DDN presents Inter University Programming Contest - IIUC Tech Fest 2025

https://toph.co/c/inter-university-iiuc-tech-fest-2025

Schedule

The contest will run for **5h0m0s**.

Authors

The authors of this contest are amirhozaifa, lelbaba, monna4335, RakibJoy, sakib_muhit, Uosmoy383, and user.Ertugrul28.

Rules

This contest is formatted as per the official rules of ICPC Regional Programming Contests.

You can use C++17 GCC 13.2, C++20 Clang 16.0, C++20 GCC 13.2, C++23 GCC 13.2, C11 GCC 13.2, C17 GCC 13.2, C23 GCC 13.2, Java 1.8, Kotlin 1.9, Kotlin 2.0, PyPy 7.3 (3.10), and Python 3.12 in this contest.

Be fair, be honest. Plagiarism will result in disqualification. Judges' decisions will be final.

Notes

There are 11 challenges in this contest.

Please make sure this booklet contains all of the pages.

If you find any discrepencies between the printed copy and the problem statements in Toph Arena, please rely on the later.

A. Grid Minimization

You are given a grid G of n rows and m columns. Each cell of the grid must be assigned a non-negative integer. Your objective is to minimize the sum of all values placed in the grid. The cell in the i-th row and j-th column is represented by $G_{i,\,j}$

There are q constraints on this grid, represented by one of the two following ways.

- $1\ r\ c_1\ c_2\ f\ (1 \le r \le n, 1 \le c_1 < c_2 \le m, 1 \le f \le 4)$ - representing a constraint on row r from columns c_1 to c_2 of the type f.
- $2\ c\ r_1\ r_2\ f\ (1 \le c \le m, 1 \le r_1 < r_2 \le n, 1 \le f \le 4)$ - representing a constraint on column c from rows r_1 to r_2 of the type f.

Where the values of f implies the following relations -

1. Increasing
2. Non-decreasing
3. Decreasing
4. Non-increasing

For example, the constraint $1\ 2\ 7\ 9\ 1$ represents $r = 2,\ c_1 = 7\ c_2 = 9\ f = 1$, which implies $G_{2,\,7} < G_{2,\,8} < G_{2,\,9}$. And the constraint $2\ 3\ 1\ 4\ 4$ implies $G_{1,\,3} \ge G_{2,\,3} \ge G_{3,\,3} \ge G_{4,\,3}$.

The grid must satisfy all given constraints simultaneously. Find the minimum sum of the values of the grid or report that such a grid cannot exist.

Input

The first line of the input contains a single positive integer: $T\ (1 \le T \le 1000)$ - the number of test cases. . Then T test cases follow.

The first line of each contains three space separated positive integers: $n\ (2 \le n \le 4 \times 10^6)$ - the number of rows in the grid, $m\ (2 \le m \le 4 \times 10^6)$ - the number of columns in the grid, and $q\ (1 \le q \le 10^5)$ - the number of constraints. Then q lines follow describing a constraint.

Each constraint contains five space separated integers. They are of the following two types -

- $1\ r\ c_1\ c_2\ f\ (1 \le r \le n, 1 \le c_1 < c_2 \le m, 1 \le f \le 4)$ - representing a constraint on row r from columns c_1 to c_2 of the type f.
- $2\ c\ r_1\ r_2\ f\ (1 \le c \le m, 1 \le r_1 < r_2 \le n, 1 \le f \le 4)$ - representing a constraint on column c from rows r_1 to r_2 of the type f.

It is guaranteed that sum of n over all test cases does not exceed 4×10^6.

It is guaranteed that sum of m over all test cases does not exceed 4×10^6.

It is guaranteed that sum of q over all test cases does not exceed 10^5.

It is guaranteed that sum of the sizes of the subsegments of each constraint over all test cases does not exceed 10^6.

Output

For each test case, print an integer on a single line - the minimum sum of the values of the grid or -1 if such a grid cannot exist.

Example

Input	Output
3 2 2 4 1 1 1 2 1 1 2 1 2 1 2 1 1 2 1 2 2 1 2 1 2 2 4 1 1 1 2 1 2 2 1 2 1 1 2 1 2 3 2 1 1 2 3 2 3 6 1 1 1 2 2 2 2 1 2 2 1 2 1 2 4 2 1 1 2 4 2 3 1 2 1 1 2 2 3 3	4 -1 9

B. Reversal Machine

In each test case, you are given a string s of length n. You are allowed to reorder the characters of s arbitrarily, forming a new string p.

After choosing p, a new string f is constructed using the following process:

- Initially, f is empty.
- For each position $i = 1, 2, \ldots, n$:
  - Append p_i to the end of f.
  - Reverse the entire string f exactly $\gcd(i, n)$ times.

