Joshua Pollock

CS 200 Project 6

31 March 2017

Joshua Pollock

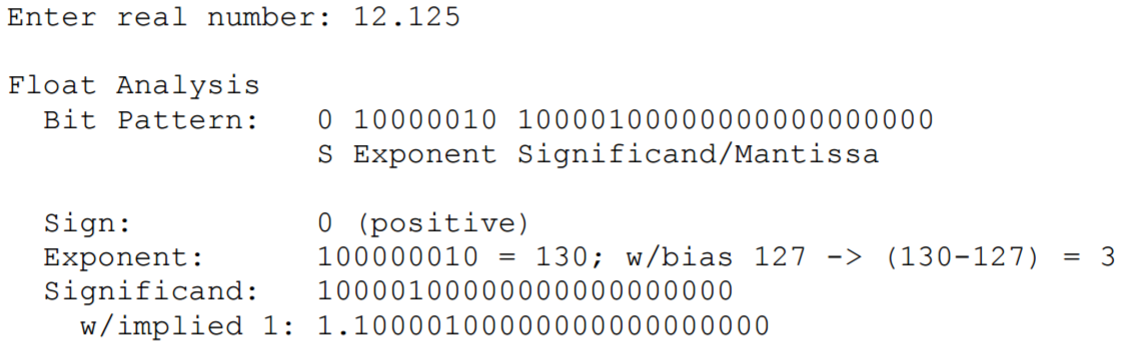
CS 200

Project 6 – Float Analysis

31 March 2017

**Purpose:**

The purpose of this project is to create a C++ program that asks a user to input a float number, and then output the floating-point representation. This project will also help us to practice using bitwise operators. The given example was this:



