**Ground Rules.** You may choose to work with one other student if you wish. Only one submission is required per group, please ensure that both group members names are on the submitted copy. Work must be submitted in hard copy by the start of class on October 27, 2017. Questions marked as COSC 583 are required for those enrolled in the 583 section. If you are in the 483 section you are more than welcome to attempt any of them, extra credit will be awarded for correct answers.

1. **Implement Free Collision Detection via Tortoise and Hare**
   You should find an untargeted collision in a hash function via the tortoise and hare algorithm. In other-words you will find two distinct values, X and Y such that \( H(X) = H(Y) \). Since these are cryptographically secure hash functions this requires a rather exhaustive search, to simplify this you will be hunting for partial collisions in the first \( n \) bits of the hash values. The larger \( n \) is, the more computations you will need to compute. You are required to succeed at an \( n \) of 16, 24, and 32, if you can succeed with \( n \) larger than 40 you will be eligible for bonus points (see: competition).

   The attack you will be implementing to find the collision is the Small Space Birthday Attack attack described in section 5.4.2 against the SHA-2 hash function. Remember, we are only looking for a collision in the first \( n \) bits! This means, as we’ll talk about in class, you only want to carry forward the first \( n \) bits each time you advance the hash. You are free to use any library to compute SHA-2 hashes. When finding your cycle you will need a starting value to hash, if you are in a group of two student that starting value should be the last names of both students concatenated together. If you are by yourself, your initial value should be your last name concatenated with the month of your birth.

   You should submit:
   - Your source code.
   - Your starting seed, the largest \( n \) you succeeded with, the two values which collide, the resultant hash of those colliding values.

2. **Does the Hare Catch the Tortoise?**
   Prove that if a cycle exists in a graph, the two pointers in the tortoise and hare algorithm are guaranteed to be equal at some point in the algorithm. In other words prove that if a cycle exists the algorithm terminates.

3. **MACs and Length**
   Show that appending the message length to the end of the message before applying basic CBC-MAC does not result in a secure MAC for arbitrary length messages.

*COSC 583* 4. **Double the Hashes, Double the Fun**
Let \((Gen_1, H_1)\) and \((Gen_2, H_2)\) be two hash functions. Define \((Gen, H)\) so that \(Gen\) runs \(Gen_1\) and \(Gen_2\) to obtain keys \(s_1\) and \(s_2\). Then define \(H^{s_1,s_2}(x) = H_1^{s_1}(x)||H_2^{s_2}(x)\).

- Prove that if at least one of \((Gen_1, H_1)\) and \((Gen_2, H_2)\) is collision resistant, then \((Gen, H)\) is collision resistant.

- Determine whether an analogous claim holds for second preimage resistance and preimage resistance respectively, prove your answer in each case.