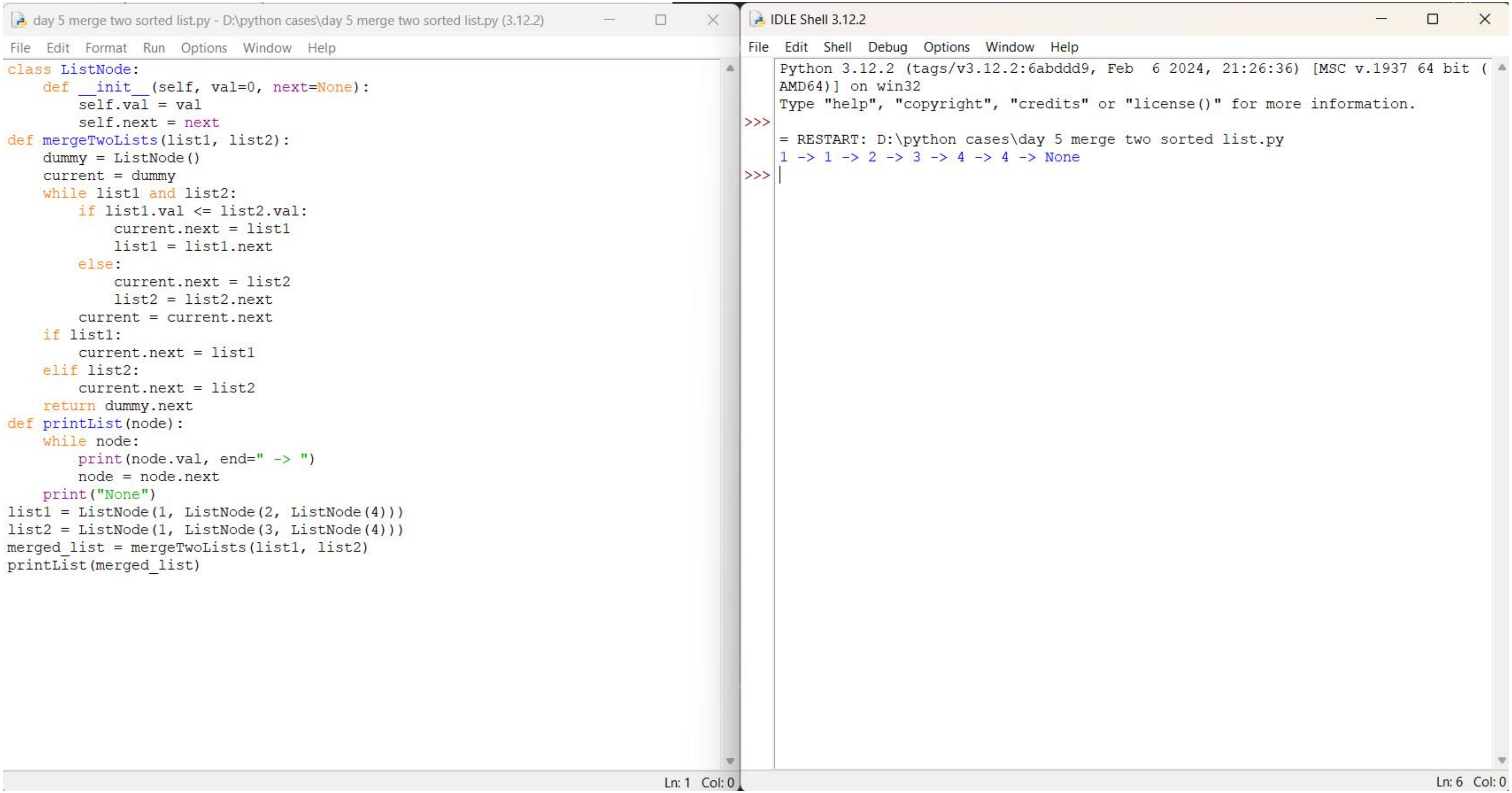
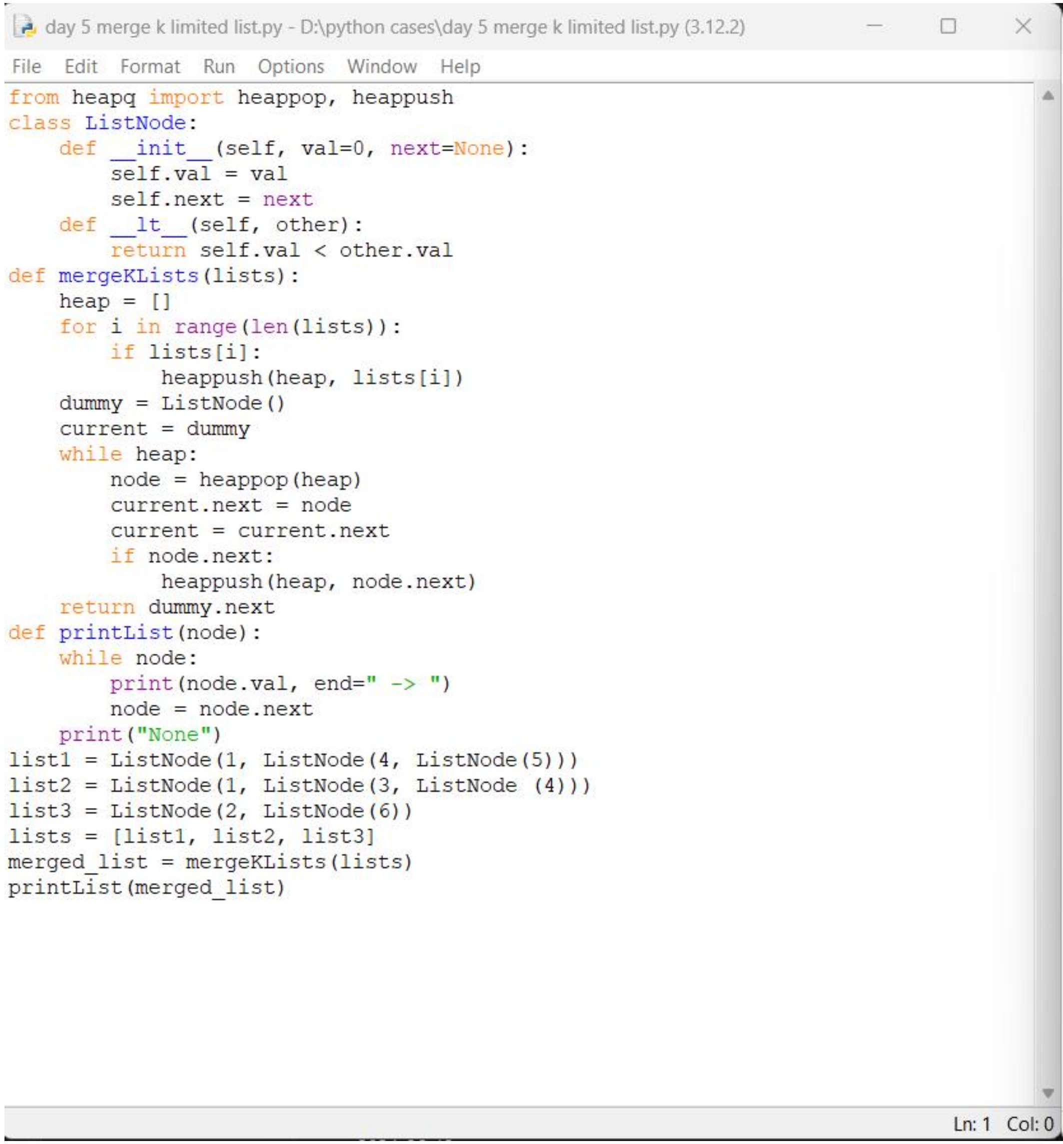
