

Incremental House of Pancakes

Problem

Every morning at The Incremental House of Pancakes, the kitchen staff prepares all of its pancakes for the day and arranges them into two stacks. Initially, the stack on the left has **L** pancakes, and the stack on the right has **R** pancakes.

This restaurant's customers behave very consistently: the i -th customer to arrive (counting starting from 1) always orders i pancakes. When the i -th customer places their order of i pancakes, you take i pancakes from the stack that has the most pancakes remaining (or from the left stack if both have the same amount). If neither stack has at least i pancakes, the restaurant closes and the i -th customer does not get served any pancakes. You never complete an order using pancakes from both stacks.

Given the initial numbers of pancakes in each stack, you want to know how many customers will be served, and how many pancakes will remain in each stack when the restaurant closes.

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each test case consists of a single line containing two integers **L** and **R**: the initial numbers of pancakes in the left and right stacks, respectively, as described above.

Output

For each test case, output one line containing `Case #x: n l r`, where x is the test case number (starting from 1), n is the number of customers who will be served, and l and r are the numbers of pancakes that will remain in the left and right stacks, respectively, when the restaurant closes.

Limits

Time limit: 20 seconds per test set.

Memory limit: 1GB.

$1 \leq T \leq 1000$.

Test Set 1 (Visible Verdict)

$1 \leq L \leq 1000$.

$1 \leq R \leq 1000$.

Test Set 2 (Hidden Verdict)

$1 \leq L \leq 10^{18}$.

$1 \leq R \leq 10^{18}$.

Sample

Sample Input

```
3
1 2
2 2
8 11
```

Sample Output

```
Case #1: 1 1 1
Case #2: 2 1 0
Case #3: 5 0 4
```

In Sample Case #1, the first customer gets 1 pancake from the right stack, leaving 1 pancake in each stack. The second customer wants 2 pancakes, but neither stack has enough for them, even though there are 2 pancakes in total.

In Sample Case #2, the first customer gets 1 pancake from the left stack, because both stacks have the same amount. This leaves 1 pancake in the left stack and 2 in the right stack. The second customer wants 2 pancakes, which you serve to them from the right stack, emptying it. When the third customer arrives, neither stack has 3 pancakes, so no more orders are fulfilled.

In Sample Case #3, the first customer is served from the right stack, leaving 8 pancakes in the left stack and 10 in the right stack. The second customer is also served from the right stack, leaving 8 pancakes in each stack. The third customer is served from the left stack, leaving 5 pancakes there and 8 in the right stack. The fourth customer is then served from the right stack, leaving 4 pancakes there. Serving the fifth customer empties the left stack, and then there are not enough pancakes remaining in either stack to serve a sixth customer.