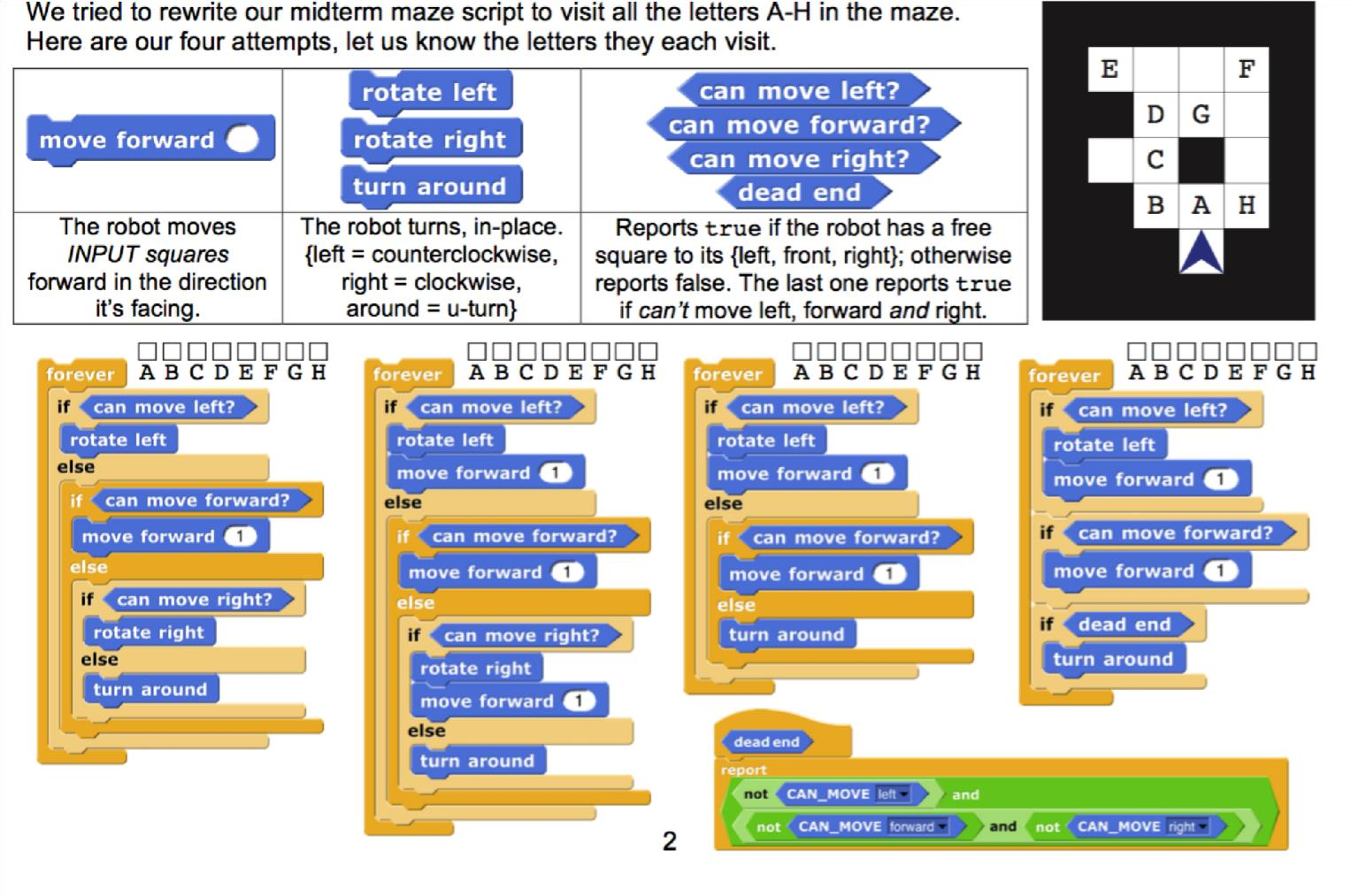
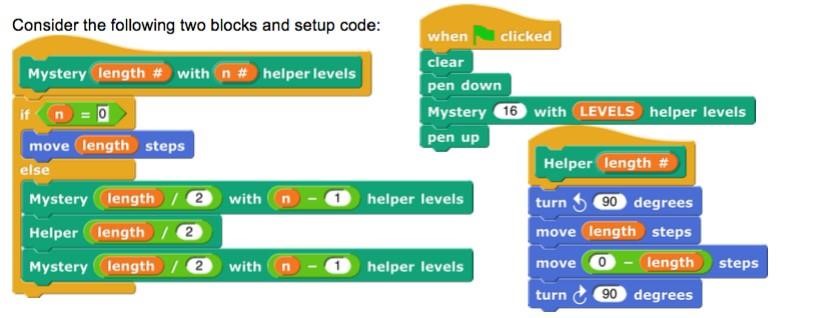
**Discussion 14: Final Review**

