1. <https://leetcode.com/problems/count-odd-numbers-in-an-interval-range/submissions/>

Sol:

class Solution:

def countOdds(self, low: int, high: int) -> int:

if low%2==1 and high%2==1:

return (high+1-low)//2+1

else:

return (high+1-low)//2

Screenshot:

A screenshot of a computer

Description automatically generated

1. <https://leetcode.com/problems/two-sum/>

sol:

class Solution:

def twoSum(self, nums: List[int], target: int) -> List[int]:

y = []

c=0

for i in nums:

q = target-i

c+=1

if q in nums[c::]:

if i==q:

for k in range(0,len(nums)):

if i==nums[k]:

y.append(k)

else:

y.append(nums.index(i))

y.append(nums.index(q))

return y

Screenshot:

Graphical user interface, text, application

Description automatically generated

1. <https://leetcode.com/problems/palindrome-number/>

sol:

class Solution:

def isPalindrome(self, x: int) -> bool:

a = [str(b) for b in str(x)]

q= ''.join([str(op) for op in a])

a.reverse()

d = ''.join([str(elem) for elem in a])

if q==d:

return True

else :

return False

Screenshot:

A screenshot of a computer

Description automatically generated

1. <https://leetcode.com/problems/remove-duplicates-from-sorted-array/>

sol:

class Solution:

def removeDuplicates(self, nums):

r = 1

for i in range(1,len(nums)):

if nums[i]!=nums[r-1]:

nums[r]=nums[i]

r+=1

return r

Screenshot:

Graphical user interface, text, application

Description automatically generated

1. <https://leetcode.com/problems/remove-element/>

sol:

class Solution:

def removeElement(self, nums, val):

i,k=0,0

while k<len(nums):

if nums[k] != val:

nums[i] = nums[k]

i+=1

k+=1

return i

Screenshot:

A screenshot of a computer

Description automatically generated