

Name:-Suryakant Upadhyay

PRN:-20220802043

Roll No.:-A20

## Practical Assignment No. 6

1. Write a c code to implement M queen(8 queen) problem using backtracking method.

```
#define N 4
#include <stdbool.h>
#include <stdio.h>
void printSolution(int board[N][N])
{
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            if(board[i][j])
                printf("Q ");
            else
                printf(".");
        }
        printf("\n");
    }
}
bool isSafe(int board[N][N], int row, int col)
{
    int i, j;
    for (i = 0; i < col; i++)
        if (board[row][i])
            return false;
    for (i = row, j = col; i >= 0 && j >= 0; i--, j--)
        if (board[i][j])
            return false;
```

```

for (i = row, j = col; j >= 0 && i < N; i++, j--)
    if (board[i][j])
        return false;

return true;
}
bool solveNQUtil(int board[N][N], int col)
{
if (col >= N)
    return true;
for (int i = 0; i < N; i++) {
    if (isSafe(board, i, col)) {
        board[i][col] = 1;
        if (solveNQUtil(board, col + 1))
            return true;
        board[i][col] = 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

```
/tmp/rbhPvy4tto.o  
. . Q .  
Q . . .  
. . . Q  
. Q . .  
|
```