Statements

Online Selection Contest
BACS Regional Programming Contest, 2018
And
BACS High School Programming Contest, 2018

Organized by

B.A.C.S
Bangladesh Advanced Computing Society

Sponsored by

CodeMarshal

www.codemarshal.com

SSL WIRELESS
www.sslwireless.com

This page is intentionally left (almost) blank.

Problem A - A Criminal

After committing a serious crime (Killing an innocent mosquito who never bites any animal) you are arrested by ACM's Defense Team. You are the **X**th (**1 based index**) criminal in the **ordered list** of **N** criminals. Defense Team are willing to punish criminals for their horrible crime. First day Defense Team take **first Y people** from criminal list and arrange them in a circle (Just like the picture bellow remember that given picture is valid when **Y** is equal to 8)

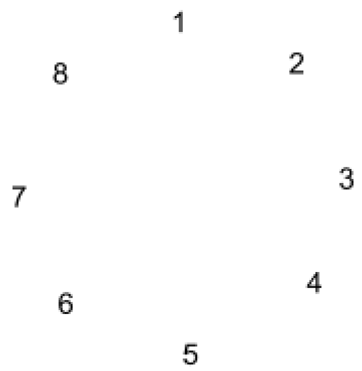


Fig: Circle That Defense Team form when **Y**=8

After forming circle they start from criminal one and simulate **clockwise**. They skip one person and punish next person from the active list (active list: person who have not been punished yet). When only one person is left in the active list they just do not punish him as he is a special person. The order of punishment when **Y** = 8 is [2, 4, 6, 8, 3 , 7, 5] 1 will not be punished in that case.

You know that Defense Team will not take **less than L** from criminal list while making circle and you also know that taking any(in range **[L,N]**) number of criminal form list is equally probable.

You are good at programming so you want to calculate the probability that you will not punished on the first day.

Input Specification

First line of the input will contain an integer **T** denoting number of test cases.
Next T lines will contain Three Integer **X , L , N**.

X: Your position on Criminal List

L: Minimum number of criminal that Defense Team Can take.
N: Total number of criminal on that list.

Constraints

$1 \le T \le 10500$
$1 \le X \le N \le 10^{15}$
$1 \le L \le N$

Output Specification

For every case print case number (See sample output for exact format) and print the answer (Probability that you will not punished on first day) as **p/q** format where **p** and **q** are coprime If the answer is **0**, then you should print **0/1**.

Sample Input	Sample Output
3 1 1 1 2 3 10 4 1 7	Case 1: 1/1 Case 2: 0/1 Case 3: 3/7

Explanation

First Case

There is only one option which is taking one person from list. and you will not be punished because remaining person in circle is 1 already(which is actually you). So ans is 1/1

Second Case

There are 8 equally probable option for Y (3,4,5,6,7,8,9,10). And for each one of those you are going to be punished. So ans is 0/1
Third Case:
This are 7 equally probable options of Y (1,2,3,4,5,6,7). When Y is 1 or 2 or 3 you will not be punished (Actually you are not even selected to be a part of the circle because your id is 4). So ans is 3/7.

Problem B - A Leap of Faith

Rio and Tokyo are two best friends on the run. They have robbed a bank in Madrid and are now hiding from the Interpol. At this moment Rio is in Moscow and Tokyo is in Denver. They cannot communicate with each other. But before they got separated, they had decided that they would meet in Osaka. But they could not decide the date when they should meet in Osaka. Tokyo has a list of flights from Moscow to Osaka and also a list of flights from Denver to Osaka. For every flight, the arrival time is given as a range. Formally, there are n flights departing from Moscow. For the i^{th} flight, two numbers x_i and **yi** are given. It means that the i^{th} flight from Moscow will need at least x_i hours from now and at most y_i hours from now to reach Osaka. Also there are m flights departing from Denver. For the i-th flight, two numbers x_i and y_i are given. It means that the i^{th} flight from Denver will need at least x_i hours from now and at most y_i hours from now to reach Osaka. Tokyo thinks that Interpol might already know that they are meeting in Osaka. So she is sure that if Rio takes the i^{th} flight from the list of flights from Moscow and she takes the j^{th} flight from the list of flights from Denver and if the ranges of the i^{th} and the j^{th} flights have a common part, then Interpol would make these flights reach Osaka at the same time to catch them both together. Tokyo is willing to take that risk because she believes that they can together evade the cops and get reunited. So now she needs to know the number of pair of flights where the first flight of the pair is from Moscow to Osaka and the second flight of the pair is from Denver to Osaka and the ranges of the arrival time of the pair have an intersection point.
Oh, and one more important information. There will be **Q** updates before Tokyo takes any flight. In the i^{th} update, three numbers source, x_i and y_i will be given. If value of source is 1 then it will mean that a new flight from Moscow to Osaka is added to the schedule whose arrival time is in between x_i and y_i. If the value of source is 2 then it will mean that a new flight from Denver to Osaka is added to the schedule whose arrival time is in between x_i and y_i. Tokyo will need to know the number of pairs of flights with intersecting range of the arrival times after every update.

