UP ACM

Algolympics 2015

Contents

Problem A
All About The Base
Time Limit: 4 seconds

Vladonis works the day shift at a local supermarket. His favorite past time is to stack the soup cans in a triangle shape. For example, if the base of the triangle contains three cans, the row above it must contain two and then one. This means that any row (aside from the lowest one) contains one less can than the row below it. This is important because he likes to put all the right cans in all the right places.

Vladiana works the night shift at the same supermarket. Her favorite past time is to stack the soup cans in a square shape. This means that if the base row contains three cans, then there must be three rows of three cans each.

This presents a slight problem. At the end of the day shift, the soup cans are stacked in a triangular shape and Vladiana takes over. She would rearrange all the cans to form a square shape at the beginning of her shift. However, she gets enraged whenever the cans used for the triangle cannot all be arranged in a square. One time, she was so enraged that she even burned every inch of the arrangement from the bottom to the top. She wasn't fired because she's the owner's daughter. Just saying.

Hence, Vladonis must make sure that the number of cans he uses for his triangular arrangement can all also be used for a square arrangement during the night shift. On his first day of work, he uses a single can as an arrangement. Vladiana is okay with this since technically, she can arrange it into a 1×1 square. On the subsequent days, he must add the least number of cans needed such that he can arrange it into a triangle and Vladiana can rearrange them into a square.

Input

Each line of input will contain a single non–negative integer $d \leq 10^6$.

Terminate the program if $d = 0$. Do not handle this case.

Output

Since this question is all about the base, you must solve for the base b of the triangular arrangement on the d^{th} day. Someone might tell you to not worry about the size, but we should do so because the value may overflow. Hence, we will ask you to output b modulo $10^7 + 6699$.

Sample Input	Sample Output
1 2 3 69 0	1 8 49 6906679

Problem B
Make Gawa This Program
Time Limit: 2 seconds

ZOMGG!!! My dad's going to take me to the One Direction concert! Finally! But, he has one condition. I need to be able to speak Tagalog sentences properly to be able to talk to people properly. W. T. F!!!!! Why Tagalog, Father??? So, anyway, I'm going to need some help. I know some Tagalog words but I usually fail in speaking proper Tagalog sentences. Please help!!!!!!!

Input

Each line of input will consist of a string with at most 1000 characters with the following format:

```
SUBJ + [space] + MAKE + [space] + VERB + " the " + NOUN + PUNC
```

where

```
SUBJ = "I" | "You" | "He" | "She" | "We" | "They"
MAKE = "will make" | "make" | "makes" | "made"
PUNC = "." | "!"
```

with `VERB` and `NOUN` being non-empty strings consisting only of letters A–Z and a–z.

Moreover, it is guaranteed that `VERB` has **at least one** vowel. Vowels are A, E, I, O, U, a, e, i, o, u.

Output

You must output a string of the following format:

```
PANDIWA + [space] + SIMUNO + " ang " + NOUN + PUNC
```

where `SIMUNO` depends on the value of `SUBJ` as follows:

SUBJ	SIMUNO
I	ko
You	mo
He	niya
She	niya
We	natin
They	nila

and `PANDIWA` depends on `MAKE` as follows:

<table>
<tr><th>MAKE</th><th>PANDIWA</th></tr>
<tr><td>will make</td><td>Repeat the first syllable of VERB. If the last vowel of VERB is “O” or “o”, replace it with “U” or “u”, respectively. If the last letter is a vowel, add “hin” at the end. If the last letter is a consonant, add “in” at the end.</td></tr>
<tr><td>made</td><td>Add the string “in” before the first vowel.</td></tr>
<tr><td>make</td><td rowspan="2">Repeat the first syllable then add the string “in” before the first vowel.</td></tr>
<tr><td>makes</td></tr>
</table>

and `NOUN`, `PUNC` are the same value as in the input. Moreover, the *first syllable* is defined to be everything before the first vowel, plus that vowel. Furthermore, if the first letter of the result is not capitalized, capitalize it.

Sample Input	**Sample Output**
He made pangako the tiket. You make sulat the program. I will make yakap the artista.	Pinangako niya ang tiket. Sinusulat mo ang program. Yayakapin ko ang artista.

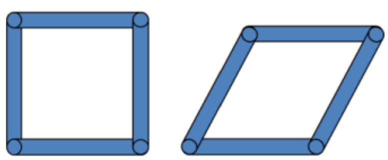
Problem C
Rigid Trusses
Time Limit: 3 seconds

What's common between the structures of bridges, billboard frames, transmission towers, and roof supports? An engineer would say, "Trusses!" One example of a truss design is a footbridge in Commonwealth (see Fig. 1).

