Jason Dorweiler
CS165 Assignment 1

Requirements:

Here's a list from each requirement for what I think needs to be done:

2.a Write some code to convert decimal number into binary and hex

2.b Write some code to convert binary numbers into hex and decimal

2.c Using the climits library, set variables to the max and min of each data type.

2.d Print out the variables from 2.c in decimal, hex, and oct.

2.e Setup user input to read in an int and then print it back out in hex and binary.

2.f Print out the max limit variables + 1, they will overflow.

2.g Add a header to the top

Possible solution:

For 2a converting to binary and hex can be done using baseflags.

I'm not really sure about 2b. There is probably a library that will do this. Or should I write a separate binary to hex/dec function?

Part 2c should be easy enough to just include the header file and assign variables to the max and min from the climit library.

Part 2d. I can just use the baseflags again.

Part 2e. Add user input using cin >>

Part f. Use cout << to print out the variable + 1. They will overflow since this is one greater than the max value.

First pseudo code implementation:

```
#Include the required libraries ( iostream, climits ….  )

int main()
{
        // use bitflags to convert the decimal numbers to hex
        cout << “decimal to hex” <<  bitset<8>( 96 );  // Then repeat this for all the decimal numbers in the problem.

        // Binary to hex, decimal
        I'm need to think about this one more...

        //use the climit macros from the class video.  For example assigning a minimum short.  Do the same for all the
        //other ones.
        shortInt = SHRT_MIN

        // Adding user input
        cout << “enter a number”
        cin >> number_entered
        cout << “you entered” << number_entered  // then print out using hex and baseflags for binary
```

```
        // add 1 to each max and they should just overflow
        cout << SHRT_MAX + 1 // repeat the same for the others.
}
```

Implementation

See comments in the source file.

I didn't see that part 2a,b had to be done by hand until after had already got it working in my program. I included another pdf file that had my hand calculations for this problem. I left the code as-is so it still does the calculations for me.

Testing and Debugging

The only input is for the one part where a user enters a number to be converted into hex and binary. The way I set this up is to convert using and 8 bit bitset<8>. Numbers that are more than 8 bits will only display their first 8 bits and ones greater than that are cutoff. Floats will get cut off at the decimal point. Anything entered other than in integer will result in a zero being printed back out.

I did some debugging with the binaryToDec function using print statements. I left the debugging line there to show this. I was checking the value of the counter and value of the intermediate results. I used this to figure out what needed to go in the pow() function to get the correct answer back.

In figuring out how the pow() function worked I did a bit of debugging by printing out the sizeof() each variable and doing the calculation by hand.

I tested the code on both my computer and the school's just to be sure the output was the same on both.

Reflection

Some parts of this assignment were pretty easy and others not so much. I though the assignment was asking for me to write a function to do the binary to decimal conversions. It wasn't until I had already spent a lot of time working on it that I saw that we were really only suppose to do it by hand.

I had at lot more trouble figuring out how to convert the binary numbers into other types than I though I would. I looked through all the course notes/videos and didn't see any way to do this. Maybe I missed it? I did some searching online and ended up using a function from a forum, which didn't fully work anyway. After a bit of debugging and more searching I finally got it working.

I added in the calculation for the max variables using the pow() function that I saw in one of the videos. This also took a bit of work just to figure out how it was working but I did that just for fun. The climits library has macros for these.