Q7. Map(STL)

For this lab section, we hope all of you learn what Map is and how to implement your own Map by std::vector and use std::sort(). In addition, you have to understand how to write a file.

We will provide you with a file (Lab7_template.cpp) and a test case (test.txt), and you need to do all TODO parts in the file.

Class member:

```
   vector<pair<string, info>> student
```

Class Methods:

```
   readFile() {}
   sortByName(){}
   sortByInfo(const char* type){}
   add(string name, info& _info){}
   find(string name){}
   erase(string name){}
   write(fstream& file) {}
   writeFile(){}
```

The detail of Class:

1. student:

String is a student's name, and info is a struct, student's id and score included.

2. readFile():

Read the file to get the command number, and the commands. The command table is listed below.

a { name } { student_ID} { score }	If a student's name exists in the vector, you need to replace the student's info with new info. Otherwise, just add the element to the vector
f { name }	If a student's name exists in the vector, you need to show the student's info. Otherwise, print ("name" is not found!)
e { name }	If a student's name exists in the vector, you need to erase the student's element in the vector. Otherwise, print ("name" is not found!)

3. add(string name, info& _info):

Add a new element to the vector. If a student's name exists in the vector, override his or her info with new info and show the changes.

4. find(string name):

Search the element in the vector. If a student's name exists in the vector, you need to show the student's info. Otherwise, print ("name" is not found!).

5. erase(string name):

Erase the element in the vector. If a student's name exists in the vector, you need to erase the student's element in the vector. Otherwise, print ("name" is not found!).

6. sortByName():

Sort the element in the vector by name in ascending order.

7. sortByInfo(const char* type):

Sort the element in the vector by the specific way in ascending order. The type table is listed below. You need to do three comparison functions.

"SC"	Compare elements by score
"ID"	Compare elements by student ID
"SCID"	Compare elements by score and student ID. If two students have the same score, and then sort two students by student ID.

8. write(fstream& file):

You should write the sorted vector to the file.

9. writeFile():

Write the results of the sorted vector by methods. Don't modify this function.

Note:

1. When you insert an element to a map(STL), the map will sort all elements by key automatically. But you don't do this in this lab, that is, you just add an element to the vector or modify an element in the vector.
2. You must use an iterator and don't use std::map.

Hint:

Map:

1. https://www.cplusplus.com/reference/map/map/
2. Teacher's slide "10_Template_STL_voice_Eng.pptx" p.42 ~ p.46

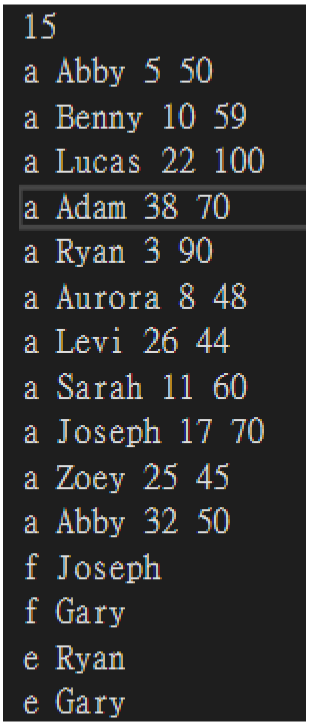
Write File:

1. Teacher's slide "10_Streams_FILE_I_O_voice_Eng.pptx"

Sort:

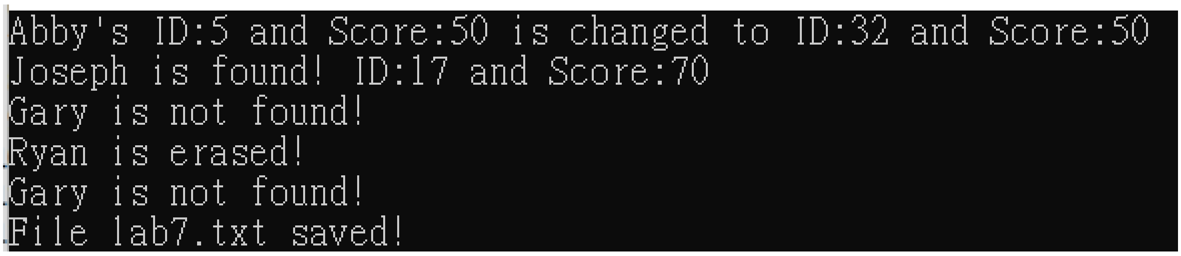
1. http://www.cplusplus.com/reference/algorithm/sort/
2. Teacher's slide "10_Template_STL_voice_Eng.pptx" p.3 ~ p.4

