

CS 61A Summer 2009 Week 6A Lab
Monday 7/27 Afternoon

This exercise concerns the Metacircular evaluator that you can find on `~cs61a/lib/mceval-final.scm`

1. In the program, we supported the `(define (foo num) ...)` notation for defining a new function by first transforming it into the `(define foo (lambda (num) ...))`.

Alternatively, we can directly process the `(define (foo num) ...)` notation and make a new function and bind a variable to it. Modify the metacircular evaluator so that we can handle the `(define (foo num) ...)` notation without first transforming it into `(define foo (lambda (num) ...))` expression.

2. Implement the **special form or** for our metacircular evaluator. Remember that `or` evaluates from left to right and stops as soon as it sees a value that isn't false. For example, using the `=61=>` interpreter:

```
> (or nay nay 23 (/ 3 0))
23
```

3. The rest of the lab will familiarize you with Logo. You will build an interpreter for Logo for your third project.

To begin, type `logo` at the Unix shell prompt — **not** from Scheme! You should see something like this:

```
Welcome to Berkeley Logo version 5.5
?
```

The question mark is the Logo prompt, like the `>` in Scheme. (Later, in some of the examples below, you'll see a `>` prompt from Logo, while in the middle of defining a procedure.)

5a. Type each of the following instruction lines and note the results. (A few of them will give error messages.) If you can't make sense of a result, ask for help.

```
print 2 + 3
```

```
print 2+3
```

```
print sum 2 3
```

```
print (sum 2 3 4 5)
```

```
print sum 2 3 4 5
```

```
2+3
```

```
print "yesterday
```

```

print "julia"

print revolution

print [blue jay way]

show [eight days a week]

show first [golden slumbers]

print first bf [she loves you]

pr first first bf [yellow submarine]

to second :stuff
output first bf :stuff
end

second "something

print second "piggies

pr second [another girl]

pr first second [carry that weight]
pr second second [i dig a pony]

to pr2nd :thing
print first bf :thing
end

pr2nd [the 1 after 909]

print first pr2nd [hey jude]

repeat 5 [print [this boy]]

if 3 = 1+1 [print [the fool on the hill]]

print ifelse 2=1+1
  [second [your mother should know]]
  [first "help]

```

```

print ifelse 3=1+2
    [strawberry fields forever]
    [penny lane]

print ifelse 4=1+2
    ["flying]
    [[all you need is love]]

to greet :person
say [how are you,]
end

to say :saying
print sentence :saying :person
end

greet "ringo

show map "first [paperback writer]

show map [word first ? last ?]
    [lucy in the sky with diamonds]

to who :sent
foreach [pete roger john keith] "describe
end

to describe :person
print se :person :sent
end

who [sells out]

print :bass

make "bass "paul

print :bass

print bass

to bass

```

```
output [johnny cymbal]
end

print bass

print :bass

print "bass

to countdown :num
if :num=0 [print "blastoff stop]
print :num
countdown :num-1
end

countdown 5

to downup :word
print :word
if empty? bl :word [stop]
downup bl :word
print :word
end
```