

## Lab 4

Get checked off for up to 3 points of incomplete work from the previous lab within the first 10 minutes of lab.

**ONLY Monday Lab Students: As there was no lab on Monday ( Jan 18 ), you can check off the lab for maximum of 10 points**

Each lab will begin with a recap of last lab and a brief demonstration by the TAs for the core concepts examined in this lab. As such, this document will not serve to tell you everything the TAs will in the demo. It is highly encouraged that you ask questions and take notes. In order to get credit for the lab, you need to be checked off by the end of lab. For non-zero labs, you can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For extenuating circumstance, contact your lab TAs and Instructor.

In this lab, you can form a group of 2-3 individuals. You must be checked off together as a group at the end of the lab. Although you perform tasks as a group, ensure that you understand the work and ask questions to TAs as needed.

### **(1 pt) Review from Lab #3**

1. Vim commands:

- a. How do you auto indent your program?
- b. Explain what the following commands do:  
**dd, y3, p, :set cindent**

### **(1 pt) VIM exercises**

**These exercises on the computer need to be repeated by each student in the pair.**

This is to ensure that both students understand how to get around in Linux!!! For this part of the lab, you will create a .vimrc file that will help you develop your C++ programs using VIM. First, we need to create a simple .vimrc file in your home directory on the ENGR (flip) server. **vim .vimrc**

In this file, you can insert the following lines so that it makes working with vim easier. Comments are prefaced with a quotation mark, “.

```
filetype on
filetype plugin on
filetype indent on
autocmd FileType c, cpp
set cindent “This allows for c-like indentation
set sw=3 “Set the sw for the auto indent to 3 spaces
set number “Show the line numbers on the left
```

```

"Change the color of the text and turn on syntax highlighting
"color desert
color torte
colorscheme evening
syntax on "Turn on syntax highlighting
set showmatch "Show matching braces
set showmode "Show current mode
"When one of the chars is typed, the matching is typed and the
cursor moves left "The line below is single quotes
inoremap ' "<Left>
"The line below is double quotes
inoremap " ""<Left>
inoremap { {}<Left>
inoremap ( ()<Left>

```

There are many more commands you can insert in this file, and here is a reference guide to some of these: <http://vimdoc.sourceforge.net/html/doc/startng.html>

## (2 pts) Testing and Debugging

Download the following cpp file. (Use wget command)

<http://classes.engr.oregonstate.edu/eecs/winter2020/cs161-010/labs/buggyCode.cpp>

You must find as many bugs as possible and fix them. Some are logic errors, some are syntax errors. Hint: there are **21** bugs in total, some are obvious, some are more complex. **Make sure you re compile and run your program after you make a single fix to a mistake to make sure you actually fixed the mistake, didn't introduce new errors, and/or eliminated other errors as a result of the fix.**

## (1 pt) Design atoi

atoi() is a common function which takes a character and returns its decimal ASCII value. Start by designing how this function will work. It should take any character found on the ASCII chart (<http://www.asciitable.com/>) and return the decimal value.

```

/*****
** Function: a_to_i
** Description: turns a character into a decimal value
** Parameters: char character
** Pre-Conditions: the input is a character
** Post-Conditions: returned the decimal value of the character
*****/

```

## (2 pts) Implement atoi

Write the a\_to\_i() function based on your design. Test your function thoroughly. Will your function properly return the decimal value of: 'A', '1', 'b', '/', ' ', etc.?

### (1 pt) Design itoa

Similar to `a_to_i()`, `i_to_a()` takes an integer and returns the character associated with that value. Design this function.

```
/******  
** Function: i_to_a  
** Description: turns a decimal value into a character value  
** Parameters: int decimal  
** Pre-Conditions: the input is an integer  
** Post-Conditions: returned the character which represents the decimal value  
*****/
```

### (2 pts) Implement itoa

Write the `i_to_a()` function. Test your function thoroughly. Will it return the correct character value for 127, 65, 97, etc.? For this exercise, you can assume that the input will not be less than 0 or greater than 127.

**Show your completed work to the TAs for credit. You will not get points if you do not get checked off!**