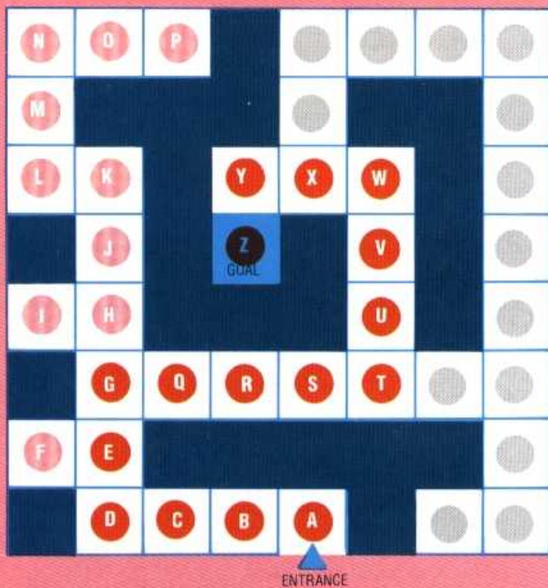
KEVIN JONES

Simple As ABC
Simple algorithms do work in maze-solving. This robot advances into empty spaces until it meets a dead-end — at squares F, I and P, for example. It then retreats to the last junction it encountered — square G here — marking all the intervening squares as useless in its mental map. If there are no untried routes from a junction the robot retreats to the junction before that, and so on, all the way back to the entrance, if necessary — in which case the maze is 'blind' or insoluble



Solving The Maze

```
49 REM*****CBM 64***********
50 REM*  MAZE SOLVER        *
51 REM*****CBM 64***********
100 GOSUB 2000              :REM INIT
150 GOSUB 9000              :REM PR.MAZE
200 FOR L=1 TO 1
250 GOSUB 7000              :REM LOOKROUND
300 GOSUB 8000              :REM MOVE
400 NEXT L
450 RO=16:CO=12:GOSUB 9500
500 IF MZ<>HM THEN PRINT"BLIND"
550 IF MZ=HM THEN PRINT"HOME"
900 PRINT:PRINT:INPUT A$:RUN
1999 REM***********************
2000 REM*   INIT              *
2001 REM***********************
2100 SZ=10:S2=SZ*SZ:GX=SZ/2:GY=SZ+2
2120 DIM MZ(SZ,SZ),R$(4),LX(4),LY(4)
2140 X=RND(-TI)
2150 DEFFNR(N)=INT(RND(1)*N+1)
2160 HM=-1:G0=-2:WL= 42:W$=CHR$(WL)
2180 CL$=CHR$(147):H$=CHR$(19)
2200 X=GX-1:Y=GY-2:DR=3
2220 D$=CHR$(17):P$=D$
2240 FORK=1TO5:P$=P$+P$:NEXT K:P$=H$+P$
2400 DATA 0,1,"D",-1,0,".",0,-1,"",1,0,""
2420 FOR K=1 TO 4:READ LX(K),LY(K),R$(K)
2490 NEXT K:RETURN
6999 REM***********************
7000 REM*  LOOK AROUND        *
7001 REM***********************
7110 RO=1:CO=1:GOSUB9500
7120 L=0:ND=0:FOR S=1 TO 4
7140 NX=X+LX(S):NY=Y+LY(S)
7200 IF NX<1 OR NX>SZ THEN NEXTS:RETURN
7220 IF NY<1 OR NY>SZ THEN NEXTS:RETURN
7230 MZ=MZ(NX,NY):IF MZ=0 THEN ND=S
7260 IF MZ=HM THEN ND=S:S=4:L=1
7280 NEXT S:RETURN
7999 REM***********************
8000 REM*   MOVE              *
8001 REM***********************
8100 MZ(X,Y)=DR:FL=1
8120 RO=Y+1:CO=X+1:GOSUB 9500
8140 IF ND=0 THEN ND=DR-2-4*(DR<3):FL=2
8160 PRINT R$(FL);X=X+LX(ND):Y=Y+LY(ND)
8180 DR=ND:IF Y>SZ THEN L=1:RETURN
8200 IF FL=2 THEN DR=MZ(X,Y)
8490 RETURN
8999 REM***********************
9000 REM* PRINT THE MAZE      *
9001 REM***********************
9040 PRINT CL$;:RO=1:CO=1
9050 WL=42:W$=CHR$(WL)
9060 S$="                  "
9070 FOR K=1 TO SZ+2:T$=T$+W$:NEXT K
9080 E$=W$+LEFT$(S$,SZ)+W$
9100 PRINT T$:FOR J=2 TO SZ+1
9120 PRINT E$:NEXT J:PRINT T$
9140 FOR K=1 TO S2/2
9150 WX=FNR(SZ):WY=FNR(SZ)
9160 MZ(WX,WY)=WL:RO=WY+1:CO=WX+1
9200 GOSUB 9500:PRINT W$:NEXT K
9250 CO=1+FNR(SZ):RO=1+FNR(SZ):GOSUB 9500
9300 PRINT"H":MZ(CO-1,RO-1)=HM
9350 RO=GY:CO=GX:GOSUB9500:PRINT"^"
9490 RETURN
9499 REM***********************
9500 REM* POSITION THE CRSR *
9501 REM***********************
9600 PRINT LEFT$(P$,RO)TAB(CO-1);:RETURN
```