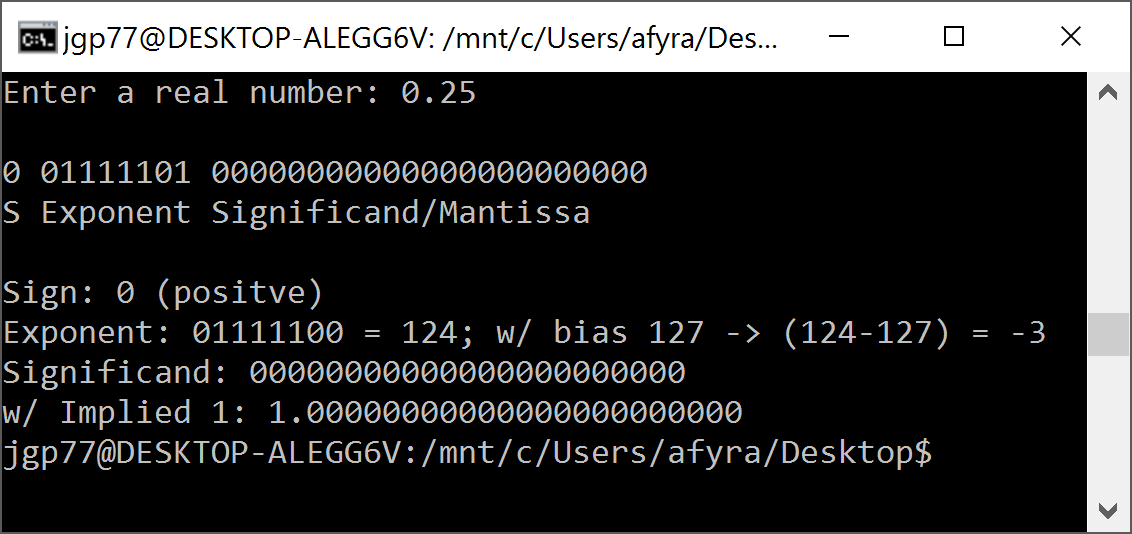
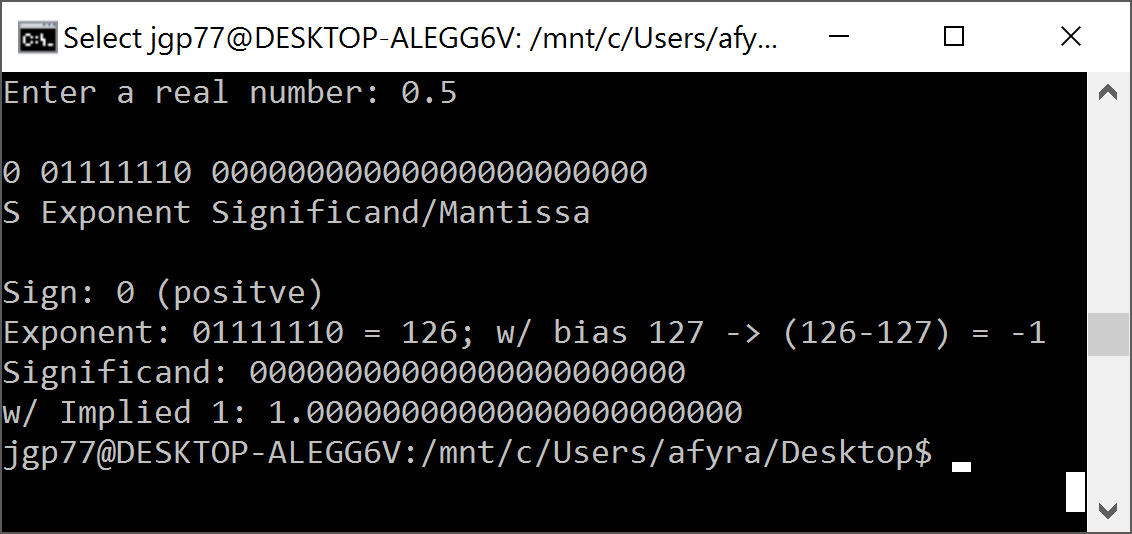
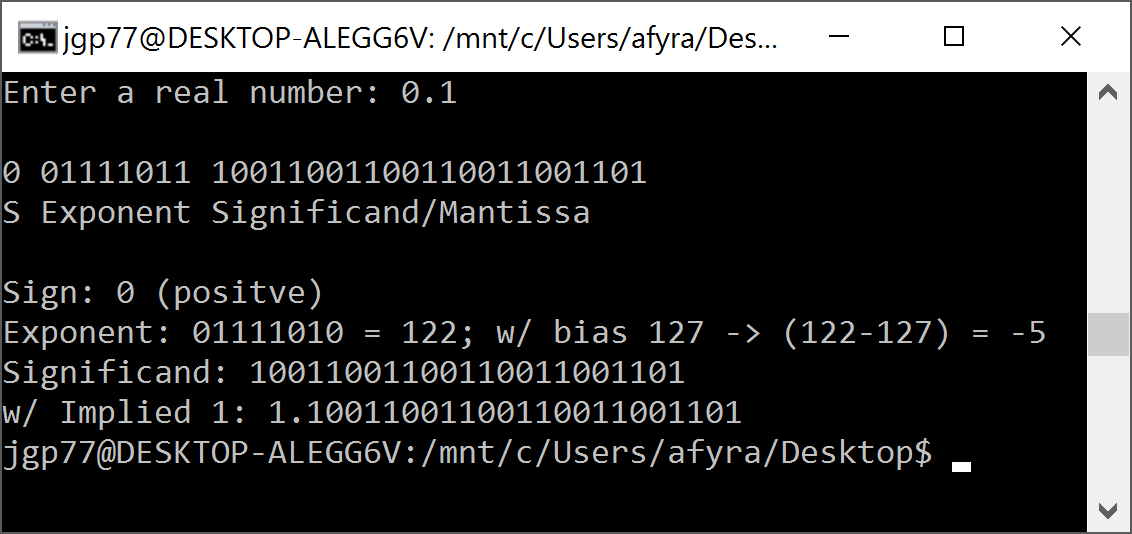
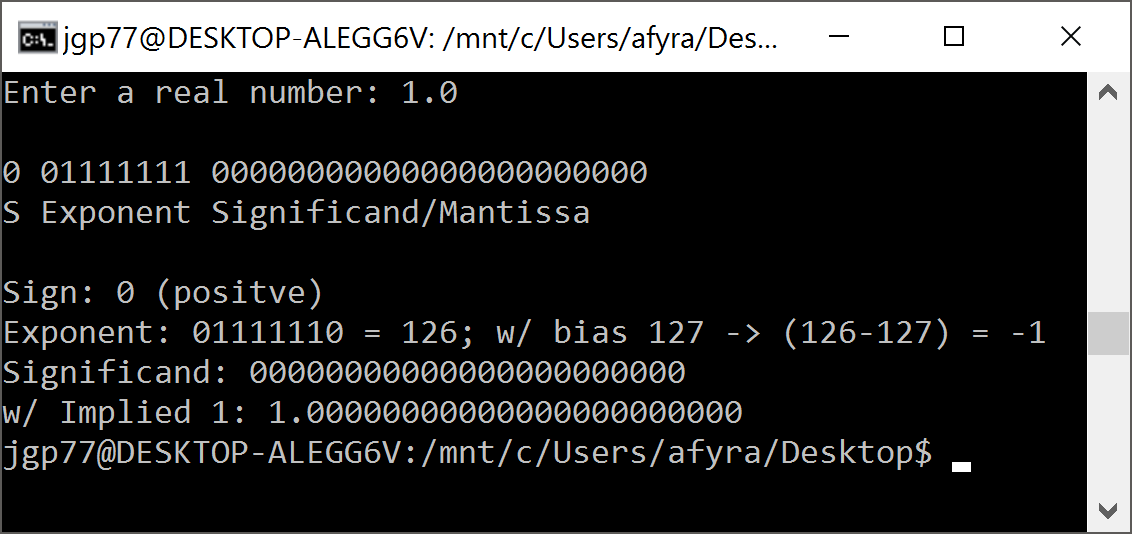
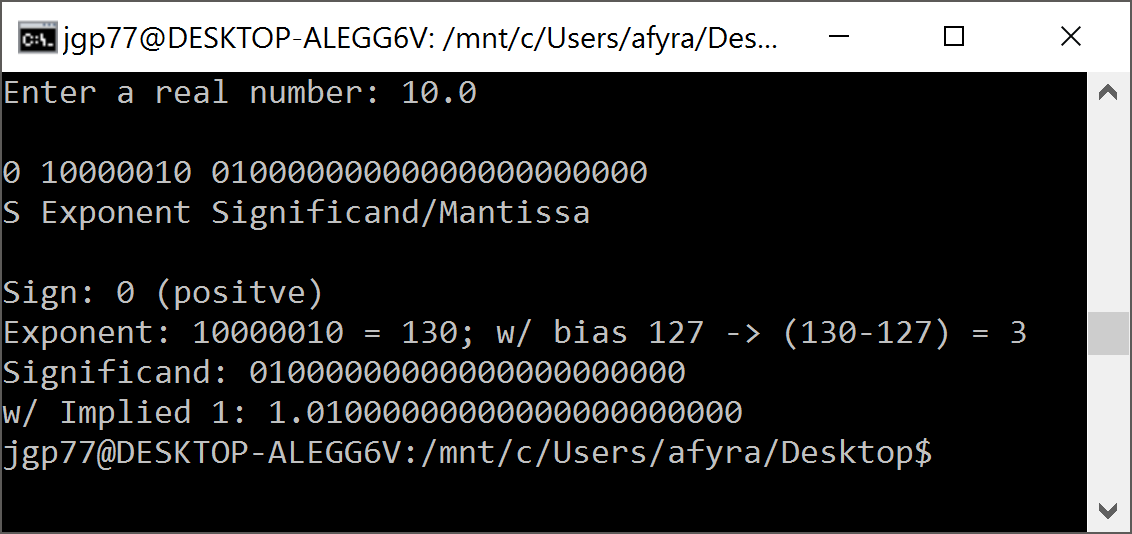
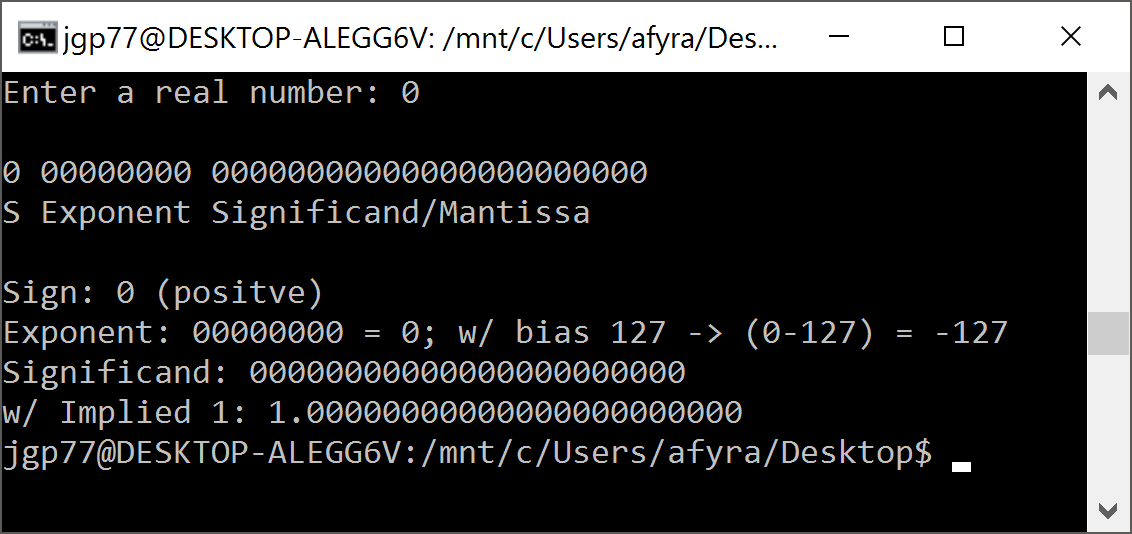
