A. Charm Is Not Always Enough

Score: 100

CPU: 0.5s

Memory: 1024MB

One day Munta was sitting under a tree when a fairy appeared before him. Being impressed by his charm, the fairy gave him a machine which can pour some mysterious liquid called “Felix Felicis”. Every time Munta pressed a button on the machine to pour down some of the liquid, exactly M litre of liquid would come down. But he couldn’t find bottles of same sizes! Munta placed each bottle in the machine and kept on pressing the button until the bottle was filled to its complete capacity. But he couldn’t change the bottle when the liquid was pouring down. As a result, some of the liquid overflowed and it got wasted. Since the liquid is very valuable, the fairy didn’t like that at all, and wanted to punish Munta for wasting so much liquid. Given the number of bottles N, amount of liquid coming down at a time M and the capacity of N different bottles C1, C2, C3,...... Cn your job is to inform the fairy how much liquid Munta wasted so that the fairy can punish Munta accordingly.

Input

The first line will contain an integer T (1<=T<=100) representing the number of test case.

For each test case, there will be two integers N and M. (1 <= N<=10^5 and 1<= M<=10^5).

Then there will be N integers representing the capacity of the bottles (1 <= Ci <= 10^5).

Output

For each test case, output the total amount of liquid wasted by Munta in separate lines.

Samples

Input	Output
1 5 10 11 12 13 15 17	32

Input	Output
2 2 100 100 55 3 100 100 150 250	45 100

B. Max and Alexis Plan to Conquer the World

Score: 100

CPU: 1.25s

Memory: 1024MB

There is this kitten Alexis. His favorite game is playing football. Then there is another kitten, Max. His favorite game is biting "hoomans". So the first time they met, naturally they couldn't agree on the game they should play together. So after two minutes of arguing (as kittens cannot focus on anything for more than two minutes), Max proposed, "Let's conquer the world of hoomans. They have been there for too long. We are the real kings and queens who should rule the kittenland." Alexis agreed with Max and immediately they started formulating their masterplan. Now they are facing a problem. There are N kittens in kittenland currently. Each year the number of kittens increase by R%. Max has decided that they wouldn't start their conquest before they have at least P kittens in their army. So now they are struggling a bit to find out how many years they have to wait till they can start their conquest. For example, let's say, there are currently 10 kittens in kittenland. If their number increases by 15% each year, then there will be 11.5 kittens after one year. So, if they need 11 kittens to be present in their army, they would need to wait for one year. [Yes, it is very much possible that there can be 11.5 kittens. You know, it's mathematics and stuff.] Now if you can solve the problem for queen Max and king Alexis, they might let you go after they conquer the world.

Input

The first line contains a positive integer T, number of test cases. In each of the following T lines, there will be 3 integers, N, R, and P. (The number of kittens currently in the kittenland, the percentage of growth of population each year and the minimum number of kittens needed for the conquest.)

Output

For each case print the case number and then a single integer which is the number of years, that Max and Alexis have to wait before starting the conquest.

Input Constraints

Subtask 1: (20 points)

1 <= T <= 1000

Subtask 2: (80 points)

1 <= T <= 250000

For both subtasks:

1 <= N <= 10^9

1 <= R <= 100

1 <= P <= 10^19

Sample

Input	Output
3 10 15 11 10 15 12 446390799 24 650597780460575459	Case 1: 1 Case 2: 2 Case 3: 99

Input file is very large. Please use faster methods for taking inputs and giving outputs. Avoid using cin/cout.

C. Being Common is Too Mainstream

‘Amra Shobai Raja’ is a mess for numbers. They have some exquisite rules. Infinitely many numbers can live in one room. And, the rent for a room is the LCM of all the numbers living in that room. But, to reduce the rent, they can do one thing. Which is dividing any two or more numbers by their any common divisor as many times as they want. But, each time, you'll have to divide all the numbers which are divisible by that common divisor once.

Some new room members of such a room have come to you seeking help. You’ll have to calculate the least amount of rent they’ll have to pay for them. As it can be pretty big, they’ll be happy if you tell it modulo 1000000001.

Input

There will be two lines of input. In the first line, there will be an integer N. Which is the number of room members. And then the next line will consist of N single integers a_i,(1 <= i <= N) the N room members. The numbers will be separated using space.

Subtask 1 : Point 10

1 <= N <= 3

1 <= a_i <= (2^61) - 1

Subtask 2 : Point 50

1 <= N <= 100000

1 <= a_i <= 601

Subtask 3 : Point 40

1 <= N <= 1000

1 <= a_i <= 63211236

Output

Print a single integer in a single line mentioned in the description. Don’t forget the newline character.

