

CMPSC 210
Principles of Computer Organization
Fall 2016

John Wenskovitch

<http://cs.allegheny.edu/~jwenskovitch/teaching/CMPSC210>

Lab 6 – C and MIPS
Due (via Bitbucket) Wednesday, 19 October 2016
30 points

Lab Goals

- Add new functionality to an existing C program
- Add new functionality to an existing MIPS program

Assignment Details

We have wrapped up our discussion of binary and hexadecimal number representations and mathematics, and are now getting ready to dig in to learning C and MIPS in more detail. This lab sets up some of that material by asking you to alter some chunks of code that you have already worked with in both languages.

Part 1: Adding Functionality to Old C Code (15 points)

In Lab 3, you were asked to write C code that populated a table with the values of four different functions (a solution to this challenge is provided as `TableSolution.c` in the lab folder).

For this lab, you should alter this code file (or start one of your own from scratch) that will allow the user to select the ordering of these functions in the table. The x column should always be displayed first, so only the $1/x^3$, \sqrt{x} , $\log_3 x$, and 1.2^x columns should be configurable. You can start by iteratively displaying a menu system to the user, asking them which column they would like first, second, etc.

Which column should appear first in the table?

1. $1/x^3$
2. `sqrt(x)`
3. `log_3 x`
4. 1.2^x

(Assuming that option 3 was chosen for the next menu presented)

Which column should appear second in the table?

1. $1/x^3$
2. `sqrt(x)`
3. 1.2^x

After the user has selected the ordering of all of the columns, you should request a maximum row value (again in the 1–50 range) to display, and recalculate the well-formatted table:

Enter an integer between 1 and 50: 32

x	<code>log_3(x)</code>	<code>sqrt(x)</code>	$1/x^3$	1.2^x
-	-----	-----	-----	-----
1	0.00000	1.00000	1.00000	1.20000
2	0.63093	1.41421	0.12500	1.44000
...
32	0.00003	5.65685	3.15465	341.82189

Part 2: Adding Functionality to Old MIPS Code (15 points)

In Lab 2, you were asked to fill in a code template that would print the characters of a string in reverse order (a solution to this challenge is provided as `StringReverseSolution.asm`).

For this lab, you should alter this code file (or start one of your own from scratch) that will continually print the same line of text in only forward order, but removing a character from the beginning or end on either iteration, alternating which side a character is removed from on each iteration. Missing characters should be replaced with spaces to keep the output aligned. The program should stop when no characters remain. For example, with the input “MIPS_is_awesome!” string, the output should look like:

```
MIPS_is_awesome!
IPS_is_awesome!
IPS_is_awesome
PS_is_awesome
PS_is_awesom
S_is_awesom
S_is_aweso
_is_aweso
_is_awes
is_awes
is_awe
s_awe
s_aw
_aw
_a
a
```

One way to solve this challenge is to keep track of the address of the first and last characters that you want to print, in addition to the beginning and end of the string. Then, a loop will print spaces from beginning to first print character, the characters of the string from first to last print characters, and then spaces again from last print character to the end of the string. Your branch checks will then simply involve comparing the current address of the character to print against each of these print segments.

An alternative solution would be to alter the input string itself, changing a character from the original input to a space on every iteration, then simply printing that input string. You can approach this problem in one of those ways, or by using your own approach.

Submission Details

For this assignment, your submission to your BitBucket repository should include the following:

1. A commented version of your variable table C code.
2. A commented version of your slow string erasing MIPS code.

Before you turn in this assignment, you also must ensure that the course instructor has read access to your BitBucket repository that is named according to the convention `cs210f2016-<your user name>`.