WEEK - 2

SORTING TECHNIQUES

2.1 OBJECTIVE:

a. Write Python script for implementing Bubble sort techniques to arrange a list of integers in ascending order.
b. Write Python script for implementing insertion sort techniques to arrange a list of integers in ascending order.
c. Write Python script for implementing selection sort techniques to arrange a list of integers in ascending order.

2.2 RESOURCES:

Python 3.4.0

2.3 PROGRAM LOGIC:

Bubble Sort Algorithm

```
Algorithm  bubblesort ( x[],  n)
 { // x[1:n] is an array of n elements
     for i := 0 to  n do
     {     for j := 0 to  n–i-1 d0
            { if (x[j] > x[j+1])
                 {
                temp = x[j];
               x[j] = x[j+1];
               x[j+1] = temp;
                }
    }       }
}
```

Example: Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are not in order.

First Pass

5	1	4	2	8	Compare the first two elements, and swaps since 5 > 1
1	5	4	2	8	Compare 5 and 4 and swap since 5 > 4
1	4	5	2	8	Compare 5 and 2 and swap since 5 > 2
1	4	2	5	8	Compare 5 and 8 and since 5 < 8, no swap.

Second Pass

1	4	2	5	8	Compare the first two elements, and as 1 < 4, no swap.
1	4	2	5	8	Compare 4 and 2, swap since 4 > 2
1	2	4	5	8	Compare 4 and 5, no swap.
1	2	4	5	8	Compare 5 and 8 and no swap.

Third Pass

1	2	4	5	8	Compare the first two elements and no swap.
1	2	4	5	8	Compare 2 and 4, no swap.
1	2	4	5	8	Compare 4 and 5, no swap.
1	2	4	5	8	Compare 5 and 8, no swap.

Fourth Pass

1	2	4	5	8	Compare the first two elements and no swap.
1	2	4	5	8	Compare 2 and 4, no swap.
1	2	4	5	8	Compare 4 and 5, no swap.
1	2	4	5	8	Compare 5 and 8, no swap.

Insertion Sort Algorithm

```
Algorithm insertionsort (a, n)
{//sort the array a[1:n] in ascending order
    for j:=2 to n do
    {
           item:=a[j];
           i=j-1;
       while((i>1) and (item< a[i])) do
       {
             a[i+1]:=a[i];
            i:=i-1;
       }
       a[i+1]:=item;
    }
}
```

Example: This is an in-place comparison-based sorting algorithm. Here, a sub-list is maintained which is always sorted. An element which is to be inserted in this sorted sub-list, has to find its appropriate place and then it has to be inserted. Consider an unsorted list with 8 elements.

14	33	27	10	35	19	42	44	Compares the first two elements 14 and 33 and these element are already in ascending order. Now 14 is in sorted sub-list.
14	33	27	10	35	19	42	44	Compare 33 with 27 and 33 is not in correct position. So swap 33 with 27.
14	27	33	10	35	19	42	44	Now we have 14 and 27 in the sorted sub-list.
14	27	33	10	35	19	42	44	Next compare 33 with 10.
10	14	27	33	35	19	42	44	Compare 27 with 10 and 14 with 10. Insert 10 in the proper place.
10	14	27	33	35	19	42	44	Compare 33 and 35, no swap.
10	14	27	33	35	19	42	44	Compare 35 with 19
10	14	27	33	35	19	42	44	Compare 33 with 19, 27 with 19,
10	14	19	27	33	35	42	44	Compare 35 with 42, no swap.
10	14	19	27	33	35	42	44	Compare 42 with 44, no swap.

Selection Sort Algorithm

```
Algorithm selectionSort ( low, high )
 { //a[low : high] is an array  of size n
     i=0, j=0, temp=0, ;
    for i: =low to high do
    {
          minindex = i;
          for  j: =i+1 to  high do
          {
                     if( a[j] < a[minindex] ) then
```

```
                minindex := j; }
            temp := a[i];
            a[i] := a[minindex];
            a[minindex] := temp;
        }
}
```

Example: The selection sort algorithm sorts an array by repeatedly finding the minimum element (considering ascending order) from unsorted part and putting it at the beginning. In every iteration of selection sort, the minimum element from the unsorted sub-array is picked and moved to the sorted sub-array.

