**EX NO: 6 SELECTION SORT AND INSERTION SORT**

**Date :**

**AIM**

To write a python program for sorting the given set of inputs using selection sort and insertion sort

sample input 0: [5,4,0,29,2]

sample output 0:[0,2,4,5,29]

sample input 1: [5,-4,0,29,2]

sample output 1:[-4,0,2,5,29]

sample input 2: [5,4.3,0,29,4.2]

sample output 2:[0,4.2,4.3,5,29]

sample input 2: ['b','a','c']

sample output 2:['a','b','c']

**ALGORITHM**

**Selection sort:**

**Step1:** Set max\_pos to location 0

**Step2:**  Search the maximum element in the list

**Step 3:** Swap with value at location max\_pos

**Step 4:** Increment max\_pos to point to the next element

**Step 5:** Repeat until list is sorted

**Insertion sort:**

**Step1:** Choose the first two elements in the list

**Step2:**  If it is sorted leave as it is

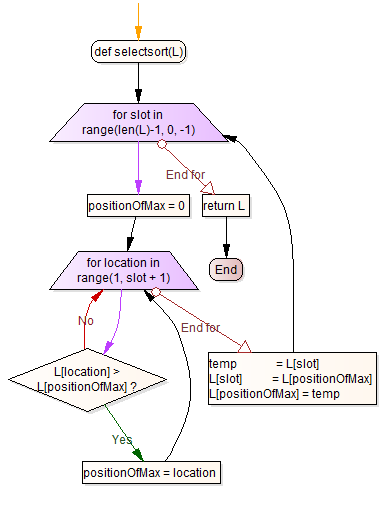
**Step 3:** If not swap the element with the previous

**Step 4:** Check whether the elements in the sub list is sorted

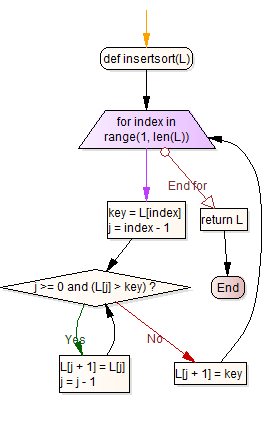
**Step 5:** Repeat until list is sorted

**FLOWCHART**

**Selection sort:**



**Insertion sort:**

****

**PRE-LAB QUESTIONS**

1. What is the output of selection sort after the 1st iteration given the following sequence of numbers: 14 9 4 18 45 2 37 6
2. What is the worst case complexity for selection sort algorithm?
3. What is the average case complexity for selection sort algorithm?
4. What is the output of selection sort after the 2nd iteration given the following sequence of numbers: 16 3 46 9 28 14?
5. What is the best case complexity for selection sort algorithm?
6. In a selection sort of n elements, how many times is the swap function called in the complete execution of the algorithm?
7. A sorting technique in which successive elements are selected in order and placed into their proper sorted positions is called?
8. In which cases are the time complexities same in selection sort?

**SOURCE CODE**

**Selection sort**

def selectsort(L):

for slot in range(len(L)-1, 0, -1):

positionOfMax = 0

for location in range(1, slot + 1):

if L[location] > L[positionOfMax]:

positionOfMax = location

temp = L[slot]

L[slot] = L[positionOfMax]

L[positionOfMax] = temp

return L

**Insertion sort**

def insertsort(L):

for index in range(1, len(L)):

key = L[index]

j = index - 1

while j >= 0 and (L[j] > key):

L[j + 1] = L[j]

j = j - 1

L[j + 1] = key

return L

**OUTPUT**

Enter number: 5, 4, 0, 29, 2

Output : 0, 2, 4, 5, 29

**RESULT**

Thus the python program for performing the selection and insertion sort was executed and verified successfully.

**POST-LAB QUESTIONS**

1. How to remove the duplicates from the resultant array?
2. Rewrite the selection sort code above to sort in ascending order