Week 5

Hashing

Avni Arora_20103153_B6_week#5

1)

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
   int a[10],count=0;
   for(int i=0;i<10;i++)
   {
      a[i]=-1;
   }
   cout<<"enter the key value to be inserted: (enter -1 to stop): ";
   while(1)
   {

      int b;
      cin>>b;
      if(b!=-1)
      {
         count++;
         int f=0;
         int c=b%9;
         int j=0;
         do
         {
```

```
            if(a[c]==-1)
            {
                a[c]=b;
                f=1;
            }
            else
            {
                c= (b+j)%9;
                j++;

            }
        }while(f==0);

    }
    else
        break;
    if(count==10)
        break;

}
cout<<"hash table: "<<endl;
for(int i=0;i<10;i++)
{
    if(a[i]!=-1)
    {
        cout<<i<<": "<<a[i]<<endl;;
    }
    else
    {
        cout<<i<<": "<<"empty"<<endl;
```

```
    }
  }

}
```

Output:

```
enter the key value to be inserted: (enter -1 to stop): 14 18 24 20 3 23 33 15 -1
hash table:
0: 18
1: 15
2: 20
3: 3
4: empty
5: 14
6: 24
7: 23
8: 33
9: empty
```

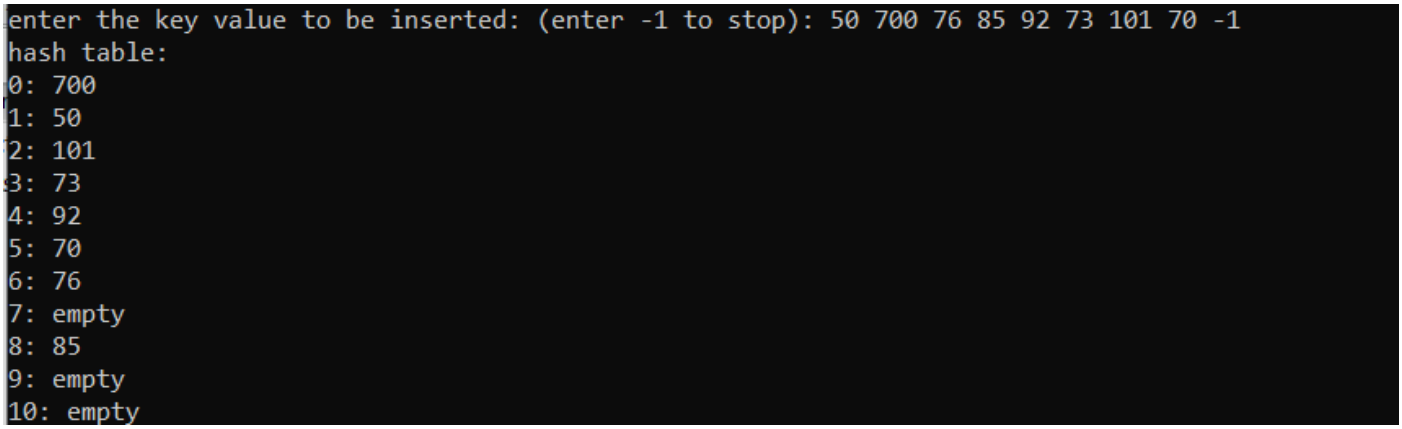
2)

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
  int a[11],count=0;
  for(int i=0;i<11;i++)
  {
    a[i]=-1;
  }
  cout<<"enter the key value to be inserted: (enter -1 to stop): ";
  while(1)
  {

    int b;
    cin>>b;
    if(b!=-1)
```

```
    {
        count++;
        int f=0;
        int c=b%7;
        int j=0;

        do
        {
            if(a[c]==-1)
            {
                a[c]=b;
                f=1;
            }
            else
            {
                c= (b+j)%11;
                j++;

            }
        }while(f==0);

    }
    else
        break;
    if(count==11)
        break;

}
cout<<"hash table: "<<endl;
for(int i=0;i<11;i++)
```

```
    {
        if(a[i]!=-1)
        {
            cout<<i<<": "<<a[i]<<endl;;
        }
        else
        {
            cout<<i<<": "<<"empty"<<endl;
        }
    }

}
```

