A01 Postmortem

Here are the most common errors we’ve found in each question while marking:

Question 2 (translation):

• Using (/ 4 3) instead of 4/3. 4/3 is the correct way to write fraction literal.
• Using (* r r) or (* (sqr r) r) instead of (expt r 3). (expt r 3) is the expected way to represent the cube of r.
• Using other mathematically equivalent expression. We expect an exact translation.
• Using some random operations that were not specified in the assignment. E.g. checking if parameters are positive (You don’t need to do it once specified in the requirement).
• Missing requirements for both parts.
• Writing the contract of b) as fib: Nat -> Nat instead of fib: Nat -> Num (fib produces inexact numbers).

Question 3 (sequences):

• Some students didn’t write any helper functions and resulted in repeating code. We expect you to avoid repeating code by writing helper functions.
• Some students forgot that (a, 0, 0, 0) forms a geometric sequence for all numbers a.

Question 4 (crl-points):

• Most common error: many students didn’t define constants for first place points, second place points, third place points, consistency bonus points, consistency bonus threshold, and major and minor deduction values.
• Many students missed the requirement that the symbol must be one of ‘good-standard, ‘minor-violation, ‘major-violation, ‘disqualified. This should be mentioned in the requirement.
• Some students didn’t write any helper functions and resulted in repeating code.
• Some students wrote ambiguous names for parameters, for example: (crl-points a, b, c, s).

**Question 5 (valid-date):**

- Some students didn’t define constants for the months used in code and for the starting and ending of missing dates in 1752.
- Some students didn’t write enough helper functions and resulted in repeating code.
- Some students’ solutions are too complicated, with a lot of long lines that are over 80 characters. We expect you to simplify your code as much as possible.
- Using `equal?` when the compared type is known. We expect you to use more specific predicate for comparison, such as `=` for comparing numbers, `symbol=?` to compare symbols
- Some students used a `cond` statement for the answer part of else.

**Global style:**

- A lot of students forgot to reference parameters in the purpose statement.
- Some students put the last 2 brackets in separate lines. This is unconventional in Racket.
- Some students use the wrong parameter names in the contract (Boolean vs Bool, Symbol/symb vs Sym).
- Some students include parameters in the contract.
- Some students put the design recipe components in wrong order.
- Some students didn’t write any design recipe for helper functions. Helper functions should include purpose, at least 1 example, and a contract.
- Several students didn’t include function names in contract.
- Some students interrupted the design recipe for main function by defining helper functions in the middle of it. Helper functions should appear in the top or bottom of the block of main function based on the rules in the style guide.