Jai Jain
1BM18CS040

Date ___/___/___

1) Dictionary operation using Hashing.

```
struct list
{   int data;
    struct list* next
} node ;
node * ptr[max], *root[max], *temp[max]

class Dict
{ public
    int index ;
    Dict()
    void search (int)
    void insert (int)
    void delete (int)
}

Dict Dict()
{ index = -1
  for (int i=0 ; i<max ; i++)
  {   root[i] = NULL
      ptr[i] = NULL
      temp[i] = NULL
  }
}
```

Saathi

Date ___/___/___

```
Did search (int key)
{   int flag = 0
    index = int (key, max)
    temp [index] = root [index]
    while ( temp [index] == NULL)
{   if (temp [index] → data == key)
    { //found  flag = 1 ; break ; }
      else
    { temp [index] = temp [index] → next
  if (flag == 0)  cout << Not found

  void Dictionary :insert (int key)
{ index = int (key, max)
  ptr [index] = (node *) malloc (sizeof (node))
   ptr [index] → data = key
  if (root [index] == NULL)
    { root [index] = ptr [index]
      root [index] → next = NULL
      temp [index] = ptr [index] ;
    }

    else
    { temp [index] = root [index]
      while (temp [index] → next != NULL)
      { temp [index] = temp [index] → next
        temp [index] → next = ptr [next]
        }
}
```

```
void Dictionary :: delete (int key)
{
    index = int (key , max)
    temp [index] = root [index]
    while (temp [index] → data ! = key && temp [index] !
    {
        ptr [index] = temp [index]
            temp [index] = temp [index] → next
    }
    ptr [index] → next = temp [index] → next
      temp [index] → data = -1
      temp [index] = NULL
      free (temp [index])
    }
}
```