Assignment -1

# Write Pseudocode and create a flowchart for a bubble sort algorithm. Provide a Brief explanation of how the algorithm works and a simple array of integers To demonstrate a dry run of your algorithm.

BUBBLE SORT:

ALGORITHM

1.START

2.SET i=1

3.SET j=0

4.IF (a[j]>a[j+1])

Swap a[j] and a[j+1]

5.J=j+1

6.IF(j<(n-i)) go to step 2

7.i=i+1

8.IF(i<n) go to step 3

9.STOP

PSEUDOCODE

1. Start
2. Read n, the array size
3. Read the elements of array A[]
4. I value varies from 1 to n-1. For each value of I do steps 5 to 8.
5. J value varies from 0 to n-i-1. For each value of j do steps 6 & 7.
6. Check j th element is greater than (j+1)th element. If yes, swap them.
7. J-j+1
8. I=i+1
9. Print the sorted array A[]
10. Stop

CODE:

Read n, the array size

For i=1 to n-1

{

For j=0 to n-i-1

{

If A[j]>A[j+1] then

{

Temp=A[j]

A[j]=A[j+1]

A[j+1]=temp

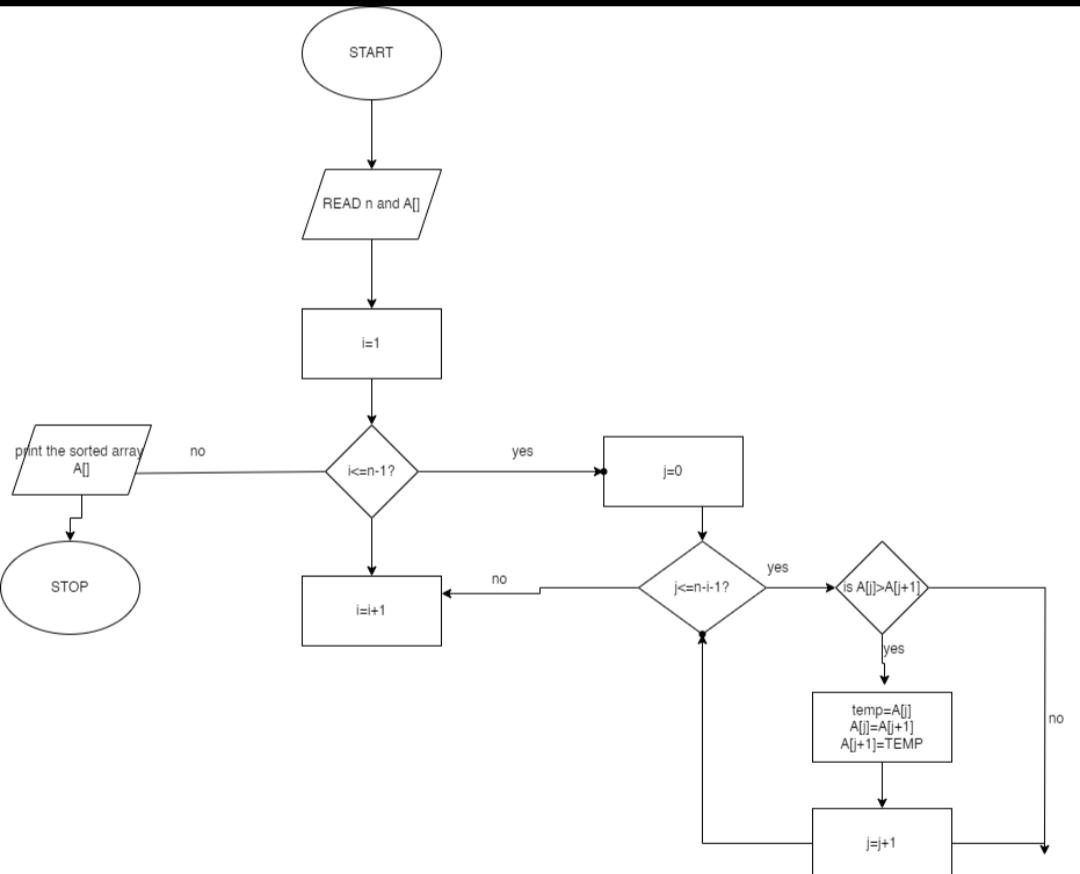
}

}

}

Print the sorted array A[]

FLOWCHART



Consider the unsorted array a = [64, 34, 25, 12, 22, 11, 90].

Pass 1 (I = 1):

Inner loop:

J = 0: Compare 64 and 34 (swap needed, a becomes [34, 64, 25, 12, 22, 11, 90])

J = 1: Compare 64 and 25 (swap needed, a becomes [34, 25, 64, 12, 22, 11, 90])

J = 2: Compare 64 and 12 (swap needed, a becomes [34, 25, 12, 64, 22, 11, 90])

J = 3: No swap needed (remaining elements are already sorted in this pass)

Outer loop continues (I becomes 2)

Pass 2 (I = 2):

Inner loop:

J = 0: Compare 34 and 25 (no swap needed)

J = 1: Compare 25 and 12 (swap needed, a becomes [34, 12, 25, 64, 22, 11, 90])

J = 2: Compare 64 and 22 (swap needed, a becomes [34, 12, 22, 64, 25, 11, 90])

J = 3: No swap needed (remaining elements are already sorted in this pass)

Outer loop continues (I becomes 3)

Pass 3 (I = 3):

Inner loop:

J = 0: Compare 34 and 12 (no swap needed)

J = 1: Compare 12 and 22 (no swap needed)

J = 2: All comparisons involve elements already in the correct order (no swaps needed)

The loop may terminate early due to no swaps in the inner loop (optimization).

Result:

After the third pass (or potentially earlier with optimization), the sorted array becomes `a = [11, 12, 22, 25,34,64,90]