Problem 182: Shopping Spree

Difficulty: Hard

Authors: Cindy Gibson and Danny Lin, Greenville, South Carolina, United States

Originally Published: Code Quest 2022

Problem Background

Online shopping has become a massive industry over the past two decades. From major corporations like Amazon to small businesses working out of their owner's basements, there are now very few things you can't order online and expect to have delivered to your doorstep a few days later. The digital "shopping cart" concept has even expanded to other applications, and now is a common, intuitive way to view and organize anything you've selected on a website.

Most shopping cart interfaces share two common features: users may only view or modify the contents of their own cart, and items in a user's cart are not purchased until they finish "checking out." This differs from a physical shopping cart like you may see in a grocery store; there, a shopper physically removes an item from the shelf, making it unavailable to other customers. Online, simply having an item in your cart usually doesn't place any sort of reservation on those items. Another person could place the same item(s) in their cart and check out before you have a chance to do so, potentially leaving those items out of stock.

Problem Description

The Lockheed Martin site you're working at is adding a new automated convenience store to allow employees to buy snacks and drinks throughout the day. Employees can select what they want from a website, and equipment in the store will drop those items into a locker for pick up. Your team has been assigned to build the website. As with any website, the site you're designing will need to process each user's commands as they arrive. You need to write a program to handle that processing. However, since multiple users can be connected at one time, the commands from one user may be mixed in with those of others.

This is a new concept, and so the store's inventory isn't very large, and it's possible the store may sell out of items rather frequently. Your program will also be provided with the store's inventory at the start of each day, and should manage that inventory as orders come in. Customers should not be allowed to add any items to their cart that are already out of stock.

Once a customer checks out, the program should total up the cost of all items they have in their cart and remove those items from the inventory. If any items went out of stock between being added to a user's cart and that user's checkout, the user should be notified that those items are not available, and should not be charged for them. The user will be able to return to the site at a later time and attempt to order the items again once they are restocked.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include:

- A line containing two positive integers, separated by spaces:
  - I, the number of types of items in the inventory
  - C, the number of commands received to the website from customers
- I lines listing items held in the inventory. Each line will contain three values, separated by spaces:
  - A positive integer indicating the number of items in inventory.
  - A text string containing the name of the item. This may contain uppercase letters, lowercase letters, and underscores (_)
  - A positive decimal value indicating the cost of a single item of this type.
- C lines listing commands received from customers. Each command will begin with a positive integer representing the customer's ID; all commands received from a particular customer will begin with the same ID value. The remainder of the command will include several values, separated by spaces. Possible commands include:
  - ADD <ITEM> <QTY> - Used to add items to the cart. <ITEM> will be the name of the item as it appears in the inventory report, and <QTY> will be replaced with a positive integer representing the number of items to add. If the requested number plus the number of those items already in the customer's cart is equal to or less than the number currently in the inventory, add that number of items to the customer's cart, but do not remove them from the inventory at this time. Otherwise, the command should be rejected.
  - REMOVE <ITEM> <QTY> - Used to remove items from the cart. <ITEM> will be the name of the item as it appears in the inventory report, and <QTY> will be replaced with a positive integer representing the number of items to remove. If the requested number and type of items exists in the customer's cart, those items should be removed from the cart. Otherwise, the command should be rejected.
  - CHECKOUT - Used to finalize the customer's order. The contents of the user's cart should be checked against the current inventory. For any items where the customer cannot receive the full requested amount, give the customer none of those items; instead, remove that item from the cart and report the shortage. All remaining items should then be removed from the inventory, and the total cost of those items reported to the user.

```
1
3 8
2 Candy_Bar 1.50
3 Soda_Bottle 1.60
2 Cheese_Pack 2.50
1 ADD Candy_Bar 1
2 ADD Candy_Bar 2
1 ADD Soda_Bottle 3
1 REMOVE Soda_Bottle 1
2 CHECKOUT
1 CHECKOUT
3 ADD Candy_Bar 1
3 ADD Cheese_Pack 2
```

Sample Output

For each test case, your program must print the results of each command received in the order they are received, as follows:

- ADD commands:
  - If the command is successful, print a single line of the format "Added <QTY> <ITEM> to customer <ID>'s cart"
  - If the command is rejected, print a single line of the format "Not enough <ITEM> for customer <ID>"
  - In both cases, replace <QTY> with the number of items added, <ITEM> with the name of the item, and <ID> with the customer's ID number.
- REMOVE commands:
  - If the command is successful, print a single line of the format "Removed <QTY> <ITEM> from customer <ID>'s cart"
  - If the command is rejected, print a single line of the format "Customer <ID> does not have that many <ITEM>s"
  - In both cases, replace <QTY> with the number of items added, <ITEM> with the name of the item, and <ID> with the customer's ID number.
- CHECKOUT commands:
  - For each item where there is insufficient stock to meet the customer's request, print a line of the format "Out of stock of <ITEM>". If multiple items have insufficient stock, print them in alphabetical order.
  - Finally, print a line of the format "Customer <ID>'s total: $<COST>", replacing <ID> with the customer's ID number, and <COST> with the total cost of all items remaining in the customer's cart, rounded to two decimal places and including any trailing zeroes.

Once all commands are processed, your program must also print one line for each item remaining in inventory, in alphabetical order, in the format "<ITEM> - <QTY>", replacing <ITEM> with the name of the item and <QTY> with the quantity remaining in inventory. Omit items that are out of stock.

```
Added 1 Candy_Bar to customer 1’s cart
Added 2 Candy_Bar to customer 2’s cart
Added 3 Soda_Bottle to customer 1’s cart
Removed 1 Soda_Bottle from customer 1’s cart
Customer 2’s total: $3.00
Out of stock of Candy_Bar
Customer 1’s total: $3.20
Not enough Candy_Bar for customer 3
Added 2 Cheese_Pack to customer 3’s cart
Cheese_Pack - 2
Soda_Bottle - 1
```