Problem A. Loyalty

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Red`

A group of n friends is going on a trip together. Each friend has a loyalty value representing the maximum group size that the friend is comfortable with.

For example, if a friend has a **loyalty value** of k, it means that he is willing to travel with any group of size up to k (including himself). This means that he is comfortable with being in a group with just themselves, or with a group of friends smaller than or equal to k.

The friends want to form groups for various activities during the trip. Each group must have at least one member, and the size of the group must not exceed the loyalty value of any of its members. A friend can only be a member of one group at a time.

Write a program to determine the **minimum** number of groups that can be formed.

Input

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

The first line of each test case contains one integer n ($1 \leq n \leq 10^5$) —the number of friends.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) —the loyalty value k of each friend.

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

Output

For each test case, print a single integer — the minimum number of groups that can be formed.

Example

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

Problem B. Play with Koko

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Black`

Koko loves cats and has a lot of them. Recently, He found it difficult to name all of them.

He found a way to solve this by giving each cat an **ID** ($2 \leq$ ID), Which is the product of its parent's IDs.

Now, Koko wants to play with you, he will give you two cats' IDs, and you have to **determine** if they have some common ancestor.

Input

The input begins with a positive integer ($1 \leq t \leq 100$), the number of test cases.

After that, follows t lines, each with two integers a_i , b_i identifying two cats' IDs. ($2 \leq a_i, b_i \leq 1000$)

Output

For each test case, print "YES"if the cats a_i and b_i share a common ancestor and "NO"otherwise.

Example

standard input	standard output
2 2 4 3 5	YES NO

Problem C. Parking Area

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Light Blue`

It's your first day working and your company decided to build a parking area for employees' cars that has a capacity of n cars.

They wanted to build a robot, so during the car entrance to the parking area, the robot would give the employee a ticket for the nearest available slot to park his car in.

Note that all slots are on a straight line and have numbers from 1 to n.

So your first task in this company is to write a program to print the nearest available slot so an employee can park in.

Input

The first line consists of an integer n $(1 \leq n \leq 10^5)$ the capacity of the parking area.

The second line consists of an integer q $(n \leq q \leq 2 * 10^5)$ the number of queries.

The following q lines given a string which is $E\ CAR_{ID}$ or $L\ CAR_{ID}$.

$E\ CAR_{ID}$ means a car with CAR_{ID} ID enters the parking area and needs a slot number from the robot.

$L\ CAR_{ID}$ means a car with CAR_{ID} ID is leaving the parking area and the slot is free now.

CAR_{ID} $(1 \leq |CAR_{ID}| \leq 10)$ consisting of uppercase latin letters.

Output

for each $E\ CAR_{ID}$ print "Good morning sir, please proceed to slot number NEAREST SLOTNUMBER".

if there is no empty slot for an employee print "Sorry sir, no available slots"

note that if there is no empty slot the employee will leave.

Example

standard input	standard output

`3`
`8`
`E QFSQS`
`E YETRDH`
`E GHDFD`
`L QFSQS`
`L GHDFD`
`E DTYJT`
`E FZBXDF`
`E RTHDFF`
	`Good morning sir, please proceed to slot number 1`
	`Good morning sir, please proceed to slot number 2`
	`Good morning sir, please proceed to slot number 3`
	`Good morning sir, please proceed to slot number 1`
	`Good morning sir, please proceed to slot number 3`
	`Sorry sir, no available slots`

Problem D. median check

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

Once upon a time, a young boy named Omar loved to go shopping. He would often go to the local market and browse through the different products available. One day, he was given a problem by his math teacher that involved buying products within a given budget and finding the number of ways to select prices that could be the median of the resulting array.

The problem required Omar to buy n products, where each product had a minimum price and a maximum price. He was given an array a of length n representing the minimum prices, and an array b of length n representing the maximum prices. He had up to m dollars to spend and could buy each product for any price within its given range. After buying all the products, he would sort the prices in an array p, and the median of this array should be equal to a given value in the array k. Omar had to find the number of elements in array k that can be the median of the resulting array p.

Input

The first line contains two integers n,m $(1 \le n \le 10^4)$ n the size of the array a,$(1 \le m \le 10^{15})$, n is always odd.

The second line contains array a $(1 \le a_i \le 10^5)$ ai is the number with index i in array a.

