Assignment No 4

Code

```
#include <iostream>
#include <vector>
using namespace std;

bool isSafe(int row, int col, vector<vector<int>> &board, int n) {
   // Check upper-left diagonal

   int r = row, c = col;

   while (r >= 0 && c >= 0) {
      if (board[r][c] == 1)
         return false;
      r--;
      c--;
   }

   // Check left row
   r = row;
   c = col;
   while (c >= 0) {
      if (board[r][c] == 1)
         return false;
      c--;
   }

   // Check lower-left diagonal
   r = row;
   c = col;
   while (r < n && c >= 0) {
      if (board[r][c] == 1)
         return false;
      r++;
      c--;
   }
   return true;
}

void printSolutions(vector<vector<vector<int>>> &ans, int n) {
   int count = 1;
   for (auto &solution : ans) {
      cout << "Solution " << count++ << ":\n";
      for (int i = 0; i < n; i++) {
```

```
            for (int j = 0; j < n; j++) {
                cout << (solution[i][j] ? "Q " : "- ");
            }
            cout << "\n";
        }
        cout << "\n";
    }
}

void solve(int col, vector<vector<int>> &board, vector<vector<vector<int>>> &ans, int n) {
    if (col == n) {
        ans.push_back(board);
        return;
    }
    for (int row = 0; row < n; row++) {
        if (isSafe(row, col, board, n)) {
            board[row][col] = 1;
            solve(col + 1, board, ans, n);
            board[row][col] = 0;
        }
    }
}

int main() {
    int n;
    cout << "Enter number of Queens to be placed: ";
    cin >> n;
    vector<vector<int>> board(n, vector<int>(n, 0));
    vector<vector<vector<int>>> ans;
    solve(0, board, ans, n);
    printSolutions(ans, n);
    return 0;
}
```

Output :