**Name:** Rohan Arun Nalawade

**Roll No:** 231012

**PRN:** 22310407  
**SY IT A**

**Assignment 7 B**

**Code implementation:**

#include <iostream>

#include <vector>

using namespace std;

bool isSafe(const vector<vector<int>>& board, int row, int col, int N) {

    // Check column

    for (int i = 0; i < row; i++)

        if (board[i][col])

            return false;

    // Check upper left diagonal

    for (int i = row - 1, j = col - 1; i >= 0 && j >= 0; i--, j--)

        if (board[i][j])

            return false;

    // Check upper right diagonal

    for (int i = row - 1, j = col + 1; i >= 0 && j < N; i--, j++)

        if (board[i][j])

            return false;

    return true;

}

bool solveNQueens(vector<vector<int>>& board, int row, int N) {

    if (row == N)

        return true;

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

        if (isSafe(board, row, col, N)) {

            board[row][col] = 1;

            if (solveNQueens(board, row + 1, N))

                return true;

            board[row][col] = 0; // Backtrack

        }

    }

    return false;

}

void printBoard(const vector<vector<int>>& board, int N) {

    cout << "One of the possible solutions:\n";

    for (int i = 0; i < N; i++) {

        for (int j = 0; j < N; j++) {

            cout << (board[i][j] ? "Q " : ". ");

        }

        cout << endl;

    }

}

int main() {

    int N;

    cout << "Enter the value of N (board size and number of queens): ";

    cin >> N;

    vector<vector<int>> board(N, vector<int>(N, 0));

    if (solveNQueens(board, 0, N))

        printBoard(board, N);

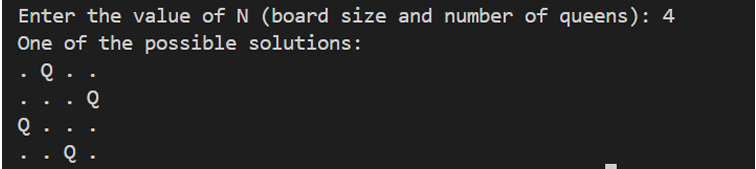
    else

        cout << "No solution exists for N = " << N << endl;

    return 0;

}

**Output:**

****