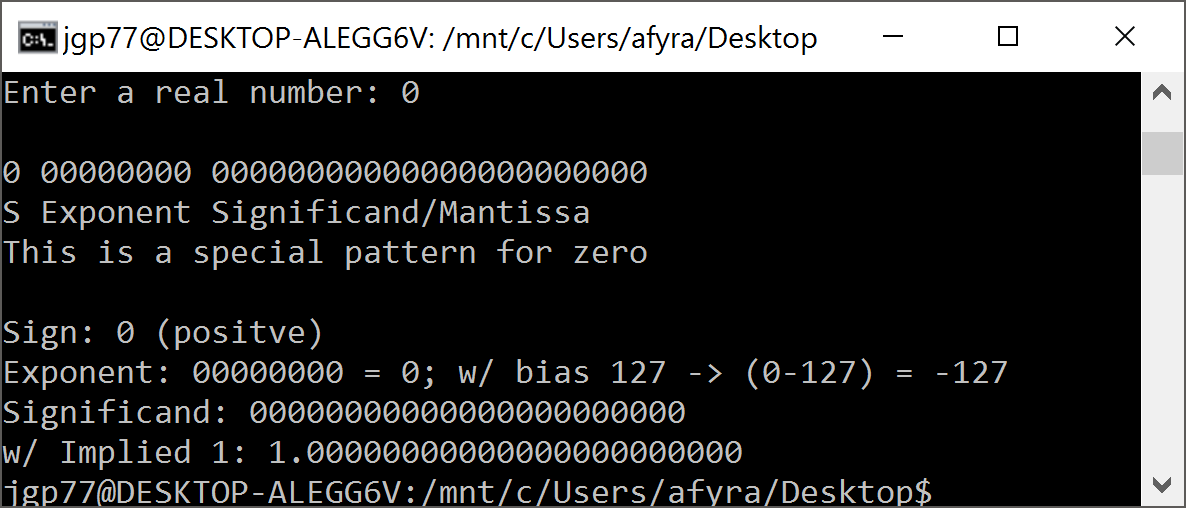
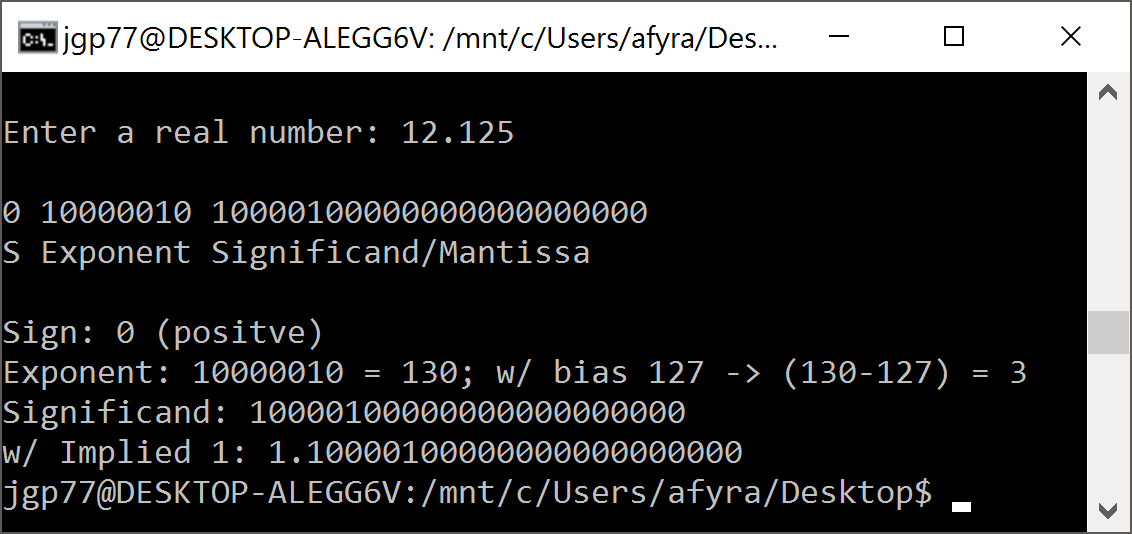
Even though we do not have to follow this style, I plan to have my code display almost exactly like this. The project is not focused around converting numbers to or from floating points, but to use bitwise operators on an existing floating point. Our program should also be able to identify the case of +/- zero through a print statement. The test values for this program are 12.125, 0, 10.0, 1.0, 0.1, 0.5, and 0.25.