Consider a list a = [64, 25, 12, 22, 11]

Index	**0**	**1**	**2**	**3**	**4**	**Remarks**
List	64	25	12	22	11	Find the minimum element in a[0 .. 4] and place it at beginning.
Iteration 1	11	25	12	22	64	Find the minimum element in a[1 .. 4] and place it at beginning of a[1 .. 4]
Iteration 2	11	12	25	22	64	Find the minimum element in a[2 .. 4] and place it at beginning of a[2 .. 4]
Iteration 3	11	12	22	25	64	Find the minimum element in a[3 .. 4] and place it at beginning of a[3 .. 4]
Iteration 4	11	12	22	25	64	Finally the list is sorted.

2.4 PROCEDURE:

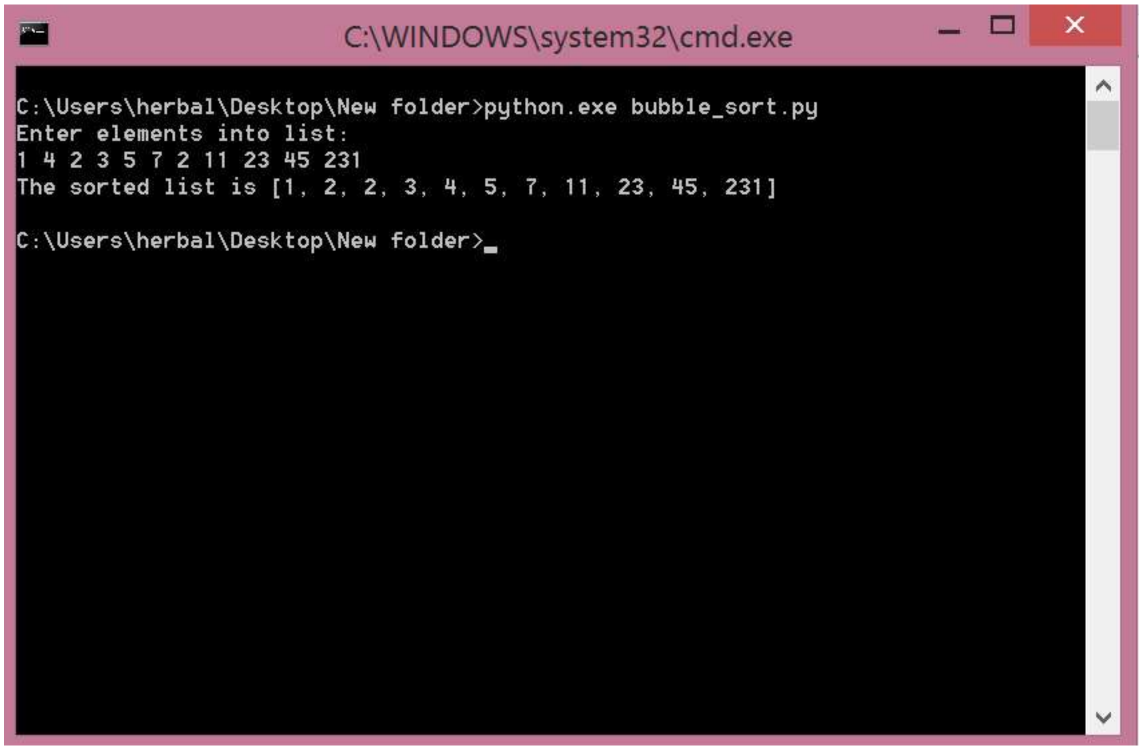
1. Create: open Python shell write a program after that save the program with .py extension.
2. Execute: F5

2.5 SOURCE CODE:

Program for implementing Bubble Sort

```
def b_sort(a):
    n = len(a)
    for i in range(n):
        for j in range(0, n-i-1):
            if a[j] > a[j+1] :
                a[j], a[j+1] = a[j+1], a[j]

print("Enter elements into list:")
a=[int(x) for x in input().split()]
b_sort(a)
print("The sorted list is ",a)
```

Output:

