

# Finger Exercises Lecture 13

The questions below are due on Wednesday October 26, 2022; 03:00:00 PM.

## 1) Question 1 of 1

Implement the function that meets the specification below.:

```
def sum_str_lengths(L):
    """
    L is a non-empty list containing either:
    * string elements or
    * a non-empty sublist of string elements
    Returns the sum of the length of all strings in L and
    lengths of strings in the sublists of L. If L contains an
    element that is not a string or a list, or L's sublists
    contain an element that is not a string, raise a ValueError.
    """
    # Your code here

# Examples:
print(sum_str_lengths(["abcd", ["e", "fg"]])) # prints 7
print(sum_str_lengths([12, ["e", "fg"]]))     # raises ValueError
print(sum_str_lengths(["abcd", [3, "fg"]]))    # raises ValueError
```

1 | # your function here

```
total = 0
try:
    for el in L:
        if type(el) is str:
            total += len(el)
        else:
            #assumed everything else would be a list instead of checking
            for sub_el in el:
                total += len(sub_el)
    return total
except:
    raise ValueError
```

You have infinitely many submissions remaining.

Here is the solution we wrote:

```
def sum_str_lengths(L):
    total = 0
    for i in L:
        if type(i) == str:
            total += len(i)
        elif type(i) == list:
            for e in i:
                if type(e) == str:
                    total += len(e)
                else:
                    raise ValueError
            else:
                raise ValueError
    return total
```

MIT OpenCourseWare  
<https://ocw.mit.edu>

6.100L Introduction to CS and Programming Using Python  
Fall 2022

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>