Iproperty

After graduating together so many years ago, Jan and Piet meet at a party for retirees. Jan arrived in a dilapidated 2hp car and Piet in the most recent Maserati model. Jan could not believe his eyes and asks Piet: *you used to be a drinking brother like me, and now you've made it far; how did you do that?* Pete replies: *I've been drinking all my life, but I didn't return my empties until I was 65, and then I was instantly rich!*

Keeping empty seems to pay off. You have followed that same strategy, and now you want to cash in. However, you are only interested in ... more booze. So you bring a number of Spa empties back to the shop and buy a number of full Spas, of course less than you brought back. After a while that will yield enough new empty bins, and then you repeat that operation. How long that can continue depends of course on the number of empty bins you started with, the price of an empty bin and the price of a full bin. At a certain point it's done, and then you have some money left. In this task we want to know how many full boxes you can *to buy* with the number of empty bins you started with, and how much money you have left at the end.

Assignment

More precisely: you start with B empties. The price of one piece of empty is PriceL. The price of a piece that is full (including the empty) is PriceV (it always applies that *PriceV* > *PriceL*). If you continuously exchange empties (and some money if necessary) for full goods (and some money if necessary), how many units of full goods can you still consume until the moment when you no longer have enough to purchase another unit of full goods, and how much money have you at that moment ?

As an example: you have 2 empty items; the price of 1 piece of empty is 2 euros; the price full is 3 euros. You exchange those 2 empty pieces for 1 full and drink that bowl of Spa; you now have 1 piece of empty and 1 euro. With this you can buy 1 full bowl, drink it empty and exchange it for cash and then at the end you have had 2 full bowls, and you have 2 euros left.

Input

The first line represents the number of test cases. One line follows per test case with three numbers separated by a blank; those numbers are

- *B*: the number of empty items (bins, for example) with which you start
- *PriceL*: the price of one piece of empty
- *PriceV*: the price of one piece of complete goods

sample input

```
2
2 2 3
7 1 8
```

Export

You have to print one line per test case: it starts with the sequence number of the test case. Then comes the number of new pieces of full goods that you were able to purchase according to that test case and then how many euros you have left. The numbers are separated by one blank.

sample output

```
1 2 2
2 0 7
```