NOV70 XDBG

COMPILATION DIAGNOSTICS

ERROR 2019 NEAR COORDINATE 0001 - ", " CANNOT FOLLOW "<ARRAY DCL>" HERE
CURRENT CONTEXT IS ") , "

ERROR 2002 NEAR COORDINATE 0007 - "A" IS UNDEFINED
CURRENT CONTEXT IS " A ("

ERROR 2016 NEAR COORDINATE 0007 - SIMPLE TYPE ERROR NO. 53
CURRENT CONTEXT IS " (C1 "

ERROR 2003 NEAR COORDINATE 0007 - SYNTAX ERROR
CURRENT CONTEXT IS ") + "

ERROR 2002 NEAR COORDINATE 0008 - "A" IS UNDEFINED
CURRENT CONTEXT IS " A ("

ERROR 2016 NEAR COORDINATE 0008 - SIMPLE TYPE ERROR NO. 53
CURRENT CONTEXT IS " (C3 "

ERROR 2003 NEAR COORDINATE 0008 - SYNTAX ERROR
CURRENT CONTEXT IS ") + "

ERROR 2002 NEAR COORDINATE 0012 - "TALLY" IS UNDEFINED
CURRENT CONTEXT IS " TALLY ("

ERROR 2016 NEAR COORDINATE 0012 - SIMPLE TYPE ERROR NO. 53
CURRENT CONTEXT IS " (K "

ERROR 2003 NEAR COORDINATE 0012 - SYNTAX ERROR
CURRENT CONTEXT IS ") := "

ERROR 2002 NEAR COORDINATE 0015 - "A" IS UNDEFINED
CURRENT CONTEXT IS " A ("

ERROR 2016 NEAR COORDINATE 0015 - SIMPLE TYPE ERROR NO. 53
CURRENT CONTEXT IS " (K "

ERROR 2003 NEAR COORDINATE 0015 - SYNTAX ERROR
CURRENT CONTEXT IS ") := "

```

? delete 2,
? run q hold
614 IS YOUR JOB NUMBER
? save source scr
"SOURCE" SCRATCHED & SAVED ON SYS14
? fetch 614 clear
DEKJOB (614) IS AWAITING EXEC , PRIO SS, CLASS Q
BACKLOGGED 3 JOB(S) 2.5 MINS.
? show time
05/31/71 14:38:20
ELAPSED TIME = 00:20:28
EDITING TIME = 1.22 SECONDS
COMPUTE TIME = 0.01 SECONDS
MEMORY USAGE = 0.01 PAGE-SECONDS
I/O ACTIVITY = 0 UNITS
? fetch 614
DEKJOB (614) IS AWAITING EXEC , PRIO SS, CLASS Q
BACKLOGGED 3 JOB(S) 2.5 MINS.
? set priority
SET: ILLEGAL.
? show time
05/31/71 14:43:11
ELAPSED TIME = 00:25:18
EDITING TIME = 1.23 SECONDS
COMPUTE TIME = 0.06 SECONDS
MEMORY USAGE = 0.06 PAGE-SECONDS
I/O ACTIVITY = 0 UNITS
? fetch 614
? listrl
329. 0041 | END
? print 614
DEKJOB (614) WILL BE PRINTED
? ***
ARE YOU STILL THERE?
? use source clear
? modify "(k)+1"
13. THEN BEGIN TALLY(K):=TALLY(K)+1;
ALTERS ?
13. THEN BEGIN TALLY(K):=TALLY(K)+1; ia(k):=jj;
ALTERS ? A(K):=JJ;
? dh "a(k):=jj;" to "" in 14

```

=> TRACING (MAIN):

```
0032      2.--|      EASY(K) := FALSE  
                K = 5; EASY(5) := FALSE; ...
```

=> TRACING (MAIN):

```
0034      1.--|      FOR I := 10 STEP 10 UNTIL 90 DO  
                I := 10;  
0035      1.--|      A(0) := I + 1  
                I = 10; A(0) := 11;  
0036      1.--|      P(I) := 1  
                I = 10; P(10) := 1;  
0037      1.--|      FIND(1)  
                -> FIND;
```

=> TRACING FIND:

```
0008      1.--|      <PARAMETER ASSIGNMENT>  
                K := #; # = 1; K' := 1;  
0009      1.--|      T1 := 5 - P(A(C1(K)) + C2(K))  
                K' = 1; C1(1) = 0; A(0) = 11; K' = 1; C2(  
0010      1.--|      T2 := 5 - P(A(C3(K)) + C4(K))  
                K' = 1; C3(1) = 0; A(0) = 11; K' = 1; C4(  
0011      1.--|      FOR J := 10 STEP 10 UNTIL 90 DO  
                J := 10;  
0012      1.--|      IF P(J) = 0 THEN  
                J = 10; P(10) = 1; * = FALSE;  
0011      |      (FOR J)  
                J := 20;  
0012      2.--|      IF P(J) = 0 THEN  
                J = 20; P(20) = ?; * = FALSE;  
0011      |      (FOR J)  
                J := 30; ...
```

```
0038      1.--|      P(I) := 0  
                I = 10; P(10) := 0;  
0034      |      (FOR I)  
                I := 20;  
0035      2.--|      A(0) := I + 1  
                I = 20; A(0) := 21;  
0036      2.--|      P(I) := 1  
                I = 20; P(20) := 1;  
0037      2.--|      FIND(1)  
                -> FIND;
```

