// DAA Practical 5 - 8 queens problem

public class Main {

final static int N = 8;

static void printBoard(int[][] board) {

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

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

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

}

System.out.println();

}

System.out.println();

}

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

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

if (board[i][col] == 1) return false;

}

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

if (board[i][j] == 1) return false;

}

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

if (board[i][j] == 1) return false;

}

return true;

}

static void solveNQueens(int[][] board, int row) {

if (row == N) {

printBoard(board);

return;

}

for (int i = 1; i < N; i++) {

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

board[row][i] = 1;

solveNQueens(board, row + 1);

board[row][i] = 0;

}

}

}

public static void main(String[] args) {

int[][] board = new int[N][N];

board[0][0] = 1;

solveNQueens(board, 1);

}

}