A. Place Your Ad Here

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

Ivan Anatolyevich's agency is starting to become famous in the town.

They have already ordered and made n TV commercial videos. Each video is made in a special way: the colors and the soundtrack are adjusted to the time of the day and the viewers' mood. That's why the i-th video can only be shown within the time range of $[l_i, r_i]$ (it is not necessary to use the whole segment but the broadcast time should be within this segment).

Now it's time to choose a TV channel to broadcast the commercial. Overall, there are m TV channels broadcasting in the city, the j-th one has c_j viewers, and is ready to sell time $[a_j, b_j]$ to broadcast the commercial.

Ivan Anatolyevich is facing a hard choice: he has to choose exactly one video i and exactly one TV channel j to broadcast this video and also a time range to broadcast $[x, y]$. At that the time range should be chosen so that it is both within range $[l_i, r_i]$ and within range $[a_j, b_j]$.

Let's define the *efficiency* of the broadcast as value $(y - x) \cdot c_j$ — the total sum of time that all the viewers of the TV channel are going to spend watching the commercial. Help Ivan Anatolyevich choose the broadcast with the maximum *efficiency*!

Input

The first line contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$) — the number of commercial videos and channels, respectively.

Each of the following n lines contains two integers l_i, r_i ($0 \le l_i \le r_i \le 10^9$) — the segment of time when it is possible to show the corresponding video.

Each of the following m lines contains three integers a_j, b_j, c_j ($0 \le a_j \le b_j \le 10^9$, $1 \le c_j \le 10^9$), characterizing the TV channel.

Output

In the first line print an integer — the maximum possible *efficiency* of the broadcast. If there is no correct way to get a strictly positive *efficiency*, print a zero.

If the maximum *efficiency* is **strictly** positive, in the second line also print the number of the video i ($1 \le i \le n$) and the number of the TV channel j ($1 \le j \le m$) in the most effective broadcast.

If there are multiple optimal answers, you can print any of them.

Examples

input
2 3 7 9 1 4 2 8 2 0 4 1 8 9 3
output
4 2 1

input
1 1 0 0 1 1 10
output
0

Note

In the first sample test the most optimal solution is to show the second commercial using the first TV channel at time $[2, 4]$. The efficiency of such solution is equal to $(4 - 2) \cdot 2 = 4$.

In the second sample test Ivan Anatolievich's wish does not meet the options of the TV channel, the segments do not intersect, so

the answer is zero.

B. Duck Hunt

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

A duck hunter is doing his favorite thing, hunting. He lives in a two dimensional world and is located at point $(0, 0)$. As he doesn't like walking for his prey, he prefers to shoot only vertically up (because in this case, the ducks fall straight into his hands). The hunter doesn't reload the gun immediately — r or more seconds must pass between the shots. When the hunter shoots up, the bullet immediately hits all the ducks who are directly above the hunter.

In a two dimensional world each duck is a horizontal segment that moves horizontally in the negative direction of the Ox axis at the speed 1 length unit per second. For each duck we know the values h_i and t_i — the x-coordinates of its head (the left end of the segment) and its tail (the right end of the segment) at time 0. The height where the duck is flying isn't important as the gun shoots vertically up to the infinite height and hits all the ducks on its way.

The figure to the first sample.

What maximum number of ducks can the hunter shoot? The duck is considered shot by the hunter if at the moment of the shot at least one of its point intersects the Oy axis. After the hunter shoots the duck, it falls and it can't be shot anymore. The hunter cannot make shots before the moment of time 0.

Input

The first line of the input contains integers n, r ($1 \le n \le 200\,000$, $1 \le r \le 10^9$) — the number of ducks and the minimum time in seconds between the shots.

Then n lines follow, each of them contains two integers h_i, t_i ($-10^9 \le h_i < t_i \le 10^9$) — the x-coordinate of the head and tail of the i-th duck at the moment 0.

Output

Print a single integer — the maximum number of ducks that can be shot by the hunter.

Examples

input
3 3 -3 0 1 3 -1 2
output
3

input
4 5 -1 1 2 4 5 9 6 8
output
3

Note

In the first sample the hunter must shoot at time 0, this shot kills ducks 1 and 3. Then the hunter needs to reload the gun and shoot again at time 3. His second shot hits the tail of duck 2.

