Below are several exercises that we will be covering in the upcoming tutorial (Friday, Mar 11). We will release these exercises in advance of the tutorial so that you get a chance to attempt the exercises yourself before we discuss them in tutorial. The course staff running the tutorial will go through each problem (time permitting) and show their process for how they would complete these questions in an assignment/exam setting. Being familiar with the questions before attending tutorial will help to ensure that you get the most out of our explanations.

Note that tutorials are not mandatory in CS135. We never cover any new material that you won’t have seen in lectures, and instead choose to highlight techniques and concepts from the most recent lectures to give you some extra practice. If you are very comfortable with the material that we covered and could easily complete the below questions, then you aren’t obligated to attend tutorials. Be very cautious with this though since sometimes questions can seem easy until you actually sit down and attempt them.

**Question 1: The Male and Female Hofstadter Functions**

The Hofstadter female (F) and male (M) sequences were first described by Douglas Richard Hofstadter in 1979. They are defined as follows:

- \( F(0) = 1, \) \( M(0) = 0 \)
- \( F(n) = n - M(F(n-1)), n > 0 \)
- \( M(n) = n - F(M(n-1)), n > 0 \)

**Part A.** Using Racket, implement the mutually recursive functions \( (M n) \) and \( (F n) \) which each consume a natural number \( n \) according to the rules above.

*Examples:*

\( (F 4) \to 3 \)

\( (M 4) \to 2 \)

**Part B.** Write a function \( (\text{count-different } n) \) which consumes a natural number \( n \). The function should produce, for all \( i \) from 0 to \( n \), the number of times \( F(i) \) and \( M(i) \) differ.

*Example:*

\( (\text{count-different } 5) \to 4 \)
Question 2: Comparing Averages

After a recent midterm, you’ve obtained a list of marks for each student in CMPUT 135. You’d like to know how well each student did compared to the average.

Write a function \((\text{compare-average marks})\) which consumes a list of marks (numbers ranging from 0.0 to 100.0) and returns a new list where, for each mark in the original list, the new list contains the difference between that mark and the average. If a mark is below average, its difference will be negative.

**Important:** You may only use local functions in your solution and must compute the average only once!

*Example:*

\((\text{compare-average (list 60 40 80 95 100)}) \rightarrow \text{(list } -15 -35 5 20 25)\)

Question 3: Line Functions

Write a function \((\text{make-line bst m b})\) which takes a slope \(m\) and an intercept \(b\) and produces a function that evaluates the line defined by \(m\) and \(b\) at the point \(x\) (where \(x\) is a parameter for the produced function). That is, the produced function should compute \(y=mx+b\).

*Examples:*

\((\text{define example-line (make-line 1.5 } -5))\)

\((\text{example-line 0}) \rightarrow -5\)

\((\text{example-line 4.5}) \rightarrow 1.3\)