

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

CS10 Fall 2026

TA: Nahee



Discussion 5: HOFs + Quest Practice

Instructions:

- If you're attending this section in-person, please log into the iClicker app for attendance!

Group Activity / Question of the Day

- Look at the person diagonally across from you (or, if in a line, pick someone two spots away). Ask them, "If you could time travel, but ONLY in one direction, would you rather go 500 years into the past or 500 years into the future? Why?"
- And what's something from today you'd bring with you or do to help survive (or thrive) in that time?

Required (Pages 2 - 5):

Section I - HOFs4z

1. Determine which of the following expressions will produce the following correct results:

a. You are given a list of numbers: [10, 2, 7, 6, 4] which is set to the variable `a`

i. You need to create a new list based on the following rules:

1. If an item is even, divide it by 2.

2. If an item is odd, multiply it by 2.

ii. The result should be: [5, 1, 14, 3, 2]



WRONG. The or block needs true or false values, so it cannot report numbers like 5 or 14. Moreover, the "or" block will output either true or false, and we will be mapping boolean values to the items of "a", not actual numbers.



WRONG. Regardless of what goes into the or block, keep items block by itself can only keep items and never modify the actual value of the items.



WRONG. The mod tests are true or false, and the item is never halved or doubled. I.e., the "or" block will output either true or false, and we will be mapping boolean values to the items of "a", not actual numbers.

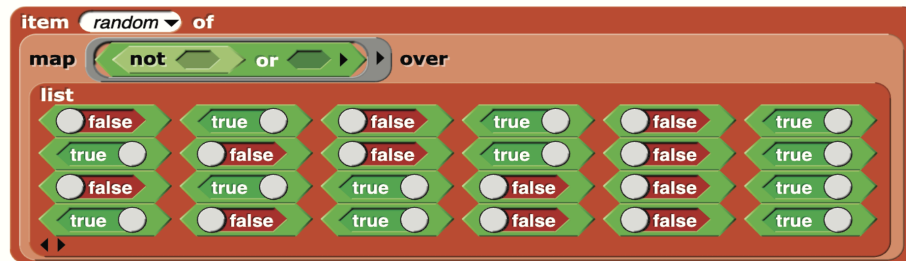


CORRECT. map changes every item, and the if then else ring reports a number each time: 10 becomes 5, 2 becomes 1, 7 becomes 14, 6 becomes 3, 4 becomes 2.



WRONG. keep items needs a true or false test (predicate) as its input function, but this ring reports numbers.

2. What does the following block report?



- ☐ True
- ☐ False
- ☐ It errors, because we can't pass in predicates in map blocks
- ☐ It depends - the output could be True or False, since selection is random.

Question 2 answer: True. Every empty slot gets the same item x, so each result is (not x) or x, which is always True.

- (not false) or false => true
- (not true) or true => true

After mapping, our input list thus becomes a list of 24 trues. Any item randomly picked from a list of all trues will always be true, so the answer never varies.

3. Using HOFs and named inputs, write a function that filters out the odd indices (only keep the even indices), and then doubles the value. Any HOFs you implement must use the named inputs. This is how the function should behave:

filter out the even indices and double value

list 1 2 3 4



Solution:

```
define: filter out the even indices and double value (lst)
report map ( (item (i) of lst) × 2 )      [named input: i]
        over keep items ( (i mod 2) = 0 ) [named input: i]
        from (numbers from 1 to (length of lst))
```

keep items finds the even indices, [2, 4]. map then looks up item 2 and item 4 of lst and doubles each one, giving [4, 8]. The name i is used in both rings, so neither ring relies on empty slots.

Section II - Quest Practice

1. Perform the following calculation and conversions

- a. $0b1011 + 0b11010$. Return the hex representation

Answer: 0x25. $0b1011$ is 11 and $0b11010$ is 26, so the sum is 37. In binary that is $0b100101$, and grouping in fours ($0010\ 0101$) gives 0x25.

- b. $0xD8 + 17_{10}$. Return the binary representation

Answer: 0b11101001. $0xD8$ is $13 \times 16 + 8 = 216$, and $216 + 17 = 233$. Since $233 = 128 + 64 + 32 + 8 + 1$, the binary form is $0b11101001$ (which is 0xE9).

- c. $0b110011 + 3_{10}$. Return the decimal representation

Answer: 54. $0b110011$ is $32 + 16 + 2 + 1 = 51$, and $51 + 3 = 54$.

