Cairo University
Faculty of Engineering
Computer Department

Expected Outcome:

This lab is expected to help you understand three concepts of functional programming:

- First Class Functions
- Pure Functions
- Currying

This lab will help you gain hands-on experience on how to implement them.

Description:

Implement a functional pipeline that processes a list of numerical values. This pipeline will include functions for filtering out even numbers, multiplying each number by a specific value, and then summing the results. Incorporate first-class functions, pure functions, and currying into your implementation.

Tasks:

1. Implement a Pure Function for Filtering: Write a pure function named **filterEven** that takes a list of numbers and returns a new list containing only the even numbers from the original list.

2. Implement a Curried Function for Multiplication: Create a curried function named **multiplyBy**. The first function takes a number **n** and returns another function. The returned function takes a list of numbers as its argument, multiplies each number by **n**, and returns the resulting list.

3. Implement a Pure Function for Summation: Write a pure function named **sumList** that takes a list of numbers and returns the sum of those numbers.
   Hint: You can use the reduce method.

4. Implement a First-Class Function for Processing: Implement a function named **processPipeline**. It should use **filterEven**, **multiplyBy**, and **sumList** to process the input list. Specifically, it should filter the list to include only even numbers, multiply each number by a specified value (use currying for this), and then sum the results.

5. Testing: Ensure your implementation works as expected by writing different test cases.