

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

CS10 Fall 2025

TA: Victoria



Discussion 9: Midterm Practice

Instructions:

- If you're attending this section in-person, please log into iClicker!
- If you missed this discussion, fill out this entire worksheet, and upload it to the Gradescope assignment titled "Discussion 9" by next Discussion.
- For the worksheet, you can either explain the process in words, show a screenshot, or draw the block/process.
- Please complete the Feedback Form tinyurl.com/fa25-disc-form

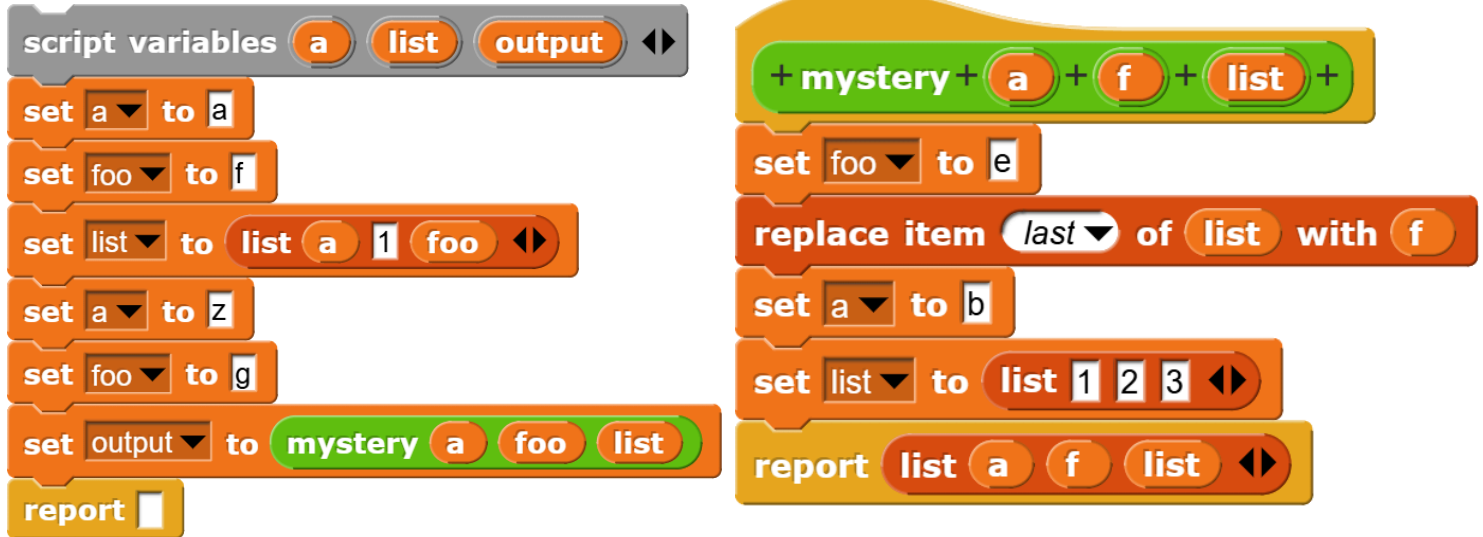
Group Activity / Question of the Day

- Do you have a favorite word? And Why? Where did you first learn it?

Required (Pages 2 - 8):

Section I - Scope

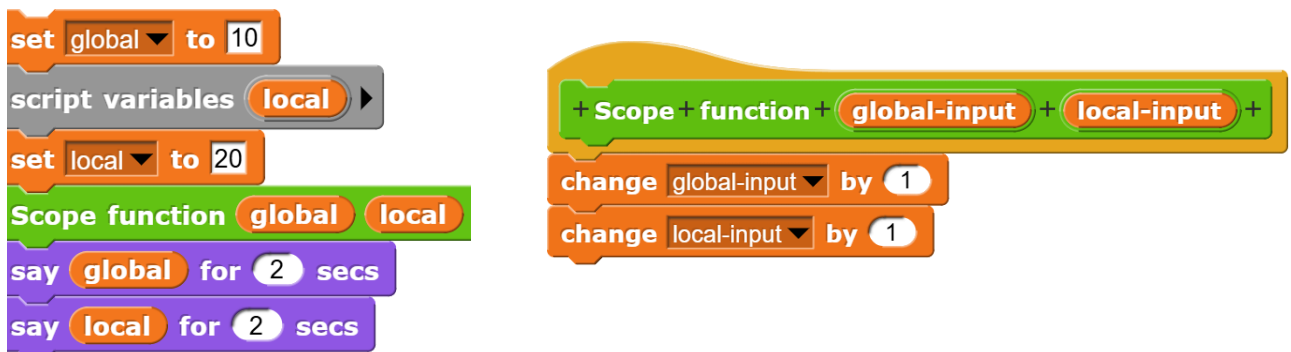
1. Determine what will be reported if the following variable was in the “report” block. You will need to trace the script to determine the outputs.



