**ASSIGNMENT NO. 2**

**NAME:-** Harshada shitole

**CLASS:** SY-B AIDS

**BATCH:** C

**ROLL NO:** 2317122

**TITLE:** Implement a knight tour of NxN size boards using stack as a data structure to keep track

of the visited places.

**CODE :**

#include <iostream>

using namespace std;

#define MAX\_N 100

bool isSafe(int x, int y, int N, int sol[MAX\_N][MAX\_N])

{

return (x >= 0 && x < N && y >= 0 && y < N && sol[x][y] == -1);

}

void printSolution(int N, int sol[MAX\_N][MAX\_N])

{

for (int x = 0; x < N; x++) {

for (int y = 0; y < N; y++) {

int value = sol[x][y]; if (value

< 10) { cout << " " << value

<< " ";

} else if (value < 100) { cout

<< " " << value << " ";

} else { cout << value

<< " ";

}

}

cout << endl;

}

}

bool solveKTUtil(int x, int y, int movei, int N, int sol[MAX\_N][MAX\_N], int xMove[], int yMove[])

{

if (movei == N \* N)

return true;

for (int k = 0; k < 8; k++) { int next\_x = x + xMove[k]; int next\_y = y +

yMove[k]; if (isSafe(next\_x, next\_y, N, sol)) { sol[next\_x][next\_y] =

movei; if (solveKTUtil(next\_x, next\_y, movei + 1, N, sol, xMove,

yMove)) return true; else sol[next\_x][next\_y] = -1;

}

}

return false;

}

bool solveKT(int N)

{

if (N <= 0 || N > MAX\_N) { cout << "Invalid board size. Please enter a size between 1 and "

<< MAX\_N << "." << endl;

return false;

}

int sol[MAX\_N][MAX\_N];

for (int x = 0; x < N; x++)

for (int y = 0; y < N; y++)

sol[x][y] = -1;

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

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

sol[0][0] = 0;

if (solveKTUtil(0, 0, 1, N, sol, xMove, yMove)) {

printSolution(N, sol); return true;

} else { cout << "Solution does not exist" <<

endl; return false;

}

}

int main()

{

int N; cout << "Enter the size of the chessboard (1 to " << MAX\_N

<< "): "; cin >> N;

solveKT(N);

return 0;

}

**OUTPUT :**

student@cvlcomp05:~$ g++ Assignment\_2.cpp

student@cvlcomp05:~$ ./a.out

Enter the size of the chessboard (1 to 100): 8

0 59 38 33 30 17 8 63

37 34 31 60 9 62 29 16

58 1 36 39 32 27 18 7

35 48 41 26 61 10 15 28

42 57 2 49 40 23 6 19

47 50 45 54 25 20 11 14

56 43 52 3 22 13 24

5 51 46 55 44 53

4 21 12

student@cvlcomp05:~$ g++ Assignment\_2.cpp

student@cvlcomp05:~$ ./a.out

Enter the size of the chessboard (1 to 100): 5

0 5 14 9 20

13 8 19 4 15

18 1 6 21 10

7 12 23 16 3 24

17 2 11 22

student@cvlcomp05:~$ g++ Assignment\_2.cpp

student@cvlcomp05:~$ ./a.out

Enter the size of the chessboard (1 to 100): 7

0 37 30 7 18 35 14

31 28 19 36 15 6 17

38 1 32 29 8 13 34

27 24 39 20 33 16 5

40 21 2 25 44 9 12

23 26 47 42 11 4 45

48 41 22 3 46 43 10