Problem A. A

Time Limit 1000 ms
Mem Limit 131072 kB

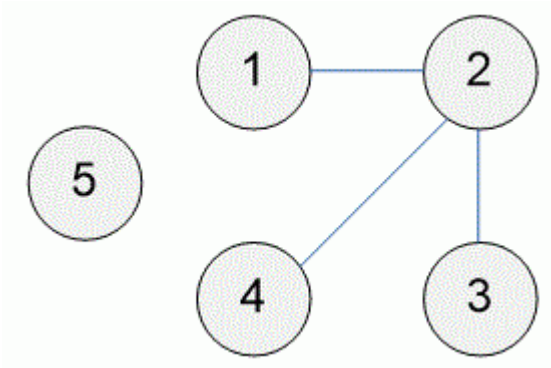
In the *Milky Way* galaxy, on the planet *Neptune*, there are n cities, some of which are connected by roads. Emperor *Maximus* of the *Milky Way* galaxy has decided to make an inventory of the roads on *Neptune*. However, as it turns out, he is not very good at math, so he asks you to count the number of roads.

Input

The first line contains the number n ($0 \leq n \leq 100$). Each of the next n lines contains n numbers, each either zero or one. If the position (i, j) in the square matrix is one, then the i-th and j-th cities are connected by a road; if it is zero, they are not connected.

Output

Print the number of roads on the planet *Neptune*.



Problem B. B

Time Limit 1000 ms
Mem Limit 131072 kB

For a given square matrix **n** x **n** of zeros and ones determine whether it can be the adjacency matrix of a simple undirected graph. The simple graph does not contain **self loops** and multiple edges.

Input

The first line contains the value of **n** (**1 ≤ n ≤ 100**). Each of the next **n** lines contains **n** numbers, describing the adjacency matrix.

Output

Print **YES**, if the graph is simple undirected and **NO** otherwise.

Sample 1

Input	Output
3 0 1 1 1 0 1 1 1 0	YES

Sample 2

Input	Output
3 0 1 1 1 0 1 0 1 0	NO

Problem C. C

Time Limit	3000 ms
Mem Limit	131072 kB

Create an undirected graph that supports the following operations:

1. **AddEdge**(**u**, **v**) — add an edge between vertices (**u**, **v**) in the graph;
2. **Vertex**(**u**) — print the list of vertices adjacent to vertex **u**.

There are no loops or multiple edges in the graph.

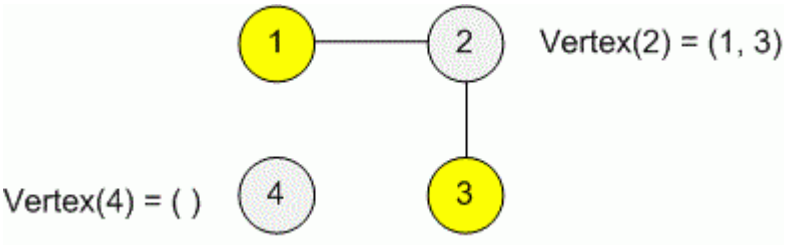
Input

The first line contains the number of vertices n ($1 \le n \le 10^5$). The next line contains the number of operations k ($0 \le k \le 10^6$). Each of the next k lines describes an operation in the format: "**1** " or "**2** ", representing, respectively, the operations **AddEdge**(**u**, **v**) and **Vertex**(**u**).

It is guaranteed that the total number of numbers printed during all **Vertex** operations does not exceed $2 \cdot 10^5$.

Output

For each **Vertex** command, print in a separate line the list of adjacent vertices for the given vertex. The vertices in each list can be printed in any order. If the given vertex has no adjacent vertices, print an empty line.



