Codeforces Round #320 (Div. 1) [Bayan Thanks-Round]

A. A Problem about Polyline

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

There is a polyline going through points $(0, 0) – (x, x) – (2x, 0) – (3x, x) – (4x, 0) – ... - (2kx, 0) – (2kx + x, x) – ....$

We know that the polyline passes through the point (a, b). Find minimum positive value x such that it is true or determine that there is no such x.

Input

Only one line containing two positive integers a and b $(1 \le a, b \le 10^9)$.

Output

Output the only line containing the answer. Your answer will be considered correct if its relative or absolute error doesn't exceed 10^{-9}. If there is no such x then output -1 as the answer.

Sample test(s)

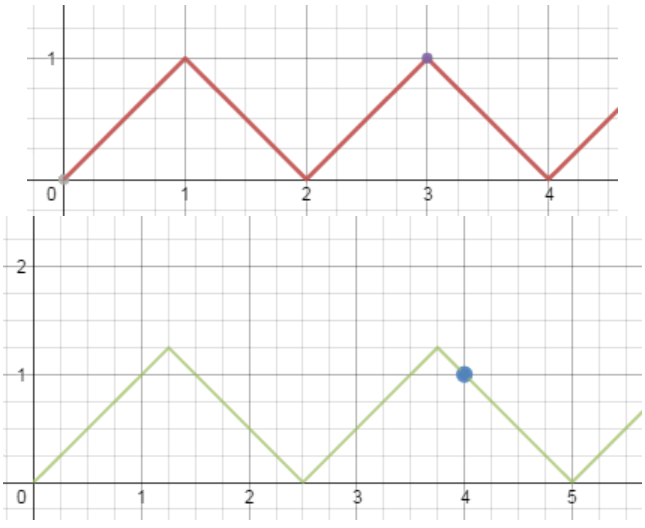
input
3 1
output
1.000000000000

input
1 3
output
-1

input
4 1
output
1.250000000000

Note

You can see following graphs for sample 1 and sample 3.



B. "Or" Game

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

You are given n numbers $a_1, a_2, ..., a_n$. You can perform at most k operations. For each operation you can multiply one of the numbers by x. We want to make $a_1 \mid a_2 \mid \ldots \mid a_n$ as large as possible, where $\mid$ denotes the bitwise OR.

Find the maximum possible value of $a_1 \mid a_2 \mid \ldots \mid a_n$ after performing at most k operations optimally.

Input

The first line contains three integers n, k and x ($1 \le n \le 200\,000$, $1 \le k \le 10$, $2 \le x \le 8$).

The second line contains n integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le 10^9$).

Output

Output the maximum value of a bitwise OR of sequence elements after performing operations.

Sample test(s)

input
3 1 2 1 1 1
output
3

input
4 2 3 1 2 4 8
output
79

Note

For the first sample, any possible choice of doing one operation will result the same three numbers $1, 1, 2$ so the result is $1 \mid 1 \mid 2 = 3$.

For the second sample if we multiply 8 by 3 two times we'll get 72. In this case the numbers will become $1, 2, 4, 72$ so the OR value will be 79 and is the largest possible result.

C. Weakness and Poorness

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

You are given a sequence of n integers $a_1, a_2, ..., a_n$.

Determine a real number x such that the *weakness* of the sequence $a_1 - x, a_2 - x, ..., a_n - x$ is as small as possible.

The *weakness* of a sequence is defined as the maximum value of the *poorness* over all segments (contiguous subsequences) of a sequence.

The *poorness* of a segment is defined as the absolute value of sum of the elements of segment.

Input

The first line contains one integer n ($1 \le n \le 200\,000$), the length of a sequence.

The second line contains n integers $a_1, a_2, ..., a_n$ ($|a_i| \le 10\,000$).

Output

Output a real number denoting the minimum possible *weakness* of $a_1 - x, a_2 - x, ..., a_n - x$. Your answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}.

Sample test(s)

input
3 1 2 3
output
1.000000000000000

input
4 1 2 3 4
output
2.000000000000000

input
10 1 10 2 9 3 8 4 7 5 6
output
4.500000000000000

Note

For the first case, the optimal value of x is 2 so the sequence becomes $-1, 0, 1$ and the max poorness occurs at the segment "-1" or segment "1". The poorness value (answer) equals to 1 in this case.

For the second sample the optimal value of x is 2.5 so the sequence becomes $-1.5, -0.5, 0.5, 1.5$ and the max poorness occurs on segment "-1.5 -0.5" or "0.5 1.5". The poorness value (answer) equals to 2 in this case.

D. LCS Again

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

You are given a string S of length n with each character being one of the first m lowercase English letters.

Calculate how many different strings T of length n composed from the first m lowercase English letters exist such that the length of LCS (longest common subsequence) between S and T is n - 1.

Recall that LCS of two strings S and T is the longest string C such that C both in S and T as a subsequence.

Input

The first line contains two numbers n and m denoting the length of string S and number of first English lowercase characters forming the character set for strings ($1 \le n \le 100\,000$, $2 \le m \le 26$).

The second line contains string S.

Output

Print the only line containing the answer.

Sample test(s)

input
3 3 aaa

output
6

input
3 3 aab

output
11

input
1 2 a

output
1

input
10 9 abacadefgh

output
789

Note

For the first sample, the 6 possible strings T are: `aab`, `aac`, `aba`, `aca`, `baa`, `caa`.

