National Collegiate Programming Contest (NCPC) 2023

Department of Computer Science and Engineering, Jahangirnagar University

Supported by

DIGITAL BANGLADESH
Skilled. Equipped. DigitalReady.

ICT DIVISION
FUTURE IS HERE

BANGLADESH COMPUTER COUNCIL
COMPUTER FOR EVERYTHING

9 March, 2024
You get 15 Pages, 11 Problems & 300 Minutes

Bangladesh Association of Problem Setters

Sponsored By

Platinum Sponsors

Gold Sponsors

Other Sponsors

Problem A

A WeirdLand

Bangladesh
Association
of
Problem
Setters

In WeirdLand, each couple continues to have children until they have at least **x** boys and at least **y** girls. As soon as they have at least **x** boys and **y** girls, they stop having children. The chance of having a girl is **g** per billion (**10⁹**), while a boy's chance is **(10⁹ - g)** per billion. Assuming an infinite number of couples, determine the following ratio:

$$\frac{Expected\ number\ of\ boys}{Expected\ number\ of\ girls}$$

Output the ratio as a simplified fraction with coprime positive numerator and denominator. Always include the denominator, even if it is **1**.

Input

The first line will contain a single integer **T ($1 \le T \le 10^4$)**. Each test case will have three integers **x**, **y**, **g ($0 \le$ x, y $\le 10^9$, $1 \le$ g $< 10^9$)**. It is guaranteed that both **x** and **y** will not be **0** at the same time.

Output

Print the case number in a single line followed by the ratio. Please see the sample for details.

Sample Input

```
2
1 1 500000000
10 10 500000000
```

Output for Sample Input

```
Case 1: 1/1
Case 2: 1/1
```

Problem B

Anagram Again

According to Wikipedia: An anagram is a word or phrase formed by rearranging the letters of a different word or phrase. For example: **"evil"** and **"vile"** are anagram of each other, **"silent"** and **"listen"** are anagram of each other. Accordingly, all the anagrams for the string **"aba"** are: **"aba"**, **"aab"**, **"baa"**.

You will be given a dictionary of **N** words. You need to answer **Q** queries. In each query, you will be given a string. You need to answer how many words are there in the dictionary whose at least one anagram is present as a subsequence in the query string.

Input

The first line of the input contains a single integer **T (1 ≤ T ≤ 100)** which denotes the number of test cases. First line of each test case has an integer **N (2 ≤ N ≤ 100)** denoting the number of words in the dictionary. Each of the next **N** lines contains one string.
Next line contains an integer **Q (1 ≤ Q ≤ 100)**, the number of queries. Each of the next **Q** lines contains one string. All the strings in the input will contain only lowercase letters and the length of any string will not exceed **100**.

Output

The first line of each output should contain the test care number in the format **"Case X:"** where **X** is the test case number. Each of the next **Q** lines should contain one number, the output for the associated query. Please see the sample input output for more clarification.

Sample Input

```
1
3
ncpc
code
champion
2
decon
hypomaniced
```

Output for Sample Input

```
Case 1:
1
2
```

Problem C

Election

Welcome to Sillyville, a beautiful country where **N** states are arranged in a straight line and these states are getting ready for its presidential election on **March 09, 2024**. Only two candidates are participating in this election. Mr. Hero Harris is the current president and wants to become president again. In the other corner, Mr. Villain Victor is his opponent. In this presidential election. Each state will act like a battleground, and whoever wins in each state gets an important point known as the electoral vote. So, there are a total of N electoral votes and the candidate who gets more electoral votes will win the election.

Government secret agencies predict that Mr. Hero Harris is set to win or lose in the ith state by X_i votes (So, X_i positive means Hero Harris will win, X_i negative means Hero Harris will lose, and $X_i = 0$ indicates a tie, and in that case, no one will get the electoral vote). To secure the biggest victory, Hero Harris aims to strategically merge some adjacent states (alter state borders). Mr. Hero Harris will win and get one electoral vote after merging states numbered from **l** to **r** if and only if $\sum_{i=l}^{r} X_i > 0$.

The future of Sillyville rests in your hands! Your task is to figure out the optimal way to merge adjacent states (change state borders) so that Hero Harris can maximize the electoral vote difference against his opponent. More formally, your task is to maximize Electoral_Vote$_{HeroHarris}$ - Electoral_Vote$_{VillainVictor}$.

