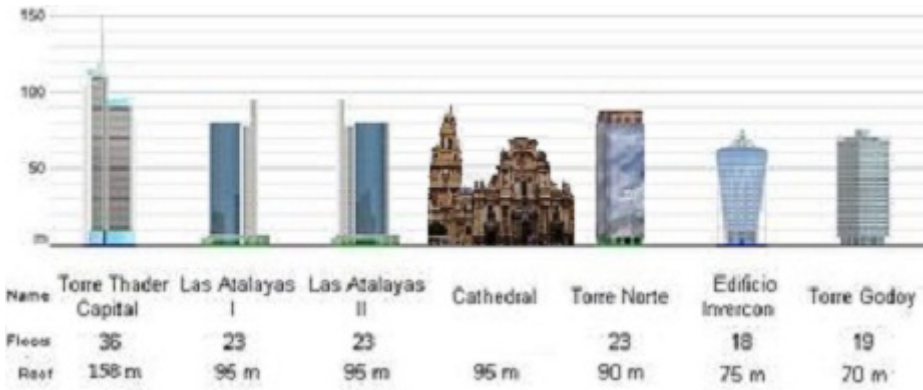


Murcia’s skyline is growing up very fast. Since the 15th century, it used to be dominated by the profile of its Baroque Cathedral. But nowadays, new skyscrapers are rising in Murcian *huerta*.

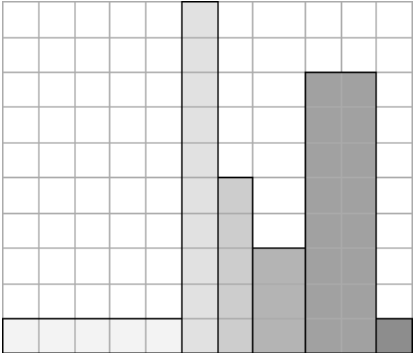


Some people say that if look at the skyline from left to right, you can observe an increasing profile; but other people say the profile is decreasing.

Looking at Murcia’s skyline from left to right, we have a series of N buildings. Each building has its own height and width. You have to discover if the skyline is increasing or decreasing.

We say the skyline is **increasing** if the longest increasing subsequence of buildings is bigger or equal than the longest decreasing subsequence of buildings; in other case, we say it is **decreasing**. A **subsequence** is a subset of the original sequence, in the same order. The **length** of a subsequence of buildings is the sum of the widths of its elements.

For example, assuming we have six buildings of heights: 10, 100, 50, 30, 80, 10; and widths: 50, 10, 10, 15, 20, 10; then we have an increasing subsequence of 3 buildings and total length 85, and a decreasing subsequence of 1 building and total length 50 (also, there is a decreasing subsequence of 4 buildings and length 45). So, in this case, we say that the skyline is increasing. You can see this example below.



Input

The first line of the input contains an integer indicating the number of test cases.

For each test case, the first line contains a single integer, N , indicating the number of buildings of the skyline. Then, there are two lines, each with N integers separated by blank spaces. The first line indicates the heights of the buildings, from left to right. The second line indicates the widths of the buildings, also from left to right.

Output

For each test case, the output should contain a line. If the skyline is increasing, the format will be:

Case i . Increasing (A). Decreasing (B).

If the skyline is decreasing, the format will be:

Case i . Decreasing (B). Increasing (A).

where i is the number of the corresponding test case (starting with 1), A is the length of the longest increasing subsequence, and B is the length of the longest decreasing subsequence.

Sample Input

```
3
6
10 100 50 30 80 10
50 10 10 15 20 10
4
30 20 20 10
20 30 40 50
3
80 80 80
15 25 20
```

Sample Output

```
Case 1. Increasing (85). Decreasing (50).
Case 2. Decreasing (110). Increasing (50).
Case 3. Increasing (25). Decreasing (25).
```