Homework 2 BCS 370 Fall 2017

HW 2 is the continuation of your lab 1. Unless you have completed lab 1 and demo-ed it to me, you must upload your code through Blackboard (Please see the submission requirements).

Task 1 [50 pts] Common Operations

We assume all the data records use name as the key, and cell as the value. You should complete the following operations on the records, including display/insert/delete/search and update.

- Your insert should run a search for the key first and return a message "pre-existing record" and do nothing if the key is found, or proceed to insert if the key is not found.
- Your delete should return an error message if the key is not found.
- Your search should return an error message if the key is not found.
- Your update should return an error message if the key is not found.

Task 2 [30 points] Growable Array.

Please complete resizeStorage function of data_management class in a similar way to the last programming question in your HW 1.

Where are resizeStorage being called? When the insertion(**you need to update your insert function**) and loading(already done in the constructor) are about to go beyond the capacity of the array.

- Declare a new dynamic array with twice the current size of the array.
- Copy over the previous value to the new array because we want to keep all the records we have.
- Delete the old array to make sure there is no memory leak.
- Store the new array in the base pointer (the member pointer that points to the array of records).
- Update the size and the capacity of the array.

With resizeStorage your application should be able to handle various-size records without any problem and all operations should work like before. I will test your task 2 with the big_records.csv .

Taks 3 [50 points] (3) Big Threes.

Please implement the big threes for data management class:

- copy constructor (15pts)
- assignment operator overloading (15pts)
- destructor (10pts)

1. You are encouraged to write a test file (10 pts, see 2). However, it is not a bad idea to convince yourself first that your code works by stepping through your code under debug mode and observing the memory usage.

2.You should write some driver code to test your code. Instead of driving the data_management object(s) through the console-based menu, you should write a small test code to verify your big threes work as expected.

Here is something you can do:

```
data_management dm("records.dat");

data_management dm2 = dm; //use the assignment operator

data_management dm3(dm2); //use the copy constructor
```

/*then you can manipulate these three objects in different ways and at the end display the different records of them to show that the copy constructor and the assignment operator overloading actually work.*/

Grading Criteria:

1) If your code does not compile, or compiles with any warnings, you will only receive up to 30% of full points.

2) Please see the grading sheet for the breakdowns of grades.

3) You will lose another 10% on your earned points if any memory-leak bug or danging-pointer bug is found.

Submission Requirements:

1) At the end of this document, there is a grading sheet. You must fill-up the **ALL** fields of self-evaluation part of the table.

- You must at least fill in comment "A" or "P" or "N" to indicate that you have completed all, part of, or nothing of the task respectively.
- You are encouraged to write down more information about your implementation, including but not limited to:
    - additional tests you have done.
    - corner cases your code might not handle.
    - better solution you come up with.
    - assumption you have made for your code.
- Even if you complete nothing on a task, if you mark "N", you will receive 15% for the question. If you do not answer, you will receive zero, and I will not review your solution for that task.

2) You must submit the grading sheet with your code, or you will receive 0 (see 3)

3) Please save the grading sheet to the same folder where your project folder is. Please keep the directory hierarchy of the project and then compress the project folder into a zip file and upload it through Blackboard.

Task 1 (50 pts, 10 pts each)	**Self-Evaluation**	**Score (used by the instructor)**
Display all the data records		
Insert a new record		
Delete a record		
Search for a record		
Update a record		
Task 2 (30 pts, 5pts each)		
Update the insertion of your code to use resizeStorage		
Declare a new array with twice size of the old array		
Copy records from the old array to the new array		
Free the old array properly		
Swap the base pointer(the member that points to the dynamic array) to the new array		
Update all the counters (the size and the capacity)		
Task 3 (50 pts)		
Copy Constructor—prototype, 2pts		
Copy Constructor – copy of non point-to members, 3pts		
Copy Constructor – copy of point-to members, 10pts		
Assignment Overloading – prototype, 2 pts		
Assignment Overloading – handle the self-assignment, 2pts		
Assignment Overloading – assign non point-to members, 3pts		
Assignment Overloading – assign point-to members, 8 pts		
Destructor, 10 pts		
Test code, 10 pts		