Name :- Suryakant Upadhyay
Subject :- Data Structure.
PRN :- 20220802043
Batch :- A1

Theory Assignment 01

Q1) Sort the given array element using Bubble Sort :-
-2, 45, 0, 11, -9.

⇒

[-2, 45, 0, 11, -9]
[(-2), (45), 0, 11, -9]
[-2, (45), (0), 11, -9]
[-2, 0, (45), (11), -9]
~~[(-2), 0, 11~~
[-2, 0, 11, (45), (-9)]
[-2, 0, 11, -9, 45] → first iteration.
[(-2), 0, 11, -9, 45].
[-2, 0, (11), (-9), 45]
[-2, 0, -9, (11), (45)]
[-2, 0, -9, 11, 45] → Second iteration.
[(-2), (0), -9, 11, 45]
[-2, (0), (-9), 11, 45].
[-2, -9, (0), (11), 45].
[-2, -9, 0, (11), (45)]
[-2, -9, 0, 11, 45]. → Third iteration.
[(-2), (-9), 0, 11, 45]
[-9, -2, 0, 11, 45] → fourth iteration.

Code:-

```
def bubblesort(arr):
    n = len(arr)

for i in range(n-1):

for j in range(0,n-i-1):
    if arr[j] > arr[j+1]:
        temp = arr[j]
        arr[j] = arr[j+1]
        arr[j+1] = temp

arr = [-2, 45, 0, 11, -9]
print("The given array is:", arr)
bubblesort(arr)
print("Sorted array is:")
for i in range(len(arr)):
    print("%d" % arr[i], end=" ")
```

Q2) Sort the given array element using Selection Sort : 20, 12, 10, 15, 2.

⇒ length of array.

length = len(array) → 5.

[20, 12, 10, 15, 2] → given array

[(20), 12, 10, 15, (2)]

[[2], (12), 10, 15, 20]

[2, 10, 12, 15, 20] → Sorted array.

Code :-

```
def SelectionSort (arr):
    n = len (arr).
for i in range (n-1)
        Small Ndx = i
for j in range (i+1, n):
            if arr [j] < arr [Small Ndx]:
Small Ndx = j.

if Small Ndx = j
        temp = arr [i]
        arr[i] = arr [Small Ndx]
        arr [Small Ndx] = temp

arr = [20, 12, 10, 15, 2]
Selection Sort (arr)
  print (*arr)
```

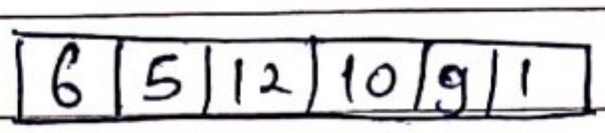
Q3.) Sort the given array element using insertion So
9, 5, 1, 4, 3.

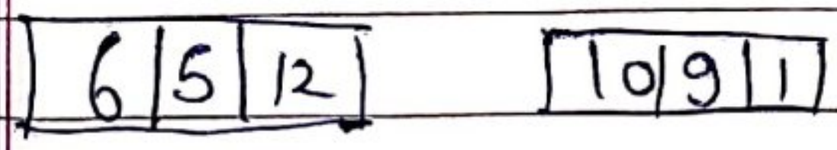
[9, 5, 1, 4, 3]
[5, 9, 1, 4, 3]
[5, 1, 9, 4, 3]
[5, 1, 4, 3, 9]
[1, 5, 4, 3, 9]
[1, 4, 3, 5, 9]
[1, 3, 4, 5, 9]

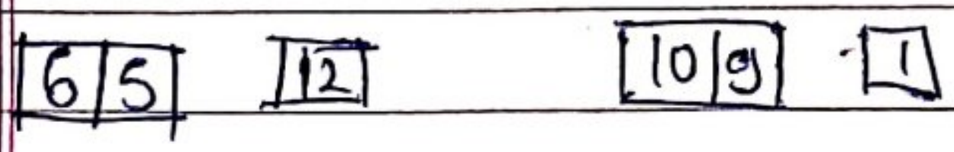
```
def Insertion Sort (arr):
        n= len (arr)
    for i in range (1,n):
        Value = arr[i].
        pos = i
    while pos >0 & Value < arr[pos-1]
     arr[pos] = arr[pos-1]
     arr[Pos] < arr[Pos-1]
      POS = 1
     arr[pos] = value

arr = [9,5,1,4,3]
InsertionSort (arr)
Print (arr)
```

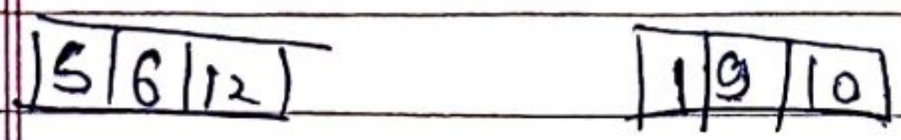
Q4Y Sort the given array element using merge sort.

