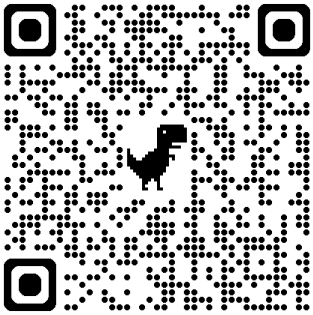
UNIVERSITY OF CALIFORNIA, BERKELEY
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS10 Spring 2025 **TA: Victoria**



Discussion 13: Concurrency + Postterm 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 13" by next Discussion.
- Please open up snap.berkeley.edu/run on your computer.
- 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/sp25-disc-form**

Question of the Day

- Do you have any specific things you do before an exam (i.e. get a specific drink, watch a certain show, listen to a pump-up jam, etc.)

Required (Pages 2 - 6):

Section I - Concurrency

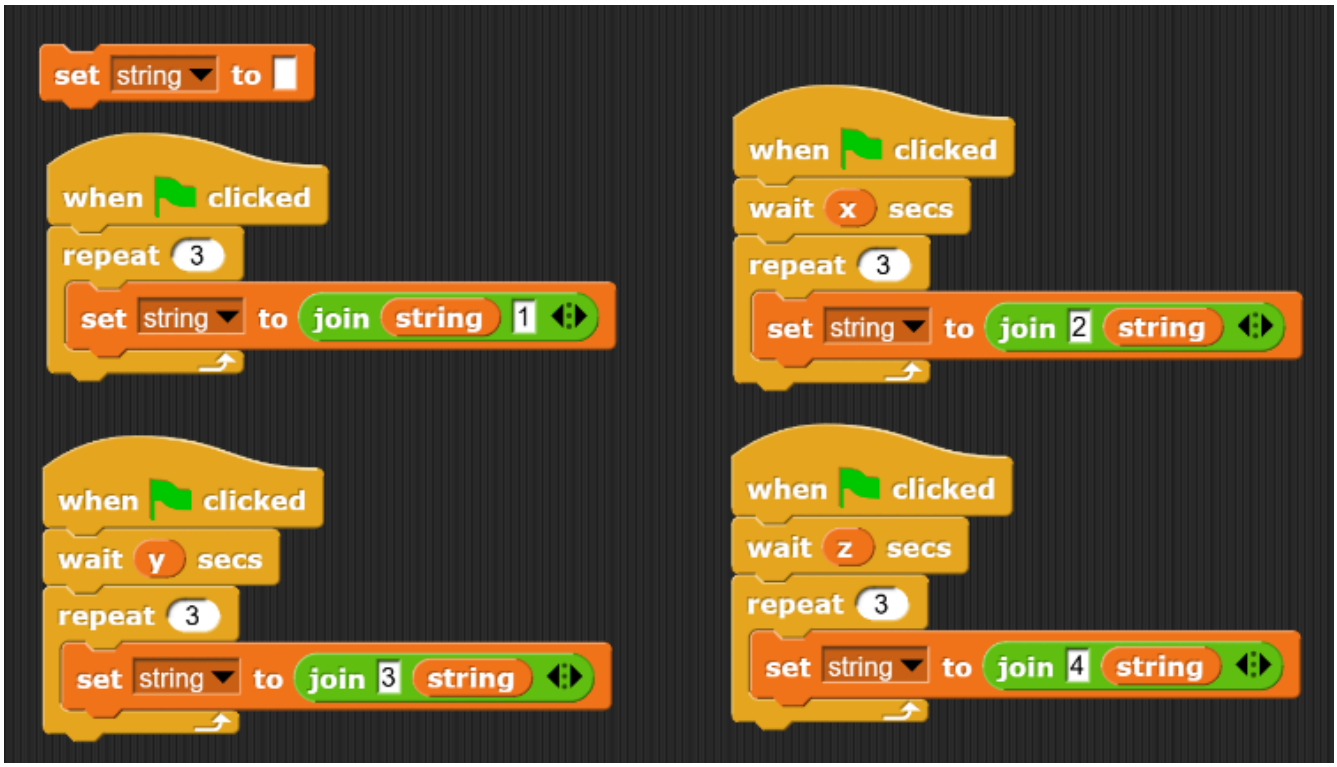
Amdahl's Law $S = \frac{1}{(1-P) + (\frac{P}{N})}$

