Problem A. Ordenar 1

Time limit	500 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

Given N integers [$0 < N \le 10^5$], count the total pairs of integers that have a difference of K. (Everything can be done with 32 bit integers).

Input

1st line contains integers N and K.

2nd line contains N integers of the set. All the N numbers are distinct.

Output

One integer - the number of pairs of numbers that have a difference of K.

Example

Input	Output
5 2 1 5 3 4 2	3

Input	Output
10 1 363374326 364147530 61825163 1073065718 1281246024 1399469912 428047635 491595254 879792181 1069262793	0

Problem B. Ordenar 2

Time limit	7000 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

In one of the internet banks thousands of operations are being performed every day. Since certain customers do business more actively than others, some of the bank accounts occur many times in the list of operations. Your task is to sort the bank account numbers in ascending order. If an account appears twice or more in the list, write the number of repetitions just after the account number. The format of accounts is as follows: **2** control digits, an **8**-digit code of the bank, **16** digits identifying the owner (written in groups of four digits), for example (at the end of each line there is exactly one space):

30 10103538 2222 1233 6160 0142

Banks are real-time institutions and they need FAST solutions. If you feel you can meet the challenge within a very stringent time limit, go ahead! A well designed sorting algorithm in a fast language is likely to succeed.

Input

t [the number of tests <= **5**]
n [the number of accounts<= **100 000**]
[list of accounts]
[empty line]
[next test cases]

Output

[sorted list of accounts with the number of repeated accounts]
[empty line]
[other results]

Example

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>2
6
03 10103538 2222 1233 6160 0142
03 10103538 2222 1233 6160 0141
30 10103538 2222 1233 6160 0141
30 10103538 2222 1233 6160 0142
30 10103538 2222 1233 6160 0141
30 10103538 2222 1233 6160 0142

5
30 10103538 2222 1233 6160 0144
30 10103538 2222 1233 6160 0142
30 10103538 2222 1233 6160 0145
30 10103538 2222 1233 6160 0146
30 10103538 2222 1233 6160 0143</pre></td><td><pre>03 10103538 2222 1233 6160 0141 1
03 10103538 2222 1233 6160 0142 1
30 10103538 2222 1233 6160 0141 2
30 10103538 2222 1233 6160 0142 2

30 10103538 2222 1233 6160 0142 1
30 10103538 2222 1233 6160 0143 1
30 10103538 2222 1233 6160 0144 1
30 10103538 2222 1233 6160 0145 1
30 10103538 2222 1233 6160 0146 1</pre></td></tr>
</table>

Problem C. Ordenar 3

Time limit 1000 ms
Mem limit 524288 kB

You are given a list of n integers, and your task is to calculate the number of *distinct* values in the list.

Input

The first input line has an integer n: the number of values.

The second line has n integers $x_1, x_2, \ldots, x_n$.

Output

Print one integers: the number of distinct values.

Constraints

- $1 \leq n \leq 2 \cdot 10^5$
- $1 \leq x_i \leq 10^9$

Example

Input	Output
5 2 3 2 2 3	2

Problem D. Ordenar 4

Time limit 1000 ms
Mem limit 524288 kB

You are given an array that contains each number between $1 \ldots n$ exactly once. Your task is to collect the numbers from 1 to n in increasing order.

On each round, you go through the array from left to right and collect as many numbers as possible.

Given m operations that swap two numbers in the array, your task is to report the number of rounds after each operation.

Input

The first line has two integers n and m: the array size and the number of operations.

The next line has n integers $x_1, x_2, \ldots, x_n$: the numbers in the array.

Finally, there are m lines that describe the operations. Each line has two integers a and b: the numbers at positions a and b are swapped.

Output

Print m integers: the number of rounds after each swap.

Constraints

- $1 \leq n, m \leq 2 \cdot 10^5$
- $1 \leq a, b \leq n$

Example

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>5 3
4 2 1 5 3
2 3
1 5
2 3</td><td>2
3
4</td></tr>
</table>

Problem E. Struct 1

Time limit 2000 ms
Mem limit 262144 kB

Problem Statement

There are N positive integers written on a blackboard: $A_1, ..., A_N$.

Snuke can perform the following operation when all integers on the blackboard are even:

- Replace each integer X on the blackboard by X divided by 2.

Find the maximum possible number of operations that Snuke can perform.

Constraints

- $1 \leq N \leq 200$
- $1 \leq A_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

N
$A_1 \ A_2 \ \ldots \ A_N$

Output

Print the maximum possible number of operations that Snuke can perform.

