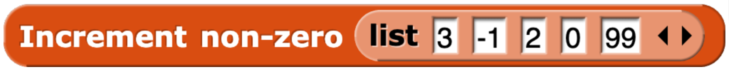
CS10 With-Computer Final (Spring 2019, Sec 1)

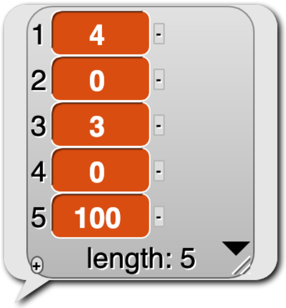
There are three questions, two Snap*!* ones and a Python one. Save your Snap*!* code into a Snap*!* file, and name it **`FinalYourfirstnameYourlastname.xml`** (e.g., **`FinalAlanTuring.xml`**). For the Python question, create a new Python file and name it **`FinalYourfirstnameYourlastname.py`** (e.g., **`FinalAlanTuring.py`**). Submit both files on GradeScope under the "online" final assignment for your lab section. All questions are independent, and each worth 5.

Snap*!* Questions: (use this starter file: http://bit.ly/2DCCPql)

We want a block that takes a list of numbers and returns a list in which 1 has been *added* to all the non-zero numbers., e.g.,



```
Increment non-zero list 3 -1 2 0 99
```



```
1  4
2  0
3  3
4  0
5  100
length: 5
```

a) Write it *recursively*. You may not use any iteration (**`repeat`**, **`repeat until`**, **`for`**, **`for each`**) or higher-order functions in this solution.

b) Write *without using recursion*. You can earn +3 bonus points if you can do it with only higher-order functions (i.e., only **`map`**, **`keep`** and **`combine`** to drive the iteration). You may write a single helper function if you need it.

Python Question:

We want to know *which TAs had at least half of their discussions full*. We have two dictionaries:

- *enrollment*: represents the number of students enrolled in each discussion
  - **Key**: TA name
  - **Value**: # of enrolled students
- *attendance*: represents the attendance for each discussion
  - **Key**: TA name
  - **Value**: list representing the number of students that attended that discussion each week

Assume we had 10 discussions. Write the function **`TAs_with_at_least_half_discussions_full`**, that returns a list of the names of TAs who had at least half (i.e., ≥ 5) of their discussions full.
(We underline the full discussions for each TA, and the TAs with at least half of them full below.)

```
>>> sp19_enrollment = {"Murtaza": 10, "Lara": 20, "Mansi":20, "Niki":15, "Brendan": 10}
>>> sp19_attendance = {"Murtaza": [10,  9, 10, 10, 10,  2,  9, 10,  9,  1],
                       "Lara":    [19, 18, 16, 14, 12, 11, 10,  5,  5,  5],
                       "Mansi":   [25, 22, 23, 24, 25, 22, 22, 20, 25, 25],
                       "Niki":    [20, 15, 15, 12, 17, 17, 17, 16, 18, 20],
                       "Brendan": [ 5,  5,  4,  2,  3, 10, 11, 12,  4,  2] }
>>> sp19_enrollment["Lara"]     # get the enrollment for Lara's discussion
20
>>> sp19_attendance["Lara"][0]  # get the attendance for Lara's first discussion
19
>>> TAs_with_at_least_half_discussions_full(sp19_enrollment, sp19_attendance)
['Murtaza', 'Mansi', 'Niki']
```