MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.037—Structure and Interpretation of Computer Programs
IAP 2019

Scheme Basics

Sum of all its parts

Write a procedure named `sum-numbers` which takes as input two integers, M and N, and returns the sum of all the numbers on the interval [M,N]. For example:

```
(sum-numbers 1 5)
```

would evaluate to `15`. Write two versions of this procedure, one which evolves an iterative process and one which evolves a recursive process.

What happens when you evaluate `(sum-numbers 5 0)`? What do you think should happen?

Fibonacci

The Fibonacci numbers are the integer sequence 0, 1, 1, 2, 3, 5, 8, 13, etc. Each subsequent element is the sum of the previous two. In other words: $F_n = F_{n-1} + F_{n-2}$ with initial values $F_0 = 0$ and $F_1 = 1$.

Write a procedure `fib` which, given a non-negative integer n, returns F_n. Implement this one in as a recursive process. How many times is the procedure `fib` applied when you evaluate `(fib 4)`?

Then, write another version which is iterative.

Feel the power

Write a procedure named `my-expt` which, given x and y, computes x^y. Assume that x is a number and y is a non-negative integer. Implement two versions, one recursive, one iterative.

How long does it take to evaluate `(my-expt 3 1000000)`? What can be done to improve this? Hint: If y is even, $x^y = (x^{y/2})^2$. Write a new procedure `fast-expt` that takes advantage of this mathematical insight. Why is the resulting procedure faster?