

Practice Problems

[A. Alien Numbers](#)

[B. Always Turn Left](#)

C. Egg Drop

[D. Shopping Plan](#)

[Questions asked](#)

Submissions

Alien Numbers

40pt Not attempted
320/432 users correct (74%)

80pt Not attempted
271/338 users correct (80%)

Always Turn Left

40pt Not attempted
108/135 users correct (80%)

80pt Not attempted
96/114 users correct (84%)

Egg Drop

40pt Not attempted
56/82 users correct (68%)

80pt Not attempted
26/53 users correct (49%)

Shopping Plan

40pt Not attempted
43/67 users correct (64%)

80pt Not attempted
16/52 users correct (31%)

Top Scores

sclo	480
jdmetz	480
lordmonsoon	480
ardiankp	480
krijgertje	480
ilyakor	400
Edu	400
Jonick	400
zibada	400
gpscale	400

Problem C. Egg Drop

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
40 points

Solve C-small

Large input
80 points

Solve C-large

Problem

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* ≥ *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* ≤ *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, **N**. **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 2³² (4294967296). (In other words, *F*_{max} = -1 if and only if *Solvable*(2³², *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.

Limits

1 ≤ **N** ≤ 100.

Small dataset

1 ≤ **F** ≤ 100,
1 ≤ **D** ≤ 100,
1 ≤ **B** ≤ 100.

Large dataset

1 ≤ **F** ≤ 2000000000,
1 ≤ **D** ≤ 2000000000,
1 ≤ **B** ≤ 2000000000.

Sample

Input	Output
2	Case #1: 7 2 1
3 3 3	Case #2: 25 3 2
7 5 3	

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform