NAME:- SAGAR ADHIKARI
ROLL NO:- PUR078BEI034
FACULTY:- ELECTRONICS(BEI)

DSA LAB SHEET NO. 8

TITLE:- IMPLEMENTATION OF SEARCHING

THEORY:-

Searching:-Searching refers to the process of finding the required information from a collection of items stored as elements in the computer memory.
Linear Search:- Linear Search is defined as a sequential search algorithm that starts at one end and goes through each element of a list until the desired element is found, otherwise the search continues till the end of the data set.
Binary Search:-Binary Search is defined as a searching algorithm used in a sorted array by repeatedly dividing the search interval in half. The idea of binary search is to use the information that the array is sorted and reduce the time complexity to O(log N).

1.Linear Search

PROGRAM CODE:-

```
#include <iostream>
using namespace std;
int LinearSearch(int A[], int n, int key)
{
    int i, flag = 0;
    for (i = 0; i < n; i++)
    {
        if (A[i] == key)
        {
            flag = 1;
            return i + 1; // Position of the element in array starts from 1 not 0.
        }
    }
    if (flag == 0)
        return -1;
}
int main()
{
    int A[100000], n, i, key;
    cout << "Enter n: ";
    cin >> n;
    for (i = 0; i < n; i++)
    {
        A[i] = rand();
        cout << A[i] << ","; // printing the random integers  generated by rand().
```

```
    }
    cout << "\nEnter key: ";
    cin >> key;
    cout << "Key is at pos: " << LinearSearch(A, n, key) << endl;
    return 0;
}
```

OUTPUT:-

Enter n: 14
41,18467,6334,26500,19169,15724,11478,29358,26962,24464,5705,28145,23281,16827,
Enter key: 26500
Key is at pos: 4

2.Binary Search

PROGRAM CODE:-

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

int BinarySearch(int A[], int l, int r, int key)
{
    int m = floor((l + r) / 2);
    while (l <= r)
    {
        if (key == A[m])
        {
            return m + 1;// to start the position from 1
        }
        else if (key < A[m])
        {
            r = m - 1;
        }
        else
            l = m + 1;
        m = floor((l + r) / 2);
    }
    return -1;
}

void swapp(int *x, int *y)
{
```

NAME:- SAGAR ADHIKARI
ROLL NO:- PUR078BEI034
FACULTY:- ELECTRONICS(BEI)

DSA LAB SHEET NO. 8

```
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

void display(int A[], int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << A[i] << " ";
    }
    cout << endl;
}

int partition(int A[], int l, int r)
{
    int x = l;
    int y = r;
    int pivot = A[l];
    while (x < y)
    {
        while (A[x] <= pivot)
        {
            x++;
        }
        while (A[y] > pivot)
        {
            y--;
        }
        if (x < y)
        {
            swapp(&A[x], &A[y]);
        }
    }
    A[l] = A[y];
    A[y] = pivot;
    return y;
}

void quicksort(int A[], int l, int r)
{
```

NAME:- SAGAR ADHIKARI
ROLL NO:- PUR078BEI034
FACULTY:- ELECTRONICS(BEI)

DSA LAB SHEET NO. 8

```
    if (l < r)
    {
        int p = partition(A, l, r);
        quicksort(A, l, p - 1);
        quicksort(A, p + 1, r);
    }
}

int main()
{
    int A[10000], n, i, key;
    cout << "Enter n: ";
    cin >> n;

    for (int i = 0; i < n; i++)
    {
        A[i] = rand();
    }

    cout << "Before sorting" << endl;
    display(A, n);

    quicksort(A, 0, n - 1);//binary search works only for sorted element

    cout << "After sorting" << endl;
    display(A, n);

    cout << "Enter key: ";
    cin >> key;
    cout << "Key is at pos: " << BinarySearch(A, 0, n - 1, key) << endl;

    return 0;
}
```

OUTPUT:-

Enter n: 10
Before sorting
41 18467 6334 26500 19169 15724 11478 29358 26962 24464
After sorting
41 6334 11478 15724 18467 19169 24464 26500 26962 29358
Enter key: 19169
Key is at pos: 6