=> EXECUTION FLOW SUMMARY

```

0000      1.--|      BEGIN
0001      |      INTEGER ARRAY P(0 :: 99);
0002      |      INTEGER ARRAY A(0 :: 9);
0003      |      INTEGER ARRAY TALLY(0 :: 9);
0004      |      INTEGER ARRAY C1, C2, C3, C4(1 :: 8);
0005      |      LOGICAL ARRAY EASY(1 :: 8);

0006      |      PROCEDURE FIND(INTEGER VALUE K);
0007      |      BEGIN
0008      |      |      INTEGER T1, T2;
0009      |      |      T1 := 5 - P(A(C1(K)) + C2(K));      T2 := 5 - P(A(C3(K)) + C4(K));
0010      |      |      FOR J := 10 STEP 10 UNTIL 90 DO
0011      |      |      |      BEGIN
0012      |      |      |      |      IF P(J) = 0 THEN
0013      |      |      |      |      |      BEGIN
0014      |      |      |      |      |      |      FOR JJ := J + 1 UNTIL J + 4 DO
0015      |      |      |      |      |      |      |      IF (P(JJ) = T1) AND (EASY(K) OR (P(JJ + 1) = T2)) THEN
0016      |      |      |      |      |      |      |      |      BEGIN TALLY(K) := TALLY(K) + 1;      A(K) := JJ;
0017      |      |      |      |      |      |      |      IF K < 8 THEN
0018      |      |      |      |      |      |      |      |      BEGIN P(J) := 1;      FIND(K + 1);
0019      |      |      |      |      |      |      |      |      P(J) := 0
0020      |      |      |      |      |      |      |      |      END ELSE
0021      |      |      |      |      |      |      |      |      IF TALLY(8) < 4 THEN
0022      |      |      |      |      |      |      |      |      |      WRITE(A(0), A(1), A(2), A(3), A(4), A(5), A(6), A(7), A(8));
0023      |      |      |      |      |      |      |      |      END
0024      |      |      |      |      |      |      |      |      END
0025      |      |      |      |      |      |      |      |      END
0026      |      |      |      |      |      |      |      |      END
0027      |      |      |      |      |      |      |      |      END
0028      |      |      |      |      |      |      |      |      END
0029      |      |      |      |      |      |      |      |      END
0030      |      |      |      |      |      |      |      |      END
0031      |      |      |      |      |      |      |      |      END
0032      |      |      |      |      |      |      |      |      END
0033      |      |      |      |      |      |      |      |      END
0034      |      |      |      |      |      |      |      |      END
0035      |      |      |      |      |      |      |      |      END
0036      |      |      |      |      |      |      |      |      END
0037      |      |      |      |      |      |      |      |      END
0038      |      |      |      |      |      |      |      |      END
0039      |      |      |      |      |      |      |      |      END
0040      |      |      |      |      |      |      |      |      END
0041      |      |      |      |      |      |      |      |      END
0042      |      |      |      |      |      |      |      |      END

```

Chronology

1:09 looked at problem

Decided to place piece in middle, oriented as given, then place other 8 cyclically

Did Monte Carlo experiment indicating roughly 91000 nodes in the search tree

Drafted program

2:17 Left office to look for a terminal

2:18 Logged on

2:34 Submitted first job to batch [JOB 602, no backlog; 0.005 minutes execution

Fixed syntax error (multiple array declarations)

0.33 sec compile
0.0 sec exec.

2:38 Submitted second job [JOB 614, backlogged.

Tried to find out how to change priority; never did find out

2:41 Job begins execution. changed 0.07 minutes compute time.

0.44 sec compile
3.46 sec exec.

2:43 I discover that the job was executed while I was inquiring about priority.

2:45 Begin reading listing of JOB 614. Two errors (gave only ^{the 161} solutions with piece 9 in center)

2:57 Corrected the errors, submitted third job [JOB 650, no backlog. 0.26 min changed 0.43 sec comp 15.00 sec exec

(Forgot to increase the expected run time from debug parameters. Noticed that 15 sec was enough for about half the computation. Checked the first solution found.)

3:05 Submitted fourth job [JOB 667, backlogged until 3:07:34. 0.45 min changed 0.44 sec comp 26.71 sec exec

3:09 logoff changed 0.13 sec compile time 2.80 sec edit time 0.13 page/seconds memory.

3:10 call McCarthy. Total computer time 45.3 sec execution
1.8 sec edit and compile.

47.1

16

Composite time $1 \times 69 = 69$

$2 \times 16 = 32$

$4 \times 35 = 140$

$20 \times 0.75 = 15$

257 minutes.