1. def meme\_sum(a,b):

a,b = str(a),str(b)

output = ''

while len(a) != len(b):

if len(a) < len(b):

a ='0'+a

else:

b='0'+b

for ele in range(len(a)):

output += str(int(a[ele])+int(b[ele]))

print(f'meme\_sum{a,b} ➞ {output}')

meme\_sum(26, 39)

meme\_sum(122, 81)

meme\_sum(1222, 30277)

Output:

meme\_sum('26', '39') ➞ 515

meme\_sum('122', '081') ➞ 1103

meme\_sum('01222', '30277') ➞ 31499

1. def next\_prime(in\_num):

in\_num\_clone = in\_num

while True:

if (in\_num-1)%6 == 0 or (in\_num+1)%6 ==0 :

temp = in\_num

for ele in range(3,in\_num+1):

if in\_num%ele == 0 and ele != in\_num:

in\_num = in\_num+1

break

if temp == in\_num:

break

else:

in\_num += 1

print(f'next\_prine({in\_num\_clone}) ➞ {in\_num}')

next\_prime(12)

next\_prime(24)

next\_prime(11)

Output:

next\_prine(12) ➞ 13

next\_prine(24) ➞ 29

next\_prine(11) ➞ 11

1. def ave\_spd(up\_time,up\_speed,down\_speed):

distance = up\_speed\*(up\_time/60)

down\_time = distance/down\_speed

output = (2\*distance)/((up\_time/60)+down\_time)

print(f'ave\_spd{up\_time,up\_speed,down\_speed} ➞ {int(output)}')

ave\_spd(18, 20, 60)

ave\_spd(30, 10, 30)

ave\_spd(30, 8, 24)

Output:

ave\_spd(18, 20, 60) ➞ 30

ave\_spd(30, 10, 30) ➞ 15

ave\_spd(30, 8, 24) ➞ 12

1. def kempner(in\_num):

def factorial(in\_num):

if in\_num == 1:

return 1

else:

return in\_num \* factorial(in\_num-1)

for ele in range(1,in\_num+1):

if factorial(ele)%in\_num == 0:

output = ele

break

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

kempner(6)

kempner(10)

kempner(5)

kempner(2)

Output:

kempner(6) ➞ 3

kempner(10) ➞ 5

kempner(5) ➞ 5

kempner(2) ➞ 2

1. def boxes(in\_list):

in\_list\_clone = in\_list.copy()

output = []

temp\_box = []

while True:

if len(in\_list) != 0:

if sum(temp\_box) <= 10:

temp\_box.append(in\_list.pop(0))

else:

in\_list.insert(0,temp\_box.pop())

output.append(temp\_box)

temp\_box = []

else:

output.append(temp\_box)

temp\_box = []

break

print(f'boxes({in\_list\_clone}) ➞ {output} ➞ {len(output)}')

boxes([2, 1, 2, 5, 4, 3, 6, 1, 1, 9, 3, 2])

boxes([5, 5, 5, 5, 5, 5, 2, 3, 4, 5, 6])

Output:

boxes([2, 1, 2, 5, 4, 3, 6, 1, 1, 9, 3, 2]) ➞ [[2, 1, 2, 5], [4, 3], [6, 1, 1], [9], [3, 2]] ➞ 5

boxes([5, 5, 5, 5, 5, 5, 2, 3, 4, 5, 6]) ➞ [[5, 5], [5, 5], [5, 5], [2, 3, 4], [5, 6]] ➞ 5