Homework #1 – Binary / Octal / Decimal / Unsigned / Hex

CS 3844 Section 0B1, Fall 2020, Dr. O'Hara

Write a C program that takes one or more numbers on the command line. Rules:

* If no numbers are given, print a usage message and quit.
* Do NOT use sscanf or similar. Process each argument one digit at a time.
* For each number given on the command line, check to see if it is a valid binary, octal, decimal and hex number. For hex numbers, allow both lowercase a-f and uppercase A-F.
* If there are any invalid characters in the argument, ignore it. E.g., a 2 is not valid for binary.
* You must also account for a leading minus sign. Additional error checking is not necessary (overflow for example).

Example usage: **./hw1 0 123 abcd -45 error -1 -4D**

Example output:

**=== Checking 0 Octal Decimal Unsigned Hex**

**Base 2: 0 0 0 0**

**Base 8: 0 0 0 0**

**Base 10: 0 0 0 0**

**Base 16: 0 0 0 0**

**=== Checking 123 Octal Decimal Unsigned Hex**

**Base 8: 123 83 83 53**

**Base 10: 173 123 123 7b**

**Base 16: 443 291 291 123**

**=== Checking abcd Octal Decimal Unsigned Hex**

**Base 16: 125715 43981 43981 abcd**

**=== Checking -45 Octal Decimal Unsigned Hex**

**Base 8: 37777777733 -37 4294967259 ffffffdb**

**Base 10: 37777777723 -45 4294967251 ffffffd3**

**Base 16: 37777777673 -69 4294967227 ffffffbb**

**=== Checking error Octal Decimal Unsigned Hex**

**=== Checking -1 Octal Decimal Unsigned Hex**

**Base 2: 37777777777 -1 4294967295 ffffffff**

**Base 8: 37777777777 -1 4294967295 ffffffff**

**Base 10: 37777777777 -1 4294967295 ffffffff**

**Base 16: 37777777777 -1 4294967295 ffffffff**

**=== Checking -4D Octal Decimal Unsigned Hex**

**Base 16: 37777777663 -77 4294967219 ffffffb3**

The columns must line up exactly as shown, but the number of spaces between columns doesn't matter.

Hint: in C, printf with %-10s will left justify a string in 10 characters, %10s will right justify it.