Exercise 5: `count_trips` (2 points)¶

For the next filtering step, we need to count how many trips there are between pairs of zones. Recall that the `'I'` and `'J'` columns of `trips` (and `trips_clean`) indicate the starting and ending zones, respectively.

Let `trip_coords` be a dataframe consisting of just two columns, `'I'`, and `'J'`, taken from `trips` or `trips_clean`, for instance.

Complete the function, `count_trips(trip_coords, min_trips=0)`, so that it counts the number of start/end pairs **and** retains only those where the count is at least a certain value.

That is, the inputs are

- `trip_coords`: a pandas `DataFrame` with two columns, `'I'` and `'J'`, indicating start/end zone pairs
- `min_trips`: the minimum number of trips to consider (the default is 0, meaning include all pairs)

The function should return a pandas `DataFrame` object with three columns derived from the input `trip_coords`:

- `'I'`: the starting zone;
- `'J'`: the ending zone; and
- `'N'`: the number of trips originating at `'I'` and ending at `'J'`.

Your function should only include start-end pairs where `'N'` >= `min_trips`.

For instance, suppose the input dataframe `trip_coords` is the following:

	I	J
	139	28
	169	51
	231	128

169	51
169	51
169	51
139	28
85	217
231	128
231	128

Then if `min_trips=3`, your function would return

I	J	N
169	51	4
231	128	3

which omits the pairs (85, 217) and (139, 28) since they appear only once and twice, respectively.

Note: If no pair meets the minimum trips threshold, your function should return a `DataFrame` with the required columns but no rows.

In [33]:

```
def count_trips(trip_coords, min_trips=0):
    ### BEGIN SOLUTION




    ### END SOLUTION
```