CIS 210 Example Midterm Questions – KEY

Note: These questions are not a comprehensive study guide! They are given here to provide you with a sense of the types of multiple choice questions that may be on the midterm.

To prepare thoroughly for the midterm you should review readings from the text, class notes, lab notes, projects and project solutions.

The format of the midterm will be multiple choice questions (Scantron), and one or more questions where you will be given a problem specification and write a solution in Python code using CIS 210 style guidelines. If the problem has a posted solution, that is, was assigned as a weekly project, a correct response will be code identical to or very close to the posted solution.

1. (1 pt.) Given:

   nestedseq = [['a', 'b'], [1, 2]]

   What is the result of executing the following code:

   >>> nestedseq[0][0]

   a) 'a'    b) ['a', 'b']    c) 1    d) [1, 2]

2. (2 pts) Given:

   def q1(alist):
       '''(list) -> dictionary
           ...
           examD = {}

           for item in alist:
               if item in examD:
                   examD[item] += 1
               else:
                   examD[item] = 1

           return examD

   What is the result of executing the following code:

   >>> q1(['hello', True, 'hello', False])

   a) {False: 1, True: 1, 'hello': 2}    c) either a or b
   b) {'hello': 2, True: 1, False: 1}    d) neither a nor b

3. (1 pt.) What will be the result of executing the following code:
>>> print(print('hello'))

a) 'hello'   b) None   c) 'hello'   d) None

4. (2 pts.) Given:

def q4aux(n):
    ''' (int) -> int
    ...'''
    if n <= 0:
        return 1
    else:
        return 2 * q4aux(n - 1)

def q4(x):
    ''' (int) -> int
    ...'''
    result = q4aux(x)
    return result

What will be the result of executing the following code:

>>> q4(5)

a) 1   b) 10   c) 16   d) 32

5. (1 pt.) Given:

def q5(li):
    ''' (list of sequences) -> list of sequences
    ...'''
    newli = []
    for item in li:
        if len(item) < 3:
            newli.append(item)

    return newli

What will be the result of executing the following code:

>>> q5(['abc', 'de', [4,5], [1, [2, 3]]])

a) ['de', [4, 5]]   b) ['de', [4, 5], [1, [2, 3]]]

c) ['abc', 'de']   d) None

6-8. (1 pt. for each) Given:
def q6(li):
    '''(list) -> Boolean

    Returns True if the number of strong passwords is greater than the number of weak passwords in li, otherwise returns False.
    '''

    frequentpsw = ['1234', 'password', 'qwerty']
    weakli = []
    strongli = []
    for item in li:
        if item in frequentpsw:
            weakli = weakli + [item]
        else:
            strongli = strongli + [item]

    return len(strongli) > len(weakli)

6. What will be the result of executing the following code:

>>> q6(['password', 'xt4j-9', 'r&s&t'])

a) True b) False c) cannot be determined d) error

7. The error in function q6 is an example of a

a) syntax error b) runtime error c) logic error d) hardware error

8. The error in function q6 stems from what characteristic of Python?

a) dynamic typing b) static typing c) strong typing d) weak typing
9. (2 pts) Given:

```python
def foo(z):
    '''exam q'''
    w = z.pop()
    return w

def bar(x):
    '''(list) -> None'''
    foo(x)
    print(x)
y = foo(x)
    print(x)
    print(y)
    return None
```

What will be the result of executing the following code:

```python
>>> bar(['grape', 'fir', 'navy'])
```

a) b) c) d)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>['grape', 'fir']</td>
<td>['grape', 'fir', 'navy']</td>
</tr>
<tr>
<td>b)</td>
<td>['grape', 'navy']</td>
<td>['grape', 'fir', 'navy']</td>
</tr>
<tr>
<td>c)</td>
<td>['grape', 'fir', 'navy']</td>
<td>['grape', 'fir']</td>
</tr>
<tr>
<td>d)</td>
<td>['grape', 'fir', 'navy']</td>
<td>['grape']</td>
</tr>
</tbody>
</table>

grape fir None
10-12. Given:

def strReverseI(s):
    '''(str) -> str

    Iterative function to reverse s; return the reversed string.
    '''

    rstr = ''
    for ch in s:
        rstr = ch + rstr

    return rstr

10. The better test suite is:

   a) >>> strReverseI('abc')
      'cba'
      >>> strReverseI('hello')
      'olleh'
      >>> strReverseI('')
      ''
      >>> strReverseI('CIS 210')
      '012 SIC'

   b) >>> strReverseI('xyz')
      'zyx'
      >>> strReverseI('cat')
      'tac'
      >>> strReverseI('foo')
      'oof'
      >>> strReverseI('rat')
      'tar'

11. Additional helpful tests could include:

   a) >>> strReverseI('a')
      'a'
      >>> strReverseI('aabbc')
      'cbba'
      >>> strReverseI('ZzZzZz')
      'zZzZzZ'

   b) >>> strReverseI('abc')
      'cba'
      >>> strReverseI('too')
      'oot'
      >>> strReverseI('ajax')
      'xaja'

12. For a string of length 13, the for loop will execute

   a) 1 time    b) 4 times    c) 13 times    d) unknown number times