

## 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.

- Suppose that `a = [6, 2, 2, 3, 5]`. What will the following commands output?
  - `print(set(a))`
  - `print(sorted(a))`
  - `print(tuple(a))`
  - `print(a.sort())`
- Suppose that `d = {"Alfred": [1, 2, 3], "Bruce": (4, 5), "Selina": 6}` What are the values of the following expressions?
  - `d["Alfred"]`
  - `d["Bruce"]`
  - `"Selina" in d`
  - `(4, 5) in d`

### 3. Tuples and tuple assignment

### 4. Nested lists, accessing elements and changing rows and individual elements.

- In a local election, votes are cast by writing the name of a candidate on a ballot. At the end of the election, all votes need to be tallied to determine the winner. Write the function `tally_votes` to calculate which candidate received the most votes. The function should accept one argument: a list of strings `votes`, where each string represents a vote for a candidate. The function should return the name of the candidate who won the election. If there is a tie between two or more candidates with the highest number of votes, the function should return the string `"tie"`. Assume all candidate names are in lowercase. Here are examples of inputs and the correct outputs for this function.

Input	Correct Output
<code>tally_votes(["alice", "bob", "alice"])</code>	<code>"alice"</code>
<code>tally_votes(["alice", "bob", "tom", "jerry"])</code>	<code>"tie"</code>

Maybe have them complete the function? It seems a little too hard? Maybe if you have a dictionary with the votes as values, then have them return the candidate with the most votes? Maybe break this up into steps? Step 1 - Find the maximum number of votes any candidate received. Step 2 - find all keys in the dictionary that received the maximum number of votes. Step 3 - Create an if-then-else statement to return the correct answer.

- 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.

7. 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`

8. 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
```

9. 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
```

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

11. 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
```

12. 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)
```

13. 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`

14. 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)
```

15. 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)`?