For the second sample, the 11 possible strings T are: `aaa`, `aac`, `aba`, `abb`, `abc`, `aca`, `acb`, `baa`, `bab`, `caa`, `cab`.

For the third sample, the only possible string T is `b`.

E. Walking!

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

There is a sand trail in front of Alice's home.

In daytime, people walk over it and leave a footprint on the trail for their every single step. Alice cannot distinguish the order of the footprints, but she can tell whether each footprint is made by left foot or right foot. Also she's certain that all people are walking by alternating left foot and right foot.

For example, suppose that one person walked through the trail and left some footprints. The footprints are `RRLRL` in order along the trail ('R' means right foot and 'L' means left foot). You might think the outcome of the footprints is strange. But in fact, some steps are resulting from walking backwards!

There are some possible order of steps that produce these footprints such as $1 \to 3 \to 2 \to 5 \to 4$ or $2 \to 3 \to 4 \to 5 \to 1$ (we suppose that the distance between two consecutive steps can be arbitrarily long). The number of backward steps from above two examples are 2 and 1 separately.

Alice is interested in these footprints. Whenever there is a person walking trough the trail, she takes a picture of all these footprints along the trail and erase all of them so that next person will leave a new set of footprints. We know that people walk by alternating right foot and left foot, but we don't know if the first step is made by left foot or right foot.

Alice wants to know the minimum possible number of backward steps made by a person. But it's a little hard. Please help Alice to calculate it. You also need to construct one possible history of these footprints.

Input

Only one line containing the string S ($1 \le |S| \le 100\,000$) containing all footprints in order along the trail from entrance to exit.

It is guaranteed that there is at least one possible footprint history.

Output

You should output 2 lines.

The first line should contain a number denoting the minimum number of backward steps.

The second line should contain a permutation of integers from 1 to $|S|$. This permutation should denote the order of footprints that may possible be used by person walked there.

If there are several possible answers, you may output any of them.

Sample test(s)

input
RRLRL
output
1 2 5 1 3 4

input
RLRLRLRLR
output
0 1 2 3 4 5 6 7 8 9

input
RRRRRLLLL
output
4 4 9 3 8 2 7 1 6 5

Note

For the first sample, one possible order is $2 \to 5 \to 1 \to 3 \to 4$, among them only the step $5 \to 1$ is backward step so the answer is 1.

For the second example one possible order is just to follow the order of input, thus there are no backward steps.

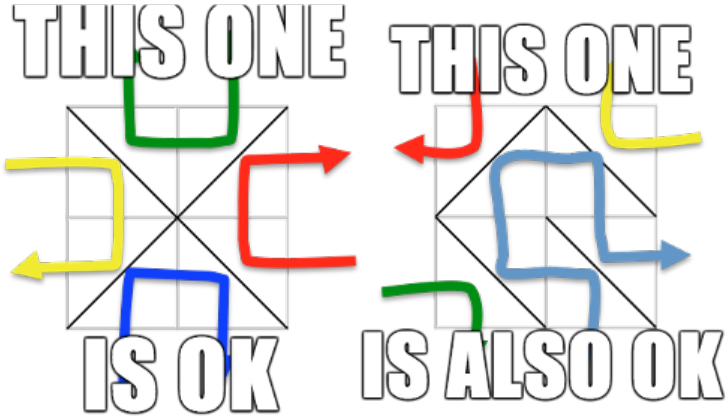
For the third sample, there will be 4 backward steps because every step from L to R will be a backward step.

F. Mirror Box

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

You are given a box full of mirrors. Box consists of grid of size $n \times m$. Each cell of the grid contains a mirror put in the shape of '\' or ' / ' (45 degree to the horizontal or vertical line). But mirrors in some cells have been destroyed. You want to put new mirrors into these grids so that the following two conditions are satisfied:

1. If you put a light ray horizontally/vertically into the middle of any unit segment that is side of some border cell, the light will go out from the neighboring unit segment to the segment you put the ray in.
2. each unit segment of the grid of the mirror box can be penetrated by at least one light ray horizontally/vertically put into the box according to the rules of the previous paragraph



After you tried putting some mirrors, you find out that there are many ways of doing so. How many possible ways are there? The answer might be large, so please find the result modulo prime number MOD.

Input

The first line contains three integers n, m, MOD ($1 \le n, m \le 100$, $3 \le MOD \le 10^9 + 7$, MOD is prime), m, n indicates the dimensions of a box and MOD is the number to module the answer.

The following n lines each contains a string of length m. Each string contains only ' / ', '\', '*', where '*' denotes that the mirror in that grid has been destroyed.

It is guaranteed that the number of '*' is no more than 200.

Output

Output the answer modulo MOD.

Sample test(s)

input
2 2 1000000007 */ /*
output
1

input
2 2 1000000007 ** \\
output
1

input
2 2 3 ** **
output
2

Note

The only way for sample 1 is shown on the left picture from the statement.

The only way for sample 2 is shown on the right picture from the statement.

For the third sample, there are 5 possibilities that are listed below:

1.

\\

\/

2.

//

\/

3.

/\

//

4.

/\

\\

5.

\/

/\

The answer is then module by 3 so the output should be 2.