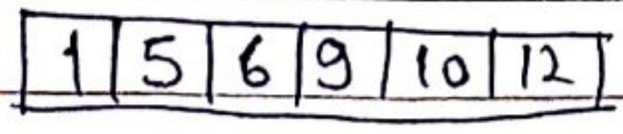
6, 5, 12, 10, 9, 1





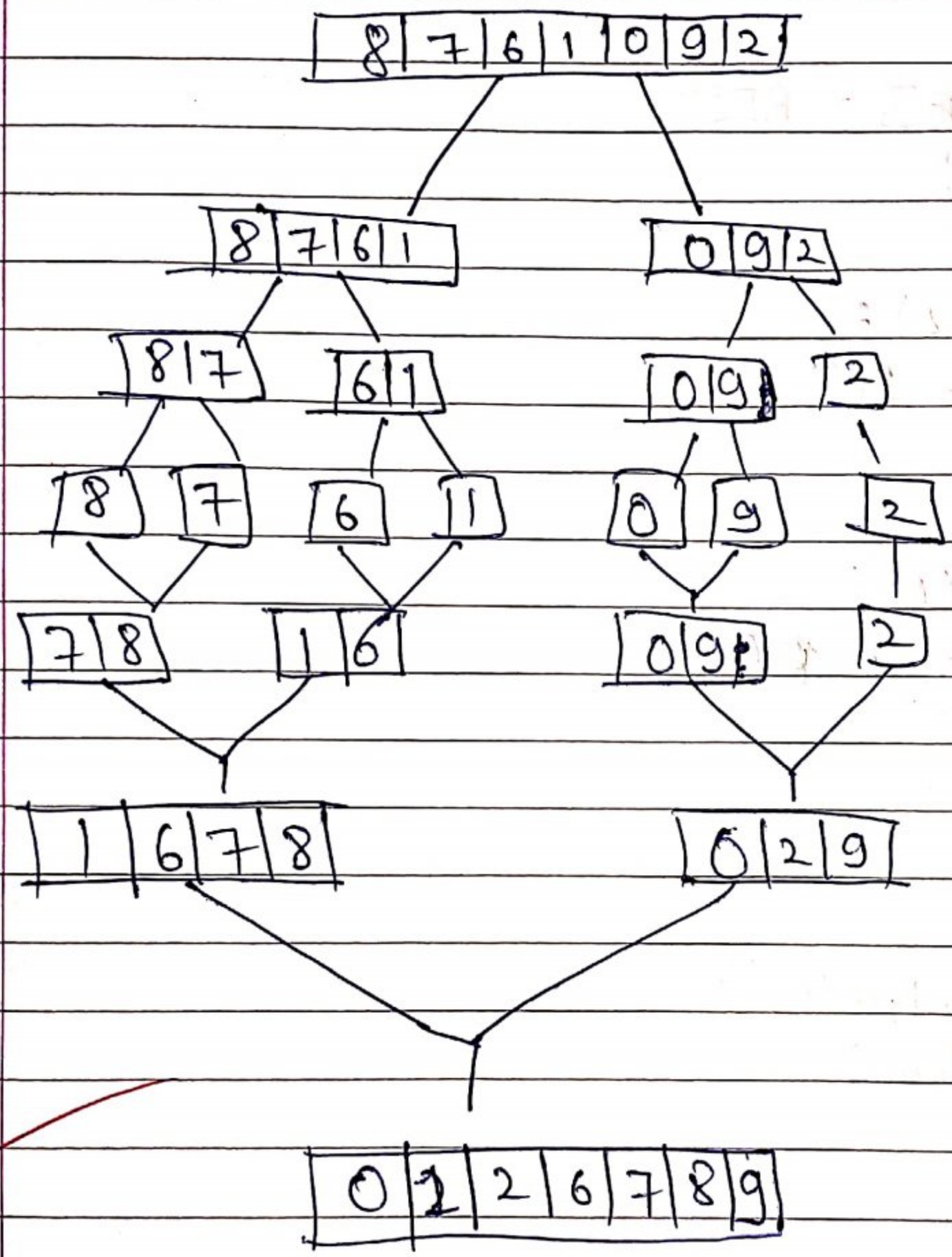






```
def Merge Sort (arr):
    if len(arr) > 1:
        mid = len(arr)//2
        L = arr[:mid]
        R = arr[mid:]
        merge sort (L)
        merge sort (R)
        i = j = k = 0
    while i < len(L) & j < len(R).
            if L[i] <= R[j]
            arr[k] = L[i]
            i += 1
    else:
            arr[k] = R[j]
            j += 1
        k += 1
    while i < len(L):
        arr[k] = L[i]
        i += 1
        k += 1
    while j < len(R):
        arr[k] = R[j]
        j += 1
        k += 1

arr1 = [6, 5, 12, 10, 9, 1]
mergesort (arr1)
print (*arr1).
```

Q5) Sort the given array element using merge sort:



Hw

```
def merge Sort (arr) :
    if len (arr) > 1 :
        mid = len(arr) // 2
        L = arr [ : mid ]
        R = arr [ mid : ]
        merge_Sort ( L )
        merge sort ( R )
        i = j = k = 0
    while i < len ( L ) & j < len ( R )
        if L [ i ] <= R [ j ]
```

```
        arr[k] = L[i]
        i += 1
    else:
        arr[k] = R[j]
        j += 1
        k += 1
while i < len(L):
    arr[k] = L[i]
    i += 1
    k += 1
while j < len(R):
    arr[k] = R[i]
    j += 1
    k += 1
~~arr = [8.~~
```

```
arr = [8, 7, 6, 1, 0, 9, 2]
mergeSort(arr)
print(*arr).
```

17/3/2023