1) Given the following UNTESTED Python program:

```python
def twice(n):
    '''(n: int) -> int
    return twice the value of n
    >>> twice(3)
    6
    '''
    print(2 * n)
    return

def thrice(n):
    '''(n: int) -> int
    return three times the value of n
    >>> thrice(4)
    12
    '''
    return twice(n) + n

def main():
    '''driver for multiples program'''
    i = 5
    result = thrice(i)
    print('Result is', result)
    return
```

What would be result of executing this program (Run Module)? (Choose all that apply.)

a) functions twice, thrice, and main would be defined in the global frame

b) the program would execute and Python would print the result: Result is 15

c) the program would execute and Python would report a runtime error: TypeError

d) the program would execute and Python would report a runtime error: NameError

e) the program would execute and Python would print a value: 10
2) Given the same UNTESTED Python program with the addition of a new final line of code:

```python
CIS 210 Fall 2020
Midterm

def twice(n):
    '''(n: int) -> int
    return twice the value of n
    >>> twice(3)
    6
    '''
    print(2 * n)
    return

def thrice(n):
    '''(n: int) -> int
    return three times the value of n
    >>> thrice(4)
    12
    '''
    return twice(n) + n

def main():
    '''driver for multiples program'''
    i = 5
    result = thrice(i)
    print('Result is', result)
    return

main()
```

What would be result of executing this program (Run Module)? (Choose all that apply.)

a) functions `twice`, `thrice`, and `main` would be defined in the global frame
b) the program would execute and Python would print the result: Result is 15
c) the program would execute and Python would report a runtime error: TypeError
d) the program would execute and Python would report a runtime error: NameError
e) the program would execute and Python would print a value: 10
3) Given the following Python code:

```python
def qred(s):
    '''(s: str) -> bool

    Returns True if s includes two or more numbers.
    >>> qred('CIS 210')
    True
    >>> qred('example question')
    False
    '''
    pass
```

Re-order the following lines of code to replace `pass` and correctly implement the specification given in the docstring:

```python
1    for c in s:
2        digits_ctr += 1
3    digits_ctr = 0
4    return (digits_ctr >= 2)
5        if c.isdigit():
```

??

4) Given the following Python code:

```python
1    def f(x, y):
2        '''(x: int, y: int) -> int'''
3        x = 2 * x
4        y = 2 * y
5        if y > x:
6            return y - x
7        else:
8            return x - y
9
10   def qblue(x, y):
11      '''(x: int, y: int) -> None'''
12      x = f(x, y)
13      y = f(y, x)
14      print(x, y)
15      return
```

When `>>> qblue(20, 5)` is called in the Shell, which diagram represents the Python call stack after line 4 has been executed the first time?
a)
global frame: \( f \rightarrow \text{function} \\
\quad q\text{blue} \rightarrow \text{function} \\
\begin{align*}
q\text{blue}: & \quad x \rightarrow 20 \\
& \quad y \rightarrow 5 \\
\end{align*}
\begin{align*}
f: & \quad x \rightarrow 40 \\
& \quad y \rightarrow 10 \\
\end{align*}
b)
global frame: \( f \rightarrow \text{function} \\
\quad q\text{blue} \rightarrow \text{function} \\
\begin{align*}
q\text{blue}: & \quad x \rightarrow 20 \\
& \quad y \rightarrow 5 \\
\end{align*}
\begin{align*}
f: & \quad x \rightarrow 20 \\
& \quad y \rightarrow 5 \\
\end{align*}
c)
global frame: \( x \rightarrow 20 \\
\quad y \rightarrow 5 \\
\begin{align*}
q\text{blue}: & \quad x \rightarrow 20 \\
& \quad y \rightarrow 5 \\
\end{align*}
\begin{align*}
f: & \quad x \rightarrow 40 \\
& \quad y \rightarrow 10 \\
\end{align*}
5) The following code does not execute as expected. What needs to be changed?

```
1 total = 0
2 astr = 'a b c d e f'
3 i = 0
4 while i < len(astr):
5     if astr[i] == ' ':
6         total += 1
7 print(total)
```

a) add new line between 6 and 7: `i += 1` – indent matches if (line 5)
b) add new line between 6 and 7: `i += 1` – indent matches `total += 1` (line 6)
c) add new line between 4 and 5: `i += 1` – indent matches while (line 4/no indent)
d) `total = 1` replaces `total = 0` (line 1)
e) edit line 6: `total += 1` so indent matches while (line 4/no indent)

6) Given the following Python code:

```
def qgreen(w, z):
    '''(w: int, z: int) -> ??
    Exam function.
    '''
    while z < w:
        w = w + 1
        z = z * 2
    return z - w
```

Complete the type contract:

??

When `>>> qgreen(6, 2)` is called, what is the value of `z` before the while loop is executed?

??

What value is returned when `>>> qgreen(6, 2)` has finished executing?

??
7) Given the following UNTESTED Python code:

```python
def testsNeeded(s):
    '''(s: str) -> int'''

    if len(s) != 0:
        prev_char = s[0]
        dup_ct = 1
        high_ct = 1
    else:
        high_ct = 0

    for i in range(1, len(s)):
        if s[i] == prev_char:
            dup_ct += 1
        else:
            prev_char = s[i]
            if dup_ct > high_ct:
                high_ct = dup_ct
            dup_ct = 1

    return high_ct
```

What is the result of executing

```python
>>> testsNeeded('abc')
```

a) 1  b) 2  c) 3  d) infinite loop  e) True

```python
>>> testsNeeded('abbccc')
```

a) 1  b) 2  c) 3  d) infinite loop  e) True
8) Given the following Python code:

```python
def circle_area(diameter):
    '''(diameter: number) -> float

    Determine area of circle with given diameter;
    return this value.

    >>> circle_area(2)
    3.141592653589793
    '''
    r = diameter * .5
    area = pi * r**2
    return area

def pizza_calculator(diameter, cost):
    '''(diameter: int, cost: number) -> float

    Calculates and returns the cost per square inch
    of pizza for a pizza of given diameter and cost.

    >>> pizza_calculator(14, 18)
    0.117
    >>> pizza_calculator(14, 20.25)
    0.132
    '''
    r = diameter / 2
    pizza_area = pi * r**2
    cost_per_inch = cost / pizza_area
    cost_per_inch = round(cost_per_inch, 3)
    print('$' + str(cost_per_inch))
    return
```

Which line(s) of code in `pizza_calculator` could be replaced by `pizza_area = circle_area(diameter)`?

??

************