

**A Practical Activity Report For  
Data Structures and Algorithms (UCS406)**

Submitted By: **Vivek Arora**

**101715178  
(ENC 8)**

Submitted To:

**Dr. Sanjay Sharma**



**ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT**

**THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY, (DEEMED TO BE UNIVERSITY),  
PATIALA, PUNJAB**

# ASSIGNMENT 4

## QUESTION 1( Various Functions of Array)

- i. Display()
- ii. Add/Append(x)
- iii. Insert(index,x)
- iv. Delete(x)
- v. LinearSearch(s)
- vi. BinarySearch(x)
- vii. Get(index)
- viii. Set(index,x)
- ix. Max()
- x. Min()
- xi. Reverse()
- xii. Shift()
- xiii. Rotate()

```
#include<iostream>
using namespace std;
void display(int a[],int n)
{
    cout<<" array : "<<endl;
    for(int i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
    }
    cout<<"\n";
}
void add(int a[],int n)
{
    int n1=n+1;
    int b[n1];
    for(int i=0;i<n;i++)
    {
        b[i]=a[i];
    }
    int value;
    cout<<endl;
    cout<<"Enter the element : "<<endl;
    cin>>value;
```

```

    b[n] = value;
    cout<<"After Adding:"<<endl;
    display(b,n1);
}
void del(int a[],int n)
{
    int b[n-1];int h,i=0,j=0,count=0,flag=0;
    cout<<"Enter the element"<<endl;
    cin>>h;
    for(i=0,j=0;i<n;j++)
    {
        if(a[i]==h)
        {
            flag=1;
            i++;
        }
        else
        {
            b[j]=a[i];
            count++;
            i++;
        }
    }
    if(flag!=1)
    {
        cout<<"\n element  not found\n";
    }
    else
    {
        display(b,count);
    }
}
void insert(int a[],int n,int b[])
{
    for(int i=0;i<n;i++)
    {
        b[i]=a[i];
    }
    int pos,value;
    cout<<"Enter the element and positon "<<endl;
    cin>>value>>pos;
    for (int i = n - 1; i >= pos - 1; i--)
        b[i+1] = b[i];

```

```

        b[pos-1] = value;
        cout<<"\nAfter Inserting:\n";
        display(b,n+1);
    }
void large(int a[],int n)
{
    int temp=a[0];
    for(int i = 1;i < n; ++i)
    {
        if(temp < a[i])
            temp = a[i];
    }
    cout << "\nMaximum element :\n " << temp;
}
void min(int a[],int n)
{
    int temp=a[0];
    for(int i = 1;i < n; ++i)
    {
        if(temp > a[i])
            temp = a[i];
    }
    cout << "\nMinimum element \n " <<temp;
}
void get(int a[],int n)
{
    int flag,i,k;
    cout<<"\nEnter the index of element to find:\n";
    cin>>k;
    for(i=0;i<n;i++)
    {
        if(i==k)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    {
        if(i==0)
            cout<<"\nThe element on given position:\n"<<a[i];
        else
            cout<<"\nThe element on given position:\n"<<a[i-1];
    }
}

```

```

void set(int a[],int n,int b[])
{
    for(int i=0;i<n;i++)
    {
        b[i]=a[i];
    }
    int position,value;
    cout<<"\nEnter the element which you wish to insert and index:\n";
    cin>>value>>position;
    for (int i = n - 1; i >= position - 1; i--)
        b[i+1] = b[i];

    b[position-1] = value;
    cout<<"\nAfter Inserting:\n";
    display(b,n+1);
}

void linearsearch(int a[],int n)
{
    int y;
    cout<<"\nEnter the number to be searched:\n";
    cin>>y;
    int flag,i;
    for(i=0;i<n;i++)
    {
        if(a[i]==y)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
    {
        cout<<"The number was found on position in linear
search:\n"<<i+1;
    }
    else
    { cout<<"The number was not found\n";}
}

int sort(int a[],int n)
{
    int temp;
    for(int i=0;i<n;i++)
    {

