UNIVERSITY OF CALIFORNIA, BERKELEY
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS10 Fall 2026 **TA: Nahee**

SOLUTIONS

Discussion 6: HOFs, Rings, Debugging

Instructions:

- If you're attending this section in-person, please log into iClicker!
- For the worksheet, you can either explain the process in words, show a screenshot, or draw the block/process.

Group Activity / Question of the Day

- What is your least favorite chore to do and why? Also, do you have a favorite chore and why?

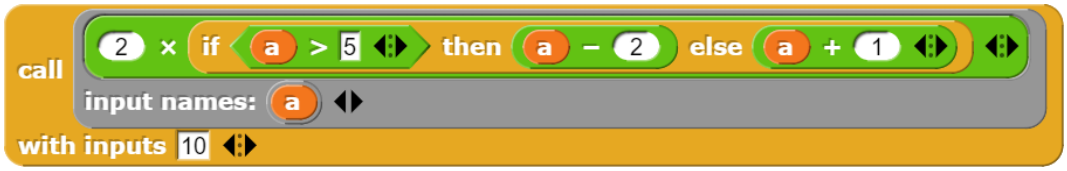
Required (Pages 2 - 4):

Section I - Rings

---

1. Trace the following functions and figure out what the output would be based on the input.

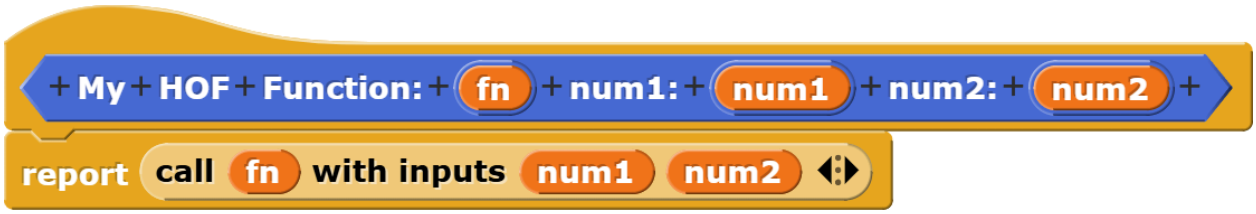
Returns TRUE



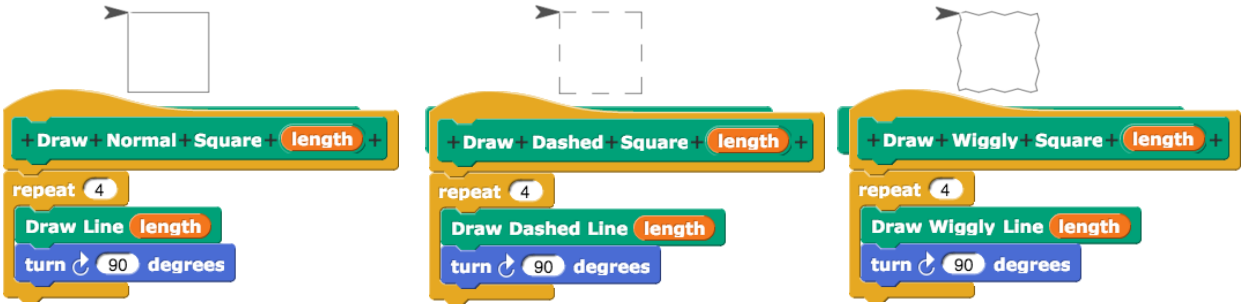
Returns 16

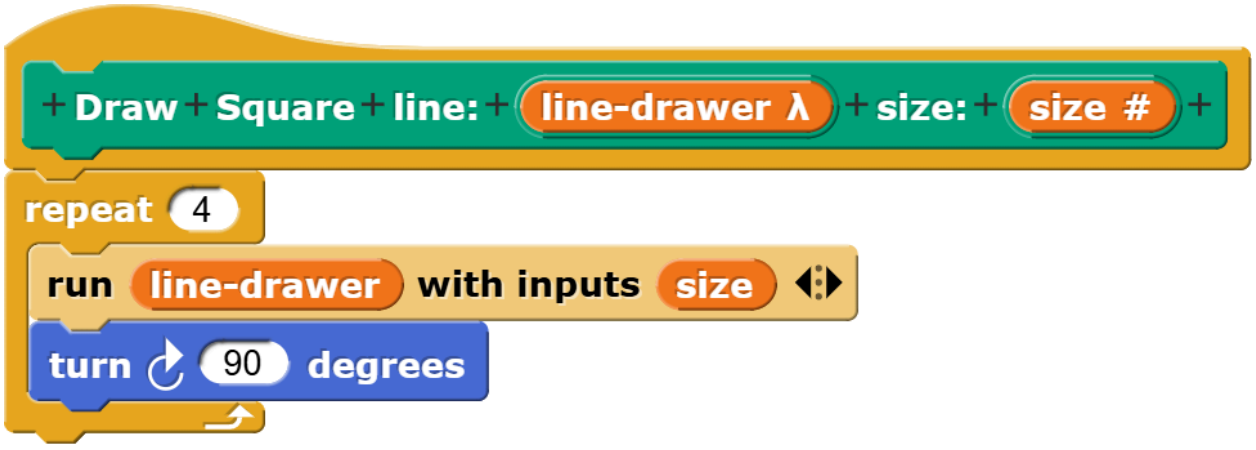
