Practice Midterm 2 - CS 261

Here are questions similar to what will be on the midterm exam. You won't be able to use any outside material during the midterm exam (no notes, computers, etc.) so see how much you can answer without looking things up. Be sure to study any Python rules and terminology that you don't know.

1. Suppose that `a = [6, 2, 2, 3, 5]`. What will the following commands output?

   (a) `print(set(a))`

   Solution: `{6, 2, 3, 5}`

   (b) `print(sorted(a))`

   Solution: `[2, 2, 3, 5, 6]`

   (c) `print(tuple(a))`

   Solution: `(6, 2, 2, 3, 5)`

   (d) `print(a.sort())`

   Solution: `None`

2. Suppose that

   ```
   d = {"Alfred": [1, 2, 3], "Bruce": (4, 5), "Selina": 6}
   ```

   What are the values of the following expressions? If there will be an error, briefly explain why.

   (a) `d["Alfred"]`

   Solution: `[1, 2, 3]`

   (b) `d["Bruce"][2]`

   Solution: Error, because the tuple `(4, 5)` has no value at index 2.

   (c) `"Selina" in d`

Solution: `True`

(d) `(4, 5) in d`

Solution: `False`

3. Determine the values of the variables `a`, `b`, `c` at the end of this code.

```
a, b = 5, 10
c = b + 2
a, b, c = c, a, b
```

Solution: `a = 12, b = 5, c = 10`

4. Consider the following code.

```
square = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

(a) For what values of `i` and `j` would `square[i][j]` be equal to 6?

Solution: `i = 1, j = 2`

(b) What is the value of `square[1]`?

Solution: `[4, 5, 6]`

(c) How could you increase the values of each number in `square` by 1? Write a nested loop that will do this.

Solution:

```
for i in range(3):
    for j in range(3):
        square[i][j] += 1
```

5. Suppose we run the following code.

```
a = [4, 7, 0, -2, 5]
b = a.sort()
c = sorted(a)
a.append(3)
```

What are the values of `a`, `b`, and `c` after this code is run?

Solution: `a = [-2, 0, 4, 5, 7, 3], b = None, c = [-2, 0, 4, 5, 7]`

6. Consider the following function which inputs a list of numbers and the code that follows it.

```
def mystery(xs):
    for i in range(1, len(xs)):
        xs[i] += xs[i - 1]

a = [1, 1, 1, 1, 1]
b = mystery(a)
print(a) # prints [1, 2, 3, 4, 5]
print(b) # prints None
```

Explain what the function `mystery()` does. What does it return? What side effects does it have?

Solution: After the loop, the value of the list `xs` at index `i` is the sum of the values of the list at the previous indices. So `mystery` mutates a list in place and fills it with the partial sums of the original input list. This function does not return anything and its only side effect is to mutate whatever list is passed into the function.

7. After the votes for each candidate have been counted, we need to figure out which candidate has the most votes. Complete the function `election_winner` below which inputs a dictionary `vote_counts` with the vote counts for each candidate and returns the name of the candidate with the most votes. If two or more candidates tie for the most votes, the function should just return the string `"tie"`.

   Here are examples of inputs and the correct outputs for this function.

Input	**Correct Output**
`{"Abe": 56, "Bob": 42, "Cal": 25}`	`"Abe"`
`{"Abe": 37, "Bob": 5, "Cal": 6, "Don": 37}`	`"tie"`
`{"Bob": 17, "Cal": 26, "Don": 37}`	`"Don"`

```
def election_winner(vote_counts):
    # Step 1: Find the integer max_votes which is the most votes anyone got.
```

Solution:

```
    max_votes = max(vote_counts.values())
```

```
    # Step 2: Make a list called top_keys with the candidates who got max_votes.
```

Solution:

```
    top_keys = [key for key, val in vote_counts.items() if val == max_votes]
```

```
    # Step 3: Return the correct output based on the name(s) in top_keys.
```

Solution:

```
    if top_keys == 1:
        return top_keys[0]
    else:
        return "tie"
```

8. Which of the following types are mutable? Circle the ones that are.

   A. Integers

   B. Lists

   C. Tuples

   D. Dictionaries

   E. Strings

   F. Sets

9. Suppose that we have a list of pairs of distinct integers, like this:

```
pairs_list = [(1, 2), (3, 5), (4, 7), (7, 4), (2, 4), (0, 1)]
```

   We will say that a pair `(x, y)` is *reversible* if both `(x, y)` and `(y, x)` are in the `pairs_list`. For example, the pair `(4, 7)` is reversible in the list above. Write a function called `count_reversible` that inputs a `pairs_list` and returns an integer equal to the total number of reversible pairs in the list. For example, your function should return 2 for the list above since the only reversible pairs are `(4, 7)` and `(7, 4)`. If the same pair appears more than once in a list, then it should only count once.

Solution:

```
def count_reversible(pairs_list):
    reversible_pairs = 0
    for pair in pairs_list:
        if (pair[1], pair[0]) in pairs_list:
            reversible_pairs += 1
    return reversible_pairs
```

10. For each of the following comprehensions, determine its value and state what type the output is.

(a) `{word: len(word) for word in ["apple", "banana", "cantalope"]}`

Solution: `{"apple": 5, "banana": 6, "cantalope": 9} # This is a dictionary`

(b) `[(n, n ** 2) for n in range(6)]`

Solution: `[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25)] # This is a list`

(c) `{c for c in "this is a string"}`

Solution: `{"t", "h", "i", "s", " ", "a", "r", "n", "g"} # This is a set`

11. Consider the following code.

```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width
    def area(self):
        return self.length * self.width

x = Rectangle(4, 5)
```

(a) Which of the following best describes `x`?

A. `x` is a method of the Rectangle class.
B. `x` is an instance of the Rectangle class.
C. `x` is an attribute of the Rectangle class.
D. `x` is the Rectangle class.

(b) What Python statement would print the width of `x`?

Solution: `print(x.width)`

(c) What Python statement would print the area of `x`?

Solution: `print(x.area())`

12. Why is it recommended to check if `__name__ == "__main__"` before running the contents of the main function in a Python program?

A. So that main will not run when you import the program as a module.
B. So that other functions that aren't named main won't get called accidentally.
C. It keeps the program from getting too complicated with too many functions.
D. It lets the program know which function is the most important one.