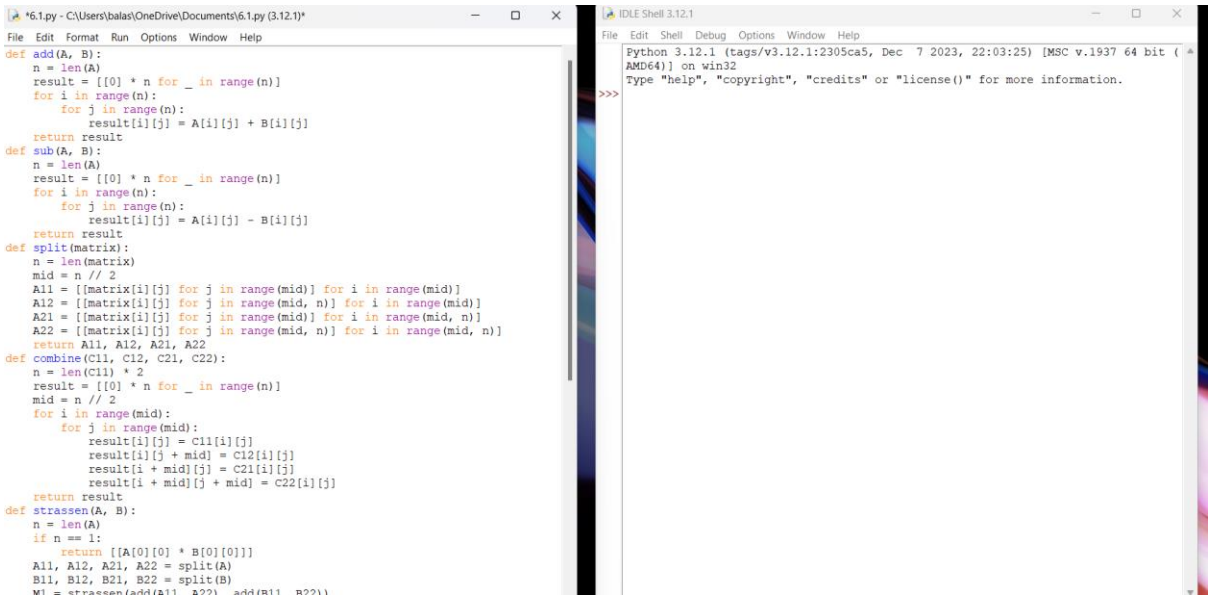
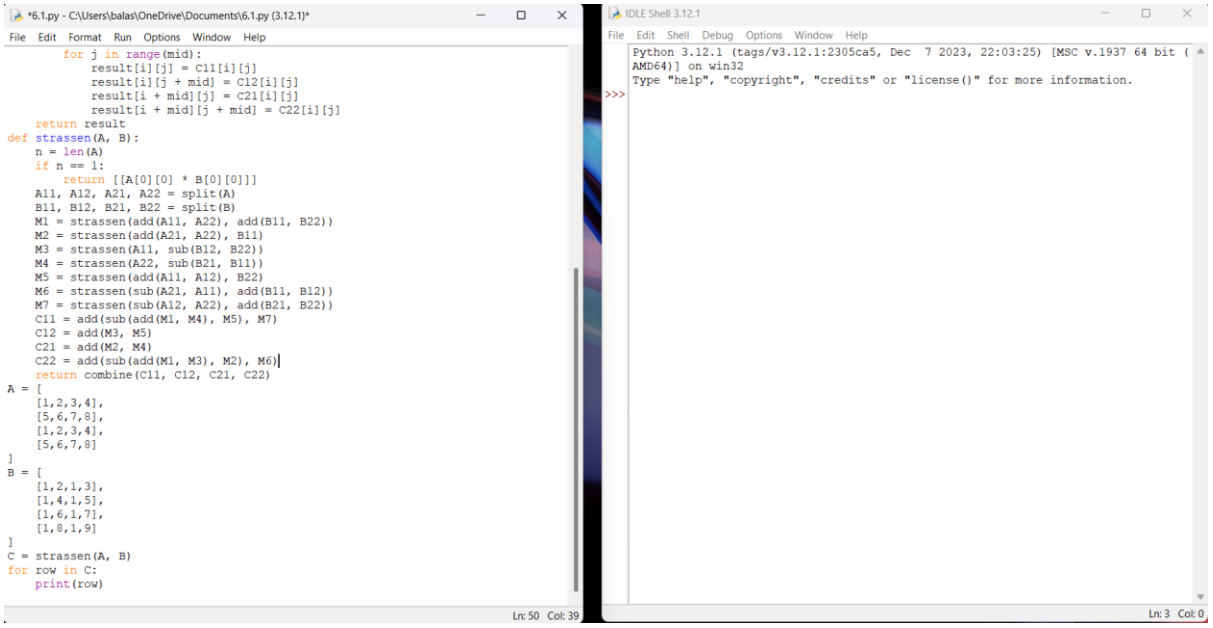
1.Perform matrix multiplication using Strassen's method.