2. Create a ring / lambda function and set it to a variable 'a'. This function should take in 2 inputs, which are numbers. It should find the max number and the min number and subtract the min from the max.

3. Create a function, called 'My HOF function' that takes in three inputs, a predicate, and two numbers. The predicate will always be a dyadic function (take in only two inputs). You should call / invoke the function and return a result. Here are some examples of how it should work:

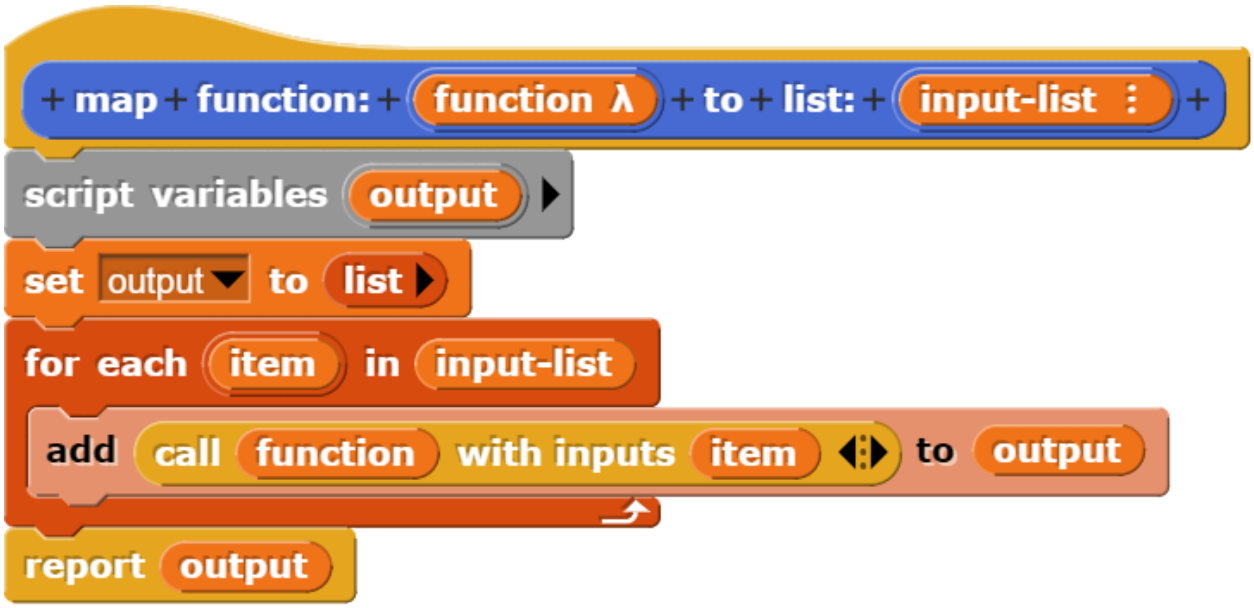


4. Suppose you have various functions from drawing different types of squares. You notice that each function is almost identical with the only difference being the "draw line". However, you want to generalize a function that can draw any of the squares below. Write a function that will draw any of the squares below. Feel free to use this starter code: [Draw Square into a HOF.](#) Example taken from Dan





