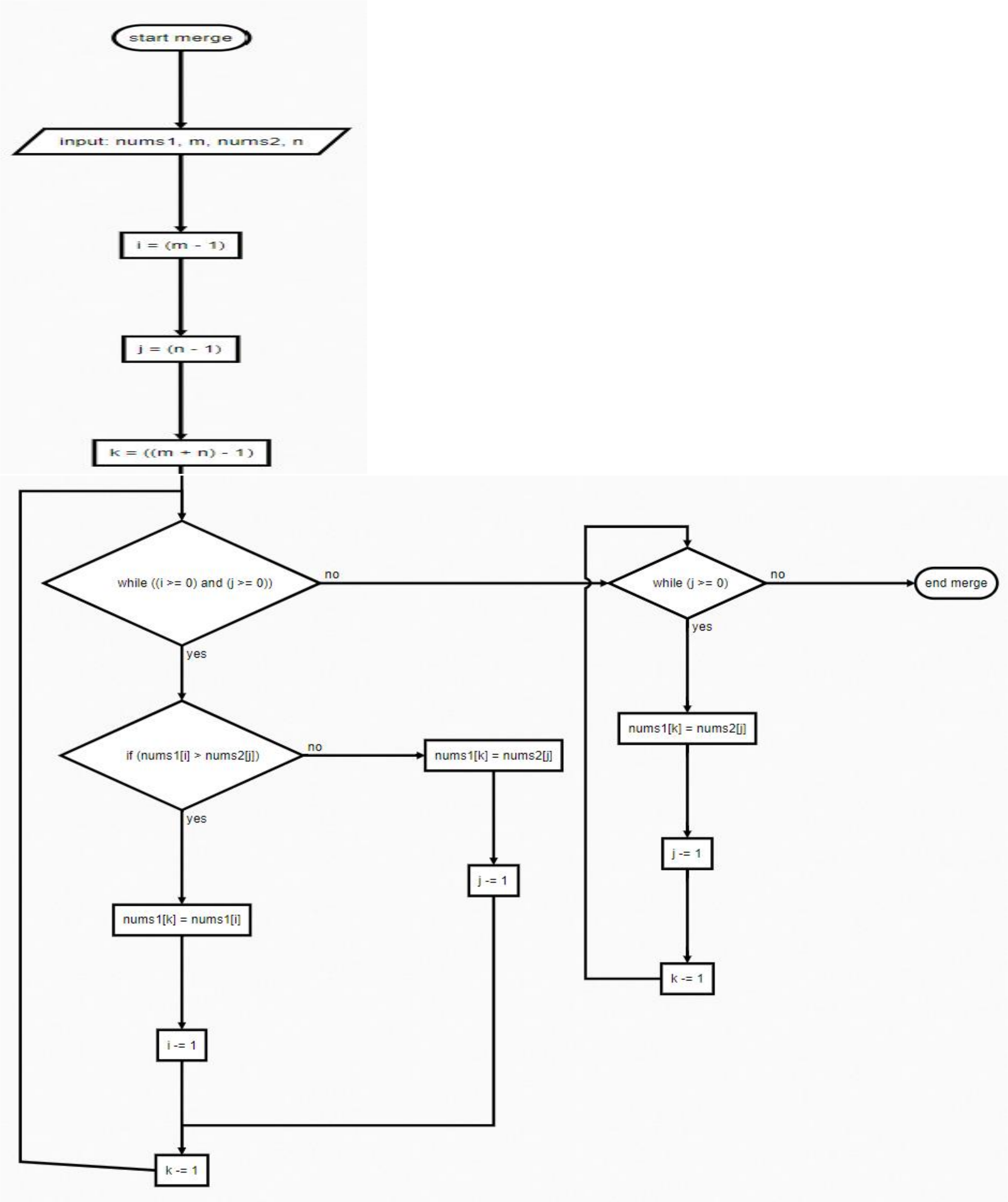
Week4Q1

Python code

```
def merge(nums1, m, nums2, n):
    # Initialize pointers for nums1, nums2, and the merged array
    i = m - 1  # Pointer for nums1
    j = n - 1  # Pointer for nums2
    k = m + n - 1  # Pointer for the merged array

    # Iterate from the end of the arrays
    while i >= 0 and j >= 0:
        if nums1[i] > nums2[j]:
            # If element in nums1 is larger, place it in the merged array
            nums1[k] = nums1[i]
            i -= 1
        else:
            # If element in nums2 is larger, place it in the merged array
            nums1[k] = nums2[j]
            j -= 1
        k -= 1

    # If there are any remaining elements in nums2, append them to nums1
    while j >= 0:
        nums1[k] = nums2[j]
        j -= 1
        k -= 1

nums1 = [1, 2, 0, 0]
m = 2
nums2 = [2, 3]
n = 2
merge(nums1, m, nums2, n)
print(nums1)
```


start merge
input: nums1, m, nums2, n
i = (m - 1)
j = (n - 1)
k = ((m + n) - 1)
while ((i >= 0) and (j >= 0))
no
yes
if (nums1[i] > nums2[j])
no
nums1[k] = nums2[j]
