Chapter 4: Practice Makes Perfect

A huge part of learning recursion is practice! Please try the following problems; there are solutions in Chapter 5 to check your work. Keep in mind that these can be pretty difficult!

```
####
# 1) we'll start off with something familiar. without looking at sum digits,
#    implement the following function so that md(number) returns the PRODUCT of
#    digits of number.
#
#    md(4023) -> 4 * 0 * 2 * 3 = 0
#    md(423) -> 4 * 2 * 3 = 24
####

def md(number):
    if __________:
        return _____

    ____________________
    ____________________

    return ____________________________
```

```
####
# 2) Exponents are basically repeated multiplication! For example,
#    2^3 (2 to the power of 3) = 2 * 2 * 2 = 8
#    Basically, it's three 2's multiplied together.
#    We can write a function for this recursively!
#    2 is the base, and 3 is the exponent in this case.
#    So, for base ^ exponent, write the following function that returns that va
#
#    HINT: remember when we did factorial, we thought about how we have to
#    make the problem smaller, and tried to relate the smaller problem
#    to our original problem. That may help here!
#
#    rec_power(2, 3) = 2 * 2 * 2 = 8
#    rec_power(4, 2) = 4 * 4 = 16
```

```
#
####

def rec_power(base, exponent):
    if _________________:
        return ________

    exponent = ____________________
    return ________________________
```

```
####
# 3) Implement the following function so that count8(number)
#    counts the number of times the number "8" appears in number.
#
#    I've given you parts of the base cases; if the number left is 8, we've
#    found an 8, so what should we return?
#    If the number left isn't 8, but can't be made any smaller, what should
#    we return then?
#
#    count8(3283) -> 1
#    count8(32883) -> 2
#    count8(8388) -> 3
####

def count8(number):
    if number == 8:
        return _____
    elif number < 10:
        return _____

    rightmost_digit = ________________
    rest_of_number = ________________

    if rightmost_digit == 8:
        return ________________________
    else:
        return ________________________
```