Post-Mortem
Assignment 05
February 22, 2018

We normally publish the post-mortem for an assignment after it has been marked and released. Here is a list of common errors provided by the graders for assignment 5.

General

• Many students are still missing examples for base case(s) throughout the assignment. When writing a recursive function, there should be an example for every base case, and an example for at least one of the recursive cases, as discussed in the style guide.

• Many students are still missing parameter references in their purpose statements. Purpose statements should meaningfully use each parameter name, and the parameter names should be written exactly as they appear in the function header.

• Constants (and helper functions) should be defined above the design recipe for the function they are used in. Some students still defining their constants and helper functions between their examples and function definition, or after the main functions specified in the assignment.

• All design recipe components, except for tests, are required for helper functions. Some students did not include these design recipe components for their helper functions.

Question 1

• In part (a), some students did not correctly account for the fact that 1 is not a prime number.

• In part (a), many students missed the requirement that the consumed natural number had to be strictly greater than zero.

• In part (b), some students did not correctly compute the result of calling (next-prime 0).

• In parts (b) and (c), some students did not use prime? as a helper function. This often lead to overly complex solutions that did not use pure structural recursion and/or had memory issues with large numbers.

Question 2

• Throughout the question, many students used (listof FileInfo) in their contracts instead of Directory. However, a Directory has additional requirements compared to a (listof FileInfo), and contracts should always use the most specific type available.

• Some students used Directory or FileInfo as parameter names in their functions. Unless you are referring to a proper noun, remember that using capital letters in identifiers goes against the style guide.
• In part (a), many students did not define a constant for the conversion rate from a kilobyte to a megabyte (or vice-versa).

• In part (b), many students did not include the full requirement that the consumed pattern string must start or end with an asterisk character, and contain exactly one asterisk.

• In part (b), many students had issues in their solutions when the consumed wildcard pattern contained either part of the file format, or part of the file name.

• In part (c), some students did not include the requirement that the consumed symbol must be one of 'name or 'size.

• In part (c), many students did not correctly account for the case where the consumed Directory would be sorted by 'size, but more than one of the files has the same size.

• In part (d), many students had solutions that would produce true if the consumed string was the empty string, even though the empty string is not considered a valid path.

Question 3

• Many students had formatting issues with their data definitions. Please refer to the course notes and take a look at the sample solutions to see how they should be formatted.

• In part (a), many students had issues with their data definition of an RLEPair. Since an RLEPair is a fixed-length list, the data definition should specify that it is a list of exactly length two. Using listof anywhere in the data definition indicates that an RLEPair can be of arbitrary size, which is not the case.

• In part (a), some students did not include the requirement that the second element of an RLEPair had to be strictly greater than zero.

• In part (b), many students did not make use of a helper function to deal with individual RLEPairs from the consumed RLEList, or their helper functions involved taking the first, second, and rest numerous times, which lead to unnecessary code complexity. Please see the model solutions on how this can be avoided.