**Homework 1**

1. Find a collision in each of the hash functions below:
   1. H(x) = x mod 712, where x can be any integer

**x = 3 and x = 712 +3**

* 1. H(x) = # of 1-bits in x, where x can be any bit string

**x = 1516 and x =7181**

* 1. H(x) = the three least significant bits of x, where x can be any bit string

**x = bear and x = pear**

1. Prove the statement: In a class of 500 students, there must be two students with the same birthday.

Proof:

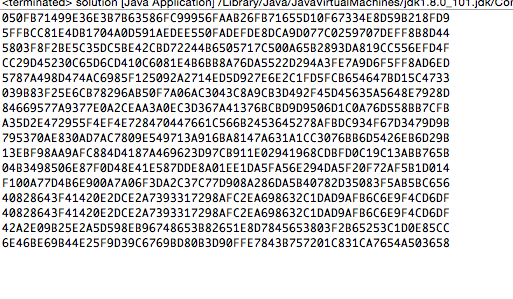
1. Every student has only one birthday.
2. There are no more than 366 days per year.
3. There are 500 students in this class.
4. Using the pigeon-hole principle, there must be two students with the same birthday.
   1. Pigeonholes: Birthday [Jan.1- Dec.31, no more than 366]
   2. Pigeons: Students [500]
   3. Collision: There must be two students “mapped” to a specific birthday.
5. Find an x such that H (x ◦ id) ∈ Y where
6. H = SHA-256
7. Id=0xED00AF5F774E4135E7746419FEB65DE8AE17D6950C95CEC3891070FBB5B03C77
8. Y is the set of all 256 bit values that have some byte with the value 0x1D.

**Solution:**

x=

050FB71499E36E3B7B63586FC99956FAAB26FB71655D10F67334E8D59B218FD9





1. To address Bob’s concern, the main idea of mechanism is to use hash function as a commitment scheme.
   1. Details of mechanism:

I would like to implement the digital analog of the following physical scheme: Alice and Bob agree on a secure hash function h. Alice chooses a random string rA and Bob chooses a random string rB. Bob tells Alice rB.

Now, Alice gives a number x between 1 to 10. Alice sends h (x, rA, rB) to Bob and asks Bob to guess that number. Let's say Bob guess a number y. Then Alice tells Bob (x, rA) and they can verify that x=y by checking that h (x, rA, rB) = h (y, rA, rB). In this way if Bob gives wrong number, then Alice can prove that he was wrong. Obviously, if Bob gives the number correctly, then the two hashes match.

* 1. Why does it work?

It's extremely hard for Alice to cheat because if Bob says "8" for example when the number was indeed "8" but Alice wants to trick him into thinking it was another number likes"7", she'd have to come up with a random string r such that h (7, r, rB) = h(8, rA, rB), which is hard by the assumption that h is a secure hash function and the fact that Bob chose rB. Essentially, the purpose of rA and h are to make Alice "commit" to her initial number x. The point of rA is so that, without it, Alice might pick some rA for which she knows another string r which might let her lie.