ACPC
AFRICA & ARAB
Collegiate Programming
Championship

The International Collegiate Programming Contest
Sponsored by ICPC Foundation

ECPC
EGYPTIAN COLLEGIATE
PROGRAMMING CONTEST

The 2020 Egyptian Collegiate Programming Qualification Contest

(Day 5 Contest Problems)

AAST
November 2020

Problem A. The 1899 Curse !

Input file:	`basement.in`
Output file:	`standard output`
Balloon Color:	`Pink`

After the last Queueforces round, **El-Khadrawy** did so well in the contest, that he thought he will finally become a Candidate Master.

Unfortunately, after the system tests, his rate settled in 1899. What a bad luck, right? El-Khadrawy, However, being a superstitious person, started to doubt whether he might be cursed. Because 1899 was the year he discovered the secret basement in this house.

To decide whether this is a curse or just a coincidence, he came up with the following idea. He will look at his rating changes a_i, $1 \le i \le N$ in all N contests that he participated in, then he will count the number of sub-array $[L, R]$, $1 \le L \le R \le N$ which satisfy the following conditions:

1. The final rating when participating in all of the contests of the sub-array $[L, R]$ is 1899. In other words, $\sum_{i=L}^{R} a_i = 1899$.
2. There is no moment where his rating exceeds 1899. Formally, for each $j : L \le j \le R$, $\sum_{i=L}^{j} a_i < 1900$.

After counting such sub-arrays, he can decide whether he is hunted by some curse or it's just a coincidence and he can still reach candidate master with a little extra hard work.

Your job is to help El-Khadrawy count these sub-arrays.

Input

The first line of the input contains one integer T — the number of test cases. Then T test cases follow.

The first line of the test case contains one integer N, $1 \le N \le 10^5$ — the number of contests. The next line contains N integers that describe the rating changes $-2000 \le a_i \le +2000$.

It is guaranteed that the sum of N does not exceed 3×10^5, $\sum_{i=1}^{T} N_i \le 3 \times 10^5$.

Output

For each test case, print the answer: the number of subarrays that satisfy the two conditions.

Example

basement.in	standard output
2 4 1898 1 1 1897 5 1898 1 0 1 1897	2 3

Problem B. Screen Numbers

Input file: screen.in
Output file: standard output
Balloon Color: Yellow

You are given an array a containing n numbers. In addition, you'll be given m TV screens. Each screen prints the sum of all numbers that has a relation with this screen. Initially, there are no relations between any screen with any number.

There are three types of queries:

- 1 i val: Set the value of the i^{th} number in a to val. Note that when a number in a is changed, all of the values displayed on screens having a relation with this number is changed.
- 2 i j: Add a relation between the i^{th} number in a and the j^{th} screen.
- 3 i: Print the number displayed on the i^{th} screen.

Can you handle q queries?

Input

The first line of the input contains a single integer number T. The number of test cases.

The first line of each test case contains two integer numbers n and m ($1 \leq n, m \leq 2 \times 10^5$). The number of elements in the array a and the number of screens. ~~arr[1000000]~~ vector <int> arr(1e9+7);

The following line contains n integer numbers ($1 \leq a_i \leq 10^9$), where a_i is the i_{th} number in a.

The following line contains a single integer number q ($1 \leq q \leq 2 \times 10^5$). The number of queries you need to handle. q

The following q lines each is one of three types:

1 i val ($1 \leq i \leq n, 1 \leq val \leq 10^9$): Set the value of the i^{th} number in a to val

2 i j ($1 \leq i \leq n, 1 \leq j \leq m$): Add a relation between the i^{th} number in a and the j^{th} screen. It is guaranteed that the relation between a number and a screen will appear at most once in the input.

3 i ($1 \leq i \leq m$): Print the number displayed on the i^{th} screen.

It is guaranteed that the sum of n in all test cases will not exceed 2×10^6.

It is guaranteed that the sum of q in all test cases will not exceed 2×10^6.

