Exercise Set 4: Backpropagation

The GitHub folder [ChainRule](https://github.com/electronicarts/cpp-ml-intro/tree/main/Exercises/4_Backprop/ChainRule) contains the same program as FiniteDifferences and DualNumbers exercises, but now uses the [chain rule](https://en.wikipedia.org/wiki/Chain_rule) to calculate the derivative.

**Your mission is to implement the body of the FDerivative function using the chain rule to make the program work.**

| **💡 Exercise Hint**  If you run into problems, you can use finite differences to understand what the derivatives *should* be (approximately). This will help you debug and discover what might be going wrong with the chain rule. |
| --- |

# More things to try

1. Try modifying the functions G,H,I and see if it still works. (Don’t forget to update the derivative functions to match!)
   1. Hint #1: With nested functions, it’s easy to accidentally make a function that doesn’t have a minimum and just descends into infinity. This will cause you to hit a NaN.
   2. Hint #2: If you find yourself hitting NaNs, don’t sweat it too much. You can always look at the *solutions.docx* file for an explanation.
2. The function F is a single nested function. Can you make it be the sum of two nested functions and implement FDerivative to accommodate that?