A. Chess For Three

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alex, Bob and Carl will soon participate in a team chess tournament. Since they are all in the same team, they have decided to practise really hard before the tournament. But it's a bit difficult for them because chess is a game for two players, not three.

So they play with each other according to following rules:

- Alex and Bob play the first game, and Carl is spectating;
- When the game ends, the one who lost the game becomes the spectator in the next game, and the one who was spectating plays against the winner.

Alex, Bob and Carl play in such a way that there are no draws.

Today they have played n games, and for each of these games they remember who was the winner. They decided to make up a log of games describing who won each game. But now they doubt if the information in the log is correct, and they want to know if the situation described in the log they made up was possible (that is, no game is won by someone who is spectating if Alex, Bob and Carl play according to the rules). Help them to check it!

Input

The first line contains one integer n $(1 \leq n \leq 100)$ — the number of games Alex, Bob and Carl played.

Then n lines follow, describing the game log. i-th line contains one integer a_i $(1 \leq a_i \leq 3)$ which is equal to 1 if Alex won i-th game, to 2 if Bob won i-th game and 3 if Carl won i-th game.

Output

Print `YES` if the situation described in the log was possible. Otherwise print `NO`.

Examples

input

```
3
1
1
2
```

output

```
YES
```

input

```
2
1
2
```

output

```
NO
```

Note

In the first example the possible situation is:

1. Alex wins, Carl starts playing instead of Bob;
2. Alex wins, Bob replaces Carl;
3. Bob wins.

The situation in the second example is impossible because Bob loses the first game, so he cannot win the second one.

B. Beautiful Divisors

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

Recently Luba learned about a special kind of numbers that she calls *beautiful* numbers. The number is called *beautiful* iff its binary representation consists of $k + 1$ consecutive ones, and then k consecutive zeroes.

Some examples of beautiful numbers:

- 1_2 (1_{10});
- 110_2 (6_{10});
- 1111000_2 (120_{10});
- 111110000_2 (496_{10}).

More formally, the number is beautiful iff there exists some positive integer k such that the number is equal to $(2^k - 1) * (2^{k - 1})$.

Luba has got an integer number n, and she wants to find its greatest beautiful divisor. Help her to find it!

Input

The only line of input contains one number n ($1 \leq n \leq 10^5$) — the number Luba has got.

Output

Output one number — the greatest beautiful divisor of Luba's number. It is obvious that the answer always exists.

Examples

input

```
3
```

output

```
1
```

input

```
992
```

output

```
496
```

C. Rumor

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

Vova promised himself that he would never play computer games... But recently Firestorm — a well-known game developing company — published their newest game, World of Farcraft, and it became really popular. Of course, Vova started playing it.

Now he tries to solve a quest. The task is to come to a settlement named Overcity and spread a rumor in it.

Vova knows that there are n characters in Overcity. Some characters are friends to each other, and they share information they got. Also Vova knows that he can bribe each character so he or she starts spreading the rumor; i-th character wants c_i gold in exchange for spreading the rumor. When a character hears the rumor, he tells it to all his friends, and they start spreading the rumor to their friends (for free), and so on.

The quest is finished when all n characters know the rumor. What is the minimum amount of gold Vova needs to spend in order to finish the quest?

Take a look at the notes if you think you haven't understood the problem completely.

Input

The first line contains two integer numbers n and m $(1 \le n \le 10^5, 0 \le m \le 10^5)$ — the number of characters in Overcity and the number of pairs of friends.

The second line contains n integer numbers c_i $(0 \le c_i \le 10^9)$ — the amount of gold i-th character asks to start spreading the rumor.

Then m lines follow, each containing a pair of numbers (x_i, y_i) which represent that characters x_i and y_i are friends $(1 \le x_i, y_i \le n, x_i \ne y_i)$. It is guaranteed that each pair is listed at most once.

Output

Print one number — the minimum amount of gold Vova has to spend in order to finish the quest.

Examples

input

```
5 2
2 5 3 4 8
1 4
4 5
```

output

```
10
```

input

```
10 0
1 2 3 4 5 6 7 8 9 10
```

output

```
55
```

input

```
10 5
1 6 2 7 3 8 4 9 5 10
1 2
3 4
5 6
7 8
9 10
```

output

```
15
```

Note

In the first example the best decision is to bribe the first character (he will spread the rumor to fourth character, and the fourth one will spread it to fifth). Also Vova has to bribe the second and the third characters, so they know the rumor.

In the second example Vova has to bribe everyone.

In the third example the optimal decision is to bribe the first, the third, the fifth, the seventh and the ninth characters.

