Hashing

Khaithik.S.Anand
IBM18CS046
ADS LAB

Insertion:

```
void 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];
    }
}
```

Hashing

Rishik. S. Anand
1BMXCS046
ADS - LAB

Searching

```
void search(int key)
{
    int flag=0
    index=int(key%max);
    temp[index]=root[index];
    while(temp[index]!=NULL)
    {
        if(temp[index]->data==key
        {
            cout<<"\nSearch key is found";
            flag=1;
            break;
        }
        else temp[index]=temp[index]->next;
    }
    if(flag==0)
        cout<<"\n search key not found";
}
```

Hashing

Khrithik.S.Anand
1BM18CS046
ADS-LAB

Delete:

```
void delete (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 been deleted ";
    temp [index ] -> data = -1;
    temp[index ] = NULL;
    free (temp[index ]);
}
```