Codeforces Round #202 (Div. 1)

A. 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 $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — 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.

Examples

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)`.

B. 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 $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^8$), 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.

Examples

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

C. 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 $a_1, a_2, \ldots, a_n$ and m sets $S_1, S_2, \ldots, S_m$ 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 $a_1, a_2, \ldots, a_n$ $(|a_i| \le 10^8)$ — 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 $S_{k,1}, S_{k,2}, \ldots, S_{k,|S_k|}$ $(1 \le S_{k,i} \le n)$ — 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.

Examples

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

D. Turtles

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

You've got a table of size $n \times m$. We'll consider the table rows numbered from top to bottom 1 through n, and the columns numbered from left to right 1 through m. Then we'll denote the cell in row x and column y as (x, y).

Initially cell $(1, 1)$ contains two similar turtles. Both turtles want to get to cell (n, m). Some cells of the table have obstacles but it is guaranteed that there aren't any obstacles in the upper left and lower right corner. A turtle (one or the other) can go from cell (x, y) to one of two cells $(x + 1, y)$ and $(x, y + 1)$, as long as the required cell doesn't contain an obstacle. The turtles have had an argument so they don't want to have any chance of meeting each other along the way. Help them find the number of ways in which they can go from cell $(1, 1)$ to cell (n, m).

More formally, find the number of pairs of non-intersecting ways from cell $(1, 1)$ to cell (n, m) modulo 1000000007 $(10^9 + 7)$. Two ways are called non-intersecting if they have exactly two common points — the starting point and the final point.

Input

The first line contains two integers n, m $(2 \le n, m \le 3000)$. Each of the following n lines contains m characters describing the table. The empty cells are marked by characters "`.`", the cells with obstacles are marked by "#".

It is guaranteed that the upper left and the lower right cells are empty.

Output

In a single line print a single integer — the number of pairs of non-intersecting paths from cell $(1, 1)$ to cell (n, m) modulo 1000000007 $(10^9 + 7)$.

Examples

input
4 5 ..... .###. .###. .....
output
1

input
2 3 ... ...
output
1

E. Pilgrims

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

A long time ago there was a land called Dudeland. Dudeland consisted of n towns connected with $n - 1$ bidirectonal roads. The towns are indexed from 1 to n and one can reach any city from any other city if he moves along the roads of the country. There are m monasteries in Dudeland located in m different towns. In each monastery lives a pilgrim.

At the beginning of the year, each pilgrim writes down which monastery is the farthest from the monastery he is living in. If there is more than one farthest monastery, he lists all of them. On the Big Lebowski day each pilgrim picks one town from his paper at random and starts walking to that town.

Walter hates pilgrims and wants to make as many of them unhappy as possible by preventing them from finishing their journey. He plans to destroy exactly one town that does not contain a monastery. A pilgrim becomes unhappy if all monasteries in his list become unreachable from the monastery he is living in.

You need to find the maximum number of pilgrims Walter can make unhappy. Also find the number of ways he can make this maximal number of pilgrims unhappy: the number of possible towns he can destroy.

Input

The first line contains two integers n ($3 \le n \le 10^5$) and m ($2 \le m < n$). The next line contains m distinct integers representing indices of towns that contain monasteries.

Next $n - 1$ lines contain three integers each, a_i, b_i, c_i, indicating that there is an edge between towns a_i and b_i of length c_i ($1 \le a_i, b_i \le n, 1 \le c_i \le 1000, a_i \ne b_i$).

Output

Output two integers: the maximum number of pilgrims Walter can make unhappy and the number of ways in which he can make his plan come true.

Examples

input
8 5 7 2 5 4 8 1 2 1 2 3 2 1 4 1 4 5 2 1 6 1 6 7 8 6 8 10
output
5 1