AIM Tech Round 3 (Div. 2)

A. Juicer

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

Kolya is going to make fresh orange juice. He has n oranges of sizes $a_1, a_2, ..., a_n$. Kolya will put them in the juicer in the fixed order, starting with orange of size a_1, then orange of size a_2 and so on. To be put in the juicer the orange must have size not exceeding b, so if Kolya sees an orange that is strictly greater he throws it away and continues with the next one.

The juicer has a special section to collect waste. It overflows if Kolya squeezes oranges of the total size strictly greater than d. When it happens Kolya empties the waste section (even if there are no more oranges) and continues to squeeze the juice. How many times will he have to empty the waste section?

Input

The first line of the input contains three integers n, b and d ($1 \le n \le 100\,000$, $1 \le b \le d \le 1\,000\,000$) — the number of oranges, the maximum size of the orange that fits in the juicer and the value d, which determines the condition when the waste section should be emptied.

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 1\,000\,000$) — sizes of the oranges listed in the order Kolya is going to try to put them in the juicer.

Output

Print one integer — the number of times Kolya will have to empty the waste section.

Examples

input
2 7 10 5 6
output
1

input
1 5 10 7
output
0

input
3 10 10 5 7 7
output
1

input
1 1 1 1
output
0

Note

In the first sample, Kolya will squeeze the juice from two oranges and empty the waste section afterwards.

In the second sample, the orange won't fit in the juicer so Kolya will have no juice at all.

B. Checkpoints

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

Vasya takes part in the orienteering competition. There are n checkpoints located along the line at coordinates $x_1, x_2, ..., x_n$. Vasya starts at the point with coordinate a. His goal is to visit at least $n - 1$ checkpoint in order to finish the competition. Participant are allowed to visit checkpoints in arbitrary order.

Vasya wants to pick such checkpoints and the order of visiting them that the total distance travelled is minimized. He asks you to calculate this minimum possible value.

Input

The first line of the input contains two integers n and a ($1 \le n \le 100\,000, -1\,000\,000 \le a \le 1\,000\,000$) — the number of checkpoints and Vasya's starting position respectively.

The second line contains n integers $x_1, x_2, ..., x_n$ ($-1\,000\,000 \le x_i \le 1\,000\,000$) — coordinates of the checkpoints.

Output

Print one integer — the minimum distance Vasya has to travel in order to visit at least $n - 1$ checkpoint.

Examples

input
3 10 1 7 12
output
7

input
2 0 11 -10
output
10

input
5 0 0 0 1000 0 0
output
0

Note

In the first sample Vasya has to visit at least two checkpoints. The optimal way to achieve this is the walk to the third checkpoints (distance is $12 - 10 = 2$) and then proceed to the second one (distance is $12 - 7 = 5$). The total distance is equal to $2 + 5 = 7$.

In the second sample it's enough to visit only one checkpoint so Vasya should just walk to the point -10.

C. Letters Cyclic Shift

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

You are given a non-empty string s consisting of lowercase English letters. You have to pick **exactly one non-empty substring** of s and shift all its letters 'z' 'y' 'x' 'b' 'a' 'z'. In other words, each character is replaced with the previous character of English alphabet and 'a' is replaced with 'z'.

What is the lexicographically minimum string that can be obtained from s by performing this shift exactly once?

Input

The only line of the input contains the string s ($1 \le |s| \le 100\,000$) consisting of lowercase English letters.

Output

Print the lexicographically minimum string that can be obtained from s by shifting letters of exactly one non-empty substring.

Examples

input
codeforces
output
bncdenqbdr

input
abacaba
output
aaacaba

Note

String s is lexicographically smaller than some other string t of the same length if there exists some $1 \le i \le |s|$, such that $s_1 = t_1, s_2 = t_2, ..., s_{i-1} = t_{i-1}$, and $s_i < t_i$.

D. Recover the String

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

For each string s consisting of characters '0' and '1' one can define four integers a_{00}, a_{01}, a_{10} and a_{11}, where a_{xy} is the number of **subsequences** of length 2 of the string s equal to the sequence $\{x, y\}$.

In these problem you are given four integers a_{00}, a_{01}, a_{10}, a_{11} and have to find any non-empty string s that matches them, or determine that there is no such string. One can prove that if at least one answer exists, there exists an answer of length no more than $1\,000\,000$.

Input

The only line of the input contains four non-negative integers a_{00}, a_{01}, a_{10} and a_{11}. Each of them doesn't exceed 10^9.

Output

If there exists a non-empty string that matches four integers from the input, print it in the only line of the output. Otherwise, print "Impossible". The length of your answer must not exceed $1\,000\,000$.

Examples

input
1 2 3 4
output
Impossible

input
1 2 2 1
output
0110

E. Centroids

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Tree is a connected acyclic graph. Suppose you are given a tree consisting of n vertices. The vertex of this tree is called *centroid* if the size of each connected component that appears if this vertex is removed from the tree doesn't exceed .

You are given a tree of size n and can perform no more than one edge replacement. *Edge replacement* is the operation of removing one edge from the tree (without deleting incident vertices) and inserting one new edge (without adding new vertices) in such a way that the graph remains a tree. For each vertex you have to determine if it's possible to make it centroid by performing no more than one edge replacement.

Input

The first line of the input contains an integer n ($2 \le n \le 400\,000$) — the number of vertices in the tree. Each of the next $n - 1$ lines contains a pair of vertex indices u_i and v_i ($1 \le u_i, v_i \le n$) — endpoints of the corresponding edge.

Output

Print n integers. The i-th of them should be equal to 1 if the i-th vertex can be made centroid by replacing no more than one edge, and should be equal to 0 otherwise.

Examples

input
3 1 2 2 3
output
1 1 1

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

Note

In the first sample each vertex can be made a centroid. For example, in order to turn vertex 1 to centroid one have to replace the edge $(2, 3)$ with the edge $(1, 3)$.