**CSCI 1411 – Lab 04 – Loops and Files**

**Goals:**

* Use for and while loops
* Read and write to files

**Development Environment:** csegrid (Centos), g++ (all students must use csegrid)

**Skills:** for loops, while loops, ifstream, ofstream

Reading: Chap 5

**Deliverables:** 1) This lab with, 5 screen shots 2) lastnameFirstLab04.cpp

**Part I – Skills Practice (5 points)**

* Log in to the csegrid. Make a directory called lab04 (remember it is case sensitive). Change directory into **lab04**. Create a file called **Lab04a.cpp** using nano (use your last name and your first initial)

**mkdir lab04  
cd lab04  
nano lab04a.cpp**

* Place the following text in the file (Do not cut and paste. You will learn more by typing it in. If you do cut and paste all of the single and double quotes will be incorrect)
* This program will provide a menu, read numbers from a file, print the next 3 numbers in a table to a file.

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ifstream infile("in.txt");

ofstream outfile("out.txt");

bool done = false;

int menu=3, item;

while (!done)

{

cout << "1. Read from and write to file\n";

cout << "2. Exit\n";

cin >> menu;

switch (menu)

{

case 1:

while (infile >> item)

{

cout << item << endl;

for (int i = 0; i < 5; i=i+2)

{

item++;

cout << item << "\t";

outfile << item << "\t";

}//for

cout << endl;

outfile << endl;

}//while

infile.close();

outfile.close();

cout << "done.\n";

break;

case 2:

done = true;

break;

default:

break;

}//switch

}//while

return 0;

}//main

* Save the file, exit out. At the command prompt type (the -o is a little Oh not a zero), the lastname and first is your lastname and first (what ever you saved the file as)
* Create an input file

