Output:-

Enter the Starting number: 1

Enter the ending number : 50

Enter the Step value: 5

1

6

11

16

21

26

31

36

41

46

17/9/25

Task 8. Implement Python genarator and decorators

Aim:-

Write a python program to Implement python generator and decorators

Q1. Write a program that includes a generator function to produce a Sequence of number. The generator should be able to:

a. Produce a Sequence of number when provided with start, end, and Step values.

Program.

```
def number_Sequence (start, end, step=1):
    Current = start
    while Current <= end
     Yeild Current
     Current += step
  Start = int (input ("Enter the starting number:"))
  end = int (input ("Enter the ending number:"))
  Step = int (input ("Enter the step value:"))
# Create the generator
Sequence_generator = number_Sequence (start, end, Step)
#print the generated Sequence of numbers
for number in Sequance_generator
  Print (number)
```

8.8 Imagine you are working on a messaging application that need to format messages differently;

Algorithm:-

1. Create Decorators
2. Define functions
3. Define Greet function
4. Execute the Program.

Program:-

```
def uppercase_decorator(func):
    def wrapper(text):
        return func(text).upper()
    return wrapper
def lowercase_decorate(func):
 def wrapper(text):
    return func(text).lower()
    return wrapper

@uppercase_decorator
def shout(text):
  return text

@lowercase_decorator
def whisper(text):
  return text
```

Output:-

HI, I AM CREATED

hi, i am created

```
def greet (func):
    greeting = func ("Hi I am Created by a function passed as
                an argument.")
    Print (greeting)
    greet (shout)
    greet (whisper.
```

VEL TECH	
EX No.	8
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	
TOTAL (20)	13
SIGN WITH DATE	16/10/25

Result :-

Thus, the python ~~do~~ program to Implement python generator and decorators was successfully executed and the output was verified.

9/6/10/25