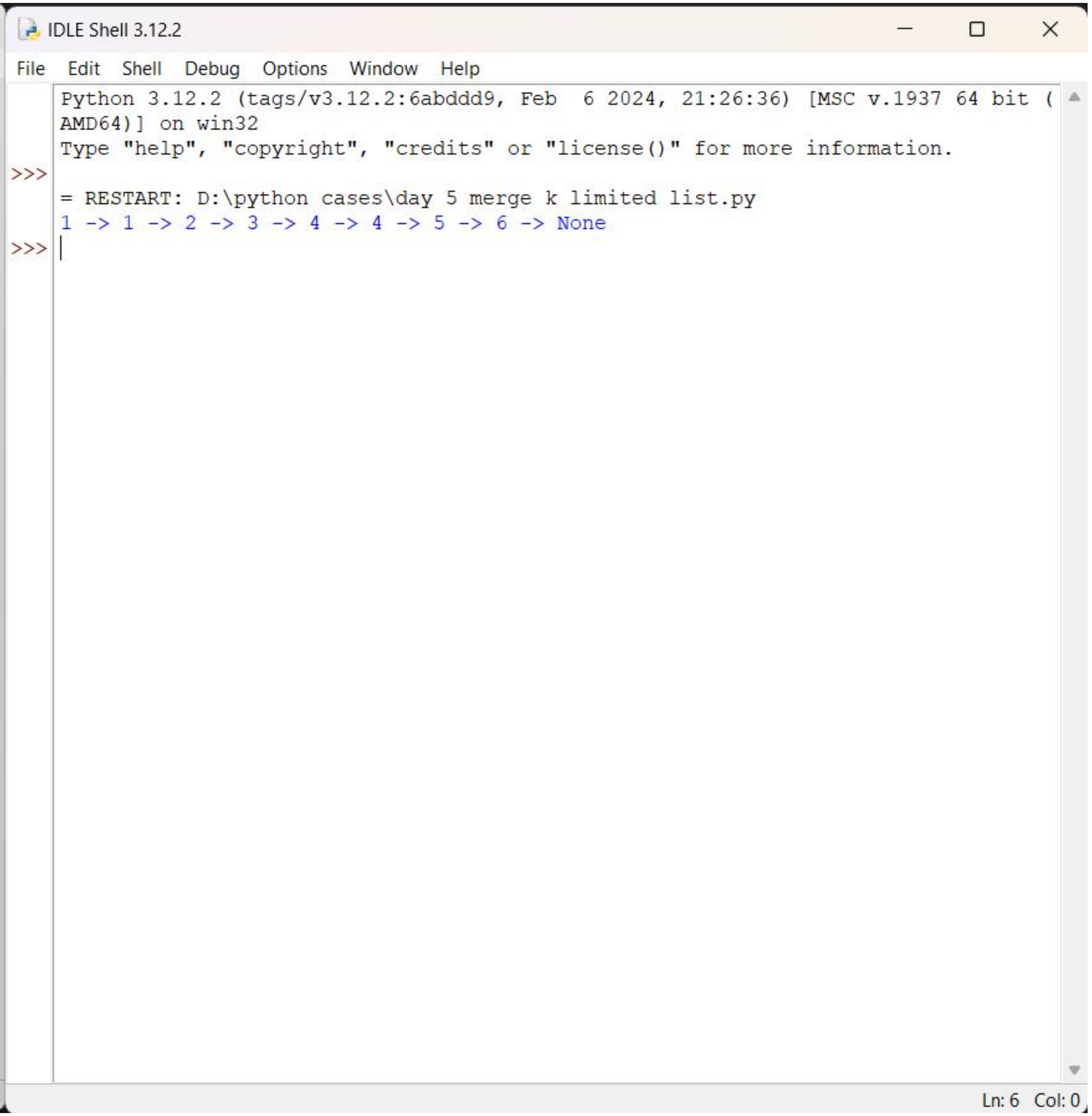

