CS1050 – Lab 7

Spring 2020

Concepts to Practice

- Pointers
- Strings
- Expand the prelab

Submission Information

Submit this assignment by following the instructions given by your TA. SUBMIT ONLY the .c file (no a.out or executable file is required). All of the lab assignments must be submitted before the end of the lab using the lab code given by the TA.

Use the following submit command:

```
mucs submit <class> <assignment_type> <filename>
```

For example:

```
mucs submit 1050 lab x-lab7.c
```

Filename must be: *sectionletter-lab7.c* (include your respective lab section, e.g., a-lab7.c)

Description

For the lab assignment, you need to implement a program that takes an input string and divides it into segments. Each segment will be 3 characters long (or less if you run out of string). The program will then reverse each segment of the string in place. For example, if the original string was "abcdefgh" the resultant string should be "cbafedhg". That is, you can think of the original string as 3 segments of "abc", "def", and "gh". Reversing these you get "cba", "fed", and "hg" or a full result of "cbafedhg".

You may **not** call any function in string.h from the Standard C Library. You **must** use pointer arithmetic to move through any strings in the program – not the index operator []. You may **not** use any global variables at all.

The main() function in your program should:

1. Print a message welcoming the user to Lab 7.
2. Call a function (written by you) to get an original string from the user.
3. Print the original string.
4. Call a function (written by you) to determine how many segments are in the original string.
5. Print how many segments are in the original string.
6. Call a function (written by you) to reverse each segment of the original string in place.
7. Print the resultant string (the modified original string).
8. Print a message thanking the user.

Note that you may assume that the original string will not be larger than 255 characters (256 with space for the null-terminator).

Functions You Must Write

You may write any functions you wish to implement this program. Here is some advice

- You need a function that will get a string from the user.
- You need a function to calculate how many segments are in the original string.
- You need a function that reverses each segment.
- **int main(void)** – Of course, you need to write a main() 😊.

Sample Output (4 different runs)

```
jimr@JimRArea51:~/CS1050/SP2020/lab7$ ./a.out
*** Welcome to Lab 7 ***
Please enter a string: abcdefgh
You entered: abcdefgh
There are 3 segments in the string.
The modified string is: cbafedhg
*** Thanks for doing Prelab 7 ***

jimr@JimRArea51:~/CS1050/SP2020/lab7$ ./a.out
*** Welcome to Lab 7 ***
Please enter a string: a
You entered: a
There are 1 segments in the string.
The modified string is: a
*** Thanks for doing Prelab 7 ***

jimr@JimRArea51:~/CS1050/SP2020/lab7$ ./a.out
*** Welcome to Lab 7 ***
Please enter a string: ab
You entered: ab
There are 1 segments in the string.
The modified string is: ba
*** Thanks for doing Prelab 7 ***

jimr@JimRArea51:~/CS1050/SP2020/lab7$ ./a.out
*** Welcome to Lab 7 ***
Please enter a string: JimRiesIsCool
You entered: JimRiesIsCool
There are 5 segments in the string.
The modified string is: miJeiRsIsooCl
*** Thanks for doing Prelab 7 ***
```

Guidelines for Grading Lab 7
40 Points Possible

General

If your program does not compile or produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of ZERO POINTS. Further, if your program does not actually follow the specifications, but merely prints out lines that make it appear to follow the specifications, you will receive a grade of ZERO POINTS. For partial credit your C program must not only compile but also produce some valid I/O that meets the lab specifications.

You program is expected to have a comment header at the top that includes your name, pawprint, the course you are taking, and the lab that you are solved (e.g., "Lab 7"). Your code should be nicely indented. **You will lose up to 10 points if you do not meet these basic requirements**.

If you use a function from string.h and/or you use the index operator [], you will lose 50% of the points you would have otherwise received.

5 points: Program correctly allows string input from a user-written function.
5 points: Program correctly prints the string input by the user.
10 points: Program correctly calculates the number of segments the string.
20 points: Program correctly reverses each segment of the string.