Input

The first line contains an integer **T (1 ≤ T ≤ 100)**, representing the number of test cases. Each test case contains two lines of input. The first line contains an integer **N (2 ≤ N ≤ 10⁵)**, representing the number of states in Sillyville. The second line contains **N** space-separated integers, representing the expected winning or losing margin X_i **(-10⁴ ≤ X_i ≤ 10⁴ , X_i≠0)**. ***Input file is large. Please use faster input/output methods.***

Output

For each test case, output a single line in the format **"Case X: Y"** where **X** is the case number and **Y** is the desired answer.

Sample Input

```
4
5
-3 4 1 2 3
5
-1 5 -10 6 -4
1
-6
2
5 -5
```

Output for Sample Input

```
Case 1: 4
Case 2: 1
Case 3: -1
Case 4: 0
```

Explanation

Case 1: Consider merging the first two states (-3 4) (1) (2) (3).
Case 2: Consider merging first two and last two states (-1 5) (-10)(6 -4)

Problem D

Equal

Given four positive integers **a, b, c** and **d** you will have to find whether $\frac{a}{b}$ equals $\frac{c}{d}$ and report accordingly.

Input

First line of the input file contains an integer **N (0 < N ≤ 240000)** which denotes the lines of inputs. Each of the next **N** lines contains for integers **a, b, c, d (0 < a, b, c, d ≤ 18*10^{18})**.

Input file is large. Please be sure to use faster input/output methods.

Output

For each line of input except the first line, produce one line of output. If both fractions are equal then print `Equal` otherwise print `Not Equal`.

Sample Input	Output for Sample Input
2 1 2 3 4 1 2 1 2	Not Equal Equal

Problem E

Function! Not Again!!!

Bangladesh
Association
of
Problem
Setters

You are given a tree of **N** nodes where each node is associated with a value. You are also given a function F whose pseudocode is given below:

```
function F(int x):
    int ans = 0
    while x > 0:
        ans = ans + (x % 10)
        x = x / 10
    return ans
```

You will be given **Q** queries in the format **X U V**. Here, **X** is a positive integer while **U** and **V** are two nodes of the tree. For each query, you need to find $\sum F(X + A_i)$ for all the values A_i on the path from **U** to **V** inclusive.

Input

The first line of the input will be a single integer **T (1 ≤ T ≤ 10)** denoting the number of test cases.

Each test case starts with a line containing a single integer **N (2 ≤ N ≤ 10^5)** which is the number of nodes in the tree. Next line of the input will contain **N** space separated integers **A$_i$ (1 ≤ A$_i$ ≤ 10^9 and 1 ≤ i ≤ N)** which are the associated values of the nodes. Next **N-1** lines contain two integers **U** and **V (U != V** and **1 ≤ U, V ≤ N)** denoting there is a bidirectional edge between node **U** and **V**. The next line of the input contains **Q (1 ≤ Q ≤ 10^5)**, the number of queries and next **Q** lines will contain one query in the form of **X U V (1 ≤ X ≤ 10^9 and 1 ≤ U, V ≤ N)**. Here **X** is an integer while **U** and **V** are nodes from the tree. It is guaranteed that **the sum of N over all test cases doesn't exceed 10^5** as well as **the sum of Q over all test cases doesn't exceed 10^5.** *Input file is large. Please be sure to use faster input/output methods.*

Output

The first line of each test case should contain the case number in the format: **"Case Z:"** where **Z** is the test case number. Then you should print **Q** lines, one for each query containing the answer. For more clarity, please take a look at the sample IO.

Sample Input

```
1
5
1 2 3 4 5
1 2
2 3
2 4
1 5
3
1 3 4
3 3 1
4 4 5
```

Output for Sample Input

```
Case 1:
12
15
28
```

Problem F

Good Path on a Tree

Oh, No!!! Not another one of these.

You are given a tree with N vertices. Each vertex is represented by a unique integer from **1** to **N**. The simple path between two different vertices $[(u, v), u < v]$ is good if the sequence created by listing the vertices on the path from **u** to **v** follows the following properties.

- The path has $2k + 1$ vertices where k is a positive integer.
- Assuming the sequence of the vertices on the path is $h_1 = u, h_2, \dots, h_{2k+1} = v$ then the path satisfies the condition $h_1 < h_2 < \dots < h_k < h_{k+1} > h_{k+2} > \dots > h_{2k} > h_{2k+1}$.

So, how many different pairs of vertices $[(u, v), u < v]$ are in the tree with a good simple path between them? Please note that the simple path is also the shortest path between two vertices.

Input

The first line of the input contains an integer **T** denoting the number of test cases. Each test starts with an integer **N ($2 \le N \le 10^5$)**, the size of the tree. The next **N-1** line each contains two integers **a** and **b ($1 \le a, b \le$ N, a != b)**, representing an edge of the tree connecting the vertices a and b. It is guaranteed that ∑**N ≤ 2×10⁵** over all test cases and the input forms a valid connected tree.

