4.5 In the first edition of this book, we presented the following version of the maze program. In retrospect, we don't much care for it. Criticize it, and contrast it with the recursive version.

```
/*   MOUSE IN A MAZE   */
/* SEARCHES PERIPHERY OF AN M X N MATRIX (MAX 50 X 50)   */
/* FOR AN ENTRY POINT.  THEN FOLLOWS PATH BY SIMULATING  */
/* MAN WITH RIGHT HAND ON WALL (TURN RIGHT IF YOU CAN,   */
/* LEFT IF YOU MUST) UNTIL HE REEMERGES.  FIRST PATH     */
/* WITH EXIT DIFFERENT THAN ENTRANCE IS ACCEPTED.        */
/* MAZE='1'B IS A PATH, MAZE='0'B IS A WALL.             */

EX510: PROCEDURE OPTIONS (MAIN);

/*** CONSTANTS ***/
   DECLARE (UP INITIAL(0), LEFT INITIAL(1),
            DOWN INITIAL(2), RIGHT INITIAL(3))
      STATIC FIXED BINARY;

/*** GLOBAL VARIABLES ***/
   DECLARE MAZE(50,50) BIT(1),
           (XPOS, YPOS)(1250) FIXED BINARY,
           L FIXED BINARY;

   ON ENDFILE STOP;

/*** INPUT AND LIST MAZE ***/
MORE:
   GET LIST (M, N);
   IF M<2 | M>50 | N<2 | N>50 THEN DO;
      PUT SKIP LIST (M, N, 'BAD DIMENSIONS');  STOP;
                                    END;

   GET EDIT (((MAZE(I,J) DO J = 1 TO N) DO I = 1 TO M))
      (COLUMN(1), (N)B(1));
   PUT PAGE EDIT (((MAZE(I,J) DO J = 1 TO N) DO I = 1 TO M))
      (LINE(33-M/2), (M)(COLUMN(40-N/2), (N)B(1)));

/*** SEARCH PERIPHERY ***/
   DECLARE FIND ENTRY (FIXED BINARY, FIXED BINARY, FIXED BINARY)
            RETURNS (BIT(1));      /* '1'B IF PATH FOUND */

   DO K = 2 TO M-1;
      IF FIND(K,1,RIGHT) THEN GOTO FOUND;
      IF FIND(K,N,LEFT)  THEN GOTO FOUND;
   END;
   DO K = 2 TO N-1;
      IF FIND(1,K,DOWN) THEN GOTO FOUND;
      IF FIND(M,K,UP)   THEN GOTO FOUND;
   END;

   PUT SKIP LIST ('NO ENTRANCE');  GOTO MORE;


/*** PRINT ANSWER ***/
   DECLARE XMAZE(50,50) CHARACTER(1);

FOUND:
   XMAZE = 'X';

   DO J = 1 TO L;       /* L, XPOS, YPOS ARE SET IN FIND */
      XMAZE(XPOS(J),YPOS(J)) = ' ';
   END;
   PUT PAGE EDIT (((XMAZE(I,J) DO J = 1 TO N) DO I = 1 TO M))
      (LINE(33-M/2), (M)(COLUMN(40-N/2), (N)A(1)));
   GOTO MORE;
```