Input Specification

In the first line, a number **T** will be given which indicates the number of test cases. For each test case, in the first line a number **n** will be given which indicates the number of available flights from Moscow to Osaka initially. Then **n** lines will follow, each for one flight. In i^{th} of these lines, two numbers x_i and y_i will be given. It means that the i^{th} flight has an arrival time in between x_i and y_i. After these **n** lines are given, the next line will contain an integer, **m** which indicates the number of available flights from Denver to Osaka initially. Then **m** lines will follow in exactly the same format as the flights above. Then a line will contain an integer **Q** which indicates the number of updates. Then **Q** lines will follow for the **Q** flights. In **i**th line, three integers source, **x**$_i$ and **y**$_i$ will be given. The meaning of these three integers are given in the problem description.

Constraints

$1 \le T \le 100$

$0 \le n, m \le 100000$

$0 \le Q \le 100000$

$1 \le xi \le yi \le 1000000000$

$\Sigma (n + m + q) \le 1.5 * 1000000$

Output Specification

For each case, in the first line print the case number and the number of pairs of flights with an intersecting arrival time range for the initial list of flights. Then print **Q** lines. The **i-th** line should contain the answer after the **i**th update.

Input	Output
2 2 1 10 4 12 3 5 13 1 10 90 100 0 1 1 10 2 5 13 90 100 2 1 4 12 2 1 10	Case 1: 4 Case 2: 1 2 4

Problem C - BACS, Scoundrel Shopkeeper and Contiguous Sequence

A dishonest grocery shopkeeper always tries to cheat on his customers. Very often the customers pick multiple pieces of stuff and ask him the total price only. He takes this as a chance and tells more than the total sum. Innocent customers believe him and pay the amount he tells. But you are not going to it, right?

Again, do you know what is a contiguous sequence? Sequence **S** which has at least two elements, will be called a contiguous sequence if the following condition is satisfied:
$S_{i+1} - S_i = 1$ for each i $(0 \le i < |S|-1)$ and $|S|$ means the length of sequence **S**.

Now you are given an integer **M** and an array **A** with **N** elements. **M** denotes the amount of money the shopkeeper asked and the array denotes the price of all the things you have bought. You have to tell if the shopkeeper lied or not.
Again, you have to say if we can find at least one pair **(i, j)** such that $A_i, A_{i+1}, \dots, A_j$ forms a contiguous sequence where $0 \le i < j < N$.

Input Specification

The first line of the input will contain an integer **T** denoting the number of test cases.
Each test case will consist of two lines. The first line of a test case will contain two integers **M** and **N**. Next line will contain **N** integers separated by spaces which actually denotes the array **A**.

Output Specification

First you have to print “Thank You BACS!!!”, without the quotes. Don't worry and don't ask. It's just a convention.

Then if the shopkeeper lied you have to print “Bojjat dokandar!!”, without the quotes. And if the shopkeeper told the truth you have to print “Thik ache.”, without the quotes.
Again, if array **A** contains at least one contiguous sequence, then your answer is “Yes”, otherwise “No” (without quotes).

For each test case, you should print one line in the format:
Case W: X Y Z

Here **W** denotes the test case number, **X** denotes the mandatory thanking, **Y** denotes the answer which will be either “Bojjat dokandar!!” or “Thik ache.” and **Z** denotes the answer which will be either “Yes” or “No”. Check Sample Output for clarification.

