Started on	Thursday, 15 September 2022, 10:50 AM
State	Finished
Completed on	Thursday, 15 September 2022, 10:59 AM
Time taken	8 mins 46 secs
Marks	1.25/2.00
Grade	**6.25** out of 10.00 (**62.5**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Match the following Racket expressions with their corresponding values.

```
((lambda (x y)
   (cond
     [(< x y) (- y x)]
     [else (- x y)]))
 2 3)
```

1 ✓

```
((lambda (x y)
   (cond
     [(> x y) y]
     [else x]))
 2 3)
```

2 ✓

```
((lambda (x y)
   (cond
     [(< x y) 1]
     [(= x (* 2 y)) (- (+ x y) 1)]
     [(< x (* 2 y)) 2]
     [else 3]))
 4 2)
```

5 ✓

```
((lambda (x y)
   (cond
     [(< x y) 1]
     [(< x (* 2 y)) 2]
     [else 3]))
 5 2)
```

3 ✓

Your answer is correct.

The correct answer is:

```
((lambda (x y)
   (cond
     [(< x y) (- y x)]
     [else (- x y)]))
 2 3)
```

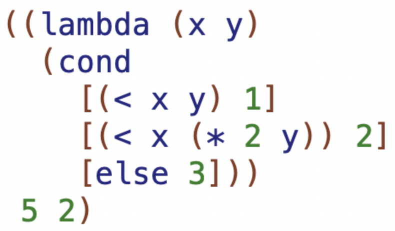
→ 1,

```
((lambda (x y)
   (cond
     [(> x y) y]
     [else x]))
 2 3)
```

→ 2,

```
((lambda (x y)
   (cond
     [(< x y) 1]
     [(= x (* 2 y)) (- (+ x y) 1)]
     [(< x (* 2 y)) 2]
     [else 3]))
 4 2)
```

→ 5,



```
((lambda (x y)
  (cond
    [(< x y) 1]
    [(< x (* 2 y)) 2]
    [else 3]))
 5 2)
```

→ 3

Question **2**

Partially correct

Mark 0.25 out of 1.00

Consider the following definition:

```
(define (compose f g x)
  (f (g x)))
```

Continue the following evaluation sequence (using Substitution Model for Racket) with the next expression (perform one step):

```
(compose (lambda (x) (+ 1 x)) (lambda (y) y) 3)
=> ((lambda (x) (+ 1 x)) ((lambda (y) y) 3))
=> ?
```

Select one:

○ a. There is no next expression, we already arrived at a value.

○ b. `(+ 1 3)`

○ c. `((lambda (x) (+ 1 x)) 3)`

● d. `(+ 1 ((lambda (y) y) 3))`✓

○ e. `4`

Your answer is partially correct.

The correct answer is: `((lambda (x) (+ 1 x)) 3)`