1. import operator

ops = {'+':operator.add,'-':operator.sub,'\*':operator.mul,'//':operator.floordiv}

def arithmetic\_operation(in\_string):

in\_list = in\_string.split(" ")

output = ops[in\_list[1]](int(in\_list[0]),int(in\_list[2])) if int(in\_list[2]) != 0 else -1

print(f'arithmetic\_operation({in\_list[0]} {in\_list[1]} {in\_list[2]}) ➞ {output}')

arithmetic\_operation("12 + 12")

arithmetic\_operation("12 - 12")

arithmetic\_operation("12 \* 12")

arithmetic\_operation("12 // 0")

Output:

arithmetic\_operation(12 + 12) ➞ 24

arithmetic\_operation(12 - 12) ➞ 0

arithmetic\_operation(12 \* 12) ➞ 144

arithmetic\_operation(12 // 0) ➞ -1

1. import math

def distance(a,b):

return math.sqrt(pow((b[1]-a[1]),2)+pow((b[0]-a[0]),2))

def perimeter(in\_array):

perimeter = []

for ele in range(len(in\_array)):

if ele == len(in\_array)-1:

perimeter.append(distance(in\_array[ele],in\_array[0]))

else:

perimeter.append(distance(in\_array[ele],in\_array[ele+1]))

print(f'perimeter({in\_array}) ➞ {sum(perimeter):.2f}')

perimeter([[15,7],[5,22],[11,1]])

perimeter([[0,0],[0,1],[1,0]])

perimeter([[-10,-10],[10,10],[-10,10]])

Output:

perimeter([[15, 7], [5, 22], [11, 1]]) ➞ 47.08

perimeter([[0, 0], [0, 1], [1, 0]]) ➞ 3.41

perimeter([[-10, -10], [10, 10], [-10, 10]]) ➞ 68.28

1. def tallest\_skyscraper(in\_list):

out\_list = []

for num in range(len(in\_list)):

count = 0

for ele in range(len(in\_list[num])):

count += in\_list[ele][num]

out\_list.append(count)

print(f'tallest\_skyscraper({in\_list}) ➞ {max(out\_list)}')

tallest\_skyscraper([[0, 0, 0, 0],[0, 1, 0, 0],[0, 1, 1, 0],[1, 1, 1, 1]])

tallest\_skyscraper([[0, 1, 0, 0],[0, 1, 0, 0],[0, 1, 1, 0],[1, 1, 1, 1]])

tallest\_skyscraper([[0, 0, 0, 0],[0, 0, 0, 0],[1, 1, 1, 0],[1, 1, 1, 1]])

Output:

tallest\_skyscraper([[0, 0, 0, 0], [0, 1, 0, 0], [0, 1, 1, 0], [1, 1, 1, 1]]) ➞ 3

tallest\_skyscraper([[0, 1, 0, 0], [0, 1, 0, 0], [0, 1, 1, 0], [1, 1, 1, 1]]) ➞ 4

tallest\_skyscraper([[0, 0, 0, 0], [0, 0, 0, 0], [1, 1, 1, 0], [1, 1, 1, 1]]) ➞ 2

1. def bonus(int\_num):

if int\_num > 48:

output = 0+(8\*325)+(8\*550)+((int\_num-48)\*600)

elif int\_num < 48 and int\_num >= 41:

output = 0+(8\*325)+((int\_num-41+1)\*550)

elif int\_num >33 and int\_num <= 40:

output = 0+((int\_num-33+1)\*325)

else:

output = 0

print(f'bonus({int\_num}) ➞ {output}')

bonus(15)

bonus(44)

bonus(37)

bonus(50)

bonus(60)

Output:

bonus(15) ➞ 0

bonus(44) ➞ 4800

bonus(37) ➞ 1625

bonus(50) ➞ 8200

bonus(60) ➞ 14200

1. def is\_disarium(in\_num):

sum = 0

output = False

for ele in range(len(str(in\_num))):

sum += int(str(in\_num)[ele])\*\*(ele+1)

if in\_num == sum:

output=True

print(f'is\_disarium({in\_num}) ➞ {output}')

is\_disarium(75)

is\_disarium(135)

is\_disarium(544)

is\_disarium(518)

is\_disarium(466)

is\_disarium(8)

Output:

is\_disarium(75) ➞ False

is\_disarium(135) ➞ True

is\_disarium(544) ➞ False

is\_disarium(518) ➞ True

is\_disarium(466) ➞ False

is\_disarium(8) ➞ True