**CS4732/5732 Cryptography Summer 2019**

**Project #3 [75 points]**

---------------------------------------------------------------------------------------------------------

Due date is Thursday, August 8th

**Overview:**

In this project you will be working with hashing functions, getting more familiar with using them in code.

**Task 1:**

For this task, I want you to work with SHA-256, a common cryptographic hash function. I want you to generate several files of different lengths (one of them very short, shorter than the output of 256 bits), another at least 1Kbyte. Document how many hashes you can do per second on your machine for each of the files. Compare this to another hashing function implemented in your library. Calculate how long it would take to find a collision through brute-force to a particular value for SHA-256 and your other algorithm. This time might be large. Document these results, including the hardware specs of your system at least to a limited degree.

**Task 2:**

In this task I want you, again working with SHA-256, to try and generate a specific hash. Your task is to start with 256-bits all set to 0. Your goal is to try and get as much of your birthdate as you can as the most significant hex digits of the hash. So for example, if my birthdate was 02231947, then I would want to find a hash starting with 02231947. I could do this by hashing my initial all-zeros, then seeing if the most significant digits (in hex) started with 02231947. Note, to get a hash equal to the entire birthdate would take a silly amount of time depending on your machine. So start looking for one that just has the first hex digit correct. Then try for a second digit or simply extrapolate from how long it took to get the first one. Document your results and logic.

Task 3:

Now let us consider if we wanted to find a SHA-256 hash that was simply less than a particular value. To experiment with this, I want you to take some string, up to you, and hash it. Then vary the string and hash it. Do this continually for 10 seconds and document the lowest hash produced. Repeat this process 10 times, to give an average of the lowest number gotten in 10 attempts. Then repeat this for 20 seconds. Then repeat for 30 seconds. Once you have done this, try and calculate a formula for the expected lowest value you would get given X seconds. Obviously, there is randomness in this, but document your thought process on your conclusions.

**Submission:**

For turnin I want a text document or pdf explaining in detail all information you learned and the steps you completed. So initial environment chosen, any source code, results obtained in the various tasks, etc.