CSCE2014 Programming Foundations II
Homework Two
By Wing Ning Li

1 Problem Description

Implement the standard library function **strtok** using C++/C basic language constructs and the functions of **Homework one**.

In concept, this assignment is similar to that of Lab three where we have implemented two simpler library functions **strlen** and **strcat**. To develop a solution for a problem, we must first understand the problem. To implement strtok, we must first understand how strtok behaves. In Lab three you have learned a little about how strtok works. You are welcome to experiment with it more (by writing code using the function and displaying the result) if there is any behavior that you are not sure about.

The following URLs provide the documentation for each function:
http://www.cplusplus.com/reference/cstring/strlen/
http://www.cplusplus.com/reference/cstring/strcat/
http://www.cplusplus.com/reference/cstring/strtok/

Try to read the documentation to see if it makes sense to you. At this point, it is normal if it doesn't make complete sense, but you will gradually be able to use this information to try out new functions without any outside help.

Typing "man strtok" on the prompt in a linux machine such as turing, we may obtain similar information.

Please pay attention to the function prototype. At this stage this is a complicated function for us so a discussion has been provided in the lecture. The example code of using it is written in C (which uses printf instead of cout for output) in the document page and see if you may replace printf with cout, testing your command of cout (you do not need to understand printf at all and hint: first run the code, observe the output and get an idea of the printf accomplishes, mimic the same output using cout). The strtok reference could be challenging for us to understand completely perhaps for now. So it is fine if you are not able to understand the example code or want to skip it, but we hope we can understand it soon, at least by the time of completing the homework.

2 Purpose

Understand and use pointers, arrays, and functions to solve a more complex problem; in particular pointers to char, arrays of char, C-style strings, and functions taking C-string as parameters and returning a pointer to char. Practice writing functions that are similar to standard library functions to show our ability and command of basic data structures (arrays) and programming in C/C++. Decompose a complex problem to some simpler problems (with

respect to the functions of **Homework one**) to get a solution. Build our confidence in using and programming arrays. Learn the function **strtok**. Learn static storage or variables within a function.

3 Design

For this part, we will follow what is provided in the documentation except for the name. We could use the identical name as used in the standard library if we create our own *namespace*, otherwise name conflict or clash may result. We will prefix the name with my.

```
char *mystrtok(char *str, const char *delim);
```

One way to solve a complex problem is to break it down to simpler problems. To help you breaking the tasks into simpler problems, we introduce the following functions for you. **You have implemented them in the last homework. You are required to use them (call them) in your implementation of mystrtok.**

```
char *find_first_not_in_the_set(char *str, const char *set);


char *find_first_in_the_set(char *str, const char *set);
```

4 Implementation

We are not allowed to use any global variable! No credit is given for using global variables.

We assume that someone (you or other folks) will use our mystrtok in their main program. We will create two files: mystrtok.h and mystrtok.cpp. Note in order our code of **Homework one**, our project needs to have helper.cpp and helper.h, and helper.h is included by include statement in mystrtok.h and mystrtok.h is included in mystrtok.cpp. Conceptually, mystrtok.h is the header file to be included for any code that needs to use our mystrtok function just as we have to include other headers in order to use the library functions or types. The mystrtok.cpp file (containing the implementation of mystrtok) is used to create the object file, say mystrtok.o, to be linked together by the compiler to create the executable file. We will have another file called homework2.cpp that contains the main function for testing our implementation of strtok, that is mystrtok function (see lab 3 set up of strlen and mystrlen, and strcat and mystrcat). Here is the idea or hint about how to implement mystrtok.

1. Inside the function, introduce a static pointer variable to remember where we left off.

2. Decide whether to use the first augment value or the value remembered to search for the next token.

3. Find the next token (idea: from the current beginning position search for a character that is not a delimiter, which is where the token starts; from that place search for a character that is a delimiter or `'\0'` character, which is where the token ends and change the delimiter character to the null character. `'\0'`)

4. make sure to update the static variable, and return the address (which is determined in the previous steps) of the first character of the token.

Note step 3 uses the two functions mentioned in the design section and you need to figure out how to call them with what arguments, how to use their return values, and so on.

5 Test and evaluation

The goal of testing is find bugs in our code (Does the code behave the way as expected?) Our main program should have several test cases for different conditions of using strtok or mystrtok function. Use the main program of Lab three as your first set of test cases. After that you need to think about other test cases, for example, what if the first parameter of strtok does not contain any delimiter at all. Or the delimiter string contains more than one delimiters. In testing of our program, we hope to design the test cases such that all situations are covered and each test case covers a new situation yet considered.

6 Report and documentation

A short report about things observed and things learned and understood. The report should also describe the test cases used in the main program and the reasons for each test case selected. Properly document and indent the source code. The source code must include the author name and as well as a synopsis of the file.

7 Project submission

Use Blackboard to submit all source files (helper.h and helper.cpp, mystrtok.h, mystrtok.cpp, homework2.cpp) and the project report in either .pdf or .doc or docx format. Note helper.h and helper.cpp is from homework1 and are the most current version as you are using them to implement mystrtok.