A. Peculiar apple-tree

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

In Arcady's garden there grows a peculiar apple-tree that fruits one time per year. Its peculiarity can be explained in following way: there are n inflorescences, numbered from 1 to n. Inflorescence number 1 is situated near base of tree and any other inflorescence with number i ($i > 1$) is situated at the top of branch, which bottom is p_i-th inflorescence and $p_i < i$.

Once tree starts fruiting, there appears exactly one apple in each inflorescence. The same moment as apples appear, they start to roll down along branches to the very base of tree. Each second all apples, except ones in first inflorescence simultaneously roll down one branch closer to tree base, e.g. apple in a-th inflorescence gets to p_a-th inflorescence. Apples that end up in first inflorescence are gathered by Arcady in exactly the same moment. Second peculiarity of this tree is that once two apples are in same inflorescence they **annihilate**. This happens with each pair of apples, e.g. if there are 5 apples in same inflorescence in same time, only one will not be annihilated and if there are 8 apples, all apples will be annihilated. Thus, there can be no more than one apple in each inflorescence in each moment of time.

Help Arcady with counting number of apples he will be able to collect from first inflorescence during one harvest.

Input

First line of input contains single integer number n ($2 \le n \le 100\ 000$) — number of inflorescences.

Second line of input contains sequence of $n - 1$ integer numbers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$), where p_i is number of inflorescence into which the apple from i-th inflorescence rolls down.

Output

Single line of output should contain one integer number: amount of apples that Arcady will be able to collect from first inflorescence during one harvest.

Examples

input

```
3
1 1
```

output

```
1
```

input

```
5
1 2 2 2
```

output

```
3
```

input

```
18
1 1 1 4 4 3 2 2 2 10 8 9 9 9 10 10 4
```

output

```
4
```

Note

In first example Arcady will be able to collect only one apple, initially situated in 1st inflorescence. In next second apples from 2nd and 3rd inflorescences will roll down and annihilate, and Arcady won't be able to collect them.

In the second example Arcady will be able to collect 3 apples. First one is one initially situated in first inflorescence. In a second apple from 2nd inflorescence will roll down to 1st (Arcady will collect it) and apples from 3rd, 4th, 5th inflorescences will roll down to 2nd. Two of them will annihilate and one not annihilated will roll down from 2-nd inflorescence to 1st one in the next second and Arcady will collect it.

B. Game with String

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

output: standard output

Vasya and Kolya play a game with a string, using the following rules. Initially, Kolya creates a string s, consisting of small English letters, and uniformly at random chooses an integer k from a segment $[0, len(s) - 1]$. He tells Vasya this string s, and then shifts it k letters to the left, i. e. creates a new string $t = s_{k+1}s_{k+2}\dots s_ns_1s_2\dots s_k$. Vasya does not know the integer k nor the string t, but he wants to guess the integer k. To do this, he asks Kolya to tell him the first letter of the new string, and then, after he sees it, open one more letter on some position, which Vasya can choose.

Vasya understands, that he can't guarantee that he will win, but he wants to know the probability of winning, if he plays optimally. He wants you to compute this probability.

Note that Vasya wants to know the value of k uniquely, it means, that if there are at least two cyclic shifts of s that fit the information Vasya knowns, Vasya loses. Of course, at any moment of the game Vasya wants to maximize the probability of his win.

Input

The only string contains the string s of length l $(3 \le l \le 5000)$, consisting of small English letters only.

Output

Print the only number — the answer for the problem. You answer is considered correct, if its absolute or relative error does not exceed 10^{-6}.

Formally, let your answer be a, and the jury's answer be b. Your answer is considered correct if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$

Examples

input
```
technocup
```
output
```
1.000000000000000
```

input
```
tictictactac
```
output
```
0.333333333333333
```

input
```
bbaabaabbb
```
output
```
0.100000000000000
```

Note

In the first example Vasya can always open the second letter after opening the first letter, and the cyclic shift is always determined uniquely.

In the second example if the first opened letter of t is "t" or "c", then Vasya can't guess the shift by opening only one other letter. On the other hand, if the first letter is "i" or "a", then he can open the fourth letter and determine the shift uniquely.

C. Teodor is not a liar!

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

Young Teodor enjoys drawing. His favourite hobby is drawing segments with integer borders inside his huge $[1;m]$ segment. One day Teodor noticed that picture he just drawn has one interesting feature: there doesn't exist an integer point, that belongs each of segments in the picture. Having discovered this fact, Teodor decided to share it with Sasha.

Sasha knows that Teodor likes to show off so he never trusts him. Teodor wants to prove that he can be trusted sometimes, so he decided to convince Sasha that there is no such integer point in his picture, which belongs to each segment. However Teodor is lazy person and neither wills to tell Sasha all coordinates of segments' ends nor wills to tell him their amount, so he suggested Sasha to ask him series of questions 'Given the integer point x_i, how many segments in Fedya's picture contain that point?', promising to tell correct answers for this questions.

Both boys are very busy studying and don't have much time, so they ask you to find out how many questions can Sasha ask Teodor, that having only answers on his questions, Sasha can't be sure that Teodor isn't lying to him. Note that Sasha doesn't know amount of segments in Teodor's picture. Sure, Sasha is smart person and never asks about same point twice.

Input

First line of input contains two integer numbers: n and m $(1 \le n, m \le 100\,000)$ — amount of segments of Teodor's picture and maximal coordinate of point that Sasha can ask about.

ith of next n lines contains two integer numbers l_i and r_i $(1 \le l_i \le r_i \le m)$ — left and right ends of ith segment in the picture. Note that that left and right ends of segment can be the same point.

