Codeforces Round #148 (Div. 2)

A. Two Bags of Potatoes

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

Valera had two bags of potatoes, the first of these bags contains x ($x \geq 1$) potatoes, and the second — y ($y \geq 1$) potatoes. Valera — very scattered boy, so the first bag of potatoes (it contains x potatoes) Valera lost. Valera remembers that the total amount of potatoes ($x + y$) in the two bags, firstly, was not gerater than n, and, secondly, was divisible by k.

Help Valera to determine how many potatoes could be in the first bag. Print all such possible numbers in ascending order.

Input

The first line of input contains three integers y, k, n ($1 \leq y, k, n \leq 10^9$; $\frac{n}{k} \leq 10^5$).

Output

Print the list of whitespace-separated integers — all possible values of x in ascending order. You should print each possible value of x exactly once.

If there are no such values of x print a single integer -1.

Examples

input
10 1 10
output
-1

input
10 6 40
output
2 8 14 20 26

B. Easy Tape Programming

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

There is a programming language in which every program is a non-empty sequence of "<" and ">" signs and digits. Let's explain how the interpreter of this programming language works. A program is interpreted using movement of instruction pointer (IP) which consists of two parts.

- Current character pointer (CP);
- Direction pointer (DP) which can point left or right;

Initially CP points to the leftmost character of the sequence and DP points to the right.

We repeat the following steps until the first moment that CP points to somewhere outside the sequence.

- If CP is pointing to a digit the interpreter prints that digit then CP moves one step according to the direction of DP. After that the value of the printed digit in the sequence decreases by one. If the printed digit was 0 then it cannot be decreased therefore it's erased from the sequence and the length of the sequence decreases by one.
- If CP is pointing to "<" or ">" then the direction of DP changes to "left" or "right" correspondingly. Then CP moves one step according to DP. If the new character that CP is pointing to is "<" or ">" then the previous character will be erased from the sequence.

If at any moment the CP goes outside of the sequence the execution is terminated.

It's obvious the every program in this language terminates after some steps.

We have a sequence $s_1, s_2, \ldots, s_n$ of "<", ">" and digits. You should answer q queries. Each query gives you l and r and asks how many of each digit will be printed if we run the sequence $s_l, s_{l+1}, \ldots, s_r$ as an independent program in this language.

Input

The first line of input contains two integers n and q ($1 \le n, q \le 100$) — represents the length of the sequence s and the number of queries.

The second line contains s, a sequence of "<", ">" and digits (`0..9`) written from left to right. Note, that the characters of s are not separated with spaces.

The next q lines each contains two integers l_i and r_i ($1 \le l_i \le r_i \le n$) — the i-th query.

Output

For each query print 10 space separated integers: $x_0, x_1, \ldots, x_9$ where x_i equals the number of times the interpreter prints i while running the corresponding program. Print answers to the queries in the order they are given in input.

Examples

input
7 4 1>3>22< 1 3 4 7 7 7 1 7
output
0 1 0 1 0 0 0 0 0 0 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 3 2 1 0 0 0 0 0 0

C. Not Wool Sequences

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

A sequence of non-negative integers $a_1, a_2, ..., a_n$ of length n is called a *wool sequence* if and only if there exists two integers l and r $(1 \le l \le r \le n)$ such that $a_l \oplus a_{l+1} \oplus \cdots \oplus a_r = 0$. In other words each wool sequence contains a subsequence of consecutive elements with xor equal to 0.

The expression $x \oplus y$ means applying the operation of a bitwise xor to numbers x and y. The given operation exists in all modern programming languages, for example, in languages *C++* and *Java* it is marked as "^", in *Pascal* — as "xor".

In this problem you are asked to compute the number of sequences made of n integers from 0 to $2^m - 1$ that are not a wool sequence. You should print this number modulo 1000000009 $(10^9 + 9)$.

Input

The only line of input contains two space-separated integers n and m $(1 \le n, m \le 10^5)$.

Output

Print the required number of sequences modulo 1000000009 $(10^9 + 9)$ on the only line of output.

Examples

input
3 2
output
6

Note

Sequences of length 3 made of integers 0, 1, 2 and 3 that are not a wool sequence are `(1, 3, 1)`, `(1, 2, 1)`, `(2, 1, 2)`, `(2, 3, 2)`, `(3, 1, 3)` and `(3, 2, 3)`.

D. Boring Partition

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

This problem is the most boring one you've ever seen.

Given a sequence of integers $a_1, a_2, \ldots, a_n$ and a non-negative integer h, our goal is to partition the sequence into two subsequences (not necessarily consist of continuous elements). Each element of the original sequence should be contained in exactly one of the result subsequences. Note, that one of the result subsequences can be empty.

Let's define function $f(a_i, a_j)$ on pairs of distinct elements (that is $i \neq j$) in the original sequence. If a_i and a_j are in the same subsequence in the current partition then $f(a_i, a_j) = a_i + a_j$ otherwise $f(a_i, a_j) = a_i + a_j + h$.

Consider all possible values of the function f for some partition. We'll call the *goodness* of this partiotion the difference between the maximum value of function f and the minimum value of function f.

Your task is to find a partition of the given sequence a that have the minimal possible goodness among all possible partitions.

Input

The first line of input contains integers n and h ($2 \leq n \leq 10^5, 0 \leq h \leq 10^8$). In the second line there is a list of n space-separated integers representing $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 10^8$).

Output

The first line of output should contain the required minimum goodness.

The second line describes the optimal partition. You should print n whitespace-separated integers in the second line. The i-th integer is 1 if a_i is in the first subsequence otherwise it should be 2.

If there are several possible correct answers you are allowed to print any of them.

Examples

input
3 2 1 2 3
output
1 1 2 2

input
5 10 0 1 0 2 1
output
3 2 2 2 2 2

Note

In the first sample the values of f are as follows: $f(1, 2) = 1 + 2 + 2 = 5$, $f(1, 3) = 1 + 3 + 2 = 6$ and $f(2, 3) = 2 + 3 = 5$. So the difference between maximum and minimum values of f is 1.

In the second sample the value of h is large, so it's better for one of the sub-sequences to be empty.

E. World Eater Brothers

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

You must have heard of the two brothers dreaming of ruling the world. With all their previous plans failed, this time they decided to cooperate with each other in order to rule the world.

As you know there are n countries in the world. These countries are connected by $n - 1$ directed roads. If you don't consider direction of the roads there is a unique path between every pair of countries in the world, passing through each road at most once.

Each of the brothers wants to establish his reign in some country, then it's possible for him to control the countries that can be reached from his country using directed roads.

The brothers can rule the world if there exists at most two countries for brothers to choose (and establish their reign in these countries) so that any other country is under control of at least one of them. In order to make this possible they want to change the direction of minimum number of roads. Your task is to calculate this minimum number of roads.

Input

The first line of input contains an integer n ($1 \le n \le 3000$). Each of the next $n - 1$ lines contains two space-separated integers a_i and b_i ($1 \le a_i, b_i \le n$; $a_i \ne b_i$) saying there is a road from country a_i to country b_i.

Consider that countries are numbered from 1 to n. It's guaranteed that if you don't consider direction of the roads there is a unique path between every pair of countries in the world, passing through each road at most once.

Output

In the only line of output print the minimum number of roads that their direction should be changed so that the brothers will be able to rule the world.

Examples

input
4 2 1 3 1 4 1
output
1

input
5 2 1 2 3 4 3 4 5
output
0

Codeforces (c) Copyright 2010-2016 Mike Mirzayanov
The only programming contests Web 2.0 platform