2.Merge Two Sorted Lists You are given the heads of two sorted linked lists list1 and list2. Merge the two lists in a one sorted list. The list should be made by splicing together the nodes of the first two lists. Return the head of the merged linked list.

6.2.py - C:\Users\balas\OneDrive\Documents\6.2.py (3.12.1)

File Edit Format Run Options Window Help

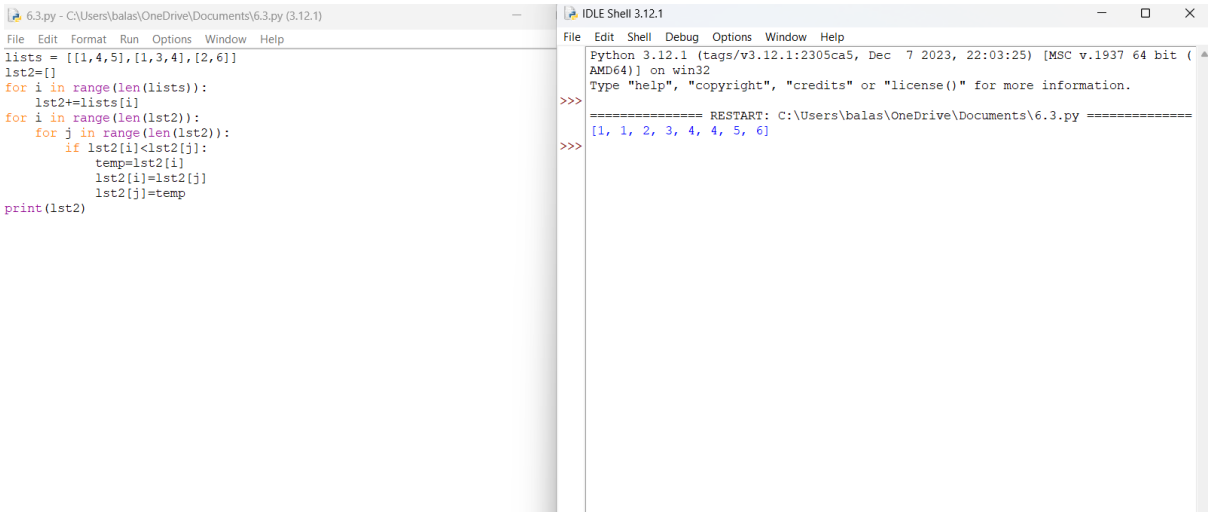
```
import ctypes
class Node:
    def init(self, data=None, next=None):
        self.data = data
        self.next = next
head = None
tail = None
n = None
def ins_end(num):
    global head, tail, n
    n = Node()
    n.data = num
    n.next = None
    if head is None:
        head = n
        tail = n
    else:
        tail.next = n
        tail = n
def sort():
    global head
    t = head
    while t is not None:
        s = t.next
        while s is not None:
            if t.data > s.data:
                t.data, s.data = s.data, t.data
            s = s.next
        t = t.next
def display():
    global head
    t = head
    while t is not None:
        print(f"{t.data},", end="")
        t = t.next
    print()
ins_end(1)
ins_end(2)
ins_end(4)
ins_end(1)
ins_end(3)
ins_end(4)
sort()
display()
```

