Questions 4-6

Recall the partial data definition from Question 3:

Suppose we want to create a data definition to represent dinner reservations at a pop up restaurant. At this restaurant people simply reserve a place for themselves - they are seated with whoever is in line with them when they show up.

A person can either reserve a spot for one of the 100 spaces available each evening, or they can be placed on the standby list which doesn't guarantee them a seat. Here are the types comment and interpretation:

```
;; Reservation is one of:
;;  - Natural[1, 100]
;;  - "standby"
;; interp.
;;    Natural[1, 100] means a guaranteed seat for dinner where the number
;;                    corresponds to which reservation (not which seat).
;;    "standby"       means a standby spot, if all the reservations show
;;                    up this person will not be seated.
```

Question 4

1 point possible (graded)

How many function examples (tests) would a function need if it consumes `Reservation` and produces `true` if the reservation is the last one? [Enter a number]

3

Answer: 3

3

Explanation

We need at least 3 tests: 2 for the interval case, because the result is `true` if the reservation is 100, and `false` otherwise, and we need at least 1 test for the second case, `"standby"`.

Submit

Question 5

1 point possible (graded)

Which is the correct template for this data definition?

```
(define (fn-for-reservation r)
  (cond [(number? r) (... r)]
        [else (...)]))
```

✔

```
(define (fn-for-reservation r)
  (cond [(string? r)(...)]
        [else (... r)]))
```

```
(define (fn-for-reservation r)
  (cond [(<= 1 r 100) (... r)]
        [else (...)]))
```

```
 (define (fn-for-reservation r)
  (cond [(string=? r "standby") (...)]
        [else (... r)]))
```

Explanation
The `cond` question must be guarded with the appropriate type predicate, and the design of the template should match the types comment.
Note that we don't have to check `(<= 1 r 100)` in the `cond` question for the first case, because if `Reservation` is a number, then it must be `Natural[1, 100]` according to this data definition.

Submit

Answers are displayed within the problem

Question 6

1 point possible (graded)
What template rules did you use? [choose all that apply]

- [ ] `one of: 3 cases`
- [x] `one of: 2 cases` ✔
- [ ] `atomic non-distinct: Number[1, 100]`
- [x] `atomic non-distinct: Natural[1, 100]` ✔
- [x] `atomic distinct: "standby"` ✔
- [ ] `atomic distinct: false`

Explanation
According to the template that we designed:

```
(define (fn-for-reservation r)
  (cond [(number? r) (... r)]
        [else (...)]))
```

The template rules are:

```
- one of: 2 cases
- atomic non-distinct: Natural[1, 100]
- atomic distinct: "standby"
```

Submit

Answers are displayed within the problem