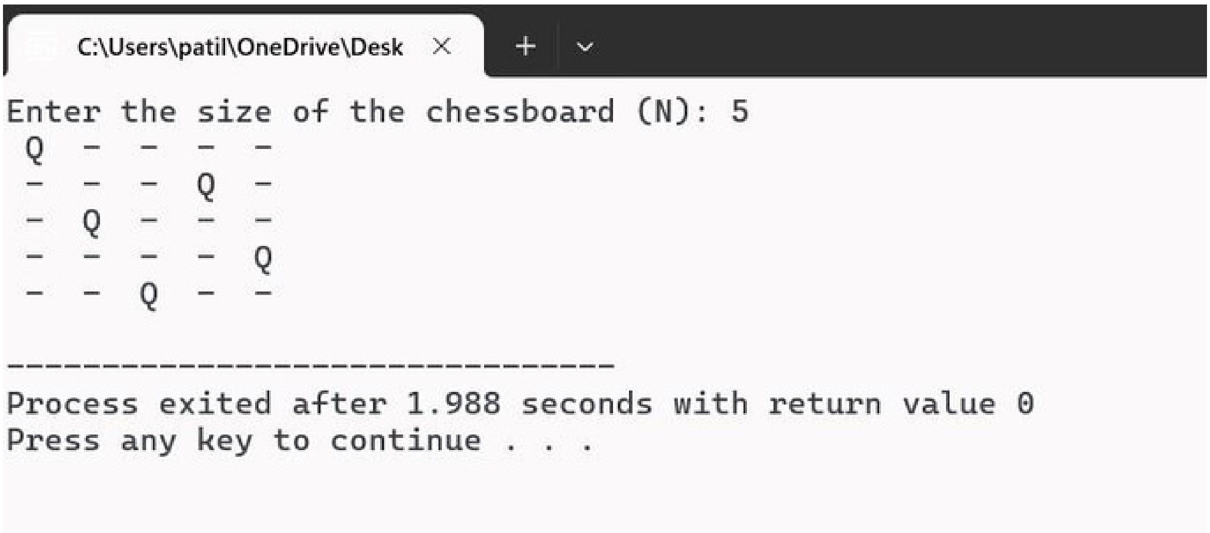
Assignment No 5 :- Design n-Queens matrix having first Queen placed. Use backtracking to place remaining Queens to generate the final n-queen‘s matrix.

Code:-

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

class NQueens
{ private: int N;

   // Function to print the chessboard void
   printBoard(const vector<vector<int> > &board) {
      for (int i = 0; i < N; i++) {
         for (int j = 0; j < N; j++) {
            cout << (board[i][j] ? " Q " : " - ");
         }
         cout << endl;
      }
   }

   // Function to check if a queen can be placed at board[row][col]
   bool isSafe(const vector<vector<int> > &board, int row, int col) {
      for (int i = 0; i < col; i++) { if (board[row][i])
         return false; // Check row
      }
      for (int i = row, j = col; i >= 0 && j >= 0; i--, j--) { if
         (board[i][j]) return false; // Check upper diagonal
      }
      for (int i = row, j = col; j >= 0 && i < N; i++, j--) { if
         (board[i][j]) return false; // Check lower diagonal
      }
      return true;
   }

   // Recursive function to solve the N-Queens problem bool
   solveNQueensUtil(vector<vector<int> > &board, int col) {
      if (col >= N) return true; // All queens are placed
      for (int i = 0; i < N; i++) { if (isSafe(board, i, col))
      {
```

```
                board[i][col] = 1; // Place queen if
                (solveNQueensUtil(board, col + 1)) return true; // Recur
                board[i][col] = 0; // Backtrack
            }
        }
        return false; // No valid position found
    }

public:
    // Function to initiate solving the N-Queens
    problem void solveNQueens(int n) { N = n;
        vector<vector<int> > board(N, vector<int>(N, 0));
        if (!solveNQueensUtil(board, 0)) {
            cout << "No solution exists." << endl;
            return;
        }
        printBoard(board);
    }
};

int main() {
    NQueens nq;
    int n; cout << "Enter the size of the chessboard
    (N): "; cin >> n; nq.solveNQueens(n); return 0;
}
```

Output :-



```
Enter the size of the chessboard (N): 5
 Q  -  -  -  -
 -  -  -  Q  -
 -  Q  -  -  -
 -  -  -  -  Q
 -  -  Q  -  -

--------------------------------
Process exited after 1.988 seconds with return value 0
Press any key to continue . . .
```