Problem 9

Bookmark this page

Problem 9

20.0 points possible (graded)

Write a Python function that takes in two lists and calculates whether they are permutations of each other. The lists can contain both integers and strings. We define a permutation as follows:

- the lists have the same number of elements
- list elements appear the same number of times in both lists

If the lists are not permutations of each other, the function returns False.
If they are permutations of each other, the function returns a tuple consisting of the following elements:

- the element occuring the most times
- how many times that element occurs
- the type of the element that occurs the most times

If both lists are empty return the tuple `(None, None, None)`. If more than one element occurs the most number of times, you can return any of them.

```
def is_list_permutation(L1, L2):
    '''
    L1 and L2: lists containing integers and strings
    Returns False if L1 and L2 are not permutations of each other.
            If they are permutations of each other, returns a
            tuple of 3 items in this order:
            the element occurring most, how many times it occurs, and its type
    '''
    # Your code here
```

For example,

- if `L1 = ['a', 'a', 'b']` and `L2 = ['a', 'b']` then `is_list_permutation` returns `False`
- if `L1 = [1, 'b', 1, 'c', 'c', 1]` and `L2 = ['c', 1, 'b', 1, 1, 'c']` then `is_list_permutation` returns `(1, 3, <class 'int'>)` because the integer 1 occurs the most, 3 times, and the type of 1 is an integer (note that the third element in the tuple is not a string).

Paste your entire function, including the definition, in the box below. Do not leave any debugging print statements.

```
# Paste your function here
```

Press ESC then TAB or click outside of the code editor to exit