**NAME – ABHIRAJ YOGESH SRIVASTAVA**

**ROLL NO. – 1906137**

**SUBJECT NAME – DESIGN AND ANALYSIS OF ALGORITHMS LAB**

**SUBJECT CODE – CSL4403**

**DATE – 22ND MARCH, 2021**

**BRANCH – CSE 2**

**ASSIGNMENT-12**

**Q12. WAP to implement N Queens problem using Backtracking.**

**Source Code in C++ Language:**

#define N 4

#include <stdbool.h>

#include <stdio.h>

int ld[30] = { 0 };

int rd[30] = { 0 };

int cl[30] = { 0 };

void printSolution(int board[N][N])

{

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

{

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

{

if(board[i][j]==0)

printf("X ");

else

printf("Q ");

}

printf("\n");

}

}

bool solveNQUtil(int board[N][N], int col)

{

if (col >= N)

return true;

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

{

if((ld[i - col + N - 1] != 1 && rd[i + col] != 1) && cl[i] != 1)

{

board[i][col] = 1;

ld[i - col + N - 1] = rd[i + col] = cl[i] = 1;

if (solveNQUtil(board, col + 1))

return true;

board[i][col] = 0;

ld[i - col + N - 1] = rd[i + col] = cl[i] = 0;

}

}

return false;

}

bool solveNQ()

{

int board[N][N] = { { 0, 0, 0, 0 },

{ 0, 0, 0, 0 },

{ 0, 0, 0, 0 },

{ 0, 0, 0, 0 } };

if (solveNQUtil(board, 0) == false) {

printf("Solution does not exist");

return false;

}

printSolution(board);

return true;

}

int main()

{

solveNQ();

return 0;

}

**Output Screenshot:**

