# EXPERIMENT -1

# Aim: Write a menu driven program to show array implementation. It should consist of input, output, insert, delete, and searching.

## Pseudo code

Function binarysearch(arr, n, key):

s = 0

e = n - 1

while s <= e:

mid = (s + e) / 2

if arr[mid] == key:

return mid

else if arr[mid] > key:

e = mid - 1

else if arr[mid] < key:

s = mid + 1

return -1

Function main():

Read n

Declare arr[n]

Read num (menu option)

Switch num:

Case 1:

Read array elements into arr

Case 2:

Output array elements from arr

Case 3:

Read pos (position) and val (value)

Insert val at pos in arr

Case 4:

Read pos (position)

Delete element at pos in arr

Case 5:

Read key (element to search)

Linearly search for key in arr and output its position

Case 6:

Read key (element to search)

Perform binary search for key in arr and output its position

End Switch

End Function

## Source code:

#include<iostream>

using namespace std;

int binarysearch(int arr[],int n,int key){

    int s=0;

    int e=n-1;

    while (s<=e){

        int mid =(s+e)/2;

        if (arr[mid]==key){

            return mid;

        }

        else if (arr[mid]>key){

            e=mid-1;

        }

        else if (arr[mid]<key){

            s=mid+1;

        }

    }

    return -1;

}

int main(){

    int n;

    cout<<"Enter the size of the array:";

    cin>>n;

    int arr[n];

    int num;

        cout<<"Menu Driven Program"<<endl;

        cout<<"1.Input"<<endl;

        cout<<"2.Output"<<endl;

        cout<<"3.Insert"<<endl;

        cout<<"4.Delete"<<endl;

        cout<<"5.Linear Search"<<endl;

        cout<<"6.Binary Search"<<endl;

        cin>>num;

    switch(num){

        case 1: { cout<<"Enter the array elements:-"<<endl;

            for(int i=0;i<n;i++){

                cin>>arr[i];

            }

        }

        case 2: {

            for(int i=0;i<n;i++){

                cout<<arr[i]<<" ";

            }

            cout<<endl;

        }

        case 3: {   cout<<"Enter position where to insert the element: "<<endl;

                    int pos;

                    cin>>pos;

                    for(int i=n;i>=pos;i--){

                        arr[i+1]=arr[i];

                    }

                    int val;

                cout<<"Enter value to be inserted: "<<endl;

                 cin>>val;

                arr[pos]=val;

                for(int i=0;i<n+1;i++){

                    cout<<arr[i]<<" ";

                }

                cout<<endl;

            }

        case 4: { cout<<"Enter position which is to be deleted: "<<endl;

            int pos;

            cin>>pos;

            for (int i=pos;i<n;i++){

                arr[i]=arr[i+1];

            }

            for(int i=0;i<n;i++){

                cout<<arr[i]<<" ";

            }

            cout<<endl;

            }

            case 5 : {  cout<<"Enter the element to be searched linearly:"<<endl;

                        int key;

                        cin>>key;

                        for ( int i=0;i<n;i++){

                            if (arr[i]==key){

                                cout<<"Position is:"<<i<<endl;

                            }

                        }

            }

        case 6 : { cout<<"Enter the element to be searched by binary:"<<endl;

                        int key;

                        cin>>key;

                        int binaryresult= binarysearch(arr,n,key);

                        if (binaryresult!=-1){

                            cout<<"Position is :"<<binaryresult<<endl;

                        }

                        else {

                       cout << "Element not found." << endl;

            }

        }

    }

}

## Output:

**Enter the size of the array:5**

**Menu Driven Program**

**1.Input**

**2.Output**

**3.Insert**

**4.Delete**

**5.Linear Search**

**6.Binary Search**

**1**

**Enter the array elements:-**

**10**

**20**

**30**

**40**

**50**

**10 20 30 40 50**

**Enter position where to insert the element:**

**1**

**Enter value to be inserted:**

**15**

**10 15 20 30 40 50**

**Enter position which is to be deleted:**

**1**

**10 20 30 40 50**

**Enter the element to be searched linearly:**

**50**

**Position is:4**

**Enter the element to be searched by binary:**

**30**

**Position is :2**

## Learning from experiment

* Missing ‘**break’** statements in ‘**switch’** cases.
* Invalid indexing in insert and delete operations.