**Experiment No. 10**

**Title :** Implementation of Binary Search

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

**Algorithm:**

**Step 1:** Start

**Step 2:** Input the array elements in ascending order.

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

**Step 4:** Searching

* Compare x with the middle element of the sorted array.
* If x matches with middle element, we return the mid index
* Else If x is greater than the mid element, then x can only lie in right half subarray after the mid element. So we recur for right half.
* Else (x is smaller) recur for the left half.

**Step 5:** Repeat the Step 5 until lower index remain smaller than higher index

**Step 6:** If lower index exceeds higher index then display element not found

**Step 7:** Stop

**Program:**

// Binary Search using Recursion

#include <iostream>

using namespace std;

// If found, location of x in the array is returned.

// otherwise -1 is returned.

int binarySearch(int arr[], int first, int last, int x)

{

if (last >= first)

{

int mid = first + (last- first) / 2;

// If the element is present at the middle of the array

if (arr[mid] == x)

return mid;

// If element is smaller than mid, then

// it can only be present in left subarray

if (arr[mid] > x)

return binarySearch(arr, first, mid - 1, x);

// Else the element can only be present

// in right subarray

return binarySearch(arr, mid + 1, last, x);

}

// If the element is not found.

return -1;

}

int main()

{

int arr[20],n,x,i;

cout<<"How many elements?";

cin>>n;//input number of elements

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

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

cin>>arr[i];

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

cin>>x;//input search element

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

if (result == -1)

cout << "Element is not present in array.";