The third line contains array b $(1 \le b_i \le 10^5)$ bi is the number with index i in array b

The fourth line contains the number k $(1 \le k \le 10^4)$.

The fifth line contains k numbers $c1, c2..., ck$ $(1 \le c_i \le 10^9)$.

Output

How many numbers in array k can be the median

Examples

standard input	standard output
3 20 1 2 3 4 5 6 3 1 2 1000000000	1
3 20 1 2 3 4 5 6 3 1 2 3	2
3 5 1 2 3 4 5 6 2 1 100000	0

Note

please note that the median is the element in the middle of the array after sorting it.

Problem E. Tug of War

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark Green`

Tug of War is a classic game played between two teams, where each team tries to pull the rope towards their side. The team that pulls the rope toward their side wins the game.

there are n people, and you are given two arrangements of **the same** people, the first arrangement is called `Team A` and the other is called `Team B`.

you know that the optimal team arrangement involves placing strong members as anchors. The front position should be reserved for the strongest person, while the $2nd$ strongest should be positioned at the back. and the $3rd$ strongest next to the strongest and so on, This way, those with less strength will be situated in the middle.

for example, if $n = 5$ and the strengths are $[1, 2, 3, 4, 5]$ the best arrangement is $[4, 2, 1, 3, 5]$.

let's denote the strength of a team as the **count** of members who are standing in the correct spots according to the rule. your task is to compare the two arrangements as if they are competing against each other, the team with greater team strength wins.

Input

The first line contains the number of members in each team n $(2 \le n \le 10^5)$

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 2 \cdot 10^5)$ — the arrangement of Team A.

The third line contains n integers $b_1, b_2, ..., b_n$ $(1 \le b_i \le 2 \cdot 10^5)$ — the arrangement of Team B.

Output

If one team has a greater team strength than the other, print the name of the better team: `Team A` for Team A and `Team B` for Team B. In case of a tie, print `Tie`.

Examples

standard input	standard output

`5`
`2 4 1 3 5`
`1 4 2 3 5`
	`Team A`

`3`
`2 3 1`
`3 1 2`
	`Tie`

`3`
`1 2 3`
`2 1 3`
	`Team B`

Note

in the first example, team A has 3 members standing in correct spots, those with strength $\{5, 3, 1\}$, while team B has only 2 members standing correctly $\{5, 3\}$.

in the second example, each team has only 1 member standing correctly so it's a Tie.

Problem F. Phoney Company

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Yellow`

A company has n employees, who form a tree hierarchy where each employee has a direct boss, except for the general director.

Employee A is considered an indirect boss for Employee B if A is the direct boss for an employee who is the direct or indirect boss for B.

For some reason, if an employee X wants to hold a meeting, he can meet with a group of employees, provided that they fulfill the following conditions:

- X is a direct or indirect boss for each employee of them.
- X can't be among them.
- There is no employee Y who can hold a meeting with them, and X is a direct or indirect boss for him.

Find for each employee from 1 to n the number of possible combinations of employees which whom he can hold a meeting with them.

Input

The first line of the input will contain a single integer T $(1 \leq T \leq 1000)$ — The number of test cases.

The first line of each test case will contain a single integer n $(1 \leq n \leq 2 \times 10^5)$ — the number of employees.

The last line of each test case will contain n integers $p_1, p_2, p_3, ..., p_n$ where p_i is the direct boss for the i_{th} employee or 0 if the i_{th} employee is the general director.

It's guaranteed that the sum of n overall test cases will not exceed 10^6.

Output

For each test case, print in a single line n integers, the i_{th} of them is the number of possible combinations of employees whom the i_{th} employee can hold a meeting with them modulo $10^9 + 7$.

Example

standard input	standard output
4 1 0 3 2 0 1 3 3 3 0 7 0 1 2 2 4 3 4	0 1 2 0 0 0 3 32 27 1 3 0 0 0

Note

No employee can be a direct or indirect boss for himself, so in the first example the second employee can hold a meeting with $[1, 3]$, and $[1]$, but he can't hold a meeting with $[3]$

Problem G. Awesome Segments

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Gold`

Given an array a of n integers, let's call a segment s of m integers awesome if at least one of the following conditions is satisfied:

- m is equal to 1.
- All elements in s are in non-decresing order, i.e. $s_1 \le s_2 \le s_3 \le ... \le s_m$.
- All elements in s are in non-increasing order, i.e. $s_1 \ge s_2 \ge s_3 \ge ... \ge s_m$.
- There is some index i such that $(1 < i < m)$ and $s_1 \le s_2 \le ... \le s_i \ge s_{i+1} \ge s_{i+2} \ge ... \ge sm$.

