

A04 Post Mortem

CS 135 Fall 2026

October 27, 2026

Contracts

- Students would put commas between data types
- Q2:
 - Many students were missing a requires statement `;; Requires: factor > 0`
 - Many students put the parameter names in their contracts instead of the data type example:
`;; after-days: factor cultures days -> (listof Nat)`
- Q3:
 - Many students thought the function consumed only `Int/Nat` and `(listof Int)/(listof Nat)` instead of `Num` and `(listof Num)` respectively
- Q4:
 - Students often wrote `(listof Num)` instead of `(listof Int)` for the second parameter of `curve-values`
 - Students often wrote `(anyof (listof Int) Bool)` or `(anyof (listof Num) Bool)` instead of `(anyof (listof Int) false)` for the produced value of `curve-values`, `Bool` is incorrect since the function only produces false and never true
 - Students often wrote `(listof Int)` to represent a curve polynomial which is a fixed length list of the form `(list Int Int 0 1)`
- Q5:
 - Students often wrote `(listof Int)` or `(listof Nat)` in their contracts for `sum-k` or `sum-c`
 - Students often wrote `(listof Any)`, `(listof Int)` or `(listof Nat)` in the contracts for `my-append`, `add-up` or `my-reverse`

Complexity

- Q2b:
 - Lots of student did not follow our recursion rule. The most common form of mistake is
`(after-days factor (next-day factor cultures) (sub1 days))`
Note that this breaks our rule because only one argument can change.
 - And a reminder to use `(sub1 days)` instead of `(- days 1)`
- Q3:
 - Please take a look at the sample solution if your code ended up using more additions or multiplications than we asked you to.
 - Some students used a helper function for `eval-poly`, but it can be condensed into one function.
 - Some students used accumulative recursion for this question, but it is not allowed yet.
- Q5d:
 - Many students did not include numerical estimates of the number of steps for a list of length 100
 - Many students stated that `(add-up (my-reverse lon))` was the line causing the problem