

Educational Codeforces Round 19

A. k-Factorization

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Given a positive integer n , find k integers (not necessary distinct) such that all these integers are strictly greater than 1, and their product is equal to n .

Input

The first line contains two integers n and k ($2 \leq n \leq 100000$, $1 \leq k \leq 20$).

Output

If it's impossible to find the representation of n as a product of k numbers, print -1 .

Otherwise, print k integers in any order. Their product must be equal to n . If there are multiple answers, print any of them.

Examples

input
100000 2
output
2 50000

input
100000 20
output
-1

input
1024 5
output
2 64 2 2 2

B. Odd sum

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given sequence a_1, a_2, \dots, a_n of integer numbers of length n . Your task is to find such subsequence that its sum is odd and maximum among all such subsequences. It's guaranteed that given sequence contains subsequence with odd sum.

Subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements.

You should write a program which finds sum of the best subsequence.

Input

The first line contains integer number n ($1 \leq n \leq 10^5$).

The second line contains n integer numbers a_1, a_2, \dots, a_n ($-10^4 \leq a_i \leq 10^4$). The sequence contains at least one subsequence with odd sum.

Output

Print sum of resulting subsequence.

Examples

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

Note

In the first example sum of the second and the fourth elements is 3.

C. Minimal string

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Petya recieved a gift of a string s with length up to 10^5 characters for his birthday. He took two more empty strings t and u and decided to play a game. This game has two possible moves:

- Extract the **first** character of s and append t with this character.
- Extract the **last** character of t and append u with this character.

Petya wants to get strings s and t empty and string u lexicographically minimal.

You should write a program that will help Petya win the game.

Input

First line contains non-empty string s ($1 \leq |s| \leq 10^5$), consisting of lowercase English letters.

Output

Print resulting string u .

Examples

input
c ab
output
a bc

input
a cdb
output
a bdc

D. Broken BST

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Let T be arbitrary binary tree — tree, every vertex of which has no more than two children. Given tree is rooted, so there exists only one vertex which doesn't have a parent — it's the root of a tree. Every vertex has an integer number written on it. Following algorithm is run on every value from the tree T :

1. Set pointer to the root of a tree.
2. Return success if the value in the current vertex is equal to the number you are looking for
3. Go to the left child of the vertex if the value in the current vertex is greater than the number you are looking for
4. Go to the right child of the vertex if the value in the current vertex is less than the number you are looking for
5. Return fail if you try to go to the vertex that doesn't exist

Here is the pseudo-code of the described algorithm:

```
bool find(TreeNode t, int x) {  
    if (t == null)  
        return false;  
    if (t.value == x)  
        return true;  
    if (x < t.value)  
        return find(t.left, x);  
    else  
        return find(t.right, x);  
}  
find(root, x);
```

The described algorithm works correctly if the tree is binary search tree (i.e. for each node the values of left subtree are less than the value in the node, the values of right subtree are greater than the value in the node). But it can return invalid result if tree is not a binary search tree.

Since the given tree is not necessarily a binary search tree, not all numbers can be found this way. Your task is to calculate, how many times the search will fail being running on every value from the tree.

If the tree has multiple vertices with the same values on them then you should run algorithm on every one of them separately.

Input

First line contains integer number n ($1 \leq n \leq 10^5$) — number of vertices in the tree.

Each of the next n lines contains 3 numbers v, l, r ($0 \leq v \leq 10^9$) — value on current vertex, index of the left child of the vertex and index of the right child of the vertex, respectively. If some child doesn't exist then number - 1 is set instead. Note that different vertices of the tree may contain the same values.

Output

Print number of times when search algorithm will fail.

Examples

input
3 15 -1 -1 10 1 3 5 -1 -1
output
2

input
8 6 2 3 3 4 5 12 6 7 1 -1 8 4 -1 -1 5 -1 -1 14 -1 -1 2 -1 -1
output
1

Note

In the example the root of the tree is vertex 2. Search of numbers 5 and 15 will return fail because on the first step algorithm will choose the subtree which doesn't contain numbers you are looking for.

E. Array Queries

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

a is an array of n positive integers, all of which are not greater than n .

You have to process q queries to this array. Each query is represented by two numbers p and k . Several operations are performed in each query; each operation changes p to $p + a_p + k$. These operations are applied until p becomes greater than n . The answer to the query is the number of performed operations.

Input

The first line contains one integer n ($1 \leq n \leq 100000$).

The second line contains n integers — elements of a ($1 \leq a_i \leq n$ for each i from 1 to n).

The third line contains one integer q ($1 \leq q \leq 100000$).

Then q lines follow. Each line contains the values of p and k for corresponding query ($1 \leq p, k \leq n$).

Output

Print q integers, i th integer must be equal to the answer to i th query.

Example

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

Note

Consider first example:

In first query after first operation $p = 3$, after second operation $p = 5$.

In next two queries p is greater than n after the first operation.

F. Mice and Holes

time limit per test: 1.5 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

One day Masha came home and noticed n mice in the corridor of her flat. Of course, she shouted loudly, so scared mice started to run to the holes in the corridor.

The corridor can be represented as a numeric axis with n mice and m holes on it. i th mouse is at the coordinate x_i , and j th hole — at coordinate p_j . j th hole has enough room for c_j mice, so not more than c_j mice can enter this hole.

What is the minimum sum of distances that mice have to go through so that they all can hide in the holes? If i th mouse goes to the hole j , then its distance is $|x_i - p_j|$.

Print the minimum sum of distances.

Input

The first line contains two integer numbers n, m ($1 \leq n, m \leq 5000$) — the number of mice and the number of holes, respectively.

The second line contains n integers x_1, x_2, \dots, x_n ($-10^9 \leq x_i \leq 10^9$), where x_i is the coordinate of i th mouse.

Next m lines contain pairs of integer numbers p_j, c_j ($-10^9 \leq p_j \leq 10^9, 1 \leq c_j \leq 5000$), where p_j is the coordinate of j th hole, and c_j is the maximum number of mice that can hide in the hole j .

Output

Print one integer number — the minimum sum of distances. If there is no solution, print -1 instead.

Examples

input
4 5 6 2 8 9 3 6 2 1 3 6 4 7 4 7
output
11

input
7 2 10 20 30 40 50 45 35 -1000000000 10 1000000000 1
output
7000000130