A. Searching Local Minimum,

2 seconds, 512 megabytes, standard input, standard output

This is an interactive problem.

Homer likes arrays a lot and he wants to play a game with you.

Homer has hidden from you a permutation $a_1, a_2, \dots, a_n$ of integers 1 to n. You are asked to find any index k ($1 \le k \le n$) which is a local minimum.

For an array $a_1, a_2, \dots, a_n$, an index i ($1 \le i \le n$) is said to be a *local minimum* if $a_i < \min\{a_{i-1}, a_{i+1}\}$, where $a_0 = a_{n+1} = +\infty$. An array is said to be a permutation of integers 1 to n, if it contains all integers from 1 to n exactly once.

Initially, you are only given the value of n without any other information about this permutation.

At each interactive step, you are allowed to choose any i ($1 \le i \le n$) and make a query with it. As a response, you will be given the value of a_i.

You are asked to find any index k which is a local minimum **after at most 100 queries**.

Interaction

You begin the interaction by reading an integer n ($1 \le n \le 10^5$) on a separate line.

To make a query on index i ($1 \le i \le n$), you should output "? i" in a separate line. Then read the value of a_i in a separate line. The number of the "?" queries is limited within 100.

When you find an index k ($1 \le k \le n$) which is a local minimum, output "! k" in a separate line and terminate your program.

In case your query format is invalid, or you have made more than 100 "?" queries, you will receive **Wrong Answer** verdict.

After printing a query do not forget to output end of line and flush the output. Otherwise, you will get Idleness limit exceeded. To do this, use:

- fflush(stdout) or cout.flush() in C++;
- System.out.flush() in Java;
- flush(output) in Pascal;
- stdout.flush() in Python;
- see documentation for other languages.

Hack Format

The first line of the hack should contain a single integer n ($1 \le n \le 10^5$).

The second line should contain n distinct integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le n$).

input

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

output

```
? 1
? 2
? 3
? 4
? 5
! 3
```

In the example, the first line contains an integer 5 indicating that the length of the array is $n = 5$.

The example makes five "?" queries, after which we conclude that the array is $a = [3, 2, 1, 4, 5]$ and $k = 3$ is local minimum.

B1. Painting the Array I,

2 seconds, 512 megabytes, standard input, standard output

The only difference between the two versions is that this version asks the *maximal* possible answer.

Homer likes arrays a lot. Today he is painting an array $a_1, a_2, \dots, a_n$ with two kinds of colors, **white** and **black**. A painting assignment for $a_1, a_2, \dots, a_n$ is described by an array $b_1, b_2, \dots, b_n$ that b_i indicates the color of a_i (0 for white and 1 for black).

According to a painting assignment $b_1, b_2, \dots, b_n$, the array a is split into two new arrays $a^{(0)}$ and $a^{(1)}$, where $a^{(0)}$ is the sub-sequence of all white elements in a and $a^{(1)}$ is the sub-sequence of all black elements in a. For example, if $a = [1, 2, 3, 4, 5, 6]$ and $b = [0, 1, 0, 1, 0, 0]$, then $a^{(0)} = [1, 3, 5, 6]$ and $a^{(1)} = [2, 4]$.

The number of segments in an array $c_1, c_2, \dots, c_k$, denoted $seg(c)$, is the number of elements if we merge all adjacent elements with the same value in c. For example, the number of segments in $[1, 1, 2, 2, 3, 3, 3, 2]$ is 4, because the array will become $[1, 2, 3, 2]$ after merging adjacent elements with the same value. Especially, the number of segments in an empty array is 0.

Homer wants to find a painting assignment b, according to which the number of segments in both $a^{(0)}$ and $a^{(1)}$, i.e. $seg(a^{(0)}) + seg(a^{(1)})$, is as **large** as possible. Find this number.

Input

The first line contains an integer n ($1 \le n \le 10^5$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le n$).

