Palestinian Collegiate Programming Contest 2025

A. Permutate Paths

3 seconds, 256 megabytes

You are given a tree with N nodes, rooted at node 1. A tree is a connected undirected graph that contains exactly $N-1$ edges.

Each node i has an integer value A_i associated with it.

For two nodes u and v, let the **path** between them be the sequence of nodes visited when moving from u to v along the tree edges.

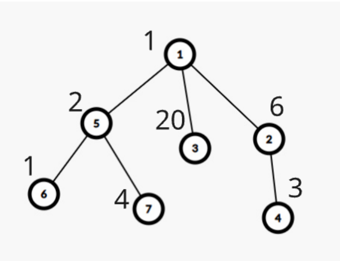
From this path, collect all node values into a multiset $\{A_x\}$.

Define $F(u,v)$ as the largest integer k such that all numbers $1,2,\ldots,k$ appear at least once among the values on the path from u to v.

In other words, $F(u,v)$ represents the size of the longest permutation that can be constructed using the values found on that path.

A **permutation of size** k is a sequence of integers from 1 to k, each appearing exactly once. For example, (1), $(3,2,1)$, and $(4,3,1,2)$ are valid permutations, while $(1,1)$ or $(2,4,5)$ are not.

You are asked to answer Q independent queries. For each query, given two nodes u and v, you must compute $F(u,v)$.



Example tree.

Input

The first line contains two integers N and Q $(2 \le N \le 10^5,\ 1 \le Q \le 10^5)$ — the number of nodes in the tree and the number of queries.

The second line contains N integers $A_1, A_2, \ldots, A_N$ $(1 \le A_i \le 10^9)$, where A_i is the value assigned to node i.

Each of the next $N-1$ lines contains two integers u and v $(1 \le u, v \le N)$ — representing an undirected edge between nodes u and v.

Each of the following Q lines contains two integers u and v $(1 \le u, v \le N)$ — describing one query.

Output

For each query, print a single integer — the value of $F(u,v)$.

input
```
7 5
1 6 20 3 2 1 4
1 2
2 4
1 3
1 5
5 6
5 7
4 7
1 6
1 3
2 4
7 6
```

output
```
4
2
1
0
2
```

In this test case, the tree has 7 nodes and 5 queries.

- The values of nodes are: $A = [1,6,20,3,2,1,4]$
- The edges are: 1–2, 2–4, 1–3, 1–5, 5–6, 5–7

Let's answer each query step by step:

1. Query $(4,7)$: Path $4 \to 2 \to 1 \to 5 \to 7$ Values on the path: $\{3,6,1,2,4\}$ The numbers $1,2,3,4$ appear, but 5 does not. Therefore, $F(4,7)=4$.
2. Query $(1,6)$: Path $1 \to 5 \to 6$ Values: $\{1,2,1\}$ The numbers $1,2$ appear, but 3 does not. So $F(1,6)=2$.
3. Query $(1,3)$: Path $1 \to 3$ Values: $\{1,20\}$ Only the number 1 appears. So $F(1,3)=1$.
4. Query $(2,4)$: Path $2 \to 4$ Values: $\{6,3\}$ The number 1 does not appear, so the result is $F(2,4)=0$.
5. Query $(7,6)$: Path $7 \to 5 \to 6$ Values: $\{4,2,1\}$ The numbers $1,2$ appear, but 3 does not. Therefore, $F(7,6)=2$.

Hence, the answers are: $[4,2,1,0,2]$

B. Min Cost

2 seconds, 1024 megabytes

In **Gaza**, a boy named **Adam** found a torn page from an old book. The page had a long word written without spaces — a mix of letters that once formed many words of hope.

Adam wants to split this word into k parts. Each part should be a continuous piece of the original word.

But the letters are not fully in order — some appear before smaller letters, making the word feel *disordered*. He calls this disorder the **cost** of a part.

The cost of a substring is the number of **inversions** in it. An **inversion** is a pair of letters (i,j) where:

- $i < j$, and
- the letter at position i is alphabetically larger than the letter at position j.

For example:

- In `ba`, there is 1 inversion because `b > a`.
- In `abc`, there are no inversions because the letters are already in order.
- In `cba`, there are 3 inversions: (c,b), (c,a), (b,a).

Adam wants to divide the word into k consecutive parts so that the total cost across all parts is as small as possible.

Can you help him find that minimum total cost?

