Homework, Week 4, part B

lightprogramming.org

September 29, 2016

This builds on the tester you worked on yesterday. This program contains a data structure that has questions and answers, and four functions, give-test (which gives the entire test), ask-question (which asks individual questions), list-answers (which lists individual choices for the multiple choice), and get-answer (which gets the response from the student).

One problem is that you cannot vary the number of test questions and tell the student the current question. Solve this problem.

There are many ways to do this. I choose to create a new function which declares and initializes three new variables. Barski probably would have done this at the top level. My answer is below. Please think hard about how you would do this before you look at my answer. I also changed `(ask-question)` to print the current question number, and `(give-test)` to end the test after the last question.

I now start the test with the simple command `(start-test 5 items)`, where 5 is the number of questions and items (as before) is the database of questions and answers.

```lisp
(defun ask-question (item)
  (format t "~%Question ~a. ~a~%" (incf current-question)(car item)) ;change to increment current-question
  (list-answers (cadr item)))

;;new version with limitation
;;part B
;;adds three variables, total-questions, current-question, and total-correct
(defun start-test (number items)
  (defparameter total-questions number)
  (defparameter current-question 0)
  (defparameter total-correct 0)
  (give-test items))

;; checks to see if the current question is greater than total-questions
(defun give-test (items)
      (cond
          ((or (>= current-question total-questions) (null items)) (format t "Test over.~%") nil) ;change to add condition
          (t (let* ((len (length items))
                    (num (random len)))
               (ask-question (nth num items))
               (give-test (remove (nth num items) items))))))
```