Codeforces Round #407 (Div. 1)

A. Functions again

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

Something happened in Uzhlyandia again... There are riots on the streets... Famous Uzhlyandian superheroes Shean the Sheep and Stas the Giraffe were called in order to save the situation. Upon the arriving, they found that citizens are worried about maximum values of the Main Uzhlyandian Function f, which is defined as follows:

In the above formula, $1 \le l < r \le n$ must hold, where n is the size of the Main Uzhlyandian Array a, and $|x|$ means absolute value of x. But the heroes skipped their math lessons in school, so they asked you for help. Help them calculate the maximum value of f among all possible values of l and r for the given array a.

Input

The first line contains single integer n ($2 \le n \le 10^5$) — the size of the array a.

The second line contains n integers $a_1, a_2, ..., a_n$ ($-10^9 \le a_i \le 10^9$) — the array elements.

Output

Print the only integer — the maximum value of f.

Examples

input
5 1 4 2 3 1
output
3

input
4 1 5 4 7
output
6

Note

In the first sample case, the optimal value of f is reached on intervals $[1, 2]$ and $[2, 5]$.

In the second case maximal value of f is reachable only on the whole array.

B. Weird journey

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

Little boy Igor wants to become a traveller. At first, he decided to visit all the cities of his motherland — Uzhlyandia.

It is widely known that Uzhlyandia has n cities connected with m bidirectional roads. Also, there are no two roads in the country that connect the same pair of cities, but roads starting and ending in the same city can exist. Igor wants to plan his journey beforehand. Boy thinks a path is *good* if the path goes over $m - 2$ roads twice, and over the other 2 exactly once. The good path can start and finish in any city of Uzhlyandia.

Now he wants to know how many different good paths are in Uzhlyandia. Two paths are considered different if the sets of roads the paths goes over exactly once differ. Help Igor — calculate the number of good paths.

Input

The first line contains two integers n, m $(1 \le n, m \le 10^6)$ — the number of cities and roads in Uzhlyandia, respectively.

Each of the next m lines contains two integers u and v $(1 \le u, v \le n)$ that mean that there is road between cities u and v.

It is guaranteed that no road will be given in the input twice. That also means that for every city there is no more than one road that connects the city to itself.

Output

Print out the only integer — the number of good paths in Uzhlyandia.

Examples

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

input
5 3 1 2 2 3 4 5
output
0

input
2 2 1 1 1 2
output
1

Note

In first sample test case the good paths are:

- $2 \to 1 \to 3 \to 1 \to 4 \to 1 \to 5$,
- $2 \to 1 \to 3 \to 1 \to 5 \to 1 \to 4$,
- $2 \to 1 \to 4 \to 1 \to 5 \to 1 \to 3$,
- $3 \to 1 \to 2 \to 1 \to 4 \to 1 \to 5$,
- $3 \to 1 \to 2 \to 1 \to 5 \to 1 \to 4$,
- $4 \to 1 \to 2 \to 1 \to 3 \to 1 \to 5$.

There are good paths that are same with displayed above, because the sets of roads they pass over once are same:

- $2 \to 1 \to 4 \to 1 \to 3 \to 1 \to 5$,
- $2 \to 1 \to 5 \to 1 \to 3 \to 1 \to 4$,
- $2 \to 1 \to 5 \to 1 \to 4 \to 1 \to 3$,
- $3 \to 1 \to 4 \to 1 \to 2 \to 1 \to 5$,
- $3 \to 1 \to 5 \to 1 \to 2 \to 1 \to 4$,
- $4 \to 1 \to 3 \to 1 \to 2 \to 1 \to 5$,
- and all the paths in the other direction.

Thus, the answer is 6.

In the second test case, Igor simply can not walk by all the roads.

In the third case, Igor walks once over every road.

