You have been hired as a systems administrator for a large company. The company head office has n computers connected by a network of m cables. Each cable connects two different computers, and there is at most one cable connecting any given pair of computers. Each cable has a latency, measured in micro-seconds, that determines how long it takes for a message to travel along that cable. The network protocol is set up in a smart way, so that when sending a message from computer A to computer B, the message will travel along the path that has the smallest total latency, so that it arrives at B as soon as possible. The cables are bi-directional and have the same latency in both directions.

As your first order of business, you need to determine which computers are connected to each other, and what the latency is along each of the m cables. You soon discover that this is a difficult task because the building has many floors, and the cables are hidden inside walls. So here is what you decide to do. You will send a message from every computer A to every other computer B and measure the latency. This will give you $n(n-1)/2$ measurements. From this data, you will determine which computers are connected by cables, and what the latency along each cable is. You would like your model to be simple, so you want to use as few cables as possible.

Input

The first line of input gives the number of cases, N (at most 20). N test cases follow. Each one starts with a line containing n ($0 < n < 100$). The next $n-1$ lines will contain the message latency measurements. Line i will contain i integers in the range $[1, 10000]$. Integer j is the amount of time it takes to send a message from computer $i+1$ to computer j (or back).

Output

For each test case, output a line containing '`Case #x:`'. The next line should contain m — the number of cables. The next m lines should contain 3 integers each: u, v and w, meaning that there is a cable between computers u and v, and it has latency w. Lines should be sorted first by u, then by v, with $u < v$. If there are multiple answers, any one will do. If the situation is impossible, print '`Need better measurements.`' Print an empty line after each test case.

Sample Input

```
2
3
100
200 100
3
100
300 100
```

Sample Output

```
Case #1:
2
1 2 100
2 3 100

Case #2:
Need better measurements.
```