**Experiment No. 9**

**Title:** Linear search.

**Problem Statement:**

Write a C++ program to search an element in an array using Linear search.

**Algorithm:**

Start

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

2. If x matches with an element, return the index.

3. If x doesn’t match with any of elements, return -1.

4.stop

**Code:**

//Linear Search

#include<iostream>

using namespace std;

int main()

{

int arr[20],n,x,i,flag=0;

cout<<"How many elements:";

cin>>n;

cout<<"\nEnter elements of the array:\n";

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

cin>>arr[i];

cout<<"\nEnter element to search:";

cin>>x;

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

{

if(arr[i]==x)

{

flag=1;

break;

}

}

if(flag)

cout<<"\nElement is found at position "<<i+1;

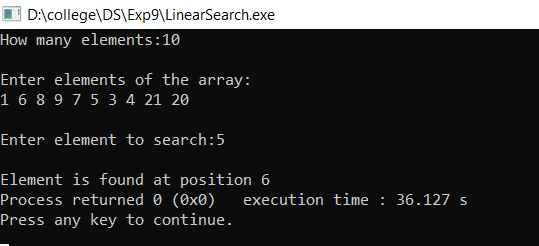
else

cout<<"\nElement not found";

return 0;

}

**Output:**



**Analysis:**

Limitations of This Implementation are as follows:

1.Inversely, when a key element matches the last element in the array or a key element doesn't matches any element then Linear search algorithm is a worst case.

2.As we have seen throughout this tutorial that Linear Search is certainly not the absolute best method for searching but do not let this taint your view on the algorithm itself.

3.People are always attempting to better versions of current algorithms in an effort to make existing ones more efficient.

4. Not to mention that Linear Search as shown has its place and at the very least is a great beginner‘s introduction into the world of searching algorithms.

5. With this is mind we progress to the asymptotic analysis of the Linear Search: