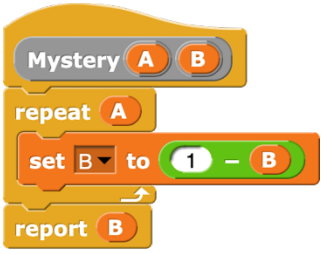
CS10 Fall 2019 Quest Answers

Question 1: Which of the following is a *true statement* regarding Abstraction? **Recipe authors who write "For a milkshake: blend ice cream, milk, vanilla and <fruit>" use *generalization*.**

Question 2: What is $100_2 + 11_{10}$? $100_2 = 1*4 + 0*2 + 0*1 = 4_{10} + 11_{10}, = 15_{10} = \mathbf{F_{16}}$.



Question 3: What does 

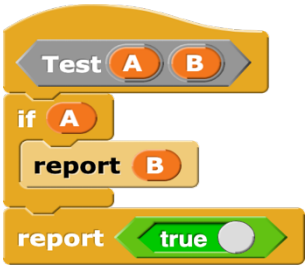
report?
B is initially 5 If A = 1, B is set to 1-B = -4. If A = 2, the first time B is set to -4 (as above). The second time B is 1-B = 5. From here, we see that B flip-flops between -4 (for odd A) and 5 (for even A) Since 200 is even, it reports **5**

What does 

report? Since 201 is odd, it reports **-4**

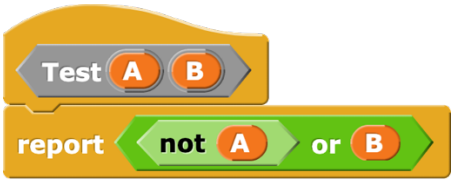
Question 4: What is your guess as to the *Domain* and *Range* of **Foo**?

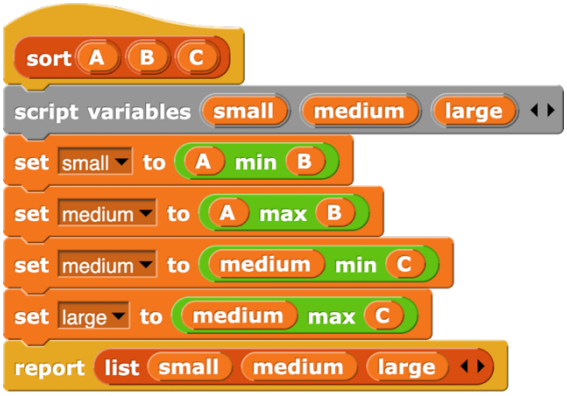
The expression doesn't cause an error. `letter D of E` returns a character so the domain of Foo is a **character**. + takes a number so the range is a **number**.



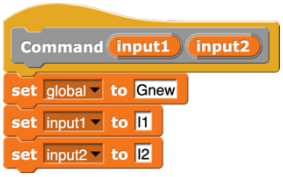
Question 5: If the output from **Test** is false, which can you say *for sure*? The only way to get false as an output is for **A to be true** (so it goes into the if) and then **B is false** (since that is what is reported).

Question 6: Which of the following are the same as the original **Test** block? Or really wants to be true – either input would make it true. So we have to find a way to make it false by having both its inputs be false. We know from above that it's only false when A is true and B is false. So if we negate A and take B unmodified (i.e., **not A or B**) then the only time that expression is false is when A is true and B is false, matching the code above perfectly.





Question 7: This script is intended to take three distinct numbers and return a list of three numbers, where the smallest number will be listed first, the middle number second, and the largest last. What do we know for sure? Will small always be the smallest? **NO**, because if C is the smallest, small never "sees" C's value. Will medium always be the middle number? **NO**, because if the largest is A and the medium is B, then after the first set, small will be set to medium and medium's value is lost. Will large always be the biggest number? **NO**, because if it's initially B, then the third set will overwrite its value with C.



Question 8: What gets said if I run the script below? Let's trace this: global gets set to G, outer script to OS. The value of these variables are passed to Command, which sets global to Gnew and input1 and input2 to values that don't matter, since all input parameters are local variables and don't affect anything outside. So when it returns, only global has been changed (to Gnew), and **GnewOS** is said.

Question 9: a) Who is the *oldest friend*? `combine` (because you're distilling many items down to one item)

b) What are the *zip codes of friends who sent you holiday cards last year*? The problem is, some zip codes contain people who did and did not send you cards! `map (mapper ()) over (keep (predicate) from (data))` (because you have to filter the people by those who sent you holiday cards first, then map the zip code extracting function over the result)

c) Who are the *friends who both owe you money AND have rich parents* they can borrow from to pay you back? `keep` (because you're keeping friends based on some criteria)

d) Given the phone number of every friend, *what is a list of the costs of an international call to them converted to Euros*? `map` (because you're transforming every friend to a cost in Euros)

e) What are *all their zip codes that are numeric palindromes*? (I.e., that are the same frontwards as backwards, like 98789) `keep items<predicate()>from(map(mapper())over(data))` (because you first need to convert every friend to their zip code, then keep those zip codes that are palindromes)