Question 1

Part a

- Some students determined the first four TensNats by adding 10 to the previous TensNat, instead of multiplying the previous TensNat by 10.

Part b

- Many students were missing the recursive call to the tensnat-template.
- Many students did not divide the consumed TensNat by 10 to get one step closer to the base case.
- Some students attempted to divide the consumed TensNat by 10, but they used the ÷ symbol or a backwards slash instead of the / Racket function.

Question 2

- Many students indicated that the most appropriate type for the names of the items was a Str. However, the number of items purchased in a transaction can vary, and the best way to represent this data of varying size is using a (listof Str).
- Many students indicated that the most appropriate type for the total amount was a Nat. However, purchase amounts can be decimal numbers as well, so Num would be the most appropriate type here.
- Some students did not include a requirement that the name(s) of the item(s) cannot be empty, whether they indicated it as a Str or a (listof Str).
- Some students did not include a requirement that the total amount must be greater than zero (greater than or equal to zero was also acceptable).
- Some students indicated that the most appropriate type for the payment option was a Str. However, symbols should be used instead of strings to represent a small collection of items, which is the case here.

Question 3

Part a

- Some students were missing contracts for their template. Remember that contracts are always required as a part of template functions.
- Some students were missing ellipsis in between the selectors in their template.

Part b

- The function definition was generally well done.
- Many students did not define constants for the different payout rates.
Part c

- Generally well done.
- Some students forgot to include the purpose and contract with their helper function(s).
- Some students had a helper function that counted medals and produced a Num rather than a Nat.
- Some students placed their conditions in the wrong order, which would produce a different result in some cases.
- Many students indicated that the function produced a (listof Str). However in this case the list is of fixed length, so it should be (list Str Str). Remember that contracts should use types that are as specific as possible.
- Some students did not use (anyof ...) to describe more than one type being produced.
- Some students produced the entire Medals structure rather than simply the name of the country.
- Some students could not properly form a list of two elements. The easiest way to do this is (list x y).
- Some students did not handle the case where values were equal.

Question 4

- Generally well done.
- Some students simplified multiple arguments in one step, or did not fully copy all of the arguments from one step to the next.
- Some students substituted constant values too early.
- Some students did not substitute argument values for all instances of parameters in the user-defined function substitution.
- Some students incorrectly substituted (posn? make-boat 'canoe 32) with the value true. In this case, if they followed the rule for (or true ...) and produced true in the next step then they were given credit for that. However, since this error trivializes the question, they were not given credit for the incorrect final result.
- Some students identified that part (c) would produce an error, but they did not describe the error. That was required to get a check mark.

Question 5

Part a

- Some students did not include comments for their tests.
- Some students did not include exclamation marks at the end of the expected results for their tests.

Part b

- Some students forgot to cons the first character in the case where the character does not match.
- Some students did not wrap the exclamation mark in a list for the base case.
• Some students did not correctly handle the conversion of the consumed string into a list of characters, and then back into a string at the end.
• Some students used `equal?` instead of `char=?` when comparing two characters.
• Some students assumed that an exclamation mark should not be added at the end when no characters in the string matched.
• Some students named their helper functions poorly, such as using "helper" or "recurse".
• Some students did not correctly triple a character when the character matched. This was often done using `(list ...)`, which creates a fixed-length list.
• Some people used string functions other than `string->list` and `list->string` when the instructions stated that this was not allowed.

**Question 6**

**Overall**
Some students misunderstood the question and thought the function should produce true in cases where there were single digits in string with many other characters. Strings such as "ablcd" do not represent a single digit integer. The name of the function is `small-int?`, not `contains-small-int?`. Furthermore, there would be no reason to specify rules regarding the "+" and "-" signs if all that was expected was to check whether a string contained a numeric character. Where possible part marks were given in part (b) for students who attempted to produce `true` in these cases. All exams were marked consistently. Do not make a remark request based on this misunderstanding of the question. The mark will not be changed.

**Part a**
• Generally well done.
• Some students did not have at least one example that produced `true` and one that produced `false`. That is expected for predicate functions.
• Some students had the function producing `true` when the value consumed was not a string.

**Part b**
• Generally poorly done.
• Most students did not check to see if `s` was a string before using functions like `string-length` or `string->list`.
• Some students did not check the length of the string before trying to access individual characters in the string.
• Some students did not handle the case for 0 properly. "+0" and "-0" should produce `false`, but "0" should produce `true`.
• Some students did not check that the length of the string needed to be 1 or 2.
• Some students tried to use `string-ref` and `string->list`. Only one of these was necessary.
• Some students tried to compare characters with numbers like -9 or 9.
• Some students tried to use `char-numeric?` on a string. It must consume a character.
• Some students used `cond` in their solutions, but the instructions stated that you were not allowed to use `cond`. 
Question 7

Parts a and b

- Generally well done.
- Some students did not correctly access the field values of the Hands and Cards.
- Some students used `make-hand` instead of `make-card` in part (b).

Part c

- Some students incorrectly indicated that the expected value of `rank->value 1` was 1.

Part d

- Some students did not include a test for the boundary point of 10.
- Some students had redundant test cases by including two tests for values less than 10.

Part e

- Many students did not indicate that the base case is when the rest of the list is empty, since the function consumes a non-empty list.
- Some students did not use `(card-suit ...)` to access the suit of the first Card in the list.
- Some students treated the recursive case as if the function should produce a symbol instead of a boolean value.
- Some students did not process the consumed value as a list, or treated the consumed value as a list of fixed length.