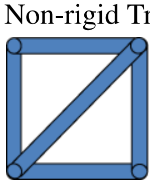
Figure 1: A truss used in a footbridge

For the purposes of this problem, let's define a truss to be any set of metal bars connected by bolts such that the whole structure behaves as a single object.

Let's start with a simple structure consisting of 2 horizontal bars and 2 vertical bars, bolted at the ends (see Figure 2a). We can see that our structure qualifies to be a truss, but it is not rigid! To make it rigid, we can bolt a diagonal bar on the opposite vertices (see Figure 2b).



(a) Non-rigid Truss



(b) Rigid Truss

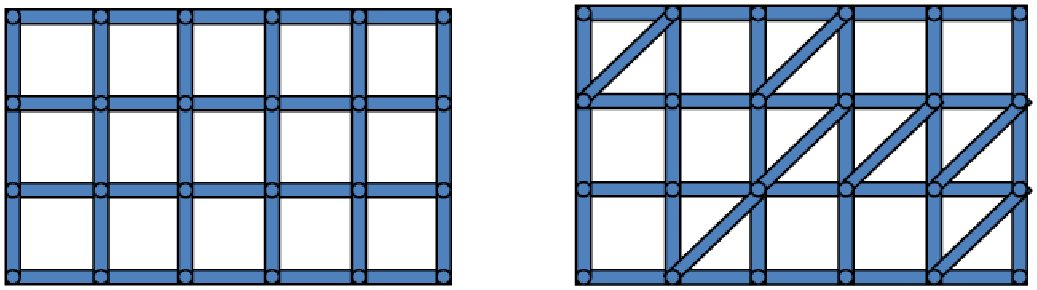
Figure 2: Trusses

To make things a bit harder, consider a grid of metal bars arranged in an $R \times C$ lattice. An example when $R = 3$ and $C = 5$ is shown in Figure 3a. In this figure, a total of $2RC + R + C = 38$ bars were used. Again, this is not rigid.

Using a number of diagonal bars strategically bolted across the structure, it can now be made rigid (see Fig. 3b). All the diagonal bars are identical, and all of them must be placed diagonally upward, connecting two opposite but adjacent bolts. There are 7 diagonal bars used in Fig. 3b.

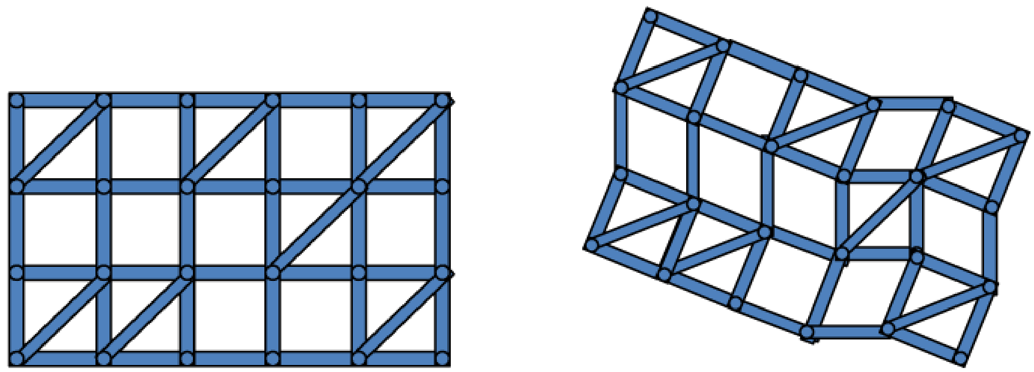
Generally, the placement and the number of diagonal bars used to ensure rigidity are important!

For example, the placement of diagonal bars in Figure 4a does not make our current rectangular truss rigid. Figure 4b shows why this is so.



(a) Rectangular Truss ($R = 3$, $C = 5$) (b) Rigid Rectangular Truss ($R = 3$, $C = 5$)

Figure 3: Rectangular Trusses



(a) Non-rigid rectangular Truss (b) Skewed non-rigid Truss

Figure 4: Non-rigid Trusses

Given a lattice grid of metal bars, possibly braced with diagonal bars, can you determine if it is rigid?

Input

The first line of input is a single number, T, denoting the number of test cases.

