

Factor Optimizations, Primes, LCM and GCD

January 25, 2023

1 Factor Count of a given number

- You will be given a positive integer n , you have to find out number of factors n is having.

```
[14]: %%time
n = int(input()) # 12
fc = 1
for i in range(1, n // 2 + 1): # 100000 500000
    if n % i == 0:
        fc += 1
print(fc)
```

1000000000

81

CPU times: total: 2.59 s

Wall time: 4.52 s

```
[26]: # Square root logic to find out the factors of a given number
n = int(input()) # 12
fc = 1
for i in range(2, int(n ** 0.5) + 1):
    if n % i == 0:
        if i == n//i:
            fc += 1
        else:
            fc += 2
print(fc)
```

17

1

```
[18]: 1000000000 ** 0.5
```

```
[18]: 10000.0
```

```
[ ]: 1 2 4 5 10 20 25 50 100
```

```
[37]: # Square root logic to find out the factors of a given number
n = int(input()) # 12
```

```

fc = 1
for i in range(2, int(n ** 0.5) + 1):
    if n % i == 0:
        if i == n//i:
            fc += 1
        else:
            fc += 2
if fc == 1:
    print('Prime')
else:
    print('Not a Prime')

```

1000000000000000000
Not a Prime

```

[ ]: 27 --> 1 3 9 27
      1 to 5
      1 3
      48
      55
      32
      63

```

```

[ ]: 100000000000
      100000
      1 100000
      1
      2
      3

```

```

[41]: # Prime
n = int(input())
is_prime = True
for i in range(2, int(n ** 0.5) + 1): # 1 2 18
    if n % i == 0:
        is_prime = False
        break

if is_prime == True:
    print('Prime')
else:
    print('Not Prime')

```

10000000000000000000
Not Prime

```
[39]: for i in range(2, 10.0):
      print(i, end = ' ')
```

```
-----
TypeError                                Traceback (most recent call last)
Input In [39], in <cell line: 1>()
----> 1 for i in range(2, 10.0):
      2     print(i, end = ' ')

TypeError: 'float' object cannot be interpreted as an integer
```

2 Least Common Multiple (LCM)

- lcm(12, 18) -> 36
- lcm(4, 5) -> 20
- lcm(6, 12) -> 12

```
[47]: a, b = map(int, input().split()) # a = 12 b = 18
i = 1
while True:
    m = a * i # m = 12 * 3 --> 36
    if m % b == 0: # 36 % 18 == 0
        print(f'LCM is {m}') # 36
        break
    i += 1 # i = 1 + 1 = 2 + 1 = 3
```

```
1234567 7654321
LCM is 9449772114007
```

```
[52]: a, b = map(int, input().split()) # a = 12 b = 18
if a < b:
    a, b = b, a
i = 1
while True:
    m = a * i # m = 12 * 3 --> 36
    if m % b == 0: # 36 % 18 == 0
        print(f'LCM is {m}') # 36
        break
    i += 1 # i = 1 + 1 = 2 + 1 = 3
```

```
7654321 1234567
LCM is 9449772114007
```

```
[56]: a, b = map(int, input().split()) # a = 12 b = 18
if a < b:
    a, b = b, a
print(a, b)
```

```
100 20
100 20
```

3 GCD (Greatest Common Divisor) or HCF (Highest Common Factor)

```
[8]: # GCD of two given numbers (Normal Approach)
a, b = map(int, input().split())
# factors of a
for i in range(1, a + 1):
    if a % i == 0 and b % i == 0:
        gcd = i
print(gcd)
```

```
123456789 987654321
9
```

3.1 Euclid's Algorithm to Find out GCD of two given numbers

```
[9]: a, b = map(int, input().split())
while a != b:
    if a > b:
        a -= b
    else:
        b -= a
# print(a)
print(b)
```

```
123456789 987654321
9
```

3.2 Extended Euclid's Algo for GCD

```
[10]: a, b = map(int, input().split()) # 12 18
while a != 0 and b != 0: # False
    if a > b: # if 12 > 6
        a %= b # a = 0
    else:
        b %= a # b = 6
if a == 0:
    print(b)
else:
    print(a)
```

```
123456789 987654321
9
```

4 Best way to find out LCM of two numbers

- is to find out GCD of those two numbers first

```
[12]: a, b = map(int, input().split()) # a = 12 b = 18
if a < b:
    a, b = b, a
i = 1
while True:
    m = a * i # m = 12 * 3 --> 36
    if m % b == 0: # 36 % 18 == 0
        print(f'LCM is {m}') # 36
        break
    i += 1 # i = 1 + 1 = 2 + 1 = 3
```

123456789 987654321

LCM is 13548070123626141

5 LCM through GCD

```
[ ]: a, b = map(int, input().split()) # 12 18
p = a * b
while a != 0 and b != 0: # False
    if a > b: # if 12 > 6
        a %= b # a = 0
    else:
        b %= a # b = 6
gcd = b if a == 0 else a
lcm = p // gcd
print(lcm)
```

```
[70]: s = int(input())
start = 9
ans = 0
while s > 0:
    if start <= s:
        s -= start
        ans = ans * 10 + start
    start += 1
print(str(ans)[::-1])
```

45

123456789