It is guaranteed that there is no integer point, that belongs to all segments.

Output

Single line of output should contain one integer number k – size of largest set $(x_i, cnt(x_i))$ where all x_i are different, $1 \le x_i \le m$, and $cnt(x_i)$ is amount of segments, containing point with coordinate x_i, such that one can't be sure that there doesn't exist point, belonging to all of segments in initial picture, if he knows only this set(and doesn't know n).

Examples

input
```
2 4
1 2
3 4
```
output
```
4
```

input
```
4 6
1 3
2 3
4 6
5 6
```
output
```
5
```

Note

First example shows situation where Sasha can never be sure that Teodor isn't lying to him, because even if one knows $cnt(x_i)$ for each point in segment $[1;4]$, he can't distinguish this case from situation Teodor has drawn whole $[1;4]$ segment.

In second example Sasha can ask about 5 points e.g. $1, 2, 3, 5, 6$, still not being sure if Teodor haven't lied to him. But once he knows information about all points in $[1;6]$ segment, Sasha can be sure that Teodor haven't lied to him.

D. Game with Tokens

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

Consider the following game for two players. There is one white token and some number of black tokens. Each token is placed on a plane in a point with integer coordinates x and y.

The players take turn making moves, white starts. On each turn, a player moves **all** tokens of their color by 1 to up, down, left or right. Black player can choose directions for each token independently.

After a turn of the white player the white token can not be in a point where a black token is located. There are no other constraints on locations of the tokens: positions of black tokens can coincide, after a turn of the black player and initially the white token can be in the same point with some black point. If at some moment the white player can't make a move, he loses. If the white player makes 10^{100500} moves, he wins.

You are to solve the following problem. You are given initial positions of all black tokens. It is guaranteed that initially all these positions are distinct. In how many places can the white token be located initially so that if both players play optimally, the black player wins?

Input

The first line contains a single integer n $(1 \le n \le 10^5)$ — the number of black points.

The $(i+1)$-th line contains two integers x_i, y_i $(-10^5 \le x_i, y_i, \le 10^5)$ — the coordinates of the point where the i-th black token is initially located.

It is guaranteed that initial positions of black tokens are distinct.

Output

Print the number of points where the white token can be located initially, such that if both players play optimally, the black player wins.

Examples

input
```
4
-2 -1
0 1
0 -3
2 -1
```
output
```
4
```

input

```
4
-2 0
-1 1
0 -2
1 -1
```

output

```
2
```

input

```
16
2 1
1 2
-1 1
0 1
0 0
1 1
2 -1
2 0
1 0
-1 -1
1 -1
2 2
0 -1
-1 0
0 2
-1 2
```

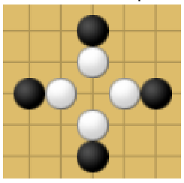
output

```
4
```

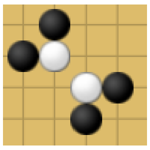
Note

In the first and second examples initial positions of black tokens are shown with black points, possible positions of the white token (such that the black player wins) are shown with white points.

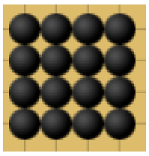
The first example:



The second example:



In the third example the white tokens should be located in the inner square 2×2, to make the black player win.



E. Coins Exhibition

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

Arkady and Kirill visited an exhibition of rare coins. The coins were located in a row and enumerated from left to right from 1 to k, each coin either was laid with its obverse (front) side up, or with its reverse (back) side up.

Arkady and Kirill made some photos of the coins, each photo contained a segment of neighboring coins. Akrady is interested in obverses, so on each photo made by him there is at least one coin with obverse side up. On the contrary, Kirill is interested in reverses, so on each photo made by him there is at least one coin with its reverse side up.

The photos are lost now, but Arkady and Kirill still remember the bounds of the segments of coins each photo contained. Given this information, compute the remainder of division by $10^9 + 7$ of the number of ways to choose the upper side of each coin in such a way, that on each Arkady's photo there is at least one coin with obverse side up, and on each Kirill's photo there is at least one coin with reverse side up.

Input

The first line contains three integers k, n and m ($1 \leq k \leq 10^9$, $0 \leq n, m \leq 10^5$) — the total number of coins, the number of photos made by Arkady, and the number of photos made by Kirill, respectively.

The next n lines contain the descriptions of Arkady's photos, one per line. Each of these lines contains two integers l and r ($1 \le l \le r \le k$), meaning that among coins from the l-th to the r-th there should be at least one with obverse side up.

The next m lines contain the descriptions of Kirill's photos, one per line. Each of these lines contains two integers l and r ($1 \le l \le r \le k$), meaning that among coins from the l-th to the r-th there should be at least one with reverse side up.

Output

Print the only line — the number of ways to choose the side for each coin modulo $10^9 + 7 = 1000000007$.

Examples

input

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

output

```
8
```

input

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

output

```
0
```

input

```
60 5 7
1 3
50 60
1 60
30 45
20 40
4 5
6 37
5 18
50 55
22 27
25 31
44 45
```

output

```
732658600
```

Note

In the first example the following ways are possible ('O' — obverse, 'R' — reverse side):

- OROOR,
- ORORO,
- ORORR,
- RROOR,
- RRORO,
- RRORR,
- ORROR,
- ORRRO.

In the second example the information is contradictory: the second coin should have obverse and reverse sides up at the same time, that is impossible. So, the answer is 0.