1. You run a profiling program on a different program to find out what percent of time within the program each function takes. You get the following results:

Function	% Time
f	20%
g	50%
h	30%

   a. Assuming that each of these functions can be parallelized by the same speedup factor (N=2), which function, if parallelized, would cause the most speedup for the entire program? *Hint: Find the speedup (S) for each function.*

2. Assume you have the following script. What would string be once all scripts are done running for the given x, y, z values? If all scripts are running at the same time, you can assume the left topmost goes first. If x = 1, y = 1, and z = 1, the string would be "112341234234"



a. x = 1, y = 2, z = 2

b. x = 0, y = 4, z = 2

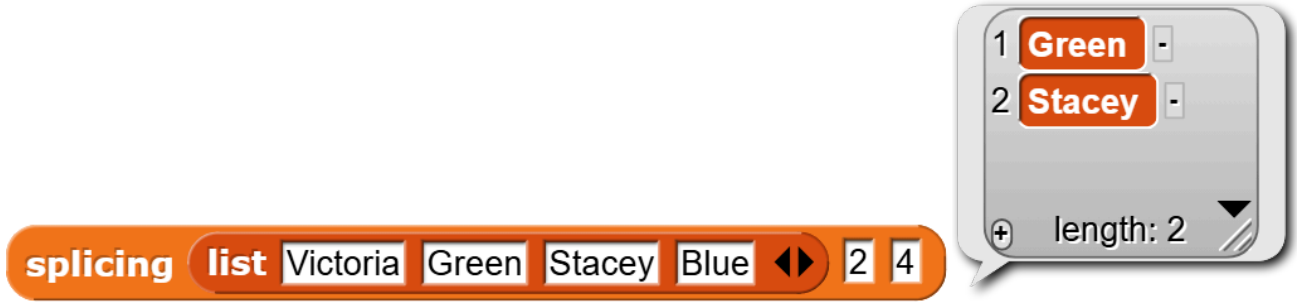
Section II - Postterm Practice

---

1. What's $\frac{AF}{101_2}$? Express your result in hexadecimal and in base-3

2. What's $\frac{BEEF_{16}}{12467_{8}}$? Express your result in decimal and in base-7

3. Write a function `splicing(data, left, right)` that works like splicing in Python, taking in a list, and two numbers (left and right), which represent the left and right indices when splicing. Items should be indexed starting at 1 though, and the right number is exclusive. You should use only HOFs.



4. Write a function `join_letters(data)` take takes in a 2D list of strings and returns a single 1D list with the first letters of each item separated by a space. The function should be completed recursively.

5. Write a function `sort_by_unicode_weight(s)` that takes a string `s` of words separated by spaces. Each word's Unicode weight is defined as the sum of the Unicode code points (ord(char)) of its lowercase alphabetic letters only. Ignore any non-alphabetic characters when calculating the score. Your function should:
    a. Compute the Unicode weight of each word.
    b. Sort the words in descending order of their Unicode weight.
    c. Return the sorted words joined by spaces.

    Here are some examples:

```
>>> sort_by_unicode_weight("Cat apple zOOm!")
'apple zOOm! Cat'
>>> sort_by_unicode_weight("hello world!")
'world! hello'
>>> sort_by_unicode_weight("Aa Bb Cc")
'Cc Bb Aa'
```

    Hint: Python has a built-in function called ord(char) that gives you the Unicode (ASCII) value of a character. For example: `ord('a')  # returns 97`

6. Return a list containing every possible substring of string having length n. You can assume len(output) < len(string). Complete the function:
    a. In Python using List comprehension
    b. in Snap*!* Using HOFs

```
>>> def substring(“Hello”, 3)
[‘Hel’, ‘ell’, ‘llo’]
>>> def substring(“World”, 2)
[‘Wo’, ‘or’, ‘rd’, ‘ld’]
```

7. Write a function named is_descending that takes an integer n as input and determines if n's digits are descending. The function should return True if each digit is in descending order, otherwise False.
    a. Write the function
        i. in Python using list comprehension
        ii. In Snap! Using HOFs
    b. Below is an example:

```
>>print(is_descending(1))
True
>>>print(is_descending(1111))
False
>>>print(is_descending(12))
False
>>>print(is_descending(321))
True
>>>print(is_descending(31))
True
```