Vasplay in the subway of tokio

Assignment

Just like in many cities with a metro, in Tokyo you have to scan your card when entering and leaving: the amount that is deducted from your prepaid card only depends on your start and end station, not on which stations you stopped on the way. You can indeed follow any route, for example to visit your favorite sushi stand, as long as you do not leave the station. The prepaid cards in Tokyo are nameless, and, in a real Belgian way, we immediately came up with the idea for an app that enables a group of people to pay a little less, namely by exchanging cards on the go.
Here's an extreme case: Kenji has to go from Shibuya station to Asakusa station in the morning; that costs 4 units. At the same time, Ronin has to go from Asakusa to Shibuya, also 4 units. Together they pay 8 units. But if they meet on the road, say in Shimbashi, and then switch their cards there, each card arrives where it left, and the total cost is 0, or the total profit is 8. At night they travel in reverse, switch again card and everyone is happy, except for the metro operators. The app that you are going to develop has a registration module for a specific time slot: a number of people enter their start and end station in it. The app then calculates how to change the cards and displays the total winnings. Of course it should not be the case that some individuals suffer a loss: see the entry later.

Stations are given as a positive number, from 1 to *N*, the number of stations. The price between two stations comes in the form of a matrix, indexed by the stations. You get this matrix completely: it is always symmetric with zeros on the diagonal and elsewhere only strictly positive integers. As an output, we expect the maximum total profit that can be made by exchanging cards without anyone incurring a loss.

Input

The first line represents the number of test cases. Per test case follows

- a line with the number of stations N; $N > 1$
- the cost matrix: N rules with each time N positive numbers separated by a blank; only on the diagonal are zeros
- a line with the number of people p that registered on the app: those persons are numbered from 1 to p; $p > 0$
- then one line comes with the p departure stations of the p persons, separated by a blank
- then one line comes with the p terminals separated by a blank

sample input

```
2
5
0 1 2 3 4
1 0 2 3 4
2 2 0 4 1
3 3 4 0 1
4 4 1 1 0
3
1  2  5
5  3  1
3
0  4  6
4  0  4
6  4  0
2
1  2
2  3
```

Export

As output we expect per test case one line with the sequence number of the test case, a blank, and then the maximum total profit that can be made without anyone incurring a loss.

sample output

```
1  8
2  0
```

In the first test case, person 1 and 3 can exchange their cards and gain 4 units each: person 2 does not switch, because that does not make it better overall.
In the second test case, the total cost without changing cards is 8. By changing the cards, the total cost becomes 6 and that is better, but the first person's card is debited with 6 instead of 4 in this scenario and that is what he wants of course not. No profit can be made here.