COMP110

Quiz 3 Review Session

Content on Quiz 3

- Importing
- Unit tests
- Sets
- Dictionaries
- A little on time complexity

Disclaimer: We haven’t seen the quiz; this review session covers the main topics in the unit.

Importing

- syntax: from <file name> import <function>

- typically imports are at the TOP of the module
- when you import a function from a module, the entire module is run

Unit Tests

```
def test_name() -> None:
    # Other code can go here!
    assert <boolean expression>
```

- Used to test that a function works for a specific case
- Basically just a function with one assert statement - if your function is written correctly, it will pass
- We use pytest:
    - test file names should end in _test.py
    - test functions should begin with test_

Sets + Dictionaries

Sets
- Items in a set must be unique
- Sets are *unordered*

Dictionaries
- key-value pairs
- similar to list, but you "index" by keys
- keys are *unique*, values can be duplicates

	Lists	**Dictionaries**	**Sets**
Ordered?	yes	complicated (Python - yes)	no
Mutable?	yes	yes	yes
Initializing empty	x: list[int] = [] x: list[int] = list()	y: dict[str, int] = {} y: dict[str, int] = dict()	z: set[int] = set()
Initializing	x: list[int] = [0, 1]	y: dict[str, int] = {"blue": 0}	z: set[int] = {1, 2, 3}
Adding new values	x.append(0)	y["red"] = 0	z.add(0)
Updating	x[0] = 1	y["red] = 0	
Removing	x.pop(0)	y.pop("red")	z.remove(0)

for in, if in

- for <variable> in <list or dict>
  - LOOP
  - when looping through list → variable will represent the *items* in the list
  - when looping through dict → variable will represent the *keys* in the dict
  - doesn't matter what you name the variable!
- if <item> in <list or dict>
  - list → checks if <item> is an *element* in the list
  - dict → checks if <item> is a *key* in the dict

Code Writing

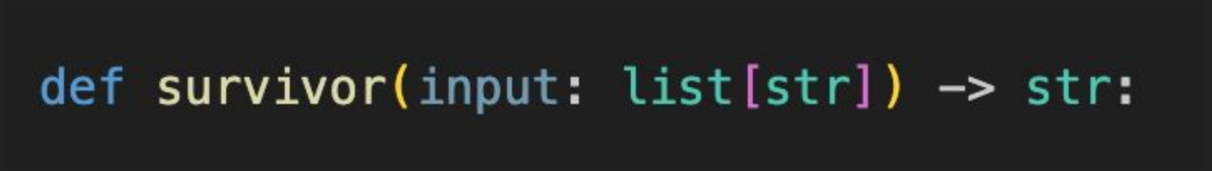
In the game show Survivor, votes are tallied to choose who should be eliminated from the island. Write a function called *survivor* that takes in a list of names and outputs the name of the person to be removed from the island. Your function should take in a *list[str*] and output a *str*.

example usage:

survivor(["Sophie", "Izzi", "Kaleb", "Sophie", "Izzi", "Sophie"])

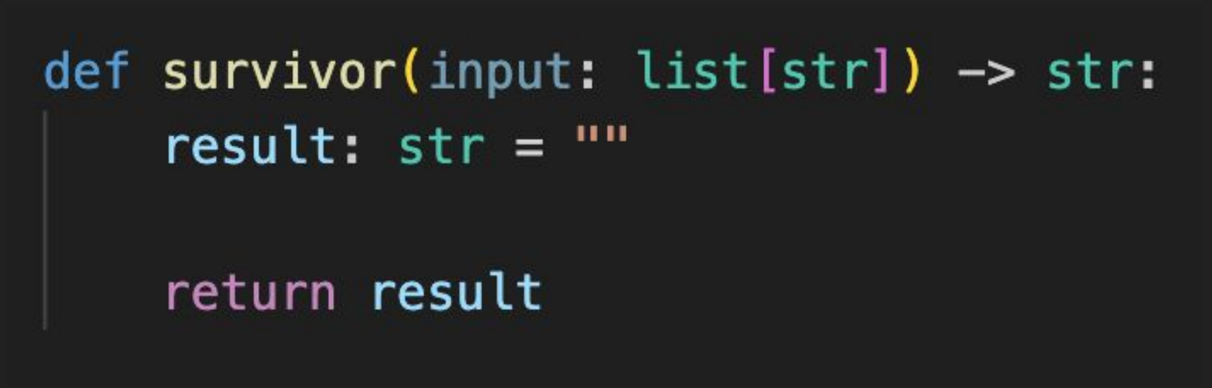
output: "Sophie"