day 5 merge two sorted list.py - D:\python cases\day 5 merge two sorted list.py (3.12.2)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0

```python
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
def mergeTwoLists(list1, list2):
    dummy = ListNode()
    current = dummy
    while list1 and list2:
        if list1.val <= list2.val:
            current.next = list1
            list1 = list1.next
        else:
            current.next = list2
            list2 = list2.next
        current = current.next
    if list1:
        current.next = list1
    elif list2:
        current.next = list2
    return dummy.next
def printList(node):
    while node:
        print(node.val, end=" -> ")
        node = node.next
    print("None")
list1 = ListNode(1, ListNode(2, ListNode(4)))
list2 = ListNode(1, ListNode(3, ListNode(4)))
merged_list = mergeTwoLists(list1, list2)
printList(merged_list)
```

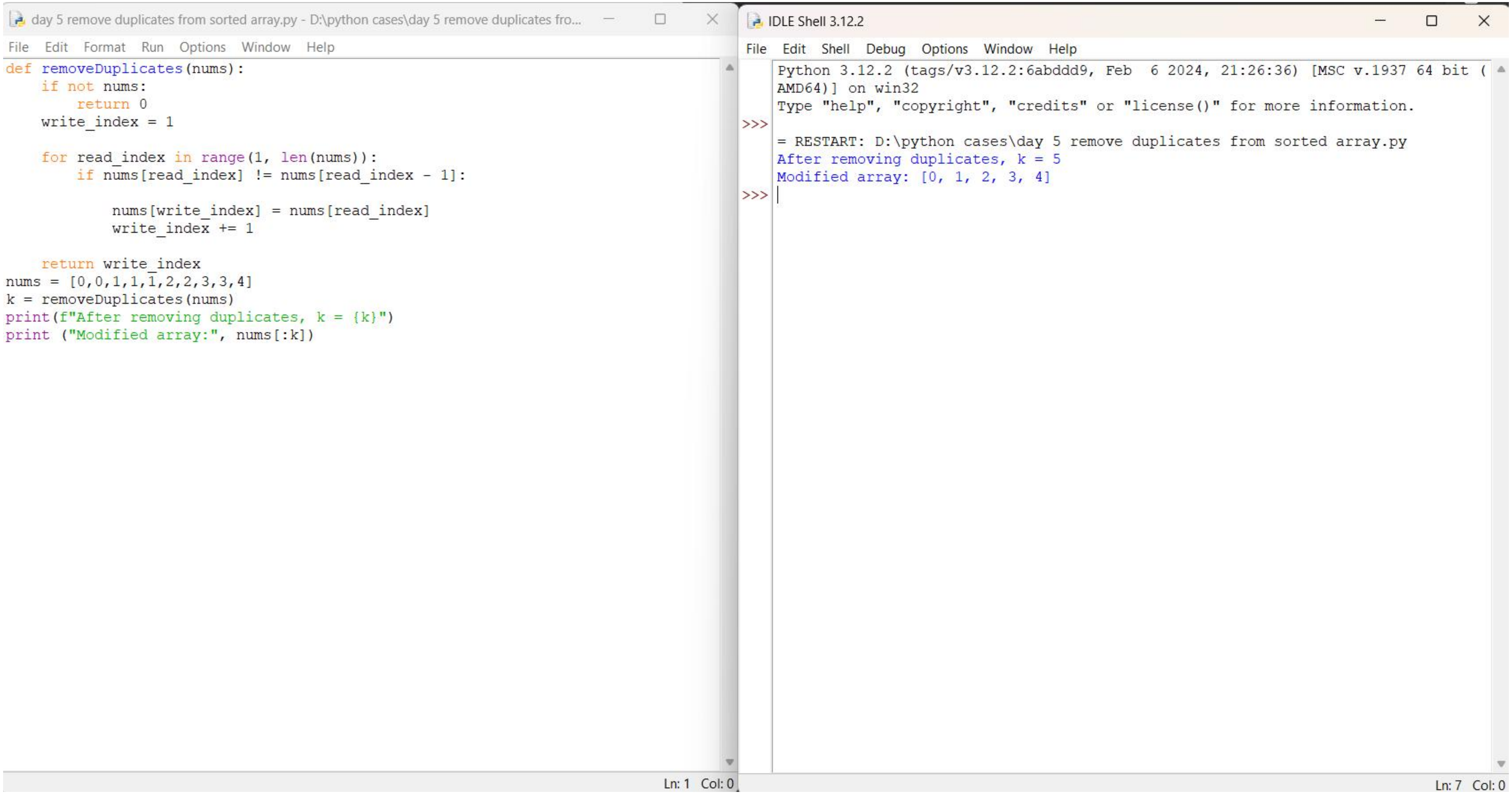
```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: D:\python cases\day 5 merge two sorted list.py
1 -> 1 -> 2 -> 3 -> 4 -> 4 -> None
>>> 
```


day 5 merge k limited list.py - D:\python cases\day 5 merge k limited list.py (3.12.2)
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```
from heapq import heappop, heappush
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
    def __lt__(self, other):
        return self.val < other.val
def mergeKLists(lists):
    heap = []
    for i in range(len(lists)):
        if lists[i]:
            heappush(heap, lists[i])
    dummy = ListNode()
    current = dummy
    while heap:
        node = heappop(heap)
        current.next = node
        current = current.next
        if node.next:
            heappush(heap, node.next)
    return dummy.next
def printList(node):
    while node:
        print(node.val, end=" -> ")
        node = node.next
    print("None")
list1 = ListNode(1, ListNode(4, ListNode(5)))
list2 = ListNode(1, ListNode(3, ListNode (4)))
list3 = ListNode(2, ListNode(6))
lists = [list1, list2, list3]
merged_list = mergeKLists(lists)
printList(merged_list)
```

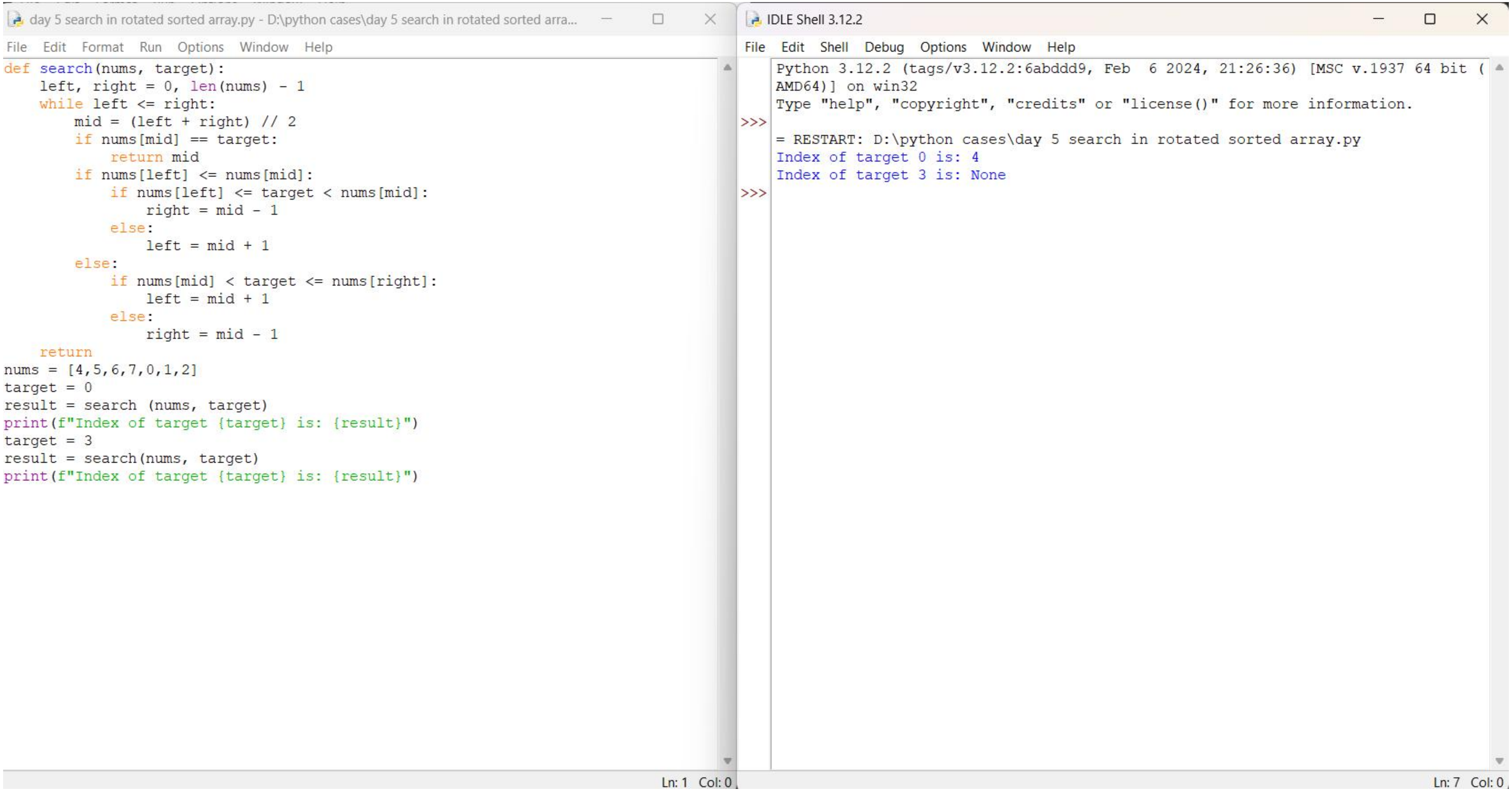


