SORTING

(I) Bubble Sorting
(II) Insertion Sorting
(III) Selection Sorting

Sorting is the process of arranging a given collection of element in some particular order

Algorithm : Bubble Sort

BUBBLESORT(numList, n)

Step 1 SET i = 0
2 WHILE i<n REPEAT STEPS 3 to 8
3 SET j = 0
4 WHILE j < n-i-1, REPEAT STEPS 5 to 7
5 IF numList[j] > numList[j+1] THEN
6 swap(numList[j], numList[j+1])
7 SET j = j+1
8 SET i = i+1

Program Implementation of Bubble Sort using Python

```
def bubble_Sort(list1)
    n = len(list1)
    for i in range(n):
        for j in range(0, n-i-1)
```

Algorithm: Selection Sort

SELECTIONSORT(numList, n)

Step 1: SET i = 0
2: WHILE i<n REPEAT STEPS 3 to 11
3: SET min = i, flag = 0
4: SET j = i+1
5: WHILE j<n, REPEAT STEPS 6 to 10
6: IF numList[j] < numList[min] THEN
7: min = j
8: flag = 1
9: IF flag = 1 THEN
10: swap(numList[i], numList[min])
11: SET i = i+1

Implementation of Selection Sort using Python

```
def selection_sort(list2)
    flag = 1        # to decide when to swap
    n = len(list2)
    for i in range(n)
        min = i
        for j in range(i+1, len(list2)):
```

```
min = j
flag = 1
if flag == 1:
        list2[min], list2[i] = l2[i], last2[i]
numList = [8, 7, 13, 1, -9, 4]
selection_Sort(numList)
print("The sorted list is:")
  for i in range(len(numList)):
        print(numList[i], end=' ')
```

INSERTION SORT

Algorithm : Insertion Sort

INSERTIONSORT(numList, n)

Step 1: SET i = 1
Step 2: WHILE i<n REPEAT STEPS 3 to 9
Step 3: temp = numList[i]
Step 4: SET j = i-1
Step 5: WHILE j >= 0 and numList[j] >
        REPEAT STEPS 6 & 7
Step 6: numList[j+1] = numList[j]
Step 7: SET j = j-1
Step 8: numList[j+1] = temp
        temp at position j
Step 9: set i = i+1