Problem A. Two Subsequences

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a permutation p of integers 1 to n. You need to partition p into two disjoint increasing subsequences such that the difference between the sizes of these subsequences is the maximum possible. Note that each element must belong to exactly one of the subsequences. The empty subsequence is also allowed.

Input

The first line contains a single integer n ($1 \le n \le 5 \cdot 10^5$), the number of elements in the permutation. The second line contains n integers p_i ($1 \le p_i \le n$), where the i-th of them equals the i-th element of the permutation. It is guaranteed that each integer from $\{1, \ldots, n\}$ appears exactly once among p_i.

Output

Output a single integer, the maximum difference of sizes of subsequences. If it is not possible to partition the permutation into two disjoint increasing subsequences, output -1.

Examples

standard input	standard output
2 2 1	0
4 4 2 3 1	-1
10 2 4 5 1 6 3 7 8 9 10	6
11 1 2 3 4 5 6 7 8 9 10 11	11

Problem B. Numbers on a Circle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given two integers n and k. Determine whether there exists an array a_0, a_1, …, a_{n-1} with the properties:

- it contains all integers from the interval $[0, n-1]$ exactly once,
- for each $i \in [0, n-1]$, we have $(a_i + a_{(i+2) \pmod n}) \pmod k = 0$.

Input

The input contains two integers n and k, the length of the array and the value of the modulo ($2 \le n \le 10^9$, $1 \le k \le 10^9$).

Output

If there exists such an array, output "`Yes`", otherwise output "`No`".

Examples

standard input	standard output
`4 2`	`Yes`
`4 4`	`No`

Problem C. Median Walk

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given an undirected graph with n vertices and m edges. The vertices are numbered by integers from 1 to n. Each vertex has a value written in it; the value in the vertex v is equal to c_v.

A *walk* between two vertices v_1 and v_k is defined as a sequence of vertices v_1, v_2, ..., v_k such that the vertices v_i and v_{i+1} are connected by an edge for all $1 \le i \le k-1$. Note that the same vertex and even the same edge can occur multiple times in a walk.

We define the *median* of k numbers to be the $\lfloor \frac{k}{2} + 1 \rfloor$-th largest number (the 1-st largest number is the smallest). The brackets $\lfloor\rfloor$ denote rounding the number to the highest integer not exceeding it. For this task, we define the value of a walk to be the median value of the values of all vertices in the walk. Note that if the walk has length k, you should consider k numbers (that is, you should consider each vertex as many times as it is visited).

Find the smallest possible value of a walk between vertices 1 and n!

Input

The first line contains two integers n ($2 \le n \le 10^5$) and m ($0 \le m \le 10^5$), the number of vertices and edges in the graph. Then n integers follow, the values c_1, ..., c_n ($1 \le c_i \le 10^9$). The next m lines contain the description of the edges, the i-th of these lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$), the numbers of the endpoints of the i-th edge. The given graph can contain multiple edges and self-loops.

Output

Output a single integer, the minimum possible value of a walk between the vertices 1 and n. If no path exists between vertices 1 and n, output -1.

Examples

standard input	standard output
4 3 4 1 3 2 1 2 2 3 3 4	3
3 3 1 2 3 1 2 2 1 3 3	-1

Problem D. Splitting Text

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a string s of length n that consists of lowercase English letters. We call a sequence of letters a *word* if it consists of at least one vowel and at least one consonant. What is the maximum number of words the given string can be partitioned into?

In this problem, we consider the vowels to be the letters `a`, `i`, `o`, `u`, `e`, `y`, and all the other letters are consonants.

Input

The first line of input contains an integer n ($1 \leq n \leq 10^5$), the length of the string. The next line contains the string s consisting of lowercase English alphabet letters.

Output

Output a single integer, the maximum possible number of words the string can be partitioned into. If it is not possible to partition the string in such a way, output 0.

