Data Structure Practical 2

Aim: Implementation of searching algorithms.

a) Linear Search

Algorithm:

```
Linear Search ( Array A, Value x)

Step 1: Set i to 1
Step 2: if i > n then go to step 7
Step 3: if A[i] = x then go to step 6
Step 4: Set i to i + 1
Step 5: Go to Step 2
Step 6: Print Element x Found at index i and go to step 8
Step 7: Print element not found
Step 8: Exit
```

Code:

```
#include<iostream>
using namespace std;
void showArray(int *entries,int size);
void linearSearch(int *ptr, int size);
int main() {
	int n,i;
	cout<<"Enter number of elements ";
	cin>>n;
	int arr[n];
	int *ptr=arr;
	for(int i=0;i<n;i++) {
		cout<<"Enter "<<i+1<<"th Element :";
		cin>>ptr[i];
	}
	ptr=arr;
	cout<<"Recorded Details \n";
	showArray(ptr,n);
	linearSearch(ptr,n);
	return 0;
```

```
}
void linearSearch(int *ptr, int size){
	int find;
	int flag=0;
	cout<<"Enter the element to search? ";
	cin>>find;
	for(int i=0;i<size;i++){
		if(ptr[i]==find){
			cout<<"element found at index = "<<i+1;
			flag=1;
			break;
		}
	}
	if(flag==0){
		cout<<"Element not present in given array.";
	}
}
void showArray(int *entries,int size) {
	for(int i=0;i<size;i++){
		cout<<entries[i]<<"\n";
	}
}
```

Output:

```
Enter number of elements 5
Enter 1th Element :45
Enter 2th Element :23
Enter 3th Element :90
Enter 4th Element :1
Enter 5th Element :67
Recorded Details
45
23
90
1
67
Enter the element to search? 1
element found at index = 4
--------------------------------
Process exited after 17.65 seconds with return value 0
Press any key to continue . . .
```

b) Binary Search

Algorithm:

1. Input an array A of n elements and “data” to be sorted
2. LB = 0, UB = n; mid = int ((LB+UB)/2)
3. Repeat step 4 and 5 while (LB <= UB) and (A[mid] ! =data)
4. If (data & lt; A[mid])
   UB = mid–1
5. Else
   LB = mid + 1
6. Mid = int ((LB + UB)/2)

If (A[mid]== data)

   Display “the data found”

8. Else
   Display “the data is not found”
9. Exit

Code:

```
#include<iostream>
using namespace std;
int binarySearch(int list[],int key,int arraySize) {
        int start = 0;
        int end = arraySize - 1;
        int pos;
        int mid = int((start+end)/2);
        while(start <= end && list[mid]!=key) {
                if(key < list[mid])
                end = mid -1;
                else
                start = mid+1;
                mid = int((start+end)/2);
        }
        if(list[mid]==key){
```

```
                pos = mid;
        }
        else {
                pos = -1;
        }
        return pos;
}
int main() {
        int arraySize,key,list[10],pos;
        cout<<"enter number of elements \n";
        cin>>arraySize;
        cout<<"enter "<<arraySize<<" no of elements in ascending order\n";
        for(int i=0;i<arraySize;i++){
                cin>>list[i];
        }
        cout<<"enter the element to search\n";
        cin>>key;
        pos = binarySearch(list,key,arraySize);
        if(pos==-1){
                cout<<"element not found\n";
        }
        else{
                cout<<"element found at the position "<<pos+1;
        }
        return 0;
}
```

Output:

```
enter number of elements
5
enter 5 no of elements in ascending order
45
90
102
679
1048
enter the element to search
90
element found at the position 2
--------------------------------
Process exited after 28.69 seconds with return value 0
Press any key to continue . . .
```