The first line of each test case contains two integers, denoting R and C. The next $2R+1$ lines contain a structure made of only the characters '/', '|', '+', '−', and ' ' (space), written in $2C+1$ columns. See sample test cases for clarity. There are no excessive whitespaces or extra lines in the input. At the very least, the structure is always a valid rectangular truss like Figure 3a.

Output

For each test case, output a line in the format: "`Case #`X`:` Y". X denotes the test case number, starting from 1. Y is 1 if the structure is rigid and 0 otherwise.

Constraints

$1 \leq T \leq 100$
$1 \leq R, C \leq 10$

Sample Input

```
2
1 1
+-+
|/|
+-+
3 5
+-+-+-+-+-+
|/| |/| |/|
+-+-+-+-+-+
| | | |/| |
+-+-+-+-+-+
|/|/| | |/|
+-+-+-+-+-+
```

Sample Output

```
Case #1: 1
Case #2: 0
```

Problem D
Slicing Number Cakes
Time Limit: 3 seconds

You are given a positive integer, N. Imagine that this integer is a cake roll and let's call it a number cake!

As a cake roll, it can be sliced anywhere between the digits. For example, if $N = 2357$, then after 1 slice, we can get any of the following: $\{2,357\}$, $\{23,57\}$, or $\{235,7\}$. In effect, if N has D digits, then you can slice it in $(D-1)$ different positions. Notice that the order of the digits is preserved.

Slices	Satisfaction
$\{2,3,57\}$	$2+3+57 =$ **62**
$\{2,35,7\}$	$2+35+7 =$ **44**
$\{23,5,7\}$	$23+5+7 =$ **35**
Max satisfaction: **62**	

If $N = 2357$ is sliced twice, then we can get any of the following: $\{2,3,57\}$, $\{2,35,7\}$, or $\{23,5,7\}$. Mogu, the cake monster, wants to eat a number cake! After making S number of slices on the number cake, N, Mogu's satisfaction is equal to the sum of the cake slices he eats. The table on the left shows the maximum satisfaction for $N = 2357$ and $S = 2$.

Therefore, the best way to slice $N = 2357$ is $\{2,3,57\}$.

Note that Mogu can eat cake slices such as "001", "0", or "00". So for $N = 10010$ and $S = 2$, the maximum satisfaction is achieved by slicing it this way: $\{100,1,0\}$. Given N and S, can you determine Mogu's maximum possible satisfaction by making exactly S slices on N?

Input

The first line of the input contains a single number, T, indicating the number of test cases. Each of the next T lines contains two integers, N followed by S, separated by a single space character. N will not contain leading zeros.

Output

For each test case, output Mogu's maximum possible satisfaction on a single line.

Constraints

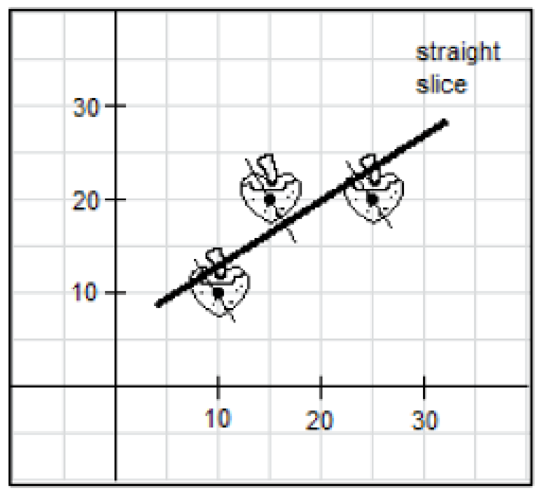
$1 \leq T \leq 500$
$10 \leq N \leq 10^{19}$
$0 \leq S <$ (number of digits of N)
N will not contain leading zeros.

Sample Input	Sample Output
3 2357 2 1234321 4 10010 2	62 439 101

Problem E
N-fruit combo
Time Limit: 3 seconds

Fruit Ninja is a popular game among mobile applications. Players slice through fruits, as they appear on-screen, using a swiping gesture with their fingers. Fruits of different shapes and sizes can appear: mangoes, bananas, apples, plums, oranges, watermelons, etc. The game also gives missions the players must accomplish. One mission reads: Get an N-fruit combo with N strawberries. Strawberries are hardest to slice because they're small. But you accepted this challenge!

The position of the strawberries' centers can be listed as (X,Y) coordinates in a screen where the lower left corner is at $(0,0)$ and the upper right corner is at $(200,150)$. We define a straight slice gesture to be a straight line on this plane, even extending beyond the screen.



