24-780 Engineering Computation Problem Set 06

You need to create a ZIP file (It may appear as a compressed folder in Windows) and submit the ZIP file via the 24-780 Canvas course. The file name of the ZIP file must be:

PS06-YourAndrewID.zip

For example, if your Andrew account is *hummingbird@andrew.cmu.edu*, the file name must be:

PS06-hummingbird.zip

If your ZIP file does not comply with this naming rule, you will automatically lose 5% credit from this assignment. If we are not able to identify who submitted the file, you will lose another 5% credit. If we finally are not able to connect you and the submitted ZIP file, you will receive 0 point for this assignment. Therefore, please make sure you strictly adhere to this naming rule before submitting a file.

The ZIP file needs to be submitted to the 24-780 Canvas course. If you find a mistake in the previous submission, you can re-submit the ZIP file with no penalty as long as it is before the submission deadline.

Notice that the grade will be given to the final submission only. If you submit multiple files, the earlier version will be discarded. Therefore, if you re-submit a ZIP file, the ZIP file MUST include all the required files. Also, if your final version is submitted after the submission deadline, late-submission policy will be applied no matter how early your earlier version was submitted.

Make sure you upload your Zip file to the correct location. If you did not upload your assignment to the correct location, you will lose 5%.

The ZIP file needs to include:

- C++ source file of your program (ps6-1.cpp, ps6-2.cpp)

Submission Due: Please see Canvas.

START EARLY!

Unless you are already a good programmer, there is no way to finish the assignment overnight.

PS6-1 Case-insensitive word search (ps6-1.cpp) [50 points]

Write a C++ program (**ps6-1.cpp**) that takes a file name and a keyword as input from the console window and finds how many times the word appears in the file. Your program needs to prompt for the file name first, and then prompt for the keyword. Your console window needs to look like:

```
Enter file name>C:/Users/soji/sampleText/text.txt
Enter keyword>Word
```

Use fgets from stdin or std::getline to take file name and keyword as input. (Make sure to remove control code at the end of line if you use fgets as demonstrated in class. Otherwise your program won't find the file.)

Your program then opens the given file, and search for the keyword. When the program finds a keyword, it needs to print the line number and the content of the line. Also your program needs to count the occurrence of the word, and print how many times the keyword is included in the text file.

To make it easier, if the keyword is "the", you can count "there" or "breathe" as one occurrence. Also, you can assume that one line is less than 256 characters long. (If a matching word span across the 256-character boundary, your program doesn't have to find the word.)

Your search must be case-insensitive. So, if the user enters the keyword "the", it must match all of "the", "The", "tHe", "THe", "thE", "ThE", "tHE", and "THE".

If the file name that the user entered does not exist, (i.e., cannot open the file in the reading mode) print "Cannot open file." and terminate the program.

If you are using macOS, make sure to turn off SandBox, or your program won't be able to read a file.

Sample console output:

```
Enter File Name>C:/Users/soji/text1.txt
Enter Keyword>small
14: practically usable solution has been found. The smallest known
19: is unknown whether 88-element solution is the smallest possible
Appeared 2 times.
```

PS6-2 Wall Clock Animation (ps6-2.cpp) [50 points]

The following sample program prints current time on the console window.

```
#include <stdio.h>
#include <time.h>

void GetLocalTimeHourMinSec(int &hour,int &min,int &sec)
{
    struct tm *localTime;
    time_t t=time(NULL);
    localTime=localtime(&t);

    hour=localTime->tm_hour;
    min=localTime->tm_min;
    sec=localTime->tm_sec;
}

int main(void)
{
    int hour,min,sec;
    GetLocalTimeHourMinSec(hour,min,sec);
    printf("%d %d %d\n",hour,min,sec);
    return 0;
}
```

In PS6-2, you write a program that draws a clock that shows the current time on the graphics window. The specification is same as Problem Set 3, except that your Second arm needs to be red.

Your program needs to update the clock until the user presses the ESC key.

You can cut & paste from the sample solution to Problem Set 3, or what you submitted for PS3.

Save your program as ps6-2.cpp and include in the zip file you submit to the Canvas.