1. [5 points] What does q1( ) print? (Recall that \( a \div b = 0 \) if \( b \) evenly divides \( a \). For example, \( 8 \div 4 = 0 \) but \( 8 \div 3 = 2 \).)

```python
def q1():
    sum = 0
    items = [1, 2, 3, 4, 5, 6]
    for el in items:
        if el % 3 == 0:
            sum += 1
    print(sum)
```

2. [5 points] What does q2( ) print? (Recall that \( \div \) is integer division. For example, \( 5 \div 2 = 2 \).)

```python
def xform(ar, f):
    for m in range(len(ar)):
        ar[m] = f(ar[m])

def half(x):
    return x \div 2

def q2():
    m = [2, 4, 6, 8]
    xform(m, half)
    tot = 0
    for el in m:
        tot += el
    print(tot)
```
3. [5 points] What does q3( ) print?

```python
def magnitude(n):
    if n > 0:
        return 1 + magnitude(n // 10)
    else:
        return 0

def q3():
    print(magnitude(1234567))
```

4. [5 points] What does q4( ) print?

```python
def scrub(li, m):
    for i in range(len(li)):
        if li[i] == m:
            li[i] = 0

def scrub_all(li, bad):
    for el in bad:
        scrub(li, el)

def q4():
    ar = [3, 7, -5, 10, -17, 20]
    negs = []
    for item in ar:
        if item < 0:
            negs.append(item)
    scrub_all(ar, negs)
    tot = 0
    for item in ar:
        tot += item
    print(tot)
```
5. [15 points] Complete the function `max_area`, consistent with its docstring.

```python
class Rect:
    def __init__(self, height, width):
        """Create rectangle. Height and width must be positive."""
        assert(height > 0 and width > 0)
        self.height = height
        self.width = width

    def area(self):
        return self.height * self.width

def max_area(li):
    """Find the area of the biggest rectangle in a list.
    Args:
        li: A list of Rect objects.
    Returns:
        the maximum of the areas of Rect objects in li.
        Returns 0 if li is empty.
    Examples:
        max_area( [ Rect(5,3), Rect(4,2), Rect(3,3) ] ) = 15
        max_area( [ Rect(1,2), Rect(2,1), Rect(1,1) ] ) = 2
        max_area( [ ] ) = 0
    """
```
6. [15 points] Finish the function `groups` below, consistent with the docstring. Your solution should run in linear time.

```python
def groups(li):
    """
    Partition li into groups of identical items
    Args:
        li: A list of integers
    Returns:
        A list containing lists of items from li, in the same order as li. Each sub-list contains a sequence of identical elements.
    Examples:
        groups( [1, 2, 2, 2, 3, 4, 5, 5, 6] ) = [[1],[2, 2, 2], [3], [4], [5, 5], [6]]
        groups( [1, 3, 4] ) = [[1], [3], [4]]
        groups( [3, 3, 3] ) = [[3, 3, 3]]
        groups( [ ] ) = [ ]
    """
```
def reformat(s, fmt):
    """
    Reformat phone number s into pattern fmt.
    Args:
        s is a string, typically of digits.
        fmt is a string in which # is a place-holder for a character from s.
    Returns:
        A string identical to fmt except each # is replaced by one character
        from s, in order. If s has more characters than fmt has #, the extras
        go at the end. If s has too few characters, the extra # are discarded.
    Examples:
        reformat("5413464140", "(###) ###-####") = "(541) 346-4140"
        reformat("15413464140", "+# ###.###.####") = "+1 541.346.4140"
        reformat("3464140", "(###) ###-####") = "(346) 414-0"
        reformat("15413464140", "(###) ###-####") = "(154) 134-64140"
    """