CS 240 - Homework 3

Assigned: Monday, 2.13.17	Due: Wednesday, 2.22.17

Make a subdirectory "hw3" in your cs240 folder for this assignment and copy the files from /courses/cs240/s17/jmcurran/GROUP/hw3.

The objective of this assignment is to familiarize you with integer and character string manipulation. You are not allowed to use any string manipulation library functions. Use the gdb debugger to help you troubleshoot your program.

INTRODUCTION TO HEADER FILES

This project provides a basic example of a header file, implemented for you.

Its use is explained in the comments. We will cover header files in more detail in the future, but here's brief overview for now.

numReader.h	Contains constants and function declarations for numReader.c
numReader.c	Contains definitions for the functions declared in numReader.h. Contains the line: #include "numReader.h" This declares the functions declared in numReader.h in numReader.c, where they are then defined.
sumLines.c	Uses the functions declared in numReader.c. Contains the line: #include "numReader.h" This makes the functions declared in numReader.h and defined in numReader.c available in sumLines.c. We must make the definitions available to the compiler, so we compile sumLines.c as follows: gcc sumLines.c numReader.c -o sumLines

Note:	The showxbits program should not prompt the user for input.
	"No prompt" is a UNIX convention for reading files from stdin. There's a good reason for the convention. It allows you run showxbits easily and cleanly either by typing in the input data or using input redirection.

numReader.c

Your main task in this file is to define the function lineValue. You are provided a stub program with includes notes about what this function does and hints about how to implement it.

There are several stubs for functions you may find helpful in implementing lineValue.

lineValue is used in sumLines.c to read an input file, convert the number on each line of the file to an integer, and then print out the sum of the integers.

For example, if the body of an input file input.in is:

```
10
0X14
invalid line
0XG
024
```

Then sumlines < input.in should print:

```
Total is: 50
```

This is because lineValue should process the lines of input as follows:

```
10            // decimal, return 10
0X14          // hex 14, i.e., decimal 20, return 20
invalid line  // invalid input, return -1
0XG           // invalid input, return -1
024           // octal 24, i.e., decimal 20, return 20
// 10 + 20 + 20 = 50
```

numReader.h

You do not need to edit this file unless you would like to write additional helper functions for your lineValue implementation.

This file provides the function declarations for lineValue and a few helper functions that may be helpful.

When you include this header in a source file, you are declaring all of these functions in that file, making them available for your use there.

sumLines.c

You should not edit this file.

sumLines.c makes use of the functions provided by numReader. This type of file is sometimes called a driver (not to be confused with hardware drivers).

You can test your lineValue implementation by compiling sumLines.c with numReader.c:

```
gcc sumLines.c numReader.c -o sumLines
```

and then running it with the provided input files:

```
sumLines < test1.in
```

You can then compare your output to test1.out

TEST FILES

I have included a directory containing three test files. You can test by copying your sumLines executable to the test directory, running it on the three .in files and checking your output against the three .out file.

DELIVERABLES

When you complete this homework, there should be 3 files in your hw3 directory:

- numReader.h
- numReader.c
- sumLines.c

START NOW AND ASK QUESTIONS ABOUT ANYTHING YOU DON'T UNDERSTAND