Input file is large. Please be sure to use faster input/output methods.

Output

For each test case print one integer, the number of different pairs of vertices **[(u, v), u < v]** with a good simple path in the tree.

Sample Input

Sample Input	Output for Sample Input
3 3 1 3 3 2 4 1 4 4 2 4 3 3 1 2 2 3	1 3 0

Problem G

Hate Me if You Can

Team **Caribbean_Pirates** is facing a crucial moment at their home ground Jungle_Universe, in a **50** overs (**300 balls**) cricket match. Captain Jack Dumb is currently on strike, gearing up to face the next ball. The team needs **Y** runs to win from the last **X** balls, and Captain Jack is **Z** runs away from reaching his milestone. His partner, being the last man, can only defend without scoring any runs.

Rules of Cricket:

Scoring Runs	The batsman (Captain Jack) on strike can score **0, 1, 2, 3, 4,** or **6** runs on each ball.
Over Structure	Each over consists of six balls delivered by the bowler. After the completion of six balls, an over concludes.
Changing Strike	The batsmen swap ends after each completed run. This means that, after scoring an odd number of runs (**1** run or **3** runs), the batsmen switch positions.
End of Over	At the end of an over, the non-striker becomes the striker for the next over. If Captain Jack (on strike) doesn't score any odd runs on the last ball of the over, the partner will take strike in the next over.
Win or Lose	The batting team must achieve the target runs within the allocated **50** overs (**300 balls**) to secure victory. When the team touches or crosses the target, the batting team is declared as the winner. If not achieved within **50 overs**, the opponent will be declared the winner.

The primary goal of Captain Jack is to win the match for his team. If the captain can win and achieve the milestone, print **"Like a Boss!"**. If the captain can win but cannot reach the milestone, print **"Bravo Captain!"**.

If the captain can't win the match, print **"Love You Captain!"**.

Input

The first line contains an integer **T (1 ≤ T ≤ 300000)**, representing the number of test cases.
Each of the next **T** lines contains three space-separated integers **X, Y** and **Z (1 ≤ X ≤ 100, 1 ≤ Y ≤ 200, 1 ≤ Z ≤ 200)**, denoting the team needs **Y** runs from **X** balls, while Captain Jack needs **Z** runs to achieve his milestone.

Input file is large. Please be sure to use faster input/output methods.

Output

For each test case, print the test case number in the format **"Case X: "**, followed by the desired answer in a single line as stated above. See the sample Input/Output section for more clarity.

Sample Input

```
3
3 18 15
7 38 40
1 10 4
```

Output for Sample Input

```
Case 1: Like a Boss!
Case 2: Bravo Captain!
Case 3: Love You Captain!
```

Explanation

Case 1: The Captain can hit 3 sixes to win the match.

Case 2: The captain can score 3 runs, and then can hit six (06) sixes to get 39 runs. In this case, he can win the match for his team, but can't achieve the personal milestone.

Case 3: It is impossible to get 10 runs from one ball.

Problem H

Palindromic Subsequence

CP Ad Agency has a billboard in Dhaka. The billboard contains an array **S** of **n** strings consisting of English alphabets. The strings are made of LED lights. So the agency needs to keep some strings on and some strings off for an ad each day. But, each day the agency is given a range **[l. r]**. The strings outside the range must be kept off on that day. Now, the agency has to decide which strings to keep on and which to keep off such that the lit on strings maintaining the order in **S** are concatenated to form a palindrome **P**. As the agency always wants the maximum profit each day, the formed palindrome **P** needs to be the longest possible. Now for **q** days, it's your task to help the agency to find the length of the longest **P** the agency can make by choosing some strings from **S** with the range **[l, r]** of each day.

Input

The first line will contain a single integer **T (1 ≤ T ≤ 100)**. Each test case will start with two integers **n (1 ≤ n ≤ 200)** and **q (1 ≤ q ≤ 10⁴)** where **n** denotes the number of strings in array **S** and **q** denotes the number of queries. The second line of each test case will contain **n** space-separated strings. The following **q** lines of each test case will contain two integers **l** and **r (1 ≤ l ≤ r ≤ n)**. All the **n** strings consist of english lowercase and uppercase letters and a string's length will **not be greater than 10**. In any input file, it is guaranteed that the **sum of n** over all test cases won't exceed **2000** as well as **the sum of q** won't exceed **10⁵.**

Input file is large. Please be sure to use faster input/output methods.

