# DS PRACTICAL 03

**AIM:** Create an array of size n and write a program to sort a given array by selection sort and bubble sort.

**(SELECTION SORT)**

**PROGRAM:**

```
# include <iostream>
using namespace std;
int main()
{
   // program for selection sort

   int ar[100],i,j,n,temp;
   cout<<"enter the size of array:";
   cin>>n;
   cout<<"\nenter elements in the array"<<endl;
   for(i=0;i<n;i++)
   {
      cout<<"enter element:";
      cin>>ar[i];
   }
   cout<<"\nyour array is :";
   for(i=0;i<n;i++)
   {
      cout<<ar[i]<<" ";
   }
   for(i=0;i<n;i++)
   {
      for(j=i+1;j<n;j++)
      {
         if(ar[i]>ar[j])
          {
            temp=ar[i];
```

```
            ar[i]=ar[j];
            ar[j]=temp;
          }
       }
    }
    cout<<"\narray after sorting is:";
    for(i=0;i<n;i++)
    {
       cout<<ar[i]<<" ";
    }
}
```

**OUTPUT**


**(BUBBLE SORT)**

**PROGRAM:**

```
# include <iostream>
using namespace std;
int main()
{
   // for bubble sort
   int ar[100],i,j,n,temp;
   cout<<"enter the size of array:";
   cin>>n;
   cout<<"enter elements in array"<<endl;
```

```
   for(i=0;i<n;i++)
   {
      cout<<"enter element:";
      cin>>ar[i];
   }
   cout<<"\nyour array is:";
   for(i=0;i<n;i++)
   {
      cout<<ar[i]<<" ";
   }

   //Logic of bubble sort
   for(i=0;i<n;i++)
   {
      for(j=0;j<n-i;j++)
      {
         if(ar[j]>ar[j+1])
         {
            temp=ar[j];
            ar[j]=ar[j+1];
            ar[j+1]=temp;

         }
      }
   }
   cout<<"\nafter sorting your array is:";
    for(i=0;i<n;i++)
   {
      cout<<ar[i]<<" ";
   }
}
```

## OUTPUT


## (BINARY SEARCH)

Write a program to search any integer in your array using binary search concept.

**PROGRAM:**

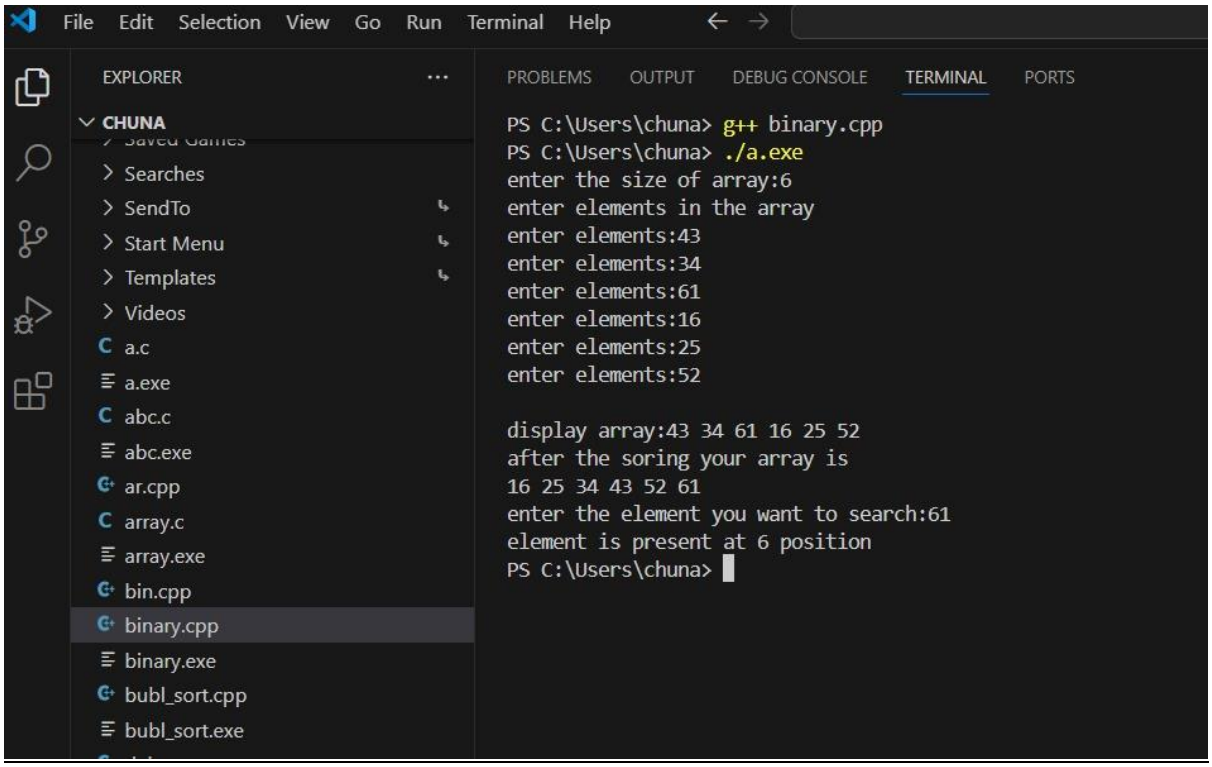
```
# include <iostream>
using namespace std;
int main()
{
   //for binary search
   int ar[100],beg,mid,end,n,num,i,j,found=0;
   cout<<"enter the size of array:";
   cin>>n;
   cout<<"enter elements in the array\n";
   for(i=0;i<n;i++)
   {
      cout<<"enter elements:";
      cin>>ar[i];

   }
   cout<<"\ndisplay array:";
   for(i=0;i<n;i++)
   {
      cout<<ar[i]<<" ";
```

```
}

// sorting array
for(i=0;i<n;i++)
{
   for(j=i+1;j<n;j++)
   {
      int temp;
      if(ar[i]>ar[j])
      {
         temp=ar[i];
         ar[i]=ar[j];
         ar[j]=temp;
      }
   }
}
cout<<"\nafter the soring your array is\n";
 for(i=0;i<n;i++)
{
   cout<<ar[i]<<" ";
}

cout<<"\nenter the element you want to search:";
cin>>num;
beg=0;
end=n;
mid=(beg+end)/2;
for(i=0;i<n;i++)
{
   if(num==ar[mid-1])
   {
```

```
        found=1;
      }
      else if(num<ar[mid-1])

        {

          end=mid-1;
        }
      else if(num>ar[mid-1])
        {
          beg=mid+1;
        }

      mid=(beg+end)/2;
   }
   mid=(beg+end)/2;
   if(found==1)
   {
      cout<<"element is present at "<<mid<<" position";
   }
   else
   {
      cout<<"element is not present in the array";
   }
}
```

# DS PRACTICAL 03

## OUTPUT




GITHUB LINK FOR PRACTICAL :

https://github.com/Nishikant-Chunarkar/DATA_STRUCTURE_PRACTICAL