CIS 210
Winter 2014 Final Exam

Your name: Ima Key

Total: _________ of 65 possible

Here is an example exam filled out by hand, with some notes.
1. [5 points] What does q1( ) print? (Recall that \( a \% b = 0 \) if \( b \) evenly divides \( a \). For example, \( 8 \% 4 = 0 \) but \( 8 \% 3 = 2 \).)

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

A little bit of annotation (like the underlines of 3 and 4, and the replaced values in the list below) are often helpful in seeing what you were thinking, which sometimes helps me give some partial credit if you had the right idea but made a simple mistake.

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 // 2

def q2():
    m = [2, 3, 6, 8]
    xform(m, half)
    tot = 0
    for el in m:
        tot += el
    print(tot)
```

10
3. [5 points] What does q3() print?

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?

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.

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
    """
    max = 0  # ok because area must be positive
    for rect in li:
        if rect.area() > max:
            max = rect.area()
    return max
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], [3], [4], [5, 5], [6]]
groups( [1, 3, 4] ) = [[1], [3], [4]]
groups( [3, 3, 3] ) = [[3, 3, 3]]
groups( [] ) = []
    """

    result = []
cur_group = []
    if len(li) == 0:
        return result
    cur_val = li[0]
    for v in li:
        if v == cur_val:
            cur_group.append(v)
        else:
            result.append(cur_group)
            cur_group = [v]
    result.append(cur_group)
    return result
```

The examples are useful test cases. They are designed to catch some potential mistakes in your code, so you should "hand execute" your code on them.

Draft and revise your code on scratch paper before transcribing the final code to the exam. Besides making it neater, you will avoid a lot of silly errors that arise when you are making decisions and writing code at the same time.

Oops ... should set `cur_val` here
7. [15 points] Finish the function `reformat` below, consistent with the docstring.

```python
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"

    result = 
    s-index = 0
    for ch in fmt:
        if ch != '#':
            result = result + ch
        elif s-index < len(s):
            result = result + s[s-index]
            s-index = s-index + 1
    while s-index < len(s):
        result = result + s[s-index]
        s-index += 1
    return result
```