Discussion 9B: Midterm Review Solutions

Recursion

1. In the space below, write a block that recursively converts binary to decimal. You may find the following blocks helpful:

2 ^ 3 ⁸ all but first letter of 1101 ¹⁰¹ length of 1101 ⁴
<pre>convert (bitstring) to decimal if length of bitstring = 1 report bitstring else</pre>
report letter 1 v of bitstring × 2 ^ length of bitstring - 1 + Convert all but first letter of bitstring to decimal

O Constant	O Logarithmic	Linear	O Quadratic	O Exponential
	-			

2. You have just gotten into a relationship and have written the block love to track your affection through your relationship (x = kisses, o = hugs, - = hanging out). The reverse block, which reverses its input text, has been provided to you.

+ if el:	+ love + day # + if day < 3 report . else report join x love day - 1 . reverse love day - 2 0 .	yelekreb
a.	a. What is love 3 ? x o	
b.	b. What is love 4 ? x x 0 0	
c.	. What are the first three and last three letters of love 9999 ? x x x, x x o	
d.	I. Which of the following are possible?	
	\Box Hanging out four times in a row (i.e. "")	
	Hug immediately followed by a kiss (i.e. "ox")	
	□ Hug twice in a row (i.e. "oo")	

3. Short Changed

When you get change back, you don't care about how many possibilities there are (silly **Count Change**). What you do want to know is what are the fewest number of coins you need to carry in your pocket. Write the **Short Changed** block that will return the minimum number of coins you can make change with.

+ Short + Changed + amount + using + coins : +
if is coins empty? report 0
else if amount - item 1 > of coins < 0
report Short Changed amount using all but first of coins
else
report
1 + Short Changed amount - item 1 of coins using coins

Hint 1: We don't need to use a minimum block.

Hint 2: This block has a few similarities with **Count Change**, but has its differences as well.

Hint 3: This block can be solved in $101_2 + (0x7B - 0b11) \div 100_2 \div F_{16}$ lines of code.

Mutability

1. We've created the copy of block to make a copy of a list, but it doesn't work as we expected. Write out what the sprite would say after the following blocks of code have been run.





2. We've created a block to square all the numbers in a list, as shown to the right. However, our tests are giving us puzzling results. Answer the questions below to help us understand the behavior of our block.



a. When we run the script below, the sprite first says "true", but then says "false." Why does this happen, and how can we fix it?



(test list) is set to [1, 2, 3] and square numbers correctly takes in (test list) and reports [1, 4, 9]. The problem is that in square numbers – since we never copied the data within the list, but instead copied the list itself – we were **mutating** (test list) the entire time. For the second test, the input (test list) is equal to [1, 4, 9], and so the output we should be expecting for the second test is the list [1, 16, 81].

b. Assuming we haven't changed the square numbers block, what do we expect the sprite to say when we run the code below? Why is this different from part a?



The sprite should now say (true) (true). This is because in part a, the list (test list) was mutated after going through square numbers the first time. Here, we have two lists that have the same elements as inputs, but are two different lists – rather than the same list (test list). This ensures that the mutating of the input list that takes place in the first test won't have an effect on the second test.