1. def playback\_duration(in\_time,playback\_speed):

time = in\_time.split(":")

time\_in\_secs = (3600\*int(time[0])+60\*int(time[1])+int(time[2]))/playback\_speed

f\_time\_in\_hours = str(int(time\_in\_secs/3600)) if time\_in\_secs > 3600 else '00'

f\_time\_in\_mins = str(int((time\_in\_secs%3600)/60)) if (time\_in\_secs)%3600 > 60 else '00'

f\_time\_in\_secs = str(int((time\_in\_secs%3600)%60)) if ((time\_in\_secs)%3600)%60 >

else '00'

output = f'{f\_time\_in\_hours}:{f\_time\_in\_mins}:{f\_time\_in\_secs}'

print(f'playback\_duration{in\_time, playback\_speed} ➞ {output}')

playback\_duration("00:30:00", 2)

playback\_duration("01:20:00", 1.5)

playback\_duration("51:20:09", 0.5)

Output:

playback\_duration('00:30:00', 2) ➞ 00:15:00

playback\_duration('01:20:00', 1.5) ➞ 00:53:20

playback\_duration('51:20:09', 0.5) ➞ 102:40:18

1. def pile\_of\_cubes(in\_volume):

out\_volume = 0

output = 0

for cube in range(1,in\_volume):

out\_volume += pow(cube,3)

if in\_volume <= out\_volume:

output = cube if in\_volume == out\_volume else None

break

print(f'pile\_of\_cubes({in\_volume}) ➞ {output}')

pile\_of\_cubes(1071225)

pile\_of\_cubes(4183059834009)

pile\_of\_cubes(16)

Output:

pile\_of\_cubes(1071225) ➞ 45

pile\_of\_cubes(4183059834009) ➞ 2022

pile\_of\_cubes(16) ➞ None

1. def find\_fulcrum(in\_list):

output = -1

for ele in in\_list:

index\_of\_ele =in\_list.index(ele)

if sum(in\_list[:index\_of\_ele]) == sum(in\_list[index\_of\_ele+1:]):

output = ele

break

print(f'find\_fulcrum({in\_list}) ➞ {output}')

find\_fulcrum([3, 1, 5, 2, 4, 6, -1])

find\_fulcrum([1, 2, 4, 9, 10, -10, -9, 3])

find\_fulcrum([9, 1, 9])

find\_fulcrum([7, -1, 0, -1, 1, 1, 2, 3])

find\_fulcrum([8, 8, 8, 8])

Output:

find\_fulcrum([3, 1, 5, 2, 4, 6, -1]) ➞ 2

find\_fulcrum([1, 2, 4, 9, 10, -10, -9, 3]) ➞ 4

find\_fulcrum([9, 1, 9]) ➞ 1

find\_fulcrum([7, -1, 0, -1, 1, 1, 2, 3]) ➞ 0

find\_fulcrum([8, 8, 8, 8]) ➞ -1

1. def sock\_merchant(in\_list):

paired\_socks = {}

output = 0

for ele in in\_list:

if ele in paired\_socks:

paired\_socks[ele]+=1

else:

paired\_socks[ele]=1

for pair in paired\_socks.values():

output += pair//2

print(f'sock\_merchant({in\_list}) ➞ {output}')

sock\_merchant([10, 20, 20, 10, 10, 30, 50, 10, 20])

sock\_merchant([50, 20, 30, 90, 30, 20, 50, 20, 90])

sock\_merchant([])

Output:

sock\_merchant([10, 20, 20, 10, 10, 30, 50, 10, 20]) ➞ 3

sock\_merchant([50, 20, 30, 90, 30, 20, 50, 20, 90]) ➞ 4

sock\_merchant([]) ➞ 0

1. import re

def negative\_sum(in\_string):

pattern = '-\d+'

output = sum([int(ele) for ele in re.findall(pattern,in\_string)])

print(f'negative\_sum("{in\_string}") ➞ {output}')

negative\_sum("-12 13%14&-11")

negative\_sum("22 13%14&-11-22 13 12")

Output:

negative\_sum("-12 13%14&-11") ➞ -23

negative\_sum("22 13%14&-11-22 13 12") ➞ -33