**nano in.txt**

**2**

**4**

**6**

**8**

**10**

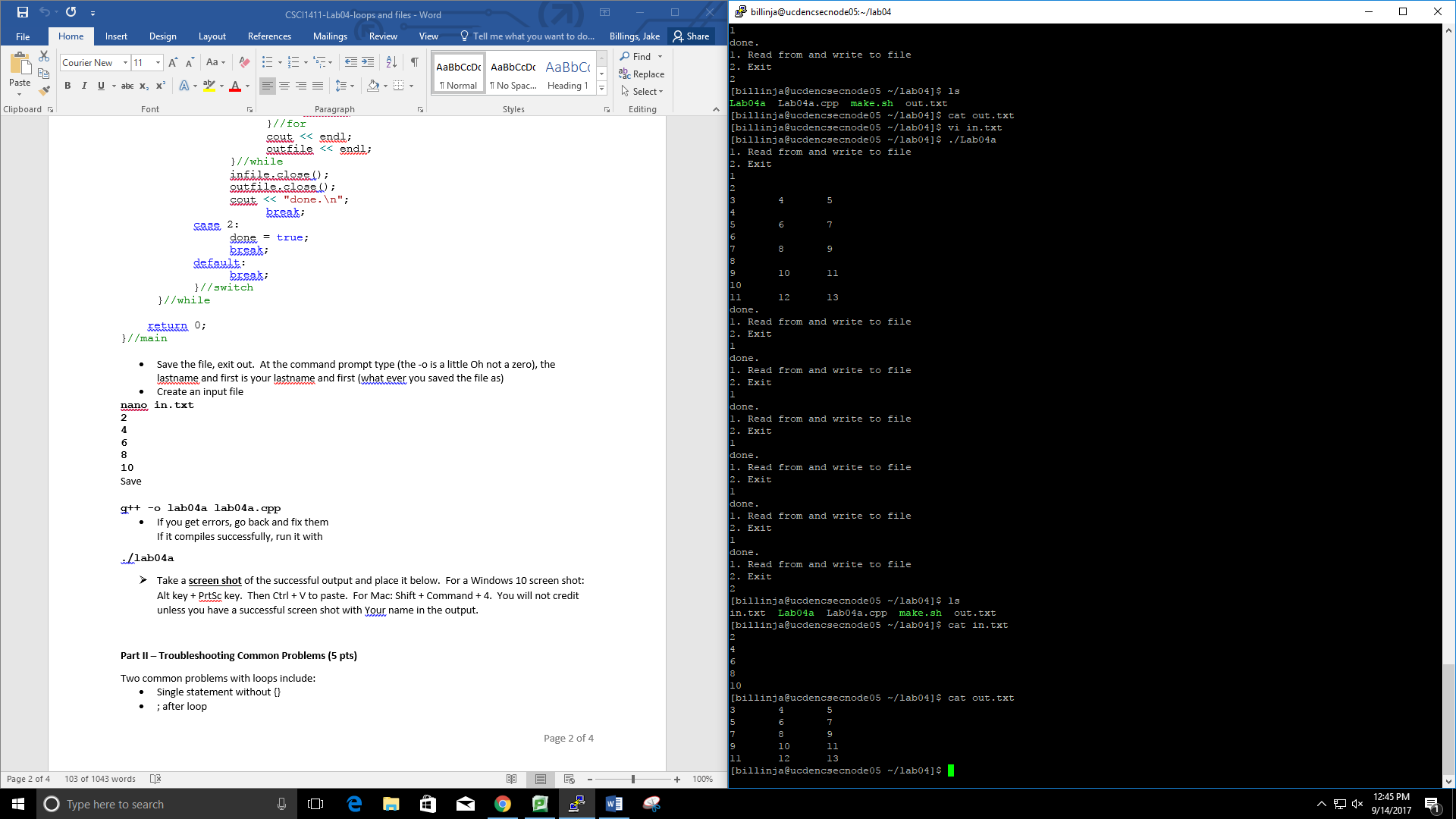
Save

**g++ -o lab04a lab04a.cpp**

* If you get errors, go back and fix them  
  If it compiles successfully, run it with

**./lab04a**

* Take a **screen shot** of the successful output and place it below. For a Windows 10 screen shot: Alt key + PrtSc key. Then Ctrl + V to paste. For Mac: Shift + Command + 4. You will not credit unless you have a successful screen shot with Your name in the output.



**Part II – Troubleshooting Common Problems (5 pts)**

Two common problems with loops include:

* Single statement without {}
* ; after loop
* Ensuring loop variable is initialized and changes.

**nano lab04b.cpp**

#include <iostream>

using namespace std;

int main()

{

for (int i = 0; i < 5; i++);

cout << "the value of i is ";

cout << i << endl;