Input

One line contains a string s of lowercase English letters and an integer k $(1 \le |s| \le 10^5,\ 1 \le k \le \min(20,\ |s|))$

Output

Print a single integer — the minimum total cost after dividing the string into exactly k parts.

input
```
abaaa 2
```

output
```
0
```

input
```
abaaa 1
```
output
```
3
```

In the first example:

- s = abaaa, $k = 2$
- One optimal split is: `ab` | `aaa`
- cost(`ab`) = 0, cost(`aaa`) = 0

Total minimal cost = 0.

In the second example:

- s = abaaa, $k = 1$
- The whole string is one part: `abaaa`
- Inversions:
  - $(b, a_3), (b, a_4), (b, a_5)$ — total 3

Total cost = 3.

C. MTI or MIT

1 second, 256 megabytes

In the bustling world of academia, two renowned universities, MTI and MIT, engage in friendly competition and academic excellence. Yet, a peculiar phenomenon surrounds their names ... MTI and MIT are not just universities but strings of infinite possibilities.

Consider the strings:

1. mti, which repeats as $mtimtimtimti\ldots$ infinitely.
2. mit, echoing as $mitmitmitmit\ldots$ infinitely.

Your task is to transform a given string S into a substring of either $mtimtimtimti\ldots$ or $mitmitmitmit\ldots$ with the least number of operations. Each operation allows you to change any character in S as needed.

However, there's a twist! If the minimum operations to fit S into either symphony are equal, output **"FAKE"** to signify the impossibility of differentiation.

Input

The first line contains an integer N ($1 \le N \le 2 \times 10^5$), the length of the string S.

The second line contains the string S of length N.

Output

Print **"FAKE"** if the minimum number of operations required to make the string S a substring of both patterns is equal, Otherwise, print "$mti\ X$" if the minimum number of operations is smaller for the pattern "mti", or "$mit\ X$" if it is smaller for the pattern "mit", where X is the minimum number of operations.

input
```
9
mtimitabc
```
output
```
FAKE
```

input
```
7
mtmierw
```
output
```
mit 4
```

D. maxA

1 second, 256 megabytes

Reda lives in a large city, C, represented as a grid of integers. Each cell in the grid represents a specific area in the city and contains a value indicating the noise level in that area.

Reda's only wish is to find a **square quiet** place to live. A square area in the city is called **square quiet** if the *median* of the noise levels in that square is **less than or equal** to a given threshold k.

Your task is to help Adel find the **maximum area** A such that **square quiet**.

Input

The first line contains three integers n, m and k ($1 \le n, m \le 500$), ($-10^9 \le k \le 10^9$) — the number of rows and columns in the grid and the noise threshold for a **square quiet** area.

The next n lines each contain m integers, where the i-th number in the j-th line represents the noise level $C_{i,j}$ ($-10^9 \le C_{i,j} \le 10^9$).

Output

Print a single integer — the greatest possible area A such that **square quiet**.

input
```
3 3 10
3 2 7
5 10 -9
-1 7 1
```
output
```
9
```

A square is defined as a subgrid with an equal number of rows and columns. The area of a square with side length s is s^2.

The median of an odd number of elements is the middle value after sorting. For an even number of elements, it's the lower of the two middle values.

For example:

- The median of $[1, 2, 3]$ is 2.
- The median of $[4, 2, 7, 6]$ is 4 (after sorting: $[2, 4, 6, 7]$).

E. Convergent Permutations

2 seconds, 256 megabytes

A permutation of length n is an array of n distinct integers from 1 to n, in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a valid permutation, whereas $[1, 2, 2]$ is not (2 appears twice) and $[1, 3, 4]$ is not.

Define an operation on a permutation q by constructing a new permutation p as follows:

$$p[i] \;=\; q\big[q[i]\big], \qquad 1 \le i \le n.$$

The permutation q is called $convergent$ if, after applying this operation infinitely many times, it eventually becomes the identity permutation (i.e., $p[i] = i$ for all $1 \le i \le n$).

Count the number of $convergent$ permutations of length n.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

Each of the next t lines contains a single integer n ($1 \le n \le 5 \times 10^5$).

Output

For each test case, print one integer — the number of convergent permutations of length n, modulo $10^9 + 7$.

input
4 1 2 3 4
output
1 2 4 16

F. Olive Harvest

1 second, 256 megabytes

In a peaceful village in **Palestine**, Adam owns n olive trees planted in a straight line. Each tree produces a certain number of olives this year.