The script is divided into two parts. The first part, under the 'script variables' tab, contains the following blocks: 'set a to a', 'set foo to f', 'set list to list a 1 foo', 'set a to z', 'set foo to g', 'set output to mystery a foo list', and 'report'. The second part, under the 'mystery' tab, contains: '+ mystery + a + f + list +', 'set foo to e', 'replace item last of list with f', 'set a to b', 'set list to list 1 2 3', and 'report list a f list'.

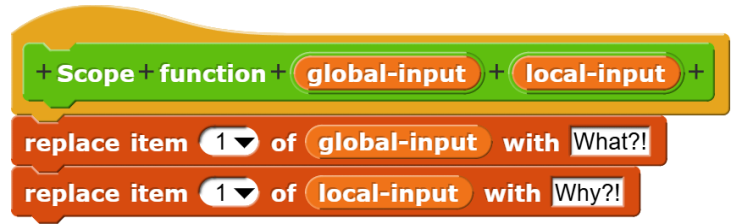
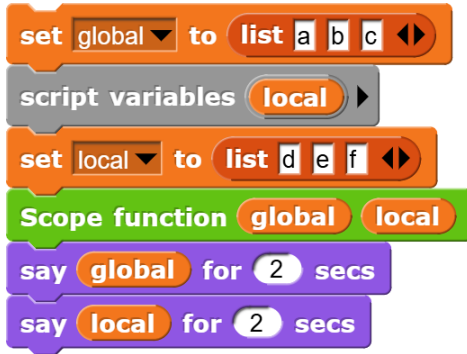
- a. 
- b. 
- c. 
- d. 

1. Determine what the ‘say’ blocks should say for each script
 - a. Script 1



The script is divided into two parts. The first part, under the 'script variables' tab, contains: 'set global to 10', 'script variables local', 'set local to 20', 'Scope function global local', 'say global for 2 secs', and 'say local for 2 secs'. The second part, under the 'Scope function' tab, contains: '+ Scope + function + global-input + local-input +', 'change global-input by 1', and 'change local-input by 1'.

b. Script 2



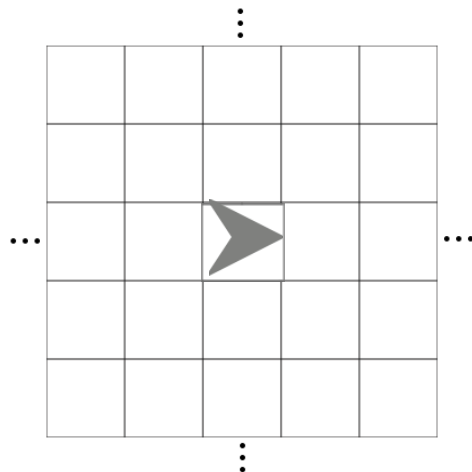
Section II – Iteration + Randomness

Here are helper blocks for controlling a robot (shaped like an arrowhead) on a grid world (infinite in both X and Y dimensions), looking down on it.

- The robot moves N squares forward in the direction it's facing.



