Topic Stream Mashup: Probability + Combinatorics

A. Rectangles

1 second, 256 megabytes

You are given $n \times m$ table. Each cell of the table is colored white or black. Find the number of non-empty sets of cells such that:

1. All cells in a set have the same color.
2. Every two cells in a set share row or column.

Input

The first line of input contains integers n and m ($1 \le n, m \le 50$) — the number of rows and the number of columns correspondingly.

The next n lines of input contain descriptions of rows. There are m integers, separated by spaces, in each line. The number equals 0 if the corresponding cell is colored white and equals 1 if the corresponding cell is colored black.

Output

Output single integer — the number of non-empty sets from the problem description.

input

```
1 1
0
```

output

```
1
```

input

```
2 3
1 0 1
0 1 0
```

output

```
8
```

In the second example, there are six one-element sets. Additionally, there are two two-element sets, the first one consists of the first and the third cells of the first row, the second one consists of the first and the third cells of the second row. To sum up, there are 8 sets.

B. Archer

2 seconds, 256 megabytes

SmallR is an archer. SmallR is taking a match of archer with Zanoes. They try to shoot in the target in turns, and SmallR shoots first. The probability of shooting the target each time is $\frac{a}{b}$ for SmallR while $\frac{c}{d}$ for Zanoes. The one who shoots in the target first should be the winner.

Output the probability that SmallR will win the match.

Input

A single line contains four integers a, b, c, d $(1 \le a, b, c, d \le 1000, 0 < \frac{a}{b} < 1, 0 < \frac{c}{d} < 1)$.

Output

Print a single real number, the probability that SmallR will win the match.

The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

input

```
1 2 1 2
```

output

```
0.666666666667
```

C. Monotonic Renumeration

2 seconds, 256 megabytes

You are given an array a consisting of n integers. Let's denote *monotonic renumeration* of array a as an array b consisting of n integers such that all of the following conditions are met:

- $b_1 = 0$;
- for every pair of indices i and j such that $1 \le i, j \le n$, if $a_i = a_j$, then $b_i = b_j$ (note that if $a_i \ne a_j$, it is still possible that $b_i = b_j$);
- for every index $i \in [1, n-1]$ either $b_i = b_{i+1}$ or $b_i + 1 = b_{i+1}$.

For example, if $a = [1, 2, 1, 2, 3]$, then two possible monotonic renumerations of a are $b = [0, 0, 0, 0, 0]$ and $b = [0, 0, 0, 0, 1]$.

Your task is to calculate the number of different monotonic renumerations of a. The answer may be large, so print it modulo 998244353.

Input

The first line contains one integer n ($2 \le n \le 2 \cdot 10^5$) — the number of elements in a.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^9$).

Output

Print one integer — the number of different monotonic renumerations of a, taken modulo 998244353.

input

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

output

```
2
```

input

```
2
100 1
```

output

```
2
```

input

```
4
1 3 3 7
```

output

```
4
```

D. Wet Shark and Flowers

2 seconds, 256 megabytes

There are n sharks who grow flowers for Wet Shark. They are all sitting around the table, such that sharks i and $i + 1$ are neighbours for all i from 1 to $n - 1$. Sharks n and 1 are neighbours too.

Each shark will grow some number of flowers s_i. For i-th shark value s_i is random integer equiprobably chosen in range from l_i to r_i. Wet Shark has it's favourite prime number p, and he really likes it! If for any pair of **neighbouring** sharks i and j the product $s_i \cdot s_j$ is divisible by p, then Wet Shark becomes happy and gives 1000 dollars to each of these sharks.

At the end of the day sharks sum all the money Wet Shark granted to them. Find the expectation of this value.

Input

The first line of the input contains two space-separated integers n and p ($3 \le n \le 100\,000, 2 \le p \le 10^9$) — the number of sharks and Wet Shark's favourite prime number. It is guaranteed that p is prime.

The i-th of the following n lines contains information about i-th shark — two space-separated integers l_i and r_i ($1 \le l_i \le r_i \le 10^9$), the range of flowers shark i can produce. Remember that s_i is chosen equiprobably among all integers from l_i to r_i, inclusive.