Output

Output a single integer, indicating the **maximal** possible total number of segments.

input

```
7
1 1 2 2 3 3 3
```

output

```
6
```

input

```
7
1 2 3 4 5 6 7
```

output

```
7
```

In the first example, we can choose $a^{(0)} = [1, 2, 3, 3]$, $a^{(1)} = [1, 2, 3]$ and $seg(a^{(0)}) = seg(a^{(1)}) = 3$. So the answer is $3 + 3 = 6$.

In the second example, we can choose $a^{(0)} = [1, 2, 3, 4, 5, 6, 7]$ and $a^{(1)}$ is empty. We can see that $seg(a^{(0)}) = 7$ and $seg(a^{(1)}) = 0$. So the answer is $7 + 0 = 7$.

B2. Painting the Array II,

2 seconds, 512 megabytes, standard input, standard output

The only difference between the two versions is that this version asks the *minimal* possible answer.

Homer likes arrays a lot. Today he is painting an array $a_1, a_2, \dots, a_n$ with two kinds of colors, **white** and **black**. A painting assignment for $a_1, a_2, \dots, a_n$ is described by an array $b_1, b_2, \dots, b_n$ that b_i indicates the color of a_i (0 for white and 1 for black).

According to a painting assignment $b_1, b_2, \dots, b_n$, the array a is split into two new arrays $a^{(0)}$ and $a^{(1)}$, where $a^{(0)}$ is the sub-sequence of all white elements in a and $a^{(1)}$ is the sub-sequenc
a. For example, if $a = [1, 2, 3, 4, 5, 6]$ and $b = [$
$a^{(0)} = [1, 3, 5, 6]$ and $a^{(1)} = [2, 4]$.

The number of segments in an array $c_1, c_2, \ldots, c_k$, denoted $seg(c)$, is the number of elements if we merge all adjacent elements with the same value in c. For example, the number of segments in $[1, 1, 2, 2, 3, 3, 3, 2]$ is 4, because the array will become $[1, 2, 3, 2]$ after merging adjacent elements with the same value. Especially, the number of segments in an empty array is 0.

Homer wants to find a painting assignment b, according to which the number of segments in both $a^{(0)}$ and $a^{(1)}$, i.e. $seg(a^{(0)}) + seg(a^{(1)})$, is as **small** as possible. Find this number.

Input

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

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$).

Output

Output a single integer, indicating the **minimal** possible total number of segments.

input
6 1 2 3 1 2 2
output
4

input
7 1 2 1 2 1 2 1
output
2

In the first example, we can choose $a^{(0)} = [1, 1, 2, 2]$, $a^{(1)} = [2, 3]$ and $seg(a^{(0)}) = seg(a^{(1)}) = 2$. So the answer is $2 + 2 = 4$.

In the second example, we can choose $a^{(0)} = [1, 1, 1, 1]$, $a^{(1)} = [2, 2, 2]$ and $seg(a^{(0)}) = seg(a^{(1)}) = 1$. So the answer is $1 + 1 = 2$.

C. Continuous City, 2 seconds, 512 megabytes, standard input, standard output

Some time ago Homer lived in a beautiful city. There were n blocks numbered from 1 to n and m directed roads between them. Each road had a positive length, and each road went from the block with the smaller index to the block with the larger index. For every two (different) blocks, there was at most one road between them.

Homer discovered that for some two numbers L and R the city was (L, R)-*continuous*.

The city is said to be (L, R)-continuous, if

1. all paths from block 1 to block n are of length between L and R (inclusive); and
2. for every $L \leq d \leq R$, there is **exactly one** path from block 1 to block n whose length is d.

A path from block u to block v is a sequence $u = x_0 \to x_1 \to x_2 \to \cdots \to x_k = v$, where there is a road from block x_{i-1} to block x_i for every $1 \leq i \leq k$. The length of a path is the sum of lengths over all roads in the path. Two paths $x_0 \to x_1 \to \cdots \to x_k$ and $y_0 \to y_1 \to \cdots \to y_l$ are different, if $k \neq l$ or $x_i \neq y_i$ for some $0 \leq i \leq \min\{k, l\}$.

