WICHITA STATE UNIVERSITY – CS 211

Homework #7 – Counting & Sorting

DUE: October 29 by 11:59:59 PM **Assigned**: October 22

Background

Take as input a sentence from the user. Count how many occurrences of each letter there are and output a sorted list.

Assignment Requirements

- The name of your source code file shall be `hw07.cpp`
- Ask for and store a sentence or phrase that is typed in by the user
- Count the occurrences of every letter (do not separate upper and lowercase)
- Print a table showing only the letters that appeared, sorted so that the letter that appeared the most frequently is at the top
    - In case of a tie, fall back to alphabetical order
- Perform the sort using bubble sort
    - You are required to write this function
        - The return type shall be void
        - The name of the function shall be bubble_sort

Sample Runs

```
Enter a phrase: It's a hard knock life
A   2
I   2
K   2
C   1
D   1
E   1
F   1
H   1
L   1
N   1
O   1
R   1
S   1
T   1
```

```
Enter a phrase: The quick brown fox jumped over the lazy dog.
E   4
O   4
D   2
H   2
R   2
T   2
```

```
U  2
A  1
B  1
C  1
F  1
G  1
I  1
J  1
K  1
L  1
M  1
N  1
P  1
Q  1
V  1
W  1
X  1
Y  1
Z  1
```

Hints

- When starting to test your code, don’t use phrases or sentences. Instead just type letters for which you control the counts
- Choose to work exclusively in upper-case letters
- Test with non-letter ASCII characters
- Your life may be made easier by utilizing `std::pair` from `<utility>`
- Pseudo-code for the bubble-sort algorithm

```
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
```

Reminders

- Be sure to include a comment block at the top of every file with the required information
  - Refer to the General Homework Requirements handout on Blackboard
- Provide meaningful comments
  - If you think a comment is redundant, it probably is
  - If you think a comment is helpful, it probably is
  - Remember that you are writing comments for other programmers, not people who know nothing (obligatory Jon Snow) about coding
  - Comments are more helpful when they explain why, not what or how
- There will be no extensions

Preparing and Submitting

- Your code must be able to compile and run on the EECS lab machines
  - You are responsible for testing your code
  - "But it runs fine on my machine!" will **not** earn you any points
- Submit **ONLY** your source code file
- Homework submission will be handled exclusively through Blackboard