

✓ Inventory Manager

Java May'19 DSA Set&Map 2 - 3 days 00:30:35

Your task is to implement a system that helps a store owner manage his inventory. New items can be added and filtered in the market. Here are the commands that the owner needs to have implemented:

[Submit solution](#)[My submissions](#)
[All submissions](#)
[Best submissions](#)

- **add ITEM_NAME ITEM_PRICE ITEM_TYPE** – adds a new item to the system;
 - **ITEM_NAME** can be any unique sequence of 3 to 20 characters;
 - **ITEM_PRICE** can be any positive floating-point number, up to 5000;
 - **ITEM_TYPE** can be any sequence of 3 to 20 characters and it may not be unique;
 - Prints: **"Ok: Item ITEM_NAME added successfully"**, if the item is added successfully;
 - Prints: **"Error: Item ITEM_NAME already exists"**, if the item already exists;
- **filter by type ITEM_TYPE** – lists the first 10 items that have the given ITEM_TYPE;
 - Prints: **"Error: Type ITEM_TYPE does not exists"**, if the given ITEM_TYPE is non-existent;
- **filter by price from MIN_PRICE to MAX_PRICE** – lists the first 10 items that have ITEM_PRICE in the given range, inclusive;
- **filter by price from MIN_PRICE** – lists the first 10 items that have a greater ITEM_PRICE than the given, inclusive;
- **filter by price to MAX_PRICE** – lists the first 10 items that have a smaller ITEM_PRICE than the given, inclusive;
- **end** – marks the end of the commands. No commands will follow.

All items that are listed by the **filter** commands must be printed in the following format:

```
Ok: LIST_OF_ITEMS
```

LIST_OF_ITEMS contains the filtered items, separated by a space and a comma (" ") and each item is represented as **"ITEM_NAME(ITEM_PRICE)"**. They must also be sorted by the following criteria:

- First by ITEM_PRICE, ascending
- Then by ITEM_NAME, ascending
- Last by ITEM_TYPE, ascending

If **LIST_OF_ITEMS** contains no items, then print **"Ok:"**.

✓ **Points:** 100
(partial)
⌚ **Time limit:** 0.6s
Java: 2.0s
📄 **Memory limit:**
64M
Java: 64M
✍ **Author:**
[martin](#)

🏷 **Tags**
Hash Tables
⬆ **Difficulty**
Intermediate

▼ **Allowed languages**
java



The input data is given at the standard input. It consists of a sequence of commands, each at a separate line, terminated by the command **end**. The commands will be valid (as described in the above list), in the specified format, within the constraints given below. There is no need to check the input data explicitly.

Java May'19 DSA Set&Map 2 - 3 days 00:30:35

Output

For each command from the input sequence print at the standard output its result as a single line.

Constraints

- All **ITEM_NAME** and **ITEM_TYPE** will consist of letters and digits only. No spaces are allowed.
- All **filter by price** commands will occur no more than 100 times in any test, and approximately 2% of all commands
- The total number of lines in the input will be in the range [1 ... 50 000]

Sample Tests

Input

Copy



```
add SmartWatch 1111.90 technology
add Candy 0.90 food
add Lemonade 11.90 drinks
add Sweatshirt 121.90 clothes
add Pants 49.90 clothes
add CowMilk 1.90 dairy
add Eggs 2.34 food
add Cheese 5.55 dairy
filter by type clothes
filter by price from 1.00 to 2.00
add FreshOrange 1.99 juice
add Aloe 2.7 juice
filter by price from 1200
add Socks 2.90 clothes
filter by type fruits
add DellXPS13 1700.1234 technology
filter by price from 1200
filter by price from 1.50
filter by price to 2.00
filter by type clothes
end
```

Java May'19 DSA Set&Map 2 - 3 days 00:30:35

Output

Copy



Ok: Item SmartWatch added successfully

Ok: Item Candy added successfully

Ok: Item Lemonade added successfully

Ok: Item Sweatshirt added successfully

Ok: Item Pants added successfully

Error: Item CowMilk already exists

Ok: Item Eggs added successfully

Ok: Item Cheese added successfully

Ok: Pants(49.9), Sweatshirt(121.9)

Ok: BulgarianYogurt(1.9), CowMilk(1.9)

Ok: Item FreshOrange added successfully

Ok: Item Aloe added successfully

Ok:

Ok: Item Socks added successfully

Error: Type fruits does not exists

Ok: Item DellXPS13 added successfully

Ok: DellXPS13(1700.1234)

Ok: BulgarianYogurt(1.9), CowMilk(1.9), FreshOrange(1.99), Eggs(2.34), Aloe(2.7),
Socks(2.9), Cheese(5.55), Lemonade(11.9), Pants(49.9), Sweatshirt(121.9)

Ok: Candy(0.9), BulgarianYogurt(1.9), CowMilk(1.9), FreshOrange(1.99)

Ok: Socks(2.9), Pants(49.9), Sweatshirt(121.9)

Java May'19 DSA Set&Map 2 - 3 days 00:30:35

? Clarifications

No clarifications have been made at this time.