Tasks:

01 . Write a Program to implement Hash table and implement the following task. Via linked lists and any Hash calculation method.

Keys=(20,34,45,70,56)

a. Insert element into the table

b. Search element from the key

c. Delete element at a key

02 . Given an array of N integers, and an integer K, find the number of pairs of elements in the array whose sum is equal to K. Use Hashing (time complexity should not be more than N worst case)

Input:

N = 4, K = 6

arr[] = {1, 5, 7, 1}

Output: 2

Explanation:

arr[0] + arr[1] = 1 + 5 = 6

and arr[1] + arr[3] = 5 + 1 = 6.

03 Write a Program to implement Hash table and implement the following task. Consider the values **3, 2, 9, 6, 11, 13, 7, 12, 23, 22, 26 where m = 10. Use Separate Chaining to Avoid Collision then Sort the each and every chain of values stored at particular index.**

04. Given an array arr[] of n integers. Check whether it contains a triplet that sums up to zero and time complexity should not exceed(n^2). Use hashing with any method

Note: Return 1, if there is at least one triplet following the condition else return 0.

Input: n = 5, arr[] = {0, -1, 2, -3, 1}

Output: 1

Explanation: 0, -1 and 1 forms a triplet

with sum equal to 0.

5. Given a set of N nuts of different sizes and N bolts of different sizes. There is a one-one mapping between nuts and bolts. Match nuts and bolts efficiently.

Comparison of a nut to another nut or a bolt to another bolt is not allowed. It means nut can only be compared with bolt and bolt can only be compared with nut to see which one is bigger/smaller.

The elements should follow the following order ! # $ % & * @ ^ ~ .

Example 1:

Input:

N = 5

nuts[] = {@, %, $, #, ^} //You can use any symbols they are not exclusive to this

bolts[] = {%, @, #, $ ^}

Output:

 # $ % @ ^

 # $ % @ ^

Note: Might sound easy but the max allowed time complexity is N*log(N) with hashes.