A strawberry, then, is considered "sliced" if and only if the shortest distance from its center to the straight slice is less than or equal to 3 units, even if this distance is measured beyond the screen.

Given N strawberries, can you determine if it is possible to slice ALL of them in ONE straight slice? This feat is an N-fruit combo! It is assumed that straight slices are done instantaneously, regardless of length.

Input

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

The first line of each test case contains a single integer, N, denoting the number of strawberries. Each of the next N lines contains two integers, X and Y. These denote the position of a strawberry center. No two centers in a test case are the same.

Output

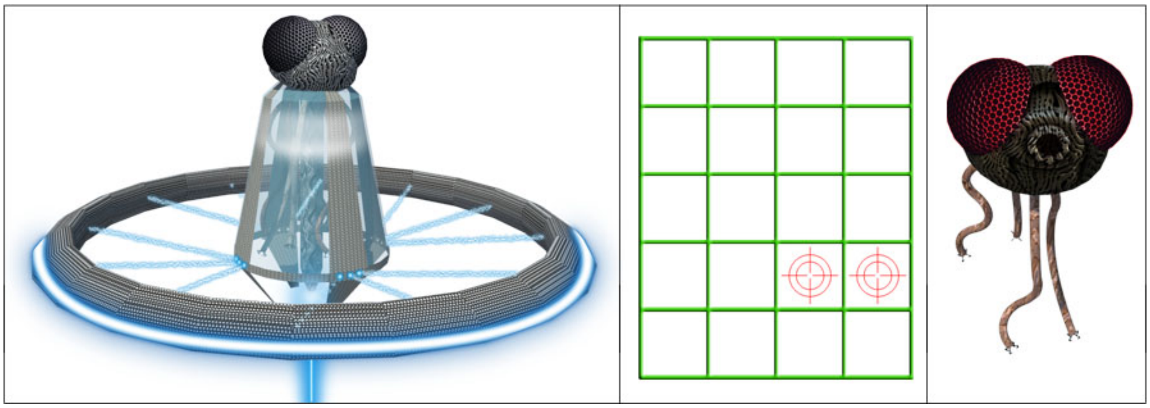
For each test case, output a line in the format "`Case #`*A*`:` *B*", where *A* denote the case number starting from 1, and *B* is 1 if an *N*-fruit combo is possible and 0 otherwise.

Constraints

$1 \leq T \leq 100$
$3 \leq N \leq 10$
$0 \leq X \leq 200$
$0 \leq Y \leq 150$
No two centers in a test case are the same.

Sample Input	Sample Output
1 3 10 10 15 20 25 20	Case #1: 1

Problem F
Alien Defense Deux
Time Limit: 2 seconds



The aliens, the Sublinears, are attacking again! The grand land of Algolympia needs your aid, brave hero, to defend against this dastardly threat. This time the Sublinears are attacking in a conservative manner, by performing devastating orbital laser strikes from space using their Quadra spacecraft. Through their concerted efforts, the Algolympians have managed to design a weapon, the Exponentia, to defend against these laser strikes. However, the Exponentia takes an entire month to deploy, and so the Algolympians need to deploy the Exponentia in a secure area safe from laser strikes before they can strike back. Through their expertise in subterfuge, the Algolympians have managed to determine the areas the Sublinears plan to attack with laser strikes. However, the Algolympians need *your* help to determine how many possible locations they can deploy the Exponentia in.

The Exponentia is a square object, of magically varying side-length. The Algolympians have helpfully partitioned their land into an $M \times N$ grid. Your goal is to find how many square-shaped areas in their grand land are viable locations for the Exponentia (that is, that are free from laser strikes).

The sides of the square-shaped areas must be parallel to the sides of the $M \times N$ land, and must coincide with the grid lines.

Input

The first line contains one integer T, the number of test cases in the input. The following lines contain the T test cases.

Each test case begins with a single line containing two integers, M and N, separated by a single space, which are the number of rows in the grid, and the number of columns in the grid, respectively.

The next M lines contain the grid. Each of the M lines will contain exactly N characters, either "`#`" (denoting an area that will be hit by a laser strike) or "`.`" (denoting an area that will not be hit by a laser strike).

