*#2/11/2022*

*# # #FOUR FUNCTION PROBLEM  
# # def add(a,b): # a,b are parameters  
# # print(a+b)  
# # def sub(a,b):  
# # print(a-b)  
# # def pro(a,b):  
# # print(a\*b)  
# # def modu(a,b):  
# # print(a%b)  
# # x=int(input("Enter First Number: "))  
# # y=int(input("Enter Second Number: ")) #x,y are arguements  
#  
# add(x,y)  
# sub(x,y)  
# pro(x,y)  
# modu(x,y)*

#**def** sub(a,b):  
 **return** a-b  
x=int(input(**""**))  
y=int(input(**""**)) Calling inside print  
print(sub(x,y))

#FACTORIAL AND MULTIPLE FUNCTION

**def** ques(a):  
 i = 1  
 fact = 1  
 **if** a == 0:  
 print(**"Factorial = 1"**)  
 **elif** a < 0:  
 print(**"Factorial of negative value not possible"**)  
 **else**:  
 **while** i <= a:  
 fact = fact \* i  
 i = i + 1  
 print(**"Factorial is :"**, fact)  
 **if** fact % 3 == 0:  
 print(**"Multiple of 3"**)  
 **else**:  
 print(**"Not a Multiple of 3"**)  
x=int(input(**""**))  
ques(x)

#FUNCTIONS TO COUNT THE DIGITS:

**def** digi(a):  
 a=str(a)  
 b=len(a)  
 **return** b  
x=int(input(**"Enter a digit: "**))  
r1=digi(x)  
print(**"Count of digits:"**,r1)

#RETURNING MULTIPLE VALUES:

**def** example(a,b,c):  
 **return** a+1,b+1,c+1  
r1,r2,r3=example(1,2,3)  
print(r1,r2,r3)

print(example(1,2,3))

#COUNTING VOWELS AND SPACES:

**def** vowels(a):  
 vcount=0  
 scount=0  
 **for** i **in** a:  
 **if** i **in "aeiouAEIOU"**:  
 vcount+=1  
 **elif** i==**" "**:  
 scount+=1  
 **return** vcount,scount  
r1,r2=vowels(**"Rahul Roy"**)  
print(r1,r2)

#ENTER NUMBERS AND FIND AVERAGE:

**def** example(\*args):  
 sum=0.0  
 a=len(args)  
 **for** i **in** args:  
 sum=sum+i  
 avg=sum/a  
 print(**"Average ="**,avg)  
s11,s12,s13=eval(input(**"Enter Marks of first Section:"**))  
s21,s22,s23,s24,s25=int(input(**"Enter Marks of second Section:"**))  
s31,s32,s33,s34,s35,s36,s37=int(input(**"Enter Marks of third Section:"**))  
example(s11,s12,s13)  
example(s21,s22,s23,s24,s25)  
example(s31,s32,s33,s34,s35,s36,s37)

#PRINTING ONLY STARTING WITH A PARTICULAR ALPHABET:

**def** example(\*\*kwargs):  
 **for** i,j **in** kwargs.items():  
 **if** j[0]==**"a"**:  
 print(i,**":"**,j,end=**" "**)  
 print()  
example(a=**"pqr"**,b=**"stu"**)  
example(a=**"aaa"**,b=**"bbb"**,c=**"ccc"**,d=**"ddd"**)  
example(a=**"abc"**,b=**"def"**,c=**"ghi"**)

#PRINTING ENTIRE ARGUMENTS:

**def** example(\*\*kwargs):  
 print(kwargs)  
example(a=**"pqr"**,b=**"abc"**)  
example(a=**"aaa"**,b=**"bbb"**,c=**"ccc"**,d=**"dddd"**)  
example(a=**"abc"**,b=**"def"**,c=**"ghi"**)

#DOCSTRING:

**def** abc():  
 *"""Function is defined to print string abc"""* print(**"abc"**)  
**def** pqr(a,b):  
 *"""Function is defined to print the product of parameters passed  
 a -- is the first parameter  
 b -- is the second parameter  
 """* print(a\*b)  
print(abc.\_\_doc\_\_)  
print(pqr.\_\_doc\_\_)

#RETURN FUNCTION:

*# def display():  
# print("Hello") even though there is no return yet python returns none internally  
# print(display())*

**def** example1(a,b):  
 **return** a\*b  
print(example1(2,3))  
**def** example2():  
 **return** 5  
print(example2())  
**def** example3(a,b):  
 **return** a,b  
print(example3(5,6))  
**def** example4():  
 print(**"Hello"**)  
 **return "Python"** print(**"world"**)  
print(example4())

