Experiment No:10

Title: Implementation of Binary Search.

Problem statement: Write a C++ program to search an element in an array using Binary search.

Algorithm:

Step1: Compare x with the middle element of the sorted array

**Step 2** **:** If x matches with middle element, we return the mid index

**Step 3 :** 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.

**Step 4:** Else (x is smaller) recur for the left half.

Code:

#include <iostream>

using namespace std;

int main()

{

int count, i, arr[30], num, first, last, middle;

cout<<"how many elements would you like to enter?:";

cin>>count;

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

{

cout<<"Enter number "<<(i+1)<<": ";

cin>>arr[i];

}

cout<<"Enter the number that you want to search:";

cin>>num;

first = 0;

last = count-1;

middle = (first+last)/2;

while (first <= last)

{

if(arr[middle] < num)

{

first = middle + 1;

}

else if(arr[middle] == num)

{

cout<<num<<" found in the array at the location "<<middle+1<<"\n";

break;

}

else {

last = middle - 1;

}

middle = (first + last)/2;

}

if(first > last)

{

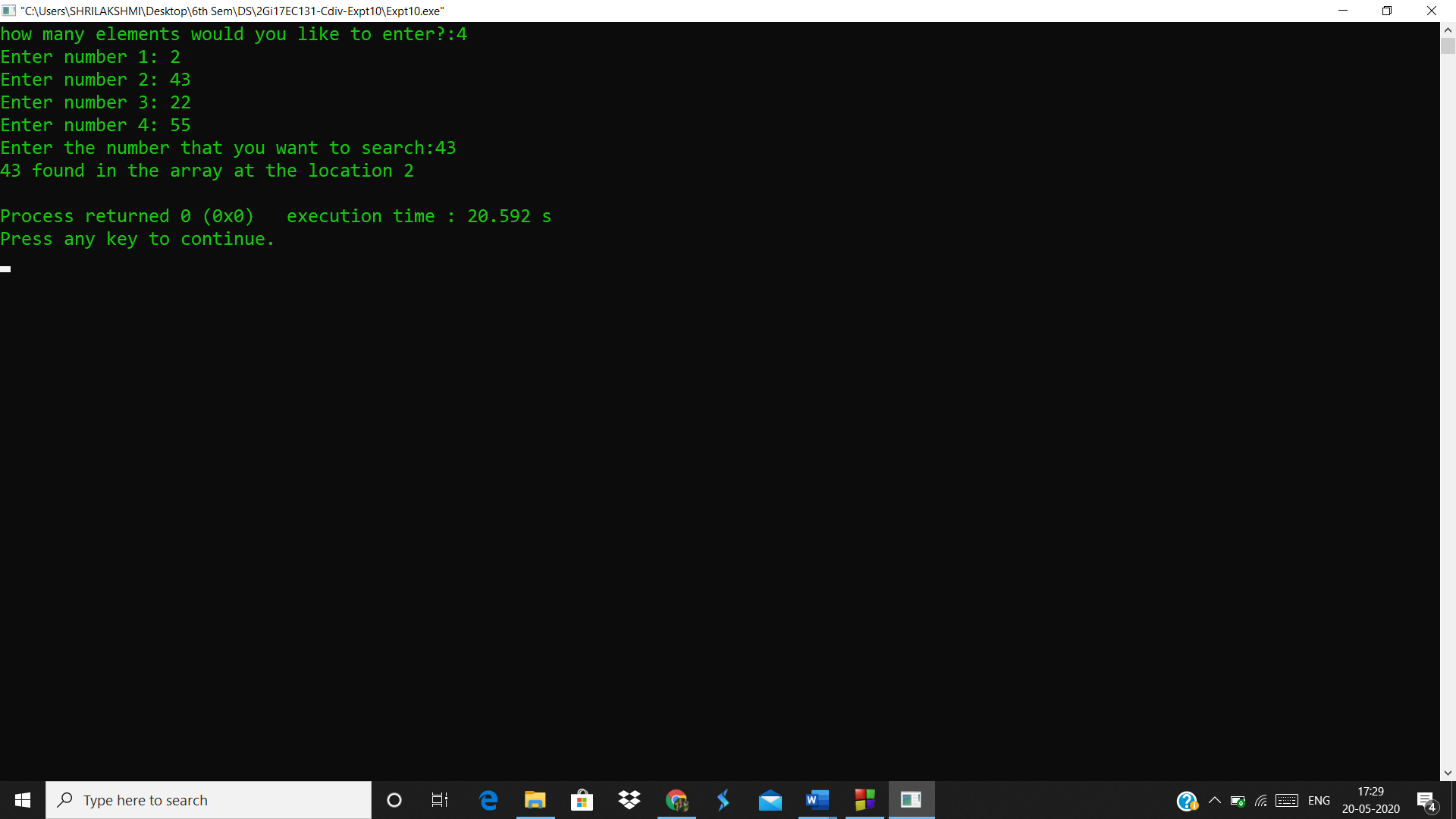
cout<<num<<" not found in the array";

