9: Hikers' Hitch

Level: Average
Time limit: 1.5 seconds

Our hitchhiking committee has announced that this year's hitchhiking weekend is going to be personalized: Each participant will receive their own location to travel to. The committee has published a list of all possible locations that could be chosen, and on the day of the event each participant will be appointed one location. This means that not all locations from the list are selected, and that certain locations may be appointed to multiple people. To make things easier, the locations are ordered by their distance from the start, and the start will be location 0.

To discourage cheating, a point system will be used, where the aim is to gain as few points as possible. Points will be added to a participant's score if the car they are in stops at a spot before their appointed destination, or for every location they reach after passing (and not stopping at) their appointed destination (overshooting a location is fine, the committee has a car of shame to help you get back). At the end, every participant's score will be added up to form a global score. You've heard that a large score is bad news for everyone, so you decide to devise a strategy together with the others.

Your team prepares for the trip by preprogramming a route into one of those newfangled self-driving taxis. You know that on the day itself, this route can't be changed, so you make the route pass through each possible destination in the given order. Depending on the ID's of everyone's destinations, you wonder what the locations are in which the car needs to stop to minimize the cumulative score?

Input

The first line has an integer representing the number of test-cases C ($1 \leq C \leq 100$). For each test case, there are two lines with on each:

An integer L, the number of possible locations along the route ($1 \leq L \leq 1.500$).

L space separated integers P_i, one for each location, where each integer corresponds the number of participants that need to get out at that location ($0 \leq P_i \leq 1.500$).

Output

The output must consist of one line per test-case, the lowest possible score that can be achieved by stopping the car at the correct locations.

Sample input 1

```
3
5
0 3 0 0 7
5
0 0 3 0 7
10
3 1 4 1 5 9 2 6 5 3
```

Sample output 1

```
7
6
67
```