Competitive Programming

Saarland University — Summer Semester 2020

Julian Baldus, Markus Bläser, Karl Bringmann, Marian Dietz, Simon Schwarz, Christoph Weidenbach, Dominic Zimmer

Assignments Week 6

Deadline: **June 16, 2020 at 16:00 sharp**

Please submit solutions to the problems in our judge system, available at `https://compro.mpi-inf.mpg.de/`.
You can find your credentials on your personal status page in our CMS.

Problem	`bridges`	`driving`	`caves`		`distance`
Points	3	3	1	2	3
Difficulty	🌶️	🌶️🌶️	🌶️🌶️	🌶️🌶️🌶️	🌶️🌶️🌶️🌶️
Time Limit	1s	1s	1s	1s	1s
Memory Limit	2 GB	2 GB	2 GB	2 GB	2 GB

Please note:

- Later, we will reopen the judge for the problems of the last weeks. However, you won't get any points for submissions of these problems.
- In the judge you can switch between the exercises of different weeks in the top-right corner.
- Your solution will be judged immediately after submitting. This may take some time, depending on the current server load.
- You can submit as many times as you want. However, don't abuse the server or try to extract the secret test cases.
- If your solution is **accepted**, you will receive the points specified in the table above.
- If you get **another verdict**, you will receive 0 points.

Bridges
Problem ID: bridges

Dieter Schlau is a quite successful criminal by now and is particularly good at hiding from the police. In fact, he is so good at hiding, he needs your help. After the last heist, Dieter hid in a city that he cannot figure out the name of. He is uncertain whether he is in *Saarbrücken*, *Zweibrücken* or any other city conveniently named after their bridges.

Can you help him identify the city by counting the bridges?

Input

The first line of the input contains two integers n ($1 \leq n \leq 10^5$) and m ($1 \leq m \leq 2 \cdot 10^5$). The following m lines each contain two integers s_i, d_i with $1 \leq s_i \leq n$, $1 \leq d_i \leq n$, representing a (bidirectional) connection between s_i and d_i. You can assume that there is at most one connection between each pair of nodes.

Output

Print a single integer, the number of bridges in the graph.

Sample Input 1	Sample Output 1
5 5 1 2 2 3 2 4 3 4 4 5	2

Sample Input 2	Sample Output 2
4 4 1 2 2 3 3 4 4 1	0

Sample Input 3	Sample Output 3
4 3 1 2 2 3 3 4	3

Autonomous Driving
Problem ID: driving

The *Excellent University for Local Engineering Research (EULER)* in Saarbrücken has invented a self-driving car. They want to test their invention in the streets of Saarbrücken. Saarbrücken consists of n places which are connected by m one-way streets. It may be possible that there are multiple streets from one place to another.

The *EULER* wants its software to be thoroughly tested. Therefore, they want the car to drive through each street once. The route should start at place 1 (which is the university) and should end at place n. However, the software is still a little bit buggy and crashes when a street is visited twice. Hence, a street should not be driven through more than once.

Can you determine whether it is possible to find a route starting from 1 and ending in n that drives through every street *exactly* once?

Input

The first line of the input contains two integers n ($2 \leq n \leq 10^5$) and m ($1 \leq m \leq 2 \cdot 10^5$). The following m lines each contain two integers s_i, d_i with $1 \leq s_i \leq n$, $1 \leq d_i \leq n$, representing an one-way street from s_i to d_i.

Output

Print "`POSSIBLE`" if there is a route fulfilling the criterions. Otherwise, print "`IMPOSSIBLE`".

Sample Input 1	Sample Output 1
3 3 1 2 2 1 1 3	POSSIBLE

Sample Input 2	Sample Output 2
3 3 1 2 2 1 2 3	IMPOSSIBLE

Sample Input 3	Sample Output 3
4 4 1 2 2 1 3 4 4 3	IMPOSSIBLE

Sample Input 4	Sample Output 4
2 5 1 2 2 1 1 2 2 1 1 2	POSSIBLE

Caves
Problem ID: caves

You are playing a game that has n caves and m one-way tunnels between them. In each cave lies a certain number of gold ingots.
What is the maximum number of gold ingots you can collect while moving through the caves when you can freely choose your starting and ending cave?

Subtasks

This problem contains subtasks. The solution to one subtask is usually an improvement over or adaption of the solution to a previous subtask. You should, therefore, think of subtasks as hints that guide you in finding a correct solution.
Due to constraints of the judge system, each subtask appears as a separate problem on the scoreboard and you have to submit your program to each of them.

- **Subtask 1** (1 point) The caves and tunnels form a directed acyclic graph.
- **Subtask 2** (2 points) The graph may contain cycles.

Input

The first line contains two integers n and m ($1 \leq n \leq 10^5$, $1 \leq m \leq 2 \cdot c10^5$), the number of caves and the number of tunnels.
The second line contains n integers v_i ($1 \leq v_i \leq 10^9$), the number of gold ingots that are in the cave i. The following m lines each contain two integers s_i, d_i ($1 \leq s_i, d_i \leq n$), indicating that there is a one-way tunnel from s_i to d_i.