Output

Print a single real number — the expected number of dollars that the sharks receive in total. You answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1, b)} \le 10^{-6}$.

input

```
3 2
1 2
420 421
420420 420421
```

output

```
4500.0
```

input

```
3 5
1 4
2 3
11 14
```

output

```
0.0
```

A prime number is a positive integer number that is divisible only by 1 and itself. 1 is not considered to be prime.

Consider the first sample. First shark grows some number of flowers from 1 to 2, second sharks grows from 420 to 421 flowers and third from 420420 to 420421. There are eight cases for the quantities of flowers (s_0, s_1, s_2) each shark grows:

1. $(1, 420, 420420)$: note that $s_0 \cdot s_1 = 420$, $s_1 \cdot s_2 = 176576400$, and $s_2 \cdot s_0 = 420420$. For each pair, 1000 dollars will be awarded to each shark. Therefore, each shark will be awarded 2000 dollars, for a total of 6000 dollars.
2. $(1, 420, 420421)$: now, the product $s_2 \cdot s_0$ is not divisible by 2. Therefore, sharks s_0 and s_2 will receive 1000 dollars, while shark s_1 will receive 2000. The total is 4000.
3. $(1, 421, 420420)$: total is 4000
4. $(1, 421, 420421)$: total is 0.
5. $(2, 420, 420420)$: total is 6000.
6. $(2, 420, 420421)$: total is 6000.
7. $(2, 421, 420420)$: total is 6000.
8. $(2, 421, 420421)$: total is 4000.

The expected value is $\frac{6000+4000+4000+0+6000+6000+6000+4000}{8} = 4500$.

In the second sample, no combination of quantities will garner the sharks any money.

E. Ilya and Escalator

2 seconds, 256 megabytes

Ilya got tired of sports programming, left university and got a job in the subway. He was given the task to determine the escalator load factor.

Let's assume that n people stand in the queue for the escalator. At each second one of the two following possibilities takes place: either the first person in the queue enters the escalator with probability p, or the first person in the queue doesn't move with probability $(1 - p)$, paralyzed by his fear of escalators and making the whole queue wait behind him.

Formally speaking, the i-th person in the queue cannot enter the escalator until people with indices from 1 to $i - 1$ inclusive enter it. In one second only one person can enter the escalator. The escalator is infinite, so if a person enters it, he never leaves it, that is he will be standing on the escalator at any following second. Ilya needs to count the expected value of the number of people standing on the escalator after t seconds.

Your task is to help him solve this complicated task.

Input

The first line of the input contains three numbers n, p, t ($1 \le n, t \le 2000$, $0 \le p \le 1$). Numbers n and t are integers, number p is real, given with exactly two digits after the decimal point.

Output

Print a single real number — the expected number of people who will be standing on the escalator after t seconds. The absolute or relative error mustn't exceed 10^{-6}.

input

```
1 0.50 1
```

output

```
0.5
```

input

```
1 0.50 4
```

output

```
0.9375
```

input

```
4 0.20 2
```

output

```
0.4
```

F. The Intriguing Obsession

1 second, 256 megabytes

— This is not playing but duty as allies of justice, Nii-chan!

— Not allies but justice itself, Onii-chan!

With hands joined, go everywhere at a speed faster than our thoughts! This time, the Fire Sisters — Karen and Tsukihi — is heading for somewhere they've never reached — water-surrounded islands!

There are three clusters of islands, conveniently coloured red, blue and purple. The clusters consist of a, b and c distinct islands respectively.

Bridges have been built between some (possibly all or none) of the islands. A bridge bidirectionally connects two different islands and has length 1. For any two islands of the same colour, either they shouldn't be reached from each other through bridges, or the shortest distance between them is **at least** 3, apparently in order to prevent oddities from spreading quickly inside a cluster.

The Fire Sisters are ready for the unknown, but they'd also like to test your courage. And you're here to figure out the number of different ways to build all bridges under the constraints, and give the answer modulo $998\,244\,353$. Two ways are considered different if a pair of islands exist, such that there's a bridge between them in one of them, but not in the other.

Input

The first and only line of input contains three space-separated integers a, b and c ($1 \le a, b, c \le 5\,000$) — the number of islands in the red, blue and purple clusters, respectively.