Your task is to choose a permutation p of s such that the resulting final string f is lexicographically the smallest possible.

Here $\gcd(a, b)$ *denotes the greatest common divisor of integers* a *and* b.

You are given T independent test cases.

Input

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

Each test case consists of two lines:

- The first line contains an integer n ($1 \le n \le 2 * 10^5$).
- The second line contains the string s of length n, consisting of lowercase English letters.

It is guaranteed that the sum of all n over all test cases does not exceed 10^6.

Output

For each test case, output a permutation of the characters of s that produces the lexicographically smallest possible final string f.

Example

Input	Output
3 4	

Input	Output
aabc 2 ca 3 aca	baac ac aca

C. Product Product Fight

You are given an array a of n positive integers and a positive integer X.

Your task is to determine the number of subsequences of array a such that the product of all elements in the subsequence is exactly equal to X. Since the result can be huge, you need to print the result modulo $10^9 + 7$.

A subsequence is a sequence that can be derived from the array by deleting zero or more elements without changing the order of the remaining elements.

The product of an empty subsequence is considered to be 1.

Input

- The first line contains the number of test cases, T.
- The first line of each test case contains two integers n and X — the size of the array and the target product.
- The following line contains n space-separated positive integers — the elements of the array a.

Constraints

- $1 \leq T \leq 5$
- $1 \leq n \leq 10^5$
- $1 \leq a_i \leq 10^6$
- $1 \leq X \leq 10^6$

Output

Print a single integer — the number of subsequences modulo $10^9 + 7$ whose product is exactly X.

Example

Input	Output
2 3 35 5 7 1 5 6 2 3 6 1 1	2 8

Subsequence $(5, 7)$ has product = 35, and subsequence $(5, 7, 1)$ also has product 35, so answer is 2.

D. Merge the Cycles (Easy Version)

You are given a permutation P of size n. For every index i, there is a directed edge $i \to P[i]$. This graph always consists of several cycles.

You may perform the following operation any number of times:

- Pick two distinct indices i, j and swap $P[i]$ and $P[j]$.
  The cost of this operation is $P[i]$ & $P[j]$. Here, & means $bitwise - AND$.

Your task is to make the permutation contain **exactly one cycle using such swaps.** Compute the **minimum total cost** needed to achieve this.

Note that $i \to P[i]$ is also a cycle when $i = P[i]$, because it forms a self-loop of length 1.

A **permutation of size** n is an arrangement of the numbers $1, 2, 3, \ldots, n$ where each number appears **exactly once**.

Input

- The first line contains a single integer n, the size of the permutation.
- The second line contains n integers $P[1], P[2], ..., P[n]$ representing the permutation.

Constraints:

- $1 \leq n \leq 10^5$
- $1 \leq P[i] \leq n$
- P is a valid permutation of size n

Output

- Print a single integer in a line — the minimum total cost needed to make the permutation contain exactly one cycle.

Examples

Input	Output
5 2 1 4 5 3	0

Input	Output
4 4 3 2 1	0

Elements 1 and 2 form a cycle, elements 3, 4, and 5 form another cycle, and swapping 2 with 5 merges them into a single cycle with cost $2 \ \& \ 5 = 0$.

E. Merge the Cycles (Hard Version)

You are given a permutation P of size n. For every index i, there is a directed edge $i \rightarrow P[i]$. This graph always consists of several cycles.

You may perform the following operation any number of times:

- Pick two distinct indices i, j and swap $P[i]$ and $P[j]$.
  The cost of this operation is $P[i] \oplus P[j]$. Here, $\oplus$ means $bitwise - XOR$.

Your task is to make the permutation contain **exactly one cycle using such swaps.** Compute the **minimum total cost** needed to achieve this.

Note that $i \rightarrow P[i]$ is also a cycle when $i = P[i]$, because it forms a self-loop of length 1.

A **permutation of size** n is an arrangement of the numbers $1, 2, 3, \ldots, n$ where each number appears **exactly once**.

Input

- The first line contains a single integer n, the size of the permutation.
- The second line contains n integers $P[1], P[2], ..., P[n]$ representing the permutation.

Constraints:

- $1 \leq n \leq 10^5$
- $1 \leq P[i] \leq n$
- P is a valid permutation of size n

Output

- Print a single integer in a line — the minimum total cost needed to make the permutation contain exactly one cycle.

Examples

Input	Output
5 2 1 4 5 3	1

Input	Output
4 4 3 2 1	2

Elements 1 and 2 form a cycle, elements 3, 4, and 5 form another cycle, and swapping 2 with 3 merges them into a single cycle with cost $2 \oplus 3 = 1$.

F. Greeting of the Cipher King

The IIUC IUPC has finally begun!
At the main gate stands a mysterious figure known as **Zorix, the Cipher King**.

