ACADEMY OF TECHNOLOGY

NAME : PRIYADARSHAN GHOSH

COLLEGE ROLL NO: 72

UNIVERSITY ROLL NO: 16900319072

DEPARTMENT: ECE-1(Y)

SEMESTER: 3rd

PAPER CODE : ES-CS391

➢ Laboratory *Assignment #9*

A.WRITE A PROGRAM TO SORT A LIST OF ELEMENTS USING INSERTION SORT ALGORITHM.

Ans:

```
#include <stdio.h>
void insertionSort(int arr[], int n)
{
	int i, key, j;
	for (i = 1; i < n; i++)
{
		key = arr[i];
		j = i - 1;
		while (j >= 0 && arr[j] > key)
{
			arr[j + 1] = arr[j];
			j = j - 1;
}
		arr[j + 1] = key;
}
}
void printArray(int arr[], int n)
{
	int i;
	for (i = 0; i < n; i++)
		printf("%d ", arr[i]);
	printf("\n");
}
int main()
```

```
{
        int a[50],n,i;
        printf("\n Enter how many numbers<=50");
        scanf("%d",&n);
        printf("\n Enter numbers one by one:");
        for(i=0;i<n;i++)
                scanf("%d",&a[i]);
                insertionSort(a, n);
                printf("Sorted array: \n");
                printArray(a, n);
}
```

OUTPUT =>

```
 Enter how many numbers<=50
10

 Enter numbers one by one:
-99
-20
10
0
5
9
7
3
1
4
```

```
Sorted array:
-99 -20 0 1 3 4 5 7 9 10


--------------------------------
Process exited after 26.99 seconds with return value 0
Press any key to continue . . .
```

B.I) WRITE PROGRAMS TO SORT A LIST OF ELEMENTS USING QUICK SORT ALGORITHM CONSIDERING FIRST ELEMENT AS A PIVOT.

Ans:

```
#include<stdio.h>
void quicksort(int number[50],int first,int last){
   int i, j, pivot, temp;

   if(first<last){
      pivot=first;
      i=first;
      j=last;

      while(i<j){
         while(number[i]<=number[pivot]&&i<last)
            i++;
         while(number[j]>number[pivot])
            j--;
         if(i<j){
            temp=number[i];
```

```
            number[i]=number[j];
            number[j]=temp;
          }
        }

        temp=number[pivot];
        number[pivot]=number[j];
        number[j]=temp;
        quicksort(number,first,j-1);
        quicksort(number,j+1,last);

    }
}

int main(){
    int i, count, number[50];

    printf("Enter how many numbers<=50: ");
    scanf("%d",&count);

    printf("Enter %d elements: ", count);
    for(i=0;i<count;i++)
        scanf("%d",&number[i]);

    quicksort(number,0,count-1);

    printf("Sorted array: ");
```

```
   for(i=0;i<count;i++)
      printf(" %d",number[i]);

   return 0;
}
```

OUTPUT =>

```
Enter how many numbers<=50:
10
Enter 10 elements:
5
9
2
8
1
0
-10
65
32
12
Sorted array:  -10 0 1 2 5 8 9 12 32 65
--------------------------------
Process exited after 26.95 seconds with return value 0
Press any key to continue . . .
```

B.II) Write programs to sort a list of elements using quick sort algorithm considering mid element as a pivot.

Ans:

```
#include<stdio.h>
int partition(int arr[], int left, int right)
{
int i = left, j = right;
int tmp;
int pivot = arr[(left + right) / 2];
while (i <= j) {
while (arr[i] < pivot)
i++;
while (arr[j] > pivot)
j--;
if (i <= j) {
tmp = arr[i];
arr[i] = arr[j];
arr[j] = tmp;
i++;
j--;
}
}
return i;
}
void quickSort(int arr[], int left, int right) {
int index = partition(arr, left, right);
```

```
if (left < index - 1)
quickSort(arr, left, index - 1);
if (index < right)
quickSort(arr, index, right);
}
int main(){
   int i, count, arr[50];

   printf("Enter how many numbers<=50: ");
   scanf("%d",&count);

   printf("Enter %d elements: ", count);
   for(i=0;i<count;i++)
      scanf("%d",&arr[i]);
   printf("\nData before sorting:");
    for (i = 0; i < count; i++)
{
        printf(" %d", arr[i]);
    }

   quickSort(arr,0,count-1);

   printf("\nData after sorting: ");
   for(i=0;i<count;i++)
      printf(" %d",arr[i]);

   return 0;
```

```
}
```

OUTPUT =>

```
Enter how many numbers<=50:
10
Enter 10 elements:
45
12
87
1
2
-98
54
0
5
36

Data before sorting: 45 12 87 1 2 -98 54 0 5 36
Data after sorting:  -98 0 1 2 5 12 36 45 54 87
--------------------------------
Process exited after 21.48 seconds with return value 0
Press any key to continue . . .
```