PRACTICAL NO.01

PROGRAM 1:-

```
#linear probing
from IPython.core.interactiveshell import sphinxify
import array as hashtable
size=7;
hashtable=hashtable.array('i',[-1]*size)
def initialize():
    display();

def display():
    for i in range(0,size):
        print("index [",i,"]= ",hashtable[i])

#fun with argument
def insert(num):
  key=num%size
  if(hashtable[key]==-1):
    hashtable[key]=num;
    #print(hashtable[key])
  else:
    print("colllision is occured at index",key)
    cnt=0;
    for i in range(0,size):
      #print(hashtable[i])
      if(hashtable[i]!=-1):
        cnt+=1
    if(cnt==size):
      print("Hash Table is Full")
    #display()
    #Liniear probing
    else:
      for j in range(0,size):
        key1=(i+1)%size;
        if(hashtable[key1]==-1):
            hashtable[key1]=num;
            break;

def search(searchno):
  for i in range(0,size):
```

```
    if(hashtable[i]==searchno):
      print("Number is found")
      break
  else:
    print("not Found")
def Delete(delno):
  key=num%size
  for i in range(0,size):
     if(hashtable[i]==delno):
        hashtable[i]=-1
        print("number is deleted")

initialize();
while True:
  print("1)Insert\n2)Display\n3)Search\n4)Delete\n5)Exit\n")
  ch=int(input("Enter your choice:"))
  if ch==1:
    op=1
    while(op):
      num=int(input('Enter the number'))
      insert(num)
      op=int(input('Enter the number o for exit or Enter 1 for more accept...'))
  elif ch==2:
    print("****HashTable***")
    display()
  elif ch==3:
    searchno=int(input("Enter the number for search:"))
    search(searchno)
  elif ch==4:
    searchnodel=int(input("Enter the number for Delete Hashing:"))
    Delete(searchnodel)
  else:
    break
```

OUTPUT:-

```
index [ 0 ]=  -1
index [ 1 ]=  -1
index [ 2 ]=  -1
index [ 3 ]=  -1
index [ 4 ]=  -1
index [ 5 ]=  -1
index [ 6 ]=  -1
index [ 7 ]=  -1
index [ 8 ]=  -1
```

```
index [ 9 ]=  -1
1)Insert
2)Display
3)Search
4)Delete
5)Exit

Enter your choice:1
Enter the number10
Enter the number o for exit or Enter 1 for more accept...1
Enter the number11
Enter the number o for exit or Enter 1 for more accept...1
Enter the number12
Enter the number o for exit or Enter 1 for more accept...1
Enter the number13
Enter the number o for exit or Enter 1 for more accept...1
Enter the number14
Enter the number o for exit or Enter 1 for more accept...1
Enter the number15
Enter the number o for exit or Enter 1 for more accept...1
Enter the number16
Enter the number o for exit or Enter 1 for more accept...1
Enter the number17
Enter the number o for exit or Enter 1 for more accept...1
Enter the number18
Enter the number o for exit or Enter 1 for more accept...1
Enter the number19
Enter the number o for exit or Enter 1 for more accept...1
Enter the number20
colllision is occured at index 0
Hash Table is Full
Enter the number o for exit or Enter 1 for more accept...0
1)Insert
2)Display
3)Search
4)Delete
5)Exit
Enter your choice:2
****HashTable***
index [ 0 ]=  10
index [ 1 ]=  11
index [ 2 ]=  12
index [ 3 ]=  13
index [ 4 ]=  14
index [ 5 ]=  15
```

```
index [ 6 ]=  16
index [ 7 ]=  17
index [ 8 ]=  18
index [ 9 ]=  19
1)Insert
2)Display
3)Search
4)Delete
5)Exit

Enter your choice:3
Enter the number for search:13
Number is found
1)Insert
2)Display
3)Search
4)Delete
5)Exit

Enter your choice:4
Enter the number for Delete Hashing:10
number is deleted
1)Insert
2)Display
3)Search
4)Delete
5)Exit
Enter your choice:2
****HashTable***
index [ 0 ]=  -1
index [ 1 ]=  11
index [ 2 ]=  12
index [ 3 ]=  13
index [ 4 ]=  14
index [ 5 ]=  15
index [ 6 ]=  16
index [ 7 ]=  17
index [ 8 ]=  18
index [ 9 ]=  19
1)Insert
2)Display
3)Search
4)Delete
5)Exit
```

```
Enter your choice:5
```

PROGRAM 2:-

#Qudratic Probing

```
from IPython.core.interactiveshell import sphinxify
import array as hashtable
size=10;
hashtable=hashtable.array('i',[-1]*size)
def initialize():
    display();

def display():
    for i in range(0,size):
        print("index [",i,"]= ",hashtable[i])

#fun with argument
def insert(num):
  key=num%size
  if(hashtable[key]==-1):
    hashtable[key]=num;
    #print(hashtable[key])
  else:
    print("colllision is occured at index",key)
    cnt=0;
    for i in range(0,size):
      #print(hashtable[i])
      if(hashtable[i]!=-1):
        cnt+=1
    if(cnt==size):
      print("Hash Table is Full")
    #display()
    #Qudratic probing
    else:
      for j in range(0,size):
        key1=(key+(j*j))%size;
        if(hashtable[key1]==-1):
            hashtable[key1]=num;
            break;

def search(searchno):
  for i in range(0,size):
    if(hashtable[i]==searchno):
      print("Number is found")
      break
```

```
  else:
    print("not Found")


def Delete(delno):
  key=num%size
  for i in range(0,size):
     if(hashtable[i]==delno):
         hashtable[i]=-1
         print("number is deleted")

initialize();
while True:
  print("1)Insert\n2)Display\n3)Search\n4)Delete\n5)Exit\n")
  ch=int(input("Enter your choice:"))
  if ch==1:
    op=1
    while(op):
      num=int(input('Enter the number'))
      insert(num)
      op=int(input('Enter the number o for exit or Enter 1 for more accept...'))
  elif ch==2:
    print("****HashTable***")
    display()
  elif ch==3:
    searchno=int(input("Enter the number for search:"))
    search(searchno)
  elif ch==4:
    searchnodel=int(input("Enter the number for Delete Hashing:"))
    Delete(searchnodel)
  else:
    break
```

OUTPUT:-

```
Enter the number o for exit or Enter 1 for more accept...1
Enter the number91
Enter the number o for exit or Enter 1 for more accept...1
Enter the number33
Enter the number o for exit or Enter 1 for more accept...1
Enter the number18
Enter the number o for exit or Enter 1 for more accept...1
Enter the number27
Enter the number o for exit or Enter 1 for more accept...1
Enter the number36
```

```
colllision is occured at index 6
Enter the number o for exit or Enter 1 for more accept...1
Enter the number62
colllision is occured at index 2
Enter the number o for exit or Enter 1 for more accept...0
1)Insert
2)Display
3)Search
4)Delete
5)Exit
Enter your choice:2
****HashTable***
index [ 0 ]=  36
index [ 1 ]=  91
index [ 2 ]=  42
index [ 3 ]=  33
index [ 4 ]=  -1
index [ 5 ]=  -1
index [ 6 ]=  16
index [ 7 ]=  27
index [ 8 ]=  18
index [ 9 ]=  -1
1)Insert
2)Display
3)Search
4)Delete
5)Exit
Enter your choice:3
Enter the number for search:36
Number is found
1)Insert
2)Display
3)Search
4)Delete
5)Exit
Enter your choice:4
Enter the number for Delete Hashing:36
number is deleted
1)Insert
2)Display
3)Search
4)Delete
5)Exit
Enter your choice:5
```