Output

For the first line of each test case, print "**Case x:**" where **x** is the test case number. For the following **q** lines of each test case, print an integer which is the length of the longest **P** you can make by choosing some strings from the sub-array $S_lS_{l+1}...S_{r-1}S_r$.

Sample Input

Sample Input	Output for Sample Input
2 5 2 aa xy b aa yx 1 5 1 4 3 1 ab bc cd 1 3	Case 1: 6 5 Case 2: 0

Problem I

Picturesque Skyline II

The city you live in has a lot of high-rise buildings. Due to some city regulations, each building has a **unique height**. One day you went to the mountains outside the city. After a long hike you looked at the city to appreciate the beauty of it all. There you noticed something weird, actually two very weird things. The first being, from the right spot all the tall buildings look like they are on a single line. The other is that the buildings are forming multiple unusual groups of pyramid-like patterns with a small gap separating two adjacent groups. Two adjacent buildings in the same group don't leave any space.

A group of $2k + 1$ buildings where k is a positive integer may form a pyramid-like pattern If they meet the following condition. Assuming the heights of buildings are h_1, h_2, ... , h_{2k+1} when sorted by position of the building from left to right. Then the group of buildings need to satisfy $h_1 < h_2 < ... < h_k < h_{k+1} > h_{k+2} > ... > h_{2k} > h_{2k+1}$. So k building in increasing heights followed by the tallest building of the group which is then followed by k buildings in decreasing heights. The city is formed of several such pyramid-like patterns (with small gaps between adjacent groups). This makes the skyline of the city quite picturesque and unique. Assume there are no visible gaps between two adjacent buildings in a group.

Below are examples of two cities. The first one has a single pyramid-like pattern of 5 buildings. The second has 6 buildings in total with two pyramid-like patterns.
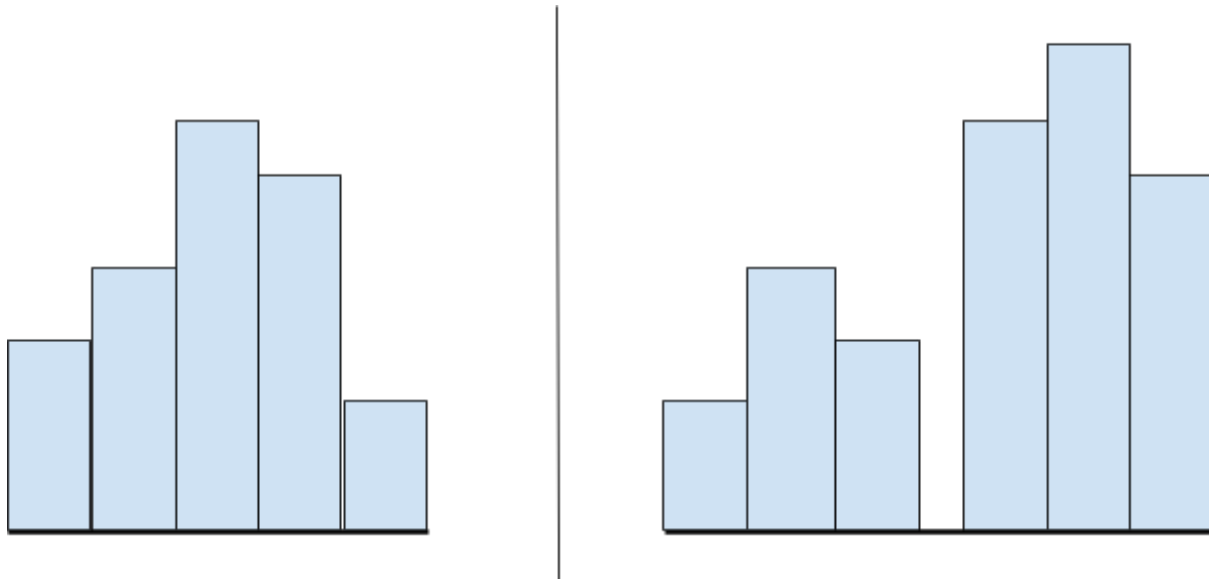

Fig: Example cities.

When you went to a different city as a tourist you noticed that the buildings in that city are **also on a straight line without any gaps between adjacent buildings** but they may not be grouped in a pyramid-like pattern. Oh the horror!! Now, hypothetically, let's assume you have two machines called Building Swapper 2000 and Group Maker 3000. Building Swapper 2000 can swap two **adjacent** buildings, but requires one unit of fuel to operate. Similarly Group Maker 3000 can split the buildings into one or more

groups so that two adjacent groups have a little bit of space between them but it is solar powered, so it doesn't need any fuel. Now, you have to follow through two steps in order.