Sample Input:



```
15
a Abby 5 50
a Benny 10 59
a Lucas 22 100
a Adam 38 70
a Ryan 3 90
a Aurora 8 48
a Levi 26 44
a Sarah 11 60
a Joseph 17 70
a Zoey 25 45
a Abby 32 50
f Joseph
f Gary
e Ryan
e Gary
```

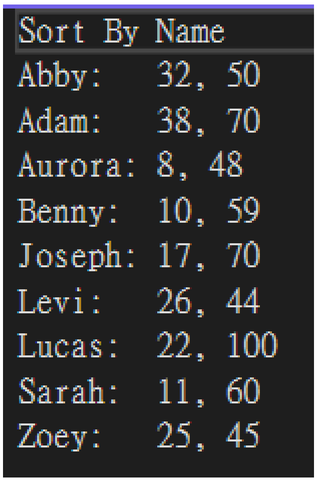
Sample Output (standard output):



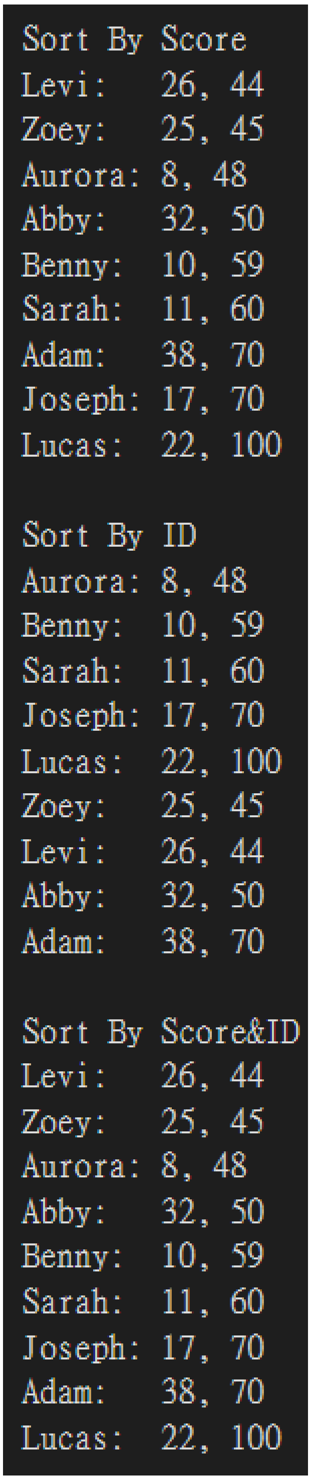
```
Abby's ID:5 and Score:50 is changed to ID:32 and Score:50
Joseph is found! ID:17 and Score:70
Gary is not found!
Ryan is erased!
Gary is not found!
File lab7.txt saved!
```

File Output:

You have to print your output and file the same as the format that we ask.



```
Sort By Name
Abby:   32, 50
Adam:   38, 70
Aurora: 8, 48
Benny:  10, 59
Joseph: 17, 70
Levi:   26, 44
Lucas:  22, 100
Sarah:  11, 60
Zoey:   25, 45
```



```
Sort By Score
Levi:   26, 44
Zoey:   25, 45
Aurora: 8, 48
Abby:   32, 50
Benny:  10, 59
Sarah:  11, 60
Adam:   38, 70
Joseph: 17, 70
Lucas:  22, 100

Sort By ID
Aurora: 8, 48
Benny:  10, 59
Sarah:  11, 60
Joseph: 17, 70
Lucas:  22, 100
Zoey:   25, 45
Levi:   26, 44
Abby:   32, 50
Adam:   38, 70

Sort By Score&ID
Levi:   26, 44
Zoey:   25, 45
Aurora: 8, 48
Abby:   32, 50
Benny:  10, 59
Sarah:  11, 60
Joseph: 17, 70
Adam:   38, 70
Lucas:  22, 100
```