Examples

standard input	standard output
13 brownfoxjumps	3
4 iota	1
3 you	0

Problem E. Ice Cream

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

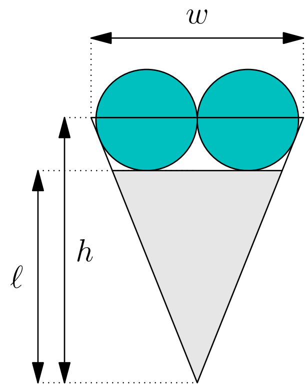
You are an ice cream man. You serve portions of ice cream filling a cone up to some level, and top it with two equally-sized flavored gelato balls. You are also generous, and want to put the largest balls you can. Both balls have to touch the main portion of the ice cream. The balls can touch, but not intersect. The balls can also touch, but not intersect the cone. What is the maximum possible radius of the ice cream balls?

Also, you live in 2D, so the cone is actually an isosceles triangle, and the balls are not spheres, but circles.

Given

- the height of the cone h,
- the level of the main ice cream portion inside the cone ℓ,
- the width of the cone top w,

calculate the maximum possible radius of the balls. Note that there can be different configurations.



Input

The first line of input contains an integer t, the number of test cases ($1 \le t \le 10^4$). Each of the next t lines contains three integers h, ℓ, w ($1 \le h, \ell, w \le 10^9$, $h > \ell$), the description of a single test case.

Output

For each test case, output a single real number, the maximum possible radius of the two balls. The output must be within 10^{-6} absolute or relative error of the correct answer.

Example

standard input	standard output
1 10 8 8	1.908131846

Problem F. Cell Borders

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a strip of paper that consists of n consecutive square cells. The cells are numbered 1 to n from left to right. Each cell has an integer written in it, which is either 0, 1, or 2. Denote the integer in the i-th cell by a_i. You can color black any number of the left or right borders of these cells (there are $n+1$ such borders in total). Is it possible to color the borders in such a way that for each cell i, the number of its borders that is colored is equal to a_i?

For example, if you are given the numbers 1 2 1 0 1 2, then it is possible to color the borders as follows:

Input

The first line of input contains a single integer n, the number of cells ($1 \le n \le 10^5$). The next line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i \le 2$), the numbers of colored borders the cells $1, \ldots, n$ should have.

Output

If it is possible, output "`Yes`" in a single line, otherwise output "`No`".

Examples

standard input	standard output
6 1 2 1 0 1 2	Yes
2 2 0	No

Problem G. Strange Queries

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given n strings, and you need to answer q queries. Each query is composed of two strings l_i and r_i. For such a query, you need to count the number of the given strings such that either it has l_i as a prefix, r_i as a suffix, or both (thus, a string that has l_i for a prefix and r_i for a suffix should be counted once).

Input

The first line contains a single integer n ($1 \le n \le 10^5$), the number of strings. Then follow n strings, each in its own line. The next line contains a single integer q ($1 \le q \le 10^5$), the number of queries. Then q lines follow, the i-th of them contains two strings l_i and r_i. None of the strings in the input is empty.

The sum of the lengths of all strings in the input does not exceed 10^5 (including queries). All strings consist of lowercase English alphabet letters only.

Output

For each query, output the number of the given strings that have either the given prefix or the given suffix (or both).

Example

standard input	standard output
3 bat eca baca 1 ba ca	3

Problem H. Optimize DFS

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

You are given a tree with n vertices. Every vertex v has two values a_v and b_v. You have to process q queries of types:

1. Given v and x, assign $a_v = x$;
2. Given v, print a_v;
3. Given v, implement the following procedure efficiently:

   First, create an array used$[1 \dots n]$ filled with `false` initially. Then run the function DFS(v):

```
DFS(v):
{
    used[v] = true;
    for u from 1 to n do
    {
        if (u and v are connected by an edge)
           and (used[u] == false)
           and (a[v] + b[to] == a[to] + b[v]) then
        {
            DFS(to);
        }
    }
    a[v] = b[v];
}
```

Note that the array used is independent for each query of the 3rd type.

Input

The first line contains two integers $1 \le n, q \le 5 \cdot 10^5$. The second line contains n integers, where the i-th is the initial value of a_i ($0 \le a_i \le 5 \cdot 10^5$). The third line contains n integers, where the i-th is the value of b_i ($0 \le b_i \le 5 \cdot 10^5$).

Each of the next $n-1$ lines describes an edge of the tree. The i-th line contains two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$), the indices of the vertices connected by the i-th edge. It is guaranteed that the given graph is a tree.

