Programming Assignment #5: Autograder Visible Test Cases' Inputs

Changelog

- v.1: Initial version.

Part #1

Cases #7 through #10 are hidden.

Case #1

```
>>> count_in_other([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [2, 3, 4, 5, 6, 9,
10, 11, 13])
8
```

Case #2

```
>>> count_in_other([17, 5, 8, 20, 4, 14, 15, 18], [6, 17, 20, 3, 15, 5])
4
```

Case #3

```
>>> count_in_other([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [13, 11, 10, 9, 6,
5, 4, 3])
7
```

Case #4

```
>>> count_in_other([40, 80, -3, 70, 20, 15, 17], [80, 17, 15, 12, -3])
4
```

Case #5

```
>>> vals = list(range(1,30))
>>> vals
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22,
23, 24, 25, 26, 27, 28, 29]
>>> count_in_other(vals, [12, 120, 1200])
1
```

Case #6

```
>>> vals = list(range(1,30))
>>> count_in_other(vals, [120, 1200])
0
```

Part #2

Cases #9 through #14 are hidden.

Case #1

```
buy_items({"item X": 30,
                            "item Z": 10, "item W": 8},
                           {"item X": 6, "item Z": 4,
                            "item W": 2},
                           {"item X": 2,
                            "item Z": 3, "item W": 4})
```

RETURN: 32

Case #2

```
buy_items({"item X": 30, "item Y": 6,
                            "item Z": 10, "item W": 8},
                           {"item X": 25, "item Z": 2},
                           {"item X": 8,
                            "item Z": 2, "item W": 1})
```

RETURN: -2

Case #3

```
buy_items({"item X": 20, "item Y": 6,
                            "item Z": 4, "item W": 8},
                           {"item X": 25, "item Z": 2},
                           {"item X": 8, "item Y": 5,
                            "item Z": 2, "item W": 1})
```

RETURN: -3

Case #4

```
buy_items({"item X": 20, "item Y": 6,
                            "item Z": 4, "item W": 8},
                           {"item X": 0, "item Y": 2},
                           {"item X": 8, "item Y": 5,
                            "item Z": 2, "item W": 1})
```

RETURN: -1

Case #5

```
buy_items({"item X": 20, "item Y": 6,
                            "item Z": 4, "item W": 8},
                           {"item X": 1, "item Y": 2},
                           {"item X": 8, "item Y": 5,
                            "item Z": 2})
```

RETURN: -2

Case #6

```
buy_items({"item X": 20, "item Y": -6,
                            "item Z": 4, "item W": 8},
                           {"item X": 1, "item Y": 2},
                           {"item X": 8, "item Y": 5,
                            "item Z": 2})
```

RETURN: -1

Case #7

```
buy_items({"item X": 20, "item Y": 6,
                            "item Z": 4, "item W": 8},
                           {"item X": 1, "item Y": 2},
                           {"item X": 8, "item Y": 5,
                            "item Z": 2, "item W": 10})
```

RETURN: 18

Case #8

```
buy_items({"item X": 17, "item Y": 12,
                            "item Z": 4, "item W": 8},
                           {"item X": 2},
                           {"item X": 8, "item Y": 4,
                            "item Z": 1, "item W": 10})
```

RETURN: 16

Part #3

Cases #4 and #5 are hidden.

Case #1

```
>>> count_less_than([[9,5,7,2],[3,2,5],[1,5,8]], 4)
4
```

Case #2

```
>>> count_less_than([[9,5,7,2],[3,0,5],[1,4,8]], 4)
-1
```

Case #3

```
>>> count_less_than([[8],[18,20,-5,2,12,10],[1,2,3,4],[5,7],[19,23,41]], 11)
10
```

Part #4

Cases #3 and #4 are hidden.

Case #1

```
        seating_chart = [["Timmy","Cosmo","Wanda"],["Jimmy","Cindy","Carl"]]
        attendance = [["","","Wanda"],["Jimmy","Cindy","Jimmy"]]
        retval = count_absences(seating_chart,attendance)
```

`retval` should be 2

Case #2

```
        seating_chart = [["Squidward"]]
        attendance = [["Squidward"]]
        retval = count_absences(seating_chart,attendance)
```

`retval` should be 0.

Part #5

Cases #4 and #5 are hidden.

Case #1

```
>>> find_in_other([17,20,5],[3,18,5,6,5,16,17,18,19,20,21])
[[6], [9], [2, 4]]
```

Case #2

```
>>> find_in_other([18,5,-2,5,20],[3,18,5,6,5,16,17,18,19,20,21])
[[1, 7], [2, 4], [], [2, 4], [9]]
```

Case #3

```
>>> find_in_other([2,10],[15,10,5,10,15,20,25,20,15,10])
[[], [1, 3, 9]]
```

Part #6

Cases #5 and #6 are hidden.

Case #1

```
>>> count_has_k_even_divisors(11, 4)
3
```

Case #2

```
>>> count_has_k_even_divisors(10, 20)
0
```

Case #3

```
>>> count_has_k_even_divisors(17, 5)
1
```

Case #4

```
>>> count_has_k_even_divisors(17, 2)
7
```