Fixed-Income Security Trade Allocation

A *portfolio* is a grouping of financial assets, such as stocks, bonds, or *fixed-income securities*. Each portfolio is managed by a *Portfolio Manager* who is in charge of sending portfolio orders to a *Trader*.

A *Trader* receives orders of varying sizes from different portfolios to buy a quantity of a *security* on the market. They then identify a *Seller* that is willing to sell units of the desired security.

In a best-case scenario, the *Seller* is selling enough of the security to fulfill all of the portfolio orders for the security; however, if that is not the case, the *Trader* must buy as much of the security as possible and *fairly* allocate it amongst the portfolios.

Fixed-Income Security

A fixed-income security has the following properties:

- minimum_trade_size The smallest number of units that can be traded with this security.
- increment The number of units the trade can be incremented with.
- $tradeable_amount = (minimum_trade_size) + (increment imes n)$, where n is a non-negative integer.
- avaliable_units The number of units of the security that are available for purchase on the market.

Fixed-Income Trade Orders

A fixed-income *Trader* has the following information:

- *portfolio_order* The number of units of the fixed-income security that a single portfolio wants to buy.
- total_order The total (sum) number of units made up of all the underlying portfolio_orders.

Defining a Proportional Allocation

If there are not enough $avaliable_units$ to fulfill all of the portfolio orders, we must find the proportional allocation for each portfolio's order of the $available_units$ on the market.

We get a portfolio's *proportional_allocation* with this expression:

$$proportional_allocation = rac{portfolio_order}{total_order} imes available_units$$

How Do We Fairly Allocate Units?

- Iterate through every underlying *portfolio_order* from smallest to largest (if two portfolios order the same number of units, then sort them lexicographically by ascending ID) and apply the following process:
 - ullet If the portfolio's $proportional_allocation$ is less than the $minimum_trade_size$, check if $proportional_allocation$ is greater than $\dfrac{minimum_trade_size}{2}$.
 - If false, do not allocate anything.

- If true, attempt to allocate the *minimum_trade_size* within the defined rules.
 - If this fails; allocate nothing.
- If the portfolio's *proportional_allocation* is greater than or equal to *minimum_trade_size*:
 - If the *proportional_allocation* is larger than or equal to the *portfolio_order*, allocate the *portfolio_order*.
 - If the *proportional_allocation* is not a *tradeable_amount*, round it down to the closest *tradeable_amount* that you can allocate within the defined rules. If you fail to find a *tradeable_amount* that satisfies the rules, allocate nothing.
- After allocating units (including the case when you allocate nothing) to a portfolio, perform the following steps to ensure that as much of the available security is purchased as is possible:
 - Recalculate the *total_order* based on the orders from the remaining portfolios (i.e., those whose orders haven't yet been allocated).
 - Subtract the quantity of units that were just allocated to an order and recalculate avaliable_units.
 - Recalculate the *proportional_allocation* of each portfolio awaiting allocation based on the remaining *avaliable_units*.

Rules That Always Hold

- A portfolio manager *only* orders tradeable amounts from the *Trader*.
- Each portfolio has to issue its own trade, hence the quantity allocated to each portfolio must be a $tradeable_amount$.
- An *untradeable amount* is a value that cannot be represented as a *tradeable amount*. Note that an exception of 0 is allowed (meaning that 0 is a tradeable amount). You must try to never leave a portfolio with leftover units ($portfolio_order allocated_amount$) that can't be traded on the market (i.e., an untradeable amount).

Given the basic information for a fixed-income security and a list of portfolio orders, find the *proportional_allocation* for each portfolio using the rules and processes defined above. Then, print each *portfolio_identifier* along with the amount of the security allocated to it as two space-separated values on a new line. Order your output alphabetically by *portfolio_identifier*.

Input Format

The first line contains an integer, T, denoting the number of portfolios hoping to place orders. The second line contains three space-separated integers denoting the respective values for the $minimum_trade_size$, increment, and $available_units$ for the fixed-income security. Each of the T subsequent lines defines a portfolio order as two space-separated values; the first value is a string denoting the $portfolio_identifier$, and the second value is an integer denoting the $portfolio_order$

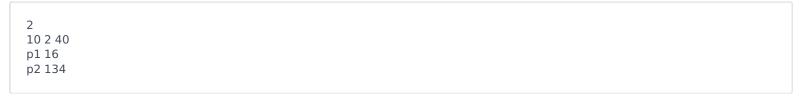
Constraints

- 0 < T < 1000
- ullet 0 < increment < $minimum_trade_size$ < $available_units$
- $portfolio_order = minimum_trade_size + increment \times n$ for some non-negative integer, n.
- $ullet minimum_trade_size imes number of portfolios < available_units$

Output Format

Print T lines where each line contains two space-separated values: a $portfolio_identifier$ followed by the number of units allocated to the portfolio. Your output must be ordered alphabetically by $portfolio_identifier$.

Sample Input



Sample Output

```
p1 0
p2 40
```

Explanation

First, we have the following information about our fixed-income security:

- \bullet $minimum_trade_size = 10$
- increment = 2
- $available_units = 40$

Next, we have T=2 portfolio orders:

- 1. p1's $portfolio_order = 16$.
- 2. p2's $portfolio_order = 134$.

We can calculate $\mathit{total_order} = 16 + 134 = 150$

$$p1_proportional_allocation = \frac{16}{16+134} \times 40 = 4.26$$

Because $p1_proportional_allocation < \frac{minimum_trade_size}{2} \implies 4.26 < 5$, nothing is allocated to p1. Therefore, all 40 $available_units$ are allocated to p2.