Step 1 - Function signature



```python
def survivor(input: list[str]) -> str:
```

Step 2 - Skeleton function



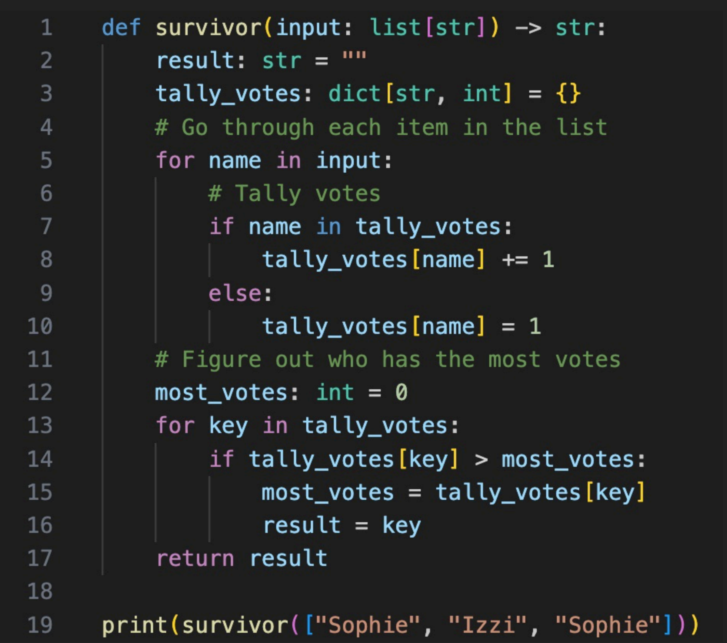
```python
def survivor(input: list[str]) -> str:
    result: str = ""

    return result
```

