

Experiment No-07: Searching and Sorting in Arrays.

Objectives

- Implement searching and sorting techniques.
- Solve various problems using one-dimensional arrays.

Example 1: A C++ Program to find the position of an element from the list of elements. [\[Linear Search\]](#)

```
#include<iostream>
using namespace std;

int main()
{
    int arr[15], n, i, to_search, loc;

    cout<<"The number of elements: "<<endl;
    cin>> n;
    cout<<"Enter the numbers: "<<endl;

    for (i = 0; i < n; i++){ //take array input
        cin>> arr[i];
    }
    cout<<"Enter the searching element: ";
    cin>> to_search;        // the number you want to take input
    loc = false;           // Assume that element does not exist in the array

    for (i = 0; i < n; i++){
        if (arr[i] == to_search)
        {
            loc = true;      // if the element is found then set True
            break;
        }
    }

    if (loc == true){

        cout<<"The element "<<to_search<<" is found at location: "<<i+1;

    }
    else{

        cout<<"The element "<<to_search<<" is not in the list."<<endl;

    }

    return 0;
}
```

Example 2: A C++ Program to sort the array elements. [\[Bubble Sort\]](#)

```
#include<iostream>
using namespace std;

int main()
{
    int arr[10] = {5, 1, 4, 2, 8}, i, j, k, temp;

    for (i=0;i<4;i++){
        for (j=0;j<4;j++){
            if (arr[j]>arr[j+1])
            {
                // value swapping
                temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
            cout<<"Iteration: "<<i+1 <<" " <<"Pass --> "<< j+1 <<endl;
            for (k=0; k<5; k++){
                cout<< arr[k]<<" ";
            }
            cout<<endl;
        }
    }
    return 0;
}
```

Example 3: A C++ Program to search an element from the array. [\[Binary Search\]](#)

```
#include<iostream>
using namespace std;

int main()
{
    int beg, last, mid, i, j, k, item ;
    int arr[15] = {11, 22,30, 33, 40, 44, 55, 60, 66, 77, 80, 88, 99};
    beg = 0;
    last = 13;

    cin>>item;
    mid = int ((beg+last)/2); // mid calculation

    while(beg<last && arr[mid]!=item ){

        if(item < arr[mid])
        {
            last = mid - 1;
        }
    }
}
```

```
else {
    beg = mid + 1;
}

mid = int ((beg+last)/2);

}
if(item == arr[mid])
{
    cout<< "Location of the item = "<<item<<" is: "<<mid+1;

}
else{
    cout<<"Item is not found"<<endl;
}

return 0;
}
```

*** Please use your reference book of C programming for a better understanding of the basic syntax of arrays and also include them in your lab report. ***

Practice Exercise

1. Write a C++ program to find the maximum and minimum elements from an array.
2. Write a C++ program to find the second largest element in an array.
3. Write a C++ program to count the total number of even and odd elements in an array.
4. Write a C++ program to print all unique elements in the array.
5. Write a C++ program to count the total number of duplicate elements in an array.
6. Write a C++ program to count the frequency of each element in an array.
7. Write a C++ program to find the reverse of an array.
8. Write a C++ program to insert an element into an array.
9. Write a C++ program to delete an element from an array.