Sample 1

Input	Output
3 8 12 40	2

Initially, $[8, 12, 40]$ are written on the blackboard. Since all those integers are even, Snuke can perform the operation.

After the operation is performed once, $[4, 6, 20]$ are written on the blackboard. Since all those integers are again even, he can perform the operation.

After the operation is performed twice, $[2, 3, 10]$ are written on the blackboard. Now, there is an odd number 3 on the blackboard, so he cannot perform the operation any more.

Thus, Snuke can perform the operation at most twice.

Sample 2

Input	Output
`4` `5 6 8 10`	`0`

Since there is an odd number 5 on the blackboard already in the beginning, Snuke cannot perform the operation at all.

Sample 3

Input	Output

`6`
`382253568 723152896 37802240 379425024`
`404894720 471526144`
	`8`

Problem F. Struct 2

Time limit 2000 ms
Mem limit 262144 kB

Problem Statement

AtCoDeer the deer is going on a trip in a two-dimensional plane. In his plan, he will depart from point $(0, 0)$ at time 0, then for each i between 1 and N (inclusive), he will visit point (x_i, y_i) at time t_i.

If AtCoDeer is at point (x, y) at time t, he can be at one of the following points at time $t + 1$: $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ and $(x, y - 1)$. Note that **he cannot stay at his place**. Determine whether he can carry out his plan.

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq x_i \leq 10^5$
- $0 \leq y_i \leq 10^5$
- $1 \leq t_i \leq 10^5$
- $t_i < t_{i+1}$ $(1 \leq i \leq N - 1)$
- All input values are integers.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
t_1 \ x_1 \ y_1 \\
t_2 \ x_2 \ y_2 \\
\vdots \\
t_N \ x_N \ y_N
\end{array}
$$

Output

If AtCoDeer can carry out his plan, print `Yes`; if he cannot, print `No`.

Sample 1

Input	Output
2 3 1 2 6 1 1	Yes

For example, he can travel as follows: $(0, 0)$, $(0, 1)$, $(1, 1)$, $(1, 2)$, $(1, 1)$, $(1, 0)$, then $(1, 1)$.

Sample 2

Input	Output
1 2 100 100	No

It is impossible to be at $(100, 100)$ two seconds after being at $(0, 0)$.

Sample 3

Input	Output
2 5 1 1 100 1 1	No

Problem G. Struct 3

Time limit	101 ms
Mem limit	1572864 kB
Code length Limit	50000 B
OS	Linux

Byteotian Interstellar Union (BIU) has recently discovered a new planet in a nearby galaxy. The planet is unsuitable for colonisation due to strange meteor showers, which on the other hand make it an exceptionally interesting object of study.

The member states of BIU have already placed space stations close to the planet's orbit. The stations' goal is to take samples of the rocks flying by. The BIU Commission has partitioned the orbit into m sectors, numbered from 1 to m, where the sectors 1 and m are adjacent. In each sector there is a single space station, belonging to one of the n member states.

Each state has declared a number of meteor samples it intends to gather before the mission ends. Your task is to determine, for each state, when it can stop taking samples, based on the meteor shower predictions for the years to come.

Input

The first line of the standard input gives two integers, n and m ($1 \le n, m \le 300\,000$), separated by a single space, that denote, respectively, the number of BIU member states and the number of sectors the orbit has been partitioned into.

In the second line there are m integers o_i ($1 \le o_i \le n$), separated by single spaces, that denote the states owning stations in successive sectors.

In the third line there are n integers p_i ($1 \le p_i \le 10^9$), separated by single spaces, that denote the numbers of meteor samples that the successive states intend to gather.

In the fourth line there is a single integer k ($1 \le k \le 300\,000$) that denotes the number of meteor showers predictions. The following k lines specify the (predicted) meteor showers chronologically. The i-th of these lines holds three integers l_i, r_i, a_i (separated by single spaces), which denote that a meteor shower is expected in sectors $l_i, l_{i+1}, \ldots, r_i$ (if $l_i \le r_i$) or sectors $l_i, l_{i+1}, \ldots, m, 1, \ldots, r_i$ (if $l_i > r_i$), which should provide each station in those sectors with a_i meteor samples ($1 \le a_i \le 10^9$).

Output

Your program should print n lines on the standard output. The i-th of them should contain a single integer w_i, denoting the number of shower after which the stations belonging to the i-th state are expected to gather at least p_i samples, or the word NIE (Polish for *no*) if that state is not expected to gather enough samples in the foreseeable future.

Example

For the input data:

```
3 5
1 3 2 1 3
10 5 7
3
4 2 4
1 3 1
3 5 2
```

the correct result is:

```
3
NIE
1
```