```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 merge k limited list.py
1 -> 1 -> 2 -> 3 -> 4 -> 4 -> 5 -> 6 -> None
>>>
```


day 5 remove duplicates from sorted array.py - D:\python cases\day 5 remove duplicates fro...
File Edit Format Run Options Window Help
Ln: 1 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 7 Col: 0

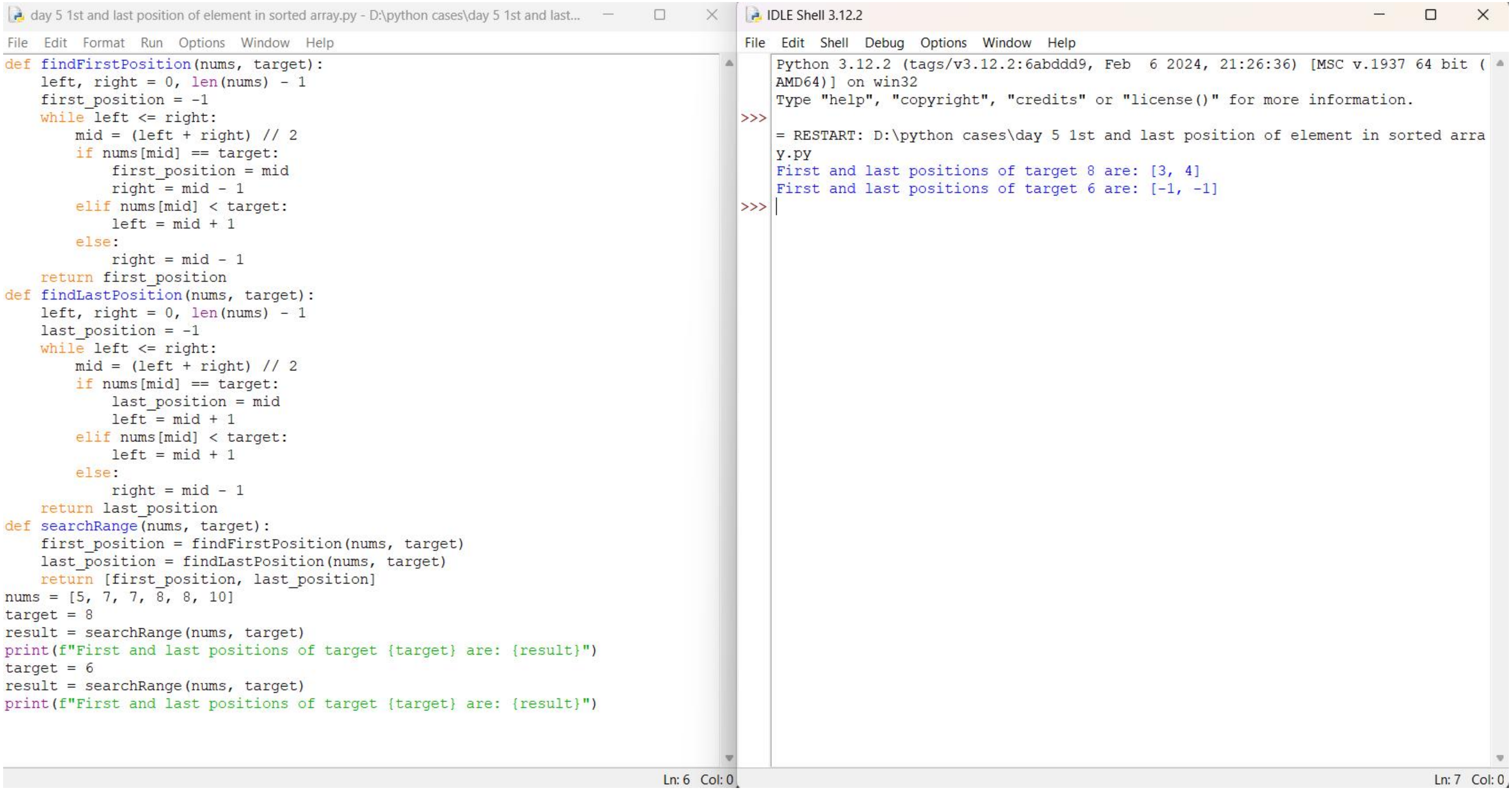
```python
def removeDuplicates(nums):
    if not nums:
        return 0
    write_index = 1

    for read_index in range(1, len(nums)):
        if nums[read_index] != nums[read_index - 1]:

            nums[write_index] = nums[read_index]
            write_index += 1

    return write_index
nums = [0,0,1,1,1,2,2,3,3,4]
k = removeDuplicates(nums)
print(f"After removing duplicates, k = {k}")
print ("Modified array:", nums[:k])
```

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 remove duplicates from sorted array.py
After removing duplicates, k = 5
Modified array: [0, 1, 2, 3, 4]
>>>
```


day 5 search in rotated sorted array.py - D:\python cases\day 5 search in rotated sorted arra...
File Edit Format Run Options Window Help
Ln: 1 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 7 Col: 0

