A. Amity Assessment

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

Bessie the cow and her best friend Elsie each received a sliding puzzle on Pi Day. Their puzzles consist of a 2×2 grid and three tiles labeled 'A', 'B', and 'C'. The three tiles sit on top of the grid, leaving one grid cell empty. To make a move, Bessie or Elsie can slide a tile adjacent to the empty cell into the empty cell as shown below:

In order to determine if they are truly Best Friends For Life (BFFLs), Bessie and Elsie would like to know if there exists a sequence of moves that takes their puzzles to the same configuration (moves can be performed in both puzzles). Two puzzles are considered to be in the same configuration if each tile is on top of the same grid cell in both puzzles. Since the tiles are labeled with letters, rotations and reflections are not allowed.

Input

The first two lines of the input consist of a 2×2 grid describing the initial configuration of Bessie's puzzle. The next two lines contain a 2×2 grid describing the initial configuration of Elsie's puzzle. The positions of the tiles are labeled 'A', 'B', and 'C', while the empty cell is labeled 'X'. It's guaranteed that both puzzles contain exactly one tile with each letter and exactly one empty position.

Output

Output "YES"(without quotes) if the puzzles can reach the same configuration (and Bessie and Elsie are truly BFFLs). Otherwise, print "NO" (without quotes).

Examples

input

```
AB
XC
XB
AC
```

output

```
YES
```

input

```
AB
XC
AC
BX
```

output

```
NO
```

Note

The solution to the first sample is described by the image. All Bessie needs to do is slide her 'A' tile down.

In the second sample, the two puzzles can never be in the same configuration. Perhaps Bessie and Elsie are not meant to be friends after all...

B. Mischievous Mess Makers

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

It is a balmy spring afternoon, and Farmer John's n cows are ruminating about link-cut cacti in their stalls. The cows, labeled 1 through n, are arranged so that the i-th cow occupies the i-th stall from the left. However, Elsie, after realizing that she will forever live in the shadows beyond Bessie's limelight, has formed the Mischievous Mess Makers and is plotting to disrupt this beautiful pastoral rhythm. While Farmer John takes his k minute long nap, Elsie and the Mess Makers plan to repeatedly choose two distinct stalls and swap the cows occupying those stalls, making **no more than one** swap each minute.

Being the meticulous pranksters that they are, the Mischievous Mess Makers would like to know the maximum messiness attainable in the k minutes that they have. We denote as p_i the label of the cow in the i-th stall. The *messiness* of an arrangement of cows is defined as the number of pairs (i, j) such that $i < j$ and $p_i > p_j$.

Input

The first line of the input contains two integers n and k ($1 \le n, k \le 100\,000$) — the number of cows and the length of Farmer John's nap, respectively.

Output

Output a single integer, the maximum messiness that the Mischievous Mess Makers can achieve by performing no more than k swaps.

Examples

input
```
5 2
```
output
```
10
```

input
```
1 10
```
output
```
0
```

Note

In the first sample, the Mischievous Mess Makers can swap the cows in the stalls 1 and 5 during the first minute, then the cows in stalls 2 and 4 during the second minute. This reverses the arrangement of cows, giving us a total messiness of 10.

In the second sample, there is only one cow, so the maximum possible messiness is 0.

C. Enduring Exodus

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

In an attempt to escape the Mischievous Mess Makers' antics, Farmer John has abandoned his farm and is traveling to the other side of Bovinia. During the journey, he and his k cows have decided to stay at the luxurious Grand Moo-dapest Hotel. The hotel consists of n rooms located in a row, some of which are occupied.

Farmer John wants to book a set of $k + 1$ currently unoccupied rooms for him and his cows. He wants his cows to stay as safe as possible, so he wishes to minimize the maximum distance from his room to the room of his cow. The distance between rooms i and j is defined as $|j - i|$. Help Farmer John protect his cows by calculating this minimum possible distance.

Input

The first line of the input contains two integers n and k ($1 \le k < n \le 100\,000$) — the number of rooms in the hotel and the number of cows travelling with Farmer John.

The second line contains a string of length n describing the rooms. The i-th character of the string will be '0' if the i-th room is free, and '1' if the i-th room is occupied. It is guaranteed that at least $k + 1$ characters of this string are '0', so there exists at least one possible choice of $k + 1$ rooms for Farmer John and his cows to stay in.

Output

Print the minimum possible distance between Farmer John's room and his farthest cow.

Examples

input

```
7 2
0100100
```

output

```
2
```

input

```
5 1
01010
```

output

```
2
```

input

```
3 2
000
```

output

```
1
```

Note

In the first sample, Farmer John can book room 3 for himself, and rooms 1 and 4 for his cows. The distance to the farthest cow is 2. Note that it is impossible to make this distance 1, as there is no block of three consecutive unoccupied rooms.

