Lab 9: Recursion

INFSCI 0201
Intermediate Programming (Spring 2025)

Don't forget - **"In order to understand recursion, one must first understand recursion."**

Submission

You are to write a complete Python program that meets the requirements outlined in the Lab 9 Tasks section.

- As with all programming assignments in this course, you must submit Lab 9 through GitHub in the git repository the instructor created for you. You must place your Lab 9 eclipse project in the folder **/labs/lab09/**
- All of your code should be in a file named "**recursion.py**". In which you will have to write three functions:
    - product_of_digits(x)
    - array_to_string(a, index)
    - log(base, value)
- If you did this all correctly, this project will be in the file **/labs/lab09/recursion.py**

Testing

The main test for this lab is in two parts.

- All three functions should perform and output values as expected.
- You must use recursive solutions to solve them. Using iterative processes (for/while loops, etc.) or calling built-in or third-party packages will not grant you any points on the specific task.

Note: Style guide violations will count as test failures! Be sure you're strictly adhering to the Python coding style guide on Canvas.

Lab 9 Tasks

Task 1: Write a recursive function with the following signature that returns the product of the digits of an integer.

product_of_digits(x)

- If x is 234, the function should return 2 * 3 * 4, that is 24.
- If x is 12, the function should return 1 * 2, which is 2.
- If x is negative, ignore the minus sign. For example, -12 and 12 both return 2.
- Hints:
    - The base case is a number between 0 and 9.
    - If x is negative, just multiply it by -1 or take an absolute value as soon as you enter the function.
    - For the recursive call, consider how computing the values x/10 and x%10 might be useful.

Task 2: Write a recursive function (DO NOT USE LOOPS) that returns a String representation of all the elements of a list of integers separated by commas. The parameters to the function should be a list of integers (a), and an integer value representing index (index). Assume *index* starts at 0 when the method is first called (it's a variable that will be super useful in making this method work recursively!)

array_to_string(a, index)

Hint: recall that the iterative solution to this problem is:

```
def array_to_string(a):
        str = ""
        for index in range(0, len(a)):
                str += a[index] + ","
        return str;
```

Consider the relationship between the condition for continuing the loop in this example, and the condition for the recursive case in your solution.

Task 3: A logarithm is a very useful mathematical function. Using a given base, b, we define Y = $\log_b X$ to be an exponent such that $b^Y = X$. If we are dealing exclusively with integer values, we typically round the log up or down using the ceiling or floor function. In this lab, we'll use the floor. This will give us the exponent Y such that $b^Y <= X$ but $b^{Y+1} > X$.

For example, consider b = 10 (i.e., we are considering base 10 logarithms):
floor (log10 (123456)) = 5 because 105 = 100000 <= 123456 and 106 = 1000000 > 123456

As another example, consider b = 2 (i.e., we are considering base 2 logarithms):
floor (log2 (64)) = 6 because 26 = 64 <= 64 and 27 = 128 > 64

To calculate the floor of the $\log_b X$, you can use repeated integer division. Divide by the base until the quotient is less than the base. The number of completed divisions is equal to the floor of the $\log_b X$.

For example, consider the following cases:

Example 1: int(log10 (123456)) ⇒ 5 divisions
123456 / 10 = 12345
12345 / 10 = 1234
1234 / 10 = 123
123 / 10 = 12
12 / 10 = 1 < 10
* It takes 5 divisions for our quotient to be less than the base (10), so the answer is 5

Example 2: int(log2 (64)) ⇒ 6 divisions
64 / 2 = 32
32 / 2 = 16
16 / 2 = 8
8 / 2 = 4
4 / 2 = 2
2 / 2 = 1 < 2
* It takes 6 divisions for our quotient to be less than the base (2), so the answer is 6

Example 3: int(log10(4567)) ⇒ 3 divisions
4567 / 10 = 456
456 / 10 = 45
45 / 10 = 4 < 10
* It takes 3 divisions for our quotient to be less than the base (10), so the answer is 3

For this task, you need to write a method to calculate a (floored) logarithm (as described in the background section) using the value and the base provided by the user and return the result. And it must be done recursively (no loops or Math function calls allowed!)

- `value` must be greater than 0, and `base` must be greater than 1.

- Both values must be integers.
- If an invalid value is passed to the function, the function should raise an error and stop running.

def log(base, value)