

Examples / check-expects

Examples / check-expects

1 point possible (graded)

Problem: Design a function that pluralizes a given word. (Pluralize means to convert the word to its plural form.) For simplicity you may assume that just adding s is enough to pluralize a word.

Continuing with the given problem, we now have a signature, purpose and stub.

```
;; String -> String
;; Produce the given string with "s" added to the end.

(define (pluralize str) "") ;stub
```

Which of the following check-expects is appropriate? (Check all that apply.)

☐ (check-expect (pluralize "cat") "s")

☐ (check-expect (pluralize "cat") "cat")

☒ (check-expect (pluralize "dog") "dogs") ✓

☒ (check-expect (pluralize "grass") "grasss") ✓

Explanation

Given "cat" this function should add "s" to produce "cats". So the first and second tests are incorrect. The third test is correct. So is the fourth, since adding "s" to "grass" should produce "grasss". Which just goes to show that this is not a very good pluralize function! But that's OK for now. The design method at least has made that clear to us.

Submit

i Answers are displayed within the problem