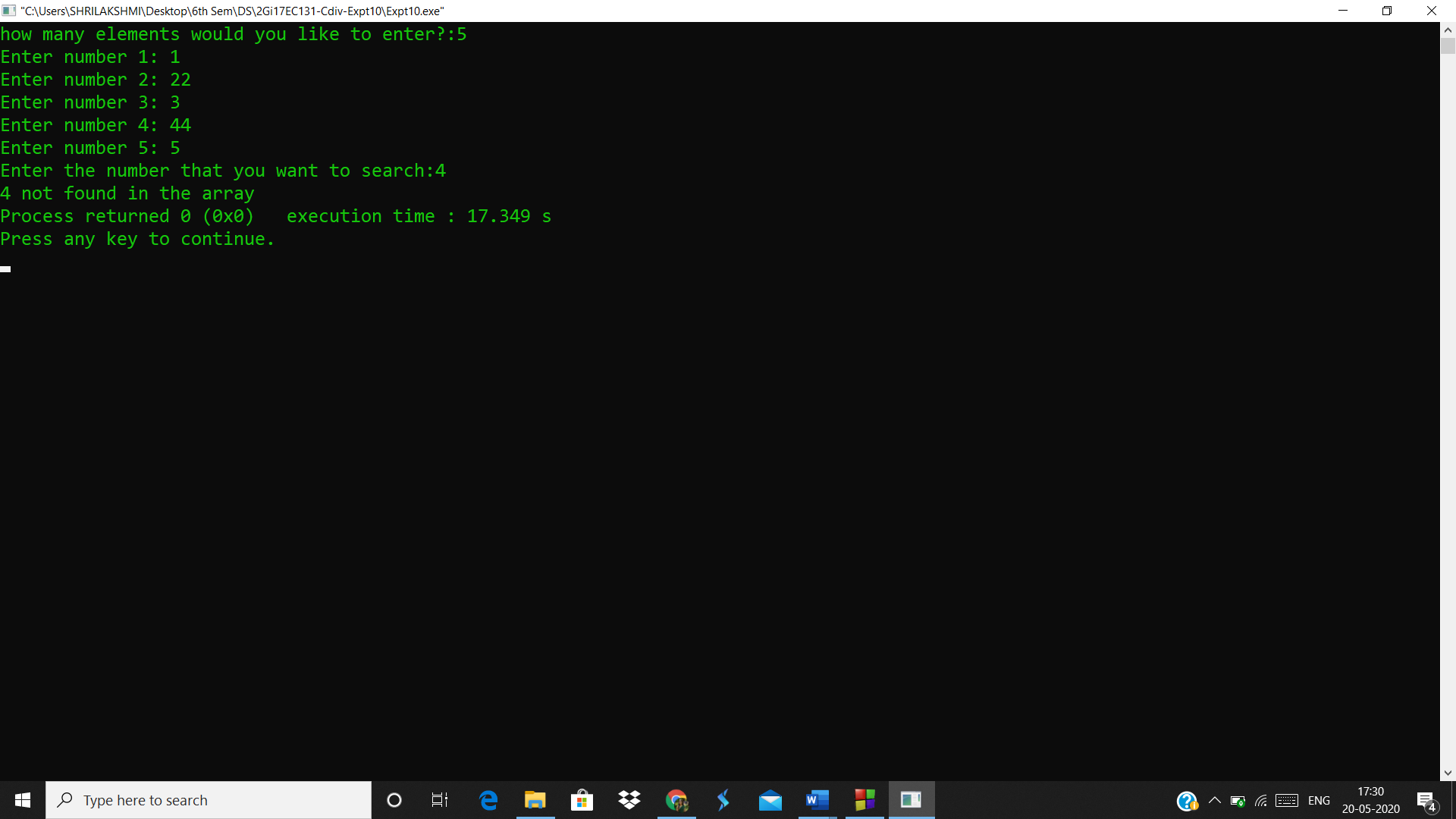
}

return 0;

}

Output:





Analysis(limitations):

* It's more complicated than linear search, and is overkill for very small numbers of elements.
* There is a great lost of efficiency if the list does not support random-access.