Objectives

- Work with lists, pairs, and a defined data type.

Activities

In this lab, you will

- define a datatype and some associated functions for complex numbers that you represent using `pairs`.

- write some functions that manipulate lists, and

1. Implementing complex numbers.

   Mathematical preliminaries

   A *complex number* is a number that can be expressed in the form $a + bi$, where a and b are real numbers and i is the imaginary unit, which satisfies the equation $i^2 = -1$. [1] For $a + bi$ we call a the real part and b the imaginary part.

   We define addition, subtraction, multiplication, and division as follows:

   Addition:
   $$(a + bi) + (c + di) = (a + c) + (b + d)i$$

   Subtraction:
   $$(a + bi) - (c + di) = (a - c) + (b - d)i$$

   Multiplication:
   $$(a + bi)(c + di) = (ac - bd) + (bc + ad)i$$

   Division:
   $$\frac{(a + bi)}{(c + di)} = \left(\frac{ac + bd}{c^2 + d^2}\right) + \left(\frac{bc - ad}{c^2 + d^2}\right)i$$

   Define the following Scheme functions that allow you to implement complex numbers:

   (a) `(make-complex a b)`, which constructs a complex number with real part `a` and imaginary part `b`

   (b) `(real x)`, which returns the real part of complex number x, and

   (c) `(imag x)`, which returns the imaginary part of complex number x. Hint: use pairs for your information, as in Monday's lecture.

2. Using the functions from the previous problem, define the following Scheme functions. These should use the functions you defined for the last question, so if you need to get the real part of a number x you should use `(real x)`, not some other function.

---

[1] from Wikipedia page, http://en.wikipedia.org/wiki/Complex_number

(a) `(complex-add x y)`, which returns the sum of two complex numbers

(b) `(complex-sub x y)`, which returns the difference of two complex numbers

(c) `(complex-mult x y)`, which returns the product of two complex numbers Note: all of these return complex numbers. You do not need to implement `complex-div`, but feel free if you have time.

3. Using your complex numbers.

   The complex conjugate of a number $z = a + bi$ is $a - bi$, and is denoted $\bar{z}$. It has the interesting property that $z\bar{z}$ is a real number; if $z = a + bi$, then $z\bar{z} = a^2 + b^2$, which follows from $i^2 = -1$ and the rules of multiplication.

   Define a Scheme function `(complex-conj x)` which returns the complex conjugate of x. Demonstrate that it works as expected by multiplying a couple of complex numbers times their conjugates.

4. Write the following functions on lists.

   (a) Write a Scheme function `(count-positives lst)` that counts the number of positive numbers in a list of numbers. See examples below.

   (b) Write a Scheme function `(sum-list lst)` that adds up the elements in a list of numbers.

   (c) The following Scheme function `(consecutive-ints a b)` evaluates to a list of numbers from a to b, where a and b are integers; if $a > b$ the result is the empty list `'()`.

```
(define (consecutive-ints a b)
  (if (> a b)
      '()
      (cons a (consecutive-ints (+ a 1) b))))
```

   In similar fashion, write a Scheme function `(consecutive-squares a b)` that evaluates to the list of perfect squares from a^2 to b^2. a and b should be integers; if $a > b$ the result should be the empty list `'()`.

```
; some examples:
> (count-positives (list 1 -23 0 -11 3 1002))
3
> (count-positives '())
0
> (sum-list '(1 2 3 4 5))
15
> (consecutive-ints -4 6)
(-4 -3 -2 -1 0 1 2 3 4 5 6)

> (consecutive-squares 1 10)
(1 4 9 16 25 36 49 64 81 100)
> (consecutive-squares 4 -6)
()
```