Remarks _____ Returned on _____

Procedure INSERTION_SORT (A,n,ele)

Description:- This procedure inserts an element from unsorted array into the sorted array A[J] and 'n' is the number of elements present in the array list.

Declaration:- Global integer A(1:n), n, ele
local integer i,j

Algorithm:-

```
For i=2 to n by +1 do
    ele = A(i)
    For j=i-1 to 1 by -1 do
        if ele < A(j)
            A(j+1) = A(j)
        else
            exit_loop
        endif
        A(j+1) = ele
    repeat
repeat
end_INSERTION_SORT
```

(1)

Listing

Roll No. ____ Expt. No. 7.3 Submitted on ____

Remarks ____ Returned on ____

Procedure BUBBLE_SORT (A, n)

Description:- This algorithm sorts elements in an array A[], 'n' is the number of elements present in an array list.

Declaration:-

Global integer A(1:n), n

local integer i, j, temp.

Algorithm:-

```
for i=1 to n-1 do
  for j=1 to n-i-1 do
    if (A(j) > A(j+1)), then
      exchange (A(j), A(j+1))
    endif
  repeat
repeat
```

END_Bubble_Sort

Remarks _____ Returned on _____

Procedure Selection_Sort (A, n)

Description:- This procedure selects a minimum element from an array A[]. 'n' is the number of elements present in an array list.

Declaration:- Global integer A(1:n), n
local integer i, j, min

Algorithm:-

```
For i = 1 to n by +1 do
    min = A(i)
    Pos_min = i
    For j = i+1 to n by 1 do
        if A(j) < min, then
            min = A(j)
            Pos_min = j
        endif
    repeat
    exchange (A(i), A(Pos_min))
repeat

End_Selection_SORT
```


Listing

Remarks _____ Returned on _____

Quick sort

Procedure QUICK_SORT (P, Q)

Description:- This procedure sorts the elements A(P)... upto A(Q) which reside in the global array A(n) into the ascending order; A(n+1) is consider to be define and must be greater than or equals to all the elements in A(P:Q); A(n+1) = + ∞

Declaration:- integer P, Q
Global n, A(n)

```
if P<Q, then
    j<-Q+1
    PARTITION (P, j)
    call QUICK_SORT (P, j-1)
    call QUICK_SORT (j+1, Q)
endif
END_QUICK_SORT
```

4

Procedure PARTITION (m,p)

Description:- Within A(m), A(m+1) .... A(p-1) the elements are rearrange in such a way that if initially temp = A(m) then after coplétion A(q) = temp for some q between m and (p-1), A(k) ≤ temp for (m ≤ k ≤ q-1) and A(k) > temp for (q ≥ k ≥ p-1) the final value of p is q

Declaration:- Global A(m:p), integer m,p,i

Algorithm:- temp ← A(m) ; i ← m

```
loop
    do
        i ← i+1
    while A(i) < temp repeat
    do
        p ← p-1
    while A(p) > temp repeat
    if i ≤ p , then
        exchange (A(i), A(p))
    else
        exit loop
    endif
repeat
A(m) ← A(p)
A(p) ← temp
```

End_PARTITION