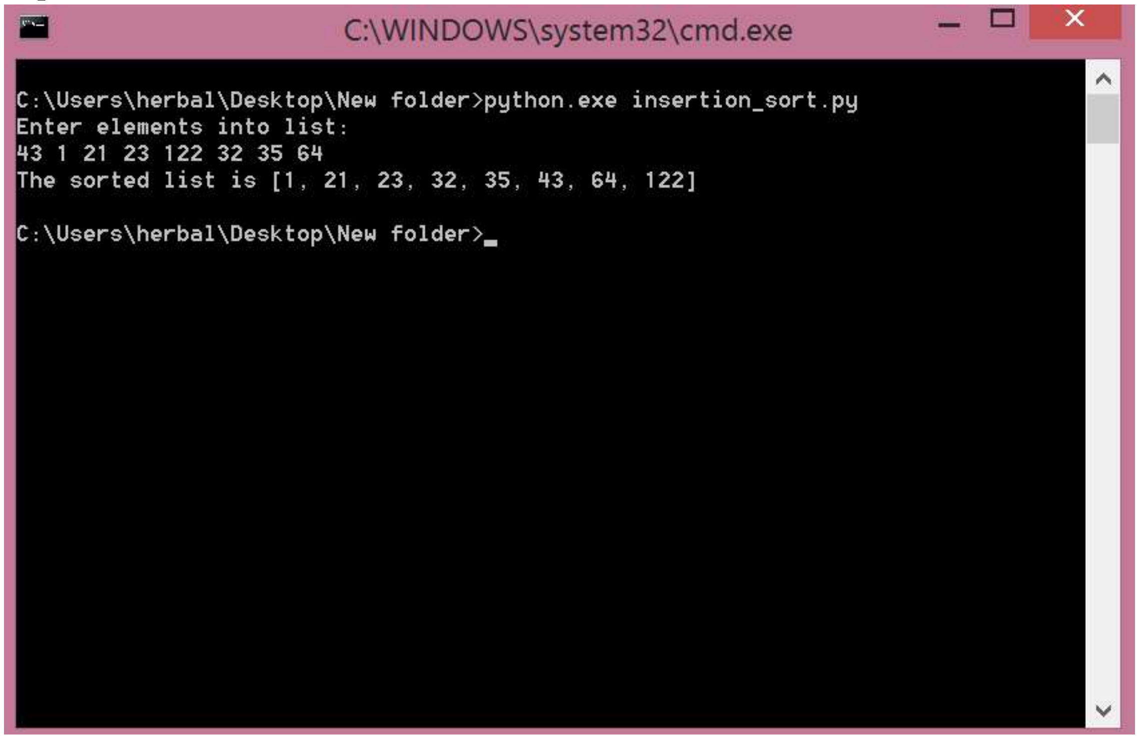


```
C:\Users\herbal\Desktop\New folder>python.exe bubble_sort.py
Enter elements into list:
1 4 2 3 5 7 2 11 23 45 231
The sorted list is [1, 2, 2, 3, 4, 5, 7, 11, 23, 45, 231]

C:\Users\herbal\Desktop\New folder>
```

Program for implementing Insertion Sort

```
def i_sort(a):
    for i in range(1,len(a)):
        temp=a[i]
        pos=i
        while pos>0 and a[pos-1]>temp:
            a[pos]=a[pos-1]
            pos-=1
        a[pos]=temp

print("Enter elements into list:")
a=[int(x) for x in input().split()]
i_sort(a)
print("The sorted list is",a)
```

Output:

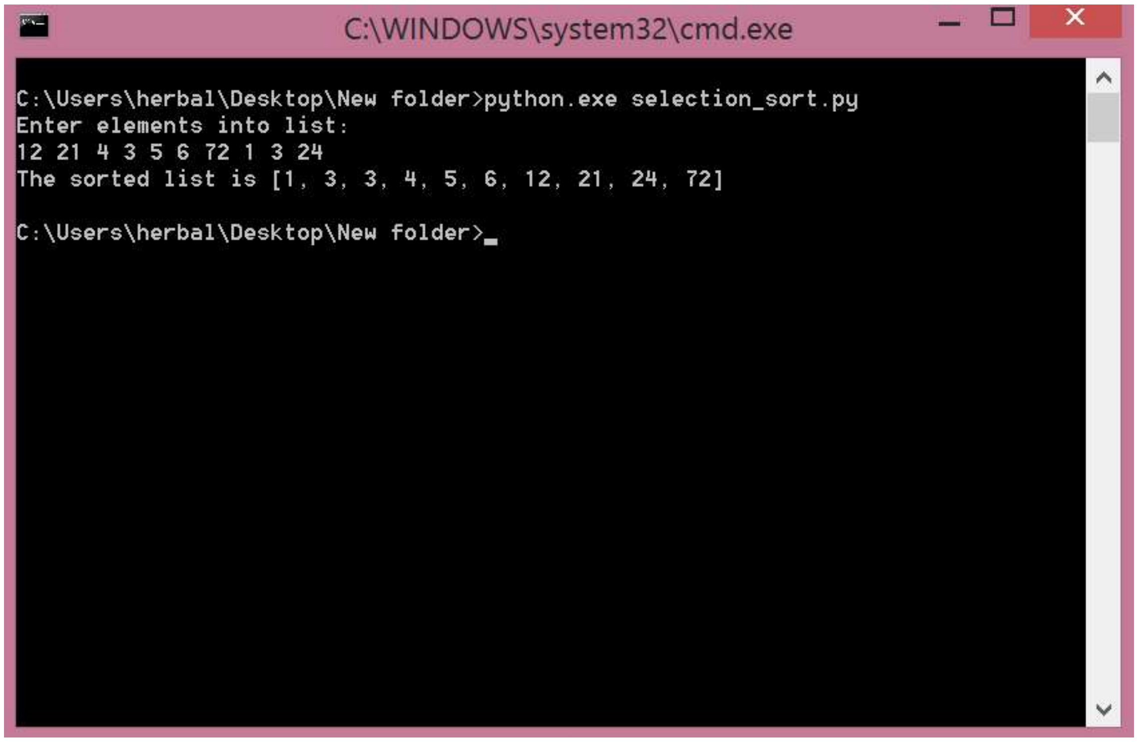


```
C:\Users\herbal\Desktop\New folder>python.exe insertion_sort.py
Enter elements into list:
43 1 21 23 122 32 35 64
The sorted list is [1, 21, 23, 32, 35, 43, 64, 122]

C:\Users\herbal\Desktop\New folder>_
```

Program for implementing Selection Sort

```
def s_sort(a):
    for i in range(len(a)):
        least=i
        for k in range(i+1,len(a)):
            if a[k]<a[least]:
                least=k
        swap(a,least,i)
def swap(a,least,i):
    temp=a[least]
    a[least]=a[i]
    a[i]=temp

print("Enter elements into list:")
a=[int(x) for x in input().split()]
s_sort(a)
print("The sorted list is",a)
```

Output:



```
C:\Users\herbal\Desktop\New folder>python.exe selection_sort.py
Enter elements into list:
12 21 4 3 5 6 72 1 3 24
The sorted list is [1, 3, 3, 4, 5, 6, 12, 21, 24, 72]

C:\Users\herbal\Desktop\New folder>_
```

2.6 PRE LAB VIVA QUESTIONS:

1. Define Sorting?
2. Differentiate between internal sorting and external sorting?
3. Explain the basic idea of Bubble Sort?
4. Explain the concept and application of Insertion Sort?

2.7 LAB ASSIGNMENT:

1. Formulate a program that implement Bubble sort, to sort a given list of integers in descending order.
2. Compose a program that implement Insertion sort, to sort a given list of integers in descending order.
3. Write a program that implement Selection sort, to sort a given list of integers in ascending order.
4. Formulate a program to sort N names using selection sort.
5. Write a program to sort N employee records based on their salary using insertion sort.
6. A class contains 50 students who acquired marks in 10 subjects write a program to display top 10 students roll numbers and marks in sorted order by using bubble sorting technique.

2.8 POST LAB VIVA QUESTIONS:

1. Write the time complexity of Bubble Sort?
2. Write the time complexity of Insertion Sort?
3. Write the other name of Bubble Sort?
4. Write the time complexity of Selection Sort?
5. Write the procedure used to sort the elements using Selection Sort?