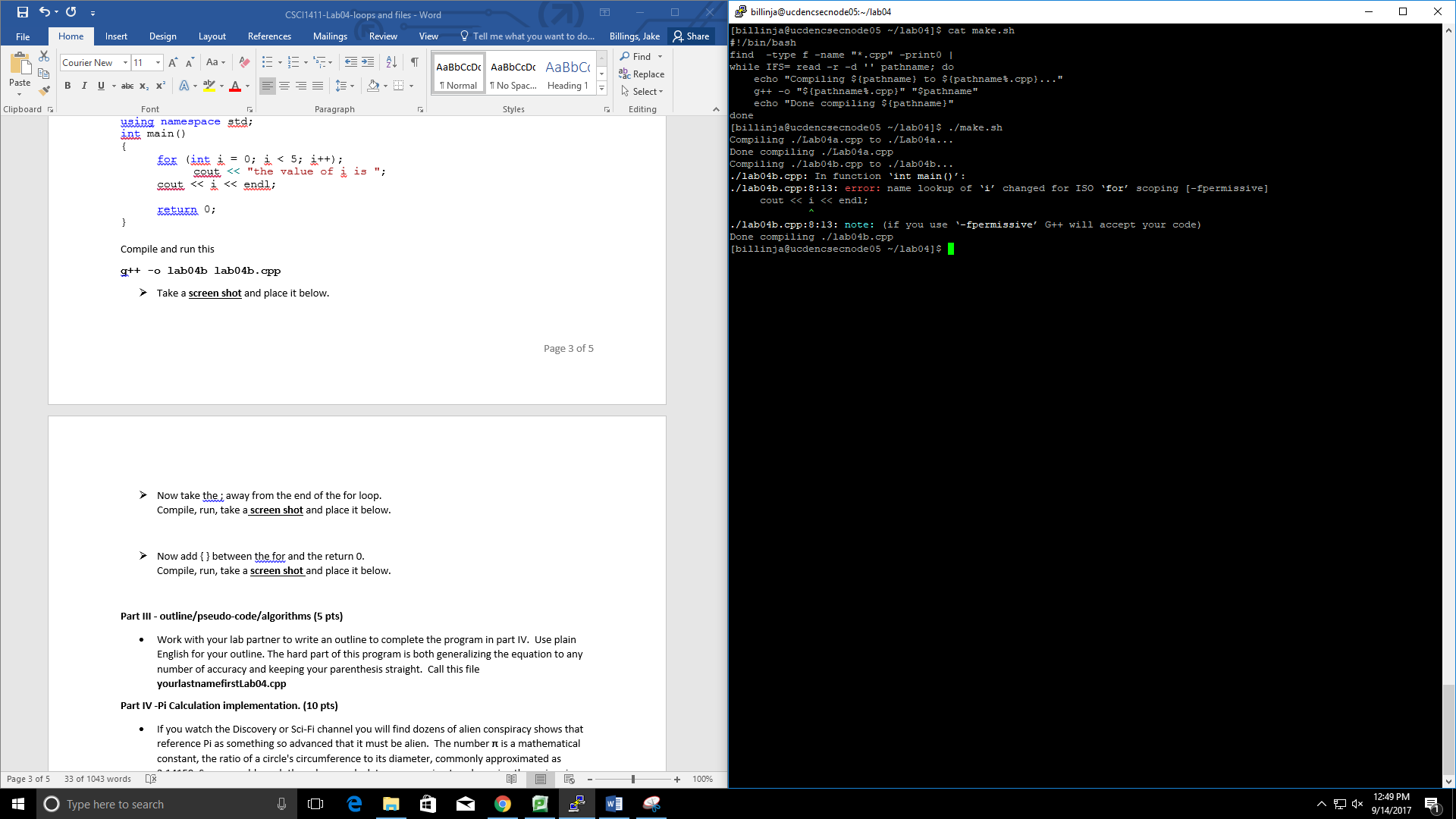
return 0;

