Instructions:

- We use the term “Scheme/Racket” as a short form for “Beginning Student Scheme/Racket.”
- Supply exactly the parts of the design recipe requested in each question. Unless otherwise told, “complete” a function means to provide just the function body.
- You may use a helper function where you feel it is needed. For each helper function, you are only required to write the function header and body.
- You may use any function defined in the exam as a helper function for any other function.
- We use arithmetic functions as well as the following built-in functions, described briefly here for your reference:
  - *max* produces the largest of two or more numbers
  - *min* produces the smallest of two or more numbers
  - *sqr* produces the square of a number
  - *even?* produces true if the consumed integer is even
  - *symbol?* produces true if the consumed value is a symbol
  - *string-length* produces the length of a string
  - *substring* produces a substring of the string that is consumed
  - *check-expect* is used for testing
- You may wish to use the following structure and data definitions:

```
(define-struct cal (month day))
;; A cal is a structure (make-cal m d) where
;; m is an integer in the range 1–12
;; d is an integer in the range 1–31
;; and m and d represent a calendar entry in a non-leap year.

(define-struct interval (start end))
;; An interval is a structure (make-interval s e) where
;; s and e are both cal structures
;; and s is no later than e in the same year.

(define-struct festival (name location cost dates))
;; A festival is a structure (make-festival n l c d) where
;; n and l are both strings,
;; c is the symbol 'free or 'charge, and
;; d is an interval.
```