package ro.alex.ag.lab04;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Lab04 {

private static int[][] adjencyMatrix;

private static int nodes;

/\*\*

\* Citeste matricea de adiacenta a grafului.

\*

\* @return

\*/

public static void read() {

System.out.print("Numarul de varfuri: ");

try {

InputStreamReader streamReader = new InputStreamReader(System.in);

BufferedReader bufferedReader = new BufferedReader(streamReader);

nodes = Integer.parseInt(bufferedReader.readLine());

adjencyMatrix = new int[nodes + 1][nodes + 1];

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

for (int j = 1; j <= nodes; j++) {

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

adjencyMatrix[i][j] = Integer.parseInt(bufferedReader

.readLine());

}

}

} catch (IOException e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

boolean ok;

int i = 1;

int[] x = new int[100];

int nsi = 0;

read();

x[i] = 0;

do {

while (x[i] < nodes) {

x[i]++;

ok = true;

if (i >= 2) {

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

for (int k = j + 1; k <= nodes; k++) {

if (adjencyMatrix[x[j]][x[k]] == 1

|| adjencyMatrix[x[k]][x[j]] == 1) {

ok = false;

}

if (ok) {

if (i > nsi) {

nsi = i;

} else {

i = i + 1;

x[i] = 0;

}

}

}

}

}

}

i--;

} while (i != 0);

System.out.println("Numarul de stabilitate interna este " + nsi);

}

}