#addavg,sub,mul:

**def** addavg(a,b):  
 sum=a+b  
 avg=sum/2  
 **return** sum,avg  
**def** sub(a,b):  
 sb=a-b  
 **return** sb  
**def** mul(a,b):  
 mu=a\*b  
 **return** mu  
x=int(input(**"Enter a number: "**))  
y=int(input(**"Enter another number: "**))  
r1,r2=addavg(x,y)  
print(**"sum, average: ({}, {})"**.format(r1,r2))  
print(**"subtraction:"**,sub(x,y))  
print(**"multiplication:"**,mul(x,y))

#GREETINGS PROGRAM FUNCTION:

**def** sayhello(username):  
 greet=**"Hello"** print(greet+**" "**+username)  
users=[**"Ram"**,**"Mahesh"**,**"Vasudha"**,**"Uma"**,**"Sekhar"**,**"John"**]  
**for** i **in** users:  
 sayhello(i)

#SIMPLE CALCULATOR:

**def** simplecal(a,b):  
 su=a+b  
 sub=a-b  
 mul=a\*b  
 print(**"addition:"**,su)  
 print(**"subtraction:"**,sub)  
 print(**"multiplicatin:"**,mul)  
  
simplecal(a=3,b=5)  
simplecal(b=4,a=5)  
simplecal(8,4)

#STRING MESSAGE CONCAT:

**def** saysomething(name,message):  
 b=**"Good"** b=b+**" "**+message  
 print(b,name)  
x=str(input(**"Enter the name: "**))  
y=str(input(**"Enter your message: "**))  
saysomething(name=x,message=y)  
saysomething(message=y,name=x)

#DEFAULT ARGUMENTS:

**def** default1(a,b,c=12):  
 print(a+b+c)  
default1(1,2)  
default1(1,2,3)

We can give multiple default arguments and even give nothing during execution

As the number of default arguments increase, our permutations will also increase

**def** simplecal(a,b=100):  
 ad=a+b  
 sub=a-b  
 mul=a\*b  
 print(**"addition: "**,ad)  
 print(**"subtraction: "**,sub)  
 print(**"multiplication: "**,mul)  
num1=int(input(**"a: "**))  
num2=int(input(**"b: "**))  
simplecal(a=num1)  
simplecal(b=num2,a=num1)

**def** calculateTax(salary,percent=20):  
 print(salary\*percent/100)  
x=int(input(**"Enter salary: "**))  
y=float(input(**"Enter Percentage: "**))  
calculateTax(salary=x)  
calculateTax(salary=x,percent=y)

#Enter arguments as many as you want and print sum:

**def** mysum(\*args):  
 su=0  
 **for** i **in** args:  
 su=su+i  
 **return**(su)  
print(mysum(1,2,3,4,5,6,7))  
print(mysum(1,2))  
print(mysum(1,2,3))

#LARGEST NUMBER (ANY NUMBER OF INPUTS):

**def** largestnumbers(\*numbers):  
 m=numbers[0]  
 **for** num **in** numbers:  
 **if** num>m:  
 m=num  
 print(**"largest:"**,m)  
  
largestnumbers(1,2,6,9,5)

LESSON 4 (UNIT 3)

x=**lambda** a: a+10  
print(x(5)) LAMBDA/ANONYMOUS FUNCTION

x=**lambda** r: 3.14\*r\*r  
print(x(2.3)) area of circle (lambda)

tax= **lambda** m1: m1\*20/100 salary lambda function  
salary=int(input(**"Please enter your salary: "**))  
print(**"Tax to be paid: "**,tax(salary))

doublenum=**lambda** x:x\*2  
num=int(input(**"Enter a number: "**)) double number   
print(**"Doubled number: "**,doublenum(num))

MAP FUNCTION:

**def** addition(n):  
 **return** n+n  
numbers=(1,2,3,4,5)  
result=map(addition,numbers)  
print(list(result)) *#or use for loop: for i in result:  
 #print(i)*

**def** area(n):  
 **return** n\*n\*3.14  
radius=(1,2,3,4,5) area of circle using mapping  
result=map(area,radius)  
print(list(result))

radius=(1,2,3,4,5,99)  
result=map(**lambda** x:x\*x\*3.14,radius)

area of circle using mapping **lambda**print(list(result))

number1=[1,2,3]  
number2=[4,5,6] two iterations  
result = map (**lambda** x,y:x+y,number1,number2)  
print(list(result))  
  
p=[100,200,300]  
r=[3,6,9]  
t=[6,7,8 simple interest(3 iterations)  
result=map(**lambda** p,r,t:p\*r\*t/100,p,r,t)  
print(list(result))

