Printing a Large Integer

```
# Prints n with usual comma-separation.
# Pre:  n is an integer.
def print_with_commas(n):

    if n < 0:        # handle negative sign, if neccessary
        print('-',end='')
        n = -n

    if ( n < 1000 ): # base case
        print(n,end='')
    else:
        print_with_commas( n // 1000) # integer division!
        print( ',',end='')
        print(n % 1000,end='')
```

Engage: Trace Printing a Large Integer
How many recursive calls?

1. 6492335 →
2. 14 →
3. 123456 →
4. 43003451999 →
5. 82842083298355 →

Recursion vs Iteration

```
# Computes the value of n!.
# Pre:  n is a non-negative integer.
def factorial(n):
    if  n < 2:                  #base case
        return 1
    return n * factorial(n-1) #recursive case
```

```
# Computes the value of n!.
# Pre:  n is a non-negative integer.
def factorial2(n):
    product = 1
    while  n > 0:
        product = n * product
        n -=1
    return product
```

Engage: Trace factorial

n	Recursive No. of recursive calls	Iterative No. of iterations
5		
7		
12		
15		
N		

```
# Computes the largest integer that divides both M and N.
# Pre:  m,n are a non-negative integers, not both 0.
# Credit:  Euclid
def gcd(m,n):
    if n == 0:              # base case
        return m
    return gcd(n, m % n)    # recursive case
```

Engage: Trace GCD precondition was not both 0

M	N	No. of recursive calls	Answer
1	0		
1	8		
4	8		
20	15		
540	288		