The next q lines contain the description of the queries. The i-th query starts with an integer t_i ($1 \le t_i \le 3$) that denotes the type of the query. If $t_i = 1$, then two integers v_i and x_i follow ($1 \le v_i \le n$, $0 \le x_i \le 5 \cdot 10^5$). Otherwise only v_i is given.

Output

For each query of the 2nd type output one integer, the value a_v.

Examples

standard input	standard output
2 6 20 102 10 90 1 2 2 1 2 2 1 2 100 3 1 2 1 2 2	20 102 10 90
6 6 20 30 30 60 60 70 10 20 30 40 50 60 1 2 2 4 2 5 1 3 3 6 1 3 40 3 1 2 1 2 6 2 2 2 4	10 60 20 60

Problem I. Strange Mex

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a multiset of integers, which is initially empty. Then you need to process q queries. With each query, you either add or remove a single integer in the multiset.

After each query you need to calculate the maximum possible mex of the multiset you can obtain, if you can apply the following operations. With a single operation, you can take an element x that appears at least twice in the multiset, remove one of them and add either $x - 1$ or $x + 1$. These operations do not affect the multiset for the next query.

Mex stands for "minimum excluded": the mex of a multiset of numbers is equal to the smallest non-negative integer which is not present in the multiset. For example, $\text{mex}(\{0, 1, 1, 2, 4, 4\}) = 3$. However, in this problem you can convert this multiset to $\{0, 1, 2, 3, 4, 5\}$, in which case mex becomes 6.

Input

The first line contains a single integer q ($1 \le q \le 10^6$), the number of queries. The next q lines describe the queries. The i-th of them contains two integers t_i, a_i ($1 \le t_i \le 2$, $0 \le a_i \le 10^6$):

- if $t_i = 1$, you need to add a_i to the multiset;
- if $t_i = 2$, you need to remove one element a_i from the multiset.

It is guaranteed that for each removal query there will be at least one such integer in the multiset.

Output

After each query, output a single integer, the maximum possible mex of the multiset you can obtain using the given operations.

Example

standard input	standard output
9 1 0 1 1 1 1 1 2 1 4 1 4 2 1 2 2 1 4	1 2 3 4 5 6 5 2 5

Problem J. Mex Grid

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given two arrays of integers a and b. The length of a is n and the length of b is m. Your task is to find any matrix C of size $n \times m$, which satisfies all of the following properties:

- all integers in the interval $[0, n \cdot m - 1]$ should be in C exactly once,
- the mex of the i-th row should be equal to a_i,
- the mex of the j-th column should be equal to b_j.

Mex stands for "minimum excluded": the mex of a set of numbers is equal to the smallest non-negative integer which is not present in the set. For example, $\text{mex}(\{0, 1, 2, 4\}) = 3$.

Input

The first line contains two positive integers n and m, the lengths of the arrays a and b ($1 \le n \cdot m \le 10^5$). The second line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i \le 10^9$). The third line contains m integers $b_1, \ldots, b_m$ ($0 \le b_i \le 10^9$).

Output

If there exists such a matrix, output "`Yes`" in the first line. Otherwise, output "`No`".

In the first case also output the required matrix. Output n rows of m integers, where the the j-th integer of the i-th row is $C_{i,j}$. Each integer from the interval $[0, n \cdot m - 1]$ should appear exactly once in C. If there exist multiple such matrices, output any of those.

Examples

standard input	standard output
2 2 2 0 1 0	Yes 0 1 3 2
2 2 1 1 1 1	No

Problem K. Binary Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a sequence of n bits $a_1, \ldots, a_n$ (each bit can be either 0 or 1). Initially all bits are set to 0. You are also given m pairs of indices (i, j), which are the possible operations you can apply to the sequence. If you apply an operation (i, j), a_i changes to $1 - a_i$ and a_j to $1 - a_j$.

You are also given a target sequence of n bits $b_1, \ldots, b_n$. Determine if it is possible to obtain b from a by applying the available operations (you can also not change the array at all). Each operation can be applied zero or multiple times, and operations may be appplied in any order.

Input

The first line of input contains a single integer, the value of n ($2 \le n \le 10^5$). The following line contains n bits $b_1, \ldots, b_n$ ($0 \le b_i \le 1$).

