Experiment – 07

(Functions)

1. Find mean, median, mode for the given set of numbers in a list, using user defined function in python.
2. Write a function cumulative product to compute cumulative product of a list of numbers.
3. Write function to compute GCD, LCM of two numbers. Each function shouldn't exceed one line.

Experiment – 08

(Files Operations)

1. Write a program to print each line of a file in reverse order.
2. Write a program to compute the number of characters, words and lines in a file.

Answers

Experiment – 7

```
def calc_stats (numbers):

mean = sum(numbers)/ len (numbers)

sort_num = sorted(numbers)

length = len (sort_num)

if length % 2! =0:

median = sort_num[length//2]

else:

median = (sort_num [length//2] + sort_num [length//2-1])/2

mode = max(set(numbers), key = numbers. count)

return mean, median, mode

numbers = [1,2,5,4,7,8,8,9,9,9]

mean, median, mode = calc_stats (numbers)

print (mean, median, mode)
```

(or)

```
import statistics

def calculate_statistics(numbers):

mean = sum(numbers) / len(numbers)

median = statistics. median (numbers)

mode = statistics. mode (numbers)

return mean, median, mode

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 10]

mean, median, mode = calculate_statistics(numbers)

print ("Mean:", mean)

print ("Median:", median)

print ("Mode:", mode)

def cumulative_product (numbers):

result = [numbers [0]]

for num in numbers [1:]:

result. append (result[-1] * num)
```

```
return result numbers = [1,2,4]

cumulative_product(numbers)

gcd = lambda a, b: a if b == 0 else gcd (b, a % b)

lcm = lambda a, b: abs (a * b) // gcd (a, b)

num1 = 12

num2 = 18

print ("GCD:", gcd (num1, num2))

print ("LCM:", lcm (num1, num2))
```