Experiment-1

**Aim:** To Perform Linear Search and Binary Search on an array

**1.Linear Search:**

**Algorithm:**

\*Start from the leftmost element of arr[] and one by one compare x with each element of arr[].

\*If x matches with an element, return the index.

\*If x doesn’t match with any of the elements, return -1.

**Source Code:**

#include<stdio.h>

int main()

{

    int a[20],i,x,n;

    printf("How many elements?");

    scanf("%d",&n);

    printf("Enter array elements:");

    for(i=0;i<n;++i)

        scanf("%d",&a[i]);

    printf("Enter element to search:");

    scanf("%d",&x);

    for(i=0;i<n;++i)

        if(a[i]==x)

            break;

    if(i<n)

        printf("Element found at index %d",i);

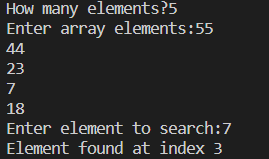
    else

        printf("Element not found");

    return 0;

}

**Output:**

****

**2.Binary Search:**

**Algorithm:**

\*Begin with the mid element of the whole array as a search key.

\*If the value of the search key is equal to the item then return an index of the search key

. \*Or if the value of the search key is less than the item in the middle of the interval, narrow the interval tothe lower half.

\*Otherwise, narrow it to the upper half.

\*Repeatedly check from the second point until the value is found or the interval is empty.

**Source Code:**

#include <stdio.h>

int binarySearch(int array[], int x, int low, int high) {

  if (high >= low) {

    int mid = low + (high - low) / 2;

    if (array[mid] == x)

      return mid;

    if (array[mid] > x)

      return binarySearch(array, x, low, mid - 1);

    return binarySearch(array, x, mid + 1, high);

  }

  return -1;

}

int main(void) {

  int array[] = {10, 12, 15, 19, 23, 29, 31};

  int n = sizeof(array) / sizeof(array[0]);

  int x = 23;

  int result = binarySearch(array, x, 0, n - 1);

  if (result == -1)

    printf("Not found");

  else

    printf("Element is found at index %d", result);

}

**Output:**