Samples

Input	Output
10 1 2 3 4 5 6 7 8 9 10	42

Input	Output
1 2305843009213693951	907850945

D. Shaat Chara

Score: 100

CPU: 1s

Memory: 1024MB

Meena and Razu have been playing Shaat Chara for many years. But they think the game is too violent. So, they came up with a new game. In this game, instead of just 1 pile, they start with n piles of stones. The game is for two players. Each player takes turn alternatively. In each turn a player can select any of the piles that has at least 1 stone left, and then remove some stone (at least one) from it. The game goes on like this until there are no stones left. The player who can't make a move loses.

At first, Razu was very happy because there was no chance of injury in this game. But he quickly grew frustrated, because Meena was winning all the games. Meena is very clever. She always plays optimally. That means, if there is a chance to win, she will always win. So, Razu came to you with a particular state of the game. He wants to know how many possible moves he may take such that Meena's win does not become guaranteed.

Input

In a single line, you will be given T, the number of test cases.

T test cases follow. In each test case, you will be given n, the number of piles. Then in a single line, you will be given n numbers ai, the number of stones in the ith (1 <=i <= n) pile.

Subtask 1 - 40 Points

T <= 100

n = 3

1 <= ai <= 50

Subtask 2 - 60 Points

T <= 100

1<=n<=10000

1 <= ai <= 100000

Output

You have to print the case number first in the form “Case 1: ”.

For each case you have to print only the number of winning moves from that particular state of the game. See sample for clarification.

Sample

Input	Output
3 1 1 2 1 1 3 1 1 1	Case 1: 1 Case 2: 0 Case 3: 3

Note that, winning moves mean the moves that will lead Meena to a losing state. For example, if the current state is {1, 3}, Razu can remove 2 stone from the second pile, which is a winning state. Because this move will lead to the state {1, 1}. Then Meena will remove 1 stone from any pile, and Razu will remove the other in the next turn. And Meena will lose the game, because there will be no remaining stone. But if Razu removed only 1 stone from the second pile, that will lead to the state {1, 2}, which is a winning state for Meena. Thus Meena will win.

Also note that, this is a normal nim game of n piles. In a normal Nim game, the player making the first move has a winning strategy if and only if the nim-sum of the sizes of the piles is nonzero. Otherwise, the second player has a winning strategy. Nim sum of n piles is defined as follows.

Nim_sum = a1 ^ a2 ^ a3 ^……… ^ an , where ^ is the Bitwise X-OR operation.

Now you might be wondering how it would even work. To put it simply, think of it this way. If the theory is correct, then the first player wins when the nim sum of the current piles are non-zero, and loses otherwise. Now, consider a case where he is in a zero nim-sum position. He must make a change to the piles, which will lead the nim-sum to non-zero, which is a winning state for the opponent. On the other hand if he is in a non-zero nim-sum position, it can be proved that he can always make the nim-sum zero in a single move, which is a losing state for his opponent.

E. Just One Swap

Score: 100

CPU: 1s

Memory: 1024MB

You are given an array of size N. How many distinct arrays can you generate by swapping two numbers for exactly once? The two selected numbers can be equal but their positions in the array must be different.

Input Specification

The first line of the input contains a single integer T, denoting the number of test cases. Every test case starts with an integer N. The next line contains N integers, the numbers of the array.

Output Specification

For each tescase output the answer in a single line.

Constraints

Subtask 1: 2 <= N <= 50 (15 points)

Subtask 2: 2 <= N <= 100000 (85 points)

In all subtasks:

1 <= T <= 5

1 <= Value of a number in the array <= 100000

Sample

Input	Output
1 5 2 3 2 3 3	7

You can generate the following arrays:

2 3 2 3 3

2 2 3 3 3

2 3 3 2 3

2 3 3 3 2

3 2 2 3 3

3 3 2 2 3

3 3 2 3 2

F. Halum and Candies

Score: 100

CPU: 1s

Memory: 1024MB

Halum has been elected as the captain of his school's football team. To celebrate this, he is going to throw a party. Halum bought candies of N different flavors for the party. There are a_i number of candies of the i_{th} flavor. To satisfy a guest he has to give him/her some positive number of candies of at least K different flavors. Otherwise the guest is not satisfied. Given this information now you have to find the maximum number of guests Halum can satisfy with the available candies.

Input Specification

First line of the input contains an integer T, denoting the number of test cases. T cases follow.

First line of each case contains two integers N and K. N space separated integers a_1, a_2… a_N follow in the next line. The i_{th} of these integers a_i denotes the number of candies of i_{th} flavor.

Output Specification

For each case print one line containing "Case X: Y" (without the quotes) where X is the case number starting from 1 and Y is an integer denoting the maximum number of guests who can be satisfied.