Constraints

1 ≤ T ≤ 100
2 ≤ N ≤ 100
1 ≤ M ≤ 10000000
1 ≤ A[i] ≤ 100000

Input	Output
4 13 5 1 3 2 3 4 20 5 1 3 2 3 4 16 5 1 6 3 2 4 100 5 1 6 3 2 4	Case 1: Thank You BACS!!! Thik ache. Yes Case 2: Thank You BACS!!! Bojjat dokandar!! Yes Case 3: Thank You BACS!!! Thik ache. No Case 4: Thank You BACS!!! Bojjat dokandar!! No

Problem D - Beauty and The Tree

You are given a rooted tree with **N** nodes numbered from **1** to **N**. Node **1** is the root. Each node **i** contains a beauty value b_i.
The beauty of the **d** - th level of the tree is the sum of the beauty values of all the nodes with depth **d** and the beauty of the tree is the largest beauty value of its levels.

For each node **v (2 ≤ v ≤ N)** , you have to output the beauty of the tree assuming that the subtree rooted at node **v**(including node **v**) does not exist in the tree.

Input Specification

The first line of input will contain a number **T**, the number of test cases. Each of the test cases contain 3 lines. The first line contains **N**, the number of nodes in the tree. The second line contains **N** integers $b_1, b_2, \dots \dots, b_n$, the beauty values of the nodes. The third line contains **N-1** integers $p_2, p_3, \dots \dots, p_n$ **(1<=pi<i)** where node p_i is the parent of node **i** in the tree.

Constraints

$1 \le T \le 30$
$1 \le N \le 100000$
$0 \le b_i \le 10000$

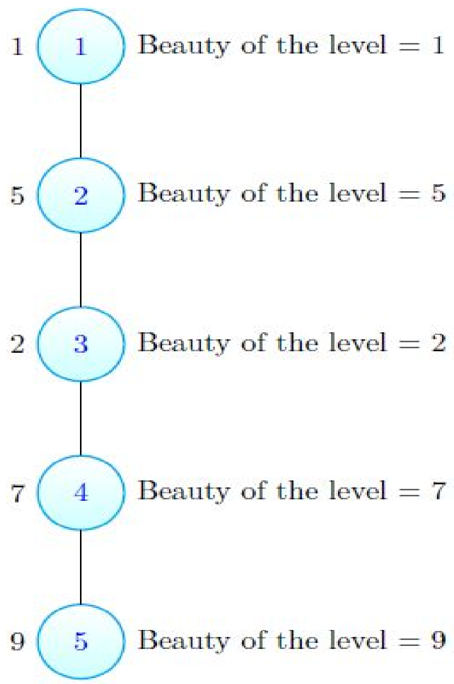
Output Specification

For each test case output **N-1** lines. The i^{th} line should contain the beauty value of the tree assuming that the subtree of node **i+1**(including node **i+1**) does not exist in the tree.

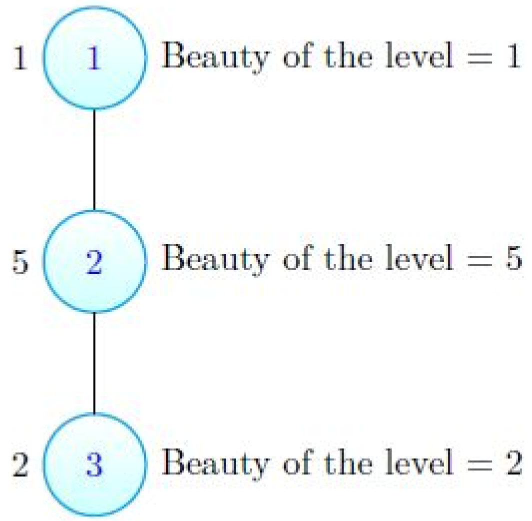
Input	Output
2 5 1 5 2 7 9 1 2 3 4 6 3 1 4 7 6 5 1 1 2 2 3	1 5 5 7 5 13 11 12 13

