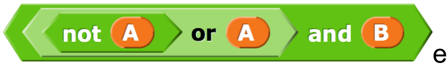
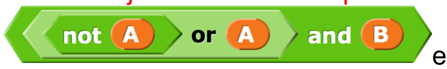
CS10 Fall 2017 Midterm 2 Answers

Question 1a: If 

```
((not A) or A) and B
```

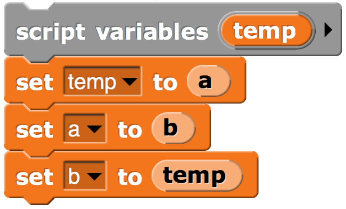
evaluates to **true**, what can you say about **A** and **B**? not A or A is always **true**. **true** and B is just B. So if the expression is **true**, B must be **true**.

Question 1b: If 

```
((not A) or A) and B
```

evaluates to **false**, what can you say about **A** and **B**? not A or A is always **true**. **true** and B is just B. So if the expression is **false**, B must be **false**.

Question 2: A code segment will be used to swap (i.e., exchange) the values of the variables **a** and **b** using the temporary variable **temp**. Which of the following code segments correctly *swaps the values* of **a** and **b**? (select ONE)



```
script variables temp
set temp to a
set a to b
set b to temp
```

Question 3: Which are the possible *values* of global variable **N** at the end of this program? (select all that apply). Left is the "+1" script, Right is the "square" script.

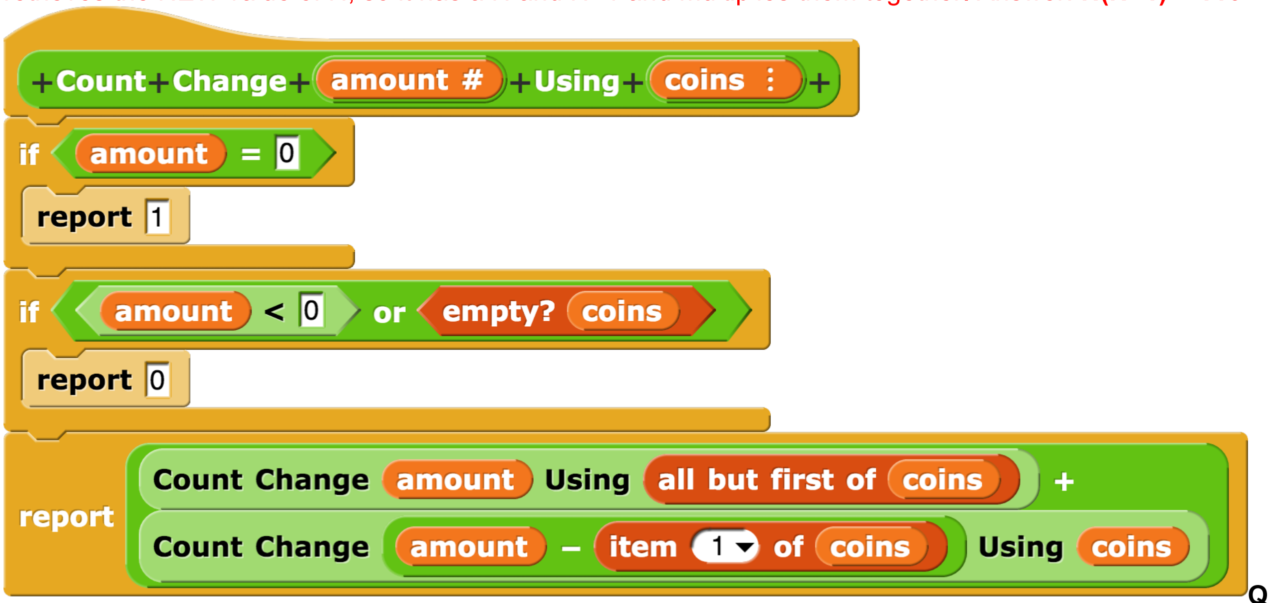
Option 1: Left gets N, adds 1, but the N←N+1 set is delayed. Then Right gets the original N twice, sets it to N^2 but that value is overridden by N+1 when that set eventually runs. Answer: **N+1 = 11**

Option 2: Right gets N twice, squares it, but the N←N*N set is delayed. Then Left gets the original N, sets it to N+1 but that value is overridden by N*N when that set eventually runs. Answer: **N*N = 100**

Option 3: Left gets N, adds 1, sets N←N+1. Then Right gets new N+1 twice, sets it to $(N+1)^2$. Answer: (**N+1)^2 = 121**

Option 4: Right gets N, squares it, sets N←N^2. Then Left gets new N^2, sets it to N^2+1. Answer: **N^2+1 = 101**

Option 5: Right gets N *once*, the other Get is delayed. Left gets N, adds 1, and executes N←N+1. Then Right's second get retrieves the NEW value of N, so it has a N and N+1 and multiplies them together. Answer: **N(N+1) = 110**



```
+Count+Change+(amount #)+Using+(coins ⋮)+
if <(amount) = 0>
    report 1
if <<(amount) < 0> or <empty? (coins)>>
    report 0
report (Count Change (amount) Using (all but first of (coins))) +
       (Count Change ((amount) − (item 1 of (coins))) Using (coins))
```

Question 4a: The number of ways of making change for **15** cents given coins**(10 5 1)** is 6, and our code returns 6.

If we *swap the order of the coins* to **(1 5 10)**, what would happen? It would report the same answer as before, the order of the coins doesn't matter in count change (thankfully!)

Question 4b: If we *swapped the order of the two **if-report**s*, when would it *change our reported value*? When both test cases would be true, then the order would matter, so it would be when the amount is zero AND coins are empty. (since the amount couldn't simultaneously be < 0 and = 0).

Question 4c: If we change the bottom-right expression **coins** to **all-but-first-of(coins)**, What'd be computed? It would report what'd happen if *each coin could only be used once, since on that recursive call you're both using a coin (amount – item 1 of coin) and not using it again (coin becomes all but first of coin).*

Question 4d: Normally if we asked for the change for **2** cents given only pennies, i.e., coins were **(1)**, it would report 1, since there's only a single way of doing it (penny + penny). What would it report if we actually had *two kinds of pennies*? I.e., what'd happen if we called count change for **2** cents with **(1 1)**? report 3 (for AA, AB, BB given pennies A,B)