/\* N-Queen \*/ Problem

#include <iostream>

using namespace std;

bool isSafe(int board[][10], int row, int col, int n) {

// Check for queens in the same row

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

if (board[row][i] == 1)

return false;

}

// Check for queens in the upper left diagonal

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

if (board[i][j] == 1)

return false;

}

// Check for queens in the lower left diagonal

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

if (board[i][j] == 1)

return false;

}

return true;

}

bool solveQueens(int board[][10], int col, int n) {

if (col >= n)

return true;

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

if (isSafe(board, i, col, n)) {

board[i][col] = 1;

if (solveQueens(board, col + 1, n))

return true;

board[i][col] = 0;

}

}

return false;

}

void printSolution(int board[][10], int n) {

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

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

cout << board[i][j] << " ";

}

cout << endl;

}

}

void solveNQueens(int n) {

int board[10][10] = {0}; // Initialize the board with zeros

if (solveQueens(board, 0, n)) {

cout << "Solution exists. Placing queens:" << endl;

printSolution(board, n);

} else {

cout << "No solution exists." << endl;

}

}

int main() {

int n;

cout << "Enter the size of the chessboard (n x n): ";

cin >> n;

solveNQueens(n);

return 0;

}