# Drawing/Movement in Snap

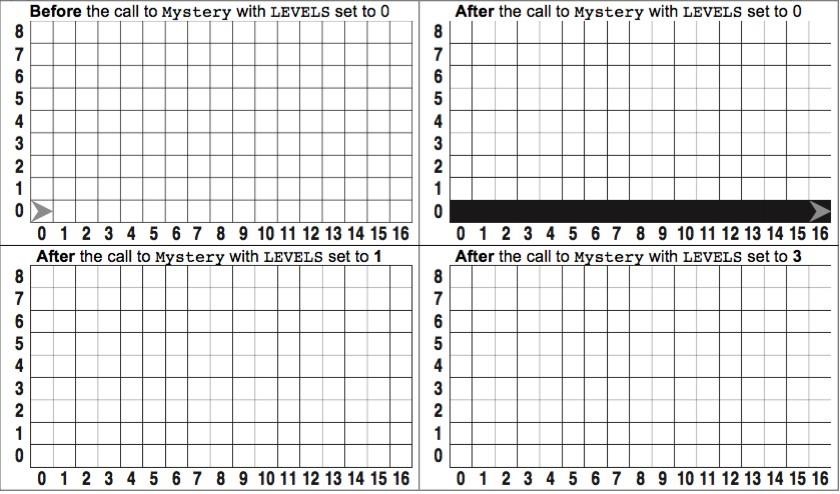
## Question 1: Mr. Robot



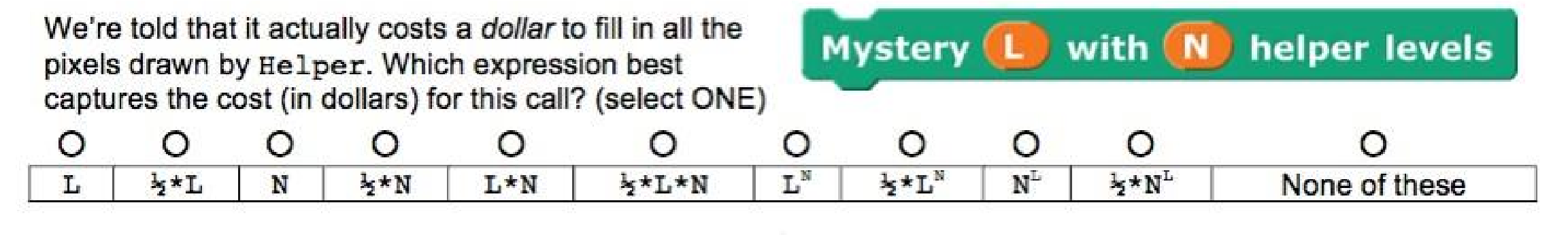
## Question 2: Magical Mystery Tour



**a.** Now, given that the sprite starts out in the bottom left corner facing right, and that the pen is in the middle of the sprite, shade in the pixels that will be colored after calls to Mystery with levels set to 1 and levels set to 3. You may use the top left grid for scratch work. Levels = 0 has been given to you.



