C++ OOP – Exam (4 September 2022)

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++11 standard.

Submit your solutions here: https://judge.softuni.org/Contests/3597/CPlusPlus-OOP-Regular-Exam-4-September-2022

Only source code will be accepted as solution for each task.

2. Console Store

The world is hungry for gaming consoles!

It is your time to shine. You decide to open a hardware console store and sell all those how new PlayStations and Xboxes. As it turns out…this is not so easy.

Your task is to study the code and implement the missing functionalities.

There are two types of consoles sold at your store:

```
enum class ConsoleType {
  PS,
  XBOX
};
```

Your program is also given several commands to execute.
After you execute the command you should print to the standard output the outcome of the command.
The possible commands are:

```
enum class CommandType {
  ADD,
  REMOVE,
  SORT_BY_PRICE,
  SORT_BY_QUALITY,
  PRINT
};
```

Where:

- ✓ `ADD` **–** creates a new console of the specified type and store it into your "Store". Additional console fields such as price, quality and console generation are also provided here.
- ✓ `REMOVE` – removes the last added console of the provided type. You are **guaranteed** that there will be such console in your Store (the input will need to validation)
- ✓ `SORT_BY_PRICE` – sort the selected console of the provided type by their **price** in descending order (the one with the biggest price is first, the one with the lowest price is last)
- ✓ `SORT_BY_QUALITY` – sort the selected console of the provided type by their **quality** in descending order (the one with the biggest quality is first, the one with the lowest quality is last)
- ✓ `PRINT` – print data for all the selected console type that are present in your Store

Input

The input is already handled for you. You don’t need to parse anything additionally.

Output

- When a command ADD is received you should print:
“Adding: *Xbox* with price: *P*, quality: *Q*”
or

“Adding: *PS* with generation *G*, price: *P*, quality: *Q*”
Where:
G – current console generation
P – current console price
Q – current console quality

- When a command REMOVE is received you should print:
(Same as the above, just change the “Adding” with “Removing”)

- When a command SORT_BY_PRICE is received you should print:
“Sorting all *Console* by price”
Where
Console – the selected console type (PS or XBox)

- When a command SORT_BY_QUALITY is received you should print:
“Sorting all *Console* by quality”
Where
Console – the selected console type (PS or XBox)

- When a command PRINT is received you should print:
“Printing all *Console* data”
Where
Console – the selected console type (PS or XBox)
Followed by individual console information **for each** console of the selected type present at your store.
“Xbox with price: *P*, quality: *Q*”

“*PS* with generation *G*, price: *P*, quality: *Q*”

See the provided examples section for concrete examples.

Restrictions

No restrictions.

Time limit: 500ms (0.50s)
Memory limit: 16 MB

Examples

Input	Output

3
0 1 200 100
0 1 100 150
1 1
	Adding: Xbox with price: 200, quality: 100
	Adding: Xbox with price: 100, quality: 150
	Removing: Xbox with price: 100, quality: 150

4
0 1 200 100
0 1 100 150
3 1
4 1
	Adding: Xbox with price: 200, quality: 100
	Adding: Xbox with price: 100, quality: 150
	Sorting all Xbox by quality
	Printing all Xbox data
	Xbox with price: 100, quality: 150
	Xbox with price: 200, quality: 100

8
0 0 50 200 1
0 1 150 250
0 1 200 100
0 1 100 150
3 1
4 1
2 1
4 1
	Adding: PS with generation 1, price: 50, quality: 200
	Adding: Xbox with price: 150, quality: 250
	Adding: Xbox with price: 200, quality: 100
	Adding: Xbox with price: 100, quality: 150
	Sorting all Xbox by quality
	Printing all Xbox data
	Xbox with price: 150, quality: 250
	Xbox with price: 100, quality: 150
	Xbox with price: 200, quality: 100
	Sorting all Xbox by price
	Printing all Xbox data
	Xbox with price: 200, quality: 100
	Xbox with price: 150, quality: 250
	Xbox with price: 100, quality: 150