```
// Binary Search Using CPP

#include<iostream>
using namespace std;
 main()
{
        int a[5]={10,20,30,40,50};
        int lr=0,up=4,f=0,mid,item;
        cout<<"\n Enter No. For Search => ";
        cin>>item;
        while(lr<=up)
        {
                mid=(lr+up)/2;
                if(a[mid]==item)
                {
                        f=1;
                        break;
                }
                if(a[mid]<item)
                  lr=mid+1;
                else
                  up=mid-1;
        }
        if(f==1)
          cout<<"\n Item is Found ";
        else
          cout<<"\n Item Not Found";


}
```

```
// Linear Search using CPP

#include<iostream>
using namespace std;
main()
{
    int a[5]={10,20,30,40,50},item,i;
    cout<<"\n Enter element for Search = >";
    cin>>item;
    while(i<5)
    {
        if(a[i]==item)
        {
            cout<<"\n Item Found";
            break;
        }
        i++;
    }
    if(i>=5)
    cout<<"\n Item Not Found";
}
```

```
// Selection Sort using CPP

#include<iostream>
using namespace std;
int findSmallest (int[],int);
int main ()
{
   int myarray[10] = {11,5,2,20,42,53,23,34,101,22};
   int pos,temp;
   cout<<"\n Unsorted List of elements \n\n";
   for(int i=0;i<10;i++)
   {
      cout<<myarray[i]<<"\t";
   }
   for(int i=0;i<10;i++)
   {
      pos = findSmallest(myarray,i);
      temp = myarray[i];
      myarray[i]=myarray[pos];
      myarray[pos] = temp;
   }
   cout<<"\n\n Sorted list of elements is \n\n\n";
   for(int i=0;i<10;i++)
   {
      cout<<myarray[i]<<"\t";
   }
   return 0;
}
int findSmallest(int myarray[],int i)
```

```
{
    int ele_small,position,j;
    ele_small = myarray[i];
    position = i;
    for(j=i+1;j<10;j++)
    {
        if(myarray[j]<ele_small)
        {
            ele_small = myarray[j];
            position=j;
        }
    }
    return position;
}
```

```
//Bubble Sort using CPP

#include<iostream>
using namespace std;
class Bubblesort
{
int a[20],i,j,n,temp;
public:
void BSort();
void getdata();
void display();
};
void Bubblesort::getdata()
{
```

```
cout<<"\n Enter how many elements:";
cin>>n;
cout<<" Enter elements:-";
for(i=1;i<=n;i++)
{

cin>>a[i];
}
}
void Bubblesort::BSort()
{
for(i=1;i<=n;i++)
{
for(j=i+1;j<=n;j++)
{
if(a[i]>=a[j])
{
temp=a[i];
a[i]=a[j];
a[j]=temp;
}
}
}
}
void Bubblesort::display()
{
cout<<"\n Elements are Sorted using Bubble sort \n";
for(i=1;i<=n;i++)
{
```

```
cout<<a[i]<<"\t";
}
}
main()
{
Bubblesort b;
b.getdata();
b.BSort();
b.display();
}
```

```
//Quick Sort using CPP
#include<iostream>
using namespace std;
int QPartition(int a[],int start,int end)
{
int pivot=a[end];
int i=(start-1);
for(int j=start;j<=end-1;j++)
{
if(a[j]<pivot)
{
i++;
int t=a[i];
a[i]=a[j];
a[j]=t;
}
}
```

```
int t=a[i+1];
a[i+1]=a[end];
a[end]=t;
return(i+1);
}
void QuickSort(int a[],int start,int end)
{
if(start<end)
{
int p=QPartition(a,start,end);
QuickSort(a,start,p-1);
QuickSort(a,p+1,end);
}
}
void Printarr(int a[],int n)
{
int i;
for(i=0;i<n;i++)
cout<<a[i]<<" ";
}
 main()
{

int a[]={23,8,28,13,18,26};
int n=sizeof(a)/sizeof(a[0]);
cout<<"Before sorting array element are:\n\n";
Printarr(a,n);
QuickSort(a,0,n-1);
cout<<"\n After sorting array elements are:\n\n";
```

```
Printarr(a,n);

}

//Insertion Sort using CPP
 #include<iostream>
 using namespace std;

void Insertionsort(int arr[],int n)
{
int i,temp,j;
for(i=1;i<n;i++)
{
temp=arr[i];
j=i-1;
while(j>=0&&arr[j]>temp)
{
arr[j+1]=arr[j];
j=j-1;
}
arr[j+1]=temp;
}
}
void Printarray(int arr[],int n)
{
int i;
for(i=0;i<n;i++)
cout<<arr[i]<<" ";
cout<<endl;
```

```
}
main()
{
int arr[]={12,11,13,5,6};
int n=sizeof(arr)/sizeof(arr[0]);
cout<<"\nBefore sorting array element:-";
Printarray(arr,n);
Insertionsort(arr,n);
cout<<"\nAfter sorting array element:-";
Printarray(arr,n);

}

// Stack Operation Using CPP

#include<iostream>
using namespace std;
int S[4],size=4,Top=-1,i,x;
void Push();
void Pop();
void Peep();
void Change();
void Display();
void Menu();

void Push()
{
 if(Top==size-1)
 {
   cout<<"Stack is Overflow ...."<<endl;
 }
 else
 {
   cout<<"Enter Value =>"<<endl;
   cin>>x;
   Top++;
   S[Top]=x;
   cout<<x <<"   is Pushed into Stack ..."<<endl;
 }
}

void Pop()
{
 if(Top==-1)
 {
    cout<<" Stack is Underflow"<<endl;
 }
 else
 {
    x=S[Top];
    cout<<x<<" is Poped from Stack"<<endl;
    Top--;
 }
}

void Peep()
{
   int pos;
   if(Top==-1)
   {
     cout<<" Stack is Underflow.."<<endl;
   }
   else
   {
      cout<<"Enter Position => "<<endl;
      cin>>pos;
      i=Top-pos+1;
      cout<<S[i]<<" is present at "<<pos<<" position"<<endl;
   }
}

void Change()
{
   int pos;
   if(Top==-1)
   {
     cout<<"Stack is Underflow"<<endl;
   }
   else
   {
     cout<<"Enter Position"<<endl;
     cin>>pos;
     cout<<"Enter the value to Change => "<<endl;
     cin>>x;
     i=Top-pos+1;
     S[i]=x;
     cout<<x<<" is changed at "<<pos<<" position"<<endl;
```

```
   }

}

void Display()
{
  if(Top==-1)
      cout<<"Stack is Underflow"<<endl;
  else
  {
    for(i=Top;i>=0;--i)
    {
       cout<<"  "<<S[i]<<endl;
    }
  }
}
void Menu()
{
     cout<<"1. Push()"<<endl;
     cout<<"2. Pop()"<<endl;
     cout<<"3. Peep()"<<endl;
     cout<<"4. Change()"<<endl;
     cout<<"5. Display()"<<endl;
}
int main()
{
    int ch=0;
    Menu();
    while(ch!=6)
    {
       cout<<"Enter Choice =>"<<endl;
       cin>>ch;
       switch(ch)
       {
               case 1:Push(); break;
               case 2:Pop(); break;
               case 3:Peep(); break;
               case 4:Change();break;
               case 5:Display(); break;
               default : cout<<"Wrong Choice ..."<<endl;
       }
    }
    return 0;
}
```