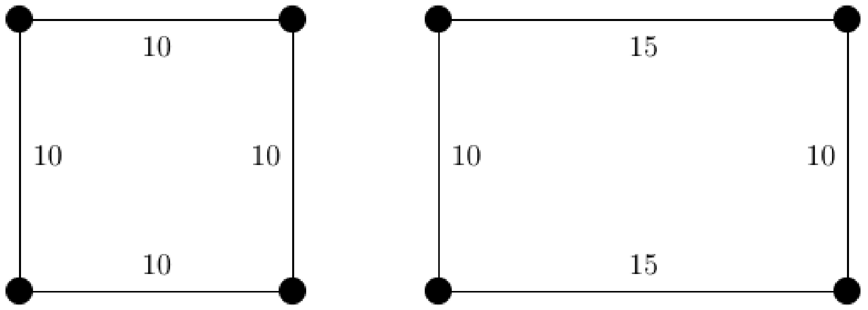
ACM-ICPC Live Archive

6879 Inspectors

Yuri is a rich business owner. His company has factories in many cities across Russia, all of which need regular, monthly inspections. Now Yuri not only has a large portfolio, but a large family as well and he would like to see some of his children start to learn how the business is run. He figures the best way to do this is to have them perform the factory inspections, but he has two concerns. First, he wants each child to be assigned at least two different factories — that way they will be able to compare different management styles at the factories and get a better idea of what works and what doesn't work. This of course requires them to travel between the factories, which brings up Yuri's second problem: while he won't be paying his children anything (their weekly allowance is more than adequate), he's worried about the total of all the travel costs, and would like it to be as small as possible. Each child is expected to inspect each of their assigned factories once a month, always ending back at the factory they started at, so to minimize cost he figures he need only assign the factories to minimize the total distance of these factory tours. The more he thinks about this, the more important saving money becomes. He's willing to use as few as only one of his children if that's the cheapest way to inspect all the factories. For example, if Yuri had four factories placed as in the figure on the left below, he could employ just one child, who could visit each factory for a monthly distance of 40 (assume here that the unlisted distances between factories are all > 10). However, if his factories were located as shown on the right, he would need to employ two children to obtain the monthly minimal distance of 40 (note: he could use two children to achieve the minimal distance in the first figure as well).



Since the number of factories changes over time, Yuri would like a program to determine the minimum distance he can expect for any layout of factories.

Input

The input file begins with a line containing a single integer m indicating the number of test cases. Each test case starts with an integer n indicating the number of factories, where $2 \leq n \leq 100$. The next $n-1$ lines contain the positive integer distances between the factories: line 1 contains $n-1$ values specifying the distances between factory 1 and factories 2, 3, 4, ..., n; line 2 contains $n-2$ values specifying the distances between factory 2 and factories 3, 4, ..., n; and so on. The maximum distance between any two factories is 1000. All distances between any three cities will satisfy the triangle inequality.

Output

For each test case display the minimum factory tour distance. You should always assume that Yuri has at least $n/2$ kids for each set of n factories.

Sample Input

```
2
4
10 20 10
10 20
10
4
15 20 10
10 20
15
```

Sample Output

```
Case 1: 40
Case 2: 40
```