Codeforces Round #397 by Kaspersky Lab and Barcelona Bootcamp (Div. 1 + Div. 2 combined)

A. Neverending competitions

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

There are literally dozens of snooker competitions held each year, and team Jinotega tries to attend them all (for some reason they prefer name "snookah")! When a competition takes place somewhere far from their hometown, Ivan, Artsem and Konstantin take a flight to the contest and back.

Jinotega's best friends, team Base have found a list of their itinerary receipts with information about departure and arrival airports. Now they wonder, where is Jinotega now: at home or at some competition far away? They know that:

- this list contains all Jinotega's flights in this year (**in arbitrary order**),
- Jinotega has only flown from his hometown to a snooker contest and back,
- after each competition Jinotega flies back home (though they may attend a competition in one place several times),
- and finally, at the beginning of the year Jinotega was at home.

Please help them to determine Jinotega's location!

Input

In the first line of input there is a single integer n: the number of Jinotega's flights ($1 \le n \le 100$). In the second line there is a string of 3 capital Latin letters: the name of Jinotega's home airport. In the next n lines there is flight information, one flight per line, in form "`XXX->YYY`", where "`XXX`" is the name of departure airport "`YYY`" is the name of arrival airport. Exactly one of these airports is Jinotega's home airport.

It is guaranteed that flights information is consistent with the knowledge of Jinotega's friends, which is described in the main part of the statement.

Output

If Jinotega is now at home, print "`home`" (without quotes), otherwise print "`contest`".

Examples

input
4 SVO SVO->CDG LHR->SVO SVO->LHR CDG->SVO
output
home

input
3 SVO SVO->HKT HKT->SVO SVO->RAP
output
contest

Note

In the first sample Jinotega might first fly from SVO to CDG and back, and then from SVO to LHR and back, so now they should be at home. In the second sample Jinotega must now be at RAP because a flight from RAP back to SVO is not on the list.

B. Code obfuscation

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Kostya likes Codeforces contests very much. However, he is very disappointed that his solutions are frequently hacked. That's why he decided to obfuscate (intentionally make less readable) his code before upcoming contest.

To obfuscate the code, Kostya first looks at the first variable name used in his program and replaces all its occurrences with a single symbol a, then he looks at the second variable name that has not been replaced yet, and replaces all its occurrences with b, and so on. Kostya is well-mannered, so he doesn't use any one-letter names before obfuscation. Moreover, there are at most 26 unique identifiers in his programs.

You are given a list of identifiers of some program with removed spaces and line breaks. Check if this program can be a result of Kostya's obfuscation.

Input

In the only line of input there is a string S of lowercase English letters ($1 \le |S| \le 500$) — the identifiers of a program with removed whitespace characters.

Output

If this program can be a result of Kostya's obfuscation, print "`YES`" (without quotes), otherwise print "`NO`".

Examples

input
abacaba
output
YES

input
jinotega
output
NO

Note

In the first sample case, one possible list of identifiers would be "`number string number character number string number`". Here how Kostya would obfuscate the program:

- replace all occurences of `number` with `a`, the result would be "`a string a character a string a`",
- replace all occurences of `string` with `b`, the result would be "`a b a character a b a`",
- replace all occurences of `character` with `c`, the result would be "`a b a c a b a`",
- all identifiers have been replaced, thus the obfuscation is finished.

C. Table Tennis Game 2

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Misha and Vanya have played several table tennis sets. Each set consists of several serves, each serve is won by one of the players, he receives one point and the loser receives nothing. Once one of the players scores exactly k points, the score is reset and a new set begins.

Across all the sets Misha scored a points in total, and Vanya scored b points. Given this information, determine the maximum number of sets they could have played, or that the situation is impossible.

Note that the game consisted of several complete sets.

Input

The first line contains three space-separated integers k, a and b ($1 \le k \le 10^9$, $0 \le a, b \le 10^9$, $a + b > 0$).

Output

If the situation is impossible, print a single number -1. Otherwise, print the maximum possible number of sets.

Examples

input
11 11 5
output
1

input
11 2 3
output
-1

Note

Note that the rules of the game in this problem differ from the real table tennis game, for example, the rule of "balance" (the winning player has to be at least two points ahead to win a set) has no power within the present problem.

D. Artsem and Saunders

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Artsem has a friend Saunders from University of Chicago. Saunders presented him with the following problem.

Let $[n]$ denote the set $\{1, ..., n\}$. We will also write $f: [x] \to [y]$ when a function f is defined in integer points $1, ..., x$, and all its values are integers from 1 to y.