IDLE Shell 3.12.1

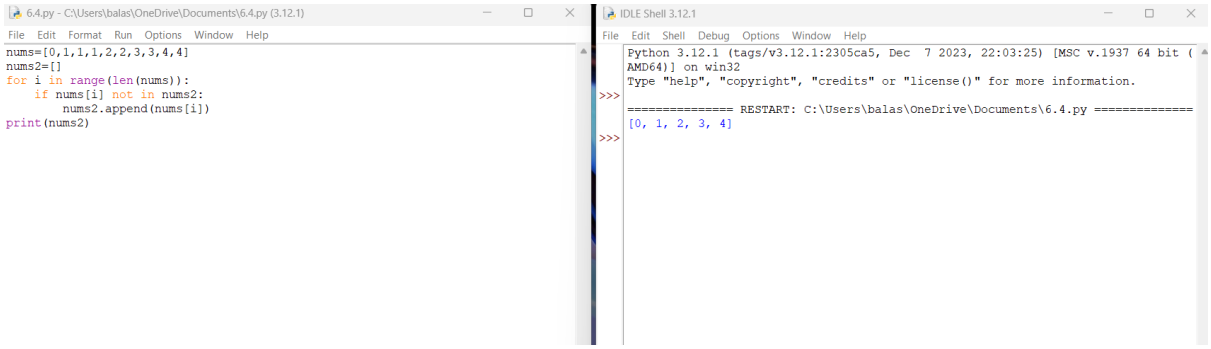
File Edit Shell Debug Options Window Help

