Design and Analysis of Algorithms

Name: Harsh Brahmecha
PRN: 20220802003
LAB 6

CODE :

```c
#include <stdio.h>

#define N 8

// Function to print the board
void printBoard(int board[N][N]) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            printf("%2d ", board[i][j]);
        }
        printf("\n");
    }
}

// Function to check if it's safe to place a queen at board[row][col]
int isSafe(int board[N][N], int row, int col) {
    int i, j;

    // Check this row on the left side
    for (i = 0; i < col; i++) {
        if (board[row][i]) {
            return 0;
        }
    }

    // Check upper diagonal on the left side
```

```
    for (i = row, j = col; i >= 0 && j >= 0; i--, j--) {
        if (board[i][j]) {
            return 0;
        }
    }

    // Check lower diagonal on the left side
    for (i = row, j = col; i < N && j >= 0; i++, j--) {
        if (board[i][j]) {
            return 0;
        }
    }

    return 1;
}

// Recursive function to solve the N-Queens problem
int solveNQueens(int board[N][N], int col) {
    if (col >= N) {
        return 1; // All queens are placed
    }

    for (int i = 0; i < N; i++) {
        if (isSafe(board, i, col)) {
            board[i][col] = 1; // Place the queen

```

```
            if (solveNQueens(board, col + 1)) {
                return 1; // If placing queen in the current row leads to a solution
            }

            board[i][col] = 0; // If placing queen in the current row does not lead to a solution, backtrack
        }
    }

    return 0; // No solution found
}

int main() {
    int board[N][N] = {0}; // Initialize the board with 0s

    if (solveNQueens(board, 0) == 0) {
        printf("Solution does not exist for N = %d\n", N);
    } else {
        printf("Solution for N = %d:\n", N);
        printBoard(board);
    }

    return 0;
}

```

OUTPUT :

```
Solution for N = 8:
 1  0  0  0  0  0  0  0
 0  0  0  0  0  0  1  0
 0  0  0  0  1  0  0  0
 0  0  0  0  0  0  0  1
 0  1  0  0  0  0  0  0
 0  0  0  1  0  0  0  0
 0  0  0  0  0  1  0  0
 0  0  1  0  0  0  0  0
```