Adam wants to prepare gift boxes for the children in his neighborhood. Each box must contain olives from a **continuous group of trees**, since he wants every box to represent a connected part of his land.

Adam believes that **even numbers bring peace and balance**, so he will only choose groups of trees whose total number of olives is even.

You need to find the number of **non-empty continuous groups of trees** (subarrays) such that the total number of olives in that group is even. Note that we are counting all possible contiguous groups, not dividing the trees into disjoint parts.

Input

The first line contains an integer t $(1 \le t \le 1000)$ — the number of test cases.

Each test case consists of:

- One line with a single integer n $(1 \le n \le 2 \times 10^5)$ — the number of trees.
- One line with n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$ — the number of olives on each tree.

The sum of all n over all test cases does not exceed 2×10^5.

Output

For each test case, print a single integer — the number of continuous groups of trees with an even total number of olives.

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

In the first test case: All non-empty continuous groups are $[3], [2], [1], [4], [3, 2], [2, 1], [1, 4], [3, 2, 1], [2, 1, 4], [3, 2, 1, 4]$. Among them, the groups with an even total number of olives are $[2], [4], [3, 2, 1], [3, 2, 1, 4]$, so the answer is 4.

In the second test case: The even-sum groups are $[1, 3]$ and $[3, 5]$, giving an answer of 2.

G. The Three Olive Trees

1 second, 256 megabytes

In a quiet village in **Palestine**, a young man named **Adam** spent his days caring for his olive trees. He had planted three trees on a small hill beside his home — each one special to him in its own way. Every morning, Adam watered them and whispered, "Grow strong, my friends. You carry the heart of this land."

When harvest season arrived, Adam counted the olives carefully: the first tree gave him a olives, the second gave him b, and the third gave him c.

He felt happy and proud, but then he thought of his friend — a farmer who lost his trees in a storm. Adam smiled and said, "I will give my friend the tree that gave the most olives, and I will keep the other two for myself."

Now Adam wants to know how many olives he will keep in total.

You are given the number of olives on each of his three trees. Help Adam find out how many olives he will have after giving away the tree with the largest harvest.

Input

The input consists of three lines:

The first line contains one integer a — olives on the first tree.

The second line contains one integer b — olives on the second tree.

The third line contains one integer c — olives on the third tree.

Output

Print a single integer — the total number of olives Adam will keep for himself.

input
5 9 3
output
8

H. Xorful Problem

2 seconds, 256 megabytes

Given a permutation p, an array a, both of size n, and an integer x, find the number of pairs $[l, r]$ such that $1 \le l \le r \le n$ and

$$(\ \mathrm{mex}(l, r)\ \oplus\ a_l\ \oplus\ a_r\ ) > \ x.$$

Here the permutation p is a permutation of the set $\{1, 2, \ldots, n\}$, i.e., each integer from 1 to n appears exactly once in p.

$mex(l, r)$ is defined as the minimum excluded positive integer in the range $[l, r]$ of the permutation p, i.e.,

$$\mathrm{mex}(l, r) = \min\{y \ge 1 \mid y \notin \{p_l, p_{l+1}, \ldots, p_r\}\}.$$

The symbol $\oplus$ denotes the bitwise XOR operation. Bitwise XOR compares numbers in binary and produces a 1 in each bit position where the operands differ and 0 where they are the same. For example,

$$5 \oplus 3 = (101)_2 \oplus (011)_2 = (110)_2 = 6.$$

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^5)$.

The first input line of each test case contains two integers n ($1 \le n \le 2 \times 10^5$) and x $(1 \le x \le 10^9)$ – the length of the array.

The second line contains n distinct integers denoting the permutation p, $1 \le p_i \le n$.

The third line contains n integers a_i $(1 \le a_i \le 10^9)$— the $i-th$ element in the array.

It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each testcase, output a single line – the answer to the problem.

```
input
1
5 1
5 2 1 3 4
10 2 7 8 6
output
10
```

I. Ayman and Elnaggar in the Escape Room

2 seconds, 256 megabytes

$Ayman$ and $ElNaggar$ are currently trapped in an Escape Room. They have successfully solved several puzzles, but they are now stuck on the final puzzle. Unfortunately, time is running out, and they need your help to solve it.

