```
/*  PROCEDURE FOR SEARCHING:  */
/* BEGINS AT BORDER CELL KX,KY AND PROBES IN DIRECTION   */
/* KW.  IF ENTRY THERE THEN FOLLOWS PATH, AND RETURNS    */
/* '1'B IF COMES OUT SOMEWHERE ELSE.                     */
/* STORES L ELEMENTS OF PATH IN XPOS, YPOS               */

   FIND:  PROCEDURE (KX, KY, KW) RETURNS (BIT(1));

   /*** CONSTANTS ***/
      DECLARE (DX(0:3) INITIAL(-1,0,1,0), DY(0:3) INITIAL(0,-1,0,1))
         STATIC FIXED BINARY;

   /*** STORAGE ***/
      DECLARE (X, Y, W) FIXED BINARY;

      IF MAZE(KX,KY) = '0'B | MAZE(KX+DX(KW),KY+DY(KW)) = '0'B
         THEN RETURN ('0'B);

   /*** EXPLORE ***/
      XPOS(1) = KX;  YPOS(1) = KY;
      XPOS(2),X = KX+DX(KW);  YPOS(2),Y = KY+DY(KW);
      W = KW;  L = 2;

      DO WHILE (X>1 & X<M & Y>1 & Y<N);
         W = MOD(W+3,4);                        /* TURN RIGHT */

         DO WHILE (MAZE(X+DX(W),Y+DY(W)) = '0'B);
            W = MOD(W+1,4);            /* TURN LEFT UNTIL OUT */
         END;

         L = L+1;
         XPOS(L),X = X+DX(W);  YPOS(L),Y = Y+DY(W);

         DO J = L-2 TO 1 BY -1;              /* TEST FOR KNOT */
            IF X = XPOS(J) & Y = YPOS(J) THEN
               IF J = 1 THEN RETURN ('0'B);
               ELSE DO;
                  L = J;  GOTO EXIT;
                    END;
         END;
   EXIT:
      END;

      RETURN ('1'B);                       /* REACHED BORDER */
   END FIND;

END EX510;
```