

# Post-Mortem

CS135 Winter 2024, Assignment 8

## Question 1b

- Many students inconsistently used **X** and **Any** in their contracts.
- Many students also used **Any** in their contracts. Although no marks was taken off, you likely got a comment saying that using a type variable such as **X** would be more specific.
- For the produced data type, many students wrote `(listof (listof ...) (listof ...))` instead of `(list (listof ...) (listof ...))`.
- Many students wrote helper functions that mimicked `filter`, even though its behaviour was banned. Luckily, only a minor deduction was warranted.
- Many students used `append` in their solutions which will lead to exponential blowup when your function is tested on large lists.
- Some students wrote multiple helper functions that was similar in behaviour (such as one to determine the elements in the list that satisfy the predicate, and one to determine the elements that did not). It should be clear that this idea can be abstracted by including an additional argument whose value is the appropriate predicate.
- As an aside: Although there was nearly no solutions that solved this question while only going through the consumed list once, other solutions that were in  $O(n)$  complexity (i.e., mainly solutions that used accumulative recursion; no `append`) still received the marks (and likely did not have a test case that did not result in their function timing out).

## Question 2

Parts a, b:

- Many students used **Any** instead of a type variable, such as **X** in their contracts.
- Some students wrote `pred` or `function` instead of writing the contract for the function when passing a function as a type.

Part c

- Many student did not describe what the binary relation operator produces in their purpose.
- Some student incorrectly described the first consumed argument in the purpose (which should be a list of two functions with the first function being a predicate and the second one being a binary relation operator).

### Question 3

- Some students did not represent the predicate as a consumed function, and instead used a name such as `pred` in their contracts. We do not have a data type named `pred`.
- Many students neglected that the predicate for Q3 consumes a `Nat`.
- Some students wrote used `Node` instead of `BT` in their contracts. The produced function should consume a Binary Tree.
- Many students did not enclose the produced and consumed functions with brackets.

### Question 4a

- Many students used `(nested-listof Any)` instead of `(nested-listof X)` in the contract.
- Some students missed the brackets around `(nested-listof X)` in the contract.
- Some students incorrectly used `cons?` instead of `list?` in the template, even though they are not interchangeable.
- Some students did not apply `nested-listof-X-template` to `(first nested-listof-X)` when `(list? (first nested-listof-X))` case was true.