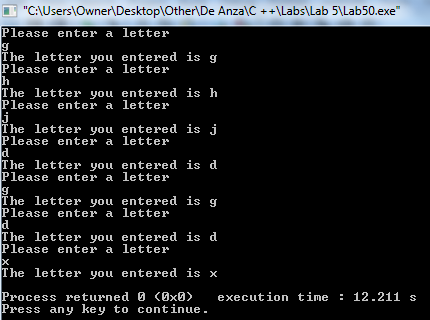
**Lab 51**

**First program:**

***This program is not user friendly. Run it a few times and explain why.***

It doesn't tell you how to exit the program.

***Take a screen shot of the output.***



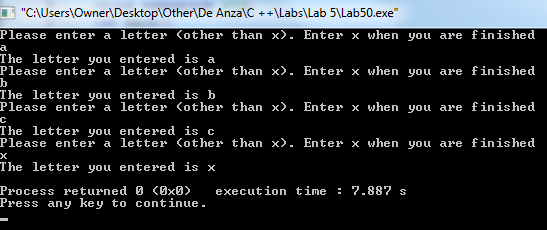
***Add to the code so that the program is more user friendly***

I added code to make it more friendly. I added a statement that allowed the user to know how to end the program.

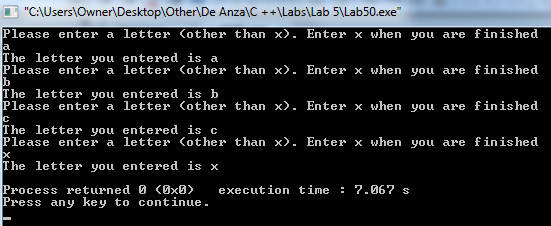
***How would this code affect the execution of the program if the while loop is replaced by a do-while loop? Try it and see. Take a screen shot of the output.***

There is no difference in having a while loop or a do while loop because of how the original code was written. Essentially, you get the same output with a while and a do while loop. The following were the outputs:

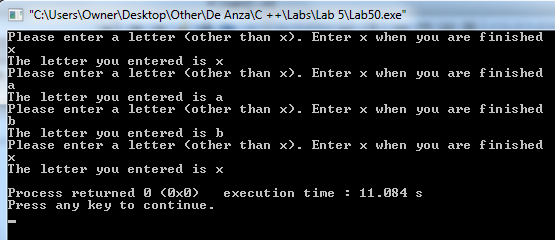
while loop output:



do while loop output:



Same output. **However**, I figured that maybe the original intention of this lab 5.1 was to show that there is a difference between these two loops. Perhaps the intention here was to have a "cin >> letter;" statement both inside the loop and before it. Because having none before it makes no sense (the differences between the while and do while loops aren't shown if this statement is present before the loop). If this is were to be the case, then there would be a difference. The do while loop, since it is a posttest loop, would allow for an "x" to be entered as the first input without ending the program (and the while wouldn’t). The following would be a do-while loop having "x" as the first input:



**Second program:**

Since the instructions didn’t ask to add a "cin >> letter;" statement before the loop (because that has nothing to do with making it "more user friendly") then my modified code will not reflect it.

***Complete the program above by filling in the code described in the statements in bold so that it will perform the indicated tas***k.

Done.

***Run the program several times with various input. Record your results. Are they correct?***

Yes. This program allows you to add all the rain fall of various months and add it all up to a final total.

***What happens if you enter -1 first?***

The program displays a message stating that no data has been entered.

***What happens if you enter only values of 0 for one or more months?***

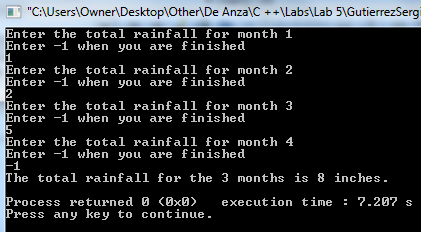
There is no problem in entering 0s. The program adds all the numeric data found in all the various months and adds them all up to a final total. 0s obviously provide no value to that total.

***Is there any numerical data that you should not enter?***

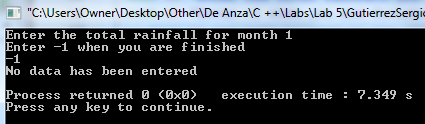
Not really. All numerical data adds up, even fractional numbers.

***Take a screen shot of the output.***

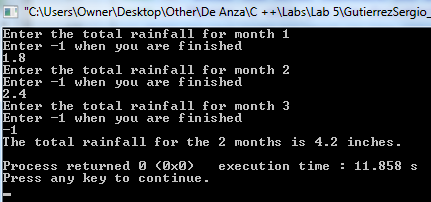
Output of 3 integer values:



Output after entering a value of -1:



Output of fractional values:



***What is the purpose of the following code in the program above?***

***if (month == 1) cout << "No data has been entered" << endl;***

***Take a screen shot of the output.***

That code is there in case the user enters an initial -1 value. By doing so, the value of "month" remains as 1 because the increment operator inside the while loop wouldn't be used to increment to initial value of months which is 1. Here is the output which reflects months having a value of 1:

