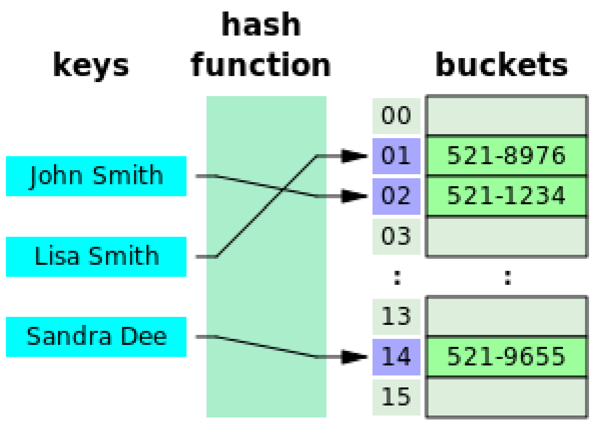
SKILL DEVELOPMENT LAB-II(2018-19)

ASSIGNMENT 7

- **Aim**: Insert the keys into a hash table of length m using open addressing using double hashing with h(k) = 1+(k mod(m-1)).
- **Objective**: To insert the data in hash table using insertion operation.
- **Theory:** Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects. Assume that you have an object and you want to assign a key to it to make searching easy. To store the key/value pair, you can use a simple array like a data structure where keys (integers) can be used directly as an index to store values. However, in cases where the keys are large and cannot be used directly as an index, you should use *hashing*. n hashing, large keys are converted into small keys by using **hash functions**. The values are then stored in a data structure called **hash table**.

  A hash table is a data structure that is used to store keys/value pairs. Hash Table uses an array as a storage medium and uses hash technique to generate an index where an element is to be inserted or is to be located from. It uses a hash function to compute an index into an array in which an element will be inserted or searched. By using a good hash function, hashing can work well. Under reasonable assumptions, the average time required to search for an element in a hash table is **O(1)**.



PHONE BOOK EXAMPLE

- **Algorithm:**

1. Calculate a hash code from the key
2. Access that hash element
   a. If the hash element is empty, add straight away
   b. If not, probe through subsequent elements (looping back if necessary), trying to find a free place

i. If a free place is found, add the data at that position
ii. If no free place is found, the add will fail.

- **Program code:**

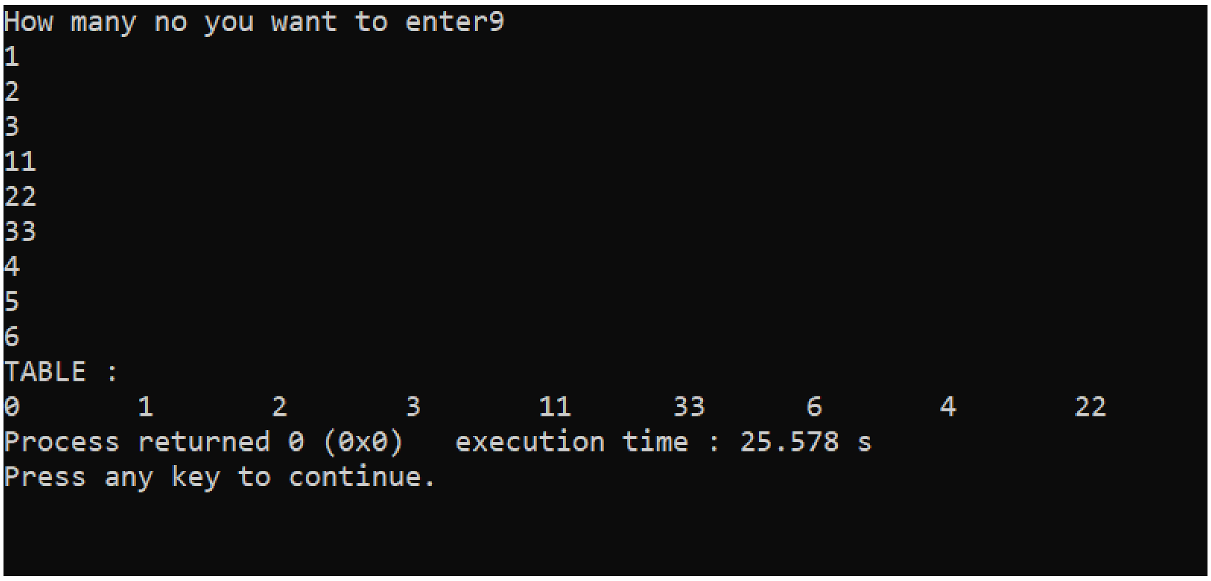
```
#include<iostream>
using namespace std;
class hashing
{
   int a[10],tab[10],i,n,h1,h,h2,j;
public:
   hashing()
   {
      for(i=0;i<9;i++)
      {
       tab[i]=0;
      }
   }
   int accept()
   {
      cout<<"How many no you want to enter";
      cin>>n;
      for(i=0;i<n;i++)
      {
         cin>>a[i];
      }
      op(a);
   }
   int op(int a[])
   {
       for(i=0;i<n;i++)
       {
          h1=a[i]%10;
          if(tab[h1]==0)
          {
             tab[h1]=a[i];
          }
          else
          {
             h2=7-(a[i]%7);
```

```
                for(j=0;j<9;j++)
                {
                    h=(h1+(j+1)*h2)%10;
                    if(tab[h]==0)
                      {
                      tab[h]=a[i];
                       break;
                      }
                }
            }

        }
        cout<<"TABLE :"<<endl;
       for(i=0;i<9;i++)
       {
        //if(tab[i]!=0)
        cout<<tab[i]<<"\t";
       }

    }
};
int main()
{
    hashing hh;
    hh.accept();
}
```

- **Output:**



```
How many no you want to enter9
1
2
3
11
22
33
4
5
6
TABLE :
0       1       2       3       11      33      6       4       22
Process returned 0 (0x0)   execution time : 25.578 s
Press any key to continue.
```

- **Conclusion:** We understand the need and concept of hashing and hash table.