- The robot turns some number of degrees clockwise (right), in-place, about its center. (remaining in the same square).
- The robot starts in the right direction



A. Fill out the grid for all possibilities of Walk(1)

☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐

C. Is there a pattern? If so, can you draw out the pattern as N gets larger?

D. If the center is $x=0$. What is the maximum x value to the right for any given N?

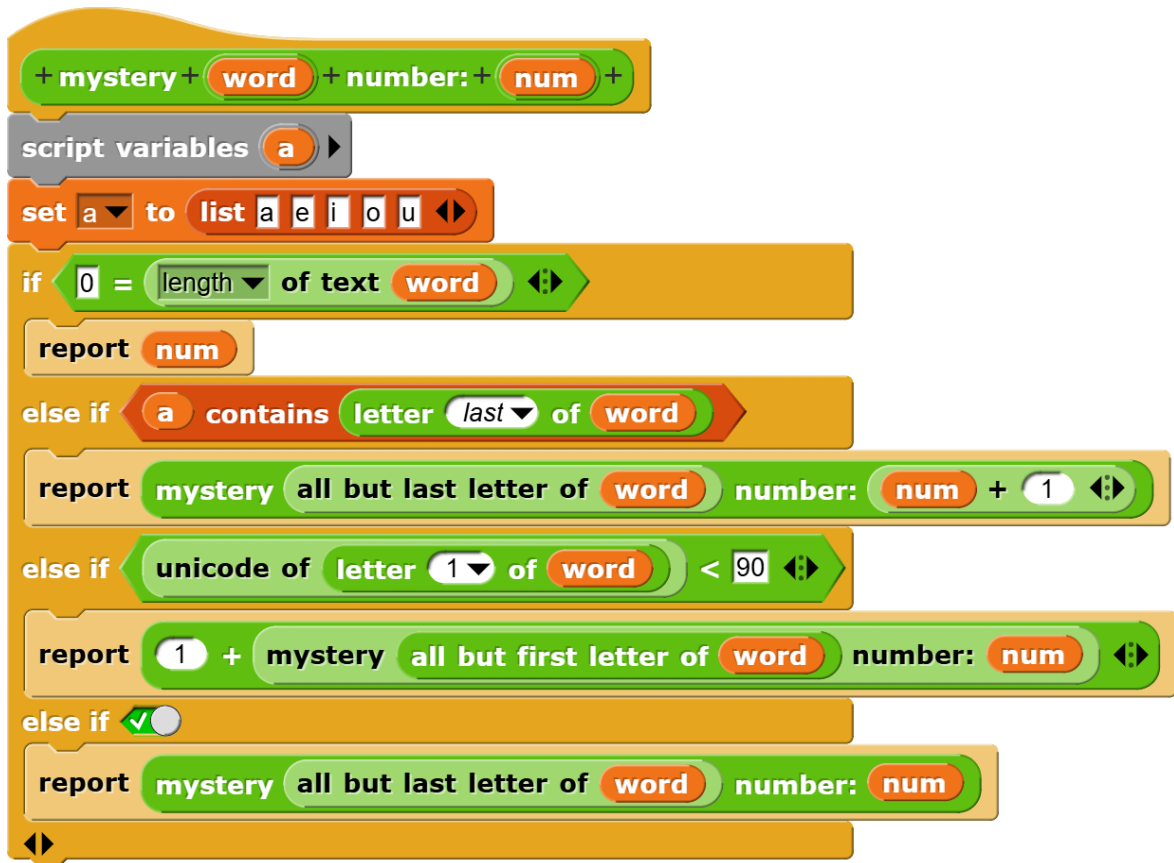
B. Fill out the grid for all possibilities of Walk(2)

☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐
☐	☐	☐	☐	☐

Section III – Recursive Tracing

1. Trace the function.
 - a. Determine what the function is doing in your own words.
 - b. Determine what the function outputs if the following is inputted:

mystery Victoria **number:** 0



- c. What value for word would cause the function to output 3? There is more than one correct answer.

- Your job is to understand and trace the function each time it is called or enters a new frame. Trace the function, and find the input value for str for that frame. By trace, we mean, figure out what the inputs are (what str is set to) for each frame. You DO NOT need to trace the value for value. You will be tracing the following call:

The Scratch code consists of the following blocks:

- Block 1:** + value: value + from string: str +
- Block 2:** if length of text str = 0
- Block 3:** report str
- Block 4:** else if letter 1 of str = value
- Block 5:** report join
 - letter 1 of str
 - value: value from string: all but first letter of str
- Block 6:** else if (checked)
- Block 7:** report join
 - value: value from string: all but first letter of str
 - letter 1 of str

The call stack trace on the right shows five calls, each with a 'str=' input field:

- Call #1: str=
- Call #2: str=
- Call #3: str=
- Call #4: str=
- Call #5: str=

- What will the final output be for: value: c from string: cats
- What would the final output be value: o from string: world for:

