Experiment No:7

Title: Implementation of Bubble sort in C.

Problem Statement: Implementation of Bubble sort in C.

Algorithm:

1. Start
2. Input the elements and store in an array format.
3. Compare adjacent elements in an array, swap the highest element gradually towards right.
4. Bubble out the highest value element to the end of array.
5. Reduce the comparison limit of elements in each Iteration.
6. Store and print the sorted array
7. Stop

CODE:

// Bubble Sort

#include<iostream>

using namespace std;

int main()

{

int array[50], n, i, j, k, temp;

cout<<"Enter the size of array: ";

cin>>n;

cout<<"Enter the array elements to be sorted: ";

//To read the array

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

cin>>array[i];

cout<<"Array before sorting :";

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

cout<<" "<<array[i];

for(i=1;i<n;++i) // i keeps track of the no. of passes

{

{

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

if(array[j]>array[j+1])

{

temp=array[j];

array[j]=array[j+1];

array[j+1]=temp;

}

}

//To print the array elements after every pass

cout<<"\n\n Array after Pass "<<i<<":" ;

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

cout<<" "<<array[k];

}

cout<<"\n\nArray after bubble sort:";

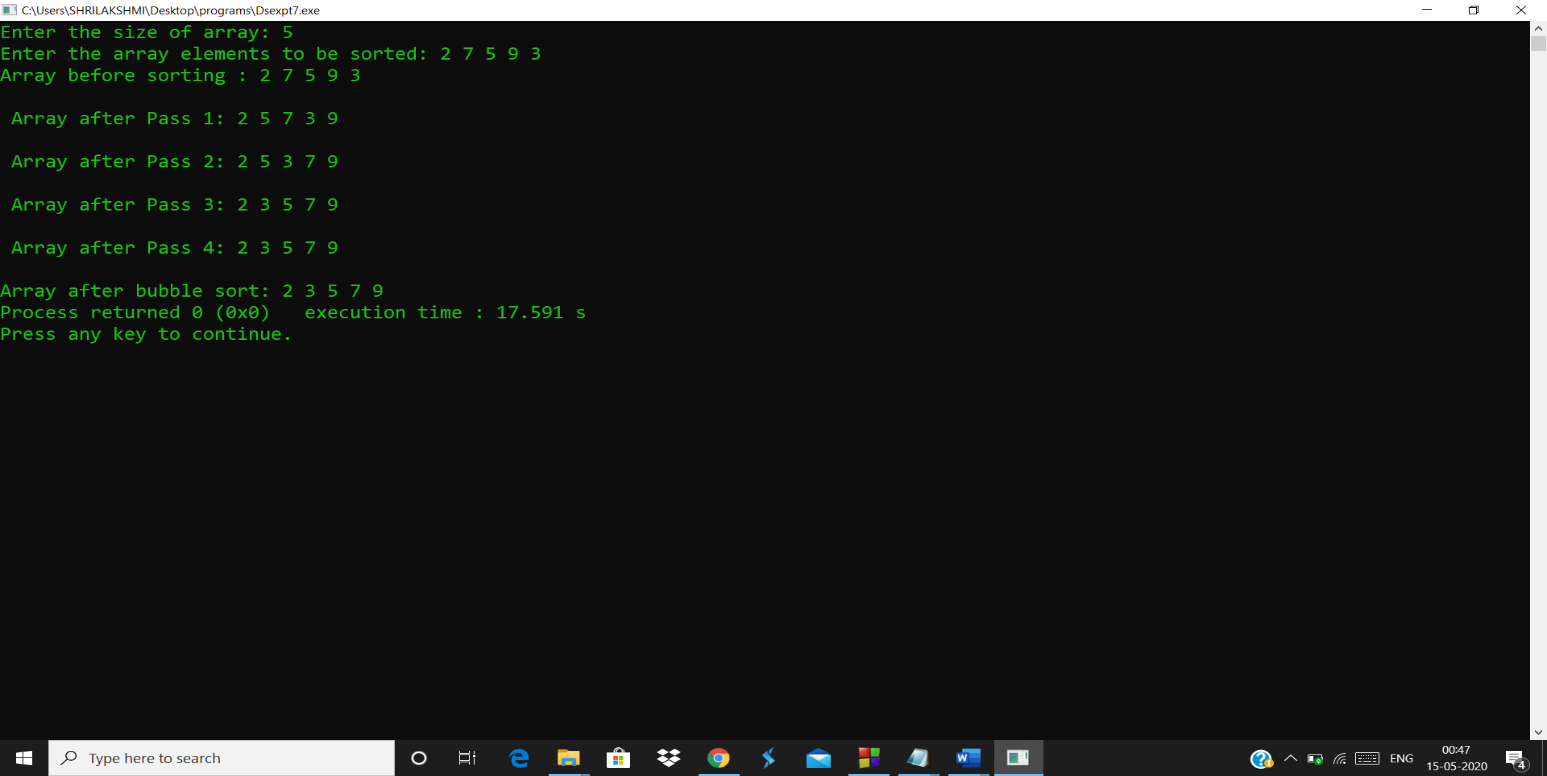
for(i=0;i<n;++i)

cout<<" "<<array[i];

return 0;

}

Output:



Analysis(Limitations):

* The major disadvantage is the amount of time it takes to sort. The average time increases almost exponentially as the number of table elements increase. Ten times the number of items takes almost one hundred times as long to sort.