Question 2

If you haven't done so already, open the Language page for reference as you work through the following stepping questions.

We have seen that `cond` is used instead of `if` when we have multiple parallel cases. For example, as seen in Question 1:

```
(define (mag1 x)
  (if (< x 0)
      "negative"
      (if (= x 0)
          "zero"
          "positive")))
```

Can be re-written as:

```
(define (mag2 x)
  (cond [(< x 0) "negative"]
        [(= x 0) "zero"]
        [else    "positive"]))
```

Now let's hand step the call to `(mag2 0)`

Question 2

1 point possible (graded)

What is the next step of the evaluation of:

```
(cond [(< 0 0) "negative"]
      [(= 0 0) "zero"]
      [else    "positive"])
```

- ( )
```
(cond [true    "negative"]
      [(= 0 0) "zero"]
      [else    "positive"])
```
- (•) ✔
```
(cond [false   "negative"]
      [(= 0 0) "zero"]
      [else    "positive"])
```
- ( )
```
(cond [(= 0 0) "zero"]
      [else    "positive"])
```

Explanation

Using the **call to a primitive** rule, the question (< 0 0) evaluates to `false`.

Submit

Answers are displayed within the problem