Output

Print the total value of gold ingots you can collect when you can choose the start and end cave freely.

Sample Inputs

Note that the first two inputs can occur in both subtasks. The last three inputs are only valid for subtask 2.

Sample Input 1	Sample Output 1
5 5 10 20 30 40 50 1 2 2 5 1 3 3 4 4 5	130

Sample Input 2	Sample Output 2
6 6 10 20 30 40 50 60 1 3 3 2 2 4 3 4 1 2 6 5	110

Sample Input 3

```
4 4
4 5 2 7
1 2
2 1
1 3
2 4
```

Sample Output 3

```
16
```

Sample Input 4

```
6 6
60 50 40 30 20 10
1 2
2 3
3 4
4 5
5 2
4 6
```

Sample Output 4

```
210
```

Sample Input 5

```
7 10
10 20 30 40 50 60 70
5 2
2 2
3 6
6 4
4 3
2 3
5 1
7 1
5 7
7 5
```

Sample Output 5

```
270
```

Keep Your Distance
Problem ID: distance

You want to go through a corridor and there are people in it. Due to corona, you want to keep a certain distance. For each person, you look if he wears his face protection properly, seems sick, or simply belongs to a risk group. According to those observations, you calculate for each person an individual distance that you want to keep to them.
People are modelled as points on a $2D$-coordinate system. Person i has the coordinates (x_i, y_i). You want to keep a distance of at least r_i to the i-th person. Passing by at distance exactly r_i is allowed.
The corridor is specified by two integers x_{min} and x_{max} with $x_{min} < x_{max}$. You want to pass the corridor from $y = -\infty$ to $y = \infty$ while staying between x_{min} and x_{max}. Note that it is allowed to walk on the borders (i.e. on x_{min} or x_{max}).

You do not need advanced geometry knowledge to solve this task, but you should know that the distance between two points (x_1, y_1) *and* (x_2, y_2) *is* $\sqrt{|x_1 - x_2|^2 + |y_1 - y_2|^2}$.

Input

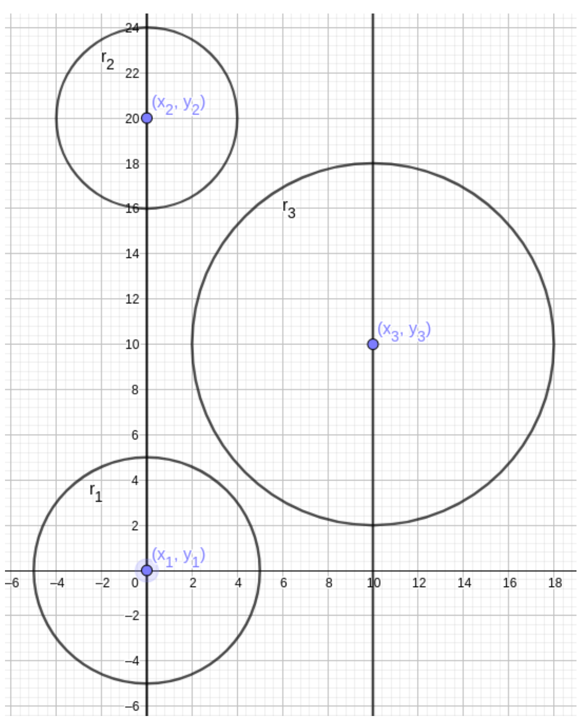
The first line contains three integers x_{min}, x_{max} and n ($0 \leq x_{min} < x_{max} \leq 10^9$, $1 \leq n \leq 2000$), the x-coordinates of the walls of the corridor and the number of people in the corridor. The following n lines each contain three integers x_i, y_i and r_i ($x_{min} \leq x_i \leq x_{max}$, $0 \leq y_i, r_i \leq 10^9$) describing a person standing at position (x_i, y_i) from whom you want to keep a distance of at least r_i.
It is not guaranteed that people keep the required distance from each other (for example, they might live in the same household).

Output

Print `possible` if you can pass the corridor without getting too close to another person, or `impossible` otherwise.

Explanation of the first sample input

A visualization of the first input is:



Note that you can clearly pass from $y = -\infty$ to $y = \infty$. Thus, the output should be “possible”.

Sample Input 1	Sample Output 1
0 10 3 0 0 5 0 20 4 10 10 8	possible

Sample Input 2	Sample Output 2
0 16 5 11 8 3 6 10 3 7 3 4 10 4 3 15 5 3	possible

Sample Input 3	Sample Output 3
0 11 5 8 11 2 10 6 2 3 7 3 4 10 2 5 15 2	possible

Sample Input 4	Sample Output 4
0 11 5 8 11 3 10 6 3 3 7 4 4 10 3 5 15 3	impossible