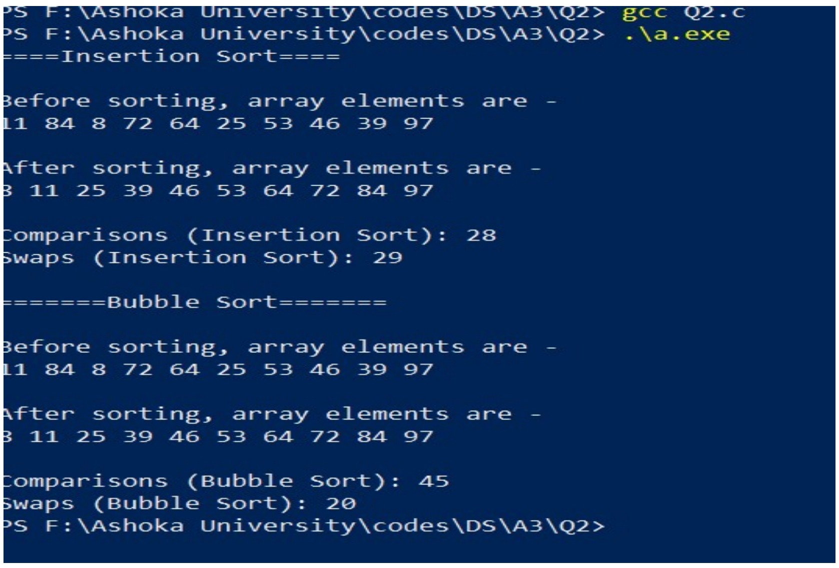
Theoretical Time Complexity:

The theoretical time complexity of Insertion Sort is O(n^2) in the average and worst-case scenarios. This is due to the nested loops involved in the algorithm. The outer loop iterates over each element in the array, and the inner loop iterates over the already sorted portion of the array to find the correct position for the current element. This results in a quadratic time complexity, O(n^2), where 'n' represents the number of elements in the array.

In the case of Bubble Sort, the theoretical time complexity is also O(n^2) in the average and worst-case scenarios. Bubble Sort compares adjacent elements in the array and swaps them if they are in the wrong order. This process is repeated for each element in the array until no more swaps are needed, indicating that the array is sorted. The nested loops in Bubble Sort, where the outer loop iterates over each element, and the inner loop compares adjacent elements, result in a quadratic time complexity, O(n^2).

Experimental Results:

For the given program in the Q2.c file, the output is as follows.



```
PS F:\Ashoka University\codes\DS\A3\Q2> gcc Q2.c
PS F:\Ashoka University\codes\DS\A3\Q2> .\a.exe
====Insertion Sort====

Before sorting, array elements are -
11 84 8 72 64 25 53 46 39 97

After sorting, array elements are -
8 11 25 39 46 53 64 72 84 97

Comparisons (Insertion Sort): 28
Swaps (Insertion Sort): 29

=======Bubble Sort=======

Before sorting, array elements are -
11 84 8 72 64 25 53 46 39 97

After sorting, array elements are -
8 11 25 39 46 53 64 72 84 97

Comparisons (Bubble Sort): 45
Swaps (Bubble Sort): 20
PS F:\Ashoka University\codes\DS\A3\Q2>
```

- For Insertion sort, Comparision 28, Swap 29
- For Bubble sort, Comparision 45, Swap 20

Where n= 10 (no. of elements in the array, thus matching with the theoretical prediction.