[1] Demonstrate the insertion of the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be h(k) = k mod 9.

[2] Show that if $|U| > nm$, there is a subset of U of size n consisting of keys that all hash to the same slot, so that the worst-case searching time for hashing with chaining is O(n).

[3] Suppose a string of r characters is hashed into m slots by treating it as a radix-128 number and then using the division method. The number m is easily represented as a 32-bit computer word, but the string of r characters, treated as a radix-128 number, takes many words. How can we apply the division method to compute the hash value of the character string without using more than a constant number of words of storage outside the string itself?

[4] Draw the 11 entry hashtable for hashing the keys 12, 44, 13, 88, 23, 94, 11, 39, 20 using the function (2i+5) mod 11, closed hashing, assuming collisions are handled by linear probing.

[5] Design a **C++ class** that implements the **Hash table** as data structure?

[6] Show the result of inserting the keys 10111101, 00000010, 10011011, 10111110, 01111111, 01010001, 10010110, 00001011, 11001111, 10011110, 11011011, 00101011, 01100001, 11110000, 01101111 into an initially empty extendible hashing data structure with M = 4.

[7] Using buckets of size 3 and a hash function of mod(key, 5) and bucket chaining enter the following records (only the key values are shown) into an empty traditional hash file. Create chains of buckets when needed.

42, 57, 16, 52, 66, 77, 12, 25, 21, 33, 32, 14

[8] Define and differentiate between Hash function and cryptographic hash function? What are the applications of randomized hashing?