In the second sample, Farmer John can book room 1 for himself and room 3 for his single cow. The distance between him and his cow is 2.

In the third sample, Farmer John books all three available rooms, taking the middle room for himself so that both cows are next to him. His distance from the farthest cow is 1.

D. Robot Rapping Results Report

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

While Farmer John rebuilds his farm in an unfamiliar portion of Bovinia, Bessie is out trying some alternative jobs. In her new gig as a reporter, Bessie needs to know about programming competition results as quickly as possible. When she covers the 2016 Robot Rap Battle Tournament, she notices that all of the robots operate under deterministic algorithms. In particular, robot i will beat robot j if and only if robot i has a higher skill level than robot j. And if robot i beats robot j and robot j beats robot k, then robot i will beat robot k. Since rapping is such a subtle art, two robots can never have the same skill level.

Given the results of the rap battles in the order in which they were played, determine the **minimum number** of first rap battles that needed to take place before Bessie could order all of the robots by skill level.

Input

The first line of the input consists of two integers, the number of robots n ($2 \le n \le 100\,000$) and the number of rap battles m ($1 \le m \le \min\left(100\,000, \frac{n(n-1)}{2}\right)$).

The next m lines describe the results of the rap battles in the order they took place. Each consists of two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), indicating that robot u_i beat robot v_i in the i-th rap battle. No two rap battles involve the same pair of robots.

It is guaranteed that at least one ordering of the robots satisfies all m relations.

Output

Print the minimum k such that the ordering of the robots by skill level is uniquely defined by the first k rap battles. If there exists more than one ordering that satisfies all m relations, output -1.

Examples

input

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

output

```
4
```

input

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

output

```
-1
```

Note

In the first sample, the robots from strongest to weakest must be $(4, 2, 1, 3)$, which Bessie can deduce after knowing the results of the first four rap battles.

In the second sample, both $(1, 3, 2)$ and $(3, 1, 2)$ are possible orderings of the robots from strongest to weakest after both rap battles.

E. Intellectual Inquiry

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

After getting kicked out of her reporting job for not knowing the alphabet, Bessie has decided to attend school at the Fillet and Eggs Eater Academy. She has been making good progress with her studies and now knows the first k English letters.

Each morning, Bessie travels to school along a sidewalk consisting of $m + n$ tiles. In order to help Bessie review, Mr. Moozing has labeled each of the first m sidewalk tiles with one of the first k lowercase English letters, spelling out a string t. Mr. Moozing, impressed by Bessie's extensive knowledge of farm animals, plans to let her finish labeling the last n tiles of the sidewalk by herself.

Consider the resulting string s ($|s| = m + n$) consisting of letters labeled on tiles in order from home to school. For any sequence of indices $p_1 < p_2 < \ldots < p_q$ we can define subsequence of the string s as string $s_{p_1}s_{p_2}\ldots s_{p_q}$. Two subsequences are considered to be distinct if they differ as strings. Bessie wants to label the remaining part of the sidewalk such that the number of **distinct subsequences** of tiles is maximum possible. However, since Bessie hasn't even finished learning the alphabet, she needs your help!

Note that empty subsequence also counts.

Input

The first line of the input contains two integers n and k ($0 \le n \le 1\,000\,000$, $1 \le k \le 26$).

The second line contains a string t ($|t| = m, 1 \le m \le 1\,000\,000$) consisting of only first k lowercase English letters.

Output

Determine the maximum number of distinct subsequences Bessie can form after labeling the last n sidewalk tiles each with one of the first k lowercase English letters. Since this number can be rather large, you should print it modulo $10^9 + 7$.

Please note, that you are not asked to maximize the remainder modulo $10^9 + 7$! The goal is to maximize the initial value and then print the remainder.

Examples

input

```
1 3
ac
```

output

```
8
```

input

```
0 2
aaba
```

output

```
10
```

Note

In the first sample, the optimal labeling gives 8 different subsequences: "" (the empty string), "a", "c", "b", "ac", "ab", "cb", and "acb".

In the second sample, the entire sidewalk is already labeled. The are 10 possible different subsequences: "" (the empty string), "a", "b", "aa", "ab", "ba", "aaa", "aab", "aba", and "aaba". Note that some strings, including "aa", can be obtained with multiple sequences of tiles, but are only counted once.

F. Cowslip Collections

time limit per test: 8 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

In an attempt to make peace with the Mischievious Mess Makers, Bessie and Farmer John are planning to plant some flower gardens to complement the lush, grassy fields of Bovinia. As any good horticulturist knows, each garden they plant must have the exact same arrangement of flowers. Initially, Farmer John has n different species of flowers he can plant, with a_i flowers of the i-th species.

On each of the next q days, Farmer John will receive a batch of flowers of a new species. On day j, he will receive c_j flowers of the same species, but of a different species from those Farmer John already has.

