#include <stdio.h>

#define NumRows 8

#define NumCols 8

#define NumSquares (NumRows\*NumCols) /\* the number of squares to visit \*/

/\* StartRow and StartCol give the initial position of the knight.

\* Positions range from (0,0) to (NumRows-1,NumCols-1).

\*/

#define StartRow 0

#define StartCol 0

#define NotVisited 0

typedef int PartialTour[NumRows][NumCols];

int Ontheboard(int r, int c)

{

if(0<=r && r<8 && 0<=c && c<8)

return 1;

else

return 0;

}

int CompleteTour(PartialTour array, int move,int row, int col)

{

int RowChange[8] = { 2, 1, -1, -2, -2, -1, 1, 2};

int ColChange[8] = { 1, 2, 2, 1, -1, -2, -2, -1};

int direction; /\* loop counter for the 8 directions a knight can move \*/

if (! Ontheboard(row,col) || array[row][col]!=NotVisited)

return 0; /\* can't visit this square, so no possible tour \*/

/\* visit the square \*/

array[row][col] = move;

if (move >= NumSquares)

return 1; /\* partial tour is already complete \*/

/\* Try adding a move and completing the new partial tour. \*/

for (direction=0; direction<8; direction++)

{ if (CompleteTour(array,move+1, row+RowChange[direction], col+ColChange[direction]))

return 1; /\* tour completed successfully \*/

}

/\* None of the eight directions led to a complete tour with this

\* move added, so backtrack.

\*/

array[row][col] = 0;

return 0;

}

int main(void)

{

int row,col; /\* row and column loop indices \*/

PartialTour array;

/\* Initialise all the elements of the array to zero's indicating that no square is visited. \*/

for (row=0; row<8; row++)

for(col=0; col<8; col++)

array[row][col]=0;

if (CompleteTour(array, 1, StartRow,StartCol)) /\* printing the solution found \*/

{

for (row=0; row<8; row++)

{

for(col=0; col<8; col++)

printf("%3d",array[row][col]);

printf("\n");

}

return 0;

}

printf("No knight's tour for %d x %d board, starting at (%d,%d)\n",

NumRows, NumCols, StartRow, StartCol);

printf("Possible positions range from (0,0) to (%d,%d)\n",

NumRows-1, NumCols-1);

return 1;

}