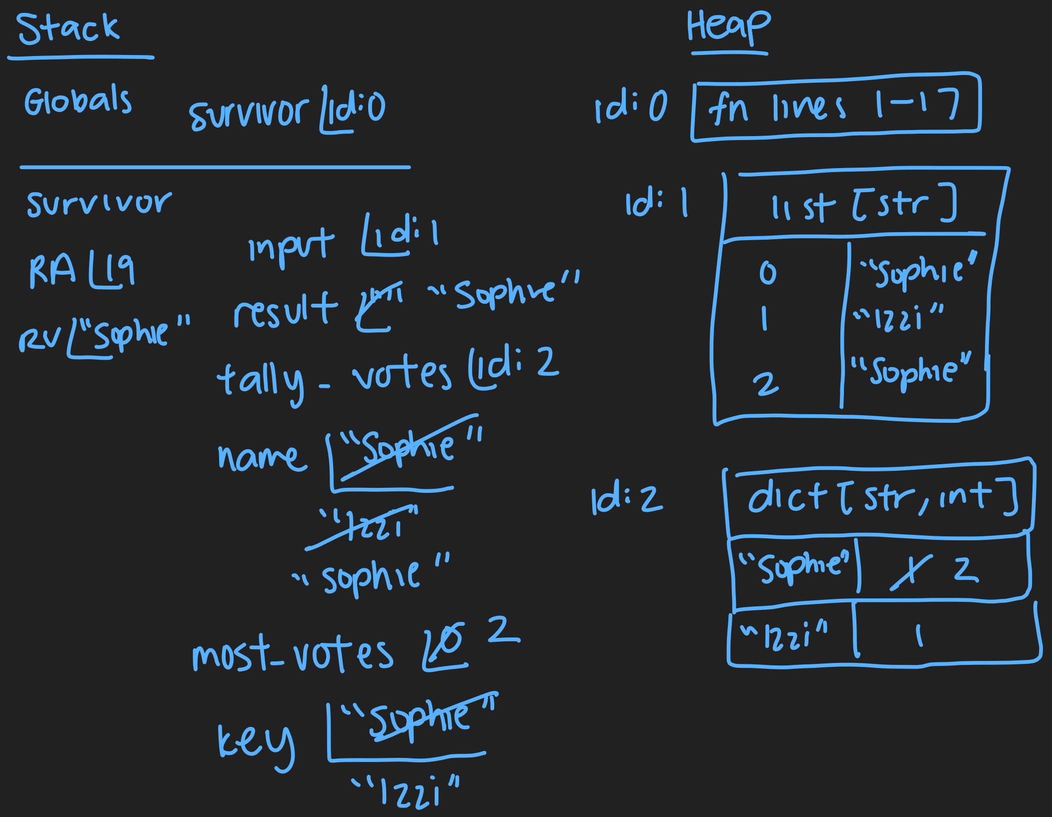
Step 3

Break down the problem - how would you solve it manually?

1. Go through each item in the list
2. Tally votes
3. Figure out who has the most votes:
    - Go through the tallied votes
    - Keep track of the highest count you've encountered so far
    - Keep track of the name of the person with the highest votes



```python
1   def survivor(input: list[str]) -> str:
2       result: str = ""
3       tally_votes: dict[str, int] = {}
4       # Go through each item in the list
5       for name in input:
6           # Tally votes
7           if name in tally_votes:
8               tally_votes[name] += 1
9           else:
10              tally_votes[name] = 1
11      # Figure out who has the most votes
12      most_votes: int = 0
13      for key in tally_votes:
14          if tally_votes[key] > most_votes:
15              most_votes = tally_votes[key]
16              result = key
17      return result
18
19  print(survivor(["Sophie", "Izzi", "Sophie"]))
```



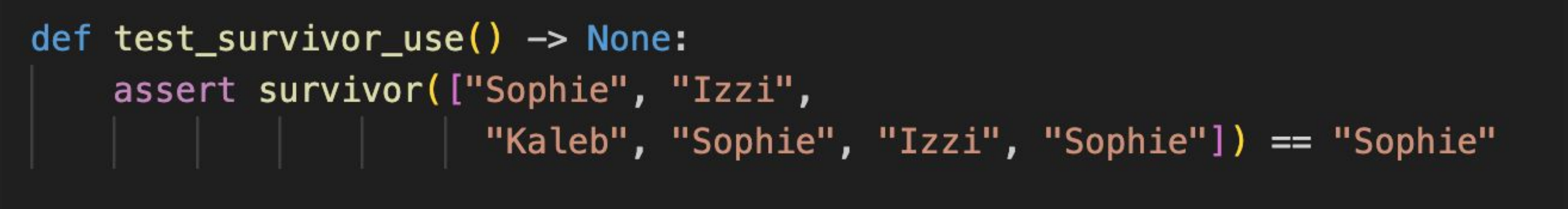
output

"Sophie"

Write a unit test for the function

Things to remember:

- a unit test should start with test_
- have an assert statement
- call the function with an input
- check that it is equal (==) to the result you expect



```python
def test_survivor_use() -> None:
    assert survivor(["Sophie", "Izzi",
                     "Kaleb", "Sophie", "Izzi", "Sophie"]) == "Sophie"
```

Questions?

Other Resources!

- Practice quiz on the course site with answers and explanations
    - We would recommend trying the problems out on your own, then checking your answers
- Tutoring
    - Thursday 3 - 5 in FB 141
- Office Hours
    - Tomorrow and Friday 11 - 5 in SN008