#Question 1: Shipping Company Optimization

def minimum\_containers(shipments, container\_limits):

    shipments.sort(reverse=True)

    container\_limits.sort(reverse=True)

    used\_containers = 0

    for shipment in shipments:

        for i in range(len(container\_limits)):

            if container\_limits[i] >= shipment:

                container\_limits[i] -= shipment

                if container\_limits[i] == 0:

                    used\_containers += 1

                break

        else:

            return -1

    return used\_containers + len([c for c in container\_limits if c > 0])

# Example

shipments = [10, 20, 30]

container\_limits = [15, 15, 20, 10]

print("Minimum number of containers needed:", minimum\_containers(shipments, container\_limits))

#Question 2: Detecting Cycle in a Linked List

class ListNode:

    def \_\_init\_\_(self, value=0, next=None):

        self.value = value

        self.next = next

def detect\_cycle(head):

    slow = head

    fast = head

    while fast and fast.next:

        slow = slow.next

        fast = fast.next.next

        if slow == fast:

            return True

    return False

# Example

head = ListNode(20)

node1 = ListNode(30)

node2 = ListNode(40)

node3 = ListNode(60)

node4 = ListNode(80)

head.next = node1

node1.next = node2

node2.next = node3

node3.next = node4

node4.next = node2

print("Cycle detected:", detect\_cycle(head))