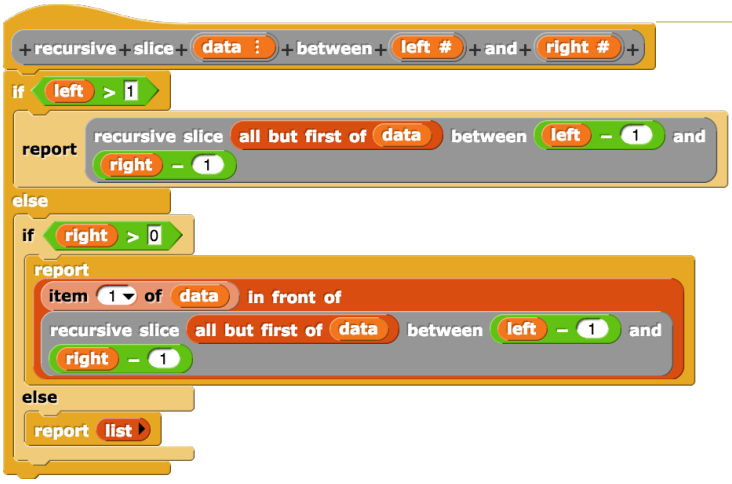
CS10 With-Computer Final (Fall 2017) Solutions

Snap*!* Questions: (use this starter file: http://bit.ly/2zzzxA5)

You want to replicate Python's list "slice" in Snap*!*. However, it should follow Snap*!*'s convention to index lists starting from 1 and include the rightmost element. You don't have to handle the case when the inputs are blank or do any error checking. That is, assume the left number ≤ the right number, and that both numbers are between 1 and the list length. If the numbers are equal, it returns a list of the element at that index.

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



Let's first outline our approach and the intuition behind it:

We only want to keep the items of our data that are between the left index and the right index. That means we need:
1. A way to tell when we should start keeping items (i.e. when we have reached a point where the indices of the items in our list are greater than the left index).
2. A way to tell when we should stop keeping items (i.e. when we have reached a point where the indices of the items in our list are greater than the right index).

How we will do this:

Every time we recurse, we will recurse on all but first of our data, decreasing the left and right indices every time so they give us the indices we should be considering in our new list. Ex: If we are trying to get indices 2-4 from the list [1, 2, 3, 4, 5], it is the same problem as getting indices 1-3 from [2, 3, 4, 5].

Then, we know we should start keeping values from our list as soon as left < 2. So once left < 2, we should start keeping the first item of our list, which we will do using:
(item 1 of data) in front of (recursive slice (all but first of (data)) between (left - 1) and (right - 1)

Since we only want to get the items between left and right, we should start keeping values once our right index is less than 1 (i.e. the last item to keep is not in the list anymore).

b) Write it using higher-order functions (only **map**, **keep** and **combine**). One helper you might find handy is the "**numbers between** () **and** ()" block.

Python Question:

Write a function that returns the *first duplicate word* of an essay whose words are all in lowercase (with no punctuation). If there are no duplicates, return the empty string. You *must* use a dictionary in your solution; if you forget any commands, remember there's **help**(*type*)and **dir**(*type*), as in **help(dict)** or **dir(str)**. To split a string into a list of words, you might find string's **split** command helpful.

first_duplicate("ask not what your country can do for you ask what")➔"ask"

first_duplicate("cs ten is the best class at cal")➔""

```
def first_duplicate(essay):
      dict = {}
      for word in essay:
            if word in dict:
                  return word
            else:
                  dict[word] = 1
      return ‘’
```