Language Assignment #1: Scheme

Issued: Wednesday, September 11
Due: Wednesday, September 25

Purpose

This assignment asks you to begin using a functional programming language named Scheme, which is a modern dialect of the venerable language Lisp. Lisp was designed by John McCarthy, at MIT, in 1958. Scheme was designed by Guy Steele and Gerald Sussman, at MIT, in 1975.

Documentation

Scheme is described in Sections 10.0–10.3.4 of our textbook.

The onyx cluster has a Scheme interpreter, which is documented at:

```
1 /usr/share/doc/sigscheme-0.8.5/index.html
```

and demonstrated by:

```
1 ~buff/classes/354/pub/sum/scheme
```

Better documentation, but for a different interpreter, is at:

```
1 http://www.gnu.org/software/mit-scheme/documentation/mit-scheme-ref
```

Assignment

Write and fully demonstrate a function named `replace`, with this interface:

```
1 (replace source target replacement)
```

The function replaces every instance of an object within `source` that matches `target` with a copy of `replacement`. Each argument can be an atom or a list.

For example:

```
(replace 1 1 2)
⇒ 2
(replace '(a (b c) d)
         '(b c)
         '(x y))
⇒ (a (x y) d)
(replace '(a (b c) (d (b c)))
         '(b c)
         '(x y))
⇒ (a (x y) (d (x y)))
```

Of course, you can write other functions and call them from `replace`.

You are required to use only the *pure* subset of Scheme:

- no side-effecting functions (e.g., `set-car!` and `set-cdr!`)
- no loops (e.g., `do`, `foreach`, and `map`)

Test your solution thoroughly. The quality of your test suite will influence your grade.

Finally, do not try to find a solution on the Internet. You'll possibly be asked to solve a similar problem on an exam, and if you have not developed a solution on your own, you will not be able to do so on the exam.