TASK 8 Implement python generator and decorators.

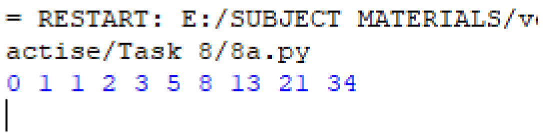
Problem 1:

Pranav wants to print Fibonacci series for first 25 numbers. Help pranav to generate Fibonacci series using **generato**r function in python.

PROGRAM

```
def fibonacci():
    a, b = 0, 1
    for _ in range(10): # Generate Fibonacci numbers using a generator function
        yield a
        a, b = b, a + b
fibonacci_series = fibonacci() # Create a generator object for Fibonacci series
for number in fibonacci_series: # Print the first 10 Fibonacci numbers
    print(number, end=" ")
```

OUTPUT



```
= RESTART: E:/SUBJECT MATERIALS/v
actise/Task 8/8a.py
0 1 1 2 3 5 8 13 21 34
```

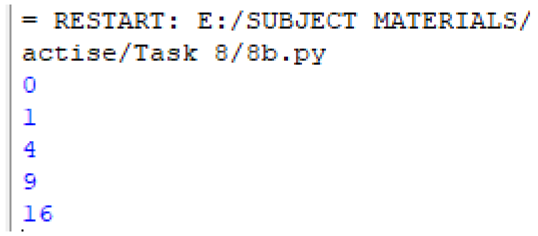
Problem 2:

Rohan wants to print the squares of the numbers 0 through 4. create the generator object that will produce the squares of the numbers 0 through 4 when iterated over.

PROGRAM

```
def squares():
        for i in range(5): # Generate squares of numbers 0 through 4
        yield i ** 2

# Create a generator object for squares of numbers 0 through 4
squares_generator = squares()

# Iterate over the generator object and print the squares
for square in squares_generator:
    print(square)
```

OUTPUT



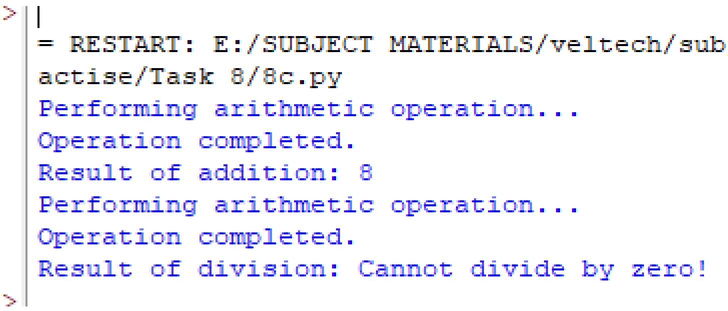
```
= RESTART: E:/SUBJECT MATERIALS/
actise/Task 8/8b.py
0
1
4
9
16
```

Problem 3:
Ms.Priya wants to decorate arithmetic operations. Help Ms.Priya to decorate arithmetic operations using **decorators** in python.

PROGRAM

```
# Decorator function to decorate arithmetic operations

def operation_decorator(func):
    def wrapper(x, y):
        print("Performing arithmetic operation...")
        result = func(x, y)
        print("Operation completed.")
        return result
    return wrapper

# Applying decorator to arithmetic operations

@operation_decorator
def add(x, y):
    return x + y

@operation_decorator
def divide(x, y):
    if y == 0:
        return "Cannot divide by zero!"
    else:
        return x / y

result_add = add(5, 3)
print("Result of addition:", result_add)

result_divide = divide(15, 0)
print("Result of division:", result_divide)
```

OUTPUT



```
>
= RESTART: E:/SUBJECT MATERIALS/veltech/subjects/WS 23-24/python
actise/Task 8/8c.py
Performing arithmetic operation...
Operation completed.
Result of addition: 8
Performing arithmetic operation...
Operation completed.
Result of division: Cannot divide by zero!
>
```