Constraints

Subtask 1: 30 points

- $1 \leq T \leq 100$
- $1 \leq K \leq N \leq 20$
- $0 \leq a_i \leq 100$

Subtask 2: 70 points

- $1 \leq T \leq 100$
- $1 \leq K \leq N \leq 1000$
- $0 \leq a_i \leq 10^9$

Sample

Input	Output
3 3 3 1 2 3 3 1 1 2 3 3 2 3 2 4	Case 1: 1 Case 2: 6 Case 3: 4

G. XOR 'em all!

Score: 100

CPU: 5s

Memory: 1024MB

You are given an array a of n integers, all integers ranging in between 0 and 2^20-1 (inclusive). You are asked to perform m operations on the array in order. All operations are either of the following two types:

1 l r v: You are asked to find a position i, such that i is in between l and r (inclusive) and the absolute difference in parity between a[i] and v is minimum, for all a[l], a[l+1], a[l+2],...., a[r]. Parity of a number is defined as the number of ones in its binary representation. If there are multiple such positions i, output the minimum among them.

2 l r: Update all elements in the range a[l], a[l+1], a[l+2], ... , a[r] by XOR-ing them with the value 2^20-1.

Input

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

Each test case starts with n, m.

The following line contains n space-separated integers, denoting a[1], a[2], a[3], ... a[n], all within the range 0 to 2^20-1 (inclusive).

The following m lines contain the descriptions of the operations. The i th such line describes the i th operation, and is of either of the two formats:

1 l r v (1<=l r<=n, 0<=v<=2^20-1)

2 l r (1<=l, r<=n)

Output

For each test case, output the case number in a single line as shown in the sample. For each operation of type 1, print an integer in a single line, representing the answer to the query.

Subtask 1 (10 points)

No operations of type 2

1<=T<=100

1<=n,m<=100

Subtask 2 (40 points)

No operations of type 2

1<=T<=5

1<=n<=10^5

1<=m<=10^5

Subtask 3 (50 points)

1<=T<=5

1<=n<=10^6

1<=m<=10^5

Samples

Input	Output
1 6 2 2 5 3 15 6 8 1 1 6 7 1 1 3 1024	Case 1: 2 1

Input	Output
2 4 3 1 5 8 2 2 3 4 1 1 4 0 1 1 2 1 5 1 21 26 16 19 71 1 2 4 71	Case 1: 1 1 Case 2: 2

N. B.

1. I/O files are huge, use faster input-output methods.
2. The second sample is valid only for task 3.

H. Simple Path

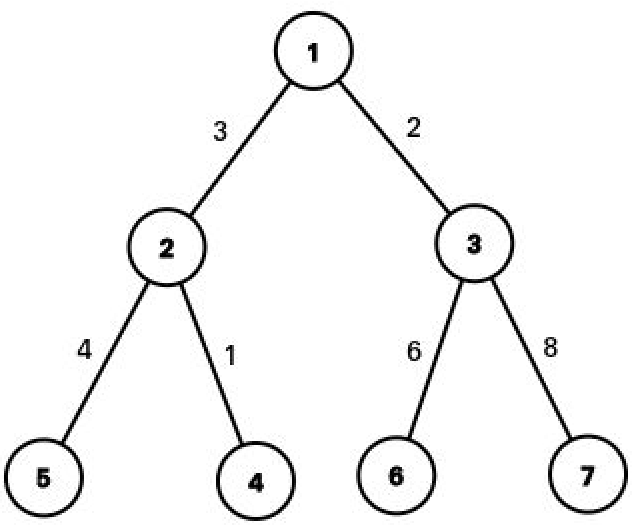
Score: 100

CPU: 1s

Memory: 1024MB

You will be given a weighted rooted tree, where root is node 1. For each subtree of that tree, you will have to compute the summation of lengths of all possible simple paths present in that subtree. Eventually you need to print the summation of those values modulo 1000000007.

Image of the tree in the first sample is given below.



Input Specification:

First line of input will contain an integer T (1 <= T <= 10) denoting the number of test cases. Each test case will begin with an integer N, indicating number of nodes. The following N-1 lines will have three integers each, U (1 <= U <= N), V (1 <= V <= N), W (1 <= W <= 1000000000), which denotes that there is an edge in the tree from node U to node V having W weight.

Input File is large. Use fast I/O methods.

Subtask 1: 1<= N <=1000 (15 points)

Subtask 2: 1<= N <=100000 (85 points)

Output Specification

For each test case, print the case number, and then print the answer to the problem modulo 1000000007 in separate lines.

Sample

Input	**Output**
2	Case 1: 212
7	Case 2: 109
1 2 3	
1 3 2	
2 4 1	
2 5 4	
3 6 6	
3 7 8	
6	
1 2 3	
1 3 2	
1 4 4	
3 5 7	
3 6 1	

Explanation of the 1st sample:

Summation of lengths of all possible sample paths for each subtree is given below:

1: 174

2: 10

3: 28

4: 0

5: 0

6: 0

7: 0

Explanation of the 2nd sample:

Summation of lengths of all possible sample paths for each subtree is given below:

1: 93

2: 0

3: 16

4: 0

5: 0

6: 0