public class NQueens {

    private static final int N = 8; // Size of the chessboard

    public static void main(String[] args) {

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

        solveNQueens(board, 0);

    }

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

        if (col >= N) {

            printSolution(board);

            return;

        }

        for (int i = 0; i < N; i++) {

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

                board[i][col] = 1;

                solveNQueens(board, col + 1);

                board[i][col] = 0; // Backtrack

            }

        }

    }

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

        // Check if it's safe to place a queen at the given position

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

            if (board[row][i] == 1) {

                return false; // Check the row

            }

        }

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

            if (board[i][j] == 1) {

                return false; // Check the upper diagonal

            }

        }

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

            if (board[i][j] == 1) {

                return false; // Check the lower diagonal

            }

        }

        return true;

    }

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

        // Print the board configuration

        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();

    }

}