}

Compile and run this

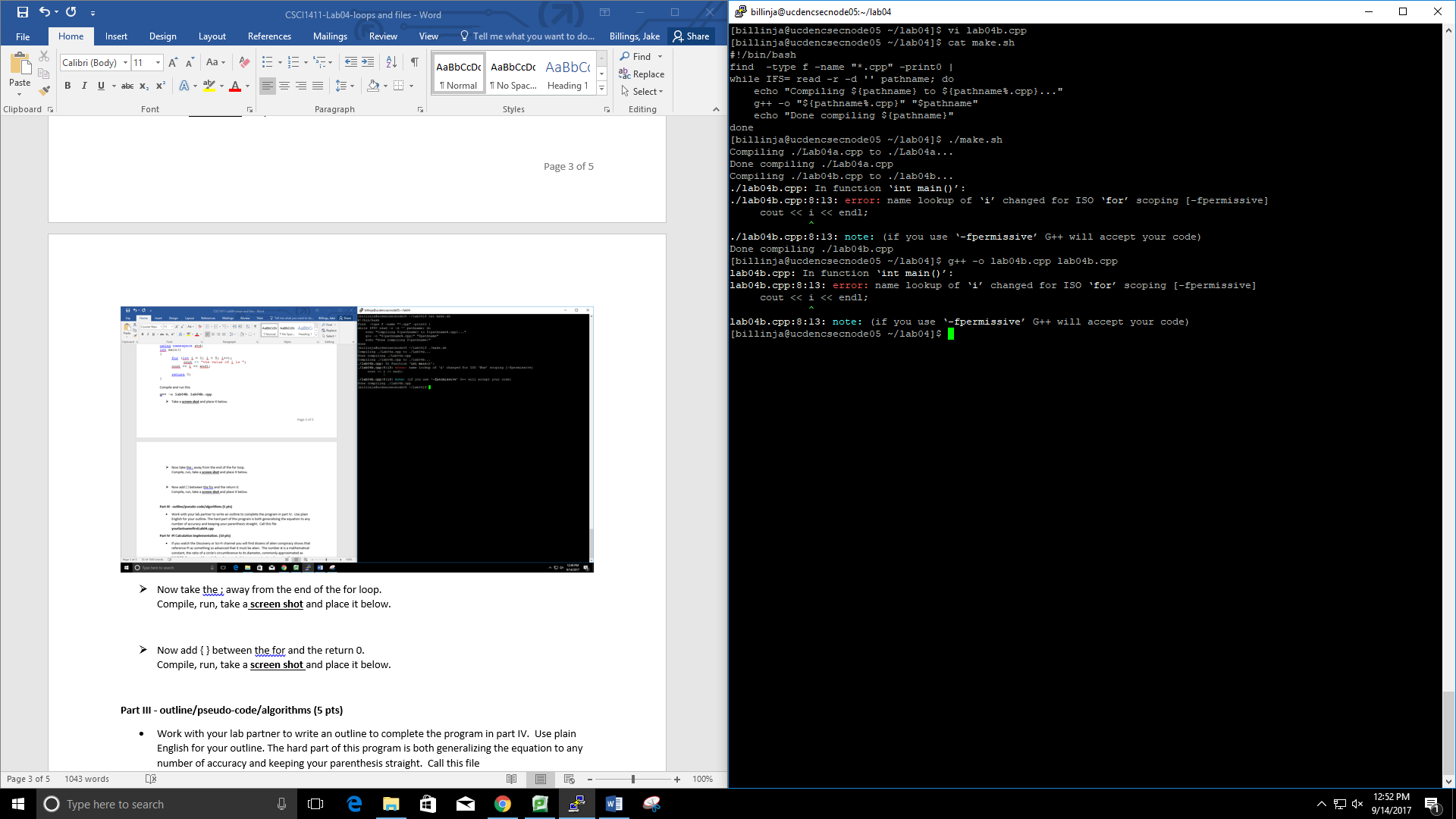
**g++ -o lab04b lab04b.cpp**

* Take a **screen shot** and place it below.



