(0) (4pts.) What are the values of a, b, and c after the following code is entered into the Python shell?

```python
>>> a = 25
>>> b = a / 5
>>> c = str(a)
>>> a += 1

>>> a
26
>>> b
5.0
>>> c
'25'
```

(1) (2 pts.) Given:

```python
x = 5
y = 10 // 3
z = 5 % 2
```

What will be the result of executing the following code?

```python
>>> (y < x) and (z < x)
True
```

(2) (7 pts.) After the following code is entered into the Python Shell:

```python
>>> x = 1.2
>>> y = 3
>>> x = x + y
>>> z = '97403'
```

What will be the result of executing the following code?

```python
>>> 'z' * 2
'zz'

>>> z * 2
'9740397403'

>>> x + y * y
13.2

>>> float(y) + round(x)
7.0

>>> [x, y, z]
[4.2, 3, '97403']
```
Questions (3) (a) through (f) are a group and should be answered in order:

(3) (12 pts.) (a) When a user enters

```python
>>> info = [['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette']]
```

into the Python shell, what does Python return?

a) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette']]`

b) `info`

c) `'info'`

d) `None`

(b) If a user then enters

```python
>>> info[1][1]
```

into the Python shell, what does Python return?

a) `['Waldo', 'Summit']`  
b) `['Broken Top', 'Jefferson']`  
c) `Jefferson`  
d) `None`

(c) If a user then enters

```python
>>> for nature in info:
    for name in nature:
        print(name)
```

into the Python shell, what does Python print?

```
Waldo
Summit
Broken Top
Jefferson
McKenzie
Willamette
```

(d) If a user then enters

```python
>>> example = info.append(['UO', 'OSU'])
```

into the Python shell, what does Python return?

a) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette'], ['UO', 'OSU']]`

b) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette'], ['UO', 'OSU']]`

c) `None`

d) `none of these`
(e) If a user then enters

```python
>>> info
```
to the Python shell, what does Python return?

a) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette']]`

b) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette'], ['UO', 'OSU']]`

c) None

d) none of these

(f) If a user then enters

```python
>>> info[0:2]
```
to the Python shell, what does Python return?

a) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette']]`

b) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson'], ['McKenzie', 'Willamette'], ['UO', 'OSU']]`

c) `[['Waldo', 'Summit'], ['Broken Top', 'Jefferson']]`

d) None

(4) (4 pts.) Given the following Python code:

```python
def display_pops(popli):
    
    return a list of the numbers from population list translated into strings in millions
    
    mill_pops = []

    for nextpop in popli:
        nextpop = str(nextpop)
        nextpop = nextpop + ',000,000'
        mill_pops.append(nextpop)

    print('Populations:',
    return mill_pops
```

What will be the output after the following Python code is executed:

```python
nation_pops = [7, 50]  
print(display_pops(nation_pops))  
```

```
('Populations:', [7, 50])
['7,000,000', '50,000,000']
```
(5) (8 pts.) Given the following Python function:

```python
def trackscore(score):
    """convert score to gradepoint; return gradepoint""
    if score >= 90:
        gradepoint = 4
    elif score >= 80:
        gradepoint = 3
    elif score >= 70:
        gradepoint = 2
    elif score >= 60:
        gradepoint = 1
    return gradepoint
```

(a) Identify the function header: `def trackscore(score):`

(b) Show the result of executing the following Python code:

```python
>>> gp = trackscore(80)
>>> gp
3
```

(c) Give another good test case for trackscore (i.e., one that reveals a bug in the code): `trackstore(50)` for example - any argument < 60 will result in a NameError for gradepoint.

(d) Tell how this problem can be fixed (precise words or code):

- initialize gradepoint before the conditional, e.g., `gradepoint = 0`
- add an else clause to the conditional, e.g., `else: gradepoint = 0`

(6) (2 pts.) What will be the output after the following Python code is executed?

```python
day = 'Saturday' + ' ' + 'Sunday'
root = day[0:5] + 'n'
print(root)
```

`Saturn`

(7)(3 pts.) Suppose we have a text file called pykeywds.txt that contains the following data:

```plaintext
and  elif  if  print
as   else  import  raise
assert except in return
break exec is try
class finally lambda while
continue for not with
def from or yield
del  global pass
```
What will be the result when the following Python code is executed?

```python
with open('pykeywds.txt', 'r') as infile:
    for i in range(2):
        aline = infile.readline()
        print(aline)

and  elif  if  print
as  else  import  raise
```

(8) (10 pts.) Given the following Python code:

```python
def testfunc(numlist):
    
    What does this function do?
    
    accumulator = 0
ctr = 0
    while ctr < len(numlist):
        for num in numlist:
            accumulator += numlist[ctr]
            accumulator += num
        ctr += 1
    return accumulator
```

(a) Write the type contract for the function:

**list of numbers** -> **number** [note: would also execute for a list of strings, for example, but we are following the hint provided by the parameter name]

(b) What will be the result when the following Python code is executed?

```python
>>> fs = [0, 1, 1, 2, 3, 5]
>>> testfunc(fs)
12
```

(c)
```python
>>> fs.reverse()
>>> fs
[5, 3, 2, 1, 1, 0]
```

(d) Mark the code with changes needed to replace the while loop with a for loop.

(9) (6 pts.) Given the following Python code:

```python
def any_uppercase(astring):
    '''If there is any uppercase character in astring, return True, else return False'''
    for c in astring:
        if c.isupper():
            return True
    #your code goes here
```

(a) Write the type contract for any_uppercase:
```
(str) -> boolean
```

(b) Replace #your code goes here with code to complete the function:
```
return False
```

c) Circle the column with the better set of test cases for function any_uppercase:

| any_uppercase('abC') | any_uppercase('AbC') |
| any_uppercase('xyZ') | any_uppercase('ABCDEFG') |
| any_uppercase('AbC') | any_uppercase('hello') |
| any_uppercase('XYz') | any_uppercase('”') |

(10) (4 pts.) The following code displays a message(s) about the acidity of a solution:

```python
ph = float(input('Enter the ph level: '))
if ph < 7.0:
    print('It is acidic.')
elif ph < 4.0:
    print('It is a strong acid.')
```

(a) What message(s) are displayed when the user enters 6.4?  
It is acidic.

(b) What message(s) are displayed when the user enters 3.6?  
It is acidic.

c) Make a small change to one line of the code so that both messages are displayed when a value less than 4 is entered.  
`change elif to if.`

(15) (2 pts.) What will be a result of executing the following Python code:

```python
def best(greeting):
    """(str) -> str
    form and return a greeting
    ""
    start = greeting.find('Best')
    return(greeting[start:])

>>> print(best('CIS122: Best wishes for a pleasant summer.'))
Best wishes for a pleasant summer.
```