A0- Racket fundamentals

Description

This course involves a lot of Racket Programming. This assignment is to get the students familiar with the parts of Racket relevant to this course. This assignment focuses on getting everyone hands-on with the Racket syntax and semantics. Although solutions to many of the problems in this assignment can be found online, it is strongly advised that the students solve the problems independently. This will ensure that the students develop enough proficiency in Racket, which is essential for future assignments.

The students can feel free to contact the TA to clarify their doubts and queries about Racket.

Reading

The Racket guide: Chapters 1 to 5 and 11 to 13.
The Racket Reference: Just for reference
RackUnit: Useful for testing

Assignment statement:

Solve the first 28 problems mentioned in Ninety-Nine Lisp Problems using only the following:

- Arithmetic (+, -, *, etc)
- car, cdr, and their variations like caddr, etc
- match
- if-then-else
- foldl and foldr
- length and append
- random (for problem 23)
- sort (for problem 28)
- Any helper functions defined by the student

Note:

1. For each question, implement at least three tests (not including provided examples).
2. Solutions for the previous problems can be reused to solve the current problem.
3. Students are advised to use recursion, higher-order function, and pattern matching extensively. However, any for/while loops will be penalized even though Racket supports loops.
4. Many of the questions can be solved directly by invoking a corresponding racket function. However, such implementation will be awarded zero points. For example, the

first question can be solved by calling the built-in last function of Racket. However, students are expected to implement it using recursion and other racket functions).

Submission instructions

The committed repo should be named as the student's *roll-number* and should have just the following two files:
rollnumber/

|- *rollnumber.rkt*
|- *rollnumber_tests.rkt*

rollnumber.rkt should have all functions required in the questions. Additional helper functions may be used.
rollnumber_tests.rkt should have all the tests cases. The testing will be automated. Therefore the submission format needs to be strictly followed. A 20% penalty will be levied if the format is not followed.