MCS 260 Spring 2012 Lowman, GCD

Iteration + Walk-thru and Recursion with Stack-Trace

Iteration:

```
def gcd(m, n):
    while n != 0:
        r = m % n
        m = n
        n = r
    return m
```

Check with walk-thru for gcd(8, 12)

m	n	r = m % n	Action
8	12	8 % 12 = 8	
12	8	12 % 8 = 4	
8	4	8 % 4 = 0	
4	0		return m = 4

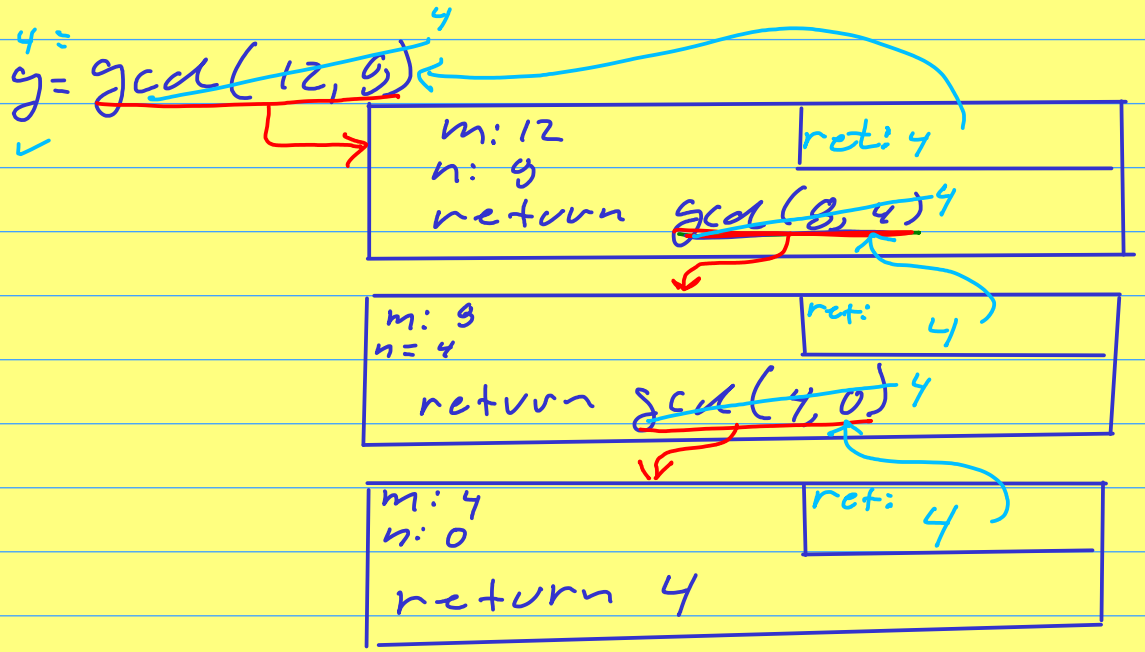
Note: Walk-thru rules are:

- time goes left to right in a row.
- If need to backup in row (can't) must start new line
- Only write values in table when memory values are over written.

Recursion:

```
def gcd(m, n):
    if n == 0:
        return m
    return gcd(n, m%n)
```

Check gcd(12,8) with a stack trace.



Stack Trace Rules:

1. arrows leave function calls and terminate on the edge of a box (stack frame)
2. Return values are stored in the return register box.

3. Return arrows leave the return register box and terminate on a function call.
4. The function call is crossed out and the return value is written above it.
5. When the calculation is complete the return value is saved in the return register. An arrow leaves the return register and ends on the function call.
6. This repeats until the original function call is crossed out and the final return value is written above it.

In all cases, your Walk-Thru / Stack-Trace should show what your code actually does and NOT what you think it should be doing