1-3. Given:

numbers = ['1', '2', '3', '4', '5', '6', '7', '8', '9', '0']
special = ['!', '@', '#', '$', '%', '&', '*']

def q1(psw):
    ''' exam function '''
    score = 0
    for ch in psw:
        if ch in numbers:
            score += 2
        elif ch in special:
            score += 3
        else:
            score += 1

    if score > 8:
        return True
    else:
        return False

1. [2 pts.] An appropriate type contract for function q1 would be

   a) (str) -> Boolean     b) (str) -> int
   c) (int) -> Boolean     d) (int) -> str

2. [2 pts.] Function q1

   a) returns a non-None value; no side effect
   b) returns None value; causes a side effect
   c) returns a non-None value; causes a side effect
   d) returns None value; no side effect

3. [2 pts.] What will be the result of executing the following code:

   >>> q1('hello!')

   a) 6       b) 8       c) True       d) False
4-6. Given:

```python
def q4(li):
    ''' exam function. li is a sorted list of integers'''
    g = 0
    for i in range(1,len(li)):
        d = li[i] - li[i-1]
        if d > g:
            g = d
    print(g)
    return None
```

4. [2 pts.] An appropriate type contract for function q4 would be

a) (int) -> None  b) (int) -> int  
c) (list) -> None  d) (list) -> int

5. [2 pts.] Function q4

a) returns a non-None value; no side effect  
b) returns None value; causes a side effect  
c) returns a non-None value; causes a side effect  
d) returns None value; no side effect

6. [2 pts.] What will be the result of executing the following code:

>>> q4([1, 2, 4, 7])

a) 0  b) 1  c) 2  d) 3

7. [3 pts.] Given:

```python
def q7(n):
    ''' (int) -> int
    n is a non-negative integer
    '''
    if n == 0:
        return 0
    else:
        return 10 + q7(n - 1)
```

What will be the result of executing the following code:

```python
>>> q7(3)
```

a) 10  b) 20  c) 30  d) 40
8. [3 pts.] Given:

\[ g = \{0:[1,4], \]
\[ 1:[2,3], \]
\[ 2:[], \]
\[ 3:[], \]
\[ 4:[3,5], \]
\[ 5:[6], \]
\[ 6:[]\} \]

```python
def q8(gr, start):
    ''' exam function '''
    print(start)
    for child in gr[start]:
        q8(gr, child)
    return None
```

What will be printed when the following code is executed (ignoring newline characters):

```python
>>> q8(g, 0)
```

a) 0 1 2 3 4 5 6  

b) 0 1 2 3 4 3 5 6

c) 0 1 4 2 3 5 6  

d) 0 1 3 4 3 2 5 6

9-10. Given:

def q9(s):
    '''(str) -> str
       Return string with information about string length.
    >>> q9('hello, world')
    'hello, world-12'
    '''
    ct = len(s)
    s = s + '-\n' + ct
    return s

9. [2 pts.] Executing function q9 will cause a

a) logic error  b) run time error – IndexError  c) run time error – TypeError

10. [2 pts.] The error in function q9 stems from what characteristic of Python?

a) static typing  b) strong typing  c) parameter passing  d) None value
11. [1 pt.] Given:

```python
def q11(s):
    ''' Exam function '''
    vowels = ['a', 'e', 'i', 'o', 'u']
    svowels = [ch for ch in vowels if ch in s]
    return svowels
```

What will be the result of executing the following code:

```python
>>> q11('quick brown fox')
```

a) ['i', 'o', 'u']  
b) ['a', 'e', 'i', 'o', 'u']  
c) 3  
d) None

12. [2 pts.] Given:

```python
def q12(s):
    ''' exam function '''
    vowels = ['a', 'e', 'i', 'o', 'u']
    vowelct = 0
    for item in s:
        if item in vowels:
            vowelct += 1
    return vowelct
```

What will be the result of executing the following code:

```python
>>> q12('the quick brown fox')
```

a) 0  
b) 1  
c) 5  
d) None

13. [2 pts.] Given:

```python
def q13(s):
    ''' exam function '''
    vowels = ['a', 'e', 'i', 'o', 'u']
    vowelct = 0
    for item in s:
        if item in vowels:
            vowelct += 1
    return vowelct
```

What will be the result of executing the following code:

```python
>>> q13('the quick brown fox')
```

a) 0  
b) 1  
c) 5  
d) None
14-17. Given:

def q14(alist):
    ''' exam function '''
    countd = {}
    for item in alist:
        if item in countd:
            countd[item] += 1
        else:
            countd[item] = 1
    countli = countd.values()
    maxct = max(countli)
    mli = []
    for k in countd:
        if countd[k] == maxct:
            mli.append(k)
    return mli

14. [2 pts.] An appropriate type contract for function q14 would be

a) \((\text{list}) \rightarrow \text{dict}\)  
   b) \((\text{dict}) \rightarrow \text{list}\)  
   c) \((\text{list}) \rightarrow \text{list}\)  
   d) \((\text{dict}) \rightarrow \text{dict}\)

15. [2 pts.] Function q14

   a) returns a non-None value; no side effect
   b) returns None value; causes a side effect
   c) returns a non-None value; causes a side effect
   d) returns None value; no side effect

