Egg Drop

Source: https://code.google.com/codejam/contest/dashboard?c=32003#s=p2

Imagine that you are in a building with **F** floors (starting at floor 1, the lowest floor), and you have a large number of identical eggs, each in its own identical protective container. For each floor in the building, you want to know whether or not an egg dropped from that floor will break. If an egg breaks when dropped from floor i, then all eggs are guaranteed to break when dropped from any floor $j \geq i$. Likewise, if an egg doesn't break when dropped from floor i, then all eggs are guaranteed to never break when dropped from any floor $j \leq i$.

We can define *Solvable(F, D, B)* to be true if and only if there exists an algorithm to determine whether or not an egg will break when dropped from any floor of a building with **F** floors, with the following restrictions: you may drop a maximum of **D** eggs (one at a time, from any floors of your choosing), and you may break a maximum of **B** eggs. You can assume you have at least **D** eggs in your possession.

Input

The first line of input gives the number of cases, 1 ≤ **N** ≤ 100. **N** test cases follow. Each case is a line formatted as:

F D B

Solvable(F, D, B) is guaranteed to be true for all input cases.

Output

For each test case, output one line containing "Case #**x**: " followed by three space-separated integers: F_{max}, D_{min}, and B_{min}. The definitions are as follows:

- F_{max} is defined as the largest value of **F'** such that *Solvable(**F'**, D, B)* is true, or -1 if this value would be greater than or equal to 2^{32} (4294967296).
  (In other words, F_{max} = -1 if and only if *Solvable(2^{32}, D, B)* is true.)
- D_{min} is defined as the smallest value of **D'** such that *Solvable(F, **D'**, B)* is true.
- B_{min} is defined as the smallest value of **B'** such that *Solvable(F, D, **B'**)* is true.

Sample Input

```
2
3 3 3
7 5 3
```

Sample Output

```
Case #1: 7 2 1
Case #2: 25 3 2
```