Problem Set 1: Dynamic Programming (15-295 Spr. 2018)

A. Boredom

time limit per test: 2.0 s
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alex doesn't like boredom. That's why whenever he gets bored, he comes up with games. One long winter evening he came up with a game and decided to play it.

Given a sequence a consisting of n integers. The player can make several steps. In a single step he can choose an element of the sequence (let's denote it a_k) and delete it, at that all elements equal to $a_k + 1$ and $a_k - 1$ also must be deleted from the sequence. That step brings a_k points to the player.

Alex is a perfectionist, so he decided to get as many points as possible. Help him.

Input

The first line contains integer n $(1 \leq n \leq 10^5)$ that shows how many numbers are in Alex's sequence.

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \leq a_i \leq 10^5)$.

Output

Print a single integer — the maximum number of points that Alex can earn.

Examples

input

```
2
1 2
```

output

```
2
```

input

```
3
1 2 3
```

output

```
4
```

input

```
9
1 2 1 3 2 2 2 2 3
```

output

```
10
```

Note

Consider the third test example. At first step we need to choose any element equal to 2. After that step our sequence looks like this $[2, 2, 2, 2]$. Then we do 4 steps, on each step we choose any element equals to 2. In total we earn 10 points.

B. Flowers

time limit per test: 2.0 s
memory limit per test: 256 megabytes
input: standard input
output: standard output

We saw the little game Marmot made for Mole's lunch. Now it's Marmot's dinner time and, as we all know, Marmot eats flowers. At every dinner he eats some red and white flowers. Therefore a dinner can be represented as a sequence of several flowers, some of them white and some of them red.

But, for a dinner to be tasty, there is a rule: Marmot wants to eat white flowers only in groups of size k.

Now Marmot wonders in how many ways he can eat between a and b flowers. As the number of ways could be very large, print it modulo 1000000007 ($10^9 + 7$).

Input

Input contains several test cases.

The first line contains two integers t and k ($1 \leq t, k \leq 10^5$), where t represents the number of test cases.

The next t lines contain two integers a_i and b_i ($1 \leq a_i \leq b_i \leq 10^5$), describing the i-th test.

Output

Print t lines to the standard output. The i-th line should contain the number of ways in which Marmot can eat between a_i and b_i flowers at dinner modulo 1000000007 ($10^9 + 7$).

Examples

input

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

output

```
6
5
5
```

Note

- For $K = 2$ and length 1 Marmot can eat (R).
- For $K = 2$ and length 2 Marmot can eat (RR) and (WW).
- For $K = 2$ and length 3 Marmot can eat (RRR), (RWW) and (WWR).
- For $K = 2$ and length 4 Marmot can eat, for example, ($WWWW$) or ($RWWR$), but for example he can't eat ($WWWR$).

C. k-Tree

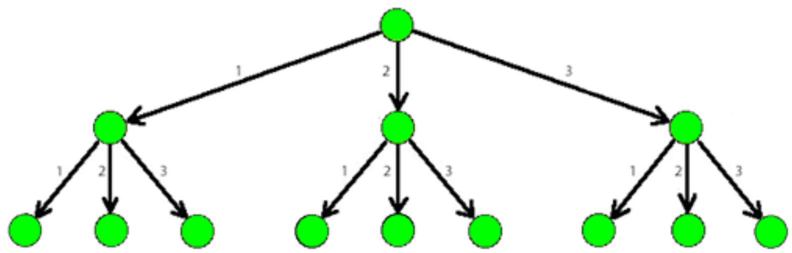
time limit per test: 2.0 s
memory limit per test: 256 megabytes
input: standard input
output: standard output

Quite recently a creative student Lesha had a lecture on trees. After the lecture Lesha was inspired and came up with the tree of his own which he called a k-tree.

A k-tree is an infinite rooted tree where:

- each vertex has exactly k children;
- each edge has some weight;
- if we look at the edges that goes from some vertex to its children (exactly k edges), then their weights will equal $1, 2, 3, ..., k$.

The picture below shows a part of a 3-tree.



