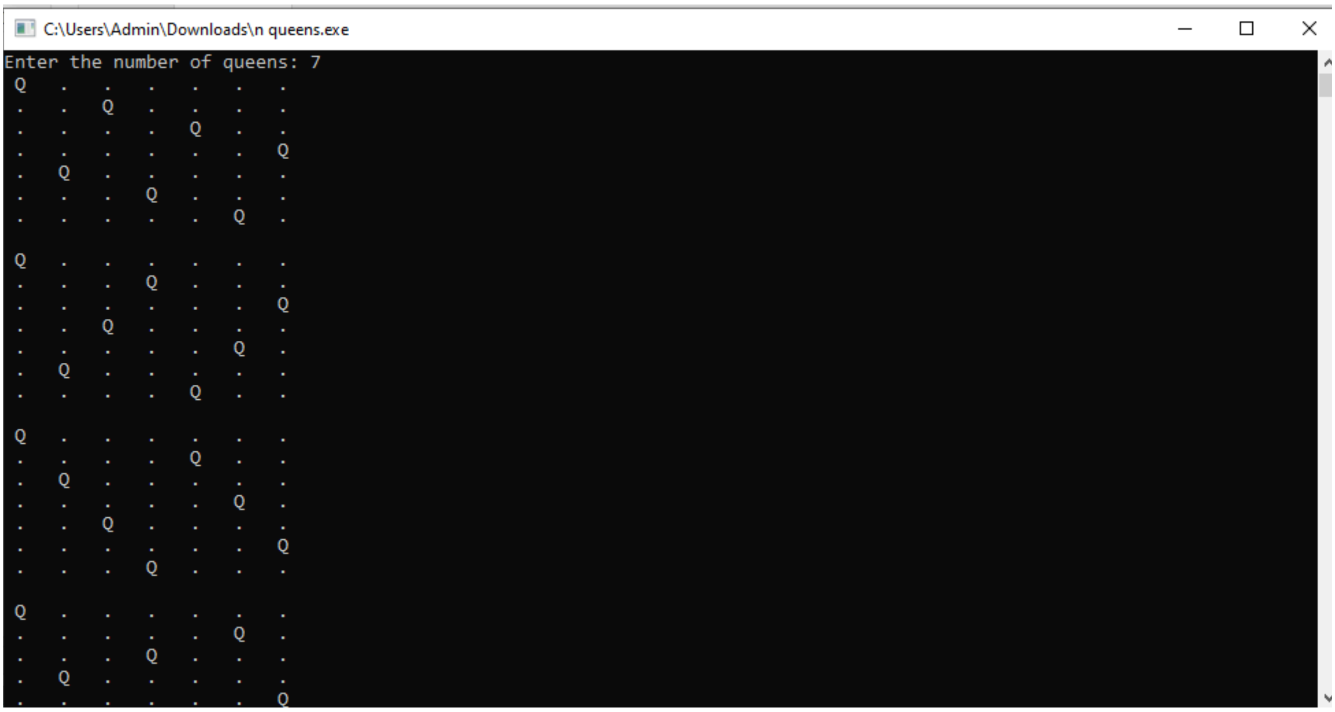
Assignment 9

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

// Function to print the chessboard
void printBoard(int board[][10], int N) {
   for (int i = 0; i < N; i++) {
      for (int j = 0; j < N; j++)
         cout << (board[i][j] ? " Q " : " . ") << " ";
      cout << endl;
   }
   cout << endl; // Print an empty line between solutions
}

// Function to check if a queen can be placed at board[row][col]
bool isSafe(int board[][10], int row, int col, int N) {
   // Check the column and both diagonals
   for (int i = 0; i < row; i++) {
      if (board[i][col] || (col - (row - i) >= 0 && board[i][col - (row - i)]) ||
         (col + (row - i) < N && board[i][col + (row - i)])) {
         return false;
      }
   }
   return true;
}

// Recursive function to solve N-Queens problem
bool solveNQUtil(int board[][10], int row, int N) {
   if (row >= N) {
      printBoard(board, N);
      return true; // Found a solution
   }
   bool foundSolution = false;
   for (int col = 0; col < N; col++) {
      if (isSafe(board, row, col, N)) {
         board[row][col] = 1; // Place the queen
         foundSolution = solveNQUtil(board, row + 1, N) || foundSolution;
         board[row][col] = 0; // Backtrack
      }
   }
   return foundSolution; // Return if a solution was found
}

// Function to solve the N-Queens problem
void solveNQueens(int N) {
   int board[10][10] = {0}; // Initialize the chessboard
   if (!solveNQUtil(board, 0, N)) {
      cout << "No solution exists for " << N << " queens." << endl;
```

```
    }
}

int main() {
    int N;
    cout << "Enter the number of queens: ";
    cin >> N;
    solveNQueens(N);
    return 0;
}
```

Output:



```
Enter the number of queens: 7
 Q   .   .   .   .   .   .
 .   .   Q   .   .   .   .
 .   .   .   .   Q   .   .
 .   .   .   .   .   .   Q
 .   Q   .   .   .   .   .
 .   .   .   Q   .   .   .
 .   .   .   .   .   Q   .

 Q   .   .   .   .   .   .
 .   .   .   Q   .   .   .
 .   .   .   .   .   .   Q
 .   .   Q   .   .   .   .
 .   .   .   .   .   Q   .
 .   Q   .   .   .   .   .
 .   .   .   .   Q   .   .

 Q   .   .   .   .   .   .
 .   .   .   .   Q   .   .
 .   Q   .   .   .   .   .
 .   .   .   .   .   Q   .
 .   .   Q   .   .   .   .
 .   .   .   .   .   .   Q
 .   .   .   Q   .   .   .

 Q   .   .   .   .   .   .
 .   .   .   .   .   Q   .
 .   .   .   Q   .   .   .
 .   Q   .   .   .   .   .
 .   .   .   .   .   .   Q
```