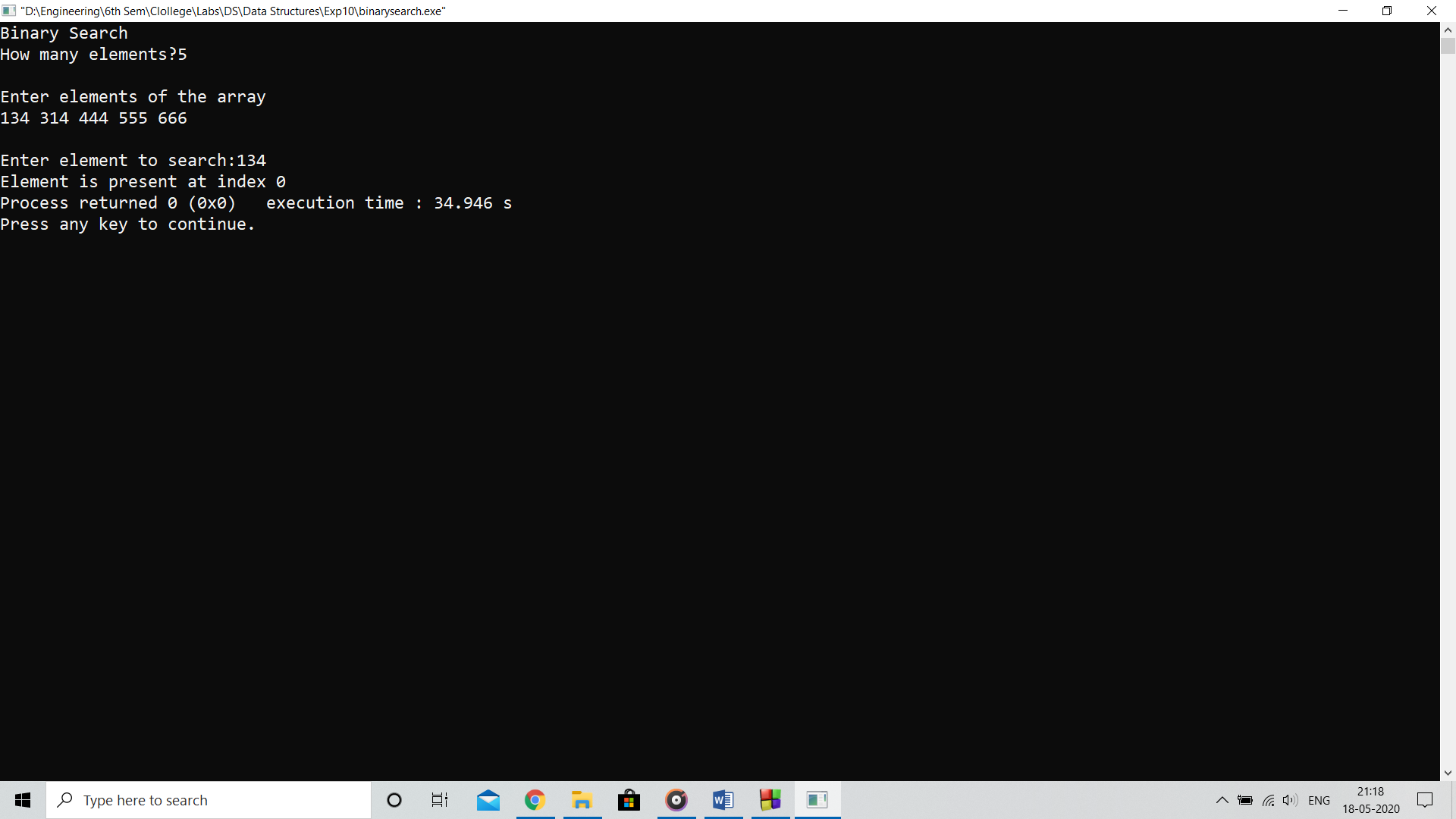
else

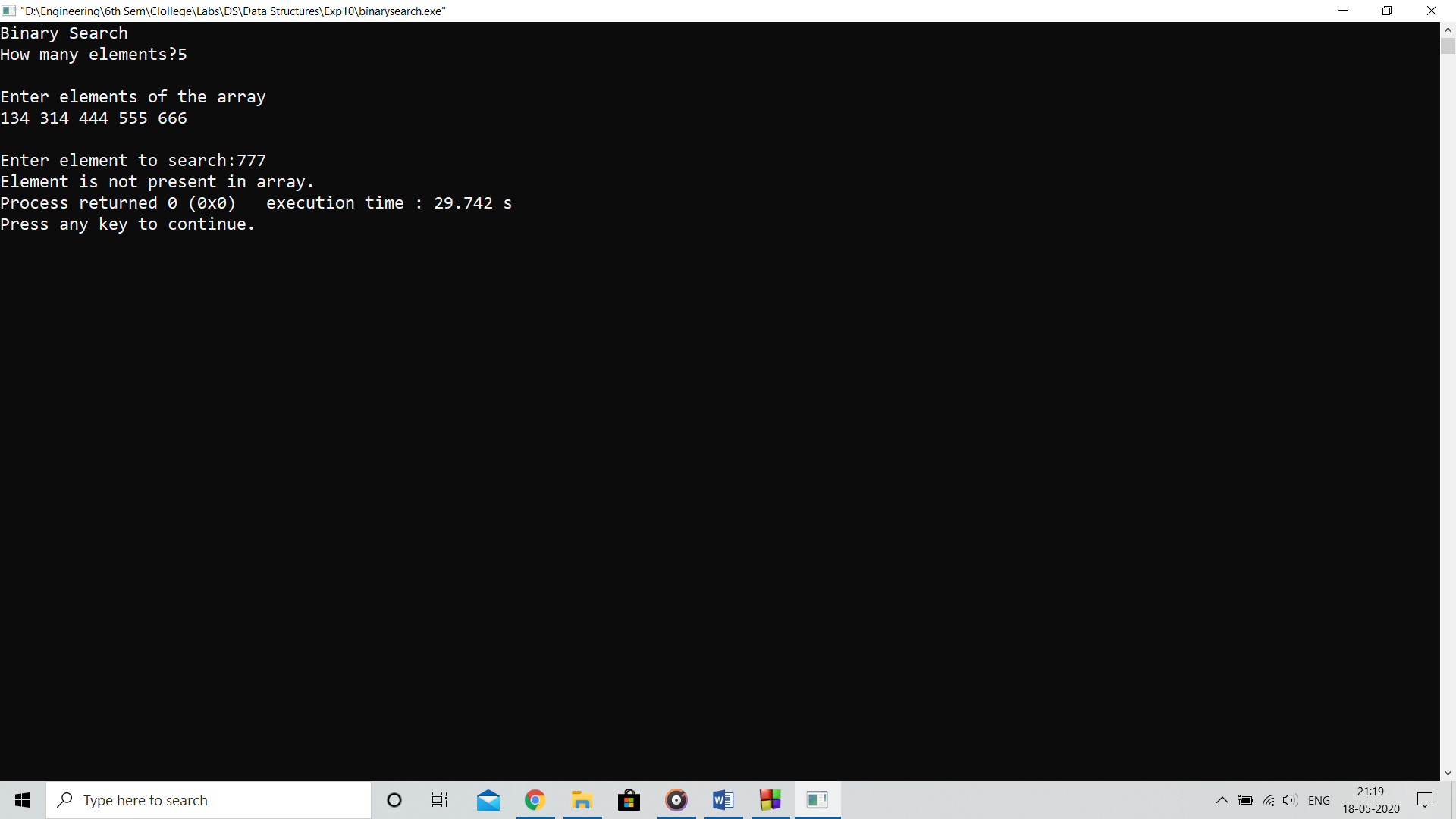
cout << "Element is present at index "<<result;

return 0;

}

**Output :**

****

****

**Analysis :**

Binary search is more efficient way than linear search as the searching is processed by dividing the array in smaller parts which reduces the computational time for searchinh

**Limitation:**

Binary search algorithm needs the data to be sorted in ascending order to work properly.