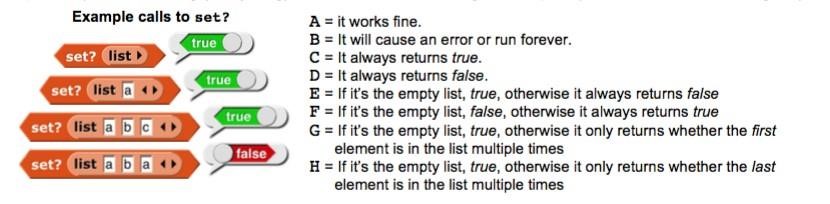
**b.**



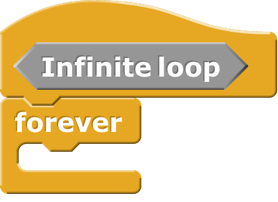
# Recursion

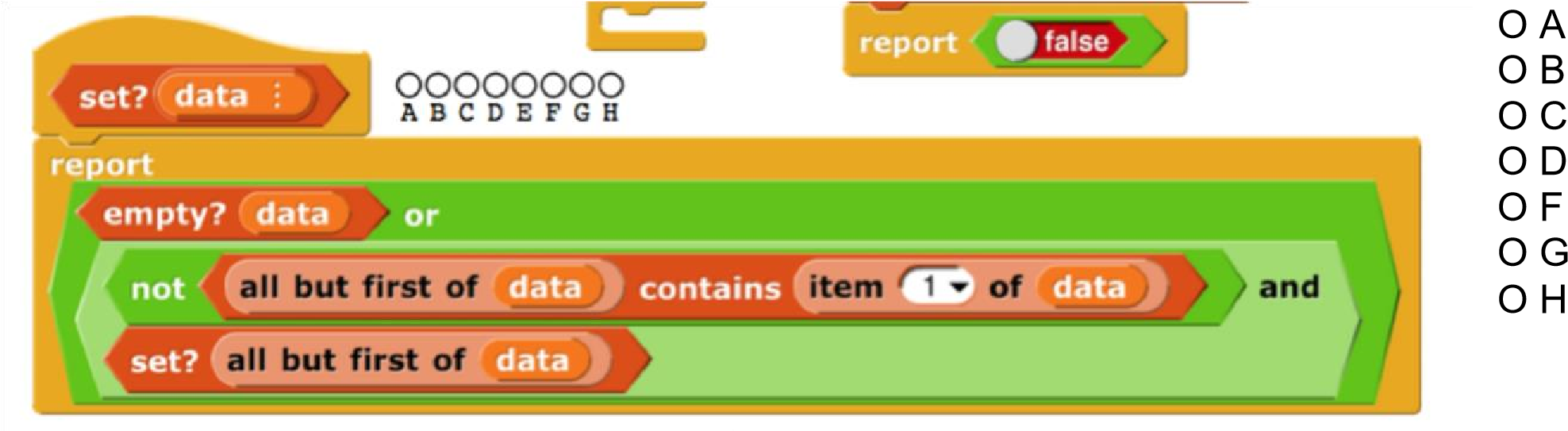
**Question 1: Ready, Set, Go!**

In this problem, we have created three different blocks to see if a given list is a set, that is, it has no duplicates. For each of the blocks below, select one of the following answer choices:

