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 6.

**General/Ongoing Errors**

- 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 making extremely poor choices for the names of their parameters and helper functions. In particular, cryptic and ambiguous function names such as `accumulate`, `recurse`, `ct`, `newlist`, `change`, `helper`, and `function` are some of the many poor function names that markers encountered.

- 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 are 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. Many students are still not including these design recipe components for their helper functions.

- Many students did not complete the assignment.

**Question 2**

- In part (a), some students indicated that the function produces (anyof Str Sym). However, the function can only produce the symbols `orange`, `krampus`, and `coal`, so explicitly stating these symbols in the contract would make the contract more specific.

- In part (b), many students did not indicate that the new list of child wishes must be sorted in non-increasing order.

**Question 3**

- In part (a), some students had a solution that produced the last column of the consumed Grid when the consumed natural number is exactly one more than the number of columns in the Grid, indexed from zero. However, empty should have been produced instead.
• In parts (a) and (b), many students did not specify that the consumed or produced list must only contain the symbols ‘B’ and ‘-’.

• In part (b), many students had a solution where they removed all the consecutive ‘B’ s at the beginning of the consumed list in their recursive calls. However, such a solution would use generative recursion. It is true that this gets closer to the base case of a list, but removing more than one element does not get one step closer to the base case.

• In part (c), some students indicated that the function produced a (listof (listof (listof Nat))) in their contracts. However, the function always produces a list of length 2, so the contract should communicate this to be as specific as possible. This is similar to the contract on question 3c of the midterm.

• In part (c), many students simply produced empty if the consumed Grid was empty. However, the assignment states that puzzle-labels always produces a list of length 2, which empty is not.

Question 4

• Some students had issues with handling the case where the last digit in the consumed list was exactly zero.

• Some students simply used the same solution in parts (a) and (b). However, doing so would not receive any marks for one of the parts, based on the type of recursion used.

• In part (b), some students had separate functions to sum up the numbers at the odd and even positions. However, this lead to two functions that are almost identical. Instead, this can all be done in one function.