BRIAN MORRIS

Basic Flavours

Make the following changes to this program:

BBC Micro

```
49 REM*******BBC************
50 REM*  MAZE SOLVER        *
51 REM*******BBC************
9040 CLS:RO=1:CO=1
9600 PRINT TAB(CO-1,RO-1);:RETURN
```

Spectrum

```
49 REM*****SPECTRUM********
50 REM*  MAZE SOLVER       *
51 REM*****SPECTRUM********
2120 DIM MZ(SZ,SZ):DIM R$(4)
2130 DIM LX(4):DIM LY(4)
2140 RANDOMIZE
2150 DEFFNR(N)=INT(RND*N+1)
9040 CLS:RO=1:CO=1
9600 PRINT AT (RO-1,CO-1);:RETURN
```

possible route, and sometimes closed to denote a wall. The mouse that reached the centre in the shortest time won the contest.

At the first British Micromouse contest, there were five entrants only. Some of these behaved in an extremely erratic fashion — one could not even travel in a straight line and even the best of the mice became quite bewildered once it had turned a couple of corners. In the same year, the European Finals of the competition were held, and mice began to arrive from Finland, Switzerland and Germany. Eventually, a mouse did succeed in negotiating the maze correctly; this was Nick Smith's 'Stirling Mouse', which was equipped with simple mechanical sensors that ran along the top of the maze walls and was powered by a simple stepper motor. Since then, interest in such competitions has grown, and in the 1984 Euromouse Contest in Madrid the fastest time to the centre of the maze was 31.4 seconds. Some contestants were still unable to reach the centre at all, but most succeeded.

MAPPING THE MAZE

So how does a robot mouse negotiate a maze? In general, the robot must have a precise method of moving itself around so that it knows its exact position at any time — this can be achieved by mounting the robot on wheels and driving it with stepper motors, often using some form of internal position feedback, such as shaft encoders. The robot also requires a set of sensors to detect the presence or absence of walls so that it can construct a 'map' of the maze. In Micromouse contests, the robots are allowed a couple of training runs, which they use to work out a plan of the course. They then make the competition run, during which they are timed in their attempts to reach the centre.

Although precise methods vary from one robot to another, one answer is to have the robot fitted with a simple tactile sensor at its front. Sitting at the centre of each square of the maze in turn, it can test to see if a wall is directly in front of it. It then turns clockwise through 90°, tests again, and repeats the sequence. Eventually it will 'know' where all the walls are in each square of the maze. This information can be stored as a single four-bit binary number — so 1111 in binary would represent a square with walls on all four sides (impossible in practice, as the robot could never enter that particular square), and 0000 would represent a square with no walls at all. 0111 would then represent a square with three walls and one opening — a cul de sac.

This information could be held in a two-dimensional array — in BASIC, DIM A(16,16) could be used to represent a maze with 16 'cells' in each direction. The robot then has to work out a route that will take it to A(8,8), if that is considered to be the centre of the maze. Often the robot has a built-in computer program that works out a tree structure for each route through the maze. Many of the branches of the tree will lead to dead ends or