Codeforces Round #202 (Div. 2)

A. Cinema Line

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

The new "Die Hard" movie has just been released! There are n people at the cinema box office standing in a huge line. Each of them has a single 100, 50 or 25 ruble bill. A "Die Hard" ticket costs 25 rubles. Can the booking clerk sell a ticket to each person and give the change if he initially has no money and sells the tickets strictly in the order people follow in the line?

Input

The first line contains integer n $(1 \le n \le 10^5)$ — the number of people in the line. The next line contains n integers, each of them equals 25, 50 or 100 — the values of the bills the people have. The numbers are given in the order from the beginning of the line (at the box office) to the end of the line.

Output

Print "YES" (without the quotes) if the booking clerk can sell a ticket to each person and give the change. Otherwise print " NO".

Sample test(s)

input
4 25 25 50 50
output
YES

input
2 25 100
output
NO

input
4 50 50 25 25
output
NO

B. Color the Fence

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

Igor has fallen in love with Tanya. Now Igor wants to show his feelings and write a number on the fence opposite to Tanya's house. Igor thinks that the larger the number is, the more chance to win Tanya's heart he has.

Unfortunately, Igor could only get v liters of paint. He did the math and concluded that digit d requires a_d liters of paint. Besides, Igor heard that Tanya doesn't like zeroes. That's why Igor won't use them in his number.

Help Igor find the maximum number he can write on the fence.

Input

The first line contains a positive integer v $(0 \le v \le 10^6)$. The second line contains nine positive integers $a_1, a_2, ..., a_9$ $(1 \le a_i \le 10^5)$.

Output

Print the maximum number Igor can write on the fence. If he has too little paint for any digit (so, he cannot write anything), print -1.

Sample test(s)

input
5 5 4 3 2 1 2 3 4 5
output
55555

input
2 9 11 1 12 5 8 9 10 6
output
33

input
0 1 1 1 1 1 1 1 1 1
output
-1

C. Mafia

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

One day n friends gathered together to play "Mafia". During each round of the game some player must be the supervisor and other $n - 1$ people take part in the game. For each person we know in how many rounds he wants to be a player, not the supervisor: the i-th person wants to play a_i rounds. What is the minimum number of rounds of the "Mafia" game they need to play to let each person play at least as many rounds as they want?

Input

The first line contains integer n $(3 \le n \le 10^5)$. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ — the i-th number in the list is the number of rounds the i-th person wants to play.

Output

In a single line print a single integer — the minimum number of game rounds the friends need to let the i-th person play at least a_i rounds.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Sample test(s)

input
3 3 2 2

output
4

input
4 2 2 2 2

output
3

Note

You don't need to know the rules of "Mafia" to solve this problem. If you're curious, it's a game Russia got from the Soviet times: `http://en.wikipedia.org/wiki/Mafia_(party_game)`.

D. Apple Tree

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

You are given a rooted tree with n vertices. In each leaf vertex there's a single integer — the number of apples in this vertex.

The *weight* of a subtree is the sum of all numbers in this subtree leaves. For instance, the weight of a subtree that corresponds to some leaf is the number written in the leaf.

A tree is *balanced* if for every vertex v of the tree all its subtrees, corresponding to the children of vertex v, are of equal weight.

Count the minimum number of apples that you need to remove from the tree (specifically, from some of its leaves) in order to make the tree balanced. Notice that you can always achieve the goal by just removing all apples.

Input

The first line contains integer n $(2 \le n \le 10^5)$, showing the number of vertices in the tree. The next line contains n integers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^8)$, a_i is the number of apples in the vertex number i. The number of apples in non-leaf vertices is guaranteed to be zero.

Then follow n - 1 lines, describing the tree edges. Each line contains a pair of integers x_i, y_i $(1 \le x_i, y_i \le n, x_i \ne y_i)$ — the vertices connected by an edge.

The vertices are indexed from 1 to n. Vertex 1 is the root.

Output

Print a single integer — the minimum number of apples to remove in order to make the tree balanced.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the sin, cout streams `cin`, `cout` or the `%I64d` specifier.

Sample test(s)

input
6 0 0 12 13 5 6 1 2 1 3 1 4 2 5 2 6

output
6

E. Subset Sums

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array $a_1, a_2, ..., a_n$ and m sets $S_1, S_2, ..., S_m$ of indices of elements of this array. Let's denote $S_k = \{S_{k,i}\}$ $(1 \le i \le |S_k|)$. In other words, $S_{k,i}$ is some element from set S_k.

In this problem you have to answer q queries of the two types:

1. Find the sum of elements with indices from set S_k: $\sum_{i=1}^{|S_k|} a_{S_{k,i}}$. The query format is "`? k`".
2. Add number x to all elements at indices from set S_k: $a_{S_{k,i}}$ is replaced by $a_{S_{k,i}} + x$ for all i $(1 \le i \le |S_k|)$. The query format is "`+ k x`".

After each first type query print the required sum.

Input

The first line contains integers n, m, q $(1 \le n, m, q \le 10^5)$. The second line contains n integers $a_1, a_2, ..., a_n$ $(|a_i| \le 10^8)$ — elements of array a.

Each of the following m lines describes one set of indices. The k-th line first contains a positive integer, representing the number of elements in set $(|S_k|)$, then follow $|S_k|$ distinct integers $S_{k,1}, S_{k,2}, ..., S_{k,|S_k|}$ $(1 \le S_{k,i} \le n)$ — elements of set S_k.

The next q lines contain queries. Each query looks like either "`? k`" or "`+ k x`" and sits on a single line. For all queries the following limits are held: $1 \le k \le m$, $|x| \le 10^8$. The queries are given in order they need to be answered.

It is guaranteed that the sum of sizes of all sets S_k doesn't exceed 10^5.

Output

After each first type query print the required sum on a single line.

Please, do not write the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Sample test(s)

input
5 3 5 5 -5 5 1 -4 2 1 2 4 2 1 4 5 2 2 5 ? 2 + 3 4 ? 1 + 2 1 ? 2
output
-3 4 9