CS-241: Data Organization
Fall 2020

Programming Assignment #3
Due by 11:59 PM on Sunday, September 13

This assignment will allow you to practice using the bitwise Boolean operators in C. Log into the host **linux.unm.edu.** Create a directory for this assignment as follows:

```
$ cd cs241
$ mkdir prog03
$ chmod 700 prog03
$ cd prog03
```

You can copy all the sample source code files related to the topics used in this assignment using the following command (do not forget to include the dot symbol at the end of this command):

```
$ cp ~soraya/cs241_Soraya/notes/wk03_modA/* .
```

You can also copy the files needed for this assignment using

```
$ cp   ~soraya/cs241_Soraya/programs/prog02/handouts/* .
```

Hashing is a technique to radically compress data into a single integer. A hash code for a file is equivalent to the fingerprint of a human being. A fingerprint does not give you much information about the entire person, but fingerprints can be used to distinguish between two different people. In the same way, a file (or any source of data) can be associated with a 32-bit **hash code**. A good hash code should be a pseudo-random number computed from the given data. To create a hash code, we basically want to "mash" all the bits of the data into a single int.

You can use the posted (with this assignment on Learn) **visualization.jar** file to see how this hashing process works on String data.

- Open the jar file by double-clicking on it
- Click on **Algorithms** along the top
- Click on **Hashing** in the drop-down menu
- Click on the **Hash Strings** button on the right
- Click on the **Pause** button in the lower left
- Type the string "alice" (without the quotes) in the TextField to the left of the Insert button
- Click on **Insert** button

You can now watch the computation of the hash code step-by-step by repeatedly clicking on the **Step** button in the lower left. Each of the five characters of "alice" are used, in turn, to contribute to the final value of the hash code. The process repeatedly adds the 8 bits of the ASCII code of the next character into the running total. Remember, a **char** is really just a small number. The running total is then shifted 4 bits to the left. The 4 bits that are shifted-out of the running total are then XORed into the running total at bit positions 18-21 (counting bit positions from the right, starting with position 0). This shift-and-XOR is **not** performed for the last character of the input string. The hash value of "alice" is 6,827,925.

Notice that the input string "alice" is very short. But this same process can be used on a file containing billions of characters/bytes (such as a video file). Consider the given file gettysburgOriginal.txt. You can hash (aka compress) this very large (OK, not so very large) input into a single integer using the command:

```
$ ./a.out $(cat gettysburgOriginal.txt)
```

The process of hashing is deeply connected to the concepts of bitcoin, blockchain, and digital signatures. If you change the input data in any way, even if only in a small way, then the resulting hash code will be completely different. Compare the hash codes of gettysburgOriginal.txt and gettysburgAltered.txt. The main purpose of blockchain is to be able to create signed contracts that can't be forged or repudiated or altered after the fact.

You should modify the given source code file **hash_handout.c** to carry out this hash computation. You should only need to write a small amount of code. Place your code in the section indicated in the file **hash_handout.c** Do not modify the source code in any other way.

If at any time when you are running your program, and you are "stuck" (the program has entered an infinite loop, or is unresponsive) you can "bail out" and halt the execution of the program using ^C (i.e., hold Control and C simultaneously).

You should be sure to include **YOUR NAME** in a comment at the top of your source code file, and make sure your code follows the code standards for this course (check out the standards in the cs241_codingStandards2020.pdf).

Rename your modified hash_handout.c file to a name that uses your last name and the initial of your first name, like this: **lastName_initialFirstName_hash_handout.c**
Submit this file for grading to Learn in the place of this assignment.