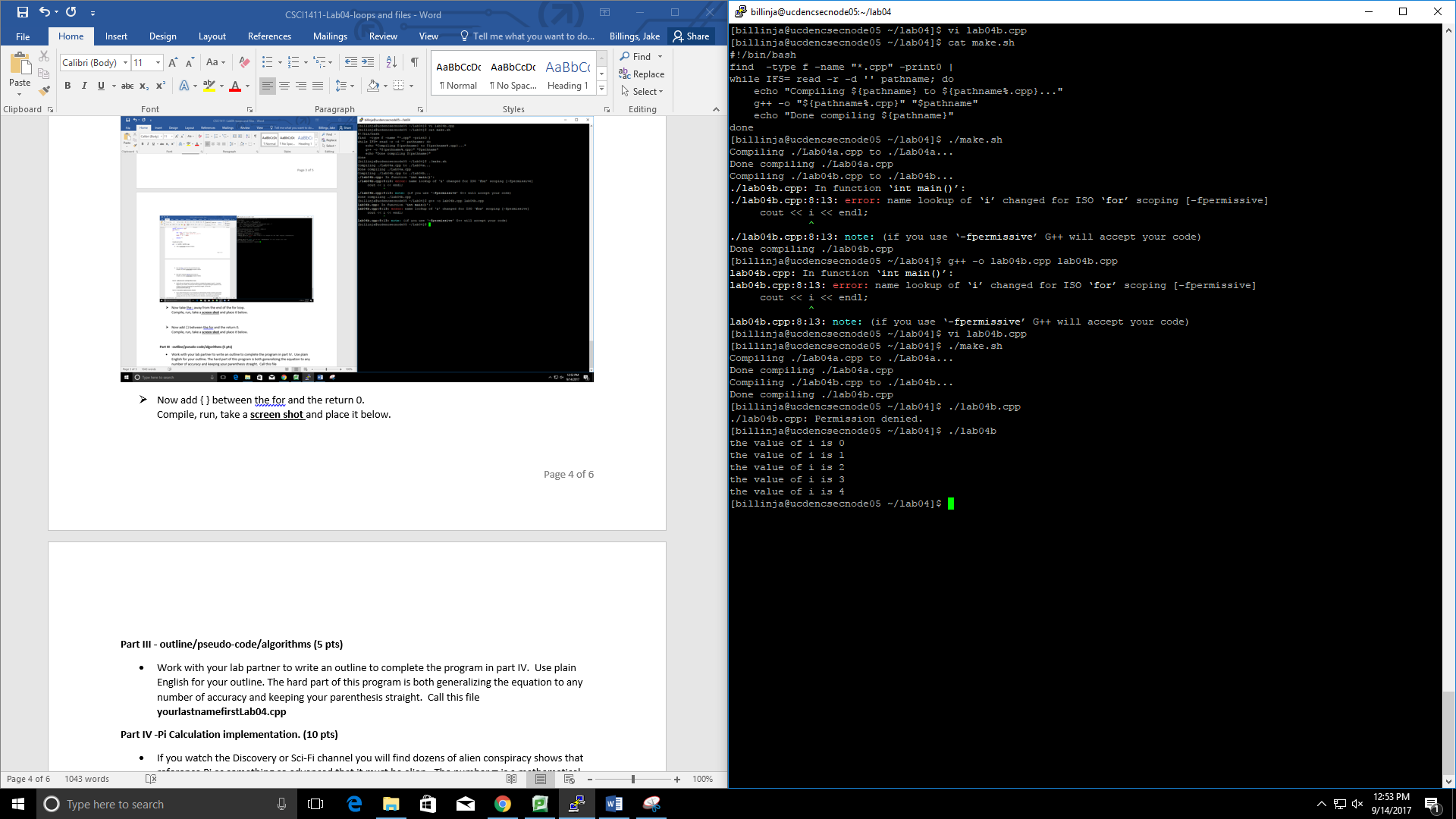
* Now take the ; away from the end of the for loop.

Compile, run, take a **screen shot** and place it below.



* Now add { } between the for and the return 0.

Compile, run, take a **screen shot** and place it below.



**Part III - outline/pseudo-code/algorithms (5 pts)**

* Work with your lab partner to write an outline to complete the program in part IV. Use plain English for your outline. The hard part of this program is both generalizing the equation to any number of accuracy and keeping your parenthesis straight. Call this file **yourlastnamefirstLab04.cpp**

**Part IV -Pi Calculation implementation. (10 pts)**

* If you watch the Discovery or Sci-Fi channel you will find dozens of alien conspiracy shows that reference Pi as something so advanced that it must be alien. The number **π** is a mathematical constant, the ratio of a circle's circumference to its diameter, commonly approximated as 3.14159. So you could graph the value or calculate an approximate value using the series given below:

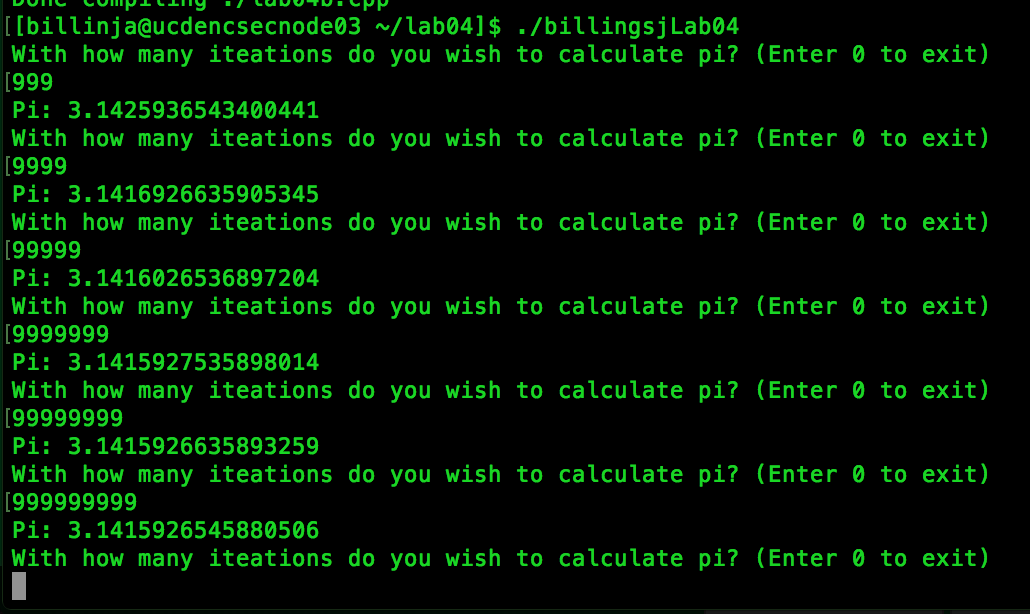
**Pi = 4 \* (1/1 – 1/3 + 1/5 – 1/7 + 1/9 … + (-1)^n / (2n +1) ).**

* Write a c++ program to calculate the approximate value of pi using this series. The program takes an input n that determines the number of values we are going to use in this series. The above series uses 5 terms (1, 3, 5, 7, 9). Then output the approximation of the value of pi. The more values in the series, the more accurate the data. Note 5 terms isn’t nearly enough. Try it with 1000, then 10,000, then 100000 to see the accuracy increase. Include a loop that allows the user to repeat this calculation for new values n until the user says they want to end the program.
* Call the program **yourlastnameFirstinitialLab04.cpp** (e.g. augustinetLab04.cpp). Every program you write should have the following block at the top in comments. Make sure to fill in the Name, Class, Description and Lab Partner. Ensure your status is accurate.

/\* Name:  
Class: CSCI 1411-00X  
Description: [fill in description]

Lab Partner:  
Status: successfully compiled and run on csegrid [if it doesn’t run or meet all of the requirements, list the actual status!/\*

* You should work with your lab partner on this, but each of you must write your own code.
* Implement the program in part I, placing the actual code under each appropriate line of the outline.
* **Implementation Detail 1:** You will want to use a double instead of a float, to get twice the decimal accuracy
* **Implementation Detail 2**: Remember that an integer divided by an integer is of type integer. Make sure you use at least one decimal point in your division (like 1.0/9) or double(numerator)/denom
* **Implementation Detail 3:** The … is not a mathematical symbol. Instead, you should figure what is the next in the series. Note that the denominator starts at 1 (1/1) and increases by two for each iteration. The numerator starts positive (1/1) and alternates each time to negative. Think of what you could multiply by the numerator to alternate between positive and negative. Remember, not every for loop starts at one, and not every for loop ends in i++!
* Implementation Detail 4: The default for the cout of a double is 5 places of accuracy. If you want to increase that you would use setprecision. <http://www.cplusplus.com/reference/iomanip/setprecision/>
* Ensure the program runs, take a **screenshot** and place it below



* Hand it in the yourlastnamefirst.cpp complete with comments from part III. Hand in this lab with 5 screenshots attached