2. Now, suppose we have a composite function $z(x) = 2|(x + 1)|^2$. Using the functions below, how can compose $h(x)$ to be in its current form. Represent your answer as $a(b(c(d(e(f(g(h(x)))))))$ form.

$$a(x) = x^2 \quad b(x) = x + 1$$

$$c(x) = 2x. \quad d(x) = |x|$$

$$e(x) = \sqrt{x} \quad f(x) = x^3$$

$$g(x) = x \div 2 \quad h(x) = \log x$$

Answer: $c(a(d(b(x))))$.

b adds 1: $x + 1$. d takes the absolute value: $|x + 1|$. a squares it. c doubles it. The functions e, f, g and h are not needed.

Check with $x = 2$: b gives 3, d gives 3, a gives 9, c gives 18, and $2 \times |2 + 1|$ squared is also 18.

- a. Now suppose we have $e(b(g(x)))$. How would this function be based on this composition using the functions above?

Answer: $e(b(g(x)))$ = square root of $((x \div 2) + 1)$. g halves x , then b adds 1, then e takes the square root.

3. Write a truth table for the following expression



a	b	$(a \neq b) \text{ and } \text{not } b$
T	T	False
T	F	True
F	T	False
F	F	False

a. If the output is `True`, what will `a` be? (either `True`, `False`, or `Neither`)

Answer: `True`. The only row that gives `True` is `a = T, b = F`.

b. If the output is `False`, what will `b` be? (either `True`, `False`, or `Neither`)

Answer: `Neither`. The output is `False` in rows `TT, FT` and `FF`, so `b` can be `True` or `False`.

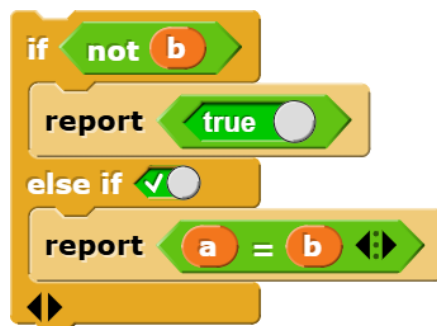
c. Which of the following statements are equivalent to the expression above?



NOT EQUIVALENT. When `a = T` and `b = F` it reports `(not a)`, which is `False`, but the original expression is `True`.



EQUIVALENT. If `b` is `True` it reports `False` (rows `TT` and `FT`). If `b` is `False` it reports `a not equal to b`, which gives `True` for `TF` and `False` for `FF`. This matches the truth table.



NOT EQUIVALENT. If `b` is `False` it always reports `True`, but in row `FF` the original expression is `False`.

4. What will the following function return if the input $n = 4$?

Answer for $n = 4$: 96.

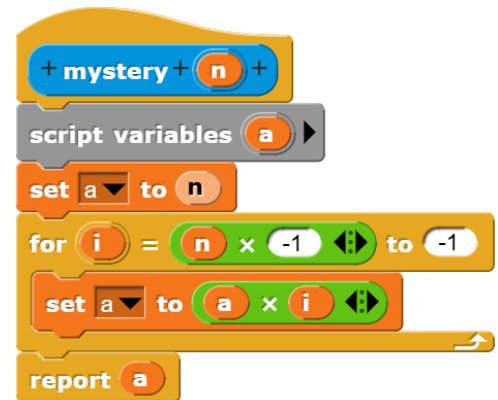
a starts at 4, and i counts from -4 up to -1. $i = -4$: $a = -16$. $i = -3$: $a = 48$. $i = -2$: $a = -96$. $i = -1$: $a = 96$.

a. In terms of n , what mathematical expression represents the mystery function outputs?

- n
- n^2
- $n^2 - 1$
- $-(n^2 - 1)$
- $-n$
- $n!$
- $-n!$
- $(-n)!$
- $n*(-n)!$
- **$n*n!$**

CORRECT. a starts at n and is multiplied by every i from $-n$ to -1 . Those values multiply to $n!$ for even n (like 4), so $a = n \times n!$, and $4 \times 24 = 96$.

- $n*(-n!)$



Walkthrough video: <https://drive.google.com/file/d/1ryLE5DIOPcyDZ0EL2I7RVcvOf6xTpSbk/view>

Starting with $a = n = 4$, the loop runs for $i = -4, -3, -2, -1$, updating $a = a \times i$ each time: $4 \times -4 = -16$, then $-16 \times -3 = 48$, then $48 \times -2 = -96$, then $-96 \times -1 = 96$. **The function returns 96.**

In terms of n : a starts at n and is multiplied by every integer from $-n$ up to -1 . That product has n factors, each with absolute value from 1 to n , so its size is $n!$. The sign is positive when n is even, which is the case for $n = 4$. So the output is **$n * n!$** , and $4 \times 24 = 96$. (For odd n the sign would flip, but $n * n!$ is the intended choice among the listed options.)