=>Using single decorator

def decorator(pattern):

def wrapper():

pattern()

print("Done")

return wrapper

@decorator

def pattern():

print("\* \* \* \* \*")

print("% % % % %")

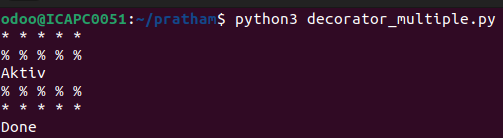
s="Aktiv"

print(s)

print("% % % % %")

print("\* \* \* \* \*")

pattern()



=>Using Multiple decorator

def decorator1(pattern):

def wrapper():

result = pattern()

return result

return wrapper

def decorator2(pattern):

def wrapper(\*args):

pattern(\*args)

print(args[0])

pattern(\*args)

return wrapper

@decorator1

def static\_string():

stringg = "aktiv"

return stringg

@decorator2

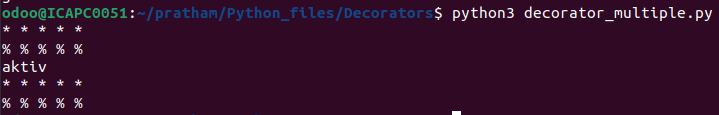
def pattern(stringg):

print("\* \* \* \* \*")

print("% % % % %")

stringg = user\_input()

pattern(stringg)



=>Factorial using Recursion and decorators  
  
def decorator(factorial):

def wrapper(n):

if n < 0:

return "Does not exist"

return factorial(n)

return wrapper

@decorator

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n - 1)

n = int(input("Enter a number:"))

print("Facorial number:",factorial(n))

