Homework 1

100 Points

Linked Lists

Project: Build and process a sorted linked list. As en entry-level programmer you have to be able to read, understand existing code and update it (add new features). One of this assignment's goals is to read about 500 lines of code (in three files: MountainList.h, MountainList.cpp, 22C_Hw1.cpp), compile and run the program, understand it, and change it as required. The existing program does the following:

A. Reads data from a text file (**`mountains.txt`**) and inserts them into a sorted linked list. The list is to be sorted in ascending order by the mountain name named **`name`** (a unique key). The **`Mountain`** structure has four fields:

`name` (string)	such as	**`Mauna Kea`**
`elevation` (int)		**`9109`**
`range`(string)		**`Sierra Nevada`**
`state` (string)		**`California`**

B. Prints the list as a table with four columns (header included)
C. Prints the number of items in the list.
D. Searches the list: prompts the user to enter the name of the mountain or QUIT to stop searching; searches for that name: if found, displays its' data, otherwise displays some message, such as "… Not found"
E. Deletes nodes from the list: prompts the user to enter the name of the mountain to be deleted from the list or QUIT to stop deleting.
F. Prints the number of items in the list.
G. Destroys the list //no memory leak

YOUR TASK is to read and understand this program. Then do the following:

--> Keep track of and display the number of cities in the list: add a count member variable of the linked list class, and a getCount() function.
--> Replace the Mountain structure by a Mountain class and place it in a new file(s).
--> Make changes to improve the displayList() function.
--> Make changes to improve the searchList() function.
--> Make changes to improve other function of the linked list class.
--> Add a new feature: prompt the user to enter a state, such as California, and display all mountains in your list that belong to that state. Continue asking the user for other states, until s/he enters QUIT.

Grading // see next page

Grading:

Create a project consisting of at least 5 source and header files: - 10

Mountain.h
Mountain.cpp *// might not be needed*
MountainList.h
Mountain.cpp
22C_Hw1.cpp
mountains.txt

Display the number of items in the list	- 10
Replace the Mountain structure by a Mountain class	- 20
Improved displayList()	- 10
Improved searchList()	- 10
Improved other functions	- 10
New feature	- 25
Self assessment	- 5

Run the program once and save the output at the end of the source file as a comment. Compress the source and header files, input and output files (if any), and upload the compressed file: 22C_LastName_FirstName_H1.zip

Self Assessment: estimate the grade you deserve based on the above grading criteria.

Create the input file using the following data:

```
Shasta
14179
Cascade Range
California

Churchill
15638
Saint Elias Mountains
Alaska

Antero
14276
Sawatch Range
Colorado

Granite Peak
12807
Beartooth Mountains
Montana

Bachelor
9068
```

```
Cascade Range
Oregon

Adams
12281
Cascade Range
Washington

Doublet Peak
13600
Wind River Range
Wyoming

Mauna Kea
13803
Mauna Kea
Hawaii

Castle Peak
9109
Sierra Nevada
California

Pyramid Peak
9985
Crystal Range
California

Torbert
11413
Tordillo Mountains
Alaska

Rainier
14411
Cascade Range
Washington

Half Dome
8836
Yosemite National Park
California

Jeff Davis Peak
12771
Snake Range
Nevada
```