Problem A: Family Tree

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

As you've been really getting into genealogy lately, you decided to write a program that can process a huge family tree containing a continuous series of one person's ancestors in such a way that queries regarding the ancestor-descendant-relationship of any two persons in it can be answered in constant time. To test your program's efficiency, you want to scan an enormous family tree and then answer a large number of such queries.

Input

The input consists of

- one line containing N ($1 \leq N \leq 10^4$) – the number of people in the tree
- N lines containing their names (the i-th line giving the name of person i), with each name being a string of length not exceeding 30 consisting only of letters and (possibly) spaces
- one line containing N integers, the i-th of which is the index of the direct descendant of person i that is included in the tree. It is 0 for the last descendant, i.e. the one all other persons are ancestors of.
- one line containing M ($1 \leq M \leq 10^4$) – the number of queries you should answer
- M lines, each containing a query in the form "a b", $1 \leq a, b \leq N$, $a \neq b$, referring to two of the persons in the tree by their indices.

Output

Answer each query "a b" on a separate line by printing *"n_a is an ancestor of n_b."* if a is an ancestor of b, *"n_b is an ancestor of n_a."* if b is an ancestor of a and *"n_a and n_b are not related."* if neither is the case, where n_a and n_b are the names of a and b, respectively.

Sample input and output

Input	**Output**
4 Abe Simpson Homer Simpson Marge Simpson Lisa Simpson 2 4 4 0 3 1 4 4 3 2 3	Abe Simpson is an ancestor of Lisa Simpson. Marge Simpson is an ancestor of Lisa Simpson. Homer Simpson and Marge Simpson are not related.

Problem B: Venetian Bridges

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

The city council of Venice, the Italian city world-renowned for its countless beautiful bridges, has once again decided to refurbish some of them.

Whenever it does this, it needs to be very careful: because bridges cannot be used during renovation and since for the tens of thousands of people that walk the city's streets every day they are by far the most important way to travel between its 118 islands, the council has to watch out never to block any *essential* bridges – that is, bridges whose blockade would break the city into two isolated parts with respect to moving on foot. Obviously, whether a bridge is *essential* depends heavily on which of the surrounding bridges are already being renovated, so this has to be taken into account as well.

You are to write an interactive program that counsels the council on which renovations it can induce. To make things a bit easier for you, they have numerated the islands $1, ..., 118$ and will communicate with you only by means of this numeration.

Input

In the beginning you get a description of the current situation: The first line of input contains M $(117 \leq M \leq 400)$ – the number of currently usable bridges. The next M lines each contain the description of a bridge, that is, two integers a and b $(1 \leq a, b \leq 118)$ denoting a (two-way) bridge between islands a and b. Be careful: while there never are bridges between an island and itself, there often are multiple bridges between the same two islands!

Then there is a line containing Q – the number of queries – $(1 \leq Q \leq 10^3)$ followed by Q lines containing queries of the following forms:

- "R a b" with $1 \leq a, b \leq 118$, telling you that the council wishes to renovate a bridge between islands a and b
- "C a b" with $1 \leq a, b \leq 118$, telling you that the renovation of a bridge between islands a and b has been completed

Output

For each query of the first kind, check whether the bridge in question is currently *essential* (it is guaranteed that the council doesn't ask to renovate a bridge that doesn't exist or is already being renovated). If so, output "Impossibile!", otherwise output "Va bene!" and assume the renovations to begin immediately (i.e. factor in that this bridge isn't usable anymore when answering the next queries).

For each query of the second kind, take into account that one more such bridge is now usable again, when answering the queries that follow.

A sample of how the interaction could look like if there were only 6 islands instead of 118:

Sample input and output

Input	**Output**
7 1 2 1 4 2 3 3 4 4 5 4 6 5 6 5 R 3 4 R 1 4 R 4 5 C 3 6 R 4 6	Va bene! Impossibile! Va bene! Va bene!

Problem C: Unavoidable Intersections

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

As he is – on principle – opposed to all forms of motorized transportation, Beppo needs to walk all the way from his home to his workplace every day. Starting at the intersection at which he lives, he needs to follow a series of bidirectional roads and intersections connecting them until he finally arrives at his workplace, which is also located at an intersection. Of course he figured out the shortest path long ago, but after a while, following it every single day became way too boring. Nowadays, he instead leaves the house trying to find an interesting new path to work. However, something about this is bothering him very much: No matter how hard he tries, some of the intersections (besides the ones where he starts and ends his commute) seem to be unavoidable, in that every path he can choose will necessarily contain them. But is this actually the case or has he simply not tried hard enough to find ways around them? Unfortunately he cannot program well enough to check this by himself, so you should help him.