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\balas\OneDrive\Documents\6.2.py ==========
1,1,2,3,4,4,
>>>
```

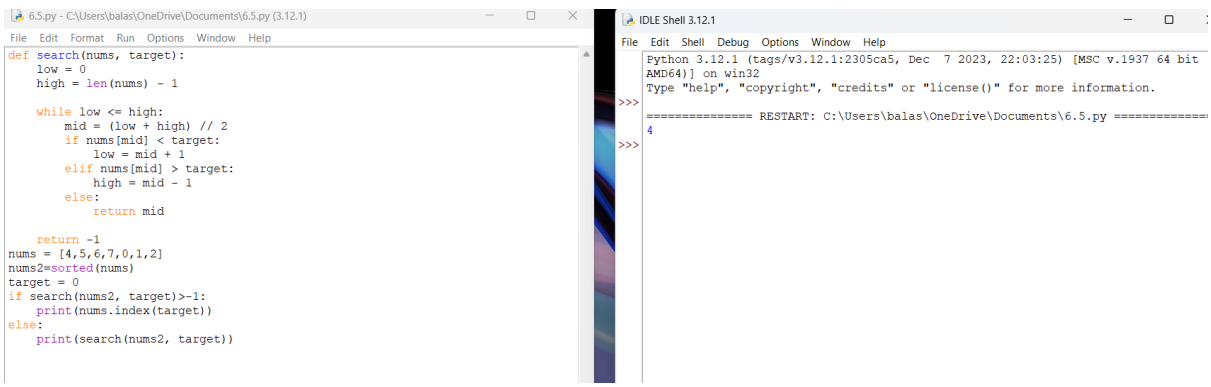
3. Merge k Sorted Lists You are given an array of k linked-lists lists, each linked-list is sorted in ascending order. Merge all the linked-lists into one sorted linked-list and return it. Example 1: Input: lists = [[1,4,5],[1,3,4],[2,6]] Output: [1,1,2,3,4,4,5,6].



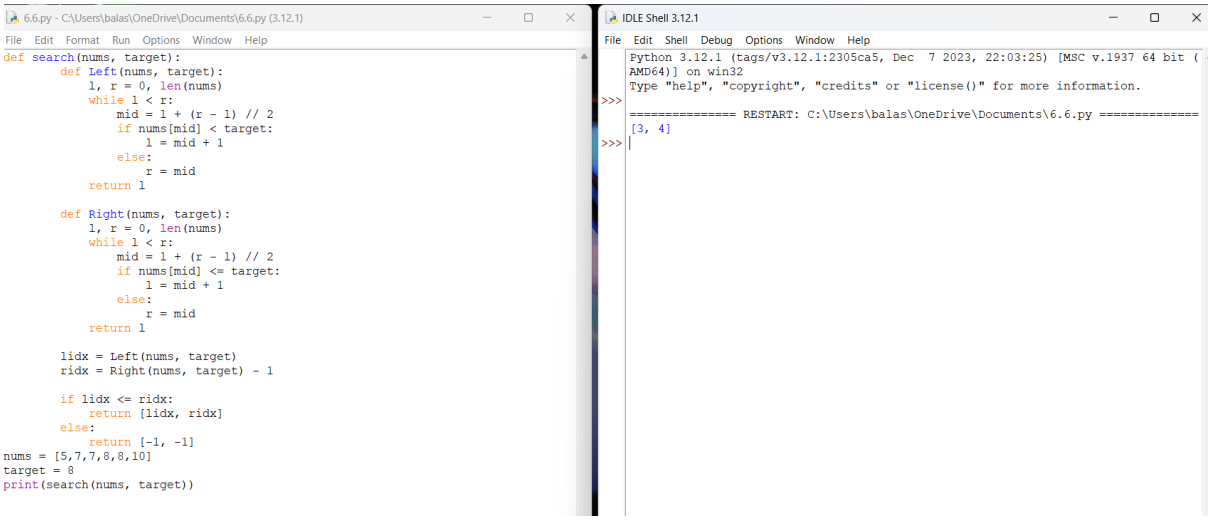
4. Remove Duplicates from Sorted Array Given an integer array nums sorted in non-decreasing order, remove the duplicates in place such that each unique element appears only once. The relative order of the elements should be kept the same. Since it is impossible to change the length of the array in some languages, you must instead have the result be placed in the first part of the array nums. More formally, if there are k elements after removing the duplicates, then the first k elements of nums should hold the final result. It does not matter what you leave beyond the first k elements. Return k after placing the final result in the first k slots of nums. Do not allocate extra space for another array. You must do this by modifying the input array in-place with O(1) extra memory.



5. Search in Rotated Sorted Array There is an integer array nums sorted in ascending order (with distinct values). Prior to being passed to your function, nums is possibly rotated at an unknown pivot index k (1 <= k < nums.length) such that the resulting array is [nums[k], nums[k+1], ..., nums[n1], nums[0], nums[1], ..., nums[k-1]] (0-indexed). For example, [0,1,2,4,5,6,7] might be rotated at pivot index 3 and become [4,5,6,7,0,1,2]. Given the array nums after the possible rotation and an integer target, return the index of target if it is in nums, or -1 if it is not in nums.



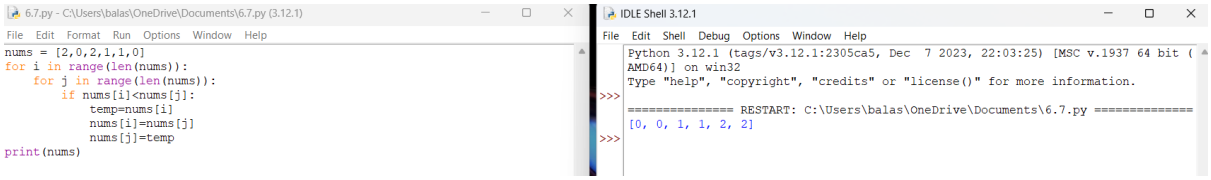
6. Find First and Last Position of Element in Sorted Array Given an array of integers nums sorted in non-decreasing order, find the starting and ending position of a given target value. If target is not found in the array, return [-1, -1].



```
def search(nums, target):
        def Left(nums, target):
            l, r = 0, len(nums)
            while l < r:
                mid = l + (r - l) // 2
                if nums[mid] < target:
                    l = mid + 1
                else:
                    r = mid
            return l

        def Right(nums, target):
            l, r = 0, len(nums)
            while l < r:
                mid = l + (r - l) // 2
                if nums[mid] <= target:
                    l = mid + 1
                else:
                    r = mid
            return l