- Use the Group Maker 3000 to split the buildings into groups of odd sizes greater than 2. After you have applied this step then there will be one or more groups of buildings (but may not be pyramid-like yet) with a small gap between two adjacent groups.
- Use the Building Swapper 2000 to swap adjacent buildings until each group becomes pyramid-like. You can only swap two adjacent buildings if they are part of the same group.

Finally, you will have some groups of buildings, where all of them are in a pyramid-like pattern and the city will have a picturesque skyline. The question for you is to calculate the minimum cost of fuel required to achieve the goal.

Input

The first line of the input contains an integer **T** denoting the number of test cases. The first line of each test contains a single integer **N (3 ≤ N ≤ 1000, N ≠ 4)**, the number of buildings in the city. The next line contains **N** integers denoting the heights of the building **(1 ≤ height ≤ 10⁹)**. It is guaranteed that Σ**N ≤ 10000** over all test cases and within each test case all the heights are unique.

Output

For each test case print one integer, the minimum amount of fuel required. It can be proven that under the restriction it is always possible to arrange the buildings to achieve the goal.

Sample Input

```
4
3
1 3 2
6
1 2 4 8 16 32
5
1 2 3 4 5
7
6 4 2 1 3 5 7
```

Output for Sample Input

```
0
2
3
9
```

Problem J

Rik and Vik

In a galaxy far, far away, Rik and Vik are space adventurers who love exploring new planets. During their journey, they discovered something amazing: the size of alphabets varies from planet to planet! Excited by this, they decide to have some fun by arranging the alphabets in different patterns.

They want to arrange the letters of the alphabet in a grid with **N** rows and **M** columns. Each cell in the grid will have one letter. But they have a rule: for any group of **K** consecutive columns (where **N x K** equals the size of the alphabet), each letter of the alphabet must appear at least once.

For instance, if the alphabet size is **9** and letters are - **'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'**. For **N = 3** and **M = 7**, here is a valid example of arrangement:

E	I	G	A	I	H	C
C	B	F	C	D	G	E
A	D	H	E	B	F	A

If you select any three consecutive columns you'll be able to find all 9 letters there.

Help Rik and Vik figure out how many different ways they can arrange the alphabet following this rule.

Input

Input starts with an integer **T (T ≤ 10000)** - the number of test cases. Each of the following **T** line consists of three integers **S (1 ≤ S ≤ 10⁶)**, **N (1 ≤ N ≤ 10⁶),** and **M (1 ≤ M ≤ 10⁹)**, where **S** represents the number of letters in the alphabet, **N** represents the number of rows in the grid, and **M** represents the number of columns.

Input file is large. Please be sure to use faster input/output methods.

Note that the input will follow these restrictions -
- **N ≤ S**
- **N x M ≥ S**
- **S is divisible by N**

Output

For each test case, output a single integer representing the number of different arrangements possible. Since the answer may be large, output the answer modulo **10⁹ + 7**.

Sample Input

Sample Input	Output for Sample Input
2 3 3 3 4 2 4	216 96

Problem K

Sort and &

You are given a permutation of size **N**. You want to sort it. In one operation you can swap the **i**th and the **j**th elements (**1 ≤ i, j ≤ N**) and it will cost you **i&j** (bitwise **AND** of i and j). Total sum of sorting the permutation is the sum of the cost of all the operations you will make. You can't make more than **3*N** operations. Print the minimum cost to sort the permutation and also print one sequence of swap operations which will sort the permutation in the minimum cost.

Input

The first line will contain a single integer **T (1 ≤ T ≤ 10⁴)**. Each test case will start with one integer **N (1 ≤ N ≤ 10⁴)**, the length of the permutation. The next line will contain **N** integers **a₁, a₂, ... , aₙ (1 ≤ aᵢ ≤ N)**. It is guaranteed that, in any input file ΣN ≤ 10⁵ across all the test cases.

Input file is large. Please be sure to use faster input/output methods.

Output

For each test case, the first line of the output should contain the minimum cost to sort the array. The next line should contain the number of required operations. Then each next line should contain two integers describing the indices being swapped. Please see the sample for details.

Sample Input

```
1
4
3 1 4 2
```

Output for Sample Input

```
0
3
1 4
1 2
3 4
```

Notes

Permutation of size **N** is an array of size **N** where each number from **1…N** appears exactly once.

i&j denotes bitwise **AND** between integer **i** and integer **j**.