Codeforces Round #503 (by SIS, Div. 1)

A. Elections

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

As you know, majority of students and teachers of Summer Informatics School live in Berland for the most part of the year. Since corruption there is quite widespread, the following story is not uncommon.

Elections are coming. You know the number of voters and the number of parties — n and m respectively. For each voter you know the party he is going to vote for. However, he can easily change his vote given a certain amount of money. In particular, if you give i -th voter c_i bytecoins you can ask him to vote for any other party you choose.

The United Party of Berland has decided to perform a statistical study — you need to calculate the minimum number of bytecoins the Party needs to spend to ensure its victory. In order for a party to win the elections, it needs to receive strictly more votes than any other party.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 3000$) — the number of voters and the number of parties respectively.

Each of the following n lines contains two integers p_i and c_i ($1 \le p_i \le m$, $1 \le c_i \le 10^9$) — the index of this voter's preferred party and the number of bytecoins needed for him to reconsider his decision.

The United Party of Berland has the index 1.

Output

Print a single number — the minimum number of bytecoins needed for The United Party of Berland to win the elections.

Examples

input
1 2 1 100
output
0

input
5 5 2 100 3 200 4 300 5 400 5 900
output
500

input
5 5 2 100 3 200 4 300 5 800 5 900
output
600

Note

In the first sample, The United Party wins the elections even without buying extra votes.

In the second sample, The United Party can buy the votes of the first and the fourth voter. This way The Party gets two votes, while parties 3, 4 and 5 get one vote and party number 2 gets no votes.

In the third sample, The United Party can buy the votes of the first three voters and win, getting three votes against two votes of the fifth party.

B. The hat

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an interactive problem.

Imur Ishakov decided to organize a club for people who love to play the famous game «The hat». The club was visited by n students, where n is even. Imur arranged them all in a circle and held a draw to break the students in pairs, but something went wrong. The participants are numbered so that participant i and participant $i + 1$ ($1 \le i \le n - 1$) are adjacent, as well as participant n and participant 1. Each student was given a piece of paper with a number in such a way, that for every two adjacent students, these numbers differ exactly by one. The plan was to form students with the same numbers in a pair, but it turned out that not all numbers appeared exactly twice.

As you know, the most convenient is to explain the words to the partner when he is sitting exactly across you. Students with numbers i and $i + \frac{n}{2}$ sit across each other. Imur is wondering if there are two people sitting across each other with the same numbers given. Help him to find such pair of people if it exists.

You can ask questions of form «which number was received by student i?», and the goal is to determine whether the desired pair exists in no more than 60 questions.

Input

At the beginning the even integer n ($2 \le n \le 100\,000$) is given — the total number of students.

You are allowed to ask no more than 60 questions.

Output

To ask the question about the student i ($1 \le i \le n$), you should print «? i». Then from standard output you can read the number a_i received by student i ($-10^9 \le a_i \le 10^9$).

When you find the desired pair, you should print «! i», where i is any student who belongs to the pair ($1 \le i \le n$). If you determined that such pair doesn't exist, you should output «! -1». In both cases you should immediately terminate the program.

The query that contains your answer is not counted towards the limit of 60 queries.

Please make sure to flush the standard output after each command. For example, in C++ use function `fflush(stdout)`, in Java call `System.out.flush()`, in Pascal use `flush(output)` and `stdout.flush()` for Python language.

Hacking

Use the following format for hacking:

In the first line, print one even integer n ($2 \le n \le 100\,000$) — the total number of students.

In the second line print n integers a_i ($-10^9 \le a_i \le 10^9$) separated by spaces, where a_i is the number to give to i-th student. Any two adjacent elements, including n and 1, must differ by 1 or -1.

The hacked solution will not have direct access to the sequence a_i.

Examples

input
8 2 2
output
? 4 ? 8 ! 4

input
6 1 2 3 2 1

0

output
? 1 ? 2 ? 3 ? 4 ? 5 ? 6 ! -1

Note

Input-output in statements illustrates example interaction.

In the first sample the selected sequence is $1, 2, 1, 2, 3, 4, 3, 2$

In the second sample the selection sequence is $1, 2, 3, 2, 1, 0$.

C. Sergey's problem

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sergey just turned five years old! When he was one year old, his parents gave him a number; when he was two years old, his parents gave him an array of integers. On his third birthday he received a string. When he was four, his mother woke him up in a quiet voice, wished him to be a good boy and gave him a rooted tree. Today he celebrates his birthday again! He found a directed graph without loops as a present from his parents.

Since Sergey is a very curious boy, he immediately came up with a thing to do. He decided to find a set Q of vertices in this graph, such that no two vertices $x, y \in Q$ are connected by an edge, and it is possible to reach any vertex $z \notin Q$ from some vertex of Q in no more than two moves.

After a little thought, Sergey was able to solve this task. Can you solve it too?

A vertex y is reachable from a vertex x in at most two moves if either there is a directed edge (x, y), or there exist two directed edges (x, z) and (z, y) for some vertex z.

Input

The first line of input contains two positive integers n and m ($1 \le n \le 1\,000\,000$, $1 \le m \le 1\,000\,000$) — the number of vertices and the number of edges in the directed graph.

Each of the following m lines describes a corresponding edge. Each one contains two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) — the beginning and the end of the i-th edge. The graph may contain multiple edges between the same pair of vertices.

Output

First print the number k — the number of selected vertices. Then print k distinct integers — the indices of the selected vertices.

If multiple answers exist you can output any of them. In particular, you don't have to minimize the number of vertices in the set. It is guaranteed, that there is always at least one valid set.

