Wichita State University CS 211 Fall 2017

Homework #8

Bubble, Bubble, Toil and Bubble Sort a Dynamically Allocated Array of Strings

---

Assigned: November 2, 2017
Due: Nov. 9 by 11:59:59 PM

Write a C++ program which will prompt the user for the size of their list, then to enter the words, and then sort the words in alphabetical order and display them.

Requirements:

- Name the source file for your program **program8.cpp**.
- Prompt the user for the number of words they would like to enter
- Dynamically allocate an array of strings (each element of the array is of type `string`) of that size.
- You can assume that each word entered by the user will not contain any spaces, so the words can be read in using "`cin >>`".
- The program must stop prompting for further input once the maximum number of words is reached
- There is **no need** for the user to type "0" to indicate that they are finished. You have already asked them how many words they want to type, so that is how many words they will type
- Sorting of the words must be done using the *Bubble Sort* method, implemented in a function with this prototype:
  `void bubbleSort(string words[], int num);`
  This function must sort the `num` strings of the `words` array from "smallest" to "largest" as determined by the `compare` function (for any strings, whether of letters or not), which corresponds to alphabetical order if the strings are of letters.
  Don't do anything special to account for upper- or lower-case letters – all upper-case letters will be considered as coming before all lower-case letters (this is known as lexicographic sorting).
- For Extra Credit, see the extra credit section
- A sample run of your program should look like:

```
How many words will you be entering? 7
Enter a word: dog
Enter a word: deer
Enter a word: Dogs
Enter a word: beaver
Enter a word: Cat
Enter a word: Mouse
Enter a word: rabbit

Your sorted list is: Cat Dogs Mouse beaver deer dog rabbit
```

Extra Credit:

- To score up to 105% on this assignment, sort the list in true alphabetical order
- This means that the upper-case and lower-case versions of a letter should be treated as if they are equals
- A sample run of your program should look like:

```
How many words will you be entering? 7
Enter a word: dog
Enter a word: deer
Enter a word: Dogs
Enter a word: beaver
Enter a word: Cat
Enter a word: Mouse
Enter a word: rabbit

Your sorted list is: beaver Cat deer dog Dogs Mouse rabbit
```

Hints:

- Be sure to track how many words were entered by the user and sort the correct number of words.
- Here is some pseudocode for Bubble Sort (from `https://en.wikipedia.org/wiki/Bubble_sort`):

```
procedure bubbleSort( A : list of sortable items )
    n = length(A)
    repeat
        swapped = false
        for i = 1 to n-1 inclusive do
            /* if this pair is out of order */
            if A[i-1] > A[i] then
                /* swap them and remember something changed */
                swap( A[i-1], A[i] )
                swapped = true
            end if
        end for
    until not swapped
end procedure
```

- For the Extra Credit, remember that if the words being compared are all lower-case or all upper-case, lexicographic order is the same as alphabetic order

Reminders:

- Be sure that your program includes your name, ID, description, etc. as shown in the General Homework Requirements Handout
- Use good style including indentation, comments, etc. Part of the grade will be for style and quality.
- Carefully test your program.
- You are welcome to write your program at home. If you do, be sure to compile and test it in the lab before submitting it.

How to submit your program:

- Submit the file `program8.cpp` electronically using `~cs211b/bin/handin 8 program8.cpp`