Output

For each query of the third type print a single line containing a single integer number. The answer to the query.

input
1
4 3
1 3 7 8
7
2 2 3
3 3
2 1 2
2 1 3
1 2 10
3 3
3 2

output
3
11
1

3 → Print tv
2 → add relation
1 → Change val. in array

Scanned with CamScanner

Example

screen.in	standard output
1 4 3 1 3 7 8 7 2 2 3 3 3 2 1 2 2 1 3 1 2 10 3 3 3 2	3 11 1

prnt
change
prnt
prnt

3 0
4 7
1 7

10
Array 1 ~~3~~ 7 8
error
+V = 3

10
1 ~~3~~ 7 8

prnt: 3 11 1

10
Array 1 ~~3~~ 7 8
+V = 3

Save changed Val

3

+V. relation (iterators)

Problem C. Sweets Cars

Input file:	`car.in`
Output file:	`standard output`
Balloon Color:	`Black`

El-Khadrawy has a problem with eating sweets and he asks you for help.

El-Khadrawy has a tree with n nodes rooted at node number 1. For each node u in this tree, there are three values assigned to it: a_u, b_u, c_u

A sweets car will start at the root (node number 1) with M liters of fuel. Assuming that a sweets car is currently at node u it can do one of the following actions:

- go to one of u's children. The cost of this action is b_v liters of fuel. (where v is the child it chose to go to).
- go to y where y is the k-th smallest node by a_v value in the subtree of u, the cost of this action is c_y liters. (if there is no such node y, it can't do this action). Please note that the node with the k-th smallest a_i in the subtree of node u is determined **without** considering node u. (See the sample below).

Note: if an action costs p liters of fuel and we have currently x liters, then we will have x-p liters of fuel after performing that action. Keep in mind that the car is not allowed to have a negative amount of fuel.

Sweets lovers are known to live in the leaves of the tree, so if we have an infinite number of sweets cars how many leaf nodes can we reach starting from the root?

A **leaf node** is any node in the tree that has at most one edge connected to it.

Input

The first line of the input contains a single integer number T - the number of test cases.

The first line of each test case contains three integer numbers n, k and M $(1 \le n \le 2 \times 10^5)$ $(1 \le k \le 100)$ $(0 \le M \le 10^9)$.

Three lines follow. Each of these lines contains n integers describing the arrays a, b, c in this order $(0 \le a_i, b_i, c_i \le 2 \times 10^5)$ (its guaranteed that all a_i are distinct)

Each of the following $n - 1$ lines contains two integers u, v $(1 \le u, v \le n)$, describing the edges of the tree.

Output

For each test case: Print one integer, the number of leaves we can reach starting from the root.

Example

car.in	standard output
1 6 2 10 2 7 9 5 3 0 3 4 11 5 7 9 9 6 8 3 2 10 1 2 2 3 1 4 4 6 4 5	1

Note

In the sample above, the only leaf node that we can reach is node 5. We can go directly from node 1 to node 5 with 2 liters of fuel, since it is the node that has the k-th smallest a_i value in the subtree of node 1.

Problem D. 2Nodes

Input file:	`path.in`
Output file:	`standard output`
Balloon Color:	`Cyan`

You are given an undirected tree of **even** number of nodes n. Each node has a color, and **no more than two nodes have the same color**. Nodes are numbered from 1 to n. Colors are numbered from 1 to $\frac{n}{2}$.

Your task is to find the length of the longest path in the tree that contains only distinct colors. The length of a path is the number of edges we traverse from source to destination.

Input

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

The first line of each test case contains a single **even** integer n ($2 \le n \le 5 \times 10^4$), the number of nodes in the tree.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le \frac{n}{2}$), where c_i is the color of the i^{th} node.

It is guaranteed that no more than two nodes will have the same color.

Each of the following $n-1$ lines will contain two integers u and v ($1 \le u, v \le n$), representing an edge between the two nodes u and v. It is guaranteed that the given graph is a tree.