Nobody knows why, but Zorix *never* speaks like a normal human.
Whenever he wants to say something, the sounds that come out of his mouth are always… strange.

People suspect this is because Zorix is always influenced by the *previous* characters of what he says.

A famous incident from last year is still remembered:

- He wanted to greet someone with **“HI ALICE”**
- But what came out was: **“GH ZKHBD”**

Today, Zorix is welcoming contestants into IIUC with this message:

VDKBNLD SN HHTB

As a programmer, you are confident that you can decode it.

Input

There is **no input** for this problem.

Output

Print the decoded greeting in capital letters of the message **“VDKBNLD SN HHTB”** without quote.

G. Number Theoretic Friendship

In a kindergarten, there are n children and each of them has a favorite integer a_i. The children are friends with each other, but some of the pairs are best friends.

As you are the teacher of the class, you tried to observe them and discovered that the children use a mathematical rule to decide whether two of them become best friends:

Two children i and j are best friends if and only if $lcm(a_i, a_j)$ divided by $gcd(a_i, a_j)$ has exactly 3 positive divisors.

Now your task is to determine how many best friend pairs there are among the children.

Input

- The first line contains the number of test cases $t(1 \le t \le 10^5)$.
- The first line of each test case contains an integer $n(1 \le n \le 2 \times 10^5)$.
- The following line contains n integers $a_1, a_2, \ldots, a_n(1 \le a_i \le 2 \times 10^5)$.
- It is guaranteed that the sum of n over all test cases does not exceed 2×10^5.

Output

For each test case, print a single integer — described in the statement.

Example

Input	Output
2 5 12 3 27 3 2 5 8 2 5 6 20	4 2

Explanation of the second test case,

- For pair $(8, 2)$, $lcm(8, 2)/gcd(8, 2)$ is 4, which has 3 divisors $(1, 2, 4)$.
- For pair $(5, 20)$, $lcm(5, 20)/gcd(5, 20)$ is 4, which also has 3 divisors.

There are no other pairs that satisfies the condition, so the answer is 2.

H. Balanced Array Transformation

You are given an array of integers $a_1, a_2, \ldots, a_n$ where each element is either -1 or 1.

An array is called balanced if, for every prefix of the array, the prefix sum lies within the range [-1, 1].

Formally, for all $1 \leq i \leq n$:

-$1 \leq (a_1 + a_2 + \ldots + a_i) \leq 1$.

You are allowed to perform the following operation any number of times:

Operation: Choose a contiguous segment of the array of length at most k ($1 \leq L \leq k$), and flip all the numbers in that segment (i.e. $1 \to -1$ and $-1 \to 1$).

Your task is to determine the minimum number of operations required to make the array balanced.

If it is impossible to make the array balanced, output -1.

Input

- The first line contains a single integer t — the number of test cases.
    - For each test case:
    - The first line contains two integers n and k — the length of the array and the maximum allowed segment length.
    - The second line contains n integers $a_1, a_2, \ldots, a_n$.

Constraints:

- 1 ≤ t ≤ 10^5
- 1 ≤ n ≤ 10^5
- 0 ≤ k ≤ n
- Each a_i is either -1 or 1
- The sum of all n over all test cases does not exceed 10^5.

Output

- For every test case, print a single integer in a new line — the minimum number of operations required to make the array balanced, or -1 if it cannot be done.

Example

Input	Output
2 4 3 1 -1 -1 -1 4 1 1 1 1 -1	1 1

For test case 1, flipping the first 3 elements once makes the array $[-1, 1, 1, -1]$, whose all prefix sums stay within $[-1, 1]$, so the answer is 1.

I. Cheap Thrills

"Come on, come on, turn the radio on

It's birthday night, and it won't be long"

Sia is celebrating her 18th birthday. She has decided to spend the whole night with her best friend and for this, she has invited Alice. Sia is usually very fond of strings. So, Alice has presented her with a string s, consisting of n lowercase English letters. As she gets a new string, she doesn't need dollar bills to have fun tonight. So, she thinks up a new game and asks Alice to do the following operation on the string **exactly m times**. So, she can,

- Select any letter from the string and change that letter to the next alphabet. For example, a becomes b, b becomes c, and so on. For this problem, the next letter of z is a.

For example, if Alice performs an operation for the third letter on $mndmz$, $mndmz$ will become $mnemz$. After that, if she again performs an operation for the fifth on $mnemz$, $mnemz$ will become $mnema$.

Sia wants to find out the lexicographically smallest string after performing exactly m operations on the string. Help Alice to find the lexicographically smallest string after exactly m operation.

