// KNIGHT'S TRAVAILS

// NAME : Akshitha Peguda ENTRY NUMBER : 2021csb1122

// NAME : Chaitanya Pedapudi ENTRY NUMBER : 2021csb1121

#include <stdio.h>

#include <stdlib.h>

int size; // size of the chess board

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

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

/\* -2 | -1 | 0 | 1 | 2 <- y-axis

------------------------

-2 | x | | x |

------------------------

-1 x | | | | x

------------------------

0 | | o | |

------------------------

1 x | | | | x

------------------------

2 | x | | x | <-(starts with (2,1) and goes anti-clockwise)

------------------------

^

|

x-axis \*/

int movements(int x, int y, int MN, int chess[size][size]); // traces the movements of the knight all over the board

int safe(int x, int y, int chess[size][size]) // the square is empty or not for the placement of knight

{

if (x >= 0 && x < size && y >= 0 && y < size && chess[x][y] == -1) // -1 denotes empty & [0,size] is the limit of the board

return 0; // safe

else

return 1; // unsafe

}

void printchess(int chess[size][size]) //prints the chess board and the respective movement number (MN)

{

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

printf("%3d", chess[i][j]);

printf("\n");

}

}

int Board()

{

int chess[size][size]; // 2D array as the chess board representor

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

chess[i][j] = -1; // intially every square of the board is empty means it has every element as -1

chess[0][0] = 0; // the first starts with (0,0)

if (movements(0, 0, 1, chess) == 0) // if the knight couldn't move even a single step means no enough space

printf("NO ENOUGH SPACE\n");

else

printchess(chess); //if everything goes well print the whole chess board

return 0;

}

int movements(int x, int y, int MN, int chess[size][size])

{

int knight, nextx, nexty;

if (MN == size \* size) // if every square on the chessboard is covered end the program

return 1;

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

{

nextx = x + xmoves[knight]; // next eligible square with respect to x-axis

nexty = y + ymoves[knight]; // next eligible square with respect to y-axis

if (safe(nextx, nexty, chess) == 0) // if the eligible square is safe then move there

{

chess[nextx][nexty] = MN;

if (movements(nextx, nexty, MN + 1, chess))

return 1;

else

chess[nextx][nexty] = -1;

}

}

return 0;

}

int main()

{

printf("Enter the size of Chess Board : ");

scanf("%d", &size);

Board();

}