The total sum of n over all test cases doesn't exceed 5×10^5.

Output

For each test case, print a single line with the length of the longest path in the tree that contains only distinct colors.

Example

path.in	standard output
2 8 3 1 1 2 2 3 4 4 4 7 8 1 3 5 4 6 3 4 4 2 1 3 2 1 1 2 1	3 0

Note

A tree is a connected graph without cycles.

Problem E. Xander and Sami

Input file:	`sami.in`
Output file:	`standard output`
Balloon Color:	`Orange`

It was the first international competition for Xander. He wanted to get to meet new people from different countries.

In one of the competition days, Xander met Sami. They agreed to play a game. They choose a random string. Then, each one reads it letter by letter. Xander reads a string from left to right. And Sami reads a string from right to left. In the game, they read the string together each one using his direction.

How many letters will be the same? In other words, if each time they both say the next corresponding letter (according to their reading directions), how many times will they say the same letter?

Input

The first line contains T, the number of test cases.

The next T line contains string S ($1 \le |S| \le 10^5$) with lowercase letters. $|S|$ denotes the length of string S.

Output

Print a single integer - The number of equal characters they will say.

Example

sami.in	standard output
2 ttlllleeee acbacba	1 3

acbacba

Problem F. Beverages

Input file:	`mountains.in`
Output file:	`standard output`
Balloon Color:	`White`

El-Khadrawy likes drinking coffee. The group (El-Khadrawy Group or El-G for short) realized that El-Khadrawy is consuming huge amounts of coffee every day. So, they introduced him to the famous beverage (5 in 1).

El-Khadrawy tried (5 in 1) for the first time, it felt like the best drink in the world. El-Khadrawy wanted to return the favor to El-G, so he decided to tell them about one of the hardest problems he has ever thought about. The problem statement goes like this:

We have N mountains, each with a known height of h_i. You are standing on one of the mountains, and you can see in the range of no more than X. In other words, you can see the previous X mountains and the next X mountains from where you stand.

When you are standing on top of mountain i, you will do the following:

1. Write on a piece of paper the number of distinct heights in the range $[max(1, i-x), min(n, i+x)]$.
2. Jump to another mountain j, such that j is in range $[max(1, i-x), min(n, i+x)]$, and $(h_j < h_i)$, then repeat the same steps for mountain j until you can't move.

You have to answer Q queries. In each query, you will be given a number i representing the index of the mountain that you should start the process from. For each query, you should print the maximum sum of numbers that you can write on your paper if you started from this mountain until reaching a mountain from which you can't move anymore.

Unfortunately, El-G didn't know how to solve this problem. Can you help them solve it?

Input

The first line of the input contains one integer T, the number of test cases.

The first line of each test case contains three integers N, X and Q $(1 \le N, Q \le 5 \times 10^5)$ $(0 \le X \le N)$, where N is the number of mountains, Q is the number of queries, and X is the range that you can see in.

The second line of each test case contains N integers $h_1, h_2 ... h_n$ $(1 \le h_i \le 2 \times 10^5)$, where h_i is the height of the $i-th$ mountain.

Each of the next Q lines will contain an integer each q_i, the index of the mountain that you should start from.

Note that the sum of N and Q over all test cases won't exceed 10^6.

Output

For each test case, print Q lines, the $i-th$ line should contain one integer S the answer to the $q_i - th$ query of that test.