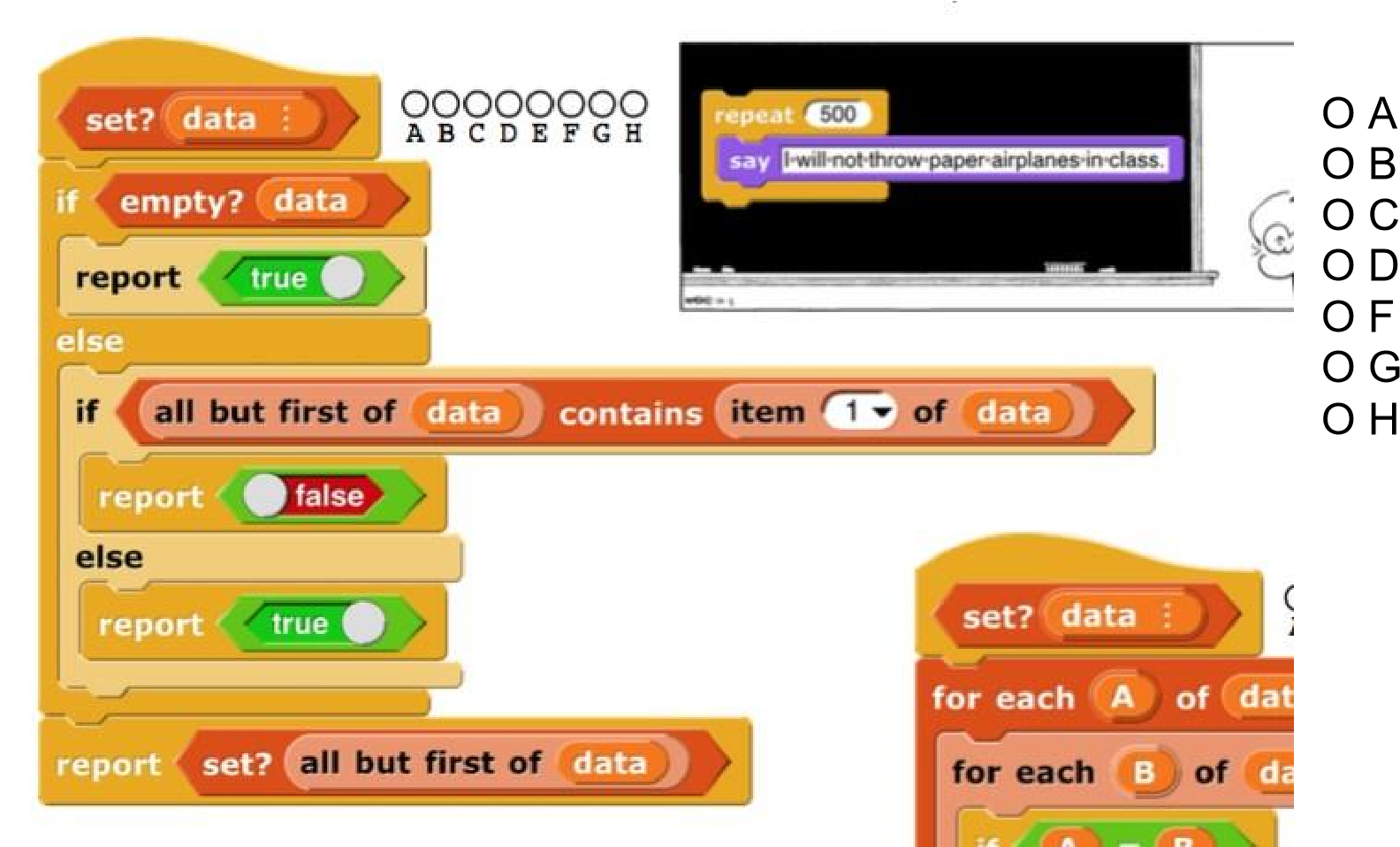


**a.** For this subpart, note that the *or* and *and* blocks don’t even look at their right input if the left one is true or false, respectively. For example,