After moving to another city, Homer only remembers the two special numbers L and R but forgets the numbers n and m of blocks and roads, respectively, and how blocks are connected by roads. However, he believes the number of blocks should be no larger than 32 (because the city was small).

As the best friend of Homer, please tell him whether it is possible to find a (L, R)-continuous city or not.

Input

The single line contains two integers L and R ($1 \leq L \leq R \leq 10^6$).

Output

If it is impossible to find a (L, R)-continuous city within 32 blocks, print "NO" in a single line.

Otherwise, print "YES" in the first line followed by a description of a (L, R)-continuous city.

The second line should contain two integers n ($2 \leq n \leq 32$) and m ($1 \leq m \leq \frac{n(n-1)}{2}$), where n denotes the number of blocks and m denotes the number of roads.

Then m lines follow. The i-th of the m lines should contain three integers a_i, b_i ($1 \leq a_i < b_i \leq n$) and c_i ($1 \leq c_i \leq 10^6$) indicating that there is a directed road from block a_i to block b_i of length c_i.

It is required that for every two blocks, there should be **no more than 1** road connecting them. That is, for every $1 \leq i < j \leq m$, either $a_i \neq a_j$ or $b_i \neq b_j$.

input
1 1
output
YES 2 1 1 2 1

input
4 6
output
YES 5 6 1 2 3 1 3 4 1 4 5 2 5 1 3 5 1 4 5 1

In the first example there is only one path from block 1 to block $n = 2$, and its length is 1.

In the second example there are three paths from block 1 to block $n = 5$, which are $1 \to 2 \to 5$ of length 4, $1 \to 3 \to 5$ of length 5 and $1 \to 4 \to 5$ of length 6.

D. Odd Mineral Resource, 5 seconds, 1024 megabytes, standard input, standard output

In Homer's country, there are n cities numbered 1 to n and they form a tree. That is, there are $(n-1)$ undirected roads between these n cities and every two cities can reach each other through these roads.

Homer's country is an industrial country, and each of the n cities in it contains some mineral resource. The mineral resource of city i is labeled a_i.

Homer is given the plans of the country in the following q years. The plan of the i-th year is described by four parameters u_i, v_i, l_i and r_i, and he is asked to find any mineral resource c_i such that the following two conditions hold:

- mineral resource c_i appears an **odd** number of times between city u_i and city v_i; and
- $l_i \leq c_i \leq r_i$.

As the best friend of Homer, he asks you for help. For every plan, find any such mineral resource c_i, or tell him that there doesn't exist one.

Input

The first line contains two integers n ($1 \leq n \leq 3 \cdot 10^5$) and q ($1 \leq q \leq 3 \cdot 10^5$), indicating the number of cities and the number of plans.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq n$).

Then the i-th line of the following $(n-1)$ lines contains two integers x_i and y_i ($1 \leq x_i, y_i \leq n$) with $x_i \neq y_i$, indicating that there is a bidirectional road between city x_i and city y_i. It i
given roads form a tree.

Then the i-th line of the following q lines contains four integers u_i, v_i, l_i, r_i $(1 \le u_i \le n, 1 \le v_i \le n, 1 \le l_i \le r_i \le n)$, indicating the plan of the i-th year.

