Raindrops Keep Falling on our Sensors

The University of Chicago Weather Service has purchased a new rainfall sensor. This sensor takes measurements every few seconds, which can be handed off to a computer for further analysis. Each measurement is a non-negative integer representing how many millimeters of rainfall have been recorded since the previous measurement.

Given this information, we would like to compute the average rainfall, which is simply the average of the non-negative integers produced by the sensor. However, there's one catch: this new sensor occasionally produces *negative* integers representing faulty measurements. These have to be discarded and shouldn't be taken into account.

Your task is to complete the function `raindrops`, which takes a list of integer measurements. If the input contains only negative numbers, your function should return the null value appropriate for your implementation language. Otherwise, your function should return the average rainfall for the dataset *rounded down* (that is, only the integer part of the average), taking into account that negative measurements should be discarded.

Here are some sample inputs and outputs:

Sample Input	Sample Output
5 10 15	10
-10 -6 -7	null value
14 -5 39 -5 7	20

Here is the skeleton code for this task:

```
'''
Distribution file of the raindrops problem.
'''

from typing import List
from typing import Union

def raindrops(measurements: List[int]) -> Union[int, None]:
    '''
    Computes the average rainfall given a list of measurements. The function
    only considers measurements that are greater than or equal to 0. If only
    negative measurements are given, the function returns None.

    Arguments:
        measurements: A list of measurements.

    Returns:
        The average rainfall as a rounded down integer. If only negative
        measurements are given, the function returns None.
    '''
    # TODO: Implement this function.
    pass
```

When you take the placement exam, you will be expected to copy the skeleton
code into a file and then complete the function. For the practice problems, we
have provided a file named problem2.py that includes the above header for your
convenience.