Python – Numbers

Purpose

This lab was designed to teach you how to use iteration to implement basic algorithms to solve various mathematical functions.

Description

Implement the 10 functions in numbers.py. You're not allowed to use Strings or any of Python's built-in functions. For example, you must implement the power(base, exponent) function and you are not allowed to use ** or call the function from the math module.

Program Shell

numbers.py is provided

```
--------------------------------------------------------------------------------
Testing digits(num) function!

1 2 3 4 5 6 7 8 9

2 4 6 8 1 0

1 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6 7 7 7 8 8 9 9 9

--------------------------------------------------------------------------------
Testing gcd(a, b) function!

gcd(10, 5)    = 5
gcd(128, 96)  = 32
gcd(90, 20)   = 10
gcd(1203, 18) = 3
gcd(-13, 48)  = 1
gcd(50, 75)   = 25

--------------------------------------------------------------------------------
Testing lcm(a, b) function!

lcm(10, 5)    = 10
lcm(128, 96)  = 384
lcm(90, 20)   = 180
lcm(1203, 18) = 7218
lcm(-13, 48)  = 624
lcm(50, 75)   = 150

--------------------------------------------------------------------------------
Testing is_prime(num) function!

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

--------------------------------------------------------------------------------
```

```
Testing print_4_perfect_numbers(n) function!

6 28 496 8128

--------------------------------------------------------------------------------
Testing is_odious(num) function!

1 2 4 7 8 11 13 14 16 19 21 22 25 26 28 31 32 35 37 38

--------------------------------------------------------------------------------
Testing is_evil(num) function!

0 3 5 6 9 10 12 15 17 18 20 23 24 27 29 30 33 34 36 39

--------------------------------------------------------------------------------
Testing power(base, exp) function!

power(3, 5)  = 243
power(2, 10) = 1024
power(4, -3) = 0.015625
power(7, -2) = 0.02040816326530612
power(8, 5)  = 32768
power(10, 9) = 1000000000
power(2, 32) = 4294967296
power(2, 64) = 18446744073709551616
--------------------------------------------------------------------------------
Testing prime_factorization(num) function!

 2 == 2
 3 == 3
 4 == 2^2
 5 == 5
 6 == 2*3
 7 == 7
 8 == 2^3
 9 == 3^2
10 == 2*5
11 == 11
12 == 2^2*3
13 == 13
14 == 2*7
15 == 3*5
16 == 2^4
17 == 17
18 == 2*3^2
19 == 19
20 == 2^2*5
21 == 3*7
22 == 2*11
23 == 23
24 == 2^3*3
25 == 5^2
26 == 2*13
27 == 3^3
28 == 2^2*7
29 == 29
30 == 2*3*5
31 == 31
32 == 2^5
33 == 3*11
34 == 2*17
35 == 5*7
```

```
36 == 2^2*3^2
37 == 37
38 == 2*19
39 == 3*13
40 == 2^3*5
41 == 41
42 == 2*3*7
43 == 43
44 == 2^2*11
45 == 3^2*5
46 == 2*23
47 == 47
48 == 2^4*3
49 == 7^2
google == 2^100*5^100
```