yes
nums1[k] = nums1[i]
j -= 1
i -= 1
k -= 1
while (j >= 0)
no
end merge
yes
nums1[k] = nums2[j]
j -= 1
k -= 1

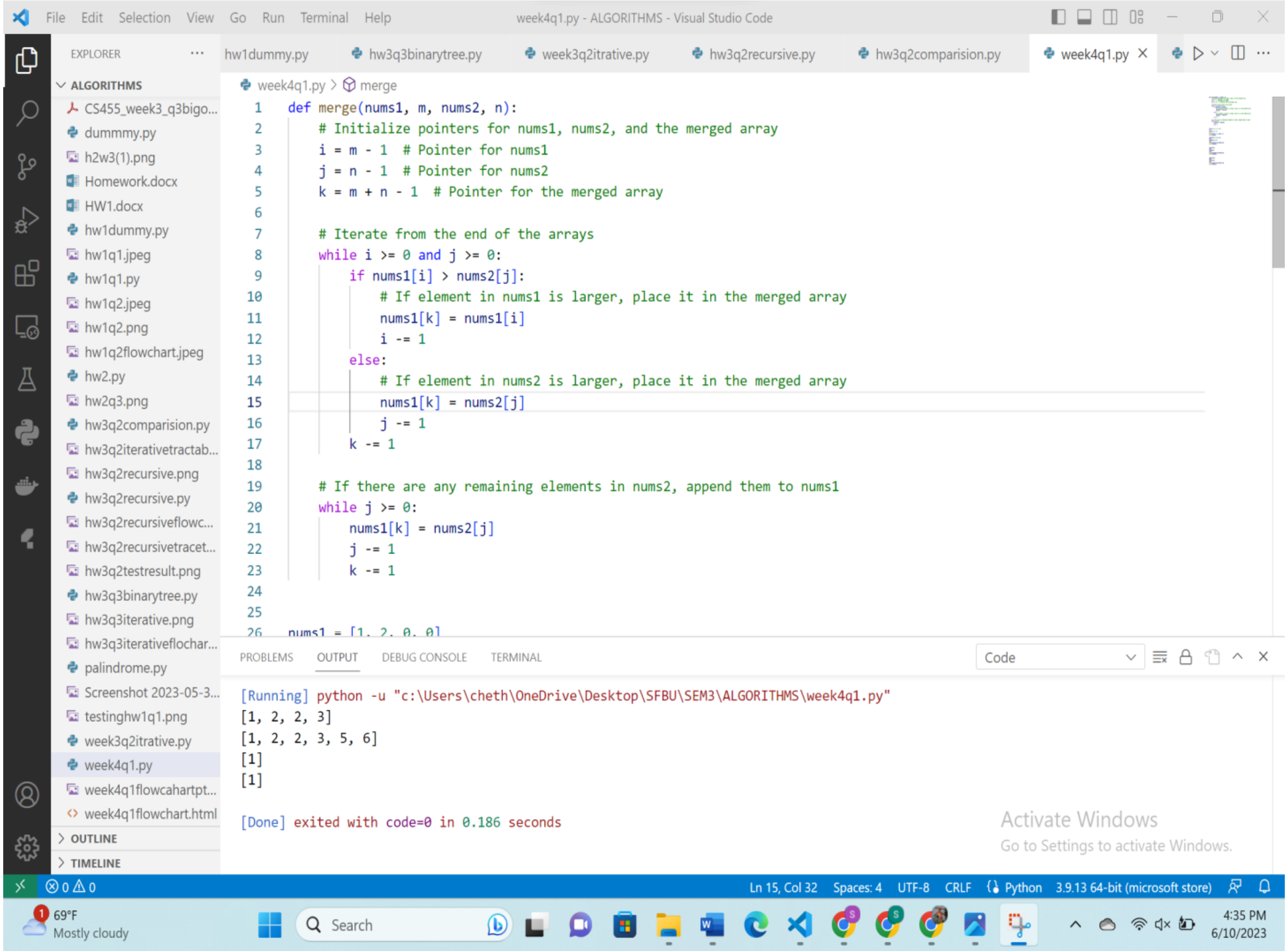
Tracetable

steps	Nums1	m	Nums2	n	i	j	k	i>=0	j>=0	i>=0 andj>=0	Nums1[i]>Nums2[j]	screen
1	[1,2,0,0]	2	[2,3]	2								
2					2-1=1							
3						2-1=1						
4							2+2-1=3					
5										true		
6											false	
7	[1,2,0,3]											
8						1-1=0						
9							2					
5										true		
6											false	
7	[1,2,2,3]											
8						0-1=-1						
9							1					
10									false			

Testcases

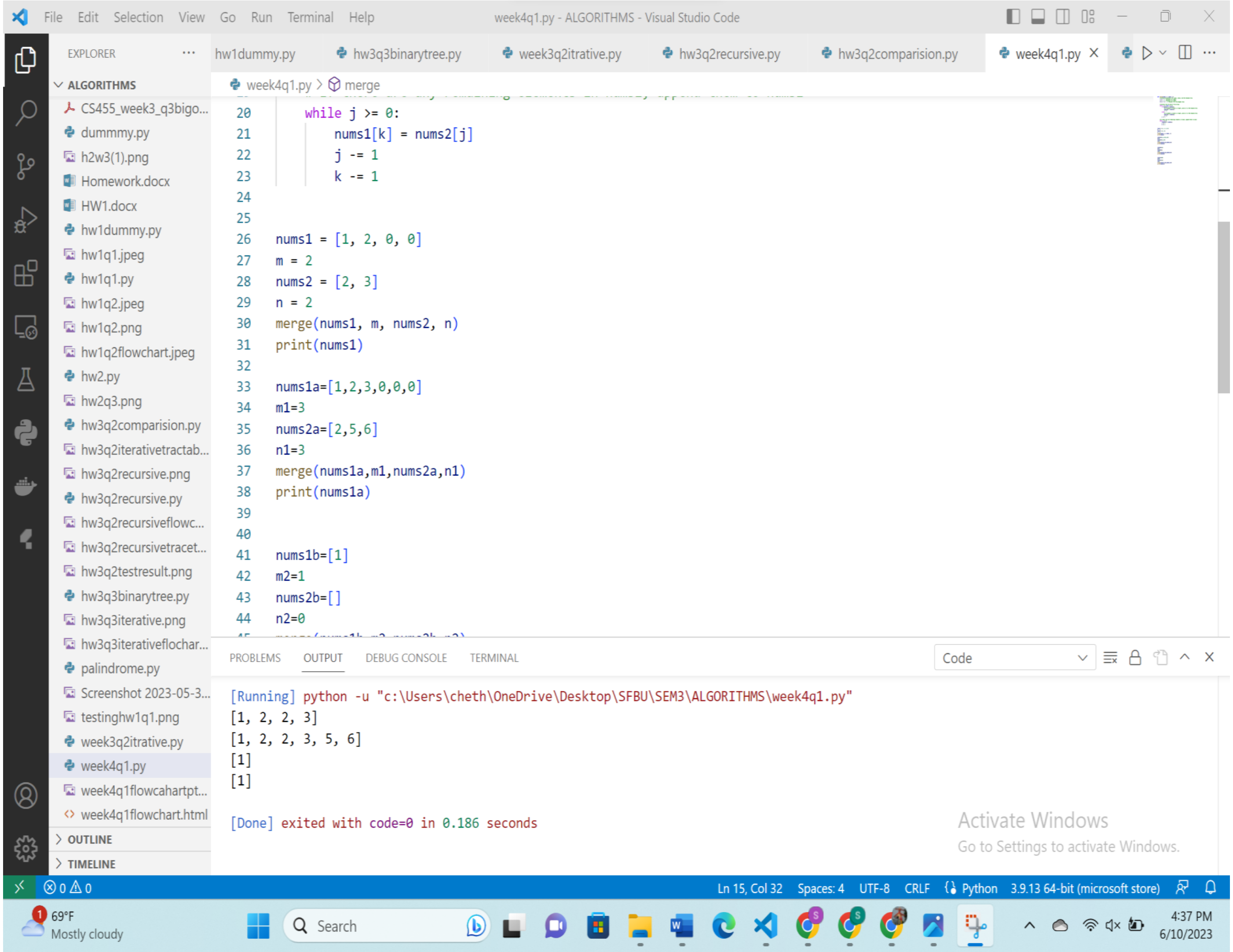
```
def merge(nums1, m, nums2, n):
    # Initialize pointers for nums1, nums2, and the merged array
    i = m - 1  # Pointer for nums1
    j = n - 1  # Pointer for nums2
    k = m + n - 1  # Pointer for the merged array

    # Iterate from the end of the arrays
    while i >= 0 and j >= 0:
        if nums1[i] > nums2[j]:
            # If element in nums1 is larger, place it in the merged array
            nums1[k] = nums1[i]
            i -= 1
        else:
