Discussion 14: Final Review

Drawing/Movement in Snap

Question 1: Mr. Robot
We tried to rewrite our midterm maze script to visit all the letters A-H in the maze. Here are our four attempts, let us know the letters they each visit.

- move forward
- rotate left
- rotate right
- turn around
- can move left?
- can move forward?
- can move right?
- dead end

The robot moves INPUT squares forward in the direction it's facing.
The robot turns, in-place. (left = counterclockwise, right = clockwise, around = u-turn)
Reports true if the robot has a free square to its (left, front, right); otherwise reports false. The last one reports true if can't move left, forward and right.

Question 2: Magical Mystery Tour
Consider the following two blocks and setup code:

- Mystery length # with n # helper levels
- move length steps
- Helper length #
- move length steps
- move 0 - length steps
- turn 90 degrees
- Mystery length / 2 with n - 1 helper levels
- Mystery length / 2 with n - 1 helper levels
- turn 90 degrees
- Mystery 16 with LEVELS helper levels
- pen up
- Mystery length / 2 with n - 1 helper levels
- Mystery length / 2 with n - 1 helper levels
- turn 90 degrees
- move length steps
- move length steps
- move length steps
- Mystery length / 2 with n - 1 helper levels
- turn 90 degrees
- Mystery length / 2 with n - 1 helper levels
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.

![Grids showing before and after calls to Mystery]

b. We’re told that it actually costs a dollar to fill in all the pixels drawn by helper. Which expression best captures the cost (in dollars) for this call? (select ONE)

![Mystery block with options]

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 = it works fine.
- B = it will cause an error or run forever.
- C = it always returns true.
- D = it always returns false.
- E = if it’s the empty list, true, otherwise it always returns false
- F = if it’s the empty list, false, otherwise it always returns true
- G = if it’s the empty list, true, otherwise it only returns whether the first element is in the list multiple times
- H = if it’s the empty list, true, otherwise it only returns whether the last element is in the list multiple times
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,

```
true or Infinite loop
false and Infinite loop
```

---

b. 

```
set? data:
if empty? data:
  report true
else:
  if all but first of data contains item 1 of data:
    report false
  else:
    report true
report set? all but first of data
```

---

c. 

```
set? data:
for each A of data:
  for each B of data:
    if A = B:
      report true
report false
```
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.

```python
>>> ['cal', 'berkeley', 'stanford'][1][2]
```

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

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.

```python
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.

```python
def dictionary_reverser(dict):
    r = {}
    for k in dict:
        if _______________ in ________________:
            ___________________.append(_______________)
        else:
            ___________________ = ___________________
    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.

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 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.

```python
>>> first_duplicate("ask not what your country can do for you ask what")
"ask"
>>> first_duplicate("cs ten is the best class at cal")
""
```