Output

Print q lines, the i-th of which contains an integer c_i such that

- $c_i = -1$ if there is no such mineral resource that meets the required condition; or
- c_i is the label of the chosen mineral resource of the i-th year. The chosen mineral resource c_i should meet those conditions in the i-th year described above in the problem statement. If there are multiple choices of c_i, you can print any of them.

input

```
6 8
3 2 1 3 1 3
1 2
1 3
2 4
2 5
4 6
3 5 1 1
3 5 1 3
3 5 1 3
1 1 2 2
1 1 3 3
1 4 1 5
1 6 1 3
1 6 1 3
```

output

```
-1
2
3
-1
3
2
2
3
```

In the first three queries, there are four cities between city 3 and city 5, which are city 1, city 2, city 3 and city 5. The mineral resources appear in them are mineral resources 1 (appears in city 3 and city 5), 2 (appears in city 2) and 3 (appears in city 1). It is noted that

- The first query is only to check whether mineral source 1 appears an odd number of times between city 3 and city 5. The answer is no, because mineral source 1 appears twice (an even number of times) between city 3 and city 5.
- The second and the third queries are the same but they can choose different mineral resources. Both mineral resources 2 and 3 are available.

E. School Clubs, 4 seconds, 512 megabytes, standard input, standard output

In Homer's school, there are n students who love clubs.

Initially, there are m clubs, and each of the n students is in exactly one club. In other words, there are a_i students in the i-th club for $1 \le i \le m$ and $a_1 + a_2 + \dots + a_m = n$.

The n students are so unfriendly that every day one of them (chosen **uniformly at random** from all of the n students) gets angry. The student who gets angry will do one of the following things.

- With probability $\frac{1}{2}$, he leaves his current club, then creates a new club himself and joins it. There is only one student (himself) in the new club he creates.
- With probability $\frac{1}{2}$, he does not create new clubs. In this case, he changes his club to a new one (possibly the same club he is in currently) with probability proportional to the number of students in it. Formally, suppose there are k clubs and there are b_i students in the i-th club for $1 \le i \le k$ (before the student gets angry). He leaves his current club, and then joins the i-th club with probability $\frac{b_i}{n}$.

We note that when a club becomes empty, students will never join it because any student who gets angry will join an empty club with probability 0 according to the above statement.

Homer wonders the expected number of days until every student is in the same club for the first time.

We can prove that the answer can be represented as a rational number $\frac{p}{q}$ with $\gcd(p, q) = 1$. Therefore, you are asked to find the value of $pq^{-1} \bmod 998\,244\,353$. It can be shown that $q \bmod 998\,244\,353 \ne 0$ under the given constraints of the problem.

Input

The first line contains an integer m $(1 \le m \le 1000)$ — the number of clubs initially.

The second line contains m integers $a_1, a_2, \dots, a_m$ $(1 \le a_i \le 4 \cdot 10^8)$ with $1 \le a_1 + a_2 + \dots + a_m \le 4 \cdot 10^8$, where a_i denotes the number of students in the i-th club initially.

Output

Print one integer — the expected number of days until every student is in the same club for the first time, modulo $998\,244\,353$.

input

```
2
1 1
```

output

```
4
```

input

```
2
1 2
```

output

```
18
```

input

```
3
1 1 1
```

output

```
21
```

input

```
1
400000000
```

output

```
0
```

input

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

output

```
737609878
```

In the first example, no matter which student gets angry, the two students will become in the same club with probability $\frac{1}{4}$. So the expected number of days until every student is in the same club should be 4.

In the second example, we note that in the first day:

- The only student in the first club will get angry with probability $\frac{1}{3}$. If he gets angry, then he will create a new club and join it with probability $\frac{1}{2}$ (In this case, there will be three clubs which have $0, 1, 2$ students in it, respectively), leave his current club and join the second club with probability $\frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3}$, or stay still with probability $\frac{1}{2} \cdot \frac{1}{3} = \frac{1}{6}$;
- Each of the two students in the second club will get angry with probability $\frac{1}{3}$. If one of them gets angry, then he will create a new club and join it with probability $\frac{1}{2}$, leave his current club and join the second club with probability $\frac{1}{2} \cdot \frac{1}{3} = \frac{1}{6}$, or stay still with probability $\frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3}$.

In the fourth example, there is only one club initially. That is, every student has already been in the same club. So the answer is 0.