```
;DEFSTRUCT-RUN-TIME - 8185, 7587
;Test Case: Observe that no gensyms appear on the lcom of a fake-compile-file'd file with a
defstruct on it. Or repeatedly macroexpand (defstruct foo a b) and its subexpressions in Sedit.
; 7587 tested 18-Apr Lyric

(do-test "AR 8185 - Defstruct uses svref on non-simple vectors"
        (defstruct (foo (:type (vector float))) bar baz)
        (let ((x (make-foo :bar 1.2 :baz 3.4)))
             (and (typep x '(vector float))
                  (= 1.2 (foo-bar x))(= 3.4 (foo-baz x)))))

;DEFSTRUCT - 8053, 8360
;       for 8053, 8360: in ARs.
; 8053 - tested 18-Apr Lyric

(do-test "AR 8360 - Defstruct makes uncompilable constructor function with :named option"
        (defstruct (foo (:type list) :named) a b)
        (equal (make-foo) '(foo nil nil)))

; DEFSTRUCT - 7753
(do-test "AR 7753 - DEFSTRUCT's BOA-constructors don't use default values"
        (defstruct (foo (:constructor make-foo ())) (a 7))
        (= 7 (foo-a (make-foo))))

;STRUCTURE-PRINT - 7437, 7438
(do-test-group ("*print-circle* vs. *print-structure*"
                        :before (defstruct graph nodes))

(do-test "AR 7437 - circle labels always go to *standard-output*"
        (let* ((foo (make-string-output-stream))
                    (*standard-output* foo)
                    (*print-structure* t)
                    (bar (make-graph)))
               (setf (graph-nodes bar) bar)
               (write bar :circle t)
               (string= (get-output-stream-string foo) "#1=#S(GRAPH NODES #1#)")))

(do-test "AR 7438 - circular structures sometimes get lost altogether"
        (let* ((a (make-graph))
                    (b (make-graph :nodes a))
                    (*print-structure* t))
               (setf (graph-nodes a) b)
               (string= (write-to-string a :circle t)
                              "#1=#S(GRAPH NODES #S(GRAPH NODES #1#))")))
)
STOP
```