Explanation

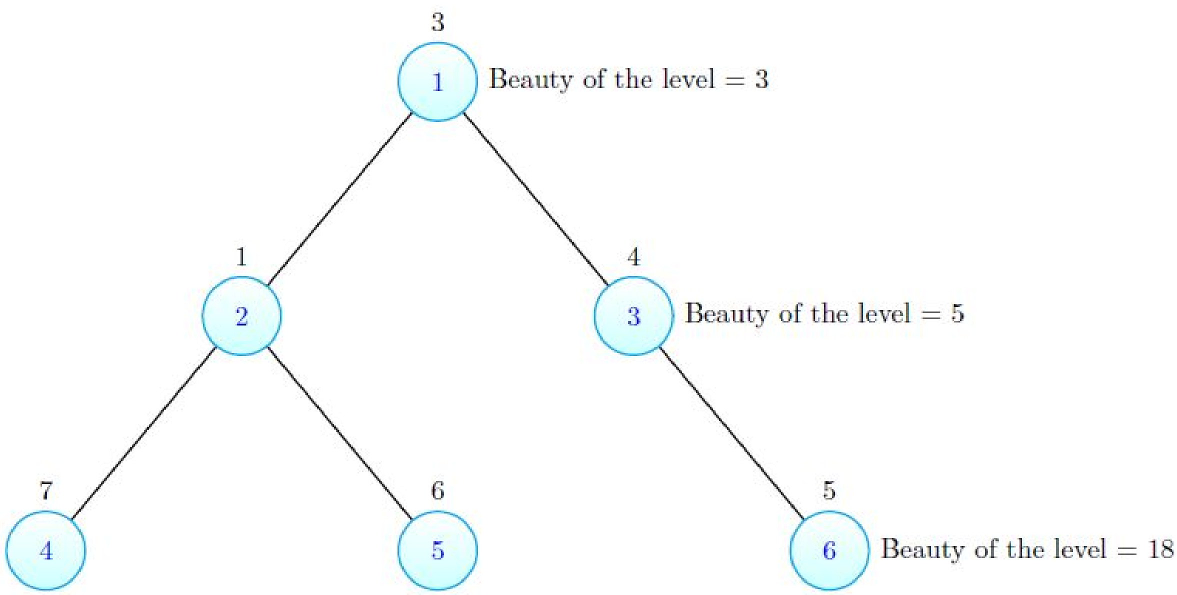
Original Tree of The First Test Case



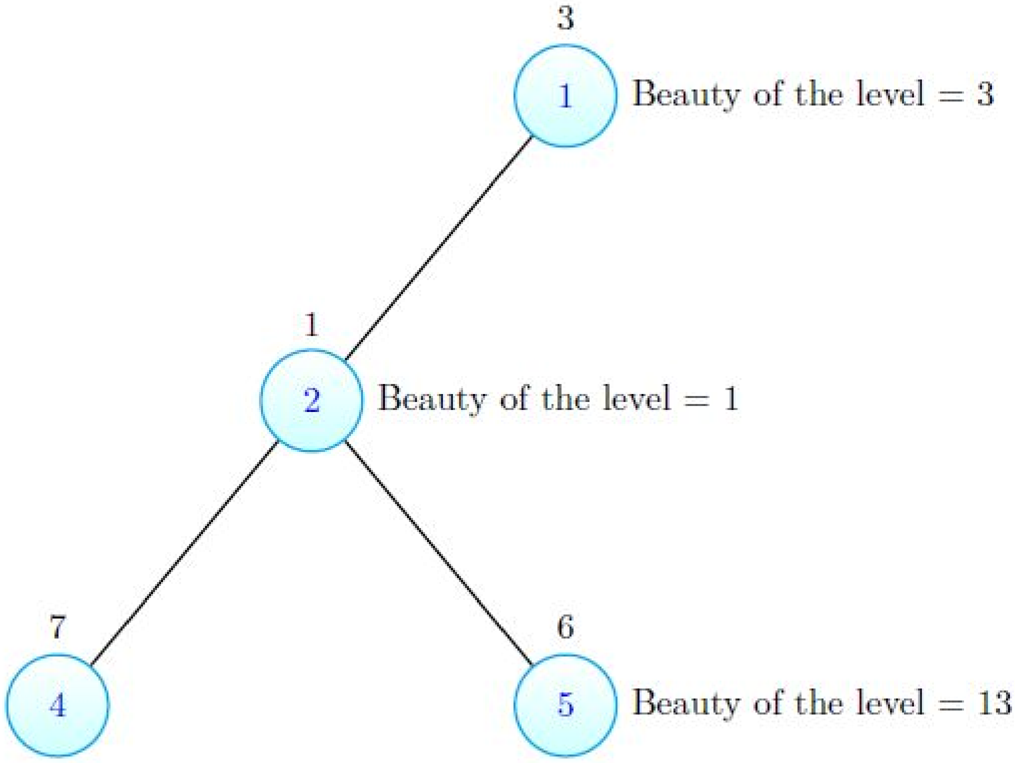
After Removing Subtree of node 4



Original Tree of The Second Test Case



After Removing Subtree of node 3



Problem E - Diverse Group

It's the year 3018 and you are the head of the Bioinformation Articulation in Cosmic Space (BACS) program at NASA. Under this program, NASA has decided to send **K** humans to planet Naboo. The whole of humanity is very excited about this and everyone wants to be a part of this voyage. But alas, there is only room for **K** people. Now, there are **N** regions in planet Earth. NASA asked each of these regions to compile a list of **M** people from that region who would be fit for this journey. The listing is complete and now you have the **N** lists at your disposal (one for each region, each containing **M** people). Among these **N*M** people, you want to select **K** people for this journey. But you want to make the selected group as diverse as possible. Therefore, you have decided not to select two person from the same region. Being a math enthusiast, you want to know how many different ways you can make this group. Two groups are different if there is someone who is in one group but not in the other.

Input Specifications

The first line contains a positive integer **T (T ≤ 1000)** denoting the number of test cases. In each of the following **T** lines, there will be three space separated integers **N, M & K** where N denotes the number of regions, M denotes the number of people from that region and **K** denotes the number of people who will sent to planet Naboo.

Constraints

2 ≤ N,M ≤ 10000000
1 ≤ K ≤ N*M

Outputput Specification: For each case, print the number of ways you can make that group. Since the answer can be rather big, print the answer modulo **1000000007 (10^9 + 7)**.

Input	Output
5 2 2 1 2 3 2 3 2 5 4 2 3 5 4 4	4 9 0 32 1280

Problem F - Football Free Kick

In a football match, when a player takes a freekick, opponent team players create a wall in front of him to prevent the goal. The freekick taker tries to shoot the ball in such a way that it can manage to go through the wall and score a goal. It is always important to know how many holes in the wall are there, so that the player can shoot the ball through a chosen hole. A segment of consecutive empty places in the wall is called the hole.
