1. Consider this code:

   ```python
   a = [1, 0]
   ```

   All of the following code fragments cause `a` to refer to `[1, 0, 8]`.

   Circle all of the code fragment(s) that create a new list.

   (a) `a.append(8)`  
   (b) `a = a + [8]`  
   (c) `a.insert(len(a), 8)`  
   (d) `a = [a[0], a[1], 8]`

   Circle all of the code fragment(s) that modify the original list.

   (a) `a.append(8)`  
   (b) `a = a + [8]`  
   (c) `a.insert(len(a), 8)`  
   (d) `a = [a[0], a[1], 8]`

2. Consider this code:

   ```python
   a = [1, 0, 8]
   b = a.sort()
   ```

   After the code above is executed, which of the following expressions evaluate to `True`? Circle those expression(s).

   (a) `a == [1, 0, 8]`  
   (b) `a == [0, 1, 8]`  
   (c) `b == [1, 0, 8]`  
   (d) `b == [0, 1, 8]`

3. Consider this code

   ```python
   a = [0, 1, 2]
   b = a
   b[2] = 100
   ```

   After the code above is executed, which of the following expressions evaluate to `True`? Circle those expression(s).

   (a) `a == [0, 1, 2]` and `b == [0, 1, 100]`  
   (b) `a == [0, 1, 2]` and `b == [0, 100, 2]`  
   (c) `a == [0, 1, 100]` and `b == [0, 1, 100]`  
   (d) `id(a) == id(b)`

4. Which of the following code fragments does not print `'na'` 12 times? Circle those expression(s).

   (a) `for i in range(12):
       print('na')`
   (b) `for i in range(1, 24, 2):
       print('na')`
   (c) `for i in range(1, 12):
       print('na')`
   (d) `for i in range(6, 12):
       print('na')`
   (d) `for i in range(6, 12):
       print('na')`