As soon as Dima, a good friend of Lesha, found out about the tree, he immediately wondered: "How many paths of total weight n (the sum of all weights of the edges in the path) are there, starting from the root of a k-tree and also containing at least one edge of weight at least d?".
Help Dima find an answer to his question. As the number of ways can be rather large, print it modulo 1000000007 ($10^9 + 7$).

Input

A single line contains three space-separated integers: n, k and d ($1 \leq n, k \leq 100$; $1 \leq d \leq k$).

Output

Print a single integer — the answer to the problem modulo 1000000007 ($10^9 + 7$).

Examples

input
3 3 2
output
3

input
3 3 3
output
1

input
4 3 2
output
6

input
4 5 2
output
7

D. Ilya and Escalator

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

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}.

Examples

input

```
1 0.50 1
```

output

```
0.5
```

input

```
1 0.50 4
```

output

```
0.9375
```

input

```
4 0.20 2
```

output

```
0.4
```

E. Connecting Vertices

time limit per test: 8.0 s
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n points marked on the plane. The points are situated in such a way that they form a regular polygon (marked points are its vertices, and they are numbered in counter-clockwise order). You can draw $n - 1$ segments, each connecting any two marked points, in such a way that all points have to be connected with each other (directly or indirectly).

But there are some restrictions. Firstly, some pairs of points cannot be connected directly and have to be connected undirectly. Secondly, the segments you draw must not intersect in any point apart from the marked points (that is, if any two segments intersect and their intersection is not a marked point, then the picture you have drawn is invalid).

How many ways are there to connect all vertices with $n - 1$ segments? Two ways are considered different iff there exist some pair of points such that a segment is drawn between them in the first way of connection, but it is not drawn between these points in the second one. Since the answer might be large, output it modulo $10^9 + 7$.

Input

The first line contains one number n ($3 \le n \le 500$) — the number of marked points.

Then n lines follow, each containing n elements. $a_{i,j}$ (j-th element of line i) is equal to 1 iff you can connect points i and j directly (otherwise $a_{i,j} = 0$). It is guaranteed that for any pair of points $a_{i,j} = a_{j,i}$, and for any point $a_{i,i} = 0$.

Output

Print the number of ways to connect points modulo $10^9 + 7$.

Examples

input

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

output

```
1
```

input

```
4
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0
```

output

```
12
```

input

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

output

```
0
```

F. Working out

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

Summer is coming! It's time for Iahub and Iahubina to work out, as they both want to look hot at the beach. The gym where they go is a matrix a with n lines and m columns. Let number $a[i][j]$ represents the calories burned by performing workout at the cell of gym in the i-th line and the j-th column.

Iahub starts with workout located at line 1 and column 1. He needs to finish with workout $a[n][m]$. After finishing workout $a[i][j]$, he can go to workout $a[i + 1][j]$ or $a[i][j + 1]$. Similarly, Iahubina starts with workout $a[n][1]$ and she needs to finish with workout $a[1][m]$. After finishing workout from cell $a[i][j]$, she goes to either $a[i][j + 1]$ or $a[i - 1][j]$.

There is one additional condition for their training. They have to meet in exactly one cell of gym. At that cell, none of them will work out. They will talk about fast exponentiation (pretty odd small talk) and then both of them will move to the next workout.

If a workout was done by either Iahub or Iahubina, it counts as total gain. Please plan a workout for Iahub and Iahubina such as total gain to be as big as possible. Note, that Iahub and Iahubina can perform workouts with different speed, so the number of cells that they use to reach meet cell may differs.

Input

The first line of the input contains two integers n and m ($3 \le n, m \le 1000$). Each of the next n lines contains m integers: j-th number from i-th line denotes element $a[i][j]$ ($0 \le a[i][j] \le 10^5$).

Output

The output contains a single number — the maximum total gain possible.

Examples

input

```
3 3
100 100 100
100 1 100
100 100 100
```

output

```
800
```

Note

Iahub will choose exercises $a[1][1] \to a[1][2] \to a[2][2] \to a[3][2] \to a[3][3]$. Iahubina will choose exercises $a[3][1] \to a[2][1] \to a[2][2] \to a[2][3] \to a[1][3]$.