```

```

        for(int j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                temp =a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    return a[n];
}
int binarySearch(int arr[], int l, int r, int x)
{
    if (r >= l) {
        int mid = l + (r - l) / 2;

        // If the element is present at the middle
        // itself
        if (arr[mid] == x)
            return mid;

        // If element is smaller than mid, then
        // it can only be present in left subarray
        if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);

        // Else the element can only be present
        // in right subarray
        return binarySearch(arr, mid + 1, r, x);
    }

    // We reach here when element is not
    // present in array
    return -1;
}
void reverse(int a[],int n,int b[])
{
    int i,j;
    for(i=n-1,j=0; i>=0;i--,j++)
    {
        b[i]=a[j];
    }
}

```

```

        cout<<"\nAfter reversing:\n";
        display(b,n);
    }
    void shift(int a[],int n)
    {
        int i,temp;
        for(i=0;i<n;i++)
        {
            temp = a[n];
            a[n] = a[i];
            a[i] = temp;
        }
        cout<<"\nAfter shifting:\n";
        display(a,n);
    }
    void rotate(int a[],int n)
    {
        int temp = a[0],i;
        for ( i = 0; i < n ; i++)
        {
            a[i] = a[i + 1];
        }
        a[n-1] = temp;
        cout<<"\nAfter rotation:\n";
        display(a,n);
    }
    int main()
    {
        int n;
        cout<<"Enter the number of elements:\n";
        cin>>n;
        int a[n];
        cout<<"Enter the array:\n";
        for(int i=0;i<n;i++)
        {
            cin>>a[i];
        }
        display(a,n);
        add(a,n);
        int n1=n+1;
        int b1[n1];
        insert(a,n,b1);
        deletele(a,n);
    }

```

```

        linearsearch(a,n);
        a[n]=sort(a,n);
        int x;
        cout<<"enter element you want to search";
        cin>>x;
        int result = binarySearch(arr, 0, n - 1, x);
        (result == -1) ? cout<<"Element is not present in array": cout<<"Element is
present at index"<<result;
        get(a,n);
        set(a,n,b1);
        large(a,n);
        min(a,n);
        int b[n];
        reverse(a,n,b);
        shift(b,n);
        rotate(b,n);
        return 0;
}

```

## **QUESTION 2 (Various functions of array )**

**i. Check if an array is sorted**

**ii. Merge arrays**

**iii. Set operations on array: Union, Intersection**

```

#include<iostream>
using namespace std;
void display(int a[],int n)
{
    for(int i=0;i<n;i++)
    {
        cout<<a[i]<<" ";
    }
}
void checksort(int a[],int n)
{
    int flag;
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                flag=1;
                break;
            }
        }
    }
}

```



```

        }
    }
}
if(flag==1)
{
    cout<<"\nThe array is not sorted\n";
}
else
{
    cout<<"\nThe array is sorted\n";
}
}
int merge(int ab[],int a[],int b[],int n,int n1,int n2)
{
    for(int i=0;i<n1;i++)
    {
        ab[i]=a[i];
    }
    for(int j=0;j<n2;j++)
    {
        ab[n1+j]=b[j];
    }
    return ab[n];
}
int sort(int a[],int n)
{
    int temp;
    for(int i=0;i<n;i++)
    {
        for(int j=i+1;j<n;j++)
        {
            if(a[i]>a[j])
            {
                temp =a[i];
                a[i]=a[j];
                a[j]=temp;
            }
        }
    }
    return a[n];
}
int unions(int a1[],int a2[],int n,int m,int u[])
{

```

```

int i,j,k,count=0;
for(i=0,j=0,k=0;i<n&& j<m;)
{
    if(a1[i]<a2[j])
    {
        u[k]=a1[i];
        i++;
        k++;
        count++;
    }
    else if(a1[i]>a2[j])
    {
        u[k]=a2[j];
        j++;
        k++;
        count++;
    }
    else{
        u[k]=a1[i];
        i++;
        j++;
        k++;
        count++;
    }
}

if(i<n){
    for(;i<n;++i){
        u[k]=a1[i];
        k++;
        count++;
    }
}
else if(j<m)
{
    for(;j<m;++j){
        u[k]=a2[j];
        k++;
        count++;
    }
}
return u[count];
}