C. The Great Mixing

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

Sasha and Kolya decided to get drunk with Coke, again. This time they have k types of Coke. i-th type is characterised by its carbon dioxide concentration . Today, on the party in honour of Sergiy of Vancouver they decided to prepare a glass of Coke with carbon dioxide concentration . The drink should also be tasty, so the glass can contain only integer number of liters of each Coke type (some types can be not presented in the glass). Also, they want to minimize the total volume of Coke in the glass.

Carbon dioxide concentration is defined as the volume of carbone dioxide in the Coke divided by the total volume of Coke. When you mix two Cokes, the volume of carbon dioxide sums up, and the total volume of Coke sums up as well.

Help them, find the minimal natural number of liters needed to create a glass with carbon dioxide concentration . Assume that the friends have unlimited amount of each Coke type.

Input

The first line contains two integers n, k ($0 \le n \le 1000$, $1 \le k \le 10^6$) — carbon dioxide concentration the friends want and the number of Coke types.

The second line contains k integers $a_1, a_2, ..., a_k$ ($0 \le a_i \le 1000$) — carbon dioxide concentration of each type of Coke. Some Coke types can have same concentration.

Output

Print the minimal natural number of liter needed to prepare a glass with carbon dioxide concentration , or `-1` if it is impossible.

Examples

input
400 4 100 300 450 500
output
2

input
50 2 100 25
output
3

Note

In the first sample case, we can achieve concentration using one liter of Coke of types and : .

In the second case, we can achieve concentration using two liters of type and one liter of type: .

D. Finding lines

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

After some programming contest Roma decided to try himself in tourism. His home country Uzhlyandia is a Cartesian plane. He wants to walk along each of the Main Straight Lines in Uzhlyandia. It is known that each of these lines is a straight line parallel to one of the axes (i.e. it is described with the equation $x = a$ or $y = a$, where a is integer called the coordinate of this line).

Roma lost his own map, so he should find out the coordinates of all lines at first. Uncle Anton agreed to help him, using the following rules:

- Initially Roma doesn't know the number of vertical and horizontal lines and their coordinates;
- Roma can announce integer coordinates of some point in Uzhlandia, and Anton then will tell him the minimum among the distances from the chosen point to each of the lines. However, since the coordinates of the lines don't exceed 10^8 by absolute value, Roma can't choose a point with coordinates exceeding 10^8 by absolute value.

Uncle Anton is in a hurry to the UOI (Uzhlandian Olympiad in Informatics), so he can only answer no more than $3 \cdot 10^5$ questions.

The problem is that Roma doesn't know how to find out the coordinates of the lines. Write a program that plays Roma's role and finds the coordinates.

Input

There is no input initially. Your program should make queries to get information.

It is guaranteed that the number of horizontal and vertical lines is at least 1 and less than or equal to 10^4 for each type.

Interaction

To make a query, print a line "`0 x y`" ($-10^8 \le x, y \le 10^8$), where x and y are the coordinates of the point. After each query you need to print end-of-line, make "`flush`" operation, and then read the answer to the query — the minimum among the distances prom this point to the Main Straight Lines of Uzhlyandia.

You can do no more than $3 \cdot 10^5$ queries.

When you are ready to print the answer, print three lines:

1. In the first line print "`1 n m`", where n is the number of vertical lines (parallel to OY), and m is the number of horizontal lines (parallel to OX).
2. In the second line print n integers $x_1, x_2, ..., x_n$ — the coordinates of the vertical lines.
3. In the third line in the same format print m integers $y_1, y_2, ..., y_m$ — the coordinates of the horizontal lines.

You can print coordinates in arbitrary order.

To make "`flush`", you can use (just after printing a query/answer and end-of-line):

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

You will get `Wrong Answer` if you make more queries than allowed or make an invalid query.

You can get `Idleness Limit Exceeded` if you don't print anything or if you forget to flush the output.

If at any moment your program reads `-1` as an answer, it should immediately exit normally (for example, by calling `exit(0)`). You will get `Wrong Answer` in this case, it means that you made more queries than allowed, or made an invalid query. If you ignore this, you can get other verdicts since your program will continue to read from a closed stream.

