

1. What is the difference between a Python tuple and Python list?
- ☐ Lists maintain the order of the items and tuples do not maintain order
  - ☒ Lists are mutable and tuples are not mutable
  - ☐ Tuples can be expanded after they are created and lists cannot
  - ☐ Lists are indexed by integers and tuples are indexed by strings
2. Which of the following methods work both in Python lists and Python tuples?
- ☐ pop()
  - ☒ index()
  - ☐ sort()
  - ☐ append()
  - ☐ reverse()
3. What will end up in the variable **y** after this code is executed?

```
1 x, y = 3, 4
```

- ☐ A dictionary with the key 3 mapped to the value 4
- ☐ 3
- ☒ 4
- ☐ A two item list
- ☐ A two item tuple

4. In the following Python code, what will end up in the variable **y**?

```
1 x = { 'chuck' : 1 , 'fred' : 42, 'jan': 100}  
2 y = x.items()
```

- ☐ A list of integers
- ☐ A tuple with three integers
- ☒ A list of tuples
- ☐ A list of strings

5. Which of the following tuples is greater than **x** in the following Python sequence?

```
1 x = (5, 1, 3)  
2 if ??? > x :  
3     ...
```

- ☒ (6, 0, 0)
- ☐ (4, 100, 200)
- ☐ (5, 0, 300)
- ☐ (0, 1000, 2000)

6. What does the following Python code accomplish, assuming the **c** is a non-empty dictionary?

```
1 tmp = list()  
2 for k, v in c.items() :  
3     tmp.append( (v, k) )
```

- ☐ It computes the largest of all of the values in the dictionary
- ☐ It computes the average of all of the values in the dictionary
- ☒ It creates a list of tuples where each tuple is a value, key pair
- ☐ It sorts the dictionary based on its key values

7. If the variable **data** is a Python list, how do we sort it in reverse order?

- ☐ data = data.sort(-1)
- ☒ data.sort(reverse=True)
- ☐ data.sort.reverse()
- ☐ data = sortrev(data)

8. Using the following tuple, how would you print 'Wed'?

```
1 days = ('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun')
```

- ☐ print(days[2])
- ☐ print(days.get(1,-1))
- ☒ print(days[2])
- ☐ print(days{2})
- ☐ print(days[1])
- ☐ print(days(2))

9. In the following Python loop, why are there two iteration variables (k and v)?

```
1 c = {'a':10, 'b':1, 'c':22}
2 for k, v in c.items() :
3     ...
```

- ☐ Because the keys for the dictionary are strings
- ☒ Because the items() method in dictionaries returns a list of tuples
- ☐ Because there are two items in the dictionary
- ☐ Because for each item we want the previous and current key

10. Given that Python lists and Python tuples are quite similar - when might you prefer to use a tuple over a list?

- ☐ For a list of items that want to use strings as key values instead of integers
- ☐ For a list of items that will be extended as new items are found
- ☒ For a temporary variable that you will use and discard without modifying
- ☐ For a list of items you intend to sort in place