

## "Confidence Table" for function sum

depth	example	answer	correct?	confident?
0	3	3	✓	yes!
	0	0	✓	
	-100	-100	✓	
1	[ ]	0	✓	yes!
	[1, 4, 8]	13	✓	
	<p>these "sublists" are all depth 0</p> <p>For all nested lists of depth 1, each sublist must have depth 0</p>			

2 | [1, [2, 3], 10] | 16 ✓

For all nested  
lists of depth 2,  
each sublist must  
have 0 or 1 depth

```
def sum_nested(obj: Union[int, List]) -> int:
    """Return the sum of the numbers in a nested list <obj>.
    """
    if isinstance(obj, int):
        # obj is an integer
        return obj
    else:
        # obj is a list of nested lists: [lst_1, ..., lst_n]
        s = 0
        for sublist in obj:
            # each sublist is a nested list
            s += sum_nested(sublist)
        return s
```