

```
# sum_numbers3.py
#
# ICS 32 Winter 2014
# Code Example
#
# This function sums the integers in a list containing *either* integers
# of sublists that are flat lists of integers.
#
# One interesting new tidbit here that you might not have seen is that you
# check the type of an object.  type(x) is a function that returns the
# type of the object x; you can compare types using == the same way you
# can compare other kinds of objects.  We use that to check whether an
# object is a list, like this:
#
#     if type(element) == list:
#
# which allows us to distinguish between elements that are integers and
# elements that are sublists.

def sum_numbers(numlist: [int or [int]]) -> int:
    '''
    Adds up the integers in a list whose elements are either integers or
    lists of integers
    '''
    sum = 0

    for element in numlist:
        if type(element) == list:
            for num in element:
                sum += num
        else:
            sum += element

    return sum

assert(sum_numbers([[1, 2, 3], 4, [5, 6], 7, 8]) == 36)
assert(sum_numbers([1, 2, 3]) == 6)
assert(sum_numbers([[1, 2], [3, 4], [5, 6]]) == 21)
assert(sum_numbers([[1], [2], [3]]) == 6)
assert(sum_numbers([[8]]) == 8)
assert(sum_numbers([], [], [], [])) == 0)
assert(sum_numbers([]) == 0)
```