**Experiment No. 9**

**Title :** Implementation of Linear Search

**Problem Statement :** Write a C++ program to implement Linear Search algorithm

**Algorithm:**

**Step 1:** Start

**Step 2:** Input the array elements.

**Step 3:** Input the element to be searched.

**Step 4:** traverse through the whole array and compare each element with the element to be searched

**Step 5:** if element matches set flag = 1 and break the traversal else flag = 0

**Step 6:** if flag is 1 display element found with the index as position else not found

**Step 7:** Stop

**Program:**

//Linear Search

#include<iostream>

using namespace std;

int main()

{

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

cout<<"How many elements ?";

cin>>n;//input number of element

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

for(i=0;i<n;++i)//input the element

cin>>arr[i];

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

cin>>x;//input the search element

for(i=0;i<n;++i)//travel and search in the array

{

if(arr[i]==x)//compare and update the flag

{

flag=1;

break;

}

}

if(flag)//depending on the flag give result

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

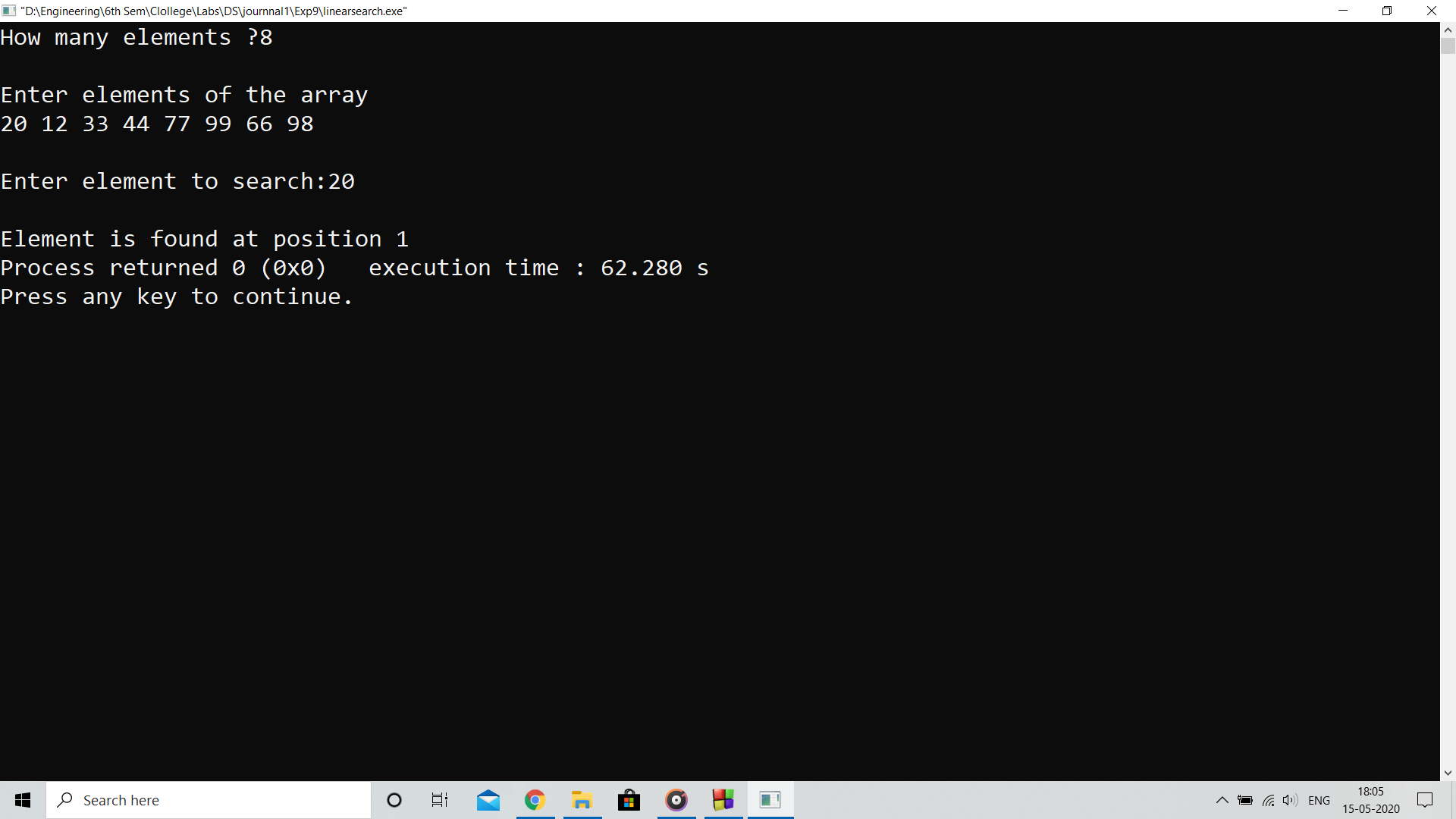
else

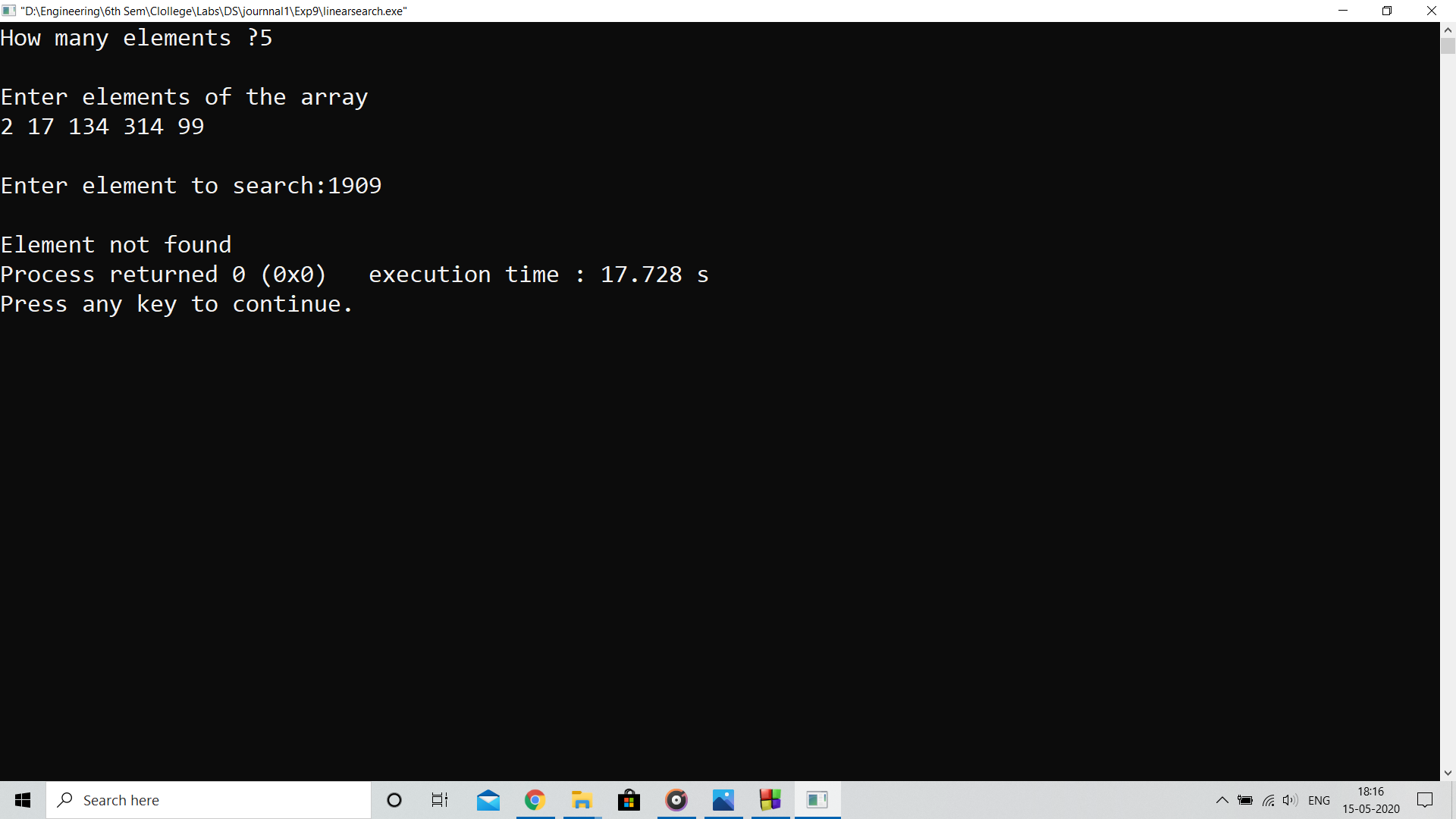
cout<<"\nElement not found";

return 0;

}

**Output :**

****

****

**Analysis :**

Linear search compares each element by element of array and tries to match thus it takes maximum time of the order O(n) to find an element in the array of ‘n’ elements

**Limitation:**

Program will take longer to find element if the total number of elements are in range of MBytes or GBytes and the element is at far index from base address which increases the latency of overall software using it.