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

**Question 1**

- A lot of students did not define a constant for \( c \).
- Some students used \((\text{define } c ~ 3 \times 10^8)\), which is not recognized by DrRacket and will cause syntax error.
- Some students used \((* m c c)\), which translates to \(m*c^2\) instead of \(mc^2\).

**Question 2**

- In part (a), many students used \(\text{boolean=?}\) or \(\text{equal?}\) to compare boolean values with \(\text{true}\) or \(\text{false}\), which evaluates to the same result as the boolean value itself or simply calling \(\text{not}\).
- In part (b), some students used nested \(\text{cond}\), which is explicitly banned in the question.
- In part (c), many students lost marks, because they did not know what general or reusable meant. Students are encouraged to practise their ability of reading and evaluating code written by others.

**Question 3**

- Some students used brackets around parameter names. Brackets should only be used for a function call.
- No marks deducted this time, but many students used nested \(\text{cond}\) in an else statement.
- In part (a), many students did not call \(\text{card-template}\) in \(\text{loc-template}\).
- In part (a), some students used \(\text{empty?}, \text{cond}, \text{and recursion}\) for \(\text{card-template}\), which they should not. By data definition, \(\text{Card}\) is a fixed-length list.
- In part (a), some students missed base case and recursive call for \(\text{loc-template}\), which is required, since \((\text{listof Card})\) has a recursive definition.
- For part (a) in general, remember templates should always closely follow data definitions.
- In part (b), some students used wrong combination of \(\text{first}, \text{rest}, \text{second}\) to access the suit of a \(\text{Card}\) in a list.
- In part (b,c), many students missed the base cases or recursive calls.
• In part (b,c), some students provided incorrect recursive calls. i.e. incorrect number of arguments, incorrect usage of cons, keeping an element when it is supposed to be removed etc.

• In part (b,c), some students used generic comparisons when the type is already specified.

• Overall, part (b) and part (c) were well done.

Question 4

• Many students missed the test where the slope is a fraction or zero.

• Some students did not refer to parameter names meaningfully in their purpose.

• Overall, this question was very well done.

Question 5

• Some students forgot to substitute all of their parameter values at once for a function call.

• Some students didn’t remove the (cond [else ...]) statement when else was evaluated, and carried it along the question.

• Overall, this question was very well done.

Question 6

Question 6a

• Some students forgot to use first when taking an element from the list of keys and values.

• Some students used cons to add an element onto a non-list.

• Some students did not correctly cons the key-value pair onto the recursive call.

• Some students used >?, =>? and =<?, instead of just >=, <=, < and >.

Question 6b

• A lot of students used = instead of char=? to compare Chars.

• Some students used (first (second al)) instead of (second (first al)) to look up the value.

Question 6c

• Many students forgot to convert the result back to a string using list->string.

• Many students did not use a helper function to deal with a (listof Char), which often resulted in incorrect argument type for a recursion call. i.e. using (listof Char) as an argument for a function call where the parameter has type Str.

• Many students didn’t use lookup-value from part b) as a helper function. This meant that some students had errors when recreating that behaviour in their encode function. It also typically meant that after searching through a number of KVPs to find the encoding of a particular character, the AL was not reset to its starting form. This meant that any lookups for KVPs that were discarded in earlier lookups could not be completed.
• No mark was deducted this time, but some students did not directly use `lookup-value` from part(b), but defined another function that is exactly the same. Students may and should directly use functions defined in earlier parts of a question as helpers.

• No mark was deducted this time, but many students defined helper functions with name “helper” or “decode/helper”. Students should give meaningful names to their functions, parameters, and constants.

Question 6d

• Many students forgot to convert the result back to a string using `list->string`.

• Many students did not use accumulative recursion (correctly) to solve this question.

• Many students accidentally used list functions on strings (or vice versa).

• Many students didn’t reset their association list to the original list between searches for different characters.

• Some students forgot to use reverse to keep their produced list in the correct order.

• Some students did not write their look up function correctly.

• Some students incorrectly compared characters.