Counting hole in a wall is easy. But the opponent team will always change their player positions constantly to confuse the freekick taker.
In this year, one football world cup team hires you to work as their technical consultant. You need to write a program for them which can calculate the number of holes in the wall after every change the opponent team makes. Length of the wall is **N**. At the beginning, first **K** positions (1 to **K**, where **K<N**) are occupied by K players.
A change can be expressed by two values **x** and **y** **(1≤ x,y≤N, x≠y)**. One player from position x will move to position y.
Note:

1. It will be always possible to move a player from position **x** to **y**. Position y will be empty before a player moves from position **x**. And also one player will always exist to move from position **x**.

Input Specification

The first line contains a positive integer **T**, the number of test cases.
Every test case starts with three integers **N, K & Q** which refers to the length of the wall, the number of players and the number of changes opponents makes accordingly.
Next, each **Q** line contains two integers **x** and **y** **(where 1 ≤ x,y ≤ N)**.
It is guaranteed that all changes are valid.

Constraints

- $T \leq 50$
- $1 \leq K < N \leq 10^9$
- $1 \leq Q \leq 10^5$

Output Specification

For each case print the case number. Next, **Q** line will contain the number of holes in the wall after **i^{th} (1 ≤ i ≤ Q)** change.
See sample output for more clarification.

Input	Output
2 10 5 10 1 10 3 7 5 6 2 9 4 8 6 1 8 3 9 5 10 2 7 4 2 1 1 1 2	Case 1: 2 4 4 3 1 1 3 4 3 1 Case 2: 1

Problem G - GCD and LCM of 3 numbers

LCM(Least common multiple) of two numbers **a** and **b** is, **LCM(a,b) = (a*b) / GCD(a,b)**, where **GCD(a,b)** is greatest common divisor of **a**, **b**.

Similarly you can assume a new function that takes three numbers **a**, **b** and **c** where **F(a,b,c) = (a*b*c) / GCD(a,b,c)**, where **GCD(a,b,c)** is greatest common divisor of **a**, **b** and **c**.

You can see **LCM** of 3 numbers, **LCM(a, b, c)** is different than **F(a, b, c)**.

You are given **F(a,b,c)** and **GCD(a,b,c)**. You have to tell number of unique triples **(a,b,c)** which satisfy above equation where **1≤ a ≤ b ≤ c**.

