Homework, Week 4, part C

November 7, 2013

This builds on the tester you worked on yesterday. It's actually quite a bit harder, probably the most difficult thing you have done so far. I don't expect you to get it right, but I do expect you to make an attempt to get it right. The answer is below.

Alter the program to keep track of correct answers and print the number of correct answers at the end of the test. Hint: we wrote the `(get-answer)` function to return a one or a zero. Use the return value of this function to increment the variable `total-correct`.

```
1  (defun ask-question (item)
2    (format t "~%Question ~a. ~a~%" (incf current-question)(car
         item)) ;B change to increment current-question
3    (list-answers (cadr item))
4    (incf total-correct (get-answer (caadr item)))) ;C calls
         get-answer and increments total-correct
5
6  ;; checks to see if the current question is greater than
       total-questions
7  (defun give-test (items)
8        (cond
9   ;B change to add condition
10  ;C change to print the number correct after the test
11          ((or (>= current-question total-questions) (null items
                )) (format t "Test over, you got ~a correct.~%"
                total-correct) nil)
12          (t (let* ((len (length items))
13                    (num (random len)))
14               (ask-question (nth num items))
15               (give-test (remove (nth num items) items))))))
```