**LABWORK 5:**

**GROUP : CE**

**Sadikshya Pokharel Roll no: 36   
 Yural Pokhrel Roll no: 37**

**INTRODUCTION:**

In this lab work, we have implemented following sorting algorithms:

* Quick sort
* Insertion sort

**IMPLEMENTATION**:

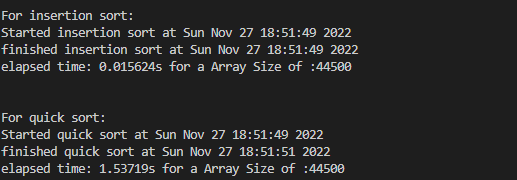
In quick sort: We use divide and conquer strategy, where we select a pivot and make partition such that the elements on left sublist is less than pivot and that in right sublist are greater than pivot. The sublist are recursively sorted to get a final sorted list.

In insertion sort: The list is divided into two parts: sorted and unsorted. In each pass of an insertion sort, the first element of the unsorted sublist is inserted into its correct location in the sorted sublist.

We have provided arrays of different sizes that are populated with random numbers. We have used rand() function for this purpose. We have calculated the start and end time of both sorting algorithms, and then calculated the time elapsed.

**OUTPUT:**

Below inserted are the screenshots output of the program and of scatter plot obtained by plotting input size vs the execution time for both algrithms.



The time complexity for quick sort is : worst case : O(n2) and best case : O(n log2 n)

The time complexity for insertion sort is : worst case : O(n2) and best case : O(n)

**Observation** : For small sized arrays, insertion sort is preferred , whereas for large sized arrays, we prefer quick sort.