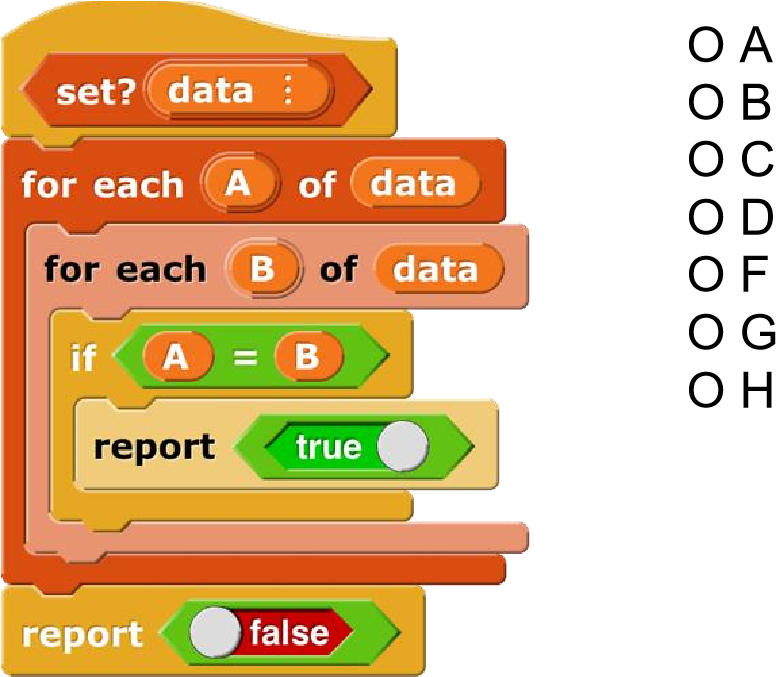




**b.**

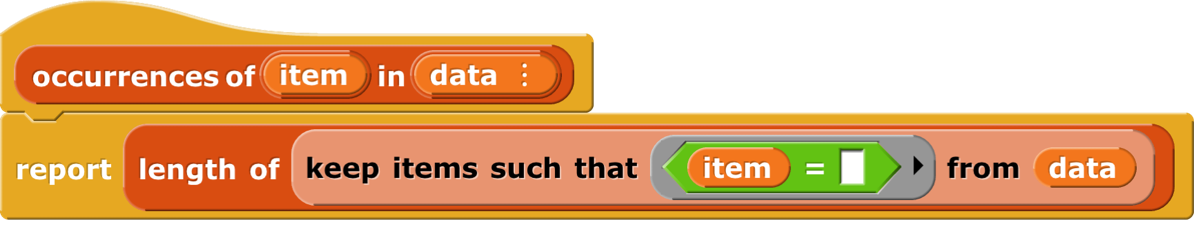


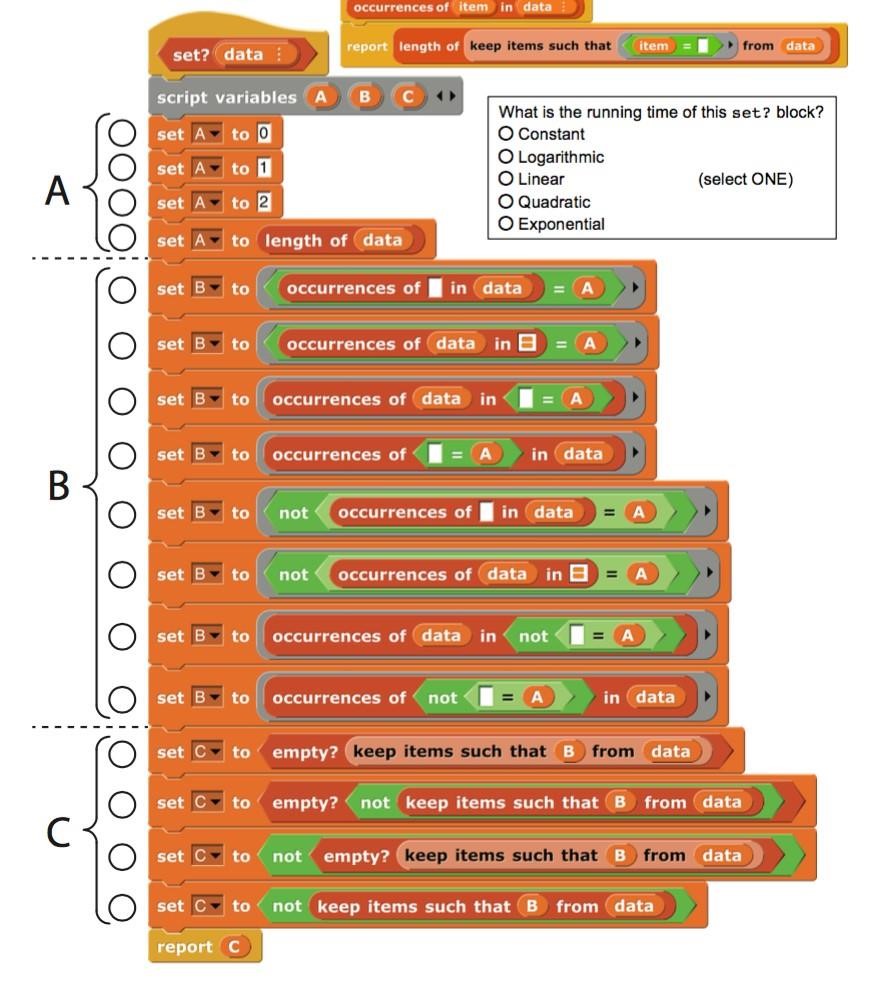
**c.**



## Question 2: Constructing the *set* block

How could we construct the *set* block using the following *occurrences of* block? Note that you may only choose one option from each section A-C.





# Python

## Question 1: Syntax

Write the output of the following lines of code.

>>> [‘cal’, ‘berkeley’, ‘stanford’][1][2]

r

>>> [x\*10 for x in range(3) if x != 1]

[0, 20]

## Question 2: Reversing a Dictionary

We want to write a dictionary reverser that takes in a dictionary and returns a new dictionary with the original values as the new keys and the original keys as a list of values.

>>> dictionary\_ reverser ({1:3, 2:3, 8:9})

{3: [1, 2], 9: [8]}

Write this function by filling in the blanks in the skeleton code below.

