import java.util.Scanner;

public class DAA\_5 {

public static boolean solveNQueens(int board[][], int col) {

int N = board.length;

if (col >= N) {

return true;

}

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

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

board[i][col] = 1;

if (solveNQueens(board, col + 1)) {

return true;

}

board[i][col] = 0;

}

}

return false;

}

public static boolean isSafe(int board[][], int row, int col) {

int N = board.length;

// left side of row...

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

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

return false;

}

}

// upper left diagonal.

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

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

return false;

}

}

// lower left diagonal...

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

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

return false;

}

}

return true;

}

public static void printSolution(int board[][]) {

int N = board.length;

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

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

System.out.print(board[i][j] + " ");

}

System.out.println();

}

}

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.println("Enter Number of queens: ");

int n = in.nextInt();

int board[][] = new int[n][n];

int firstQueenRow = 0;

int firstQueenCol = 0;

board[firstQueenRow][firstQueenCol] = 1;

if (solveNQueens(board, firstQueenCol + 1)) {

System.out.println("Solution exists");

printSolution(board);

} else {

System.out.println("No solution exists");

}

}

}