

### Experiment 3

Date \_\_\_\_\_  
Page \_\_\_\_\_

Quest:-

Bubble Sort

begin BubbleSort(list)

for all elements of list

if  $\text{list}[i] > \text{list}[i+1]$

swap( $\text{list}[i]$ ,  $\text{list}[i+1]$ )

end if

end for

return list

end Bubble Sort

#### Pseudocode

procedure bubbleSort(list : array of items)

loop = list.count;

for  $i = 0$  to loop - 1 do :

swapped = false

for  $j = 0$  to loop - 1 do :

if  $\text{list}[j] > \text{list}[j+1]$  then

swap( $\text{list}[j]$ ,  $\text{list}[j+1]$ )

swapped = true

end if

end for

if (not swapped) then

break

end if

end for

end procedure return list



## Recursive Version

```
bubbleSort (int arr [], int n)
{
    if n == 1
        return;
    for all elements of array
        if arr[i] > arr[i+1]
        {
            temp = arr[i]
            arr[i] = arr[i+1]
            arr[i+1] = temp
        }
    bubbleSort (arr, n-1);
}
```

## Ques 2:- Shell Sort

procedure shellSort()

A : array of items

while interval < A.length/3 do :

interval = interval \* 3 + 1

end while

while interval > 0 do

for outer = interval; outer < A.length; outer++ do:

valueToInsert = A[outer]

inner = outer;



```

while inner > interval - 1 && A[inner - interval] >= valueToInsert do :
    A[inner] = A[inner - interval]
    inner = inner - interval
end while
A[inner] = valueToInsert
end for
interval = (interval - 1) / 3;
end while
end procedure.

```