* Assignment No. 7 *.

• Title: program for constructing telephone book database of a client. make use of a hash table implementation to quickly lookup client telephone.no.

* Objective :
  1) Basic concept.
  2) Concept of recursion of iteration.

* problem statement.
  consider telephone book database n clients. make use of hash table implementation to quickly look up telephone number.

* Outcomes:
  Input :
    1) Enter name.
    2) Enter phone.number.

  Output:
    Insert element int table Search element from key delete element of a key.

• Hardware Requirement :-
  8GB RAM, 500GB and intel Processor.

• Software Requirement :
  gedit s/w terminal, gcc/g++ compiler.

* Theory :-

Hash table are an efficient implemented of a key array data structure. a struct sometime known as an associative array.

* Hashing function :

We are designing a container which will be ~~wired~~ used to holds some number of items of given set k. In this context we all the element of the set k key the general key in the array is given by function is called a hash function.

∘ method of hash function :

1) Divison method
2) midsquare method.
3) floding method.
4) Digit analysis.
5) length dependent method,.
6) Algebric coding.
7) multicative hashing.

* Collosion Resolution :

collosion resolution in the main program in hashing. If the element to be inserted is mapping to the same locate when element is insered then we have a collosion is it must be resolved. There are seveal stategies for collusion resolution. The most commonly used are

collousion resolution statergies.

1) seperaate channing
2) open addressing
    a) Linear problem
    b) Quadradic probling.
    c) double hashing.

° Consider an Example.:

Sr. No.	Name	mobileno.
1.	Shreya	7298182696.
2.	Deepika Nimbalkar	9826688210.
3.	Harshali kale	9928342612
4.	Neha Landge	9712118018.
5.	Teju Salunkke	7421349868
6.	Sneha kale	984442816.

In above table it is a telephone breaks list. of peoples. Now by using linear probing we can create a hash tables by using abosoulue of list name letter of name lie.

H(83)%10 = 3
H(72)%10 = 2
H(68)%10 = 8
H(78)%10 = 8 collosion.
H(84)%10 = 4
H(83)%10 = 3 collosion.

	Name	mod. no.
0		
1	Harshali kale	
2.	Harsh.	
3.	Shreya padue.	9928342618
4.	Teju salunke.	7299182618 7
5.	Sneha kale.	8087125421
6		
7		
8	Deepika Nimbalkar	8018762182
9	Neha landge.	9712118012

◦ Basic Operation:

following are the primary operation of a hash table.

1) search.: search on element in a hash table.
2) Insert an element in a hash table.
3) delete an element from a hash table.

◦ Search Operation..:

Whenever an element is to the compare the hash table code of the key passed, and locate the element using hash table code is index in the array. use it. the element is not found at the computed hash code

Example.:

```
        struct node
        {
            int data;
            int key;
        };
```

```
struct node * search (int key).
  {
        int hashindex = hash code (key).
        while (hash Array [hash Index] != key)
          {
              if (hash Array[hash Index] -> key == key)
               return hash Array [hash Index];
               hash index = hash index % = size;
          }
         return NULL;
      }.
```

• Insert operation :)
Whenever an element is to be insert compute the hash table code of key passed code as an index in the array. Use linear probing for empty location. If an element is found at the computed hash code.

Ex.:

```
     Void insert (int key, int data).
     {
            Struct node * item (struct node k) malloc
       (size of (struct node));
         item -> data = data;
          item -> key = key;
         int hash Index = hash code (key);
          hash Array [hash Index] -> key != -1);
          {
               ++hash Index ;
               hash Index % = size;
          }
```

```
    hashArray[hashArray]=item;
}.
```

• Delete Operation :

Whenever an element is to be deleted compute the hash code of the key deleted compute the hash code and locate using that use linear probing to get the element has if an element is not found at the computed item there to keep performace of the hash table.

• Conclusion :

Hence, we have studied and implementation the telephone book distonry using hashing.

Scanned with CamScanner