Section IV – Algorithmic Complexity

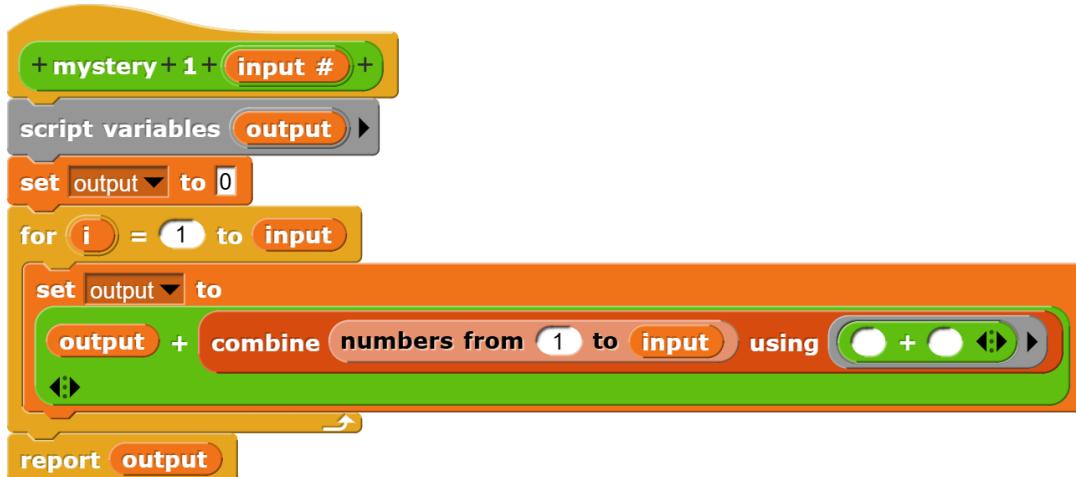
- For each of the following scenarios, determine what the worst-case runtime would be:
 - Count how many students raised their hands. You scan a line of N students once and tally hands up.
 - You have a list of N movies to watch this month. You want to decide which ones to include in your marathon, so you try every possible combination of

movies (watch some, skip others) to see which set gives the highest total enjoyment.

- c. You have N students and N locker numbers, all on sticky notes and shuffled. You compare every pair of notes to see if two students have the same locker.
- d. Guess a secret number with hot/cold feedback on a fixed range. The number is somewhere from 1 to a known large maximum. You're told "higher" or "lower" after each guess; you must guarantee finding it.
- e. You're a security guard walking down a hallway. You scan a badge once to unlock every door in the building.
- f. You are going to a party where you need to bring three different dishes. There are N different dishes you can pick up. You find every combination of three dishes and then pick one combination of three dishes you like the best.
- g. You want to check a list of restaurants to see if each is vegetarian. Instead of going to each restaurant's website, you access a site that allows you to check at one time if all the restaurants you're interested in offer a vegetarian option.
- h. You have a list of N movies released from 2000 to 2020, sorted by year. You want to find the first movie that made over \$100 million. Each time you check a movie, you jump to the middle of the remaining range. For example, you will check 2010 first.

2. For each function, determine the worst-case runtime.

a. Mystery 1



b. Mystery 2



Section V – HOFs

1. Consider the following list



3	A	B	C
1	Victoria	Berkeley	Green
2	Andrew	San Francisco	Blank
3	Alonzo	Snap!	Yellow

The objective is to report the following flattened list using 3 functions/operations

report Function 1(Function 2 (Function 3))

1	Victoria, Berkeley, Green
2	Andrew, San Francisco, Blank
3	Alonzo, Snap!, Yellow

length: 3

Choose the corresponding functions that will allow us to do so

item 1 of

+

append

keep items from

add to

map over

letter 1 of

join

combine using

2. Using the same input list, we now want to output the list below. Circle the options below that will allow us to return the desired output (may be more than one)

3	A	B	C	D
1	Victoria	Berkeley	Green	CS10
2	Andrew	San Francisco	Blank	CS10
3	Alonzo	Snap!	Yellow	CS10

map append list join CS10 over data

map reverse of CS10 in front of reverse of over data

map list item 1 of item 2 of item 3 of join CS10 over data

combine data using map list join CS10 over data