The next line contains a single integer, the value of m ($1 \le m \le 10^5$). The next m lines each contains two integers l_i and r_i, the indexes of an operation ($1 \le l_i < r_i \le n$). No two operations are equal.

Output

Output "`Yes`", if it is possible, and "`No`" otherwise.

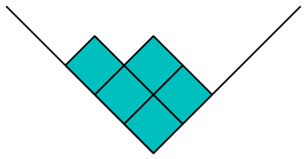
Examples

standard input	standard output
5 1 1 0 1 1 3 1 3 3 4 2 5	Yes
2 0 1 1 1 2	No

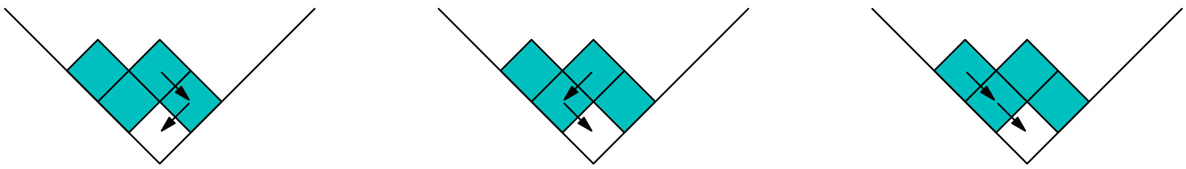
Problem L. Falling Boxes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are in charge of a storage room, which is formed by two infinite diagonal walls joining in a single point at the bottom. There are n equally sized square boxes in the storage. The boxes are naturally pulled by gravity. For example, the placement of the boxes could be as follows:



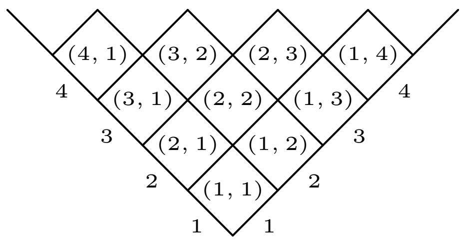
Then the bottom box is removed from the storage. The other boxes start to fall into the now empty space until a stable arrangement is reached again. The boxes can fall in different ways, depending on whether the left or the right upper box fills the empty slot. For example, in the previous figure, there are 3 different options:



Find the total number of different possible falling scenarios modulo $10^9 + 7$.

Input

The first line contains a single integer, the number of boxes n ($1 \le n \le 10^5$). The next n lines describe the boxes, the i-th of them contains two integers x_i and y_i ($1 \le x_i, y_i \le n$). These are the coordinates of the box in the following system: the rows of boxes perpendicular to the left storage wall are numbered by integers starting from 1 and we denote this axis by x; the rows of boxes perpendicular to the right storage wall are numbered by integers starting from 1 and we denote this axis by y. The box at coordinates (x_i, y_i) is located at the intersection of the x_i-th and y_i-th such rows, accordingly.



It is guaranteed that the given arrangement of boxes is stable.

Output

Output a single integer, the number of possible scenarios of falling boxes modulo $10^9 + 7$.

Example

standard input	standard output
5 1 1 1 2 2 2 3 1 2 1	3

Problem M. Big Sum

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given an infinite table A. The rows and columns are indexed by integers 0, 1, 2, and so on. The number in the i-th row and j-th column is equal to $A_{i,j} = 2^i 3^j$.

	0	1	2	3	...
0	1	3	9	27	...
1	2	6	18	54	...
2	4	12	36	108	...
3	8	24	72	216	...
⋮	⋮	⋮	⋮	⋮	⋱

You need to process q queries of the form (i_1, i_2, j_1, j_2), where $i_1 \le i_2$ and $j_1 \le j_2$. For each query, you need to calculate the value

$$\sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} A_{i,j}$$

modulo $10^9 + 7$.

Input

The first line of the input contains a single integer q, the number of queries ($1 \le q \le 10^4$). The i-th of the next q lines contains four integers i_1, i_2, j_1, j_2, the description of the i-th query ($0 \le i_1 \le i_2 \le 10^9$, $0 \le j_1 \le j_2 \le 10^9$).

Output

For each query, output the value of the sum modulo $10^9 + 7$.

Example

standard input	standard output
1 1 2 0 1	24