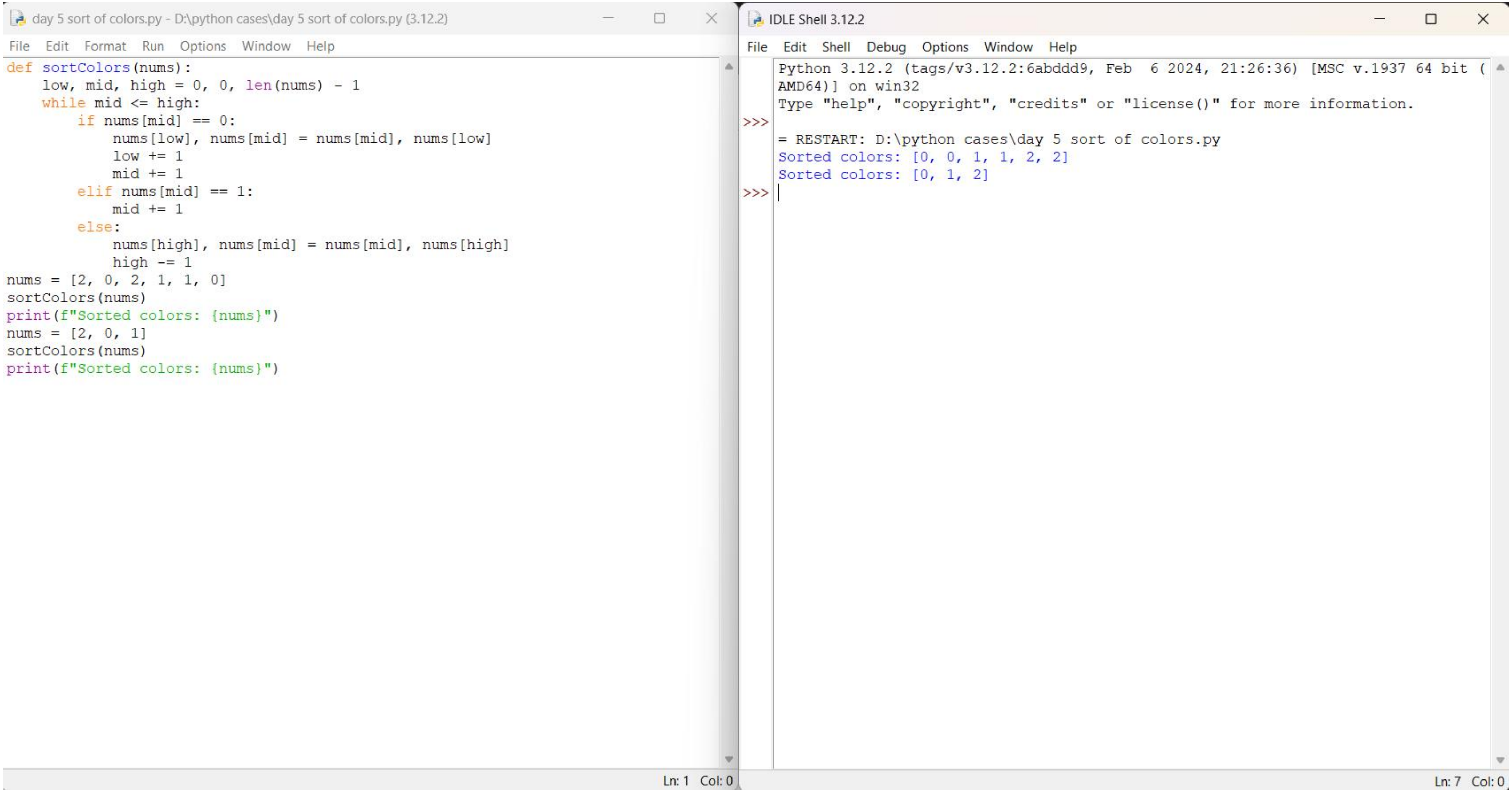
```python
def search(nums, target):
    left, right = 0, len(nums) - 1
    while left <= right:
        mid = (left + right) // 2
        if nums[mid] == target:
            return mid
        if nums[left] <= nums[mid]:
            if nums[left] <= target < nums[mid]:
                right = mid - 1
            else:
                left = mid + 1
        else:
            if nums[mid] < target <= nums[right]:
                left = mid + 1
            else:
                right = mid - 1
    return
nums = [4,5,6,7,0,1,2]
target = 0
result = search (nums, target)
print(f"Index of target {target} is: {result}")
target = 3
result = search(nums, target)
print(f"Index of target {target} is: {result}")
```

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 search in rotated sorted array.py
Index of target 0 is: 4
Index of target 3 is: None
>>>
```


day 5 1st and last position of element in sorted array.py - D:\python cases\day 5 1st and last...
File Edit Format Run Options Window Help
Ln: 6 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 7 Col: 0

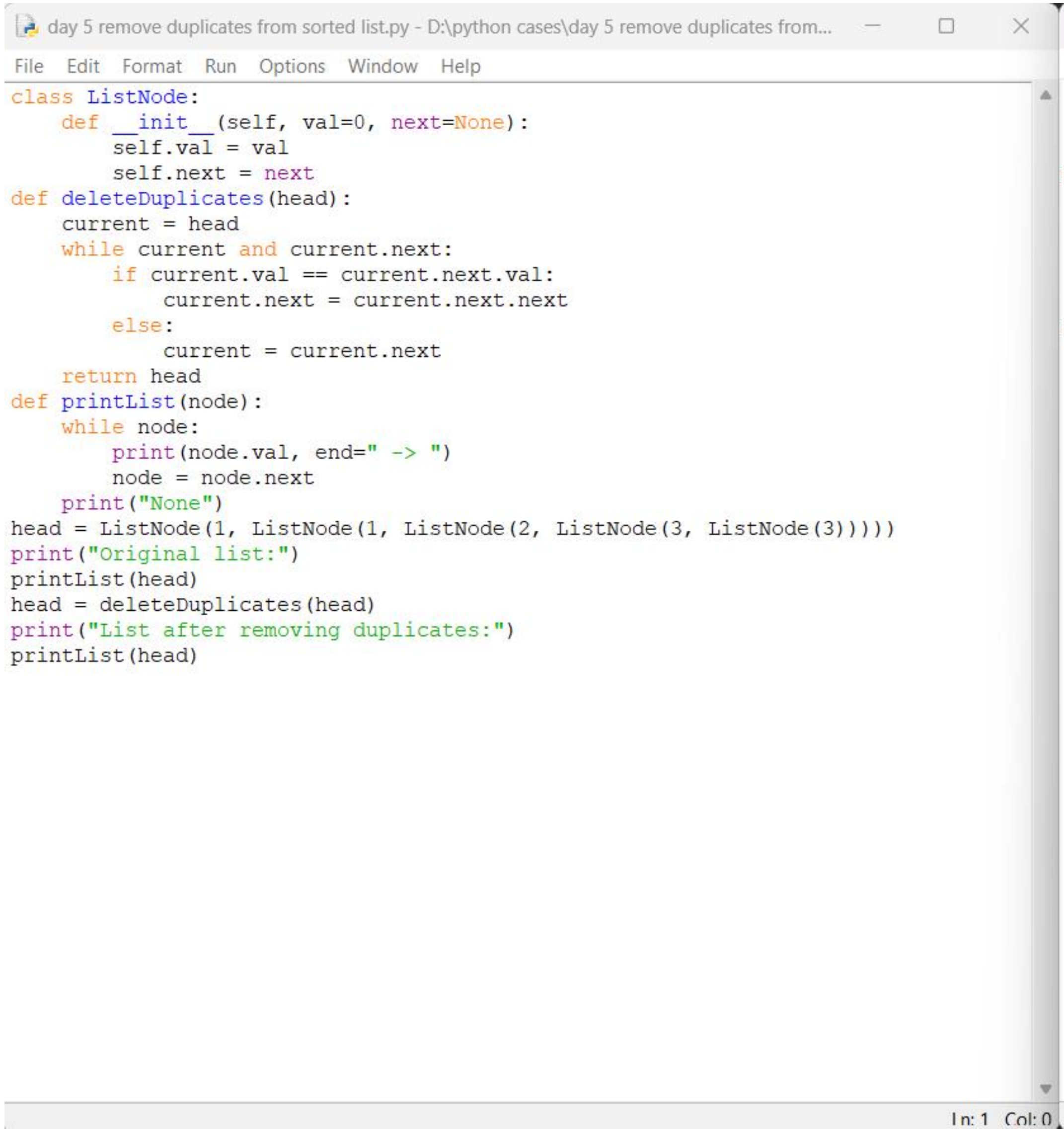
```python
def findFirstPosition(nums, target):
    left, right = 0, len(nums) - 1
    first_position = -1
    while left <= right:
        mid = (left + right) // 2
        if nums[mid] == target:
            first_position = mid
            right = mid - 1
        elif nums[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return first_position
def findLastPosition(nums, target):
    left, right = 0, len(nums) - 1
    last_position = -1
    while left <= right:
        mid = (left + right) // 2
        if nums[mid] == target:
            last_position = mid
            left = mid + 1
        elif nums[mid] < target:
            left = mid + 1
        else:
            right = mid - 1
    return last_position
def searchRange(nums, target):
    first_position = findFirstPosition(nums, target)
    last_position = findLastPosition(nums, target)
    return [first_position, last_position]
nums = [5, 7, 7, 8, 8, 10]
target = 8
result = searchRange(nums, target)
print(f"First and last positions of target {target} are: {result}")
target = 6
result = searchRange(nums, target)
print(f"First and last positions of target {target} are: {result}")
```

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 1st and last position of element in sorted arra
y.py
First and last positions of target 8 are: [3, 4]
First and last positions of target 6 are: [-1, -1]
>>>
```


day 5 sort of colors.py - D:\python cases\day 5 sort of colors.py (3.12.2)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 7 Col: 0

