CS2810 OOAIA: A2

Design Submission Deadline: 16:40 January 20, 2020

Code Submission Deadline: 23:45 January 22, 2020

Problem Statement

The problem is to maintain an inventory of items under additions and deletions. The items are pens and pencils. There are three classes Inventory item, Pen, and Pencil. Each class has data members and member functions, as mentioned below. Each object of the Inventory item class is related to the object of class Pen and Pencil using unique id. So, the deletion of the object of the class, either pen or pencil with id say 23 leads to deletion of the object of class inventory item with the same id (This must be implemented using destructor logic). However, the deletion of objects of a class inventory item is not possible. Apart from insertion operation and deletion operation, the print operation should print all the objects of the specified class in ascending order of their id.

Data Structures

Inventory item class has the following interface:

- Inventory Item (id: int, price: float, manufacturer: string, type: string)
- ~Inventory Item()
- int getId()
- float getPrice()
- string getManufacturer()
- string getType()

Note: Exact method names can be changed.

Pen class has the following interface:

- Pen (id: int, width: float, color: string, style: string)
- ~Pen()
- int getId()
- float getWidth()
- string getColor()
- string getStyle()

Note: Exact method names can be changed.

Pencil class has the following interface:

- Pencil (id: int, width: float, hardness mark: string, point size: string)
- ~Pencil()
- int getId()
- float getWidth()
- string getHmark()
- string getPointsize()

Note: Exact method names can be changed.

Also we recommend to use static arrays (as explained below) to store the object references of each class.

Static Data Members

When a data member of a class is declared as static, only one copy of that member is created for that class and is shared by all the objects of that class. All static data is initialized to zero when the first object is created if no other initialization is present. Static members are mainly used to maintain values that are common to the entire class.

Declaration:
```
static int count;  // static variable
static int arr[N];   // static array
```

Example:
```
class staticExample{
        static int count;
        // data members

        //member functions
}
int staticExample::count;
```

Note: The type and scope of each static member variable must be defined outside the class definition. This is necessary because the static data members are stored separately rather than as a part of an object.

Input Format

First line of input contains an integer *t* denoting number of operations.

There are total type of 3 operations:

Addition operation: Operation <a, b, c> creates object of type inventory item, pen and pencil respectively.
So operation <a, b, c> is followed by data (as shown in sample input).

Deletion operation: Operation d delete object having particular id.
Note: delete operation can’t delete object of class inventory item. Object of this class can be deleted only by deleting object of class pen or pencil with same id. Delete operation deletes the object of class either pen or pencil based on where matching id is present.

Print operation: Operation <p, q, r> will print all the objects of class inventory item, pen and pencil respectively in sorted order (according to id).

Each Input file has certain number of "blocks". Each block has a set of addition operations followed by deletion operations followed by a set of print operations. The addition operation adds some inventory items then some pens and then some pencils. The IDs for the pens and pencils to be added are chosen from the inventory items that have just been added in the same block. The deletion operations delete certain number of pens and pencils. The set of print operations just uses <p, q, r> to print the state of the inventory items.

Note: The deletions in a block could be a pen/pencil that was added in any of the previous blocks. The additions are "independent" in the sense that each block adds items with a unique set of IDs

Constraints

1 <= *t* <= 1000

Output format

Each line in output file represents particular object of class either inventory item or pen or pencil as shown in sample output.

Sample Input

```
17
a 10 23.32 abc xyz
a 11 43.4 cdf uvw
b 10 3.32 ghi lmp
c 11 34.43 jkl omn
a 20 23.32 abc xyz
a 21 43.4 cdf uvw
a 22 67.3 abc pqr
b 20 3.32 ghi lmp
d 10
d 11
p
q
r
```

```
d 20

p

q

r
```

Sample Output

```
20 23.32 abc xyz
21 43.4 cdf uvw
22 67.3 abc pqr
20 3.32 ghi lmp
21 43.4 cdf uvw
22 67.3 abc pqr
```

Explanation

First thirteen lines of input forms the single block. Out of them first eight line are for addition operation. Next two line are deletion operation and last three line are print operations. Hence output after processing the first block is:

```
20 23.32 abc xyz
21 43.4 cdf uvw
22 67.3 abc pqr
20 3.32 ghi lmp
```

Note: how delete operation: ‘d 10’ deleted both the objects of class inventory item and pen. Also note that when object of class pen/pencil is added in a block and later deleted in some other block, it is never "re-added" also ids are unique around the classes pen, pencil and are subset of id present in inventory item class.

Deletion of objects with missing id is not permitted.

Challenge link

https://www.hackerrank.com/cs2810-a2