B: BallDrops

The classic *Two Glass Balls* brain-teaser is often posed as:

> "Given two identical glass spheres, you would like to determine the lowest floor in a 100-story building from which they will break when dropped. Assume the spheres are undamaged when dropped below this point. What is the strategy that will minimize the worst-case scenario for number of drops?"

Suppose that we had only one ball. We'd have to drop from each floor from 1 to 100 in sequence, requiring 100 drops in the worst case.

Now consider the case where we have two balls. Suppose we drop the first ball from floor n. If it breaks we're in the case where we have one ball remaining and we need to drop from floors 1 to $n-1$ in sequence, yielding n drops in the worst case (the first ball is dropped once, the second at most $n-1$ times). However, if it does not break when dropped from floor n, we have reduced the problem to dropping from floors $n+1$ to 100. In either case we must keep in mind that we've already used one drop. So the minimum number of drops, in the worst case, is the minimum over all n.

You will write a program to determine the minimum number of drops required, in the worst case, given B balls and an M-story building.

Input

The first line of input contains a single integer P, $(1 \leq P \leq 1000)$, which is the number of data sets that follow. Each data set consists of a single line containing three (3) decimal integer values: the problem number, followed by a space, followed by the number of balls B, $(1 \leq B \leq 50)$, followed by a space and the number of floors in the building M, $(1 \leq M \leq 1000)$.

Output

For each data set, generate one line of output with the following values: The data set number as a decimal integer, a space, and the minimum number of drops needed for the corresponding values of B and M.

Sample Input

```
4
1 2 10
2 2 100
3 2 300
4 25 900
```

Sample Output

```
1 4
2 14
3 24
4 10
```