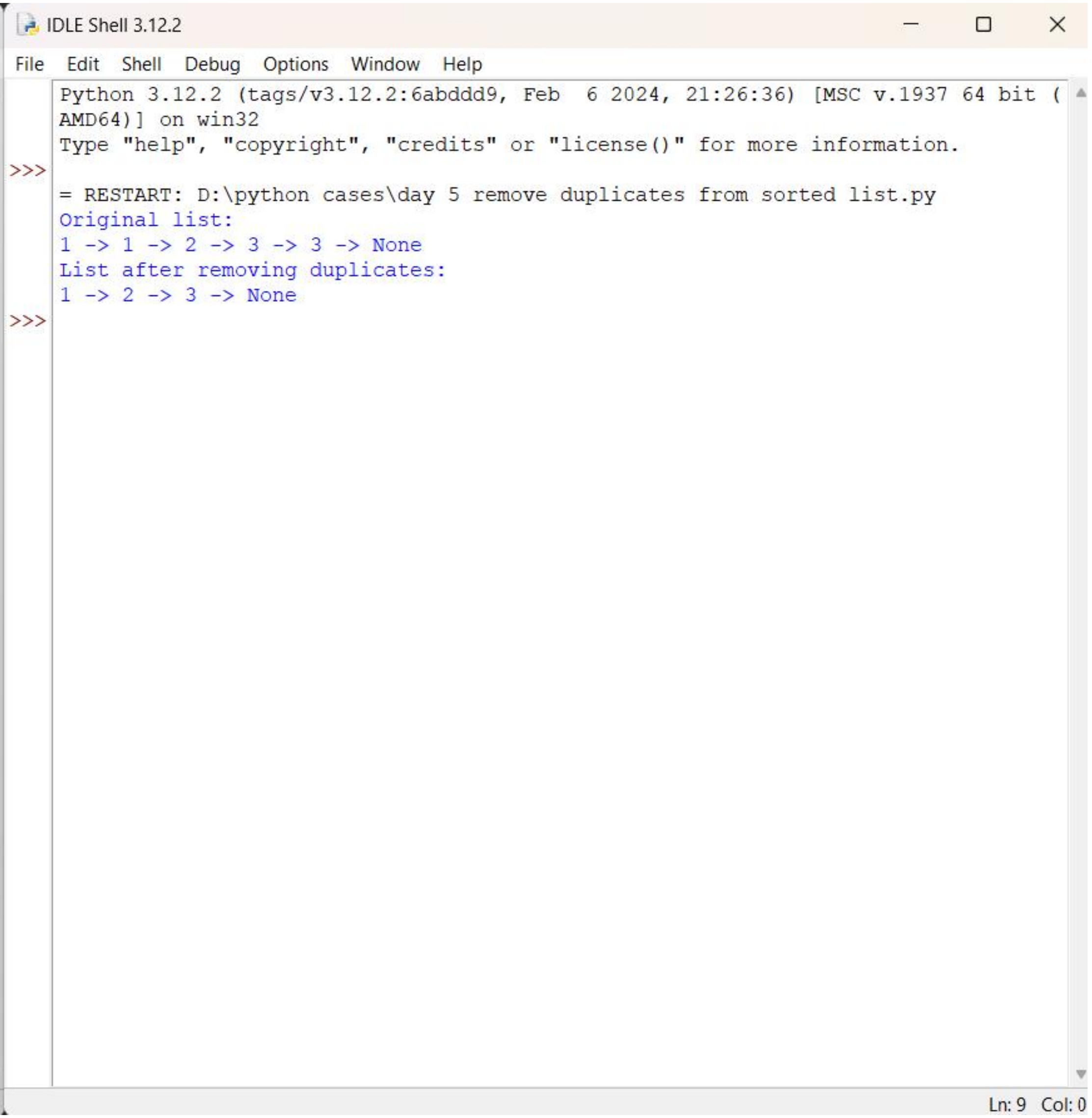
```
def sortColors(nums):
    low, mid, high = 0, 0, len(nums) - 1
    while mid <= high:
        if nums[mid] == 0:
            nums[low], nums[mid] = nums[mid], nums[low]
            low += 1
            mid += 1
        elif nums[mid] == 1:
            mid += 1
        else:
            nums[high], nums[mid] = nums[mid], nums[high]
            high -= 1
nums = [2, 0, 2, 1, 1, 0]
sortColors(nums)
print(f"Sorted colors: {nums}")
nums = [2, 0, 1]
sortColors(nums)
print(f"Sorted colors: {nums}")
```