D. Credit Card

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

Recenlty Luba got a credit card and started to use it. Let's consider n consecutive days Luba uses the card.

She starts with 0 money on her account.

In the **evening** of i-th day a transaction a_i occurs. If $a_i > 0$, then a_i bourles are deposited to Luba's account. If $a_i < 0$, then a_i bourles are withdrawn. And if $a_i = 0$, then the amount of money on Luba's account is checked.

In the **morning** of any of n days Luba can go to the bank and deposit any **positive** integer amount of burles to her account. But there is a limitation: the amount of money on the account can never exceed d.

It can happen that the amount of money goes greater than d by some transaction in the evening. In this case answer will be «-1».

Luba must not exceed this limit, and also she wants that **every day her account is checked** (the days when $a_i = 0$) the amount of money on her account is non-negative. It takes a lot of time to go to the bank, so Luba wants to know the minimum number of days she needs to deposit some money to her account (if it is possible to meet all the requirements). Help her!

Input

The first line contains two integers n, d $(1 \le n \le 10^5, 1 \le d \le 10^9)$ —the number of days and the money limitation.

The second line contains n integer numbers $a_1, a_2, \dots a_n$ $(- 10^4 \le a_i \le 10^4)$, where a_i represents the transaction in i-th day.

Output

Print `-1` if Luba cannot deposit the money to her account in such a way that the requirements are met. Otherwise print the minimum number of days Luba has to deposit money.

Examples

input

```
5 10
-1 5 0 -5 3
```

output

```
0
```

input

```
3 4
-10 0 20
```

output

```
-1
```

input

```
5 10
-5 0 10 -11 0
```

output

```
2
```

E. Counting Arrays

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

You are given two positive integer numbers x and y. An array F is called an y-*factorization* of x iff the following conditions are met:

- There are y elements in F, and all of them are integer numbers;
- $\prod_{i=1}^{y} F_i = x$.

You have to count the number of pairwise distinct arrays that are y-factorizations of x. Two arrays A and B are considered different iff there exists at least one index i $(1 \le i \le y)$ such that $A_i \ne B_i$. Since the answer can be very large, print it modulo $10^9 + 7$.

Input

The first line contains one integer q $(1 \le q \le 10^5)$ — the number of testcases to solve.

Then q lines follow, each containing two integers x_i and y_i $(1 \le x_i, y_i \le 10^6)$. Each of these lines represents a testcase.

Output

Print q integers. i-th integer has to be equal to the number of y_i-factorizations of x_i modulo $10^9 + 7$.

Example

input

```
2
6 3
4 2
```

output

```
36
6
```

Note

In the second testcase of the example there are six y-factorizations:

- $\{-4, -1\}$;
- $\{-2, -2\}$;
- $\{-1, -4\}$;
- $\{1, 4\}$;
- $\{2, 2\}$;
- $\{4, 1\}$.

F. Subtree Minimum Query

time limit per test: 6 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given a rooted tree consisting of n vertices. Each vertex has a number written on it; number a_i is written on vertex i.

Let's denote $d(i, j)$ as the distance between vertices i and j in the tree (that is, the number of edges in the shortest path from i to j). Also let's denote the k*-blocked subtree* of vertex x as the set of vertices y such that both these conditions are met:

- x is an ancestor of y (every vertex is an ancestor of itself);
- $d(x, y) \le k$.

You are given m queries to the tree. i-th query is represented by two numbers x_i and k_i, and the answer to this query is the minimum value of a_j among such vertices j such that j belongs to k_i-blocked subtree of x_i.

Write a program that would process these queries quickly!

Note that the queries are given in a modified way.

Input

The first line contains two integers n and r $(1 \le r \le n \le 100000)$ — the number of vertices in the tree and the index of the root, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$ — the numbers written on the vertices.

Then $n - 1$ lines follow, each containing two integers x and y $(1 \le x, y \le n)$ and representing an edge between vertices x and y. It is guaranteed that these edges form a tree.

Next line contains one integer m $(1 \le m \le 10^6)$ — the number of queries to process.

Then m lines follow, i-th line containing two numbers p_i and q_i, which can be used to restore i-th query $(1 \le p_i, q_i \le n)$.

i-th query can be restored as follows:

Let $last$ be the answer for previous query (or 0 if $i = 1$). Then $x_i = ((p_i + last) \; mod \; n) + 1$, and $k_i = (q_i + last) \; mod \; n$.

Output

Print m integers. i-th of them has to be equal to the answer to i-th query.

Example

input

```
5 2
1 3 2 3 5
2 3
5 1
3 4
4 1
2
1 2
2 3
```

output

```
2
5
```