CS1203 - Monsoon 2023 - Question 2

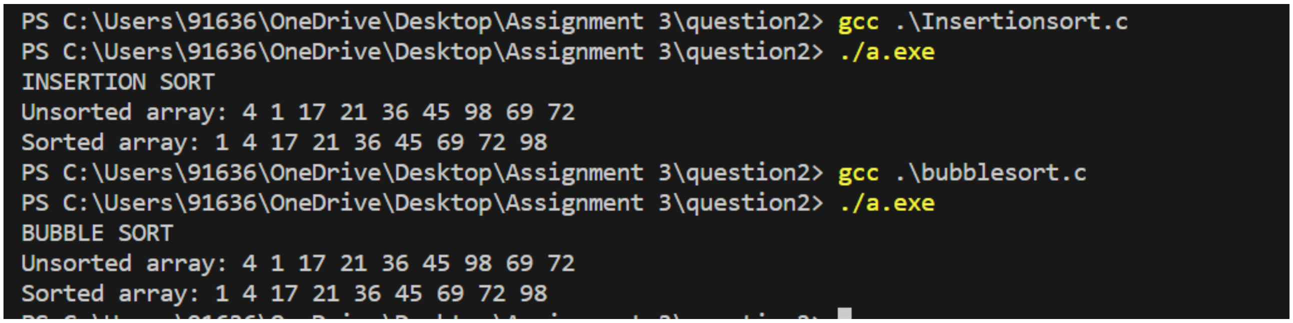
Niyati Rao, Student ID: 1020211512

Collaborators: NONE

Question 2

Comparing the time complexity of insertion sort with bubble sort.

- **INSERTION SORT:**
  Insertion sort is an in-place sorting algorithm, meaning it doesn't require additional memory for sorting. and has a space complexity of O(1). It builds a sorted list one element at a time by comparing and inserting each element into its correct position within the sorted portion of the list. going from left to right by comparing with predecessors one at a time.
  - **Best case:**
    The best case scenario occurs when the array is sorted in the case of insertion sort this would require one to traverse through the array at least once there the time complexity would be O(N)
  - **Worst case:** In the worst-case scenario where the array to be sorted in ascending order is in descending order or vice versa the time complexity would be O(N^2) this is because on iteration with each element in the array, it is compared to all the previous elements in order to insert it at its correct position in the worst case.
- **BUBBLE SORT:**
  Like insertion sort bubble sort is also an in-place sorting algorithm, meaning it doesn't require additional memory for sorting. and has a space complexity of O(1). Bubble sort is a basic sorting algorithm that repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. This process is repeated until the entire list is sorted.
  - **Best case:**
    The best case scenario occurs when the array is sorted in the case of insertion sort this would require one to traverse through the array the time complexity would therefore be O(N)
  - **Worst case:** In the worst-case scenario where the array to be sorted in ascending order is in descending order or vice versa the time complexity would be O(N^2) this is because on iteration with each element in the array, it is compared to all the later elements in order to bubble it to its final position. On each iteration the algorithm solves for one position from the end.



```
PS C:\Users\91636\OneDrive\Desktop\Assignment 3\question2> gcc .\Insertionsort.c
PS C:\Users\91636\OneDrive\Desktop\Assignment 3\question2> ./a.exe
INSERTION SORT
Unsorted array: 4 1 17 21 36 45 98 69 72
Sorted array: 1 4 17 21 36 45 69 72 98
PS C:\Users\91636\OneDrive\Desktop\Assignment 3\question2> gcc .\bubblesort.c
PS C:\Users\91636\OneDrive\Desktop\Assignment 3\question2> ./a.exe
BUBBLE SORT
Unsorted array: 4 1 17 21 36 45 98 69 72
Sorted array: 1 4 17 21 36 45 69 72 98
```

Figure 1: An instance of execution of Insertion and Bubble sort