Lab 9 A

Problem: Stock Value Management

You are tasked with managing a stock portfolio, keeping track of various stocks, their quantities, and their prices. You need to read stock data from a file and compute the average value of each stock based on the quantity and price. Additionally, the program should be able to print the top 5 stocks by average value.

The input file contains a series of commands and stock entries. Each stock entry consists of a stock name, quantity, and price. The program should also handle a special query command, `PRINT_TOP_5`, which instructs the program to output the top 5 stocks by their average value.

Input Format:

The input.txt file contains a series of lines, each of which can either be:

- A stock entry in the format:

  ```
  <stock_name> <quantity> <price>
  ```

  Where:
  - `<stock_name>` is the name of the stock (a string of up to 20 characters).
  - `<quantity>` is an integer representing the number of stocks bought.
  - `<price>` is a floating-point number representing the price of the stock.
- A special query:

  ```
  PRINT_TOP_5
  ```

  This query does not contain stock data. Instead, it instructs the program to compute and print the top 5 stocks by their average value.

Output Format:

When a `PRINT_TOP_5` query is encountered, print the top 5 stocks (or fewer if there are less than 5 stocks) sorted by average value in descending order. If two stocks have the same average value, sort them alphabetically by their name.

The output should be in the format:

```
Top 5 Stocks by Average Value:
<stock_name1>: Avg Value = <avg_value1>
<stock_name2>: Avg Value = <avg_value2>
...
```

Constraints:

- The number of different stocks can be up to `100` (maximum value of `N`).
- Stock quantities can range from `1` to `1000`.
- Stock prices can range from `1.00` to `5000.00`.

Example:

Input File:

```
AAPL 10 150.50
GOOGL 5 2800.10
AAPL 15 160.75
PRINT_TOP_5
MSFT 8 300.20
GOOGL 10 2750.25
```

Output:

```
Top 5 Stocks by Average Value:
GOOGL: Avg Value = 2800.10
AAPL: Avg Value = 156.65
```

Notes:

- Ensure that the `PRINT_TOP_5` query only prints the top 5 stocks by average value.
- The input file may contain multiple `PRINT_TOP_5` queries, and the output should reflect the top 5 stocks at that point in time.

Problem 2: E-Commerce Cart System

In an e-commerce platform, a system is needed to manage items in a shopping cart based on various operations provided in an input file. The system should keep track of the quantities and values of items in the cart, allowing for easy updates and display of item details.

The input file, named `students.txt`, contains a list of operations to be applied to items in the shopping cart. Each operation will either initialize an item's price, add or remove quantities of an item

Each line in the file follows one of the formats below:

- **Price Initialization:** `ItemName Price` – Sets the price for an item. If the item already exists, updates the price.
- **Add/Remove Quantity:** `ItemName Add Quantity` or `ItemName Remove Quantity` – Adds or removes a specified quantity of the item in the cart. The quantity in the cart should not go below zero.

Your task is to:

1. Read and process each operation from the `students.txt` file.
2. Maintain a record of each item's price and quantity in the cart.
3. In the end, calculate the item's total value in the cart and print the item's name, quantity, and total value (rounded to two decimal places).

Constraints:

- Each line in `students.txt` represents a valid operation for the cart system.
- Number of Operations <= 100
- The quantity for each `Add` or `Remove` operation is a positive integer.
- 0 < Item_Price <= 1000.00

Input Format:

The file `students.txt` contains lines formatted as follows:

```
ItemName Price
ItemName Add Quantity
ItemName Remove Quantity
```

Output Format:

Output the item's details in the following format:

```
ItemName Quantity TotalValue
```

Example File:

Suppose `students.txt` contains the following:

```
Item1 10.0
Item2 5.0
Item1 Add 3
Item2 Add 2
Item1 Remove 1
```

Example Output:

```
Item1 2 20.00
Item2 2 10.00
```

Notes:

- The program reads and parses each line of data, maintaining the current quantity and value of each item in the cart.
- For each `Show ItemName` command, the item's total value is calculated as `price * quantity` and printed rounded to two decimal places.
- If an item's quantity becomes zero, its total value should also be zero.

Submission Guidelines

Do not rename any files given in the handout. Only write the code in the specified C files in the respective directories.