# Experiment No- 04

**Title:-** Python program to compute Selection Sort and Bubble sort algorithm

**Objectives:-** To understand the use of sorting using algorithm

# Problem Statement:-

Write a Python program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using

1. Selection Sort
2. Bubble sort and display top five scores.

# Theory-

**Selection Sort Algorithm (Ascending order)**

Step 1 − Set MIN to location 0

Step 2 − Search the minimum element in the list Step 3 − Swap with value at location MIN

Step 4 − Increment MIN to point to next element Step 5 − Repeat until list is sorted

Consider the following array as an **example**.



For the first position in the sorted list, the whole list is scanned sequentially. The first position where 14 is stored presently, we search the whole list and find that 10 is the lowest value.



So we replace 14 with 10. After one iteration 10, which happens to be the minimum value in the list, appears in the first position of the sorted list.



Following is a pictorial depiction of the entire sorting process −



# Bubble Sort Algorithm: (Ascending order)

Step 1. Starting with the first element(index=0) compare the current element with the next element of the list.

Step 2. If the current element is greater than the next element of the list swap them. Step 3. If the Current element is less than the next element, move to the next element.

Repeat step 1

**Example**. Bubble sort takes Ο(n2) time so we're keeping it short and precise.



Bubble sort starts with very first two elements, comparing them to check which one is greater.



In this case, value 33 is greater than 14, so it is already in sorted locations. Next, we compare 33 with 27.



We find that 27 is smaller than 33 and these two values must be swapped.



The new array should look like this −



Next we compare 33 and 35. We find that both are in already sorted positions.



Then we move to the next two values, 35 and 10.



We know then that 10 is smaller 35. Hence they are not sorted.



We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this −



To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this −



Notice that after each iteration, at least one value moves at the end.



And when there's no swap required, bubble sorts learns that an array is completely sorted.



# Algorithm:-

1. Start
2. Declare the variables such as,n array,ele,ch,i=0
3. Input how many no. of students in first year
4. Input the Enter percentage marks of students
5. Print Menu:
   1. Selection Sort
   2. Bubble Sort and display Top 5 scores
   3. Exit

ch= Input enter your choice

1. If ch==1 then call function SelectionSort(array,n)
2. else if ch==2 then call function BubbleSort(array,n)
3. else If ch==3 then go to step 9 else print wrong choice
4. Stop

# Conclusion:

In this way, we perform Sorting of marks using Selection sort and Bubble sort algorithm.