Output

For each test case, output a set of lines in the following format:

- The first line should contain the phrase "`Case` t`:`", where t is the test case number (where the first test case has the test case number 1).
- If there are no square-shaped areas that are free from laser strikes, the next line should contain the phrase "`Wala`", and no other lines should follow for this test case.
- Otherwise, for each size of laser-strike-free square-shaped areas (in increasing order of side-length), print the string: "s`*`s`:` u", where s is the side-length of the square-shaped area, and u is the number of areas with the given size that are laser-strike-free.

Constraints

$1 \leq T \leq 12$
$1 \leq M \leq 900$
$1 \leq N \leq 900$

Sample Input	**Sample Output**
``3 3 5 .... 5 4 ....## .... 2 2 ## ##```	```Case 1: 1*1: 15 2*2: 8 3*3: 3 Case 2: 1*1: 18 2*2: 8 3*3: 2 Case 3: Wala```

Problem G
For Science™!
Time Limit: 12 seconds

Your friend, Elrond Trustworthy Hubble, has just started a new company called the Sciencology Club. The Sciencology Club provides Science™ to the masses.

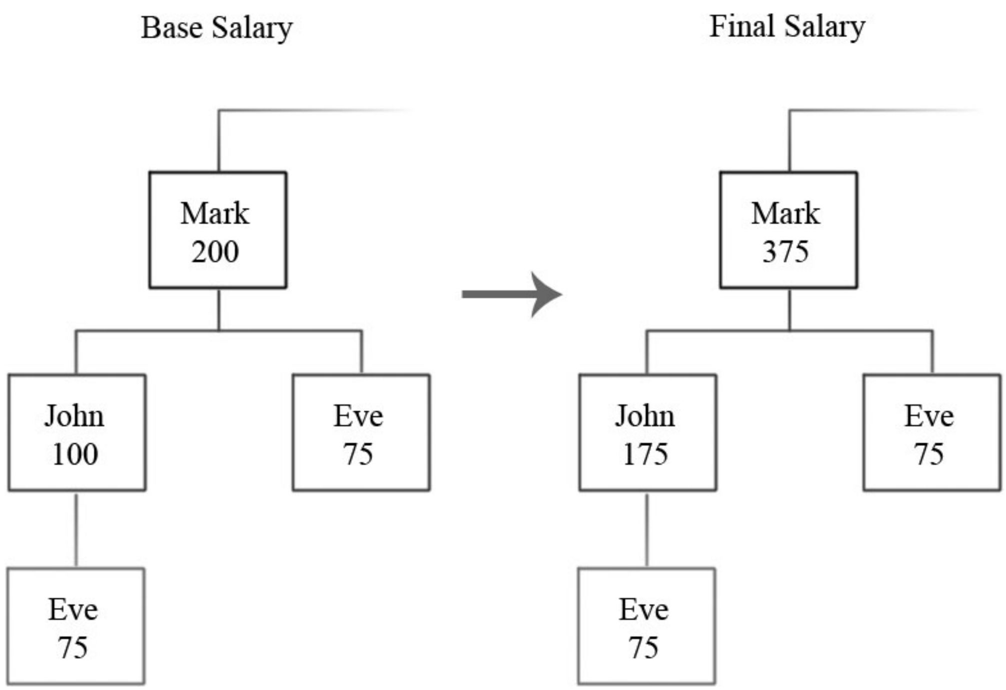
Because the Sciencology Club is still a new company, Elrond decided to grow it using what he calls the *Peer-Amid Scheme*. The Peer-Amid Scheme works by surrounding yourself with peers and getting them to join the Sciencology Club, that is, being amid peers. Each new recruit will be under the member that recruited them, thus forming a tree hierarchy. Of course, Elrond is the root of this tree since he started the company himself.

A member's monthly salary is computed as the sum of his base salary plus the base salaries of all of his recruits, plus his recruits' recruits, plus his recruits' recruits' recruits, and so forth. In other words, a member's salary is the sum of the base salaries of his subtree (including his own).

Every member is assigned a base salary based on only his name. Which is totally okay because Science™ has proven this to be reliable. Furthermore, no two people with different names will ever get the same base salary because Science™ demands it.

There is only one problem though. Because Science™ has shown that if more than one person with the same name are under one person's subtree, it is unscientific to give that person both of their base salaries. Thus, that person is only entitled to one of their base salaries.

As an example, Mark recruits John Smith and Eve Ville. Then John recruits Eve Ning. Mark gets a base salary of 200. John gets 100. And each Eve gets 75. Mark now has two people named Eve in his subtree, so he's only entitled to one of their salaries. So Mark gets a final salary of 375. John gets a final salary of 175.