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 sort of colors.py
Sorted colors: [0, 0, 1, 1, 2, 2]
Sorted colors: [0, 1, 2]
>>>
```

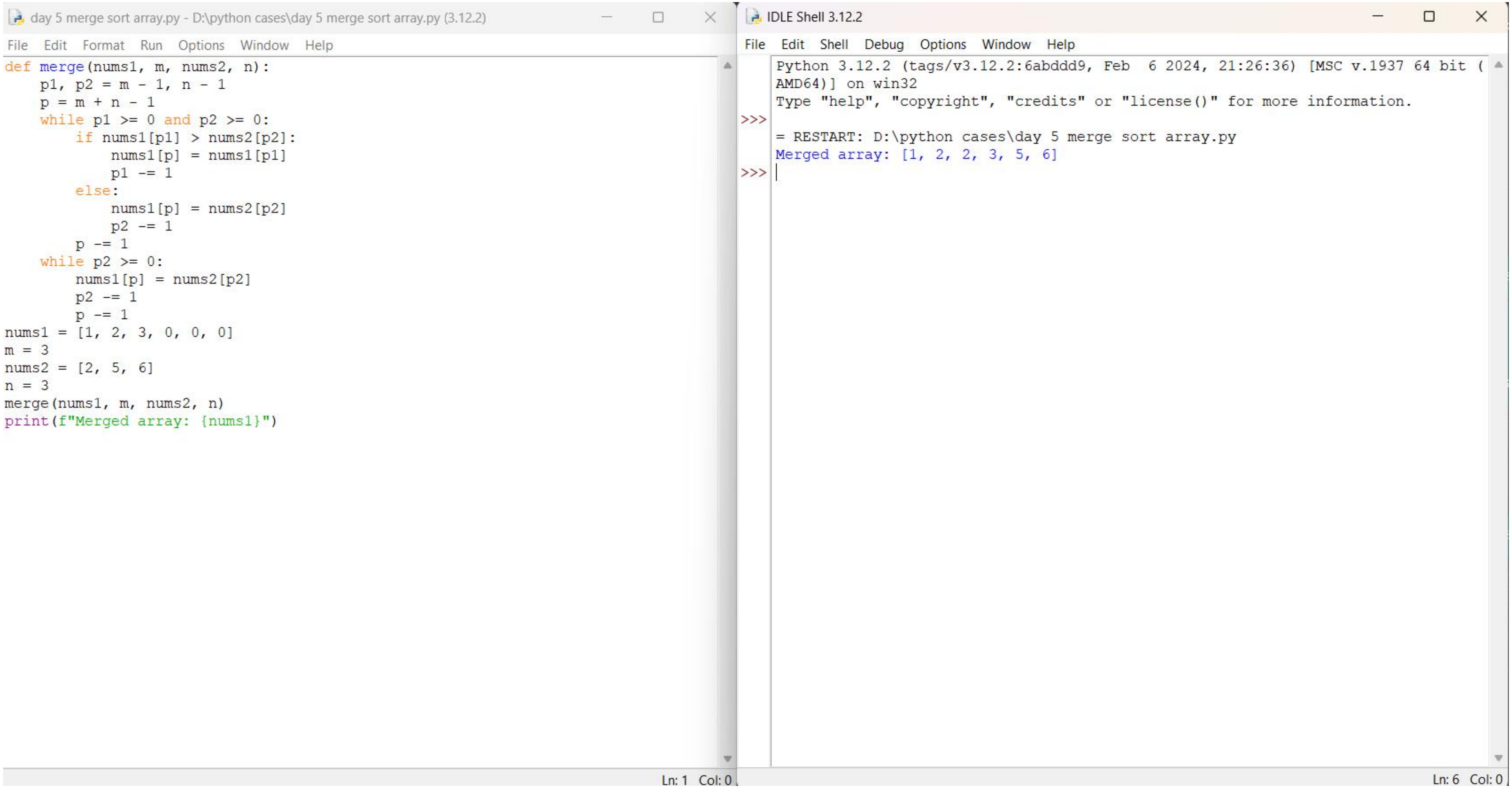


day 5 remove duplicates from sorted list.py - D:\python cases\day 5 remove duplicates from...
File Edit Format Run Options Window Help
Ln: 1 Col: 0

```python
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
def deleteDuplicates(head):
    current = head
    while current and current.next:
        if current.val == current.next.val:
            current.next = current.next.next
        else:
            current = current.next
    return head
def printList(node):
    while node:
        print(node.val, end=" -> ")
        node = node.next
    print("None")
head = ListNode(1, ListNode(1, ListNode(2, ListNode(3, ListNode(3)))))
print("Original list:")
printList(head)
head = deleteDuplicates(head)
print("List after removing duplicates:")
printList(head)
```

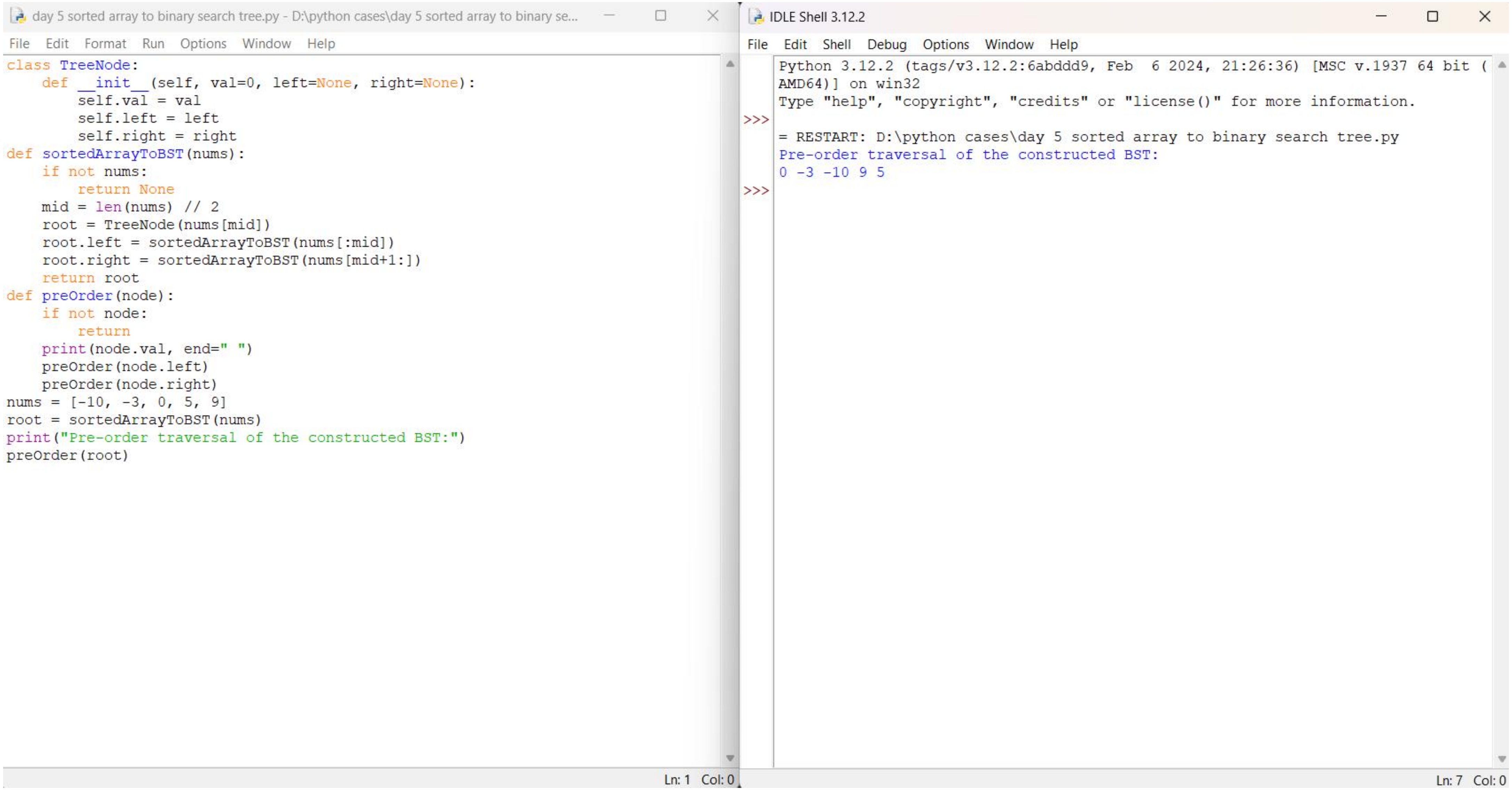


```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 remove duplicates from sorted list.py
Original list:
1 -> 1 -> 2 -> 3 -> 3 -> None
List after removing duplicates:
1 -> 2 -> 3 -> None
>>>
```


day 5 merge sort array.py - D:\python cases\day 5 merge sort array.py (3.12.2)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 6 Col: 0