Input Specification

In the first line of input, there will be an integer **T**, number of test case. In next **T** lines, there will be two integer numbers which represent **F(a, b, c)** and **GCD(a, b, c)**.

Output Specification

For each case, output the numbers of unique triples (a,b,c) will satisfy above formula.

Constraints

1 ≤ F(a, b, c), GCD(a, b, c) ≤ 1e14

Input	Output
5 7 1 12 1 8 2 18 3 34 1	1 4 1 1 2

Explanation

When F(a,b,c) = 7 and GCD(a,b,c) = 1, only valid triple is,
{1,1,7} as (1*1*7) / GCD(1,1,7) = 7/1 = 7 = F(a,b,c)

When F(a,b,c) = 12 and GCD(a,b,c) = 1, valid triples are,
{1,1,12} as (1*1*12) / GCD(1,1,12) = 12/1 = 12 = F(a,b,c)
{1,2,6} as (1*2*6) / GCD(1,2,6) = 12/1 = 12 = F(a,b,c)
{1,3,4} as (1*3*4) / GCD(1,3,4) = 12/1 = 12 = F(a,b,c)
{2,2,3} as (2*2*3) / GCD(2,2,3) = 12/1 = 12 = F(a,b,c)

When F(a,b,c) = 8 and GCD(a,b,c) = 2, only valid triple is,
{2,2,4} as (2*2*4) / GCD(2,2,4) = 16/2 = 8 = F(a,b,c)

When F(a,b,c) = 18 and GCD(a,b,c) = 3, only valid triple is,
{3,3,6} as (3*3*6) / GCD(3,3,6) = 54/3 = 18 = F(a,b,c)

When F(a,b,c) = 34 and GCD(a,b,c) = 1, valid triples are,
{1,1,34} as (1*1*34) / GCD(1,1,34) = 34/1 = 34 = F(a,b,c)
{1,2,17} as (1*2*17) / GCD(1,2,17) = 34/1 = 34 = F(a,b,c)

Problem H - Little T2 and Derangements

Little T2 has a dream to live in the Great Chinese Kingdom of Tanguzi. People of Tanguzi are very fond of playing with permutations and derangements. So, Little T2 started playing with permutations and derangements too. One day her mother got angry with her and gave her a task involving permutations and derangements and told her if she is unable to solve this task, she can not live in the Great Chinese Kingdom of Tanguzi. But the task is too hard for her. Can you help her to fulfill her dream!

A permutation of length **n** is a sequence of distinct positive integers of length **n**, where each integer is at most n. Example: {1,2,3}, {1,3,2}, {2,1,3}, {2,3,1}, {3,1,2}, {3,2,1} are all possible permutations of length 3. In permutation {1,2,3} all the integers are sorted in ascending order. Let's call this permutation Initial_Permutation.

A derangement of length n is a permutation of length n where each integer does not belong to it's original position. For each integer, the original position is where it should be in the Initial_Permutation. Example: {2,3,1} and {3,1,2} are the only two derangement of length 3.

A permutation $\mathbf{P_a}$ is lexicographically smaller than permutation $\mathbf{P_b}$ if there is a position where the permutations have their earliest mismatch then the element of that position in $\mathbf{P_a}$ is smaller than the element of that position in $\mathbf{P_b}$. All the permutations of length 3 shown above are in lexicographically sorted order from smaller to larger.

Little T2 is given two integers **N** and **K**. You have to help her to tell what is the $\mathbf{K}^{th}$ lexicographically smallest derangement of length **N**.

Input Specification

The first line of input contains an integer **T**, denoting the number of test cases. In each next **T** lines there will be two integers **N** and **K**.

Output Specification

For each test case you have to print the $\mathbf{K}^{th}$ lexicographically smallest derangement of length **N** in a single line. Two integers should be separated by a single space. Check the sample output for clarification.

Constraints

1 ≤ T ≤ 1000
2 ≤ N ≤ 100
1 ≤ K ≤ 2,000,000,000,000,000 (2e15)

For each subtask it is guaranteed that **K** will always be less than or equal to the number of possible derangements of length **N**.