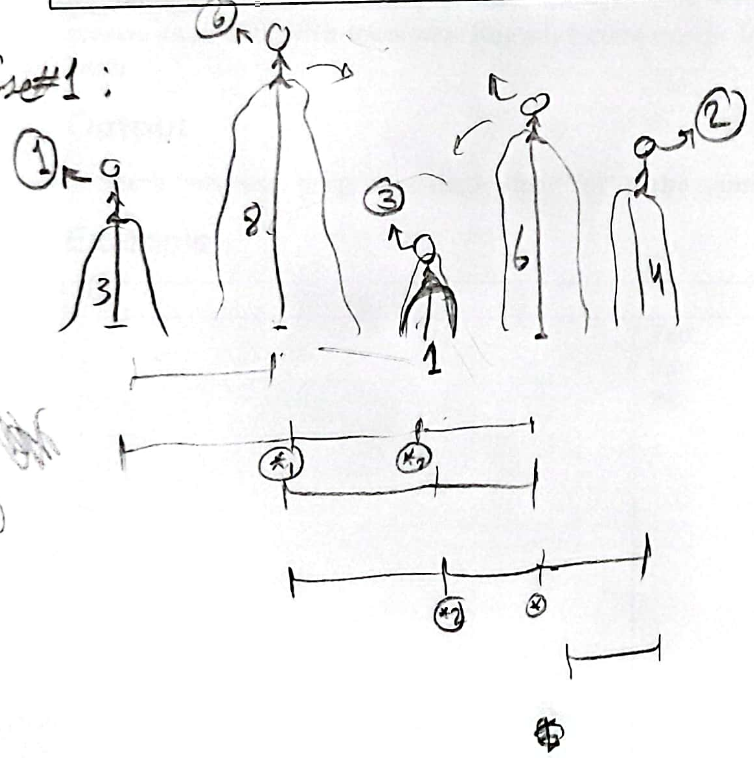
Example

heights

queries

mountains.in	standard output
2 5 1 5 3 8 1 6 4 1 2 3 4 5 5 1 5 5 5 5 5 5 1 2 3 4 5	2 6 3 6 2 1 1 1 1 1

Case#1 :



Problem G. A Valid Naming

Input file:	`naming.in`
Output file:	`standard output`
Balloon Color:	Blue

Preparing a contest is not as easy as you might think. It takes a lot of hard work and dedication. The judges did their job; they prepared everything about the problem set except for the problem names.

A valid naming of the problems is when you can arrange the problems in some order such that the first problem's name begins with the letter A, the second problem's name begins with the letter B ... and so on.

The problem set consists of N problems. Ali will suggest their names, and you will help him decide whether they form a valid naming or not.

Input

The first line of the input consists of an integer T the number of test cases.

In the first line of each test case, you will get an integer N ($1 \le N \le 26$) the number of problems in the problem set, the following N lines will contain a string S in each line such that its length $|S|$ is not greater than 100, with lowercase English letters except for the first letter which is an uppercase English letter.

Output

For each test case, print on a single line "Yes" if the naming is valid, print "No" otherwise.

Example

naming.in	standard output
3	Yes
4	No
Dima	No
Alice	
Carl	
Bob	
3	
David	
Dean	
Acm	
4	
Alice	
Acm	
Bob	
Carl	

set.insert(s[0])
arr ['A']
1
arr[s[0]] = 1
check => 0
map ✓

Note

In the first test case you can rearrange the names as follows :

the first one is *Alice* which starts with A. the second one is *Bob* which starts with B. the third one is *Carl* which starts with C. the fourth one is *Dima* which starts with D.

So, the answer is "Yes".

In the second test case, there is no name starts with B or with C. So, the answer is "No".

Problem H. Cute Numbers

Input file:	`cute.in`
Output file:	`standard output`
Balloon Color:	`Gold`

El-Khadrawy loves math so much. He invented an array called the cute array.

The cute array is an array of cute distinct numbers. Cute numbers are numbers that have the same digits. For example, $(11, 1, 999, -5555, -1, 0)$ are cute, while $(12, 15, 123456)$ are not.

El-Khadrawy was killing time so he invented a problem. He wants to make an array of N numbers a cute array. However, he wanted to make it cute with minimum numbers of operations and asked you for help.

He thought that it won't be fun unless it was under difficult circumstances. You can take any element of the array and either increase it or decrease it by one as many times as you want.

For example, if El-Khadrawy started with $30, 10, 20, 10, 10$, the minimum cost to make it cute would be 9 and the array elements become $33, 9, 22, 11, 8$.

