1. Given the key values for some students {2341, 1234, 1189, 2839, 430, 422, 454, 597, 2920}, students key ranges between 400 to 3000. Show the resulting hash tables after inserting the student data in the given order with each of these collision strategies.
   1. Using 1:1 direct mapping, no hashing
   2. Using open hashing with a linked list emerging from each bucket; bucket size = 7
   3. Using open hashing with a linked list emerging from each bucket; bucket size = 11
   4. Using closed hashing with a hash table of size 40 and second hashing function hi(x) = (x +i ) % 40
   5. Using closed hashing with a hash table of size 40 and second hashing function hi(x) = (x+ 3\*i) % 40
2. Why choosing a uniform function during hashing is important, and Which hashing function is more uniform in the range 0:9 assume you have key values between 200 and 300:
   1. H= x%10
   2. H= (x\*x)%10
   3. H= int(sqrt(x)) %10
   4. H = x%3
   5. H = random\_number(0,9)
3. What are the Hash functions good metrics?
4. Design two level of hashing for storing records about the cars in Egypt. In the first level we use the first three letters in the car sign in the second level we use the remaining three letters. Discuss the required data structures and hence the resulting space and big O time

//1st level distuributed by N/D1 second level N/D1\*D2,so complexity O(N/D1\*D2) if last level stored in linked list if BST O(logN/D1\*D2)

1. How good are the following hash functions:
   1. 10 digit phone number → ph → H(ph) = first\_three\_digits(ph)//bad 012-011-010 all start with that
   2. 10 digit phone number → ph → H(ph) = last\_three\_digits(ph)//good
   3. Memory locations → address → H(address) = address%1000//good uniform
2. Design a hashing system (including hashing function, primary and secondary data structure if needed) to take student first names and create an index to quickly access students data.
3. A hash table with a function h(i) = i - offset, where offset is a constant has a disadvantage of \_\_\_\_\_\_.
4. When to use Hashing or Binary Search Tree to implement the key-value mapping?
5. Design a data structure to store your contacts to support the following:
   1. Lookup the phone number by name.
   2. Determine who is calling given their phone number.
6. Given an unsorted array A of n integers. What are the possible solutions for getting pairs that sum to n and what are their complexity?

11.What can be done to compute the hash key value of a string?

* 1. Convert them all to their ASCII values
  2. Generate random numbers for the letters every time
  3. Give them each a value according to their place in the alphabet
  4. a and b only
  5. a and c only
  6. b and c only

a and c only.  
The idea is that you want the hashing function to generate the same output for the same string.  
B will just generate a new value each time you use the hashing function (i.e: You won’t be able to search for your data since you don’t know where it’s stored!)