**Research:**

One of the first things I looked at was the “Data Representation in Computer Systems” PDF file on BBLearn. This was the lesson where we learned about floating-point representations. I wanted to look this over to just refresh my mind on floating-points. After looking over the PDF, I began to input the given code from the Project 6 PDF. There wasn’t much code here but it did contain some helpful lines and examples, such as “exponent=(bits>>23)&0xff;” and “int bits=\*((int\*)&f);”.   
  
 During my coding process, I had to look up how I could print out an array. I found two methods and ended up using the one from <http://www.cplusplus.com/doc/oldtutorial/arrays/>. On this website, they had a method that would allow them to pass in a length and array, and print it out using cout<<. I implemented and edited this so that it would fit my needs in the program. The other method that I could use was something called stringstream. Stringstream would fit my program but was far too complicated to implement.

**Program:**

Creating the program for this project was much easier than creating the code for the last project. At first I forgot to check to see if the number was 0 or -0, but I did end up going back and creating a method in order to check the arrays. This method was quite easy to create. It takes in two arrays along with their lengths, and the sign of the floating-point. I implemented a method instead of going back and editing the main function to avoid creating possible errors within my working code. In the end, I was able to get the program to detect 0 and -0 to satisfy the project requirements.



**Conclusion:**

I greatly enjoyed this project. It was much easier that the project from last week, and helped to clear up many things I was needed improvement on or uncertain about in C/C++. I’m much more confident is using the simple input/output like: cout, cin, and printf. With researching into arrays, I have an understanding about how to use them more efficiently in my coding. This project also helped to increase my knowledge of bit manipulation.   
  
 In the end, I was able to complete all requirements given by this project. I did not attempt the extra credit because I did not understand it. My code was compiled on a Windows 10 machine using the Unbuntu terminal with the command “g++ Proj6.cpp” and “./a.out”.