Output

Output one line containing an integer — the number of different ways to build bridges, modulo $998\,244\,353$.

input

```
1 1 1
```

output

```
8
```

input

```
1 2 2
```

output

```
63
```

input

```
1 3 5
```

output

```
3264
```

input

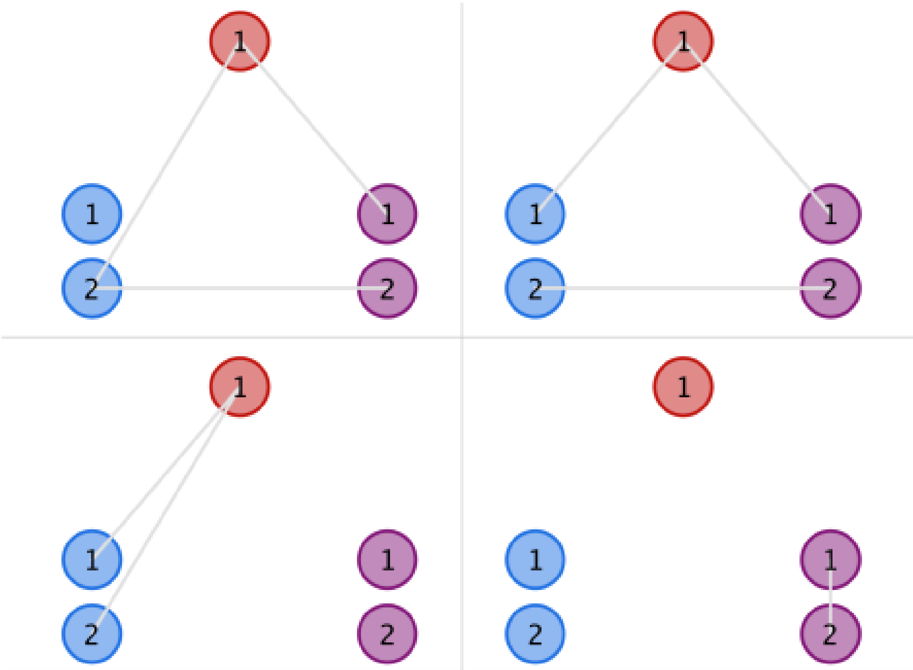
```
6 2 9
```

output

```
813023575
```

In the first example, there are 3 bridges that can possibly be built, and no setup of bridges violates the restrictions. Thus the answer is $2^3 = 8$.

In the second example, the upper two structures in the figure below are instances of valid ones, while the lower two are invalid due to the blue and purple clusters, respectively.



G. Coprime Subsequences

2 seconds, 256 megabytes

Let's call a non-empty sequence of positive integers $a_1, a_2 \ldots a_k$ *coprime* if the greatest common divisor of all elements of this sequence is equal to 1.

Given an array a consisting of n positive integers, find the number of its *coprime* subsequences. Since the answer may be very large, print it modulo $10^9 + 7$.

Note that two subsequences are considered different if chosen indices are different. For example, in the array $[1, 1]$ there are 3 different subsequences: $[1]$, $[1]$ and $[1, 1]$.

Input

The first line contains one integer number n $(1 \le n \le 100000)$.

The second line contains n integer numbers $a_1, a_2 \ldots a_n$ $(1 \le a_i \le 100000)$.

Output

Print the number of *coprime* subsequences of a modulo $10^9 + 7$.

input

```
3
1 2 3
```

output

```
5
```

input

```
4
1 1 1 1
```

output

```
15
```

input

```
7
1 3 5 15 3 105 35
```

output

```
100
```

In the first example *coprime* subsequences are:

1. 1
2. $1, 2$
3. $1, 3$
4. $1, 2, 3$
5. $2, 3$

In the second example all subsequences are *coprime*.

H. Square Subsets

4 seconds, 256 megabytes

Petya was late for the lesson too. The teacher gave him an additional task. For some array a Petya should find the number of different ways to select non-empty subset of elements from it in such a way that their product is equal to a square of some integer.

Two ways are considered different if sets of indexes of elements chosen by these ways are different.

Since the answer can be very large, you should find the answer modulo $10^9 + 7$.

Input

First line contains one integer n $(1 \le n \le 10^5)$ — the number of elements in the array.

