CodersCup Sophomore Round B

QUESTION #1:

Picking Numbers

Given an array of integers, find and print the maximum number of integers you can select from the array such that the absolute difference between any two of the chosen integers is less than or equal to **1.** For example, if your array is a = **[1, 1, 2, 2, 4, 4, 5, 5, 5]**, you can create two subarrays meeting the criterion: **[1, 1, 2, 2]** and **[4, 4, 5, 5, 5].** The maximum length subarray has **5** elements.

Input

The first line contains a single integer n, the size of array a.

The second line contains n space-separated integers a[i].

Output

A single integer denoting the maximum number of integers you can choose from the array such that the absolute difference between any two of the chosen integers is **≤ 1.**

Constraints

- (2 ≤ **n** ≤ 100).
- .(0 ≤ **a[i]** ≤ 100).
- The answer will be **≥ 2**

Input

6
4 6 5 3 3 1

Output

3

QUESTION #2:

Max Min

You will be given a list of integers, **arr,** and a single integer **k**. You must create an array of length k from elements of arr such that its unfairness is minimized. Call that array subarr. Unfairness of any array can be calculated as

Max(subarr)-min(subarr)

Where,
Max denotes the largest number in the array
Min denotes the smallest number in the array

As an example, consider the array [1, 4, 7, 2] with a k of 2. Pick any two elements,test subarr = [4, 7].
Unfairness = max(4,7) – min(4,7) = 7 – 4 = 3.

Testing for all pairs, the solution [1, 2] provides the mimimum unfairness.
Note: Integers in arr may not be unique.

- K is an integer
- Arr: array of integers

Constraints

- ($2 \leq \mathbf{n} \leq 10^5$).
- ($2 \leq \mathbf{k} \leq n$)
- ($0 \leq \mathbf{arr[i]} \leq 10^9$)

Input

The first line contains an integer **n**, the number of elements in array **arr**.

The second line contains an integer **k**.

Each of the next **n** lines contain an integer arr[i] where $0 \leq i \leq n$.

Output

An integer that denotes the minimum possible value of unfairness.

Input:

7
3
10
100
300
200
1000
20
30

Output:

20

Explanation

Here k = 3 selecting the 3 integers 10,20,30 unfairness equals

max(10,20,30) - min(10,20,30) = 30 - 10 = **20**

QUESTION #3:

DFS

Let **T** be a tree on **n** vertices. Consider a graph G_0, initially equal to **T**. You are given a sequence of **q** updates, where the **i**-th update is given as a pair of two distinct integers **ui** and **vi**.

For every **i** from 1 to **q**, we define the graph G_i as follows:

If G_{i-1} contains an edge **{ui,vi},** then remove this edge to form G_i.

Otherwise, add this edge to G_{i-1} to form G_i.

Formally, $G_i := G_{i-1} \triangle \{\{ui,vi\}\}$ where △ denotes the set symmetric difference.

Furthermore, it is guaranteed that **T** is always a subgraph of G_i . In other words, an update never removes an edge of **T**.

Consider a connected graph **H** and run a depth-first search on it. One can see that the tree edges (i.e. the edges leading to a not yet visited vertex at the time of traversal) form a spanning tree of the graph **H**. This spanning tree is not generally fixed for a particular graph — it depends on the starting vertex, and on the order in which the neighbors of each vertex are traversed.

We call vertex **w** good if one can order the neighbors of each vertex in such a way that the depth-first search started from w produces T as the spanning tree. For every **i** from 1 to **q**, find and report the number of good vertices.

Input

The first line contains two integers **n** and **q** ($3 \le n \le 2 \cdot 10^5$, $1 \le q \le 2 \cdot 10^5$) — the number of nodes and the number of updates, respectively.

Output

For each update, print one integer **k** — the number of good vertices **w** after the corresponding update.

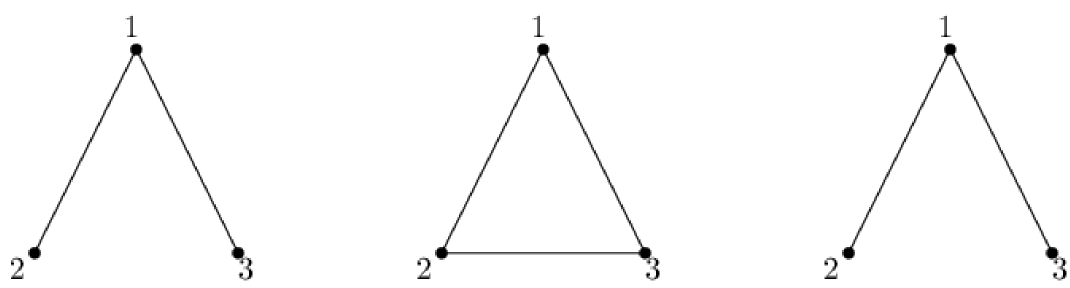
Input:

3 2
1 2
1 3
2 3
3 2

Output:

2
3

Explanation:



$T = G_0$ G_1 G_2

After the first update, **G** contains all three possible edges. The result of a DFS is as follows:

Let the starting vertex be **1**. We have two choices of ordering the neighbors of **1**, either **[2,3] or [3,2].**

If we choose the former, then we reach vertex **2**. Regardless of the ordering of its neighbors, the next visited vertex will be **3**. Thus, the spanning tree generated by this DFS will contain edges **{1,2}** and **{2,3},** which does not equal to **T**.

If we choose the latter, we obtain a spanning tree with edges **{1,3}** and **{2,3}.**

Hence, there is no way of ordering the neighbors of vertices such that the DFS produces **T**, and subsequently **1** is not a good vertex.

Let the starting vertex be **2**. We have two choices of traversing its neighbors. If we visit **3** first, then the spanning tree will consist of edges **{2,3}** and **{1,3}**, which is not equal to **T**. If we, however, visit **1** first, then we can only continue to **3** from here, and the spanning tree will consist of edges **{1,2}** and **{1,3}**, which equals to **T**. Hence, **2** is a good vertex.

The case when we start in the vertex **3** is symmetrical to starting in **2**, and hence **3** is a good vertex.

Therefore, the answer is **2.**

After the second update, the edge between **2** and **3** is removed, and **G=T.** It follows that the spanning tree generated by DFS will be always equal to T independent of the choice of the starting vertex. Thus, the answer is **3.**

QUESTION #4:

FUTURE FAILURE

Alice and Bob are playing a game with a string of characters, with Alice going first. The string consists n characters, each of which is one of the first k letters of the alphabet. On a player's turn, they can either arbitrarily permute the characters in the words, or delete exactly one character in the word (if there is at least one character). In addition, their resulting word cannot have appeared before throughout the entire game. The player unable to make a valid move loses the game.

Given n, k, p, find the number of words with exactly n characters consisting of the first k letters of the alphabet such that Alice will win if both Alice and Bob play optimally. Return this number modulo the prime number p.

Input

The first line of input will contain three integers n, k, p

($1 \le n \le 250\,000$, $1 \le k \le 26$, $10^8 \le p \le 10^9 + 100$, p will be prime).

Output

Print a single integer, the number of winning words for Alice, modulo p.

Example

Input:

4 2 100000007

Output:

14