        lidx = Left(nums, target)
        ridx = Right(nums, target) - 1

        if lidx <= ridx:
            return [lidx, ridx]
        else:
            return [-1, -1]
nums = [5,7,7,8,8,10]
target = 8
print(search(nums, target))
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\balas\OneDrive\Documents\6.6.py ==========
[3, 4]
>>>
```

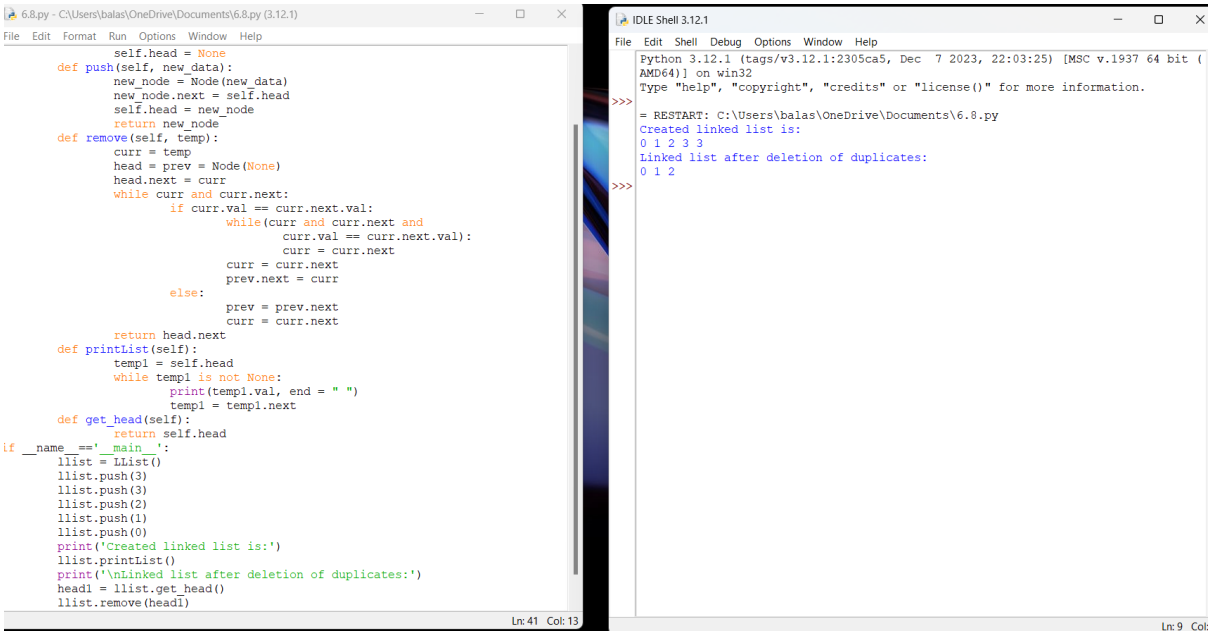
7. Sort Colors Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue. We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively. You must solve this problem without using the library's sort function.



```
nums = [2,0,2,1,1,0]
for i in range(len(nums)):
    for j in range(len(nums)):
        if nums[i]<nums[j]:
            temp=nums[i]
            nums[i]=nums[j]
            nums[j]=temp
print(nums)
```

```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\balas\OneDrive\Documents\6.7.py ==========
[0, 0, 1, 1, 2, 2]
>>>
```

8. Remove Duplicates from Sorted List Given the head of a sorted linked list, delete all duplicates such that each element appears only once. Return the linked list sorted as well.

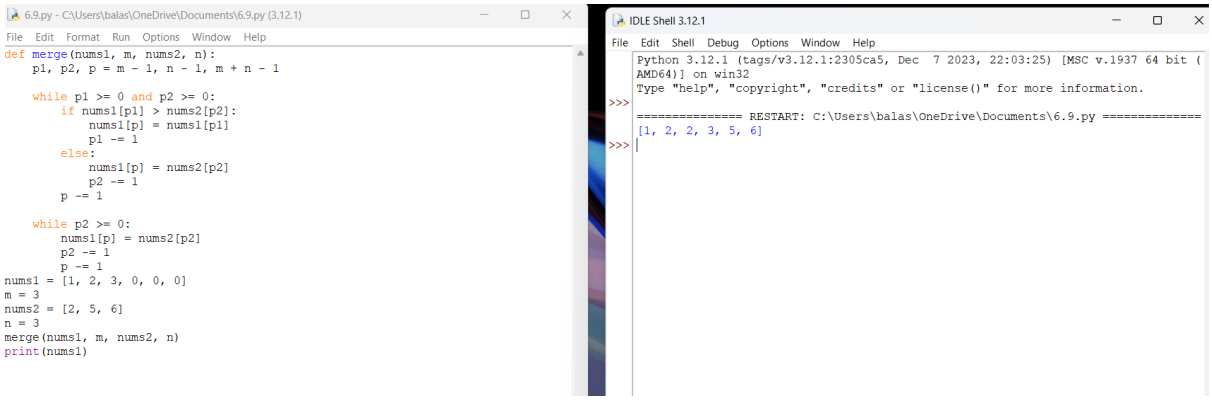