Output:



3)

```
#include<bits/stdc++.h>
using namespace std;
struct node
{
    int data;
    node *next;
};
int main()
{
    cout<<"enter no of buckets: ";
```

```
int n;
cin>>n;
node * a= new node[n];
for(int i=0;i<n;i++)
{
   a[i].data=i;
   a[i].next=NULL;
}
cout<<"enter the key value to be inserted: (enter -1 to stop): ";
while(1)
{

   int b;
   cin>>b;
   if(b!=-1)
   {
      int c=b%n;
      node *p=new node;
      p->data=b;
      p->next=NULL;
      if(a[c].next==NULL)
      {
         a[c].next=p;
      }
      else
         {
            p->next=a[c].next;
            a[c].next=p;

         }
```

```
        }
        else
            break;

    }

    cout<<"hash table :"<<endl;
    for(int i=0;i<n;i++)
    {
        if(a[i].next!=NULL)
        {
            cout<<i<<" : ";
            node *p=a[i].next;
            while(p!=NULL)
            {
                cout<<p->data<<"->";
                p=p->next;
            }
            cout<<endl;
        }
    }
}
```

Output:

```
enter no of buckets: 5
enter the key value to be inserted: (enter -1 to stop): 11 12 13 15 22 34 44 31 64 -1
hash table :
0 : 15->
1 : 31->11->
2 : 22->12->
3 : 13->
4 : 64->44->34->

Process returned 0 (0x0)   execution time : 33.946 s
Press any key to continue.
```

4)

Q1 using quadratic probing)

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
   int a[10],count=0;
   for(int i=0;i<10;i++)
   {
      a[i]=-1;
   }
   cout<<"enter the key value to be inserted: (enter -1 to stop): ";
   while(1)
   {

      int b;
      cin>>b;
      if(b!=-1)
      {
         count++;
         int f=0;
         int c=b%9;
         int j=0;
         do
         {

            if(a[c]==-1)
            {
```

```
            a[c]=b;
            f=1;
        }
        else
        {
            c= (b+(j*j))%9;
            j++;

        }
    }while(f==0);

  }
  else
    break;
  if(count==10)
    break;

}
cout<<"hash table: "<<endl;
for(int i=0;i<10;i++)
{
  if(a[i]!=-1)
  {
    cout<<i<<": "<<a[i]<<endl;;
  }
  else
  {
    cout<<i<<": "<<"empty"<<endl;
  }
}
```

}

Output:

```
enter the key value to be inserted: (enter -1 to stop): 14 18 24 20 3 23 33 15 -1
hash table:
0: 18
1: 23
2: 20
3: 3
4: 15
5: 14
6: 24
7: 33
8: empty
9: empty
```

Q2 with quadratic probing)

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
   int a[11],count=0;
   for(int i=0;i<11;i++)
   {
      a[i]=-1;
   }
   cout<<"enter the key value to be inserted: (enter -1 to stop): ";
   while(1)
   {

      int b;
      cin>>b;
      if(b!=-1)
      {
```

```
        count++;
        int f=0;
        int c=b%7;
        int j=0;

        do
        {
           if(a[c]==-1)
           {
              a[c]=b;
              f=1;
           }
           else
           {
              c= (b+(j*j))%11;
              j++;

           }
        }while(f==0);

    }
    else
        break;
    if(count==11)
        break;

}
cout<<"hash table: "<<endl;
for(int i=0;i<11;i++)
{
```

```
        if(a[i]!=-1)
        {
            cout<<i<<": "<<a[i]<<endl;;
        }
        else
        {
            cout<<i<<": "<<"empty"<<endl;
        }
    }

}
```