            # If element in nums2 is larger, place it in the merged array
            nums1[k] = nums2[j]
            j -= 1
        k -= 1

    # If there are any remaining elements in nums2, append them to nums1
    while j >= 0:
        nums1[k] = nums2[j]
        j -= 1
        k -= 1


nums1 = [1, 2, 0, 0]
m = 2
nums2 = [2, 3]
n = 2
merge(nums1, m, nums2, n)
print(nums1)

nums1a=[1,2,3,0,0,0]
m1=3
nums2a=[2,5,6]
n1=3
merge(nums1a,m1,nums2a,n1)
print(nums1a)


nums1b=[1]
m2=1
```

```
nums2b=[]
n2=0
merge(nums1b,m2,nums2b,n2)
print(nums1b)


nums1c=[0]
m3=0
nums2c=[1]
n3=1
merge(nums1c,m3,nums2c,n3)
print(nums1c)
```



```
def merge(nums1, m, nums2, n):
    # Initialize pointers for nums1, nums2, and the merged array
    i = m - 1  # Pointer for nums1
    j = n - 1  # Pointer for nums2
    k = m + n - 1  # Pointer for the merged array

    # Iterate from the end of the arrays
    while i >= 0 and j >= 0:
        if nums1[i] > nums2[j]:
            # If element in nums1 is larger, place it in the merged array
            nums1[k] = nums1[i]
            i -= 1