Farmer John, knowing the right balance between extravagance and minimalism, wants exactly k species of flowers to be used. Furthermore, to reduce waste, each flower of the k species Farmer John chooses must be planted in some garden. And each of the gardens must be identical; that is to say that each of the k chosen species should have an equal number of flowers in each garden. As Farmer John is a proponent of national equality, he would like to create the greatest number of gardens possible.

After receiving flowers on each of these q days, Farmer John would like to know the sum, over all possible choices of k species, of the maximum number of gardens he could create. Since this could be a large number, you should output your result modulo $10^9 + 7$.

Input

The first line of the input contains three integers n, k and q ($1 \leq k \leq n \leq 100\,000$, $1 \leq q \leq 100\,000$).

The i-th ($1 \leq i \leq n$) of the next n lines of the input contains an integer a_i ($1 \leq a_i \leq 1\,000\,000$), the number of flowers of species i Farmer John has initially.

The j-th ($1 \leq j \leq q$) of the next q lines of the input contains an integer c_j ($1 \leq c_j \leq 1\,000\,000$), the number of flowers of a new species Farmer John receives on day j.

Output

After each of the q days, output the sum of the maximum possible number of gardens, where the sum is taken over all possible choices of k species, modulo $10^9 + 7$.

Examples

input

```
3 3 2
4
6
9
8
6
```

output

```
5
16
```

input

```
4 1 2
6
5
4
3
2
1
```

output

```
20
21
```

Note

In the first sample case, after the first day Farmer John has $(4, 6, 9, 8)$ of each type of flower, and $k = 3$.

Choosing $(4, 6, 8)$ lets him make 2 gardens, each with $(2, 3, 4)$ of each flower, respectively. Choosing $(4, 6, 9)$, $(4, 9, 8)$ and $(6, 9, 8)$ each only let him make one garden, since there is no number of gardens that each species can be evenly split into. So the sum over all choices of $k = 3$ flowers is $2 + 1 + 1 + 1 = 5$.

After the second day, Farmer John has $(4, 6, 9, 8, 6)$ of each flower. The sum over all choices is $1 + 2 + 2 + 1 + 1 + 2 + 2 + 3 + 1 + 1 = 16$.

In the second sample case, $k = 1$. With x flowers Farmer John can make x gardens. So the answers to the queries are $6 + 5 + 4 + 3 + 2 = 20$ and $6 + 5 + 4 + 3 + 2 + 1 = 21$.

G. Armistice Area Apportionment

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

After a drawn-out mooclear arms race, Farmer John and the Mischievous Mess Makers have finally agreed to establish peace. They plan to divide the territory of Bovinia with a line passing through at least two of the n outposts scattered throughout the land. These outposts, remnants of the conflict, are located at the points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$.

In order to find the optimal dividing line, Farmer John and Elsie have plotted a map of Bovinia on the coordinate plane. Farmer John's farm and the Mischievous Mess Makers' base are located at the points $P = (a, 0)$ and $Q = (-a, 0)$, respectively. Because they seek a lasting peace, Farmer John and Elsie would like to minimize the maximum difference between the distances from any point on the line to P and Q.

Formally, define the **difference** of a line ℓ relative to two points P and Q as the smallest real number d so that for all points X on line ℓ, $|PX - QX| \le d$. (It is guaranteed that d exists and is unique.) They wish to find the line ℓ passing through two distinct outposts (x_i, y_i) and (x_j, y_j) such that the difference of ℓ relative to P and Q is minimized.

Input

The first line of the input contains two integers n and a ($2 \le n \le 100\,000$, $1 \le a \le 10\,000$) — the number of outposts and the coordinates of the farm and the base, respectively.

The following n lines describe the locations of the outposts as pairs of integers (x_i, y_i) ($|x_i|, |y_i| \le 10\,000$). These points are distinct from each other as well as from P and Q.

Output

Print a single real number—the difference of the optimal dividing line. Your 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 $\frac{|a-b|}{\max(1,b)} \le 10^{-6}$.

Examples

input

```
2 5
1 0
2 1
```

output

```
7.2111025509
```

input

```
3 6
0 1
2 5
0 -3
```

output

```
0.0000000000
```

Note

In the first sample case, the only possible line ℓ is $y = x - 1$. It can be shown that the point X which maximizes $|PX - QX|$ is $(13, 12)$, with $|PX - QX| = |\sqrt{(13-5)^2 + (12-0)^2} - \sqrt{(13-(-5))^2 + (12-0)^2}|$, which is $2\sqrt{13} \approx 7.2111025509$.

In the second sample case, if we pick the points $(0, 1)$ and $(0, -3)$, we get ℓ as $x = 0$. Because $PX = QX$ on this line, the minimum possible difference is 0.