Find the number of ways to split the array a into **non-empty** contiguous awesome segments modulo $10^9 + 7$.

Input

The first line of the input will contain a single integer T $(1 \le T \le 1000)$ — The number of test cases.

The first line of each test case will contain a single integer n $(1 \le n \le 10^6)$.

The last line of each test case will contain n integers $a_1, a_2, a_3, ..., a_n$ $(-10^9 \le a_i \le 10^9)$.

It's guaranteed that the sum of n overall test cases will not exceed 2×10^6.

Output

For each test case, print a single line containing the number of ways to split a into non-empty contiguous awesome segments modulo $10^9 + 7$.

Example

standard input	standard output
3 2 1 2 3 1 2 3 4 1 3 4 2	2 4 8

Problem H. The Music Game

Input file: `standard input`
Output file: `standard output`
Balloon Color: `White`

Malak and her n friends are waiting for the lecture to begin, her friends start playing "The Music Game"till the lecturer arrives.

This game's rules are:

- each player has a **unique** power p_i.
- The music player will stop the music $n - 1$ time.
- All the players should listen carefully to the lyrics
- when the music stops the last letter will decide who will be disqualified!
- the disqualification process is taking this letter and converting it to its alphabetical order **(from 1 to 26)** and see which player has the maximum absolute difference between his power and the letter order and disqualify him.
- If there are two players who have the same absolute difference, you have to disqualify the player with the smaller power and the maximum absolute difference.
- the disqualified player is not going to participate in the next rounds

Malak decided to be the music player and she wanted to know who is the winner.

Input

In the first line given an integer n $(1 \leq n \leq 10^5)$ the number of Malak friends.

in the second line given an n integers $(1 \leq p_i \leq 10^9)$ the power of each player.

given in the next $n - 1$ lines an string s consisting of lowercase and uppercase Latin letters.

Output

you have to output the power p_i of the winner.

Example

standard input	standard output
7 3 12 5 25 1 17 23 IJustHeardYouFoundTheOneYouHaveBeenLooking YouHaveBeenLookingFor IWishIWouldHaveKnownThatWasNotMe CauseEvenAfterAllThisTimeIStillWonder WhyICanNotMoveOn JustTheWayYouDidSoEasily	17

Problem I. Change permutation

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Silver`

Charlie loved to solve challenging math problems. One day, they were given a problem that involved finding the number of common subsequences between n permutations of length k, where the length of the longest common subsequence was x.

Alice, Bob, and Charlie were excited to solve this problem and started working on it immediately. They first tried to find the longest common subsequence between the given permutations, which would help them find the number of common subsequences of length x and they will add one rule to make the problem very hard.

The rule is: the start of every subsequence is unique.

They realized that finding the longest common subsequence between n permutations of length k was a difficult problem in itself Can you help them one we add this rule ?

Input

The first line contains n,k $(1 \leq n \leq 500)$ n the number of permutations $(1 \leq k \leq 500)$ the length of permutions.

then we have n lines every line contain a permutation of length k.).

Output

the length of the longest common subsequence and how many common sub sequence there length $=$ the length of the longest commonn subsequence Where the start of every subsequence is unique.

Examples

standard input	standard output
2 3 1 2 3 3 1 2	2 1
2 3 1 2 3 1 2 3	3 1
2 2 1 2 2 1	1 2

Note

We call the start unique if the index which starts subsequence is unique

Problem J. The Whole Nine Chars

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Orange`

Bondok is a cat which finds itself in a bad situation,

he is standing on a rock in the middle of a lake and wants to cross the lake over to the other bank, but there is a problem he can't touch the water he only can jump on the rocks in the lake, rocks are represented as lowercase English letters, rock (a) denotes the closest rock to the first lakeside which *Bondok* came from, and rock(z) represents the other lakeside, that it wants to reach.

