Tasks: Implement python grenerator and de coratons; Aim! Write a python program to implement python generator and de coraton. 8.1 write a python program that includes a generodor function to produce a sequence of numbers. with (101t, end and stop a)produce a sequence of numbers when provided b) produce a default sequence of number starting from a ending attorna with a step of 1 if no value are provided. Algorithm: 1. Define generator Function: * Define the function number - sequence (start, end, step=1) 2. Initialize Crument value: * set (unicide to the value of start. 3. Generate Sequence: x white current is less than or equal-to end. * Vield the current value of current. * Increment current by step. Ur Get wer Input: -* Read the starting number (start) from user input. * fead the ending number(end) from user input. or head the step value (step) from when input. 5-create Generator object: -* (reat e a generated object by calling number - sequence (store, end, stop) with mer-provided values. 6. Print Generated sequence: * Interate over the values produced by the generator object. & print each value. def number - sequence (start, end, step=1): 8.1 program; current = start while current 2= end : Yeild current current + = step start = int (input ("Enter the starting number:")) end = int (mput ("Enter the ending number:")) Step=int (input ("Enver the Step valuel:")) sequence - generator = number - sequence (start, end, step) for number in sequence generator: Print Lhamber) Manithm: * Define the function my generator (n) that takes a parometer n. 1. Start function: 2. Initialize counter: * set value to o. 3 Generate values: * while value is less than n:

ou Pit. End or the staneing number: 1 Enter the ending number:50 Enter the step value : I Mar may 6 March 1 and the second of the 16 21 26 31 36 4 46 or the state of th Secretarian Secretarian the angle of the second of the second

* yield the current value .. * Incrementualite by 1 d. Create Generator object: *(all my - generator (11) to create a generator object. 5. Iterate and print values: * For each value produced by the generator object: * print value. 8.1(P) bidiam; def my_generator(n): value = 0 while valuezn: yield value valuet = 1 for value in my_ generator (3): print (value). 8.2 Imagine you are working on a messaging application that needs to format message differently based on the wer's preferences. Algorithm: 1. Creat e Decorators: -*Define uppercase_decorator to convert the result of a function to uppercase. *Define lower case_decorator to convert the result of a function to convert ask. 2. Define functions: -* Define shout function to return the input text-apply @ uppercase_decorationto * Define whisper function to return to return the input text. Apply @ lower ax_ decorated to this function. 3. Define Greet function:-* Define greet function that: * Accepts a function (func) as input * call this function with the text "Hi, I am created by a function passed as an argument! & print the result 4. Execute the Program:-+ coll greet (shout) to print the greeting in upper case. *(all greet Luhisper)-to print the greating in lower case. Program! def upperlase _ decorator(func). def wrapper(text): return func (text).upper() return warpper def lawrease _d econoxon (funi): def wapper (text):

output: Migration and a way to the second and and the second of the property of the second the of the process of mostly points or requests the model that the points of THE THE SEA SEA OF THE all times a leave with to the state you former as off with we are may be recome from the time out to other qu CARLOLAND OF THE COMPANY BUILD OF MAKE STORY " granger of and are \$5: 1/0

MINIT AM CREATED BY A PUNCTION AS AN ARGUMENT, hi, iam created by a function passed as an argument. man was no registed to become and burners on the Consolly prior on the me sufficiency THE PROPERTY OF THE PROPERTY O 新日子をよる。2017年 日 日の新日本・新信仰である。2017年 and the contract of the state of the state of the state of the contract of the state of the stat was a second of the second of er and a part of the ensy that I am Consequences of the sequences FIRST VITES

Teturn wrappe?

Qupper case _ decoration

de 1 Shout (text):

return wrapper

Quoter case _ decorator

de 1 whisper (text):

return text

de 1 greet (func):

greeting = func ("Hi, I am created by a function provint (greeting))

meturn func (text) fower()

del greet (func):

greeting = func ("Hi, 1 am created by a function passed as an argument")

print (greeting)

greet (shout)

greet (whisper)

EX No.	CH
	8
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	3
VIVA VOCE (5)	
RECORD (5))
TOTAL (28)	
SIGN WITH DATE	

Result: Thus the python program to implement python generator and decorators was successfully executed and the output was verified.