16. [2 pts.] What will be the result of executing the following code:

```python
>>> q14([1, 1, 1, 2, 2, 2, 3, 3, 3])
```

a) [ ]  
   b) [1, 2, 3]  
   c) {1:3, 2:3, 3:3}  
   d) [1, 1, 1, 2, 2, 2, 3, 3, 3]

17. [2 pts.] The better test suite for function q14 would be

a) q14([5, 7, 1, 3])  
   q14([1, 2, 2, 3, 99])  
   q14([99])  
   q14([0, 0, 1, 1])

b) q14([1, 2, 3, 4])  
   q14([100, 99, 98, 97])  
   q14([2, 4, 6, 8])  
   q14([1, 7, 5, 3])
18-19. Given:

```python
def q18_aux(codeli, resultli):
    ''' exam function '''
    for item in codeli:
        resultli.append(item[0])
    return None

def q18_main():
    ''' exam function '''
    codeli = [('a', 'z'), ('b', 'y'), ('c', 'x')]
    resultli = []
    q18_aux(codeli, resultli)

    # checkpoint 1
    print(codeli)

    # checkpoint 2
    print(resultli)
    return None
```

18. [3 pts.] What will be printed at checkpoint 1?

```python
>>> q18_main()
[('a', 'z'), ('b', 'y'), ('c', 'x')]
```

a) `[('a', 'z'), ('b', 'y'), ('c', 'x')]`  
b) `[('a', 'z')]`

c) `['a', 'b', 'c']`  
d) None

19. [3 pts.] What will be printed at checkpoint 2?

```python
>>> q18_main()
[]
```

a) `[]`  
b) `['a']`

c) `['a', 'b', 'c']`  
d) None
20-22. Given:

def foo(li, item):
    ''' exam function '''
    li.append(item)
    print(li)
    li = li.append(item)
    print(li)
    return item + 1

def bar():
    '''() -> None'''
    ...
    x = 99
    foo([], x)
    # checkpoint 1
    x = foo([], x)
    # checkpoint 2
    return None

What will be the result of executing the following code:

>>> bar()

20. [2 pts.] What will have been printed after the first call to foo?

<table>
<thead>
<tr>
<th>a)</th>
<th>b)</th>
<th>c)</th>
<th>d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[99]</td>
<td>None</td>
<td>99</td>
<td>[99]</td>
</tr>
<tr>
<td>None</td>
<td>[99]</td>
<td>[99]</td>
<td>[99, 99]</td>
</tr>
</tbody>
</table>

21. [2 pts.] What is the value of x at checkpoint 1?

a) [99]  b) 99  c) 100  d) None

22. [2 pts.] What is the value of x at checkpoint 2?

a) [99]  b) 99  c) 100  d) None
24. [6 pts.] Write function, `rainfall`, with one parameter, `rainli`, a list of floating point numbers that record the daily rainfall in Eugene from March 1 through March 15. Function `rainfall` should return the average (mean) daily rainfall for Eugene in this time period. Code should be written according to CIS 210 style guidelines, including a docstring with a type contract; brief description that explicitly refers to parameters, return value, and side effects (if any); and at least one example of use.

```python
def rainfall(rainli):
    '''(list) -> float

    Returns average (mean) daily rainfall from data in rainli.
    
    >>> rainfall([1, 2, 3])
    2.0
    '''
    totalrain = 0
    for rainct in rainli:
        totalrain += rainct

    #avg_rain = sum(rainli) / len(rainli)
    avg_rain = totalrain / len(rainli)

    return avg_rain
```
25. [6 pts.] Write a function, raindata, with one parameter, f, the name of a file containing rain data (daily rainfall in Eugene from March 1 through March 15).

The file will have two lines: a header line, and another line of comma separated values, which are the daily rainfall amounts:

```
#rainfall data for final exam CIS 210 W17 03/01-15/2017 Eugene, OR .2, .19, .16, .2, .16, .15, .2, .12, .15, .16, .16, .16, .16, .16, .2
```

Function raindata should create and return a rain dictionary, where the dictionary keys are the days of the month (1 through 15) and the associated value is the rainfall for that day.

Code should be written according to CIS 210 style guidelines, including a docstring with a type contract; brief description that explicitly refers to parameters, return value, and side effects (if any); and at least one example of use. Example of use may refer to a small example file, e.g.,

```
#rainfall data .2, .19
```

```python
def raindata(f):
    '''(str) -> dict

    Rainfall data from file f are stored in a data dictionary by day of month.

    >>> raindata('rainfall.txt')
    {1 : .2, 2 : .19}
    '''
    with open(f) as rainf:
        rainf.readline()
        rain = rainf.readline().rstrip().split(',

    raindict = {}
    day = 1
    for rainct in rain:
        raindict[day] = float(rainct)
        day += 1

    return raindict
```

→ TURN OVER FOR 1 MORE PROBLEM
23. [1 pt.] Given:

def best(greeting):
    '''(str) -> None

    print a greeting
    '''
    start = greeting.find('Best')
    print(greeting[start:])
    return None

What will be the result of executing the following code:

>>> best('CIS 210: Best wishes for a pleasant break.')

a) Best wishes for a pleasant break.