Problem 3 – Online Market

You are given an online market. Products can be added and queried in the market. You are given a sequence of commands that must be implemented:

- **`add PRODUCT_NAME PRODUCT_PRICE PRODUCT_TYPE`** – adds a new product to the market
  - **`PRODUCT_NAME`** can be any unique sequence of 3 to 20 characters
  - **`PRODUCT_PRICE`** can be any positive floating-point number, up to 5000
  - **`PROCUCT_TYPE`** can be any sequence of 3 to 20 characters. Product type may not be unique
  - Print "**`Ok: Product PRODUCT_NAME added successfully`**" if the product is added
  - Print "**`Error: Product PRODUCT_NAME already exists`**" if the product already exists
- **`filter by type PRODUCT_TYPE`** – lists the first 10 products that have the given PRODUCT_TYPE
  - Print "**`Error: Type PRODUCT_TYPE does not exists`**", if the given PRODUCT_TYPE is non-existent
- **`filter by price from MIN_PRICE to MAX_PRICE`** – lists the first 10 products that have PRODUCT_PRICE in the given range, inclusive
- **`filter by price from MIN_PRICE`** – lists the first 10 products that have a greater PRODUCT_PRICE than the given, inclusive
- **`filter by price to MAX_PRICE`** – lists the first 10 products that have a smaller PRODUCT_PRICE that the given, inclusive
- **`end`** – marks the end of the commands. No commands will follow

All products that are listed by the **`filter`** commands must be printed in the format "**`Ok: LIST_OF_PRODUCTS`**". LIST_OF_PRODUCTS contains the filtered products, separated by a space and a comma (", ") and each product is represented as "**`PRODUCT_NAME(PRODUCT_PRICE)`**". If the result from **the filtering by price is 0** products, then print "**`Ok:`** ". They must also be sorted by the following criteria:

- First by PRODUCT_PRICE, ascending
- Then by PRODUCT_NAME, ascending
- Last by PRODUCT_TYPE, ascending

Input

The input data is given at the standard input. It consists of a sequence of commands, each at a separate line, ending 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.

Output

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

Constraints

- All **PRODUCT_NAME** and **PRODUCT_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 in a test
- The total **number of lines** in the input will be in the range [1 … 50 000]
- Allowed working time for your program: 0.6 seconds.
- Allowed memory: 64 MB.

Example

Input	Output
add Milk 1.90 dairy add Yogurt 1.90 dairy add Notebook 1111.90 technology add Orbit 0.90 food add Rakia 11.90 drinks add Dress 121.90 clothes add Jacket 49.90 clothes add Milk 1.90 dairy add Socks 2.90 clothes filter by type dairy filter by price from 1.00 to 2.00 filter by price from 1.50 filter by price to 2.00 filter by type clothes end	Ok: Product Milk added successfully Ok: Product Yogurt added successfully Ok: Product Notebook added successfully Ok: Product Orbit added successfully Ok: Product Rakia added successfully Ok: Product Dress added successfully Ok: Product Jacket added successfully Error: Product Milk already exists Ok: Product Socks added successfully Ok: Milk(1.9), Yogurt(1.9) Ok: Milk(1.9), Yogurt(1.9) Ok: Milk(1.9), Yogurt(1.9), Socks(2.9), Rakia(11.9), Jacket(49.9), Dress(121.9), Notebook(1111.9) Ok: Orbit(0.9), Milk(1.9), Yogurt(1.9) Ok: Socks(2.9), Jacket(49.9), Dress(121.9)
add Milk 1.90 dairy add Yogurt 1.90 dairy add Notebook 1111.90 technology add Orbit 0.90 food add Rakia 11.90 drinks add Dress 121.90 clothes add Jacket 49.90 clothes add Milk 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 CappyOrange 1.99 juice add Nestey 2.7 juice filter by price from 1200 add Socks 2.90 clothes filter by type fruits add MacBookPro 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	Ok: Product Milk added successfully Ok: Product Yogurt added successfully Ok: Product Notebook added successfully Ok: Product Orbit added successfully Ok: Product Rakia added successfully Ok: Product Dress added successfully Ok: Product Jacket added successfully Error: Product Milk already exists Ok: Product Eggs added successfully Ok: Product Cheese added successfully Ok: Jacket(49.9), Dress(121.9) Ok: Milk(1.9), Yogurt(1.9) Ok: Product CappyOrange added successfully Ok: Product Nestey added successfully Ok: Ok: Product Socks added successfully Error: Type fruits does not exists Ok: Product MacBookPro added successfully Ok: MacBookPro(1700.1234) Ok: Milk(1.9), Yogurt(1.9), CappyOrange(1.99), Eggs(2.34), Nestey(2.7), Socks(2.9), Cheese(5.55), Rakia(11.9), Jacket(49.9), Dress(121.9) Ok: Orbit(0.9), Milk(1.9), Yogurt(1.9), CappyOrange(1.99) Ok: Socks(2.9), Jacket(49.9), Dress(121.9)