Task 1:

Imagine we have the variable `units` which stores the list `[ ["inch", "foot", "yard", "mile"], ["teaspoon", "tablespoon", "cup", "pint", "quart"] ]`

Using the `units` variable, write expressions to produce the following:

1. The string `"inch"`
2. The string `"h"`
3. The list `["inch", "foot", "yard", "mile"]`
4. The list `["cup", "pint"]`

Task 2:

Imagine we have the variable `numbers` which stores the list `[4353, 2314, 2956, 3382, 9362, 3900]`. Using list methods, do the following:

a. Remove 3382 from the list.
b. Get the index of 9362.
c. Insert 4499 in the list after 9362.
d. Extend the list by adding [5566, 1830] to it.
e. Reverse the list.
f. Sort the list.

Task 3:

Write a function, `somewhat_buggy_count(list, item1, item2)` that takes a list and two items to find in the list. Your function should return the sum of how many times both elements were found in the list, with a couple of exceptions:

- If `item1` is not in the list, return 6
- If `item2` is not in the list, return 4
- If both `item1` and `item2` are not in the list, return -3

The signature of this function is `(list, any, any) --> int`.