CS1050 – Lab 9

Fall 2021

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_name> <filename>
```

For example:

```
mucs submit 1050 lab9 lab9.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 5 characters long (or less if you run out of string). The program will then reverse the order of the segments in-place in the original string. For example, if the original string was "abcdefgh" the resultant string should be "fghabcde". That is, you can think of the original string as 2 segments of "abcde" and "fgh". Reversing the order of these segments you get "fgh" and "abcde" or a full result of "fghabcde".

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 9.
2. Call a function (written by you) to get an original string from the user. Note that the storage for the original string should be declared in main().
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 the order of the segments 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 the order of the segments.
- **int main(void)** – Of course, you need to write a main() 😊.

Hints

- Set a pointer to the last segment in the string, and work backward.
- Have a temporary string where you can put each segment as you find it.
    - So, for the test string “abcdefgh” your temporary string would change as follows
        - “” (empty to begin with)
        - “fgh” (now it has just the last segment in it)
        - “fghabcde” (now it has both segments in it)
- If you use a temporary string, you can just copy it over the top of the original string at the end.
- To debug this, **load your code with printf() calls**! The easier it is to see what is going on (or going wrong) in your code, the easier it is to make things work correctly.
- Submit early and submit often! If you get something to work (at all), go ahead and submit it. Each time you improve your code, go ahead and submit again. That way, you won’t end up with seconds left in the lab and nothing working. We only grade the last valid (on-time and compiling) submission.

Honors Extension

For the honors extension, you should request a segment size from the user instead of hard-coding it to be 5. Hint – if you put a space at the beginning your scanf() format string, scanf() will “eat” whitespace characters that might be left over in the input buffer until it gets to the integer you want.

Non-honors Sample Output (6 different runs)

```
jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcdefgh
You entered: abcdefgh
There are 2 segments in the string.
The modified string is: fghabcde
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: a
You entered: a
There are 1 segments in the string.
The modified string is: a
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcd
You entered: abcd
There are 1 segments in the string.
The modified string is: abcd
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcde
You entered: abcde
There are 1 segments in the string.
The modified string is: abcde
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcdef
You entered: abcdef
There are 2 segments in the string.
The modified string is: fabcde
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: JimRiesIsCool
You entered: JimRiesIsCool
There are 3 segments in the string.
The modified string is: oolesIsCJimRi
*** Thanks for doing Lab 9 ***
```

Honors Sample Output (5 different runs)

```
jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcdefgh
Please enter a segment length:3
You entered: abcdefgh
There are 3 segments in the string.
The modified string is: ghdefabc
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ clear
jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcdefgh
Please enter a segment length:3
You entered: abcdefgh
There are 3 segments in the string.
The modified string is: ghdefabc
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcdefgh
Please enter a segment length:2
You entered: abcdefgh
There are 4 segments in the string.
The modified string is: ghefcdab
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: a
Please enter a segment length:1
You entered: a
There are 1 segments in the string.
The modified string is: a
*** Thanks for doing Lab 9 ***

jimr@JimRArea51:~/CS1050/FS2021/labs/lab9$ ./a.out
*** Welcome to Lab 9 ***
Please enter a string: abcdefghijklmnopqrstuvwxyz
Please enter a segment length:10
You entered: abcdefghijklmnopqrstuvwxyz
There are 3 segments in the string.
The modified string is: uvwxyzklmnopqrstabcdefghij
*** Thanks for doing Lab 9 ***
```

Guidelines for Grading Lab 9
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 9"). 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 the index operator [], you will lose 50% of the points you would have otherwise received. If you use a global variable you will get zero points.

Non-Honors

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.
10 points: Program correctly reverses the order of each segment of the string when all segments are the same length.
5 points: Program correctly reverses the order of each segment of the string when there is a segment that is not the same length as other segments.
5 points: Program uses the original string in main() as the reversed string at the end.

Honors

5 points: Program correctly allows string input from a user-written function.
5 points: Program gets the segment length from the user rather than hard-coding it.
5 points: Program correctly prints the string input by the user.
5 points: Program correctly calculates the number of segments the string.
10 points: Program correctly reverses the order of each segment of the string when all segments are the same length.
5 points: Program correctly reverses the order of each segment of the string when there is a segment that is not the same length as other segments.
5 points: Program uses the original string in main() as the reversed string at the end.