```python
def threeSumClosest(self, nums: List[int], target: int) -> int:
    nums.sort()
    closest_sum=float('inf')
    for i in range(len(nums)-2):
        left ,right=i+1 , len(nums)-1
        while(left<right):
            current_sum=nums[i]+nums[left]+nums[right]
            if abs(current_sum-target)<abs(closest_sum-target):
                closest_sum=current_sum
            if current_sum<target:
                left+=1
            elif current_sum>target:
                right-=1
            else:
                return current_sum
    return closest_sum

def fourSum(self, nums: List[int], target: int) -> List[List[int]]:
    nums.sort()
    res=[]
    for i in range(len(nums)-3):
        if i>0  and nums[i]==nums[i-1]:
             continue
        for j in range(i+1 , len(nums)-2):
            if j>i+1 and nums[j]==nums[j-1]:
                continue
            left , right=j+1 , len(nums)-1
            while(left<right):
                four_sum=nums[i]+nums[j]+nums[left]+nums[right]
                if four_sum==target:
                    res.append([nums[i] , nums[j] ,nums[left] ,nums[right]])
                    left+=1
                    right-=1
                    while left<right and nums[left]==nums[left-1]:
                        left+=1
                    while left<right and nums[right]==nums[right+1]:
                        right-=1
                elif four_sum<target:
                    left+=1
                else:
                    right-=1

    return res
```

```
def addTwoNumbers(self, l1: Optional[ListNode], l2: Optional[ListNode]) ->
Optional[ListNode]:

        dummeyHead=ListNode(0)
        tail=dummeyHead
        carry=0

        while l1 is not None or l2 is not None or carry!=0:
            digit1=l1.val if l1 is not None else 0
            digit2 =l2.val if l2 is not None else 0

            sum=digit1+digit2+carry
            digit=sum%10
            carry=sum//10

            newNode=ListNode(digit)
            tail.next=newNode
            tail=tail.next

            l1=l1.next if l1  is not  None else None
            l2=l2.next if l2 is not None else None

        result=dummeyHead.next
        dummeyHead.next=None
        return result
```

```
def  mergelist(list 1 l1 ,list l2 ):
    cur = dummy = ListNode()
     while list1 and list2:
        if list1.val < list2.val:
            cur.next = list1
            list1, cur = list1.next, list1
        else:
            cur.next = list2
            list2, cur = list2.next, list2

     if list1 or list2:
        cur.next = list1 if list1 else list2

     return dummy.next
```