```
            self.head = None
        def push(self, new_data):
            new_node = Node(new_data)
            new_node.next = self.head
            self.head = new_node
            return new_node
        def remove(self, temp):
            curr = temp
            head = prev = Node(None)
            head.next = curr
            while curr and curr.next:
                if curr.val == curr.next.val:
                    while(curr and curr.next and
                          curr.val == curr.next.val):
                          curr = curr.next
                    curr = curr.next
                    prev.next = curr
                else:
                    prev = prev.next
                    curr = curr.next
            return head.next
        def printList(self):
            temp1 = self.head
            while temp1 is not None:
                print(temp1.val, end = " ")
                temp1 = temp1.next
        def get_head(self):
            return self.head
if __name__=='__main__':
        llist = LList()
        llist.push(3)
        llist.push(3)
        llist.push(2)
        llist.push(1)
        llist.push(0)
        print('Created linked list is:')
        llist.printList()
        print('\nLinked list after deletion of duplicates:')
        head1 = llist.get_head()
        llist.remove(head1)
```

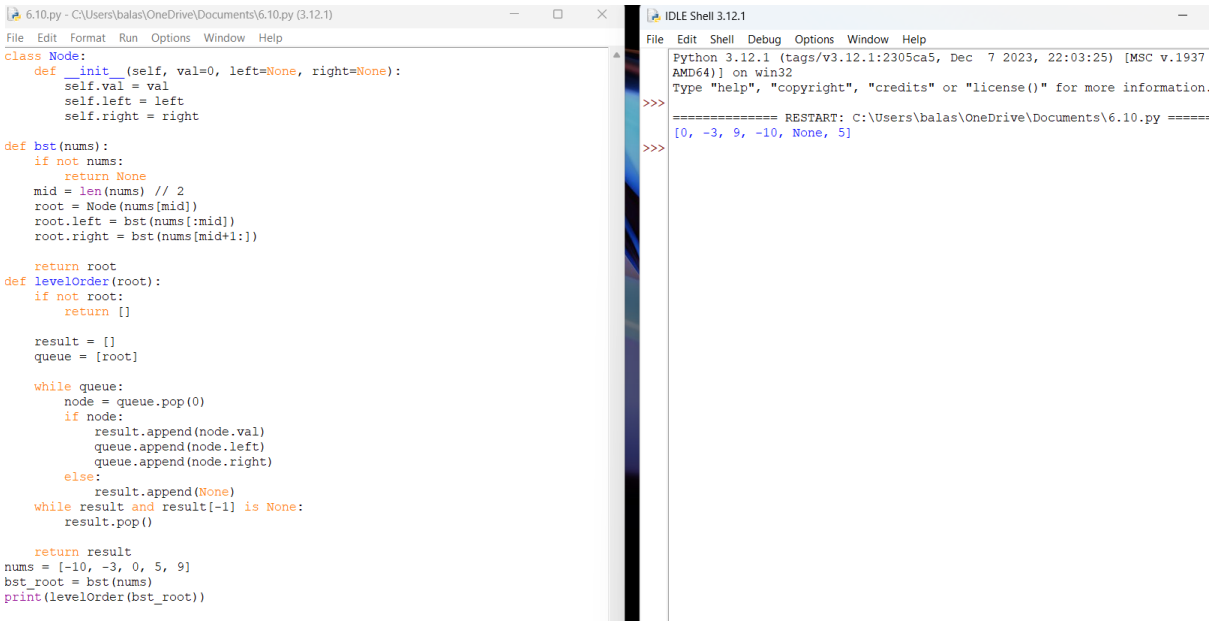
```
Python 3.12.1 (tags/v3.12.1:2305ca5, Dec  7 2023, 22:03:25) [MSC v.1937 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\balas\OneDrive\Documents\6.8.py
Created linked list is:
0 1 2 3 3
Linked list after deletion of duplicates:
0 1 2
>>>
```