Output:

```
enter the key value to be inserted: (enter -1 to stop): 50 700 76 85 92 73 101 70 -1
hash table:
0: 700
1: 50
2: 101
3: 73
4: 92
5: 70
6: 76
7: empty
8: 85
9: empty
10: empty
```

5)

Q1 with chain hashing:

```
#include<bits/stdc++.h>
using namespace std;
struct node
{
    int data;
    node *next;
};
int main()
{
```

```
int n=9;
node * a= new node[n];
for(int i=0;i<n;i++)
{
   a[i].data=i;
   a[i].next=NULL;
}
cout<<"enter the key value to be inserted: (enter -1 to stop): ";
while(1)
{

   int b;
   cin>>b;
   if(b!=-1)
   {
      int c=b%n;
      node *p=new node;
      p->data=b;
      p->next=NULL;
      if(a[c].next==NULL)
      {
         a[c].next=p;
      }
      else
         {
            p->next=a[c].next;
            a[c].next=p;

         }
```

```
        }
        else
            break;

    }

    cout<<"hash table :"<<endl;
    for(int i=0;i<n;i++)
    {
        if(a[i].next!=NULL)
        {
            cout<<i<<" : ";
            node *p=a[i].next;
            while(p!=NULL)
            {
                cout<<p->data<<"->";
                p=p->next;
            }
            cout<<endl;
        }
    }
}
```

Output:

```
enter the key value to be inserted: (enter -1 to stop): 14 18 24 20 3 23 33 15 -1
hash table :
0 : 18->
2 : 20->
3 : 3->
5 : 23->14->
6 : 15->33->24->
```

Q2 with chain hashing

```
#include<bits/stdc++.h>
using namespace std;
struct node
{
   int data;
   node *next;
};
int main()
{
   int n=7;
   node * a= new node[n];
   for(int i=0;i<n;i++)
   {
      a[i].data=i;
      a[i].next=NULL;
   }
   cout<<"enter the key value to be inserted: (enter -1 to stop): ";
   while(1)
   {

      int b;
      cin>>b;
      if(b!=-1)
      {
         int c=b%n;
         node *p=new node;
         p->data=b;
         p->next=NULL;
```

```
        if(a[c].next==NULL)
        {
            a[c].next=p;
        }
        else
            {
                p->next=a[c].next;
                a[c].next=p;

            }


    }
    else
        break;

}

cout<<"hash table :"<<endl;
for(int i=0;i<n;i++)
{
    if(a[i].next!=NULL)
    {
        cout<<i<<" : ";
        node *p=a[i].next;
        while(p!=NULL)
        {
            cout<<p->data<<"->";
            p=p->next;
        }
```

```
            cout<<endl;
        }
    }
}
```

Output:

```
enter the key value to be inserted: (enter -1 to stop): 50 700 76 85 92 73 101 70 -1
hash table :
0 : 70->700->
1 : 92->85->50->
3 : 101->73->
6 : 76->
```

6)

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int a[11],count=0;
    for(int i=0;i<11;i++)
    {
        a[i]=-1;
    }
    cout<<"enter the key value to be inserted: (enter -1 to stop): ";
    while(1)
    {

        int b;
        cin>>b;
        if(b!=-1)
        {
```

```
        count++;
        int f=0;
        int c=b%11;
        int j=0;
        do
        {

           if(a[c]==-1)
           {
              a[c]=b;
              f=1;
           }
           else
           {
              c= ((b%7)+j*(b%3))%11;
              j++;

           }
        }while(f==0);

    }
    else
        break;
    if(count==10)
        break;

}
cout<<"hash table: "<<endl;
for(int i=0;i<10;i++)
{
```

```
        if(a[i]!=-1)
        {
            cout<<i<<": "<<a[i]<<endl;;
        }
        else
        {
            cout<<i<<": "<<"empty"<<endl;
        }
    }

}
```

Output:

```
enter the key value to be inserted: (enter -1 to stop): 50 700 76 85 92 73 101 70 -1
hash table:
0: 70
1: empty
2: 101
3: 73
4: 92
5: empty
6: 50
7: 700
8: 85
9: empty
```