```python
def merge(nums1, m, nums2, n):
    p1, p2 = m - 1, n - 1
    p = m + n - 1
    while p1 >= 0 and p2 >= 0:
        if nums1[p1] > nums2[p2]:
            nums1[p] = nums1[p1]
            p1 -= 1
        else:
            nums1[p] = nums2[p2]
            p2 -= 1
        p -= 1
    while p2 >= 0:
        nums1[p] = nums2[p2]
        p2 -= 1
        p -= 1
nums1 = [1, 2, 3, 0, 0, 0]
m = 3
nums2 = [2, 5, 6]
n = 3
merge(nums1, m, nums2, n)
print(f"Merged array: {nums1}")
```

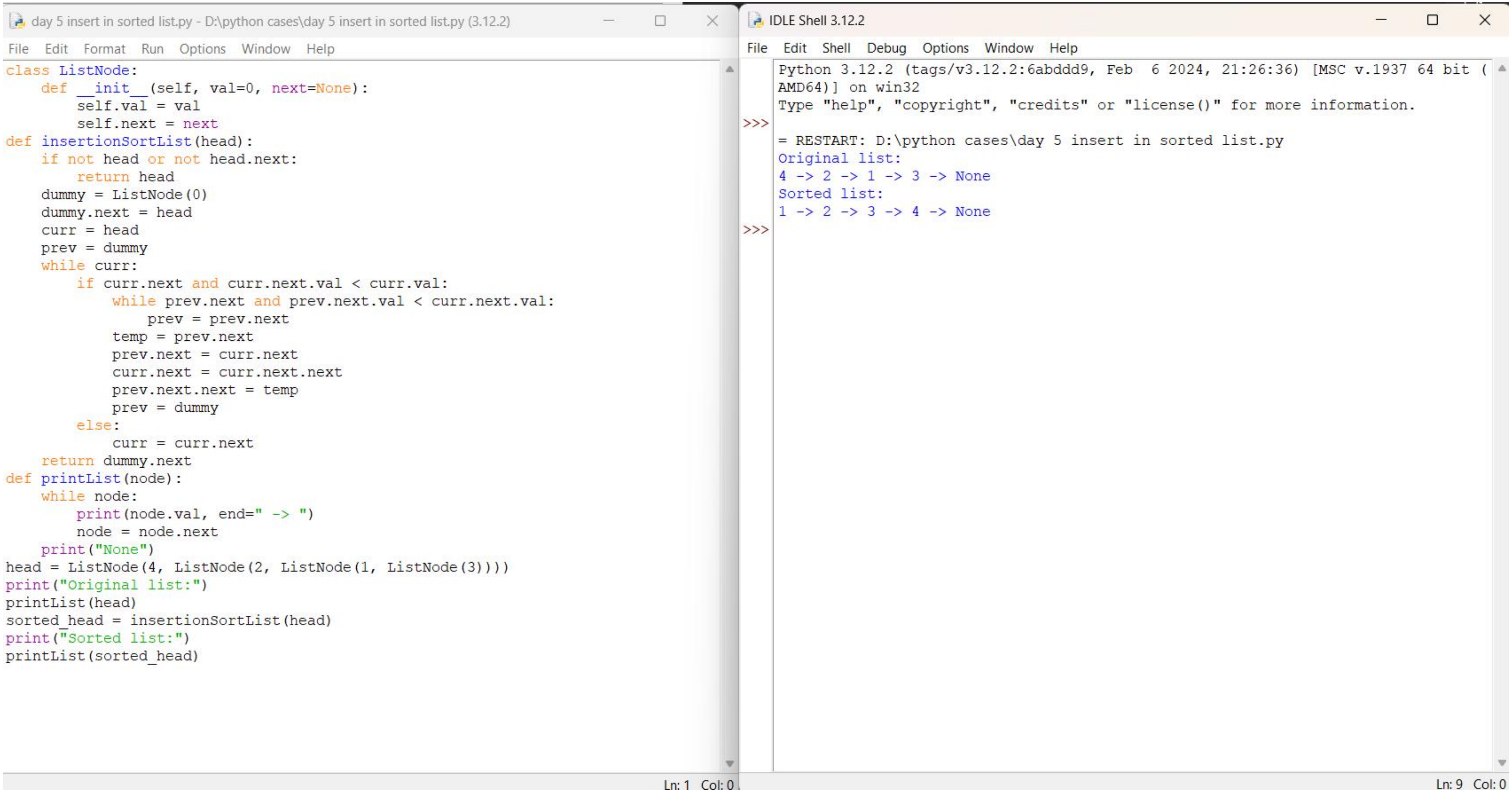
```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: D:\python cases\day 5 merge sort array.py
Merged array: [1, 2, 2, 3, 5, 6]
>>> 
```



day 5 sorted array to binary search tree.py - D:\python cases\day 5 sorted array to binary se...
File Edit Format Run Options Window Help
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 1 Col: 0
Ln: 7 Col: 0

```python
class TreeNode:
    def __init__(self, val=0, left=None, right=None):
        self.val = val
        self.left = left
        self.right = right
def sortedArrayToBST(nums):
    if not nums:
        return None
    mid = len(nums) // 2
    root = TreeNode(nums[mid])
    root.left = sortedArrayToBST(nums[:mid])
    root.right = sortedArrayToBST(nums[mid+1:])
    return root
def preOrder(node):
    if not node:
        return
    print(node.val, end=" ")
    preOrder(node.left)
    preOrder(node.right)
nums = [-10, -3, 0, 5, 9]
root = sortedArrayToBST(nums)
print("Pre-order traversal of the constructed BST:")
preOrder(root)
```

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 sorted array to binary search tree.py
Pre-order traversal of the constructed BST:
0 -3 -10 9 5
>>>
```


day 5 insert in sorted list.py - D:\python cases\day 5 insert in sorted list.py (3.12.2)
File Edit Format Run Options Window Help
Ln: 1 Col: 0
IDLE Shell 3.12.2
File Edit Shell Debug Options Window Help
Ln: 9 Col: 0

```python
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
def insertionSortList(head):
    if not head or not head.next:
        return head
    dummy = ListNode(0)
    dummy.next = head
    curr = head
    prev = dummy
    while curr:
        if curr.next and curr.next.val < curr.val:
            while prev.next and prev.next.val < curr.next.val:
                prev = prev.next
            temp = prev.next
            prev.next = curr.next
            curr.next = curr.next.next
            prev.next.next = temp
            prev = dummy
        else:
            curr = curr.next
    return dummy.next
def printList(node):
    while node:
        print(node.val, end=" -> ")
        node = node.next
    print("None")
head = ListNode(4, ListNode(2, ListNode(1, ListNode(3))))
print("Original list:")
printList(head)
sorted_head = insertionSortList(head)
print("Sorted list:")
printList(sorted_head)
```

```
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\python cases\day 5 insert in sorted list.py
Original list:
4 -> 2 -> 1 -> 3 -> None
Sorted list:
1 -> 2 -> 3 -> 4 -> None
>>>
```