```
f __name__=='__main__':
        llist = LList()
        llist.push(3)
        llist.push(3)
        llist.push(2)
        llist.push(1)
        llist.push(0)
        print('Created linked list is:')
        llist.printList()
        print('\nLinked list after deletion of duplicates:')
        head1 = llist.get_head()
        llist.remove(head1)
        llist.printList()
```

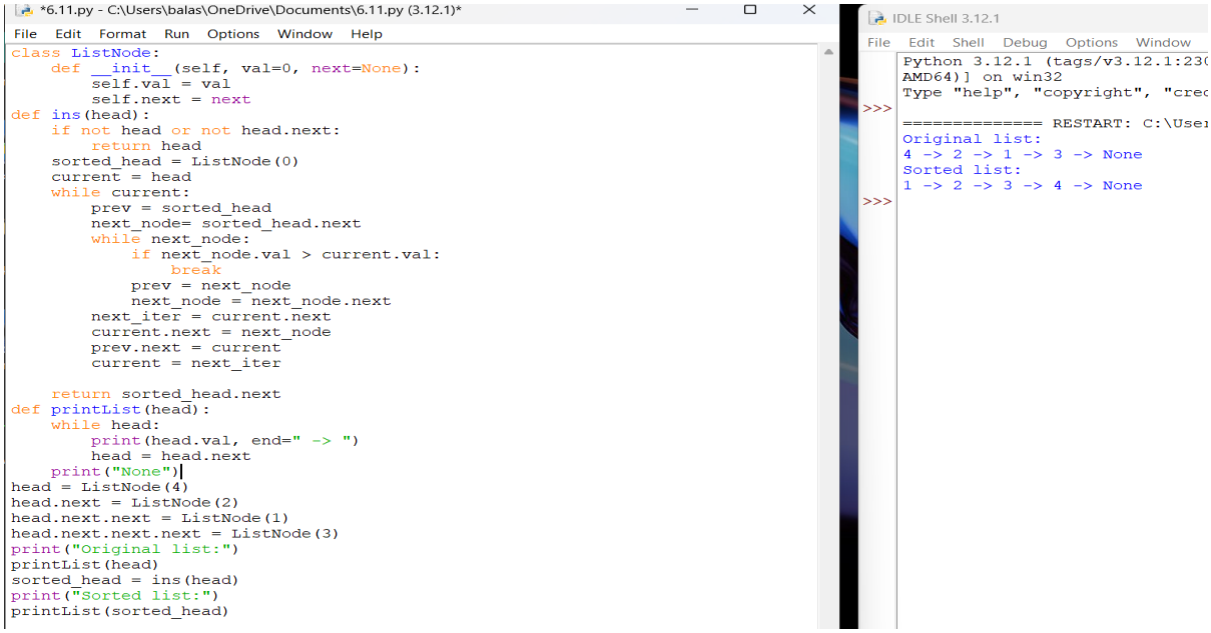
9. Merge Sorted Array You are given two integer arrays nums1 and nums2, sorted in non-decreasing order, and two integers m and n, representing the number of elements in nums1 and nums2 respectively. Merge nums1 and nums2 into a single array sorted in non-decreasing order.



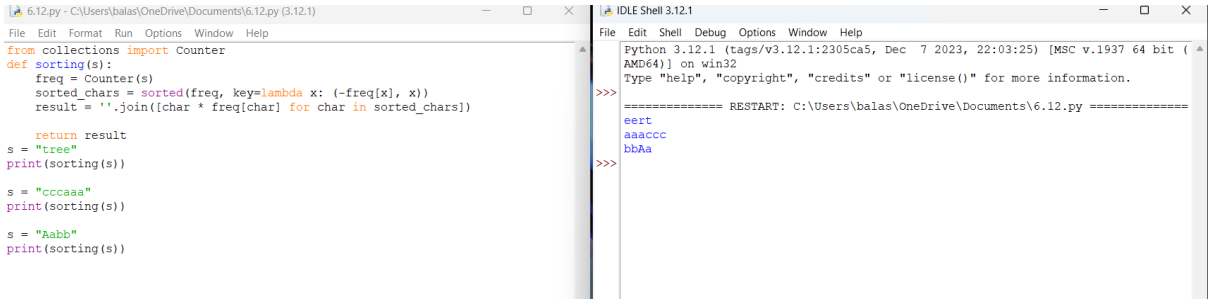
10. Convert Sorted Array to Binary Search Tree Given an integer array nums where the elements are sorted in ascending order, convert it to a height-balanced binary search tree.



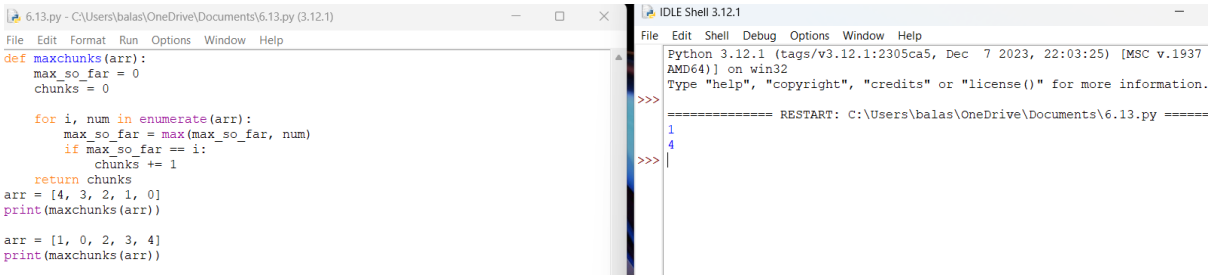
11. Insertion Sort List Given the head of a singly linked list, sort the list using insertion sort, and return the sorted list's head.



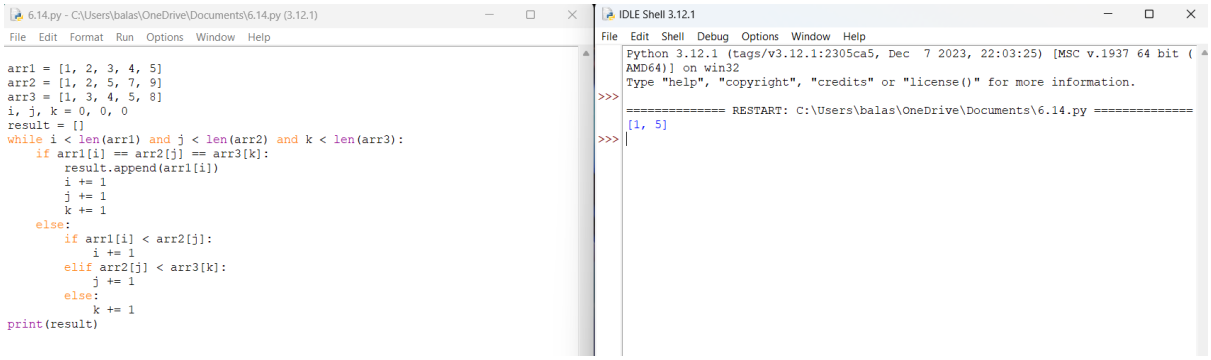
12. Sort Characters By Frequency Given a string s, sort it in decreasing order based on the frequency of the characters. The frequency of a character is the number of times it appears in the string. Return the sorted string. If there are multiple answers, return any of them.



13. . Max Chunks To Make Sorted You are given an integer array arr of length n that represents a permutation of the integers in the range [0, n - 1]. We split arr into some number of chunks (i.e., partitions), and individually sort each chunk. After concatenating them, the result should equal the sorted array. Return the largest number of chunks we can make to sort the array.



14. Intersection of Three Sorted Arrays Given three integer arrays arr1, arr2 and arr3 sorted in strictly increasing order, return a sorted array of only the integers that appeared in all three arrays.



15. 4. Sort the Matrix Diagonally A matrix diagonal is a diagonal line of cells starting from some cell in either the topmost row or leftmost column and going in the bottom-right direction until reaching the matrix's end. For example, the matrix diagonal starting from mat[2][0], where mat is a 6 x 3 matrix, includes cells mat[2][0], mat[3][1], and mat[4][2]