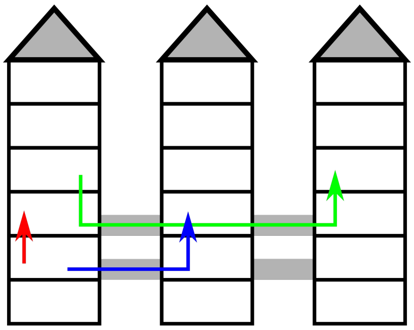
A. New Building for SIS

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

You are looking at the floor plan of the Summer Informatics School's new building. You were tasked with SIS logistics, so you really care about travel time between different locations: it is important to know how long it would take to get from the lecture room to the canteen, or from the gym to the server room.

The building consists of n towers, h floors each, where the towers are labeled from 1 to n, the floors are labeled from 1 to h. There is a passage between any two adjacent towers (two towers i and $i+1$ for all i: $1 \le i \le n-1$) on every floor x, where $a \le x \le b$. It takes exactly one minute to walk between any two adjacent floors of a tower, as well as between any two adjacent towers, provided that there is a passage on that floor. It is not permitted to leave the building.



The picture illustrates the first example.

You have given k pairs of locations (t_a, f_a), (t_b, f_b): floor f_a of tower t_a and floor f_b of tower t_b. For each pair you need to determine the minimum walking time between these locations.

Input

The first line of the input contains following integers:

- n: the number of towers in the building ($1 \le n \le 10^8$),
- h: the number of floors in each tower ($1 \le h \le 10^8$),
- a and b: the lowest and highest floor where it's possible to move between adjacent towers ($1 \le a \le b \le h$),
- k: total number of queries ($1 \le k \le 10^4$).

Next k lines contain description of the queries. Each description consists of four integers t_a, f_a, t_b, f_b ($1 \le t_a, t_b \le n$, $1 \le f_a, f_b \le h$). This corresponds to a query to find the minimum travel time between f_a-th floor of the t_a-th tower and f_b-th floor of the t_b-th tower.

Output

For each query print a single integer: the minimum walking time between the locations in minutes.

Example

input

```
3 6 2 3 3
1 2 1 3
1 4 3 4
1 2 2 3
```

output

```
1
4
2
```

B. Badge

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

In Summer Informatics School, if a student doesn't behave well, teachers make a hole in his badge. And today one of the teachers caught a group of n students doing yet another trick.

Let's assume that all these students are numbered from 1 to n. The teacher came to student a and put a hole in his badge. The student, however, claimed that the main culprit is some other student p_a.

After that, the teacher came to student p_a and made a hole in his badge as well. The student in reply said that the main culprit was student p_{p_a}.

This process went on for a while, but, since the number of students was finite, eventually the teacher came to the student, who already had a hole in his badge.

After that, the teacher put a second hole in the student's badge and decided that he is done with this process, and went to the sauna.

You don't know the first student who was caught by the teacher. However, you know all the numbers p_i. Your task is to find out for every student a, who would be the student with two holes in the badge if the first caught student was a.

Input

The first line of the input contains the only integer n ($1 \le n \le 1000$) — the number of the naughty students.

The second line contains n integers p_1, ..., p_n ($1 \le p_i \le n$), where p_i indicates the student who was reported to the teacher by student i.

Output

For every student a from 1 to n print which student would receive two holes in the badge, if a was the first student caught by the teacher.

Examples

input

```
3
2 3 2
```

output

```
2 2 3
```

input

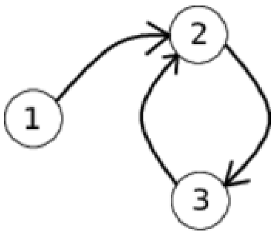
```
3
1 2 3
```

output

```
1 2 3
```

Note

The picture corresponds to the first example test case.



When $a = 1$, the teacher comes to students 1, 2, 3, 2, in this order, and the student 2 is the one who receives a second hole in his badge.

When $a = 2$, the teacher comes to students 2, 3, 2, and the student 2 gets a second hole in his badge. When $a = 3$, the teacher will visit students 3, 2, 3 with student 3 getting a second hole in his badge.

For the second example test case it's clear that no matter with whom the teacher starts, that student would be the one who gets the second hole in his badge.

C. 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.

D. 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 $1, 2, 3, 2, 1, 0$.

E. 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

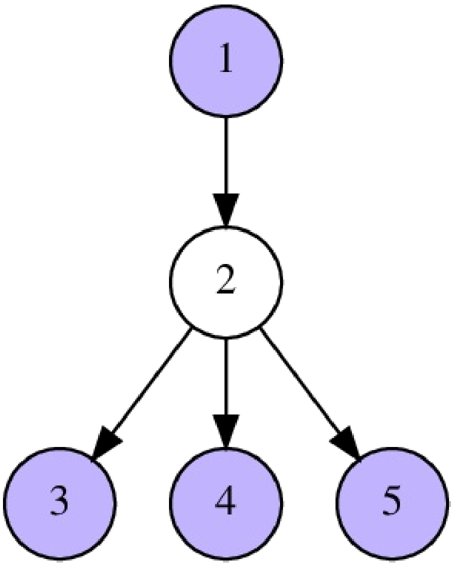
```
1
3
```

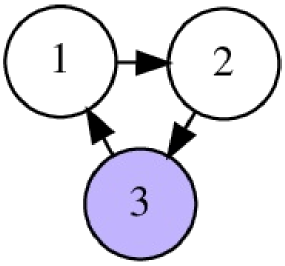
Note

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.


1
2
3
4
5


1
2
3