Problem D. D

Time Limit 2000 ms
Mem Limit 131072 kB

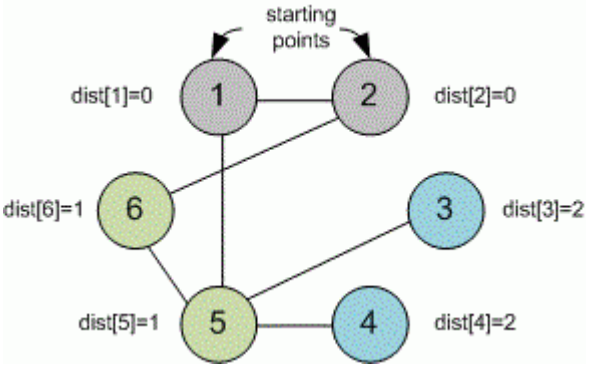
There is a common web in front of you. However, as an experienced contester, you noticed that it is a connected graph with n vertices and m edges. If you fire some vertex, it will light up, after a second all adjacent vertices light up, then all adjacent ones with already burning will light up, etc. You know which vertices will be fired in the web (all at the same time). Find how many seconds will pass until the last vertex lights up and find this vertex.

Input

The first line contains integers n $(1 \le n \le 10^5)$ and m $(n-1 \le m \le 10^5)$. Each of the next m lines contains two numbers --- the vertex numbers connected with an edge. The vertices are numbered starting from 1. The next line contains number k $(1 \le k \le n)$ --- the number of points to fire. Next line contains the numbers of k vertices to be fired.

Output

In the first line print the time when the last vertex will light up. In the second line print the number of this vertex. If there are several of them, print the one with minimum number.



Problem E. E

Time Limit 100 ms
Mem Limit 131072 kB

In the table with **n** rows and **n** columns, some cells are occupied with balls, others are free. You choose a ball and a place where you want to move it. In one step the ball can be moved to the next horizontal or vertical empty cell. Determine, is it possible to move the ball from the initial cell to the given, and if it is possible, find a way with the minimum number of steps.

Input

The first line contains integer **n** (**2** ≤ **n** ≤ **40**), each of the the next **n** lines contains **n** symbols. The empty cell is marked with a point, the ball on the table is marked with letter **O**, the initial position of a ball that must be moved - with **@**, and the place where the ball must be moved - with a letter **X**.

Output

Print **Y** in the first line , if the path from the start to the end exists, or **N** otherwise. If the path exists, print **N** rows with **N** symbols like it is given at input, but symbol **X** and all points on the route must be changed to pluses.

Examples

Input	Output
5 ....X .0000 ..... 0000. @....	Y +++++ +0000 +++++ 0000+ @++++

Input	Output
5 ..X.. ..... 00000 ....@..	N

Input	Output
5 ...X. ..... 0.000 ....@	Y ..++. .++.. 0+000 .++++ ....@

Problem F. F

Time Limit 1000 ms
Mem Limit 131072 kB

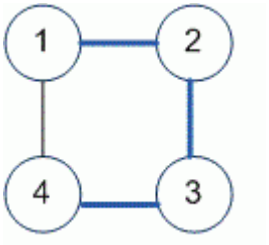
A connected undirected graph without loops or multiple edges is given. It is allowed to delete edges from the graph. The goal is to obtain a tree.

Input

The first line contains two integers n $(1 \leq n \leq 100)$ and m — the number of vertices and the number of edges in the graph. The next m pairs of integers represent the edges. It is guaranteed that the graph is connected.

Output

Print $n - 1$ pairs of integers — the edges that form a tree. The edges can be printed in any order.



Problem G. G

Time Limit 1000 ms
Mem Limit 131072 kB

Write a program that determines whether one of two given vertices in a tree is a parent of the other.

Input

The first line contains one integer n $(1 \leq n \leq 10^5)$ — the number of vertices in the tree.