*def* dictionary\_reverser(*dict*):

  r = {}

  for k in *dict*:

    if *dict*[k] in r:

        r[*dict*[k]].append(k)

    else:

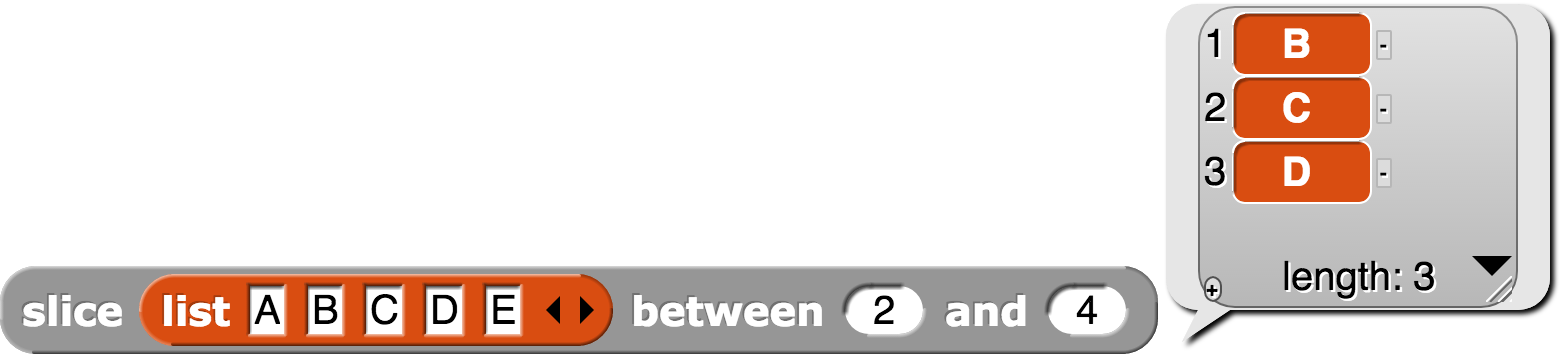
          r[*dict*[k]] = [k]

  return r

# Online Final Questions

## \*\*Note: You should complete all of the below questions either on a separate sheet of paper or on your computer. There is not sufficient space to write the solutions here.\*\*

**Question 1: Slicing in Snap!**

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.

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



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



## Question 2: Strings and Dictionaries in Python

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.split():

       if word in *dict*:

             return word

       else:

*dict*[word] = 1

          return ''