A string a is lexicographically smaller than a string b if and only if one of the following holds:

- a is a prefix of b, but $a\ /\ b.=$

  In the first position where a and b differ, the string a has a letter that appears earlier in the alphabet than the corresponding letter in b.

Input

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

Each test case consists of three lines containing three elements: n, the length of the string; s, the string itself; and m, the number of operations. $(1 \leq n \leq 10^5, 0 \leq m \leq 10^9)$.

It is guaranteed that the sum of n overall test cases does not exceed 10^6.

Output

In each query, print the lexicographically smallest string after exactly m operation. Each answer should be printed on a separate line.

Example

Input	Output
4 3 zzz 3 6 abcdef 17 1 z 0 4 mndm 33	aaa abcdew z aads

J. The River of Time

In the mystical land of Xander, the River of Time was a legendary river that flowed through the middle of the country. This river was not just a means of transportation; it was a living entity with its own rules. The river connected all the villages to the market city of Sakkar, where the rare and magical elements grown in Xander were traded. However, the journey along the River of Time was no ordinary journey.

Mr. X, a dedicated farmer from Serenity, was known for his exceptional skill in cultivating these rare elements. He dreamed of selling them in Sakkar, where they fetched the highest prices. To reach Sakkar, he had to rely on the River of Time, which had a peculiar behavior. He had to start his journey from the first hour, and his boat could travel $1km$ per hour.

For the first 12 hours, the river flowed steadily in one front direction, carrying Mr. X closer to Sakkar. However, after the 12th hour, the river's behavior became unpredictable. In any hour h (where $h > 12$), if h could not be expressed as the product of three distinct numbers greater than 1, the river's current would reverse direction, pushing Mr. X backward for the entire hour. Otherwise, the river would continue flowing forward, allowing him to make progress.

So, in short, if the current hour is h:

The river will help move Mr. X forward if there are three different numbers and their product is equal to $h(h = a \times b \times c, where(a, b, c) > 1)$. Otherwise, for the entire current hour, the river pushes Mr. X backward.

To be more specific, assume Mr. X is currently at the dth kilometer. If the condition is not satisfied for the current hour, he moves backward for the entire hour and ends at the $(d-1)th$ kilometer. On the other hand, if the condition is satisfied, he moves forward during the hour and ends at the $(d+1)th$ kilometer.

The river could never push him beyond his starting point, so if it flowed backward, he would simply remain at the source city.

Mr. X faced a critical challenge. He had to reach Sakkar within a given time to sell his elements. For each query, he was given two values:

d : The distance from Serenity to Sakkar (in kilometers).

T: The time remaining (in hours) to reach Sakkar.

Mr. X needed to determine whether he could reach Sakkar in time, considering the unpredictable behavior of the River of Time.

Your Task Given Q queries, for each query with values d and T, determine if Mr. X can reach Sakkar within the given time.

Input

- The first line contains a single integer Q — the number of queries.
- Each of the next Q lines contains two integers d and T:
  - d — the distance from Serenity to Sakkar (in kilometers).
  - T — the total time available (in hours).

Constraints:

- $Q \leq 10^5$
- $0 \leq d \leq 10^5$
- $1 \leq t \leq 10^6$

Output

For each query, print:

- **"YES"** — if Mr. X can reach Sakkar within **T** hours.
- **"NO"** — otherwise.
- Each answer should be printed on a separate line.

Example

Input	Output
4 12 33 45 99 100 10000 100 100	YES NO YES NO

K. The Broken Bridge Forest

Once upon a time, there was a magical forest with n **villages** numbered from 1 to n. These villages were connected by $n-1$ **bridges**, so that from any village you could reach any other village. The king decided to test the strength of the forest by **removing bridges one by one**, in the order they were built.

For each day i ($1 \leq i \leq n-1$):

1. The first i **bridges** are removed.
2. After removing these bridges, the forest breaks into several smaller groups of villages.
3. The king wants to know the **largest distance between any two villages** among all the groups **that are still reachable using the remaining bridges**.

Note: The distance between two villages is measured as the **number of bridges you need to pass** to go from one village to the other.

Input

- The first line contains a single integer T, the number of test cases.
- Each test case starts with a new line followed a single integer n the number of nodes in the tree.
- The next $n-1$ lines each contain two integers u and v, representing an edge between node u and node v.

Constraints:

- $2 \leq T \leq 10^5$
- $2 \leq n \leq 10^5$
- $1 \leq u, v \leq n$
- Sum of n over all test case 10^6

Output

- Each test case output is in a new line.

- For each i from 1 to $n-1$ a space separately print a single integer — the maximum diameter among all trees in the forest after deleting the first i edges in the given order.

Example

Input	Output
2 5 1 2 1 3 2 4 2 5 4 2 4 2 3 4 1	2 2 1 0 1 1 0