# CS 1410 Introduction to Computer Science – CS2
# Section 1: MWF 10:30 a.m. – 11:20 a.m.
# Section 2: MWF 1:00 p.m. – 1:50 p.m.
# Instructor: Xiaojun Qi
# Assignment #6

**Given: Sunday, March 2, 2014**
**Due: 11:59 p.m. Sunday, March 9, 2014**
**Total Points: 30 points**

In this assignment, you will make use of a few of C-string functions and write your own "modularized" functions. Here is a list of possible functions that you may consider to use for solving the problem:

- strlen, strcpy, and strstr
- isalpha, isalnum, isdigit
- toupper, tolower
- getline

Make sure that you include the right header files for the chosen functions. You may consider using istringstream and ostringstream to simplify your I/O related operations since they are subclasses of istream and ostream and work similarly as the cin and cout. Refer to slides 16-19 of class notes for some simple examples. Please carefully read all posted examples for Chapter 12 as well.

The file that you will use to test your program is your Program Execution Environment (PEE). When your program starts executing, C++ sets a runtime variable char **environ; to point to the beginning of the environment array. By declaring this variable in your program as illustrated below, you can access this array. Note that the array of C-strings is terminated with environ[k] is equal to NULL, where k is the number of lines in the environment description. **Make sure to copy the following section in your .cpp file to avoid any possible typos!**

```
#include <iostream>
using namespace std;
int main()
{
        extern char **environ;   // needed to access your execution environment

        int k = 0;

        while (environ[k] != NULL)
        {
                cout << environ[k] << endl;
                k++;
        }
        return 0;
}
```

The output generated by this program will vary depending on your operating system (OS). However, much of it will be the same on any PC. In any case, the grader will check your programs on the same PC and the results will be the same for your programs if they work correctly. A partial sample of the output for my machine is given below:

ALLUSERSPROFILE=C:\ProgramData
.... ...

OS=Windows_NT
PATH=C:\Program Files\Common Files\Microsoft Shared\....

PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
... ...

Your task is to write a C++ program which executes as follows:

- Read in the system's PEE and output it to the console
- **Output a count for each word (i.e., the number of times each word occurring in the PEE) and a count of the total number of words.**
    - A word is any string of at least two characters, i.e., letters, and/or digits, and/or underscores "_". To be a word, **the string must have at least one letter.** Thus x86 is a word while 86 and C: are not words. Any other characters including <newline> and whitespace should be interpreted as delimiter separators between words and non-words (or words). For example, the string "SystemRoot C:\Users" consists of 2 words, where SystemRoot is a word and Users is a word. C:\Users is not a word because the characters ":" and "\" are not legal word characters.
- Finally, the program should accept from the user, on a separate line, **a string of up to 40 characters**. Any character string with or without spaces, **except <newline> and "END"**, is legal. Your program should output all the lines in which that string occurs without distinguishing between upper case and lower case letters. **It should also output the total number of times that the string occurs over the entire PEE file**. For example, for the line containing "abc 1ABC //ABcd3", the string "abc" would be found 3 times since the user input "abc" should be viewed the same as the string "ABc" or "ABC".
    - If a string occurs more than once in a line, count each occurrence but output the line only once.
    - A string occurring as a substring in a string should be counted. For example, for the following line (a fake example):

      CommmonProgramW6432=C:\Program Files\Commmon Files

      If the string to search for is "MM", the line has **4** occurrences of "MM". If the string to search for is "M", the line has 8 occurrences of "M."
- Your program should loop accepting strings to search for until the string "END" is the input. The program should then terminate execution.