

# Problem J

## Travelling Tom

Tom is a used popsicle salesman. His recent sales at his stand on Gløshaugen haven't been very good, so he's decided that he wants to travel the world selling his merchandise. He has already compiled a list of cities to visit, and have also gathered the costs of the plane trips between them. Now he just have to figure out if he has enough money for the trip. He has of course heard that this travelling salesman problem is really really hard, so the task is down to the smartest people he knows. That's you (if you don't feel very smart at the moment, keep in mind that Tom doesn't know that many people). You need to write a program that computes the lowest cost of Tom visiting all these cities in the given order.



### Input specifications

The first line of input contains a single number  $T$ , the number of test cases to follow. Each test case begins with a line containing a single number,  $N$ , the number of cities Tom will visit. The second line of each test case contains  $N$  numbers  $a_i$ , the order in which Tom wants to visit the  $N$  cities. He starts his trip in the first city described, and will travel back there once he has visited the last one. Then follow  $N$  lines, each containing  $N$  integers, describing the cost  $c_{ij}$  of travelling from city  $i$  to each of the  $N$  cities.

### Output specifications

For each test case, output a line containing a single number, the lowest possible cost of visiting the cities. If it is not possible to visit all the cities, output **impossible** instead.

## Notes and Constraints

- $0 < T \leq 100$
- $0 < N \leq 200$
- $0 \leq a_i < N$
- $-1 \leq c_{ij} \leq 10000$
- The list of cities visited will always be a permutation of the numbers from 0 to  $N - 1$ , inclusive.
- The cost of travelling from a city to itself is always 0.
- If and only if there is no plane going from city  $i$  to city  $j$ , then the  $j$ -th number of the  $i$ -th line of costs will be -1.
- Note that Tom is returning to the starting city at the end of the tour.
- Cities can be visited several times, but will only count when (also) in the correct relative order. For the order 1 0 2, for instance, 1 2 0 2 1 is a valid tour, while 1 2 0 1 is not.

## Sample input

```
2
3
0 2 1
0 1 2
1 0 1
1 3 0
2
0 1
0 -1
1 0
```

## Output for sample input

```
5
impossible
```