**Code:**

#include <bits/stdc++.h>

using namespace std;

class Solution {

public: void solve(int col, vector < string > & board, vector < vector < string >> & ans, vector < int > & leftrow, vector < int > & upperDiagonal, vector < int > & lowerDiagonal, int n) {

if (col == n) {

ans.push\_back(board);

return;

}

for (int row = 0; row < n; row++) {

if (leftrow[row] == 0 && lowerDiagonal[row + col] == 0 && upperDiagonal[n - 1 + col - row] == 0) {

board[row][col] = 'Q';

leftrow[row] = 1;

lowerDiagonal[row + col] = 1;

upperDiagonal[n - 1 + col - row] = 1;

solve(col + 1, board, ans, leftrow, upperDiagonal, lowerDiagonal, n);

board[row][col] = '.';

leftrow[row] = 0;

lowerDiagonal[row + col] = 0;

upperDiagonal[n - 1 + col - row] = 0;

}

}

}

public: vector < vector < string >> solveNQueens(int n) {

vector < vector < string >> ans;

vector < string > board(n);

string s(n, '.');

for (int i = 0; i < n; i++) board[i] = s;

vector < int > leftrow(n, 0), upperDiagonal(2 \* n - 1, 0), lowerDiagonal(2 \* n - 1, 0);

solve(0, board, ans, leftrow, upperDiagonal, lowerDiagonal, n);

return ans;

}

};

int main() {

int n;

cout << "Enter the value of n: ";

cin >> n;

Solution obj;

vector < vector < string >> ans = obj.solveNQueens(n);

for (int i = 0; i < ans.size(); i++) {

cout << "Arrangement " << i + 1 << "\n";

for (int j = 0; j < ans[0].size(); j++) {

cout << ans[i][j];

cout << endl;

}

}

return 0;

}

**Output:**

