

1.1

A) ArrayA = {3, 7, 1, 6} e N = 4

BubbleSort

def bubble(vt):

 for i in range(0, 4):

 for j in range(i+1, 4):

 if(vt [i] > vt [j]):

 aux = vt [i]
 vt [i] = vt [j]
 vt [j] = aux

1ª Iteração: i = 0

j = 1

vt [i] = vt [0] = 3

vt [j] = vt [1] = 7

Condição if: vt [0] > vt [1] → FALSO → Não troca os valores!!

j = 2

vt [i] = vt [0] = 3

vt [j] = vt [2] = 1

Condição if: vt [0] > vt [2] → VERDADEIRO → Troca os valores!!

aux = vt [0]
vt [0] = vt [2]
vt [2] = aux

ArrayA = {1, 7, 3, 6}

j = 3

vt [i] = vt [0] = 1

vt [j] = vt [3] = 6

Condição If: vt [0] > vt [3] → FALSO → Não troca os valores!!

2ª Iteração: i = 1

j = 2

vt [i] = vt [1] = 7

vt [j] = vt [2] = 3

Condição If: vt [1] > vt [2] → VERDADEIRO → Troca os valores!!

aux = vt [1]
vt [1] = vt [2]
vt [2] = aux

ArrayA = {1, 3, 7, 6}

j = 3

vt [i] = vt [1] = 3

vt [j] = vt [3] = 6

Condição If: vt [1] > vt [3] → FALSO → Não troca os valores!!

3ª Iteração: i = 2

j = 3

vt [i] = vt [2] = 7

vt [j] = vt [3] = 6

Condição If: vt [2] > vt [3] → VERDADEIRO → Troca os valores!!

aux = vt [2]

vt [2] = vt [3]

vt [3] = aux

ORDENADO → ArrayA = {1, 3, 6, 7}

SelectionSort

ArrayA = {3, 7, 1, 6}

def selection(vt):

 for i in range(0, 4):

 posMen = i
 valMen = vt[i]

 for j in range(i+1, 4):

 if(valMen > vt[j]):
 posMen = j
 valMen = vt[j]

 aux = vt[i]
 vt[i] = vet[posMen]
 vet[posMen] = aux

1ª Iteração: i = 0

j = 1 ; vt [j] = vt [1] = 7

posMen = 0 ; valMen = vt[0] = 3 ;

Condição If: valMen > vt [1] → FALSO → Não troca os valores!!

j = 2 ; vt [j] = vt [2] = 1

posMen = 0 ; valMen = vt[0] = 3 ;

Condição If: valMen > vt [2] → VERDADEIRO → Troca os valores!!

 posMen = 2
 valMen = 1

j = 3 ; vt [j] = vt [3] = 6

posMen = 2 ; valMen = vt[2] = 1 ;

Condição If: valMen > vt [3] → FALSO → Não troca os valores!!

 aux = vt[0]
 vt[0] = vet[2]
 vet[2] = aux

ArrayA = {1, 7, 3, 6}

2ª Iteração: i = 1

j = 2 ; vt [j] = vt [2] = 3

posMen = 1 ; valMen = vt[1] = 7 ;

Condição If: valMen > vt [2] → VERDADEIRO → Troca os valores!!

 posMen = 2
 valMen = 3

j = 3 ; vt [j] = vt [3] = 6

posMen = 2 ; valMen = vt[2] = 3 ;

Condição If: valMen > vt [3] → FALSO → Não troca os valores!!

aux = vt[1]

vt[1] = vet[2]

vet[2] = aux

ArrayA = {1, 3, 7, 6}

3ª Iteração: i = 2

j = 3 ; vt [j] = vt [3] = 6

posMen = 2 ; valMen = vt[2] = 7 ;

Condição If: valMen > vt [3] → VERDADEIRO → Troca os valores!!

posMen = 3

valMen = 6

aux = vt[2]

vt[2] = vet[3]

vet[3] = aux

ORDENADO → ArrayA = {1, 3, 6, 7}

InsertionSort

ArrayA = {3, 7, 1, 6}

```
def insertion(vt):  
    for i in range(1, 4):  
        aux = vt[i]  
        j = i - 1  
  
        while( ( j >= 0 ) and ( aux < vt[ j ] ) ):  
  
            vt[ j+1 ] = vt[ j ]  
            j -= 1  
  
        vt[j+1] = aux
```

1ª Iteração: i = 1

j = 0 ; aux = vt [1] = 7
Laço while: (0 >= 0) and (aux < vt[0]) → FALSO → Sai do While!
Vt [0+1] = 7

ArrayA = {3, 7, 1, 6}

2ª Iteração: i = 2

j = 1 ; aux = vt [2] = 1
Laço while: (1 >= 0) and (aux < vt[1]) → VERDADEIRO → Entra no While!
Vt[1+1] = 7
j -= 1 logo, j = 0

ArrayA = {3, 7, 7, 6}

Laço while: (0 >= 0) and (aux < vt[0]) → VERDADEIRO → Entra no While!
Vt [0+1] = 3
j -= 1 logo, j = -1

vt [-1 + 1] = vt [0] = 1

ArrayA = {1, 3, 7, 6}

2ª Iteração: i = 3

j = 2 ; aux = vt [3] = 6
Laço while: (2 >= 0) and (aux < vt[2]) → VERDADEIRO → Entra no While!
Vt [2+1] = 7
j -= 1 logo, j = 1

ArrayA = {1, 3, 7, 7}

Laço while: ($i \geq 0$) and ($aux < vt[i]$) \rightarrow FALSO \rightarrow Sai do While!

$Vt[i+1] = 6$

ORDENADO \rightarrow ArrayA = {1, 3, 6, 7}