Round 1 - Programming , DigiCred Technologies Pvt. Ltd.

Rules:

- All the questions are compulsory
- You can use any programming language
- Plagiarism is not allowed (ie - don't copy your answers from anywhere) document each line of code
- You have 48 hours to submit your answers

Steps to submit your assignments:

- Create a github account if you don't have one.
- Fork the following repo: https://github.com/DigiCred/upes-assignments
- Create a folder your name.
- Create a separate file for each question/answer in that folder (name the file by question number)
- Send a pull request to the the master remote.

Question 1: Uber has introduced UberPool to make sure there are fewer cars on the road, which means less traffic, faster travel times and cheaper rides. Here is how UberPool works: it matches different riders heading in the same direction, giving a driver two pick-up and two drop-off locations on the same trip. UberPool is limited to two riders per trip, which means that sometimes a driver has to choose which request to accept.

Consider a city represented as integer points on the Cartesian plane, with roads parallel to the axes. There's a driver who picked up a passenger at point A and is heading to point B with them, taking one of the shortest possible routes. When the driver reaches point C, she receives two more requests: one from a rider at point X, another from a rider at point Y. Both riders are also going to point B.

Your task is to find out which additional rider should be picked up (if any), taking into account that the final length of the trip can't be more than two times longer than the length of the trip with no extra riders. If both requests meet this condition, accept the one with the shortest route. If the lengths of the routes are the same, accept the first request.

Example

For A = [0, 0], B = [3, 3], C = [3, 1], X = [5, 0] and Y = [2, 2] the answer should be 2.

The initial distance is 6, the route with the passenger at point X is of length 12, the route with the passenger at point Y is of length 8. Both pick-ups are possible; however, it's better to pick up the passenger at Y, since the route with him is shorter. See the picture below for details.

[input] array.integer A

The departure point, represented as array of two integers - x and y coordinates.
[input] array.integer B

The destination point.
[input] array.integer C

The point at which two new requests came in.
[input] array.integer X

Coordinates of the first request.
[input] array.integer Y

Coordinates of the second request.
[output] integer

Return 1 if the passenger at point X should be picked up, 2 if the passenger at point Y should be picked up, or -1 if it's impossible to accept any of the additional requests.

Question 2 : Consider a lonely Uber driver working in a big city.
He has N requests that he has to grant exactly in the given order.
The route is already mapped out for him and we can assume that he picks up a new passenger at exactly the same spot where he drops off the previous one.

Each request consists of two numbers:
1. the time by which the rider will be ready to go
2. the time by which the rider will cancel the request if the car is not there.
This means that if the driver arrives too early, he'll have to wait to pick up the passenger. If he arrives too late, the ride will be canceled.

Knowing the requests' details and the time it takes to travel from one to the next, determine the minimum amount of time the given route will take, or return -1 if it is impossible to service all the requests.

Note that as you're given the driver's daily plan, the answer is smaller than the number of seconds in a day (24 * 60 * 60 = 86400).

[input] array.integer travelTimes
An array on N - 1 positive integers; travelTimes[i] defines the travel time (in seconds) between the ith and the (i+1)th request.

[input] array.integer readyTimes
The times at which riders will be ready for pick up as an array of N integers in the range [0, 86400 - 1].

[input] array.integer cancelTimes
The times at which riders will cancel the ride if the driver is not there as an array of N integers in the range [0, 86400 - 1].

It is guaranteed that cancelTimes[i] > readyTimes[i] for each i in range [0, N - 1].
[output] integer

The minimum amount of time the given route will take, or -1 if it is impossible to service all the requests.

Question 3: Vidhi went to a magic show last week where she was astounded by a magic trick performed by the great Mandwarf, the brown. His trick was as follows :

- Ask a volunteer from the audience to write down a list L of N integers.
- Ask another volunteer from the audience to provide three integers A, B, C
- Ask another volunteer from the audience to provide N length string called S where each letter is either 'R', 'A' or 'M'
- Close his eyes for a split second and give the output of The Ancient Algorithm on this input.

We all know that The Ancient Algorithm is as follows :
for i from 1 to N do

if i^{th} letter of S is 'R'
reverse L[i...N]
else if i^{th} letter of S is 'A'
add A to all numbers of L[i..N].
else if i^{th} letter of S is 'M'
multiply B to all numbers of L[i..N].

for all number in L[i..N], module them by C.

announce L[i] out loud

end

Vidhi's boyfriend got jealous when he saw her getting impressed by Mandwarf, the brown's wisdom. He wants to learn the trick to gain her undivided admiration. How about you help him?

Input:

First line contains a single integer T, denoting the number of test cases. Then follow

Ttest case scenarios. Each test case begins with an integer N, the size of the list L. Then in next line, you'd find N space separated integers - the list L itself. In next line, there'd be three space separated integers A, B, C followed by string S in the next line.

Output:

For each test case you've to output N space separated integers - the numbers announced by Mandwarf, the brown.

Example

Input:

2

3

1 1 1

2 3 1000

ARM

4

1 2 3 4

0 1 1000

AMAM

Output:

3 3 9

1 2 3 4

Question 4: Devasena was the princess of an unknown kingdom (we'll all know which one, in Baahubali 2 ;)), and her father arranged for a Swayamvara to get her married. He gave all of them a question, and the question was so hard that nobody was able to answer it (Yes you guessed it right, both Baahubali and Bhallaladeva were not present there for the Swayamvara). We all know that Amarendra Baahubali married her later and as a consequence of so many things, she was imprisoned for 25 years. Who knows, if someone else had answered the question that day, then she would have got married to him and things could have been different. Phew! But the bad part, we wouldn't have had the story of Baahubali. :)

Although everything is history now, recently archaeologists discovered the secret question that was asked at the Swayamvara, and you think - "Well, it's such a simple question. I could use a computer to solve it!". The question goes as follows:

You are given N integers (not necessarily distinct) => A_1, A_2, A_3, ..., A_N. There are 2^Npossible subsets (including the empty subset). The GCD of a subset is defined as the greatest common divisor of all the integers in that subset. You need to find the product of the GCDs of all the 2^N possible subsets you can construct from A. Since the answer can be large, you need to output the answer modulo 1000000007. Do you think you can solve this question?

Input

The first line of input consists of a single integer T denoting the number of test cases. The description of T test cases follow. The first line of each test case consists of a single integer N. The second line of each test case consists of N space separated integers A_1, A_2, ..., A_N

Output

For each test case, output an single integer on a separate line denoting the answer for that test case. Note that you need to output all the values modulo 1000000007 ($10^9 + 7$).

Example

Input:

3

1

1

2

1 2

3

1 2 2

Output:

1

2

8