Questions 1-3

Question 1

1 point possible (graded)

Let's work through the HtDF recipe for the `on-tick` handler for the countdown program. Here's what the wishlist entry currently looks like.

```
;; Countdown -> Countdown
;; advances the countdown by subtracting 1, if the countdown is zero it remains at zero
;; !!!
(define (advance-countdown cd) 0)  ;stub
```

How many tests do we need for this function?

- 0
- 1
- 2 ✔
- 3

Explanation

We need to test both cases of our function. When Countdown is zero and when it is positive, because each case will be handled differently.

Submit

Answers are displayed within the problem

Question 2

1 point possible (graded)

Choose the correct function body for `advance-countdown`:

```
(define (advance-countdown cd)
  (- 1 cd))
```

```
(define (advance-countdown cd)
  (- cd 1))
```

```
(define (advance-countdown cd)
  (if (< cd 0)
      0
      (- cd 1)))
```

```
(define (advance-countdown cd)
  (if (= cd 0)
      0
      (- cd 1)))
```

✔

Explanation
We must first consider if the countdown has already reached 0, if so it should stay there, otherwise `advance-countdown` decreases the Countdown by 1.

Submit

Answers are displayed within the problem

Question 3

1 point possible (graded)
Choose the correct body for the function `render-countdown`:

```
(define (render-countdown cd)
  (place-image cd
               CTR-X
               CTR-Y
               MTS))
```

```
(define (render-countdown cd)
  (place-image (text cd TEXT-SIZE TEXT-COLOUR)
               CTR-X
               CTR-Y
               MTS))
```

```
(define (render-countdown cd)
  (place-image (text (number->string cd) TEXT-SIZE TEXT-COLOUR)
               CTR-X
               CTR-Y
               MTS))
```

✔

Explanation
Since `Countdown` is a natural number, we first have to convert it to a string, and then the string to an image using `text`.

Submit

Answers are displayed within the problem