In the second sample the hunter can make shots at times 0 and 6 to hit three ducks.

C. Idempotent functions

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

Some time ago Leonid have known about *idempotent functions*. *Idempotent function* defined on a set $\{1, 2, ..., n\}$ is such function $g : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$, that for any $x \in \{1, 2, \ldots, n\}$ the formula $g(g(x)) = g(x)$ holds.

Let's denote as $f^{(k)}(x)$ the function f applied k times to the value x. More formally, $f^{(1)}(x) = f(x)$, $f^{(k)}(x) = f(f^{(k-1)}(x))$ for each $k > 1$.

You are given some function $f : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, n\}$. Your task is to find minimum positive integer k such that function $f^{(k)}(x)$ is *idempotent*.

Input

In the first line of the input there is a single integer n ($1 \le n \le 200$) — the size of function f domain.

In the second line follow $f(1), f(2), ..., f(n)$ ($1 \le f(i) \le n$ for each $1 \le i \le n$), the values of a function.

Output

Output minimum k such that function $f^{(k)}(x)$ is *idempotent*.

Examples

input
4 1 2 2 4
output
1

input
3 2 3 3
output
2

input
3 2 3 1
output
3

Note

In the first sample test function $f(x) = f^{(1)}(x)$ is already idempotent since $f(f(1)) = f(1) = 1$, $f(f(2)) = f(2) = 2$, $f(f(3)) = f(3) = 2$, $f(f(4)) = f(4) = 4$.

In the second sample test:

- function $f(x) = f^{(1)}(x)$ isn't idempotent because $f(f(1)) = 3$ but $f(1) = 2$;
- function $f(x) = f^{(2)}(x)$ is idempotent since for any x it is true that $f^{(2)}(x) = 3$, so it is also true that $f^{(2)}(f^{(2)}(x)) = 3$.

In the third sample test:

- function $f(x) = f^{(1)}(x)$ isn't idempotent because $f(f(1)) = 3$ but $f(1) = 2$;
- function $f(f(x)) = f^{(2)}(x)$ isn't idempotent because $f^{(2)}(f^{(2)}(1)) = 2$ but $f^{(2)}(1) = 3$;
- function $f(f(f(x))) = f^{(3)}(x)$ is idempotent since it is identity function: $f^{(3)}(x) = x$ for any $x \in \{1, 2, 3\}$ meaning that the formula $f^{(3)}(f^{(3)}(x)) = f^{(3)}(x)$ also holds.

D. Superhero's Job

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

It's tough to be a superhero. And it's twice as tough to resist the supervillain who is cool at math. Suppose that you're an ordinary Batman in an ordinary city of Gotham. Your enemy Joker mined the building of the city administration and you only have several minutes to neutralize the charge. To do that you should enter the cancel code on the bomb control panel.

However, that mad man decided to give you a hint. This morning the mayor found a playing card under his pillow. There was a line written on the card:

The bomb has a note saying "$J(x) = A$", where A is some positive integer. You suspect that the cancel code is some integer x that meets the equation $J(x) = A$. Now in order to decide whether you should neutralize the bomb or run for your life, you've got to count how many distinct positive integers x meet this equation.

Input

The single line of the input contains a single integer A ($1 \le A \le 10^{12}$).

Output

Print the number of solutions of the equation $J(x) = A$.

Examples

input
3
output
1

input
24
output
3

Note

Record $x|n$ means that number n divides number x.

$\gcd(a, b)$ is defined as the largest positive integer that divides both a and b.

In the first sample test the only suitable value of x is 2. Then $J(2) = 1 + 2$.

In the second sample test the following values of x match:

- $x = 14, J(14) = 1 + 2 + 7 + 14 = 24$
- $x = 15, J(15) = 1 + 3 + 5 + 15 = 24$
- $x = 23, J(23) = 1 + 23 = 24$

E. Playing on Graph

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

Vova and Marina love offering puzzles to each other. Today Marina offered Vova to cope with the following task.

Vova has a non-directed graph consisting of n vertices and m edges without loops and multiple edges. Let's define the operation of *contraction* two vertices a and b that are **not connected by an edge**. As a result of this operation vertices a and b are deleted and instead of them a new vertex x is added into the graph, and also edges are drawn from it to all vertices that were connected with a or with b (specifically, if the vertex was connected with both a and b, then also exactly one edge is added from x to it). Thus, as a result of *contraction* again a non-directed graph is formed, it contains no loops nor multiple edges, and it contains $(n - 1)$ vertices.

