1. What does the following Python code fragment produce?
   ```python
code = 
numbers = list(range(2,14,3))
aSlice = numbers[::3]
print(aSlice)
```
A. [ ] (i.e., an empty list)
B. [2]
C. [2, 5, 8, 11]
D. An error message
E. None of the above

2. What does the variable `grades` contain once the following Python code fragment has executed?
   ```python
grades = ['B', 'A', 'D']
grades.insert(2, 'E')
```
A. ['B', 'A', 'E']
B. ['B', 'A', 'E', 'D']
C. ['B', 'E', 'A', 'D']
D. An error message.
E. None of the above

Consider the following Python code fragment and answer the following 4 questions:
```python
mystery = 0
someNumbers = [9,8,7,6,5,4,3,2,1]
for digit in someNumbers:
    if digit % 2 == 0:
        mystery += digit
    print("digit is %d." %digit)
    print("And so far, mystery is %d." %mystery)
print("Final mystery is ", mystery)
```

3. Once the Python code fragment above has executed, what is the content of the variable `mystery`?
   A. 0 (zero)
   B. [2, 4, 6, 8]
   C. [1, 2, 3, 4, 5, 6, 7, 8, 9]
   D. [1, 3, 5, 7, 9]
   E. None of the above

4. If the second statement of the Python code fragment above was replaced with
   ```python
someNumbers = list()
```
   what would the content of the variable `mystery` be once the modified Python code fragment had executed?
   A. 0 (zero)
   B. [2, 4, 6, 8]
   C. [1, 2, 3, 4, 5, 6, 7, 8, 9]
   D. [1, 3, 5, 7, 9]
   E. None of the above

5. What is the content of the variable `digit` during the 4th iteration of the `for` loop?
   A. 0 (zero)
   B. [ ] (i.e., an empty list)
   C. 6
   D. 4
   E. None of the above

6. What is the content of the variable `someNumbers` once the `for` loop has executed, i.e., has finished iterating?
   A. [9, 8, 7, 6, 5, 4, 3, 2, 1]
   B. [ ] (i.e., an empty list)
   C. [2, 4, 6, 8]
   D. [1, 3, 5, 7, 9]
   E. None of the above
Consider the following Python code fragment and answer the following 6 questions:

```python
count = 0
if condition_1:
    count += 1
if not condition_3 or condition_4:  # Here
    count += 2
else:
    count += 3
    count += 20
else:
    if condition_2:
        count += 4
    elif condition_5:  # Line 1
        count += 5  # Line 2
    else:  # Line 3
        count += 6  # Line 4
    count += 10
print(count)
```

7. What does the Python code fragment above produce when all conditions are True?
   - A. 3
   - B. 24
   - C. 4
   - **D. 23**
   - E. None of the above

8. What does the Python code fragment above produce when all conditions are False?
   - A. 4
   - B. 5
   - C. 6
   - **D. 0**
   - E. None of the above

9. Let us assume the execution flow reaches # Here, which values could condition_3 and condition_4 have in order for the statement count += 2 to be executed?
   - A. True, False
   - B. As long as condition_3 is False.
   - C. As long as condition_4 is False.
   - D. As long as the compound condition composed of not condition_3 and condition_4 evaluates to False.
   - E. None of the above

10. If # Line 1 and # Line 2 are replaced by
    ```python
    else:
    if condition_5:
        count += 5
    ```
        what should be done with # Line 3 and # Line 4?
        - A. They should stay where they are in the Python code fragment above.
        - B. # Line 3 should be moved right so that it aligns with the new if (in the box above).
        - C. # Line 4 should be moved right so that it stays indented in relation to # Line 3.
        - D. B and C
        - E. None of the above

11. Let us assume that all conditions are initially set to True. What is the minimum number of conditions that must be set to False in order for the Python code fragment above to produce 15?
    - A. 1
    - **B. 2**
    - C. 3
    - D. 0
    - E. None of the above

12. Let us assume that all conditions are initially set to False. What is the minimum number of conditions that must be set to True in order for the Python code fragment above to produce 23?
    - A. 1
    - **B. 2**
    - C. 3
    - D. 0
    - E. None of the above