

## Problem G. Binary Tree

### Background

Binary trees are a common data structure in computer science. In this problem we will look at an infinite binary tree where the nodes contain a pair of integers. The tree is constructed like this:

- The root contains the pair (1, 1).
- If a node contains (a, b) then its left child contains (a + b, b) and its right child (a, a + b)

### Problem

Given the contents (a, b) of some node of the binary tree described above, suppose you are walking from the root of the tree to the given node along the shortest possible path. Can you find out how often you have to go to a left child and how often to a right child?

### Input

The first line contains the number of scenarios.

Every scenario consists of a single line containing two integers  $i$  and  $j$  ( $1 \leq i, j \leq 2 \cdot 10^9$ ) that represent a node (i, j). You can assume that this is a valid node in the binary tree described above.

### Output

The output for every scenario begins with a line containing "Scenario #i:", where  $i$  is the number of the scenario starting at 1. Then print a single line containing two numbers  $l$  and  $r$  separated by a single space, where  $l$  is how often you have to go left and  $r$  is how often you have to go right when traversing the tree from the root to the node given in the input. Print an empty line after every scenario.

### Sample Input

```
3
42 1
3 4
17 73
```

### Sample Output

```
Scenario #1:
41 0

Scenario #2:
2 1

Scenario #3:
4 6
```