Second line contains n integers a_i $(1 \le a_i \le 70)$ — the elements of the array.

Output

Print one integer — the number of different ways to choose some elements so that their product is a square of a certain integer modulo $10^9 + 7$.

input

```
4
1 1 1 1
```

output

```
15
```

input

```
4
2 2 2 2
```

output

```
7
```

input

```
5
1 2 4 5 8
```

output

```
7
```

In first sample product of elements chosen by any way is 1 and $1 = 1^2$. So the answer is $2^4 - 1 = 15$.

In second sample there are six different ways to choose elements so that their product is 4, and only one way so that their product is 16. So the answer is $6 + 1 = 7$.

I. First Digit Law

2 seconds, 256 megabytes

In the probability theory the following paradox called Benford's law is known: "In many lists of random numbers taken from real sources, numbers starting with digit 1 occur much more often than numbers starting with any other digit" (that's the simplest form of the law).

Having read about it on Codeforces, the Hedgehog got intrigued by the statement and wishes to thoroughly explore it. He finds the following similar problem interesting in particular: there are N random variables, the i-th of which can take any integer value from some segment $[L_i;R_i]$ (all numbers from this segment are equiprobable). It means that the value of the i-th quantity can be equal to any integer number from a given interval $[L_i;R_i]$ with probability $1 / (R_i - L_i + 1)$.

The Hedgehog wants to know the probability of the event that the first digits of at least $K\%$ of those values will be equal to one. In other words, let us consider some set of fixed values of these random variables and leave only the first digit (the MSD — most significant digit) of each value. Then let's count how many times the digit 1 is encountered and if it is encountered in at least K per cent of those N values, than such set of values will be called a good one. You have to find the probability that a set of values of the given random variables will be a good one.

Input

The first line contains number N which is the number of random variables $(1 \le N \le 1000)$. Then follow N lines containing pairs of numbers L_i, R_i, each of whom is a description of a random variable. It is guaranteed that $1 \le L_i \le R_i \le 10^{18}$.

The last line contains an integer K ($0 \le K \le 100$).

All the numbers in the input file are integers.

Please, do not use `%lld` specificator to read or write 64-bit integers in C++. It is preffered to use `cin` (also you may use `%I64d`).

Output

Print the required probability. Print the fractional number with such a precision that the relative or absolute error of the result won't exceed 10^{-9}.

input
```
1
1 2
50
```
output
```
0.500000000000000
```

input
```
2
1 2
9 11
50
```
output
```
0.833333333333333
```

J. Mike and Geometry Problem

3 seconds, 256 megabytes

Mike wants to prepare for IMO but he doesn't know geometry, so his teacher gave him an interesting geometry problem. Let's define $f([l, r]) = r - l + 1$ to be the number of integer points in the segment $[l, r]$ with $l \le r$ (say that $f(\varnothing) = 0$). You are given two integers n and k and n closed intervals $[l_i, r_i]$ on OX axis and you have to find:

$$\sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} f([l_{i_1}, r_{i_1}] \cap [l_{i_2}, r_{i_2}] \cap \ldots \cap [l_{i_k}, r_{i_k}]).$$

In other words, you should find the sum of the number of integer points in the intersection of any k of the segments.

As the answer may be very large, output it modulo 1000000007 ($10^9 + 7$).

Mike can't solve this problem so he needs your help. You will help him, won't you?

Input

The first line contains two integers n and k ($1 \le k \le n \le 200\,000$) — the number of segments and the number of segments in intersection groups respectively.

Then n lines follow, the i-th line contains two integers l_i, r_i ($-10^9 \le l_i \le r_i \le 10^9$), describing i-th segment bounds.

Output

Print one integer number — the answer to Mike's problem modulo 1000000007 ($10^9 + 7$) in the only line.

input
```
3 2
1 2
1 3
2 3
```
output
```
5
```

input
```
3 3
1 3
1 3
1 3
```
output
```
3
```

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

In the first example:

$f([1, 2] \cap [1, 3]) = f([1, 2]) = 2$;

$f([1, 2] \cap [2, 3]) = f([2, 2]) = 1$;

$f([1, 3] \cap [2, 3]) = f([2, 3]) = 2$.

So the answer is $2 + 1 + 2 = 5$.