**Lab 10**

**Hashing**

**Introduction:**

Hash tables are good for doing a quick search on things. If we have an array full of data (say 100 items), if we knew the position where a specific item is stored in the array then we could quickly access it. For instance, we just happen to know that the item that we want is at position 3.

Then we can apply **myitem=myarray[3];**

* **Linear Probing:** When a new identifier is hashed into a full bucket, we need to find another bucket for this identifier. The simplest solution is to find the closest unfilled bucket. This is called linear probing or linear open addressing.



**2. Chaining:** In chaining you have an array of linked lists as shown in Figure 10.3. All the data in the "same link", have Colliding Hash values.



**Objective:**

The objective of this lab is to implement data insertion and search from Hash table using linear probing and chaining as overflow handling techniques, then compare the two techniques on the basis of time taken to execute search.

**APPLICATION:**

* A **hash** table is a data structure which is used to store key-value pairs.
* **Hash function** is used by **hash** table to compute an index into an array in which an element will be inserted or searched.
* Message Digest.
* Password Verification.
* Data Structures(Programming Languages)
* Compiler Operation.
* Rabin-Karp Algorithm.
* Linking File name and path together.

**ISSUE:**

Faced issue in measure time I do not how it works.

**CONCLUSION:**

Hashing is the most widely used data structure as it takes constant time O (1) for insert, delete, and search operations. Hashing is mostly implemented by using a hash function that computes a unique smaller key value for large data entries. We can implement hashing using arrays and linked lists.

Whenever one or more data entries equate to the same values of keys, it results in a collision. We have seen various collision resolution techniques including linear probing, chaining, etc. We have also seen the implementation of hashing in C++.