Making test for hacking

The first line should contain two integers n and m ($1 \le n, m \le 10^4$).

The second line should contain n distinct integers x_i ($-10^8 \le x_i \le 10^8$) — the coordinates of the vertical lines.

The third line should contain m distinct integers y_i ($-10^8 \le y_i \le 10^8$) — the coordinates of the horizontal lines.

You can write coordinates in arbitrary order.

You can see the example case in the notes.

Example

input
1 1 3

```
2
```

output

```
0 1 2
0 -2 -2
0 5 6
0 -2 2
1 1 2
2
0 -3
```

Note

The example test is

```
1 2
2
0 -3
```

The minimum distances are:

- from $(1, 2)$ to $x = 2$;
- from $(- 2, \ - 2)$ to $y = \ - 3$;
- from $(5, 6)$ to $x = 2$;
- from $(- 2, 2)$ to $y = 0$.

E. New task

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

On the 228-th international Uzhlyandian Wars strategic game tournament teams from each country are called. The teams should consist of 5 participants.

The team of Uzhlyandia will consist of soldiers, because there are no gamers.

Masha is a new minister of defense and gaming. The prime duty of the minister is to calculate the efficiency of the Uzhlandian army. The army consists of n soldiers standing in a row, enumerated from 1 to n. For each soldier we know his *skill* in Uzhlyandian Wars: the i-th soldier's skill is a_i.

It was decided that the team will consist of three players and two assistants. The skills of players should be same, and the assistants' skills should not be greater than the players' skill. Moreover, it is important for Masha that one of the assistants should stand in the row to the left of the players, and the other one should stand in the row to the right of the players. Formally, a team is five soldiers with indexes i, j, k, l, p, such that $1 \le i < j < k < l < p \le n$ and $a_i \le a_j = a_k = a_l \ge a_p$.

The efficiency of the army is the number of different teams Masha can choose. Two teams are considered different if there is such i such that the i-th soldier is a member of one team, but not a member of the other team.

Initially, all players are able to be players. For some reasons, sometimes some soldiers become unable to be players. Sometimes some soldiers, that were unable to be players, become able to be players. At any time any soldier is able to be an assistant. Masha wants to control the efficiency of the army, so she asked you to tell her the number of different possible teams modulo 1000000007 $(10^9 + 7)$ after each change.

Input

The first line contains single integer n $(1 \le n \le 10^5)$ — the number of soldiers in Uzhlyandia.

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ — the soldiers' skills.

The third line contains single integer m $(1 \le m \le 10^5)$ — the number of changes.

The next m lines contain the changes, each change is described with two integers t and x $(1 \le t \le 2,\ 1 \le x \le n)$ on a separate line. If $t = 1$, then the x-th soldier is unable to be a player after this change. If $t = 2$, then the x-th soldier is able to be a player after this change.

It is guaranteed that before each query of the first type the soldier is able to be a player, and before each query of the second type the soldier is unable to be a player.

Output

Print m integers — the number of distinct teams after each change.

Print the answers modulo 1000000007 $(10^9 + 7)$.

Examples

input
6 1 1 1 1 1 1 2 1 3 2 3
output
1 6

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

Note

In the first example, after the first change the only team consists of soldiers $[1, 2, 4, 5, 6]$. After the second change any five soldiers can form a team.

In the first example after the first change the only team is soldiers $[1, 2, 3, 7, 8]$. After the second change the possible teams are: $[1, 2, 3, 5, 7]$, $[1, 2, 3, 5, 8]$, $[1, 2, 3, 7, 8]$, $[1, 2, 5, 7, 8]$, $[1, 3, 5, 7, 8]$, $[2, 3, 5, 7, 8]$. After the third change the possible teams are: $[1, 3, 5, 7, 8]$, $[2, 3, 5, 7, 8]$.