EXPERIMENT-5

Program:

WAP to implement Bubble Sort using c/c++ and write the complexity.

Pseudo Code:

```
function BUBBLESORT(ARRAY)
   # loop through the array multiple times
   loop INDEX from 0 to size of ARRAY – 1
      # consider every pair of elements except the sorted ones
      loop INDEX2 from 0 to size of ARRAY – 2 – INDEX
         if ARRAY[INDEX2] > ARRAY[INDEX2 + 1] then
            # swap elements if they are out of order
            TEMP = ARRAY[INDEX2]
            ARRAY[INDEX2] = ARRAY[INDEX2 + 1]
            ARRAY[INDEX2 + 1] = TEMP
         end if
      end loop
   end loop
end function
```

Input:

```
#include <stdio.h>
#include <stdlib.h>
#define N 100
void swap(int *a, int *b)
{
   int c;
   c=*a;
   *a=*b;
   *b=c;
}
void BubbleSort(int a[],int n)
{
```

```
    int i,j;
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n;j++)
        {
            if (a[j]<a[j+1])
            swap(&a[j],&a[j+1]);
        }
    }
}
void printArray(int a[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }
}
int main()
{
printf("Boddu Asmitha Bhavya_A2305221386");
    int a[N],i,n;
    printf("\nThe Number of elements in the array:");
    scanf("%d",&n);
    printf("\nEnter the elements in the array:");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    BubbleSort(a,n);
    printf("\nThe sorted array is:");
```

```
    printArray(a,n);

    return 0;

}
```

Output:

```
Boddu Asmitha Bhavya_A2305221386
The Number of elements in the array:5

Enter the elements in the array:3 7 3 5 1

The sorted array is:7   5       3       3       1
```

Time Complexity: O(N2)

EXPERIMENT-6

Program:

WAP to implement the Selection Sort using c/c++ and write the complexity.

Pseudo Code:

```
procedure selection sort
  list  : array of items
  n     : size of list
  for i = 1 to n - 1
  /* set current element as minimum*/
     min = i
   /* check the element to be minimum */
     for j = i+1 to n
        if list[j] < list[min] then
           min = j;
        end if
     end for
     /* swap the minimum element with the current element*/
     if indexMin != i  then
        swap list[min] and list[i]
     end if
  end for
end procedure
```

Input:

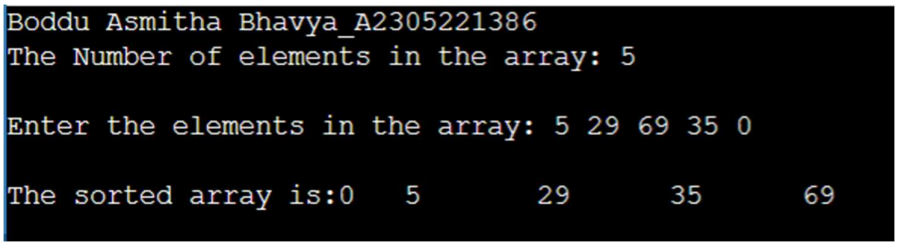
```
#include <stdio.h>
#include <stdlib.h>
#define N 100
void swap(int*a,int *b)
{
   int c;
   c=*a;
   *a=*b;
```

```
    *b=c;
}
void SelectionSort(int a[], int n)
{
    int i, j, min;
    for(i=0;i<n-1;i++)
    {
        min=i;
        for(j=i+1;j<n;j++)
        {
            if (a[j]<a[min])
            min=j;
        }
        swap(&a[min],&a[i]);
    }
}
void printArray(int a[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }
}
int main()
{
    printf("Boddu Asmitha Bhavya_A2305221386");
    int a[N],i,n;
    printf("\nThe Number of elements in the array:");
    scanf("%d",&n);
    printf("\nEnter the elements in the array:");
```

```
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    SelectionSort(a,n);
    printf("\nThe sorted array is:");
    printArray(a,n);
    return 0;

}
```

Output:



Time Complexity:

The time complexity of Selection Sort is O(N2) as there are two nested loops:

One loop to select an element of Array one by one = O(N)

Another loop to compare that element with every other Array element = O(N)

Therefore, overall complexity = O(N) * O(N) = O(N*N) = O(N2)