CS221 - Lab05

Paul Haskell

INTRODUCTION

In this lab you will work with binary and hex. This lab is fairly short, since we will kick off of our first large project next time.

Program for this Lab

Your only program for this lab is called **conversions.c** . This program reads a string from the first command-line argument. The string contains a hexadecimal number without a starting "0x". The letters 'a' through 'f' might be represented in uppercase or lowercase, so you might want to use your `tolower()` method from the previous lab. Your program should convert this string of hexadecimal digits into an `unsigned short` numerical value.

Now multiply that unsigned value by -1 and save the result to a `short` . Finally, print out the `short` in binary, as a sequence of 16 '0's and '1's. Hint: if the `short` value is negative, then the first binary digit should be '1'.

Approach

How to approach this? First, let's look at the string-to-unsigned short conversion. What are some very simple cases to handle? Single-digit cases, e.g:
4
B

Can you make those work? How will you handle two-digit cases, e.g:
10

This should have a decimal value of 16. "ff" should have a value of 255. If you can make these cases work, can you handle longer inputs?

Ok, how to handle the conversion of a `short` value into binary digits? The bitwise operators will help. How can you look at the value of the least significant bit?

```
value & 1
```

How about the next-least-significant bit?

```
value & 2 (or value & (1<<1)
```

I will not test with inputs larger than 15 bits, and the hex input cannot be negative.

CONCLUSION:

Please push the program to GitHub before the lab deadline. The deadline for completion of Lab05 is 11:59pm Wednesday February 12.

Task	Score, points
conversions.c gives correct output values for each of 5 test inputs	20
Software and design quality as judged by the grader	10