Don’t forget to create readme and makefile.

**Step1: Basic C++ (10 points)**

Practice: g++ compiler and playing with iostream libraries.
Here I want you to get practice with the g++ compiler. Please take a look at the class notes and create a simple C++ program to do the following:

You will ask the user to input a list of numbers. We don’t know how many numbers will be inputted, so you will need to dynamically allocate memory to store the numbers as you go. When the user inputs a blank (this will take some thinking on how to do it…..) you will output the following information:

1) number of integers inputted  
2) sum of all the numbers  
3) maximum number  
4) average of all the numbers

Note: there is no reason negative numbers can’t be inputted.

Remember we are dealing with cout and cin…. 

**Step 2 – Basic C++ Classes (10 points)**

In this step we will be practicing your C++ skills. We are creating a basic StringList class in C++: do not use the string class from the std c++ libs, reuse the string2 class we did in class 😊

Main.cpp – the file which will have your main test routines  
StringList.cpp - the cpp class code for the StringList class  
StringList.h the cpp header info for the class StringList 

Please don’t forget to sprinkle comments liberally in your code, it will only help your grade.

You should be familiar with the idea of a String class and String list from java. We did the string2 code in class, so you know what a string class looks like. You might need to
adopt the code from class to fit the assignment. Create a primitive StringList class in cpp, with the minimum following functions:

1) Constructor for taking a character pointers (c string) and inserting it into your list. The string internal class does deep copies (You need to allocate memory and copy data into your own data representation) which means clean up after yourself…..
2) Destructor in each class to clean up after yourself
   we have memory allocated, so don’t forget to delete it
3) Insert function to insert c strings into the list
4) Find function to return 0/1 (false/true)
   if the String passed in exists in this list
5) Size function to return number of items in our list.
6) Print function to print out the contents of the StringList in some way

Summary: we want to represent a bunch of strings as a list and print them out

**Step 3 – More advanced practice (10 points)**

In addition to what we’ve done in last step, we would like to overload the

1) `<<` (left shift)
2) `+` operator
3) `!=` operator
4) `=` operator (remember its not just sizes but also items that you might need to compare

I would like to be able to code the following code:

```cpp
StringList test;     //default constructor triggered here
StringList test2;
    test.insert("Hello!");
    test2.insert("Bye");
StringList test3 = test + test2;
    //Hint: a constructor is called here.

if ( test != test2 ) {
    cout << "the lists are not equal" << "List 1 is " << test << "List 2 is " << test2 << endl;

    cout<<"list 3 is now: " <<list3<<endl;
}
```

Hint: Look up the signature of each operator in order to overload it. For example overloading the left shift operator the following way:

```cpp
friend ostream& operator<<( ostream&, const StringList &); 
```
and define the following method:
ostream& operator <<(ostream &os, const StringList &obj)
{
    // your code goes here
    // basically loop through the items
    return os;
}

Hint: do each operator one at a time, and test before starting on next, will make the lab easier for you.

friend String operator+(const StringList& s1, const StringList& s2);