```

```

int inter(int a1[],int a2[],int m,int n)
{

    int i = 0, j = 0;
    while (i < m && j < n)
    {
        if (a1[i] < a2[j])
            i++;
        else if (a2[j] < a1[i])
            j++;
        else
        {
            cout << a2[j] << " ";
            i++;
            j++;
        }
    }
}

```

```

int main()
{
    int n1,n2,n;
    cout<<"Enter the Number of elements:\n";
    cin>>n1;
    int a[n1];
    cout<<"Enter the array:\n";
    for(int i=0;i<n1;i++)
    {
        cin>>a[i];
    }
    checksort(a,n1);
    cout<<"Enter the number of second elements:\n";
    cin>>n2;
    int b[n2];
    cout<<"Enter the second array:\n";
    for(int i=0;i<n2;i++)
    {
        cin>>b[i];
    }
    n=n1+n2;
    int ab[n];
    ab[n]=merge(ab,a,b,n,n1,n2);
    cout<<"The elements in combined array is:\n";
}

```

```

    display(ab,n);
    a[n1]=sort(a,n1);
    b[n1]=sort(b,n2);
    int u[n];
    u[n]=unions(a,b,n1,n2,u);
    cout<<"\nThe elements after union is: \n";
    display(u,n);
    cout<<"\nThe elements after intersection is: \n";
    inter(a,b,n1,n2);
    return 0;
}

```

### **QUESTION 3 (Various Function of Array )**

- i. Finding single element in an array**
- ii. Finding multiple elements in an array**
- iii. Finding duplicates in a sorted array**
- iv. Finding duplicates in an unsorted array**
- v. Finding a pair of elements with sum k**
- vi. Finding a pair of elements with sum k in sorted array**
- vii. Finding max and min in a single scan**

```

#include<iostream>
using namespace std;

void single(int temp[])
{int key=0,size=sizeof(temp)/sizeof(temp[0]);

cout<<"Enter the element to search"<<endl;
cin>>key;

for(int i=0;i<size;i++)
{
    if(temp[i]==key)
    {
        cout<<"Element is present at index "<<i<<endl;
        break;
    }
    else
        cout<<"Element is not present"<<endl;
}

}

```

```

void multiple(int temp[])
{
int key=0,size=sizeof(temp)/sizeof(temp[0]);
cout<<"Enter the element to search"<<endl;
cin>>key;
for(int i=0;i<size;i++)
{
if(temp[i]==key)
{
cout<<"The element is found at index "<<i<<endl;
}

}
}

```

```

void sortedduplicate(int temp[])
{
for(int i=0;i<size-1;i++)
{
if(temp[i]==temp[i+1])
{
cout<<"Element repeated is : "<<temp[i]<<endl;
}
}
}

```

```

void unsortedduplicate(int temp[])
{
int size=sizeof(temp)/sizeof(temp[0]);
for(int i=0;i<size-1;i++)
{
for(int j=i+1;j<size;j++)
{
if(temp[i]==temp[j])
{
cout<<"Element repeated is : "<<temp[i];
break;
}
}
}
}
void sum(int temp[])

```

```

{
    int k=5,t=0,size=sizeof(temp)/sizeof(temp[0]);
    for(int i=0;i<size;i++)
    {
        t=k-temp[i];
        for(int j=0;j<size;j++)
        {
            If(temp[j]==t)
            { cout<<"Elements at indexes "<<i<<" and "<<j<<" have the sum = "<<k;
              break;
            }
        }
    }
}

```

```

void maxmin(int temp[])
{
    int max=0,min=0,size=sizeof(temp)/sizeof(temp[0]);
    max=temp[0];
    min=temp[0];
    for(int i=1;i<size;i++)
    {
        if(temp[i]<min)
        {
            min=temp[i];
        }
        if (temp[i]>max)
        {
            max=temp[i];
            <#statements#>
        }
    }
    cout<<" Maximum:"<<max<<"Minimum: "<<min<<endl;
}

```

```

int main()
{
    int a[]={4,6,3,4,6,7};
    single(a);
    multiple(a);
    sortedduplicate(a);
    unsortedduplicate(a);
}

```

```
sum(a);  
maxmin(a);  
    return 0;  
}
```