Quiz 12b Rubric

1. (2 points) Which of the following is NOT an instance where you should force an expression? (circle one)
2. When something is being printed to the screen
3. The operation part of a procedure application
4. **All arguments for a special form**
5. All arguments for a primitive procedure
6. These are all instances where you should force expressions
7. (2+2 points) For the following questions, if any expression generates an error or infinite loop, just write ERROR. If it is a thunk, write THUNK.

Consider the following input to the lazy evaluator.

(define count 0)

(define (foo arg)

(let ((baz count))

(set! count 1)

(set! baz arg))

baz))

(define (bar)

(set! count (+ count 3))

15)

(define (foobar) (foo (bar)))

After evaluating these expressions, what is the value of count? **0**

What is the value of foobar? **A lambda**

Now consider this input:

(define count 0)

(define (foo arg)

(let ((baz count))

(set! count 1)

(set! baz arg)

baz))

(define (bar)

(set! count (+ count 3))

15)

(define foobar (foo (bar))) ;;; This has changed

After evaluating these expressions, what is the value of count? **1**

What is the value of foobar? **THUNK**

1. (4 points) The following interaction is run in the lazy evaluator:

(define (seleno a b c)

(if (> a 2)

(\* a b)

c))

(define (parac a b)

(if (> (\* a 2) b)

a

1))

> (seleno 3 (parac 2 (\* 3 1)) (\* 3 2))

How many times is \* called if thunks are memoized? **3**

How many times is \* called if thunks are not memoized? **3**

How many times is \* called in the normal metacircular evaluator? **4**

Which is more efficient for the code above? (circle one)

**LAZY evaluator**  NORMAL metacircular evaluator