QUIZ • 30 MIN

Week 2 Exercise

Submit your assignment

DUE DATE Aug 23, 2:59 PM CST **ATTEMPTS** 3 every 8 hours

Receive grade

TO PASS 80% or higher

Week 2 Exercise

TOTAL POINTS 10

1. Consider this code: — 1 point

```
def high_low(guess, actual):
    """ (int, int) -> str

    Return "your guess is low" if guess is lower than actual,
    "your guess is high" if guess is higher than actual, and
    "correct" if guess is equal to actual.

    >>> high_low(4, 10)
    "your guess is low"
    """
```

Which set of test cases is the *best* choice of tests cases for this function? (Think about the **Boundaries** category for choosing test cases.)

- ○ • `guess` refers to `"higher"` and `actual` refers to `"lower"`
  - `guess` refers to `"correct"` and `actual` refers to `"correct"`
  - `guess` refers to `"lower"` and `actual` refers to `"higher"`
- ○ • `guess` and `actual` refer to negative values
  - `guess` and `actual` refer to zero
  - `guess` and `actual` refer to positive values
- ◉ • `guess` refers to a value that is less than the value referred to by `actual`
  - `guess` refers to a value equal to the value referred to by `actual`
  - `guess` refers to a value that is greater than the value referred to by `actual`
- ○ • `guess` and `actual` refer to odd values
  - `guess` and `actual` refer to zero
  - `guess` and `actual` refer to non-zero even values

2. Consider this code: — 1 point

```
def first_two_items(L):
    """ (list of str) -> list of str

    Return the first two items in L. If there are fewer than
    two items in L, return all of the items.

    >>> first_two_items(["apple", "pear", "grape"])
    ["apple", "pear"]
    """
```

Which of these sets of values for **L** is the *best* choice of tests cases for this function? (Think about the **Size** category for choosing test cases.)

- ◉ • `[]`
  - `["one"]`
  - `["one", "two"]`
  - `["one", "two", "three", "four"]`
- ○ • `[]`
  - `["one"]`
- ○ • `["one", "two"]`
  - `["one", "two", "three"]`
  - `["one", "two", "three", "four"]`
- ○ • `[]`
  - `["one"]`
  - `["one", "two"]`
  - `["one", "two", "three"]`
  - `["one", "two", "three", "four"]`
  - `["one", "two", "three", "four", "five"]`
  - `["one", "two", "three", "four", "five", "six"]`
  - More tests like this, with lists up to 20 items long.

3. Consider this code: — 1 point

```
def is_preschooler(age):
    """ (int) -> bool

    Precondition: age >= 0

    Return True if and only if age is between 3 and 5
    inclusive.

    >>> is_preschooler(4)
    True
    """
```

Which set of test cases is the *best* choice of tests cases for this function? (Think about which test case category you might want to consider.)

- ◉ • 0 or 1
  - 2
  - 3
  - 4
  - 5
  - 6
  - one number over 6
- ○ • one number under 0
  - 0
  - one number over 0
- ○ • one number between 0 and 2
  - 4
  - one number over 5
- ○ • one number between 0 and 2
  - one number between 3 and 5
  - one number over 5

4. Consider this code: — 1 point

```
def count_occurrences(s, ch):
    """ (str, str) -> int

    Precondition: len(ch) == 1

    Return the number of occurrences of ch in s.

    >>> count_occurrences("hello", "l")
    2
    """
```

Which of these sets of values for **s** and **ch** is the *best* choice of tests cases for this function? (There are several test case categories for this question, including at least **Size** and **Dichotomies**.)

- ◉ • `s` refers to '', `ch` refers to `'a'`
  - `s` refers to `'a'` `ch` refers to `'a'`
  - `s` refers to `'a'`, `ch` refers to `'b'`
  - `s` refers to `'abc'`, `ch` refers to `'b'`
  - `s` refers to `'abc'`, `ch` refers to `'d'`
  - `s` refers to `'abcabca'` `ch` refers to `'a'`
- ○ • `s` refers to '', `ch` refers to `'b'`
  - `s` refers to `'a'`, `ch` refers to `'b'`
  - `s` refers to `'aaaaaa'`, `ch` refers to `'b'`
- ○ • `s` refers to `''`, `ch` refers to `'a'`
  - `s` refers to `'a'`, `ch` refers to `'a'`
  - `s` refers to `'1'`, `ch` refers to `'1'`
  - `s` refers to `'a'`, `ch` refers to `'b'`
  - `s` refers to `'abc'`, `ch` refers to `'b'`
  - `s` refers to `'123'`, `ch` refers to `'2'`
  - `s` refers to `'abc'`, `ch` refers to `'d'`
  - `s` refers to `'abcabca'`, `ch` refers to `'a'`
- ○ • `s` refers to '', `ch` refers to `'a'`
  - `s` refers to `'a'`, `ch` refers to `'a'`
  - `s` refers to`'aaaaaa'`, `ch` refers to `'a'`

5. Consider this code: — 1 point

```
def can_vote(age):
    """ (int) -> bool

    Precondition: age >= 0

    Return True if and only if a person aged age can vote
    in Canada. The legal voting age in Canada is 18 years and
    older.
    """

    return age > 18
```