PRACTICE WITH RECURSION

Working with a partner/group, use the following steps to solve each of the following problems.

(a) Plan your code on the white board (either on the classroom wall or on Zoom). Write out your entire program. Think about what errors might occur and how to fix them.

(b) Plan multiple test cases. What input will you send to your function? For each input, what value should be returned?

(c) *Only after you have completed steps (a) and (b) should you type your code in Python.*

(d) After you have typed your code, run your test cases. Does your code work? If not, how can you fix it?

1. **Factorials**: Recall that the factorial of an integer n, denoted $n!$, is the product of all positive integers less than or equal to n. For example, the factorial of 5 is

$$5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$$

Write a recursive function to compute the factorial of a number.

First, think about how the problem of computing the factorial of n can be reduced to the problem of computing the factorial of a smaller number — this gives you a recursive strategy. Then, identify the base case that will cause the recursion to stop.

2. **First uppercase letter**: Write a recursive function that finds the first uppercase letter in a string.

First, think about how the problem of finding the first uppercase letter in a string can be reduced to a smaller problem. Then identify the base case that will cause the recursion to stop.

3. **Recamán's sequence**: Recamán's sequence is a sequence of integers $a_0, a_1, a_2, \ldots$ defined recursively as follows. First, $a_0 = 0$. Then, for $n > 0$:

$$a_n = \begin{cases} a_{n-1} - n & \text{if } a_{n-1} - n \text{ is positive and not already in the sequence,} \\ a_{n-1} + n & \text{otherwise.} \end{cases}$$

The first few terms of the sequence are 0, 1, 3, 6, 2, 7, 13, 20, ….

Write a recursive function `recaman(n)` that computes the first n terms of Recamán's sequence.

4. **Bonus – Tower of Hanoi**: Write a recursive function that prints the moves necessary to solve the Tower of Hanoi puzzle for n disks. See en.wikipedia.org/wiki/Tower_of_Hanoi for an explanation of the puzzle.