Input	Output
5 2 1 3 2 4 3 5 26 7 666	2 1 3 1 2 2 4 1 3 4 3 1 5 2 4 1 7 5 6 2 3

Problem I - Marbelous Meena

Meena likes marbles a lot. She has a huge collection of marbles. Today Meena found some more marbles in a basket and, as usual, started playing with them. Soon she lost control and scattered all the marbles around into **N** piles, the **i**-th of which consists of **A[i]** marbles. That's when her mom came in. Apparently, mom kept those marbles for decorating purposes. She became very angry seeing all this mess. She asked Meena to sort this out right away, by moving all marbles to a single pile.

But as a punishment, Meena can't move the marbles around freely. She can move some marbles from one pile to another pile if and only if the size of the latter pile gets doubled after moving those marbles there. For example, if there are two piles where one has **5** marbles and the other has **3**: Meena can't move **2** marbles from the first one to the second, but can certainly move **3** marbles since then the size of the second pile will be **6**, twice the previous size.

Meena is not sure if the task is indeed possible. If it is not possible to move all marbles to a single pile this way, she will change the size of some of the piles by adding some marbles from her collection to the piles secretly, so that the task becomes possible.

Meena asks for your help. Please tell her if the task is possible, and if not possible, at least how many piles need to be changed to make the task possible.

Input Specification

The first line of the input file contains a single integer **T**, the number of test cases. The following lines describe the **T** test cases. Each test case consists of two lines. The first line contains a single integer **N**, the number of piles. The next line contains **N** integers: the $\mathbf{i}^{\text{th}}$ integer is the size of the $\mathbf{i}^{\text{th}}$ pile.

Output Specification

For each case on a single line, print the case number first. If the task is possible, print "YES" (without quotes). Otherwise print "NO" (without quotes) and the minimum number of piles to change (see sample cases for better understanding).

Constraints

- $0 < T < 10^5$
- $1 < N < 10^5$
- $0 < A[i] < 2^{40}$
- There are at most 10 tests with N > 2

Input	Output
3 2 1 3 2 2 4 3 6 9 9	Case 1: YES Case 2: NO 1 Case 3: YES

Problem J - Non Super Boring Substring

You'll be given a string **S** and an integer **K**. You have to find the number of non super-boring substring inside **S**. A substring is called super-boring if it contains any palindromic substring of length ≥ **K** inside it.

For example, if **S** = “ababba” and **K** = **3**, then the substring “abab” is a super-boring substring because it contains a palindrome “aba” whose length is ≥ 3. But the substring “bba” is a valid non super-boring substring.

Find the number of substrings which are not super-boring. Two substrings are considered different only if their starting or ending positions are different.

Input Specification

The first line of input contains an integer **T**, denoting the number of test cases. The first line of each test cases contains an integer **K** and the next line contains the string **S** consisting of lower case English alphabets.

Constraints

$1 \leq T \leq 100$
$K \leq |S| \leq 10^5$
Sum of $|S|$ over all test cases $<= 10^6$

Note: Here |S| denotes the length of the string S.

Output Specification

For each test case you have to print the number of non super-boring substring of the string **S**.

Input	Output
3 3 bababe 4 abcbbcbc 3 aabacecc	12 27 20

Explanation

In the first test case, here is the list of 12 non super-boring substrings: b, ba, a, ab, b, ba, a, ab, abe, b, be, e.

Problem K - Ray Ray Array

You will be given an array **A** which contains **N** numbers. You have to perform **Q** updates.

Update type 1: **1 X Y**
Which means you will subtract Y from those numbers which are currently less than or equal to X in the array.

Update type 2: **2 X Y**
Which means you will add Y to those numbers which are currently greater than or equal to X in the array.

All updates are given as 1-based indexing.
After performing all the updates, print the final array.

When a number is updated due to subtraction or addition with Y, we call this event a **change**.
It is guaranteed that, there will be **at most N changes** due to **negative Y** in a test case.
See the sample test case and explanation for better understanding.

Input Specification

In the first line there will be one integer **T**, denoting the number of test cases.
The second line will contain two Integers **N** and **Q**, they are size of the array and number of updates to perform.
There will be **N** integers in the next line, the initial values of the array.
Each of the next **Q** lines will contain three integers: **P X Y**. P = 1 corresponds to update of type 1 and P = 2 corresponds to update of type 2, as described above.

Output Specification

After performing all the updates one after another, print N integers of final array.

Constraints

$1 \leq T \leq 5$
$1 \leq N, Q \leq 10^5$
$-10^9 \leq A[i] \leq 10^9$
$-10^9 \leq X \leq 10^9$
$-10^9 \leq Y \leq 10^9$

Summation of all N in a test file will be at most 2×10^5 and summation of all Q in a test file will be at most 2×10^5.