Vova must perform the *contraction* an arbitrary number of times to transform the given graph into a *chain* of the maximum length. A *chain* of length k ($k \geq 0$) is a connected graph whose vertices can be numbered with integers from 1 to $k + 1$ so that the edges of the graph connect all pairs of vertices $(i, i + 1)$ ($1 \leq i \leq k$) and only them. Specifically, the graph that consists of one vertex is a chain of length 0. The vertices that are formed as a result of the *contraction* are allowed to be used in the following operations of *contraction*.

The picture illustrates the contraction of two vertices marked by red.

Help Vova cope with his girlfriend's task. Find the maximum length of the chain that can be obtained from the resulting graph or else determine that it is impossible to obtain the chain.

Input

The first line contains two integers n, m ($1 \leq n \leq 1000, 0 \leq m \leq 100\,000$) — the number of vertices and the number of edges in the original graph.

Next m lines contain the descriptions of edges in the format a_i, b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$), which means that there is an edge between vertices a_i and b_i. It is guaranteed that there is at most one edge between each pair of vertexes.

Output

If it is impossible to obtain a chain from the given graph, print -1. Otherwise, print the maximum possible number of edges in the resulting chain.

Examples

input
5 4 1 2 2 3 3 4 3 5
output
3

input
4 6 1 2 2 3 1 3 3 4 2 4 1 4
output
-1

input
4 2 1 3 2 4
output
2

Note

In the first sample test you can contract vertices 4 and 5 and obtain a chain of length 3.

In the second sample test it is initially impossible to contract any pair of vertexes, so it is impossible to achieve the desired result.

In the third sample test you can contract vertices 1 and 2 and obtain a chain of length 2.

F. Quest

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

Polycarp is making a quest for his friends. He has already made n tasks, for each task the boy evaluated how interesting it is as an integer q_i, and the time t_i in minutes needed to complete the task.

An interesting feature of his quest is: each participant should get the task that is best suited for him, depending on his preferences. The task is chosen based on an interactive quiz that consists of some questions. The player should answer these questions with "yes" or "no". Depending on the answer to the question, the participant either moves to another question or goes to one of the tasks that are in the quest. In other words, the quest is a binary tree, its nodes contain questions and its leaves contain tasks.

We know that answering any of the questions that are asked before getting a task takes exactly one minute from the quest player. Polycarp knows that his friends are busy people and they can't participate in the quest for more than T minutes. Polycarp wants to choose some of the n tasks he made, invent the corresponding set of questions for them and use them to form an interactive quiz as a binary tree so that no matter how the player answers quiz questions, he spends at most T minutes on completing the whole quest (that is, answering all the questions and completing the task). Specifically, the quest can contain zero questions and go straight to the task. Each task can only be used once (i.e., the people who give different answers to questions should get different tasks).

Polycarp wants the total "interest" value of the tasks involved in the quest to be as large as possible. Help him determine the maximum possible total interest value of the task considering that the quest should be completed in T minutes at any variant of answering questions.

Input

The first line contains two integers n and T ($1 \le n \le 1000$, $1 \le T \le 100$) — the number of tasks made by Polycarp and the maximum time a quest player should fit into.

Next n lines contain two integers t_i, q_i ($1 \le t_i \le T$, $1 \le q_i \le 1000$) each — the time in minutes needed to complete the i-th task and its interest value.

Output

Print a single integer — the maximum possible total interest value of all the tasks in the quest.

Examples

input
5 5 1 1 1 1 2 2 3 3 4 4
output
11

input
5 5 4 1 4 2 4 3 4 4 4 5
output
9

input
2 2 1 1 2 10
output
10

Note

In the first sample test all the five tasks can be complemented with four questions and joined into one quest.

In the second sample test it is impossible to use all the five tasks, but you can take two of them, the most interesting ones.

In the third sample test the optimal strategy is to include only the second task into the quest.

Here is the picture that illustrates the answers to the sample tests. The blue circles represent the questions, the two arrows that go from every circle represent where a person goes depending on his answer to that question. The tasks are the red ovals.