Examples

input
5 4 1 2 2 3 2 4 2 5

output
4 1 3 4 5

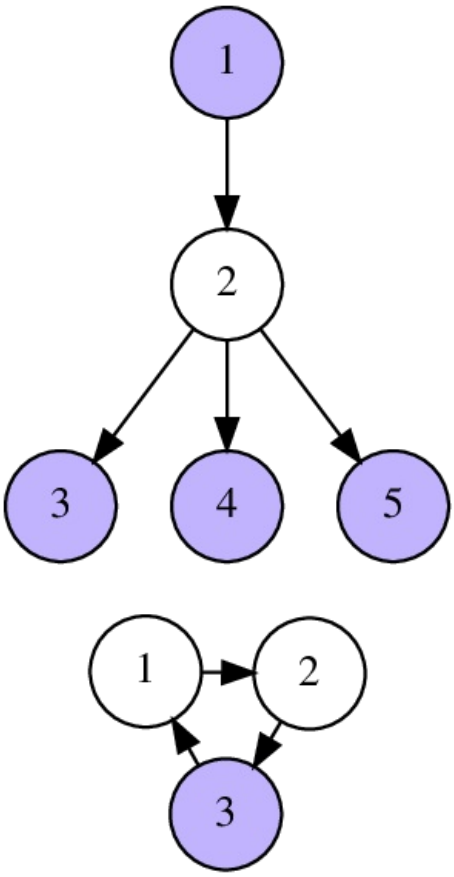
input
3 3 1 2 2 3 3 1

output
1 3

Note
In the first sample, the vertices $1, 3, 4, 5$ are not connected. The vertex 2 is reachable from vertex 1 by one edge.

In the second sample, it is possible to reach the vertex 1 in one move and the vertex 2 in two moves.

The following pictures illustrate sample tests and their answers.



D. Large Triangle

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

> There is a strange peculiarity: if you connect the cities of Rostov, Taganrog and Shakhty, peculiarly, you get a triangle
>
> «Unbelievable But True»

Students from many different parts of Russia and abroad come to Summer Informatics School. You marked the hometowns of the SIS participants on a map.

Now you decided to prepare an interesting infographic based on this map. The first thing you chose to do is to find three cities on this map, such that they form a triangle with area S.

Input
The first line of input contains two integers n and S ($3 \le n \le 2000$, $1 \le S \le 2 \cdot 10^{18}$) — the number of cities on the map and the area of the triangle to be found.

The next n lines contain descriptions of the cities, one per line. Each city is described by its integer coordinates x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$).

It is guaranteed that all cities are located at distinct points. It is also guaranteed that no three cities lie on the same line.

Output
If the solution doesn't exist — print «No».

Otherwise, print «Yes», followed by three pairs of coordinates (x, y) — the locations of the three cities, which form the triangle of area S.

Examples

input
3 7 0 0 3 0

0 4

output
No

input
4 3 0 0 2 0 1 2 1 3

output
Yes 0 0 1 3 2 0

E. Raining season

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

By the year 3018, Summer Informatics School has greatly grown. Hotel «Berendeetronik» has been chosen as a location of the school. The camp consists of n houses with $n - 1$ pathways between them. It is possible to reach every house from each other using the pathways.

Everything had been perfect until the rains started. The weather forecast promises that rains will continue for m days. A special squad of teachers was able to measure that the i-th pathway, connecting houses u_i and v_i, before the rain could be passed in b_i seconds. Unfortunately, the rain erodes the roads, so with every day the time to pass the road will increase by a_i seconds. In other words, on the t-th (from zero) day after the start of the rain, it will take $a_i \cdot t + b_i$ seconds to pass through this road.

Unfortunately, despite all the efforts of teachers, even in the year 3018 not all the students are in their houses by midnight. As by midnight all students have to go to bed, it is important to find the maximal time between all the pairs of houses for each day, so every student would know the time when he has to run to his house.

Find all the maximal times of paths between every pairs of houses after $t = 0$, $t = 1$, ..., $t = m - 1$ days.

Input

In the first line you are given two integers n and m — the number of houses in the camp and the number of raining days ($1 \le n \le 100\,000$; $1 \le m \le 1\,000\,000$).

In the next $n - 1$ lines you are given the integers u_i, v_i, a_i, b_i — description of pathways ($1 \le u_i, v_i \le n$; $0 \le a_i \le 10^5$; $0 \le b_i \le 10^9$). i-th pathway connects houses u_i and v_i, and in day t requires $a_i \cdot t + b_i$ seconds to pass through.

It is guaranteed that every two houses are connected by a sequence of pathways.

Output

Print m integers — the lengths of the longest path in the camp after a $t = 0, t = 1, \ldots, t = m - 1$ days after the start of the rain.

Example

input
5 10 1 2 0 100 1 3 0 100 1 4 10 80 1 5 20 0

output
200 200 200 210 220 230 260 290 320 350

Note

Let's consider the first example.

In the first three days ($0 \le t \le 2$) the longest path is between 2nd and 3rd houses, and its length is equal to $100 + 100 = 200$ seconds.

In the third day ($t = 2$) the road between houses 1 and 4 has length 100 and keeps increasing. So, in days $t = 2, 3, 4, 5$ the longest path is between vertices 4 and (1 or 2), and has length $180 + 10t$. Notice, that in the day $t = 2$ there are three pathways with length 100, so there are three maximal paths of equal length.

In the sixth day ($t = 5$) pathway between first and fifth houses get length 100. So in every day with $t = 5$ and further the longest path is between houses 4 and 5 and has length $80 + 30t$.