```scheme
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; Structure and Interpretation of Computer Programs, 2. ed.    ;;;;;
;;;;; Instructor Manual, Section 1.1, Exercise M1.2                ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; Student: Abrantes Araújo Silva Filho                          ;;;;;
;;;;; Date: 2019-02-11                                              ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;; Use the evaluation rule of section 1.1.3 to describe the process
;;;; of evaluating the expression:
(* pi (* radius radius))

; 1) This is a combination, so the interpreter evaluates the
;    subexpressions *, pi, (* radius radius), in any order.
;
; 2) The result of this initial evaluation is:
;
;       * => compound procedure to multiply
;
;       pi => value associated with the name "pi"
;
;       (* radius radius) => this is another combination, so the
;       interpreter evaluates the subexpression *, radius, radius, in
;       any order:
;
;          * => compound procedure to multiply
;
;          radius => value associate with the name "radius"
;
;          radius => value associate with the name "radius"
;
;       Now the interpreter APPLY the value of the first subexpression
;       "*" to the other "radius" and "radius", and RETURN the square
;       of the radius.
;
;       Now the interpreter APPLY the value of the first subexpression
;       "*" to the other "pi" and "square of radius".
;
; 3) The "tree accumulation" is as follows:
;       .
;     / | \
;    *  pi .
;          |  \      \
;          * radius radius
```