Midterm Post-Mortem

CS 135 Fall 2026

Q2a

- A concerning number of students did not understand the substitution rules with functions and variables.
- These students substituted `(f false 2 1)` and the constant `z` in the first step when the first step should have only substituted the function f
- Some students did not short-circuit the **`and`** function.

Q2b and Q2c

- Many students did not understand the **`cond`** substitution rules. For more details, please see the L03 slides.
- A concerning number of students evaluated the answer inside answer/question pair rather than short circuit the expression.
- Some students mistakenly believed that when multiple conditions in a **`cond`** statement evaluate to true, the function throws an error. In reality, the **`cond`** statement produces the value in the first true question/answer pair.

Q3a

- Many students had the wrong order for the divisions, with `(/ 2 ...)` instead of `(/ ... 2)`, while others simply forgot the denominator.
- Some values were added within parenthesis, such as `(2)`, which would produce an error where a function is expected.
- There were cases where the function inputs didn't match the values used in the function, e.g. `(define (trap-area p1 p2) (/ (* (- x2 x1) (+ y2 y1)) 2))`. While others ignored that the inputs should be points, and had `(define (trap-area x1 y1 x2 y2) ...)`.
- Finally, there were cases where the approach was more complex than needed, using: recursion, helper functions, cond, or absolute values. Not all of those made solutions incorrect, but they opened up possibility for mistakes, such as wrong cases for cond, or helper functions that were not properly defined.

Q3b

- Many students had incorrect contracts
- Many students used `(= length 1)` in their base case instead of `(empty? (rest lst))`
- Many students did not properly use the helper function `trap-area` (defined in Q3a) correctly
- Many students either only tested their function with 2 cases with the same number of points or with the empty list which is not a valid input

Q4a

- Many students still do not know the format of a contract or the different between a purpose and contract.
- Many students had incorrect bracketing around `(listof ...)`.
- Many students were missing `(anyof ...)` to combine types together.
- Many students did not capitalize the type name or capitalized the L in listof.

Q4b

- A large amount of students still do not know the difference between *cons*, *list*, and *append*, which caused a number of errors in their code.
- A large amount of students did not use (empty? lst) and (zero? n) for their base cases. The midterm did not cover general recursion and as such, base case conventions still needed to be followed.
- A lot of people did not have their base cases written such that (zero? n) was checked first. When this base case was not checked first, solutions would generally return the wrong output when n was equal to the length of the list.
- Some students did not use lockstep recursion in their recursive calls. mark should've been called as (mark (sub1 n) (rest lst)).
- A lot of people used accumulative recursion. This was not the correct approach for a number of reasons.
  a) Students were told not to use accumulative recursion, unless explicitly requested.
  b) All accumulative recursion solutions were inefficient compared to the intended solution.
  c) Most implementations would return the list in the reverse order of what it should've returned.
- A lot of students unnecessarily used helper functions.
  a) No helper functions were needed for this question.
  b) A lot of people used wrapper functions where they were not needed as there were no additional parameters used.
  c) Generally, the helper functions used made the code less readable, less efficient, and produced incorrect results.

Q5a

- Some students had the parameter `n` as a Num or an Int in their contracts, when it should've been a Nat.
- Some students had the parameters `x-sum` and `x2-sum` as Nats or Ints in their contracts, when they should have been Nums.
- Some students made mistakes in the computation of the variance, like squaring `x-sum` before dividing by `n` or subtracting the two terms the wrong way around.

Q5b

- A concerning number of students didn't use accumulative recursion.
- A lot of students produced 0 for the base case for their accumulative recursion.

- A few students did not properly handle the length of the list.
- A lot of students used too many helper functions.
- Many students did not use `var/sums` properly.

Q6

- A concerning number of students did not write the contract correctly.
- Many students returned only pairs that appeared adjacent in the given list
- Many students got ‘cons‘, ‘append‘, and ‘list‘ mixed up
- Some students used general recursion (L14 material)
- Many students tried to use accumulators and failed to output the result in the correct order. (they would output it in the reverse order then not use reverse to put it in the correct order)
- Many students did not use a helper function or they used more than one helper function which did not end up with a correct solution