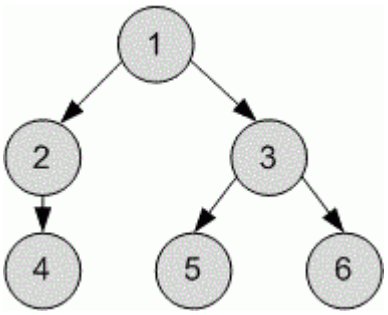
The second line contains n integers, where the i-th number represents the parent of the vertex numbered i. If the value is zero, the vertex is the root of the tree.

The third line contains an integer m $(1 \leq m \leq 10^5)$ — the number of queries.

Each of the following m lines contains two distinct integers a and b $(1 \leq a, b \leq n)$, representing a query.

Output

For each of the m queries, print 1 in a separate line if vertex a is a parent of vertex b, and 0 otherwise.



Examples

Input	Output
6 0 1 1 2 3 3 5 4 1 1 4 3 6 2 6 6 5	0 1 1 0 0

Problem H. H

Time Limit 1000 ms
Mem Limit 131072 kB

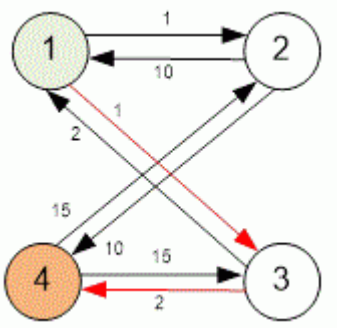
There are n cities in the country, some of which are connected by roads. To travel along one road, one tank of gasoline is required. In each city, the cost of a full tank of gasoline is different. Your task is to get from the first city to the n-th city with minimum expenses.

Input

The first line contains the number of cities n $(1 \leq n \leq 100)$. The second line contains n numbers, where the i-th number represents the cost of gasoline in the i-th city (all numbers are integers from 0 to 100). The third line contains the number of roads m in the country, followed by the description of the roads. Each road is represented by two numbers — the numbers of the cities it connects. All roads are bidirectional (that is, you can travel in both directions), there is at most one road between any two cities, and there are no roads that lead from a city to itself.

Output

Print a single integer — the minimum cost of the route, or -1 if it is impossible to reach the n-th city.



Examples

Input	Output
4 1 10 2 15 4 1 2 1 3 4 2 4 3	3

Input	Output
4 1 10 2 15 0	-1

Problem I. I

Time Limit 1000 ms
Mem Limit 131072 kB

Between some villages of Vasyuki district there are buses routes. As the passenger traffic is not very large, the buses run only a few times a day.

Maria Ivanovna wants to get from the village d to the village v as soon as possible (it is assumed that at time 0 she is in the village d).

Input

First given the total number of villages n $(1 \leq n \leq 100)$, d and v, then the number of bus lines r $(0 \leq r \leq 10000)$. Then given the description of bus trips. Each route is given by the number of the starting village, time of departure, the destination village and the arrival time (all times are integers from 0 to 10000). If at time t the passenger arrives in the village, he can leave it at any time, starting from t.

Output

Print the minimum time when Maria Ivanovna can reach the village v. If she fails with these bus trips to get from d to v, output -1.

Examples

Input	Output
3 1 3 4 1 0 2 5 1 1 2 3 2 3 3 5 1 1 3 10	5

Problem J. J

Time Limit 1000 ms
Mem Limit 131072 kB

A graph is given. Determine does it contain a cycle of negative weight, and if so, print it.

Input

The first line contains the number of vertices **n** (**1 ≤ n ≤ 100**). Each of the next **n** lines contains **n** numbers - the adjacency matrix of the graph. Weights of the edges do not exceed **10000** by absolute value. If the edge is absent, the corresponding value is **100000**.

Output

Print in the first line "**YES**", if a cycle exists, or "**NO**" otherwise. In the presence of the cycle output in the second row the number of vertices in it (assuming the same first and last) and the third row - tops included in this series in order of traversal. If several cycles - output any.

Examples

Input	Output
2 0 -1 -1 0	YES 3 1 2 1