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.

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

   (b) print(sorted(a))

   (c) print(tuple(a))

   (d) print(a.sort())

2. Suppose that

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

   What are the values of the following expressions?
   (a) d["Alfred"]

   (b) d["Bruce"]

   (c) "Selina" in d

   (d) (4, 5) in d

3. Tuples and tuple assignment

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?

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

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?

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?

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.









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









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

8. In a school, each student can nominate others as their friends. These nominations are collected into a list of pairs, each pair representing a one-way friendship nomination (i.e., friendships are not necessarily reciprocated). Write the function `count_mutual` to analyze friendship nominations in a classroom. The function should accept one argument: a list of tuples nominations, where each tuple contains two strings representing the nominator and the nominee. The function should return an integer representing the total number of pairs who have mutually nominated each other as friends.

   A mutual friendship exists when student A nominates student B, and student B also nominates student A. Your function must accurately count these mutual relationships without considering repeated pairs. Assume all names are lowercase.

9. For each of the following Python expressions, write down the output when the expression is evaluated using a Python interpreter. Write **error** if you think the expression will raise an error.
   (a) 15 % 3

   (b) 5 + 3 * 4 == 100

   (c) "HSC" + "2024"

   (d) 100+"1"

   (e) 6 / 2

10. Suppose that we type the following assignments and expressions in a Python shell in the given order.

```
>>> a = 10
>>> b = 20
>>> c = a + b
```

   (a) What will Python output if we enter the following (after those first three statements)?

```
>>> a + 5
```

   (b) What will Python output if we enter this statement (after the one from part (a))?

```
>>> a + b
```

   (c) What will Python output when we enter these two statements next?

```
>>> b = b + 1
>>> c
```

   (d) What about if we enter this last?

```
>>> a + b
```

11. Consider the following function. What will it return if you call mystery(5)?

```
def mystery(n):
    a = n
    a + 5
    if a > 10:
        return a
    elif a == 10:
        return 100
    else:
        return n
```

12. Write a function called average3 that inputs any three numbers and returns their average.

13. Consider the simple function below.

```
def twice(n):
    print(2 * n)
```

Why do we get False if we enter the following into the shell? Explain why we get the output below.

```
>>> twice(5) == 10
10
False
```

14. Describe in words what the following function will do when you call it.

```
def loop_function():
    for i in range(100):
        if i % 3 == 0:
            print(i)
```

15. Identify the type of the value (str, int, float, bool, or list) for each of the following expressions.

(a) ”hello” * 5

(b) 7 // 2

(c) ”hello” == 5

(d) (”He” in ”Hello”) or (”Yes” in ”No”)

(e) 3 + 4 / 7

16. Determine the values of the variables x, y, and z after the following lines of code are run.

```
x = 5
y = x
z = 2 * y / 2
x += 4
y = (y - 1) * 2
x = (x == z + 4)
```

17. Consider the following recursive function:

```
def recurse(n, s):
    if n == 0:
        return s
    else:
        return recurse(n - 1, s + n)
```

   (a) What will recurse(3, 0) return?

   (b) What will happen if you call recurse(−1, 0)?