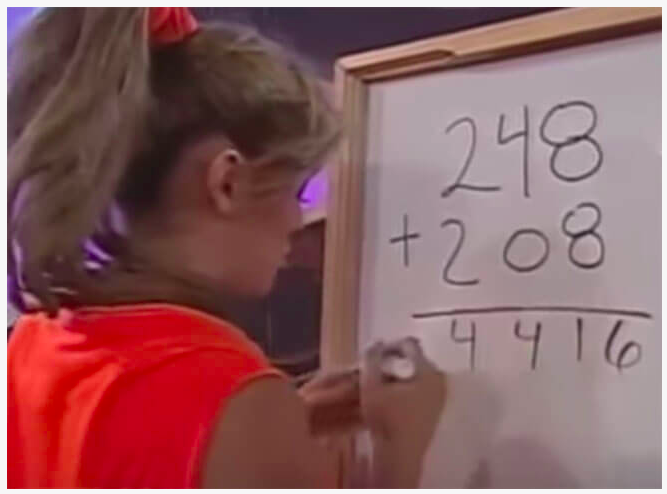
1. For this challenge, forget how to add two numbers together. The best explanation on what to do for this function is this meme:



**Examples**

meme\_sum(26, 39) ➞ 515

# 2+3 = 5, 6+9 = 15

# 26 + 39 = 515

meme\_sum(122, 81) ➞ 1103

# 1+0 = 1, 2+8 = 10, 2+1 = 3

# 122 + 81 = 1103

meme\_sum(1222, 30277) ➞ 31499

**Ans:**

def meme\_sum(a, b):

x=str(a)

y=str(b)

result=""

if len(x)>len(y):

y=y.zfill(len(x))

else:

x=x.zfill(len(y))

for i in range(len(x)):

result+=str(int(x[i])+int(y[i]))

print(result)

2. Given an integer, create a function that returns the next prime. If the number is prime, return the number itself.

**Examples**

next\_prime(12) ➞ 13

next\_prime(24) ➞ 29

next\_prime(11) ➞ 11

# 11 is a prime, so we return the number itself.

**Ans:**

def next\_prime(num):

while True:

found=True

for i in range(2, (num//2)+1):

if num%i==0:

found=False

break

if found:

print(num)

break

else:

num+=1

3. If a person traveled up a hill for 18mins at 20mph and then traveled back down the same path at 60mph then their average speed traveled was 30mph.

Write a function that returns the average speed traveled given an uphill time, uphill rate and a downhill rate. Uphill time is given in minutes. Return the rate as an integer (mph). No rounding is necessary.

**Examples**

ave\_spd(18, 20, 60) ➞ 30

ave\_spd(30, 10, 30) ➞ 15

ave\_spd(30, 8, 24) ➞ 12

**Ans:**

def ave\_spd(t1, s1, s2):

d=s1\*(t1/60)

t2=(d/s2)

out=(2\*d)/((t1/60)+t2)

print(int(out))

4. The Kempner Function, applied to a composite number, permits to find the smallest integer greater than zero whose factorial is exactly divided by the number.

kempner(6) ➞ 3

1! = 1 % 6 > 0

2! = 2 % 6 > 0

3! = 6 % 6 === 0

kempner(10) ➞ 5

1! = 1 % 10 > 0

2! = 2 % 10 > 0

3! = 6 % 10 > 0

4! = 24 % 10 > 0

5! = 120 % 10 === 0

A Kempner Function applied to a prime will always return the prime itself.

kempner(2) ➞ 2

kempner(5) ➞ 5

Given an integer n, implement a Kempner Function.

**Examples**

kempner(6) ➞ 3

kempner(10) ➞ 5

kempner(2) ➞ 2

**Ans:**

def kempner(num):

for i in range(1, num+1):

if fact(i)%num==0:

print(i)

break

def fact(n):

fac=1

for i in range(1, n+1):

fac\*=i

return fac

5. You work in a factory, and your job is to take items from a conveyor belt and pack them into boxes. Each box can hold a maximum of 10 kgs. Given a list containing the weight (in kg) of each item, how many boxes would you need to pack all of the items?

**Example**

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

# Box 1 = [2, 1, 2, 5] (10kg)

# Box 2 = [4, 3] (7kg)

# Box 3 = [6, 1, 1] (8kg)

# Box 4 = [9] (9kg)

# Box 5 = [3, 2] (5kg)

**Ans:**

def boxes(lst):

dic={}

l=[]

sum=0

temp\_sum=0

count=1

for i in lst:

key="# Box "+str(count)

temp\_sum+=int(i)

if temp\_sum<=10:

key="# Box "+str(count)

sum+=int(i)

l.append(i)

dic[key]="{0} ({1}kg)".format(l, sum)

else:

count+=1

temp\_sum=0

sum=0

l=[]

temp\_sum+=int(i)

key="# Box "+str(count)

sum+=int(i)

l.append(i)

dic[key]=dic[key]="{0} ({1}kg)".format(l, sum)

for key in dic.keys():

print("{0} = {1}".format(key, dic[key]))