Input

The input consists of

- one line containing N ($3 \leq N \leq 2 \cdot 10^4$) and M ($N - 1 \leq M \leq 2 \cdot 10^5$) – the number of intersections and roads in the town where Beppo lives, respectively
- one line containing h and w ($1 \leq h, w \leq N, h \neq w$) – the indices of the intersections where Beppo's home and workplace are located
- M lines containing descriptions of the roads – each description consists of two integers a and b ($1 \leq a, b \leq N$), denoting a road bidirectionally connecting intersections a and b. It is guaranteed that each intersection can be reached from each other intersections through a series of roads.

Output

First output the number of unavoidable intersections as described above. If there are any, output their indices on the next line, in the order in which Beppo visits them on his way to work.

Note that, even though intersections h and w are of course part of every h-w-path, Beppo accepts that as natural and doesn't view them as the kind of unavoidable intersections he despises (and wants you to calculate).

Sample input and output

Input	**Output**
3 2 1 2 1 3 1 2	0
3 2 1 3 1 2 2 3	1 2
9 12 9 6 1 2 2 3 4 5 2 6 2 7 8 9 1 3 1 4 1 5 6 7 3 8 3 9	2 3 2

Problem D: Precedence graph

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

A friend of yours is currently in the process of creating his own operating system. He recently told you in confidence how he can't seem to find a good way to handle the scheduling of codependent processes. So far, he is already able to let the processes in question create a precedence graph (i.e. a graph in which the vertices are the processes and an edge from vertex a to vertex b denotes that process a has to be done before process b can be started) but could not yet figure out an efficient way to find an order of computation that is consistent with this graph.

Back when he told you this you had no idea on how to solve it either, but in the latest *Advanced Algorithms for Programming Contests* lecture you learned about an algorithm that seems to exist for this exact purpose, so you decided to surprise your friend by using it to write a program that solves his problem.

Input

The input consists of

- one line containing N $(2 \leq N \leq 10^4)$ and M $(1 \leq M \leq 10^5)$ – the number of processes and the number of dependencies between them, respectively
- M lines each containing PIDs a_i and b_i $(1 \leq a_i, b_i \leq n)$, denoting that process a_i needs to be done computing before b_i may start.

Output

Output the PIDs $(1, ..., N)$ separated by spaces in an order that is consistent with the given precedence graph. If there are multiple solutions, assume that processes with smaller PID have higher priority and should therefore be first in the list (give back the list that is smallest lexicographically).

Sample input and output

Input	**Output**
4 3 1 3 1 2 4 2	1 3 4 2

Problem E: Fire safety

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

In the city of Göttham there are N houses, some of which are connected by one-way roads. Lately there had been increasingly many fires in the city, so the people of Göttham decided to finally build some fire stations.

But first they need to solve a problem: While a fire truck can of course use streets in the wrong direction when heading to an emergency, it has to obey the rules of Göttham and drive in the right directions on its return.

Obviously fire trucks should be able to return to their respective station from wherever in the city they needed to go. However, building fire stations is very costly, so the people on the main council of Göttham decided to build the minimum number of fire stations necessary to satisfy this demand. Moreover, in order to save money, stations will be built next to already existing houses.

Your task is to write a program that calculates optimal positions for the fire stations and checks how many are needed.

Input

The input consists of

- one line containing N, $(1 \leq N \leq 3000)$ – the number of houses
- one line containing M – the number of roads $(1 \leq M \leq 10^5)$
- M lines corresponding to the roads – each of them containing two integers a_i and b_i $(1 \leq a_i, b_i \leq n)$, denoting that there is a road in the city on which vehicles are allowed to drive from house a_i to house b_i.

Output

Output a single integer – the minimum number of fire stations required in Göttham.

Sample input and output

Input	Output
5 7 1 2 2 3 3 1 2 1 2 3 3 4 2 5	2

Problem F: Links

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds
Memory: 512 MB

Problem description

Surfing around on Wikipedia you can reach many pages by only following the links on other pages. However, you can not reach every page from any other page only following links, which is absolutely unacceptable! Since adding a lot of links is troublesome, you decided that you need a program to find the minimum number of links that have to be added in order to enable surfing all pages only by clicking links from any start page.

Input

The input consists of

- one line containing N and M $(1 \leq N \leq 10^5, 0 \leq M \leq 10^5)$ – the numbers of pages and already existing links, respectively
- M lines containing each integers a and b $(1 \leq a, b \leq N)$, meaning that there exists a link from page a to page b.

Output

Output a single integer – the minimum number of extra links that have to be added in order for every page to be reachable from any other page.

Sample input and output

Input	**Output**
3 2 1 2 2 1	2
3 2 1 2 2 3	1