Since Elrond is the provider of Science™ to all mankind, nothing is suspicious about any of this. Again, Trustworthy is his middle name.

Your job, as one of the members of the Sciencology Club, is to write a program that computes everyone's final salaries given the base salary structure of the company. Do it for Science™!

Input

The first line contains one integer T, the number of test cases in the input. The following lines contain the T test cases.

Each test case begins with a single line containing two integers, N and E, separated by a single space, which are the number of Sciencology Club members and Elrond's number respectively. Members are assigned a number from 1 to N.

The next line contains N integers. The i^{th} number denotes the base salary of the i^{th} person. Elrond is person number E.

The following $N-1$ lines contain two numbers each, u and v denoting that member u recruited member v, or vice versa.

It is guaranteed that the given data is valid, and that everyone is under Elrond.

Output

For each test case, output one line containing N integers separated by single spaces. The i^{th} number must be the final salary for the i^{th} person.

Constraints

$1 \leq T \leq 10$
$1 \leq N \leq 100{,}000$
Base salaries are from 1 to 100,000,000

Warning: The input file is large (approximately 20Mb)! Please use fast I/O methods (e.g. `BufferedReader` in Java, `printf`/`scanf` in C/C++)

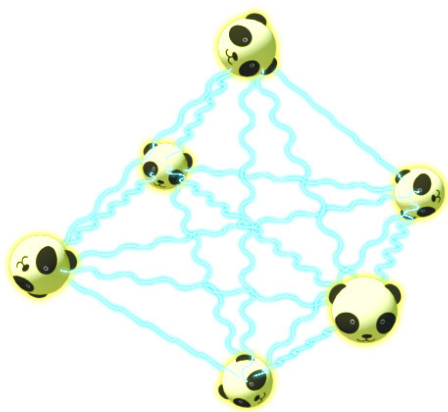
Sample Input

```
3
4 1
200 100 75 75
1 2
2 3
1 4
3 2
11 24 11
1 2
2 3
5 2
100 200 100 200 200
1 2
2 4
4 3
4 5
```

Sample Output

```
375 175 75 75
11 35 11
100 300 100 300 200
```

Problem H
Algols for Algolympia
Time Limit: 20 seconds



The people of the city of Algolympia is nearing extinction, having almost exhausted their oil supply for electricity since a programmer typed in "maximum" instead of "minimum".

Luckily, you are a Genius particle physicist AND a Genius astronaut AND a Genius programmer! After narrowly escaping death by crashing onto the newly discovered planet Algolympus, you encountered a sea of floating firefly-like particles called **algols** (yes, yes, the ones on the left are glowing pandas, but you are an imaginative person :D). After observing them, you concluded that these particles could be the key to solving the energy crisis of Algolympia!

Analyzing the particles, you noticed the following:

1. The fundamental type of particle in the planet Algolympus is the **algol**.
2. The amount of energy of an isolated algol varies with every algol.
3. An algol can pair off with another algol to form an **algolic bond**. The amount of energy of an algolic bond is the sum of the energy of the two algols in it.
4. If an algol algolically bonds with two or more other algols, then those algols also algolically bond with each other.
5. A group of algols all connected with algolic bonds is called an **algolymer**.

Surprisingly, an algolymer is an amazing source of energy. The total energy that can be extracted from an algolymer is the product of the energies of the algolic bonds in it.

In your hurry to go back to Algolympia, you were only able to obtain an algolymer with N algols. You have measured the amount of energy of each algol in the algolymer. Before extracting the energy in it, you would like to first compute the amount of energy you can extract from it. However, being a genius programmer, you are very lazy and only want the answer modulo 1,000,000,007.

Please hurry and get the job done, so that you can resume playing Offense of the Moderns!

Input

The first line contains one integer T, the number of test cases in the input. The following lines contain the T test cases.

Each test case consists of two lines. The first line contains a single integer, N, denoting the number of algols. The second line contains N space-separated integers $A_1, A_2, \ldots, A_N$ where A_i is the amount of energy in the i^{th} algol.

Output

For each test case, output one line containing a single integer, which is the amount of energy that can be extracted from the algolymer, modulo 1,000,000,007.

Constraints

$1 \le T \le 5$
$2 \le N \le 200{,}000$
$1 \le A_i \le 200{,}000$

Sample Input	Sample Output
2 4 1 1 2 4 3 499 500 502	2700 1998991