CIS 210 Winter 2017 Midterm Practice Questions - Key

(0) Create a string variable, name, that is assigned to your name:

    name = 'Midterm Practice Questions - Key'

(1) What is the result of executing >>> q1(90)?

    def q1(score):
        ''' No docstring on the exam '''
        gradepoint = 0

        if score >= 90:
            gradepoint = 4

        if score >= 80:
            gradepoint = 3

        if score >= 70:
            gradepoint = 2

        if score >= 60:
            gradepoint = 1

        return gradepoint
(2) Replace the ?? with the results of executing the following code in the Python Shell (indicate 'error' if the result would be a Python error message)

```python
>>> x = 210
>>> y = '210'
>>> len(x)
error
>>> len(y)
3
>>> x / 3
70.0 [1 pt. for number, 1 pt. for float]
>>> y / 3
error
>>> x = y
>>> y = '211'
>>> x + '_212'
'210_212' [1 pt. for string, 1 pt. for quotes]
>>> y + '_212'
'211_212'
```

(3) Replace ?? (2 places) with the expected results:
```
def twice(x):
    ... ...
    result = 2 * x
    print(result)
    return None
```

```python
>>> x = 99
>>> twice(10)
20
>>> x
99
```
(4) An approximation for the square root of \( n \) can be generated using the following equation:

\[
x_{k+1} = \frac{1}{2} \left( x_k + \frac{n}{x_k} \right), \text{where } x_0 = 1
\]

Each value of \( x \) should be a better approximation for the square root of \( n \).

(a) Supply the type contract for function `approx_sqrt` below, consistent with this equation. [2 pts.]

(b) Replace the `??` (2 places) with the code needed to implement the approximation.

```python
def approx_sqrt(num, iterations):
    '''(number, int) -> float
    
    Generates an approximate square root of num, a positive number, via an iterative process that runs iterations times. The approximate square root is returned.
    
    >>> approx_sqrt(1, 1)
    1.0
    >>> approx_sqrt(4, 1)
    2.5
    >>> approx_sqrt(4, 5)
    2.000000000000002
    ...
    
    value = 1
    for ctr in range(iterations):
        value = .5 * (value + num/value)
    
    return value
```
import math
def isInCircle(x, y, r):
    '''(number, number, number) -> Boolean

    Returns True if point (x, y) is in
    the circle with radius r.

    >>> isInCircle(.5, .5, 1)
    True
    >>> isInCircle(1, 2, 1)
    False
    '''
    d = math.sqrt(x**2 + y**2)
    return (d <= r)

(6) (a) The following code does not execute as expected. Add the missing code:
    total = 0
    astr = 'a b c d e f'
    i = 0
    while i < len(astr):
        if astr[i] == ' ':
            total += 1
        i += 1
print(total)

(6) (b) Rewrite the while loop as a for loop:
    for i in range(len(astr)):
        if astr[i] == ' ':
            total += 1
(7)(a) Complete the docstring for function q7.

(7)(b) What is the result of executing >>> q7('hello')?  'helo'

def q7(myStr):
    '''(str) -> str

    removes duplicate chars
    from myStr; returns the
    new string

    >>> q7('hello')
    'helo'
    '''
    newStr = ''
    for nextCh in myStr:
        if nextCh not in newStr:
            newStr += nextCh
    return newStr

(8) What is the result of executing >>> q8(7, 2)?  12 12 0

def q8(w, z):
    '''(int, int) -> None
    Exam docstring has type contract only.
    '''
    while z < w :
        w += 1
        z += 2
    print(w, z, z - w)
    return None