LAB #3
Practice vim, input, and conditional execution

Due to the cancellation of Monday labs, you are allowed to do this work on your own or in a Tuesday – Friday lab. Since lab #3 attendance is not required, you need to get lab #3 (and possibly lab #2) checked off at the beginning of lab #4.

Defining Terms (2 pts)

You and your partner need to define the following terms:

> **Compiler, data type, int, double, type cast, declaration, variable, assignment statement, operation, operand**
>
> In addition, write down any term that has been used in the course that you do not understand.

VIM Exercises (3 pts)

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!!!

1. First, create a lab3 directory in your labs directory, and change into the lab3 directory.
   cd labs
   mkdir lab3
   cd lab3

2. Now, let's practice using the vim editor following the instructions below. (If needed, refer to Lab #1 for a reference guide to the basic commands)
   1. Open a test file for practice – **vim test.cpp**
   2. Save the file using – **:w**<Press Return>
   3. Go into insert mode by pressing **i**
   4. Type the standard input/output header and main function in your test.cpp file.
   5. Get out of insert mode by pressing <Esc>.
   6. Show the line numbers in vim by typing **:set number**
   7. Go to the third line (or whichever line has main) in your code by typing **:3**
   8. Delete this line of code by typing **dd**
   9. Put the line of code back underneath the first line by moving your cursor to the top line and typing **p**
   10. Redo the delete by typing **u** to undo the paste.
   11. Paste the line back using **p**
   12. Now, copy the line of code by typing **yy** (If you want to copy >1line, you type

one less than the number of lines you want to copy after the y, i.e. y4 copies 5 lines)

13. Paste the line of code by moving your cursor to a line and typing **p**
14. Undo the paste by typing **u**.
15. Try using the **j, k, h,** and **l** keys to move around the screen. You can also use the arrow keys, but these don't always work.
16. Try searching for "main" in vim with **/main**.
17. Go back into insert mode, **i**, and type some random letters, dkfjdsl, then press <Esc> to go back into command mode.
18. Use the **x** to delete all the characters from this random string you typed.
19. Press **i** to begin typing text again.
20. Now, write a small program:
    - Ask the user if she/he likes vi as an editor.
    - Read the 0 or 1 integer value from the user
    - If the user says true, then display a message, "You love vi!"
    - If the user says false, then display a message, "You hate vi!"
21. Now, press <Esc> to get back into command mode, and go to the first line in your program by typing **:0**.
22. Then, type **=G** to auto indent your program. You can set the number of spaces you prefer by typing **:set sw=3**, then you want to auto indent again, **=G**. (If you want to auto indent while you type, then use **:set cindent**)
23. You can change your color scheme or background by using one of these commands:
    :colorscheme evening
    :set background=dark

    Now, write and quit the file by typing **:wq**

C++ Program (3 pts)

Now, write the C++ program from your design in Lab #2.

Input: amount of artificial sweetener needed to kill a mouse
weight of the mouse
weight of the dieter, when done dieting
how long you will live
how many diet sodas in a week

Output: amount of artificial sweetener needed to kill you
amount of artificial sweetener consumed by person in lifetime
write "yes" if the person will consume enough to cause death
write "no" if the person will not consume enough to cause death

Use Strings (2 pts)

You will practice using C++ strings to help with your assignment #2.

Problem Statement:
Read a string from the user, and output the number of characters in the string followed by the string of characters printed vertically, instead of horizontally. For example:

```
Enter a message: hello
Number of characters: 5
h
e
l
l
o
```

Things you need to accomplish this. First, investigate the API for the string class. http://www.cplusplus.com/reference/string/string/

- The first thing you need to do is include the string library to create a string object.
  ```
  #include <string>
  ```

- Then, you can create a string object, where you create variables in main.
  ```
  std::string message;
  ```

- You can read into a string using cin or getline, depending if you want to read word or line of text.
  ```
  std::cin >> message;    //read a word
  getline(std::cin, message);    //read a line
  ```

- Now you can use functions to access length and characters.
  ```
  std::cout << message.length() << endl;
  std::cout << "First char: " << message.at(0) << std::endl;
  ```

Hint for accessing all the characters in the string:

```
for(int i=0; i<message.length(); i++) {
     //You can access each character in here using i

}
```