ASSIGNMENT NO:2 **DATE:06/02/2016**

PROGRAM TITLE:Sort an array using Bubble Sort.

THEORY:

The basic idea of bubble sort is to compare two adjoining values and exchange them if they are not in proper order. This process is repeated for every pair of adjoining elements in the array. In every pass, the heaviest element settles as its appropriate position in the bottom.

PROGRAM CODE:

```
#Shell Program to perform Bubble Sort
read -p "Enter the number of elements::" n
i=0
echo Enter the elements
while [ $i -lt $n ]
do
      read a[$i]
      i=`expr $i + 1`
done
flag=1
m=`expr $n - 1`
while [ $flag -eq 1 ]
do
      flag=0
      i=0
      while [ $i -lt $m ]
      do
            j=`expr $i + 1`
            if [ ${a[$i]} -gt ${a[$j]} ]
            then
                  temp=${a[$i]}
                  a[$i]=${a[$j]}
                  a[$j]=$temp
                  flag=1
            fi
            i=`expr $i + 1`
      done
done
echo The sorted array is::
i=0
while [ $i -lt $n ]
do
      echo ${a[$i]}
      i=`expr $i + 1`
done
```

OUTPUT:

```
Enter the number of elements::5
Enter the elements
100
25
35
1
```

```
69
The sorted array is::
1
25
35
69
100
```

DISCUSSION:

1. The Complexity of Bubble Sort is $O(n^2)$.
2. Using flag stops the program when no swapping has occurred due to a pass.