        else:
            # If element in nums2 is larger, place it in the merged array
            nums1[k] = nums2[j]
            j -= 1
        k -= 1

    # If there are any remaining elements in nums2, append them to nums1
    while j >= 0:
        nums1[k] = nums2[j]
        j -= 1
        k -= 1


nums1 = [1, 2, 0, 0]
```

```
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week4q1.py"
[1, 2, 2, 3]
[1, 2, 2, 3, 5, 6]
[1]
[1]

[Done] exited with code=0 in 0.186 seconds
```



```python
    # If there are any remaining elements in nums2, append them to nums1
    while j >= 0:
        nums1[k] = nums2[j]
        j -= 1
        k -= 1


nums1 = [1, 2, 0, 0]
m = 2
nums2 = [2, 3]
n = 2
merge(nums1, m, nums2, n)
print(nums1)

nums1a=[1,2,3,0,0,0]
m1=3
nums2a=[2,5,6]
n1=3
merge(nums1a,m1,nums2a,n1)
print(nums1a)


nums1b=[1]
m2=1
nums2b=[]
n2=0
merge(nums1b,m2,nums2b,n2)
```

```
[Running] python -u "c:\Users\cheth\OneDrive\Desktop\SFBU\SEM3\ALGORITHMS\week4q1.py"
[1, 2, 2, 3]
[1, 2, 2, 3, 5, 6]
[1]
[1]

[Done] exited with code=0 in 0.186 seconds
```