Input	Output
1 5 4 4 2 6 2 -5 1 2 3 2 1 3 1 1 1 1 -1 -1	7 -1 9 -1 -8

Explanation

After first update the array becomes: 4 -1 6 -1 -8
After 2nd update the array becomes: 7 -1 9 -1 -8
After 3rd update the array becomes: 7 -2 9 -2 -9
After last update the array becomes: 7 -1 9 -1 -8

Here total number of "changes" is 11. 3 numbers were reduced in the first update, 2 numbers were increased after the second update, 3 numbers were reduced in the 3rd update, and 3 numbers were increased in the last update. Please note, here total number of "changes" due to negative Y is 3.

Problem L - School Reunion

So, the alumni association of your school is arranging it's annual reunion tomorrow. **N+1** people will attend the ceremony, which includes Mr. Kashem, a notable alumnus of the school. But he can't stay for a long period of time, because you know, he has other important things to do. However, Mr. Kashem wants to meet at least **P** persons. His assistant has acquired information about tomorrow's plan of the **N** other persons - the time they will arrive, and the time they will exit. Now he needs to figure out the minimum amount of time Mr. Kashem has to stay on the reunion to meet at least **P** persons. It is obviously complicated to solve this problem by hand, and Mr. Kashem's assistant has zero knowledge on programming. Can you help him solving this problem?

Input Specification

First line of input contains an integer **T**, indicating the number of test cases. Each test case starts with a line containing two integers **N** and **P**. Here **N** indicates the number of people other than Mr. Kashem, who will attend the reunion. And **P** indicates the minimum number of people Mr. Kashem wants to meet. Following **N** lines contain two integers each, $\mathbf{st_i}$ and $\mathbf{en_i}$, denoting the entry and exit time of the $\mathbf{i}^{th}$ person.

Constraints

$1 \le T \le 30$
$1 \le P \le N \le 100000$
$1 \le st_i \le en_i \le 1000000000$
Use fast input method. For example, prefer using scanf to cin in C/C++.

Output Specification

For each test case, print the case number starting from 1, followed by the answer to the case in a separate line. See sample I/O for better understanding.

Input	Output
2 5 3 1 3 8 12 6 9 14 17 2 7 3 3 1 6 4 7 6 9	Case 1: 1 Case 2: 0

Explanation

In case 1, if Mr. Kasem enters at time 7 and exits at time 8, then he'll be able to meet person 2, 3 and 5. Here he stays for (8-7)=1 unit time.

In case 2, he can enter and exit at time 6. Yes, it is allowed to enter and exit at the same time. In this arrangement he gets to meet all three persons.

Problem M - TFF

Let **F(x)** and **G(x)** be two functions of **x** with all positive integer co-efficient and let **H(x)** be their multiplication.

Formally,

$F(x) = a_0 + a_1*x + a_2*x^2 + ... + a_{n-1}*x^{n-1}$
$G(x) = b_0 + b_1*x + b_2*x^2 + ... + b_{m-1}*x^{m-1}$
$H(x) = F(x) * G(x)$
$= c_0 + c_1*x + c_2*x^2 + ... + c_{k-1}*x^{k-1}$

Where $a_{n-1} != 0$ and $b_{m-1} != 0$ and $c_{k-1} != 0$
Given **G(x)** and **H(x)**, find **F(x)**.

Input Specification

First line will contain an integer **t** denoting the number of test cases. Then the description of the **t** test cases will follow. The first line of every test case will contain the integer **m**. The next line will contain m integers which denotes b_0, b_1, b_2, ... , b_{m-1}. Description of **H(x)** will be given in the following two lines in exactly the same format.

Output Specification

For every test case you need to print the **F(x)** function in the same format as **G(x)** and **H(x)**, which were given in the input.

Constraints

$1 <= t <= 10$
$1 <= a_i, b_i <= 1000$
$1 <= n, m <= 1000$

Sample

Input	**Output**
1 2 7 3 3 14 41 15	2 2 5

Explanation of sample 1

Here,
G(x) = 7 + 3*x
H(x) = 14 + 41*x + 15*x^2
= (7+3*x) *(2+5*x)
So, F(x) = 2 + 5 * x