El-Khadrawy wants to know what is the minimum number of operations to make the array cute, and also wants to apply this operation and print a cute array using the initial array and this cost. If there are several cute array with the same minimum cost, print any of them.

Input

The first line contains an integer T denoting the number of test cases.

Each test case starts with a line containing N $(1 \le N \le 100)$ denoting the length of the array.

The next line contains N integers A_i $(-10^9 \le A_i \le 10^9)$.

Output

for each test, you are asked to output the minimum number of operations to make the array cute. On the next line, print the array after applying the operations.

Example

cute.in	standard output
2 5 10 10 20 30 10 3 1 3 5	9 8 9 22 33 11 0 1 3 5

10 10 10 11

1 1 1

11

$\frac{n(n+1)}{2}$ $\frac{5 \times 6}{2}$

Problem I. Just a Game

Input file:	`game.in`
Output file:	`standard output`
Balloon Color:	`Green`

Two little friends: Alice, Bob are playing with an array. The array contains N elements. They want to play a game of two steps on this array:

1. Alice will choose a random pair (A, B), such that $1 \leq A \leq B \leq N$, with equal probability of choosing any pair.
2. Bob will take the sum of all elements between indices A and B, including A and B.

You watched the friends play this game for many times that you started to wonder, if they played it one more time, what would be the expected value of the sum that Bob will get in his turn of the game?

Input

The first line of the input contains one integer T, the number of test cases.
The first line of each test case contains one integer N, $(1 \leq N \leq 10^5)$ the size of the array.
The second line of each test case contains N integers $a_1, a_2 \ldots a_n$ $(-1000 \leq a_i \leq 1000)$, where a_i is the value of the $i-th$ element.
Note that the sum of N over all test cases won't exceed 10^6

Output

For each test case, print one real number E, the expected value of the sum that Bob will get on his turn in the game. Your answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}

Example

game.in	standard output
3 5 1 9 6 3 2 3 -1 3 0 4 1 9 -4 1	11 1.5 3.8

test case: 4
-4 -5 -2 -1

ans
$\boxed{d_{min} = 10}$
$\boxed{-62}$
$\boxed{\text{final} = -6.2}$
-16 -30 -12 -4
Cut
Sum

4 3 2 1
3 2 1
2 1
1
4 6 6 4

Problem J. The Palindrome

Input file:	`pals.in`
Output file:	`standard output`
Balloon Color:	`Silver`

El-Khadrawy has two strings s and t of lowercase Latin letters. He can do the following operation any number of times:

- Choose an integer $1 \le i \le |s|$, where $|s|$ is the length of string s. Then,
- Choose an integer $1 \le j \le |t|$, where $|t|$ is the length of string t. Then,
- Swap the i_{th} letter of s and the j_{th} letter of t.

El-Khadrawy wants to make the string s palindrome. Can you help him checking whether he can or not?

Input

The first line of the input contains a single integer number T. The number of test cases.

The first line of each test case contains two integer numbers n_s and n_t ($1 \le n_s, n_t \le 10^5$). The length of string s and the length of string t, respectively.

The second line of each test case contains a string s. It is guaranteed that string s contains n_s lowercase Latin letters only.

The third line of each test case contains a string t. It is guaranteed that string t contains n_t lowercase Latin letters only.

It is guaranteed that the sum of n_s in all test cases is less than or equal to 3×10^5, and the sum of n_t in all test cases is less than or equal to 3×10^5

Output

For each test case print a single line containing "YES" if El-Khadrawy can make the string palindrome. Otherwise, print "NO".

Example

pals.in	standard output
2 3 4 abc bbcc 4 4 abgd aefc	YES NO

x y z x y x

t:→ b c
2 2

s: a b c
1 1 1

لو فيه odd
ندور الاول على الحرف
ونطلع واحد
لو مالقيناهوش ← هنشوف اي حاجة تانية وناخد (-2) منه

a b c d

دايماً %2