Bondok can jump up to 9 rocks per jump,

One jump from any letter means that he is currently standing on the rock with the letter after it in alphabetical order,

$ForExample$:

if *Bondok* is currently on the rock (a) and he wants to jump two rocks It will end up on the rock with the letter (c).

a to b first jump and b to c second jump

You are given the current Rock *Bondok* is standing on, help the poor cat to reach the other lakeside as minimum jumps as possible.

Input

consists of only one character C which denotes the current rock *Bondok* is standing on.

Output

The minimum amount of jumps *bondok* has to take so he can reach the other lakeside

from the rock he is currently on.

Examples

standard input	standard output
`a`	`3`
`x`	`1`

Problem K. Bouncing Ball

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark Blue`

Being burned out from training so hard for *ICPC*, Magdy wants to have some fun to restore his energy. He asked Halaby if he knew any game that he could play. So Halaby gave him a simple one.

There will be n blocks numbered from 1 to n and their lengths are 1 unit. You will get i points if the ball lands on block i. The game goes like that:

- You pick a starting block i to set the ball at it.
- The ball will start bouncing with power p.
- If the ball is at block i and bounced with power k it will land on block $i + k$.
- After the ball bounces with power k its bounce power will decrease by one (e.g. the ball will now bounce with power $k - 1$).
- The ball keeps bouncing as long as its bouncing power is positive and it didn't bounce out of the n blocks.

If Magdy can choose the ball's initial position, He wants to know the maximum number of points he can get, and because the number may be very big print it modulo $10^9 + 7$.

Input

The first line of the input data contains a single integer t $(1 \le t \le 20)$ — the number of test cases.

The first and only line of each test case contains two integers n and p $(1 \le n, p \le 10^{18})$ — the number of blocks and the initial power of the ball, respectively.

Output

Output t integers, each of which is the answer to the corresponding test case. As an answer, output the maximum number of points Magdy can get modulo $10^9 + 7$.

Example

standard input	standard output
2 10 5 6 1	17 11

Note

In the first test case of the example, to get the maximum number of points you can start at block number 1 which will make the ball bounce on blocks $1, 6, 10$ collecting a total of 17 points.

Problem L. Hajar's Game

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Bronze`

Hajar has devised an exciting game for you. She presents an array of numbers with n elements and challenges you to transform each number into a value equal to or greater than 10^6. Here are the rules of the game:

1. In each step of the game, select an index i. You have the option to choose a number x from a group of k numbers located directly before or after the i_{th} number. More specifically, you can select x from the elements at positions $i-1, i-2, ..., i-k$ or $i+1, i+2, ..., i+k$, provided that those positions exist in the array.
2. To proceed, you will add the chosen number x to the i_{th} number. For example, if the current number is a_i, you will update it as $a_i = a_i + x$.
3. It's important to note that these modifications do not affect the original array.
4. The goal is to find the most efficient strategy that requires the fewest possible steps to modify each number in the array and achieve a value of at least 10^6.

For example, if we have an array $A = [1, 2, 3]$, $n = 3, k = 2$. We can use at most 2 elements on the left and at most 2 elements on the right for the conversion of each element. So, for the first element, for example, the best way for conversion is to add the number 3 to it until reaching 10^6 or bigger, and so on for other elements.

Could you beat Hajar and find an optimal answer to this game?

Input

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

The first line of each test case contains two integers n, k ($1 \le k < n \le 10^5$) — the length of the array and number of integers you can use on the left and right for each element in conversion.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^6$).

It is guaranteed that the sum of n overall test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print a single integer — the total number of operations you need to convert all the elements of the array.

Example

standard input	standard output

`1`
`3 2`
`1 2 3`
	`1166665`

Note

conversion of each element is not considered before finishing conversion of all elements of the array.

Problem M. Meow-san XORing

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Pink`

In the village of Numerica, there are N houses with unique door numbers in array A. Alice wants to find how many pairs of houses (i, j) exist, where $i < j$ and the XOR of A_i and A_j is an even number. Can you help Alice solve this puzzle?

Input

The first line of input contains an integer N $1 \leq N \leq 10^5$, the size of the array.

The second line contains N space-separated integers, the elements of the array $1 \leq A_i \leq 10^5$.

Output

Print a single integer, the number of pairs (i, j) such that $i < j$ and A_i XOR A_j is an even number.

Example

standard input	standard output
5 1 2 3 4 5	4