CHITRANJEEVI
1BM19CS403

8. Implement function of Dictionary using hashing

```
Void Dictionary :: search(int key)
{
  int flag=0;
  index= int(key %max);
  temp[index].root(index];
  while(temp[index] != NULL)
    {
      if (temp[index -> data== key)
      {
      cout<< "\n search success";
       flag= 1;
       break;
     }

  else
  temp[index]= temp [index]->next;

}
if (flog==0)
    cout<< "\n Search Unsuccessful";

}
Dictoronary :: Dictionary(){
        index=-1;
      for (int i=0; i<max ; i++){
            arr[i]= NULL;
            Ptr[i]= NULL;
            temp[i]= NULL;
          }

}
```


chf.

IBM19CS403
CHIRANJEEVI

```
void Dictionary::insert(int key)
{
  index = int(key % max);
ptr[index]= (node_type)* malloc(sizeof(node_type));
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[index];
   }
}
void Dictionary :: delete_ele (int key){
        index = int(key % max);
        temp[index]= root[index];
    while (temp[index] -> data != key && temp[index] != NULL)
    {
     ptr[index]= temp[index]
     temp[index] = temp[index] -> next
    }
    ptr[index] -> next = temp[index] -> next;
cout << "\n" << temp[index] -> data << " has deleted";
      temp[index] -> data = -1;
       temp[index] = NULL;
        free (temp[index]);
}
```


chf.