The final puzzle involves a sequence of non-negative integers, represented by an array A of length N. $Ayman$ and $ElNaggar$ need to find the longest contiguous subarray of A that meets the following criteria:

1. All elements in the subarray must be unique.
2. The subarray must have the maximum possible MEX.

The MEX is the smallest non-negative integer not present in the subarray, for example:

- The MEX of [1,2,3] is 0, because 0 does not exist in the array.
- The MEX of [0,1,3] is 2, because 2 does not exist in the array.

Input

The first line contains an integer N $(1 \le N \le 2 \times 10^5)$, the length of the array.

The second line contains N non-negative integers $A_1, A_2, \dots, A_N$ $(0 \le A_i \le N)$, representing the elements of the array A.

Output

Output a single integer, the length of the longest contiguous subarray of the given array that satisfies the given criteria.

```
input
6
0 1 3 2 1 0
output
4
```

```
input
10
1 2 3 4 5 6 7 7 0 1
output
3
```

J. Mundo Game

1 second, 256 megabytes

Mundo is fighting N monsters. Each monster i has a health of H_i. At the i-th second, the i-th monster starts fighting Mundo, joining any previous monsters that are still alive. Every second, Mundo deals d damage to each monster fighting him. A monster dies as soon as its health becomes zero or negative.

At the beginning of each second i:

1. The $i-th$ monster joins the battle.
2. If the number of attacking monsters is $\ge K$ — in other words, if all monsters from index 1 to i that are still alive are $\ge K$ — Mundo loses immediately.
3. If Mundo hasn't lost, he deals d damage to all currently attacking monsters. Dealing d damage reduces an attacking monster's health by d.

Your task is to determine the minimum possible value of d such that Mundo never loses. It is guaranteed that a solution exists.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^3)$. The description of the test cases is provided below.

The first line of each test case contains two integers N and K ($2 \le K \le N \le 2 \times 10^5$).

The second line of each test case contains N integers $H_1, H_2, \dots, H_N$ $(1 \le H_i \le 10^9)$.

It is guaranteed that the sum of N over all test cases does not exceed 2×10^5.

Output

For each test case, output the minimum possible value of d such that Mundo never loses.

```
input
3
5 2
4 7 5 1 4
5 3
4 9 3 7 2
4 3
6 4 8 5
output
7
3
3
```

K. String Ordering

1 second, 256 megabytes

Mazen struggles with string-related problems, and he just encountered a tricky one!

You are given a string s of length n consisting of lowercase English letters. Additionally, you have k strings.

Your task is to determine whether there exists a way to reorder the k strings such that, when checked sequentially, each appears **non-overlapping** in the given string s in that order. You can skip characters between occurrences, but **the substrings cannot overlap** in s.

Input

The first line contains two integers n and k $(1 \le k \le 7, 1 \le n \le 1000)$ — the length of the main string and the number of strings.

The second line contains the main string s of length n.

The next k lines contain the k strings, each with a length of at most 1000.

Output

Print **"YES"** if there exists an order of the k substrings such that each appears in order within the main string; otherwise, print **"NO"**. You can output the answer in any case (for example, **"yEs"**, **"yes"**, **"Yes"**, and **"YES"** will all be accepted as positive responses).

input
6 3 abcdef a cd f
output
YES

input
6 3 abcdef ab bc cd
output
NO

In the first example, we can order the substrings as `a, cd, f`. In this order, each substring appears in s and does not overlap.

In the second example, any ordering of the substrings would require **overlapping** (e.g., `bc` and `cd` both use `c`), so the answer is **"NO"**.

L. gcd ?

1 second, 256 megabytes

$Maryam$ is studying gcd for the first time, and she thought it was pretty cool. So she decided to give you a simple challenge just for fun!

She gives you an array a of n integers and q queries. Each query consists of a single integer k.

For each query, determine whether it is possible to **change at most one** element of the array a (i.e., change zero or one elements)

such that the greatest common divisor of the resulting array **gcd(** $a_1 , a_2 , a_3 , . . . , a_n$**)** becomes exactly equal to k .

Input

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

The second line contains n integers $a_1 , a_2 , a_3 , . . . , a_n$ $(1 \le a_i \le 10^7)$.

The third line contains q $(1 \le q \le 10^6)$ queries , each query contains an integer k $(1 \le k \le 10^9)$.

Output

For each query output **"Yes"** if it is possible, and **"No"** otherwise.

You can output the answer in any case (upper or lower). For example, the strings **"yEs"** , **"yes"**, **"Yes"**, and **"YES"** will be recognized as positive responses.

input
4 4 16 8 32 4 1 2 3 4
output
YES YES NO YES