**def** fun(variable):  
 letters=[**"a"**,**"e"**,**"i"**,**"o"**,**"u"**]  
 **if** (variable **in** letters):  
 **return True  
 else**: vowel check  
 **return False**sequence=[**"g"**,**"g"**,**"r"**,**"a"**,**"h"**,**"u"**,**"l"**,**"r"**,**"o"**,**"y"**]  
filtered = filter(fun,sequence)  
print(list(filtered))

**def** even(x):  
 **if** x%2==0:  
 **return True  
 else**: filter **for** even number  
 **return False**num=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]  
eve=filter(even,num)  
print(list(eve))

seq=[0,1,2,3,4,5,6,7,8,9]

result=filter(**lambda** x: x%2!=0,seq)  
print(list(result)) 2 filters   
result=filter(**lambda** x: x%2==0,seq)  
print(list(result))

*LIST COMPREHENSION:*l1=[1,2,3,44,55]  
l2=[x\*x **for** x **in** l1]  
print(l2)

fruits=[**"apple"**,**"orange"**,**"kiwi"**,**"mango"**]  
newlist=[x **for** x **in** fruits **if "a" in** x] **"a"** check  
print(newlist)  
  
l1=[1,2,3,4] even number square  
l2=[x\*x **for** x **in** l1 **if** x%2==0]  
print(l2)

COMMON NUMBER CHECK USING FILTER AND COMPREHENSION

a=[1,2,3,5,7,9]  
b=[2,3,6,7,9,8]  
print(list(filter(**lambda** x: x **in** a,b)))  
print([x **for** x **in** a **if** x **in** b])

GLOBAL VARIABLE :

globvar=**"Hello"  
def** test1():  
 **global** globvar  
 globvar=**"Good Morning"  
  
def** test2():  
 globvar=**"Good Morning"**print(globvar)  
test1()  
test2()  
print(globvar)

x=10  
**def** example():  
 x=20  
**def** example2():  
 **global** x  
 x=x+1  
**def** example3():  
 **global** x  
 x+=2  
example()  
example2()  
example3()  
print(x)

a=int(input(**"a: "**))  
**def** changeglobal():  
 **global** a  
 a=200  
**def** changelocal():  
 a=500  
 print(a)  
print(a)  
changeglobal()  
changelocal()  
print(a)

FUNCTION COMPOSITION:

**def** square(x):  
 x=x\*\*2  
 **return** x  
**def** double(x): FUNTION COMPOSITION  
 x=x\*2 doubling the value **and   
 return** x printing square  
a=int(input(**""**))  
print(square(double(a)))

**def** square(x):  
 x=x\*\*2  
 **return** x  
**def** double(x):  
 x=x\*2 checking the square of a number is divisible by   
 **return** x 5 or not  
**def** multiple(x):  
 **if** x%5==0:  
 **return True  
 else**:  
 **return False**a=int(input(**""**))  
print(square(double(a)))  
print(multiple(square(double(a))))

**def** compose(\*functions):  
 **def** inner(arg):  
 **for** f **in** reversed(functions):  
 arg=f(arg)  
 **return** arg  
 **return** inner  
  
**def** square(x):  
 **return** x\*\*2  
**def** increment(x):  
 **return** x+1  
**def** half(x):  
 **return** x/2  
a=int(input(**"Number: "**))  
composed=compose(square,increment,half)  
print(composed(a))

TYPICAL CODE

**def** compose(\*functions):  
 **def** inner(arg):  
 **for** f **in** reversed(functions):  
 arg=f(arg)  
 **return** arg  
 **return** inner

**def** square(x):  
 **return** x\*\*2  
**def** increment(x):  
 **return** x+1  
**def** half(x):  
 **return** x/2  
a=int(input(**"Number: "**))  
composed=compose(square,increment,half)  
print(composed(a))  
composed=compose(square,increment)  
print(composed(a))

RECURSION:

**def** sum(x):  
 **if** x==1 **or** x==0:  
 **return** x  
 **else**: sum till nth number  
 **return** x+sum(x-1)  
n=int(input(**"Enter n: "**))  
**if** n<0:  
 print(**"Enter a positive number"**)  
**else**:  
 print(**"Sum is : "**,sum(n))

**def** reverse(n,x):  
 **if** (n==0):  
 **return** x  
 **else**: reverse of a number  
 x=(x\*10)+(n%10)  
 **return** reverse(n//10,x)  
num=int(input(**""**))  
print(reverse(num,0))