Now then, you are given a function $f: [n] \to [n]$. Your task is to find a positive integer m, and two functions $g: [n] \to [m]$, $h: [m] \to [n]$, such that $g(h(x)) = x$ for all , and $h(g(x)) = f(x)$ for all , or determine that finding these is impossible.

Input

The first line contains an integer n ($1 \le n \le 10^5$).

The second line contains n space-separated integers — values $f(1), ..., f(n)$ ($1 \le f(i) \le n$).

Output

If there is no answer, print one integer `-1`.

Otherwise, on the first line print the number m ($1 \le m \le 10^6$). On the second line print n numbers $g(1), ..., g(n)$. On the third line print m numbers $h(1), ..., h(m)$.

If there are several correct answers, you may output any of them. It is guaranteed that if a valid answer exists, then there is an answer satisfying the above restrictions.

Examples

input
3 1 2 3
output
3 1 2 3 1 2 3

input
3 2 2 2
output
1 1 1 1 2

input
2 2 1
output
-1

E. Tree Folding

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Vanya wants to minimize a tree. He can perform the following operation multiple times: choose a vertex v, and two disjoint (except for v) paths of equal length $a_0 = v, a_1, ..., a_k$, and $b_0 = v, b_1, ..., b_k$. Additionally, vertices $a_1, ..., a_k, b_1, ..., b_k$ must not have any neighbours in the tree other than adjacent vertices of corresponding paths. After that, one of the paths may be merged into the other, that is, the vertices $b_1, ..., b_k$ can be effectively erased:

Help Vanya determine if it possible to make the tree into a path via a sequence of described operations, and if the answer is positive, also determine the shortest length of such path.

Input

The first line of input contains the number of vertices n ($2 \le n \le 2 \cdot 10^5$).

Next $n - 1$ lines describe edges of the tree. Each of these lines contains two space-separated integers u and v ($1 \le u, v \le n, u \ne v$) — indices of endpoints of the corresponding edge. It is guaranteed that the given graph is a tree.

Output

If it is impossible to obtain a path, print `-1`. Otherwise, print the minimum number of edges in a possible path.

Examples

input
6 1 2 2 3 2 4 4 5 1 6
output
3

input
7 1 2 1 3 3 4 1 5 5 6 6 7
output
-1

Note

In the first sample case, a path of three edges is obtained after merging paths $2 - 1 - 6$ and $2 - 4 - 5$.

It is impossible to perform any operation in the second sample case. For example, it is impossible to merge paths $1 - 3 - 4$ and $1 - 5 - 6$, since vertex 6 additionally has a neighbour 7 that is not present in the corresponding path.

F. Souvenirs

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Artsem is on vacation and wants to buy souvenirs for his two teammates. There are n souvenir shops along the street. In i-th shop Artsem can buy one souvenir for a_i dollars, and he cannot buy more than one souvenir in one shop. He doesn't want to introduce envy in his team, so he wants to buy two souvenirs with least possible difference in price.

Artsem has visited the shopping street m times. For some strange reason on the i-th day only shops with numbers from l_i to r_i were operating (weird? yes it is, but have you ever tried to come up with a reasonable legend for a range query problem?). For each visit, Artsem wants to know the minimum possible difference in prices of two different souvenirs he can buy in the opened shops.

In other words, for each Artsem's visit you should find the minimum possible value of $|a_s - a_t|$ where $l_i \le s, t \le r_i, s \ne t$.

Input

The first line contains an integer n ($2 \le n \le 10^5$).

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

The third line contains the number of queries m ($1 \le m \le 3 \cdot 10^5$).

Next m lines describe the queries. i-th of these lines contains two space-separated integers l_i and r_i denoting the range of shops working on i-th day ($1 \le l_i < r_i \le n$).

Output

Print the answer to each query in a separate line.

Example

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

G. Math, math everywhere

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

If you have gone that far, you'll probably skip unnecessary legends anyway...

You are given a binary string and an integer . Find the number of integers k, $0 \le k < N$, such that for all $i = 0, 1, ..., m - 1$

Print the answer modulo $10^9 + 7$.

Input

In the first line of input there is a string s consisting of 0's and 1's ($1 \le |s| \le 40$).

In the next line of input there is an integer n ($1 \le n \le 5 \cdot 10^5$).

Each of the next n lines contains two space-separated integers p_i, α_i ($1 \le p_i, \alpha_i \le 10^9$, p_i is prime). All p_i are distinct.

Output

A single integer — the answer to the problem.

Examples

input
1 2 2 1 3 1
output
2

input
01 2 3 2 5 1
output
15

input
1011 1 3 1000000000
output
411979884