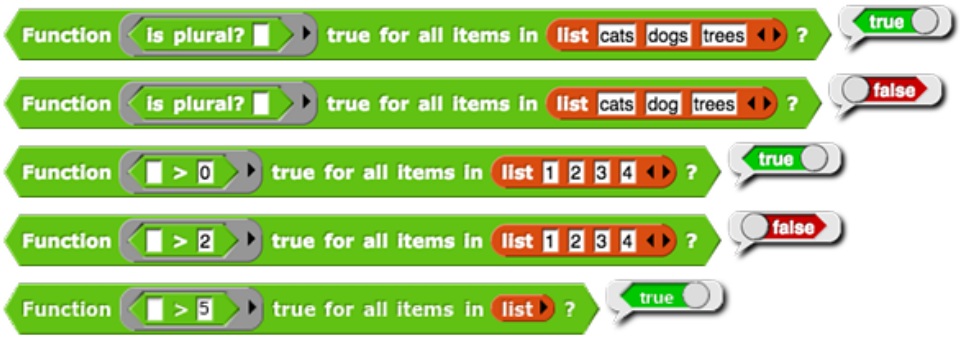
5. Create the map block from scratch. You should be invoking a function on every item from a list. Do this without using map. Instead, use iteration! Here's an example of how it should work:



Section II - Higher Order Functions

---

6. We would like to create the predicate HOF block which takes in a predicate function and a list. The predicate function should take a single argument. Our



HOF block should report True if every item in the given list makes the given predicate function report True. Below are some examples. The output of the block is True when the input list is empty, regardless of the predicate.

a. Your solution should only use ONE map, keep, or combine

b. Now, find another way to write this function using one or more map, keep, or combine.

c. Finally, we want to make a third version using iteration!

```
+ Function + func λ + true + for + all + item + in + list + list + (C) +
for each item in list
    if not call func with inputs item
        report false
report true
```

Section III – Debugging

---

1. For each of the following functions, determine where the error is occurring, why, what the output currently is, and what it should be:
   a. Function 1
   b. Function 2

```
+ sum + the + squares + from + 1 + to + N: + N +
script variables sum
set sum to 0
repeat until N ≤ 1
    set sum to sum + N × N
    set N to N − 1
report sum
```

We can change N <= 1 to be N < 1.

The output originally if n=3 was 14

Now it will return the correct value 15

```
+ word: + word + contains + vowel +
script variables i vowels
set i to 1
set vowels to list a e i o u
repeat until i > 1
    if vowels contains letter 1 of word
        report true
    set i to i + 1
report false
```

The issue is with letter 1 of word

Instead of looking at each letter in word (letter i of word), we are only ever looking at the 1 letter of word

So if word = “banana”, the function would return false since the first letter (“b”) is not a vowel

Optional Section (Extra Practice):

Optional Section I - Rings

Create the keep block from scratch. You should be invoking a function on every item from a list. Do this without using map or keep. Instead, use iteration!

Optional Section II - Higher Order Functions

---

1. Write a function “apply twice” that takes a function f and a value x (list) as arguments and returns the result of applying f to x twice. Here’s an example:

   a. Now, find another way to write this function using one or more map, keep, or combine.

   b. Finally, we want to make a third version using iteration!

Section III – Debugging

2. For each of the following functions, determine where the error is occurring, why, what the output currently is, and what it should be:

   a. Function 1

```
+ factorial + of + N: + (N) +
script variables (output)
set output to 0
repeat until (N < 1)
    set output to (N × output)
    set N to (N − 1)
report (output)
```

   b. Function 2

```
+ is + word: + (word) + a + palindrome +
script variables (first) (last)
set first to 1
set last to (length of text (word))
repeat until (first ≥ last)
    if (letter (first) of (word) = letter (last) of (word))
        report (true)
    set first to (first + 1)
    set last to (last − 1)
report (false)
```