Question 1

- Some students tried to use the $\pi$ symbol to refer to the value of pi. Racket’s built-in pi constant is just $\pi$.
- Overall, this question was done well.

Question 2

Part a

- A common answer was (anyof Int Bool). Since we may be dividing two integers (in the else case), which may produce a Num, the correct answer would be (anyof Num Bool).

Part b

- Since zero is even, the function never divides by zero. Hence, no requirements were necessary.

Part c

- Most students forgot to check the boundary case for when $a = 64$.

Question 3

Part a

- The most common answer was 4 or 6 (but the correct answer was 2). It is important to read a question thoroughly and carefully before attempting it.

Part b

- A common (incorrect) answer was (list 42 (list "W" empty)). Please note that (cons item list) simply adds item to the beginning of list.
- Many students forgot that empty was an element of the given list.

Part c

- A common (incorrect) answer was (rest (first (first my-lst))). A good rule of thumb to avoid this is by remembering that rest produces a list.

Part d

- Many students explained that quotation notation would turn empty into ’empty, but the correct way to represent empty in quote notation is ’().
- A few students said that quote notation would not work because the list has strings in it, which is incorrect.
- Some students wrote empty instead of ’() in their quoted notation representation, causing it to turn into the symbol ’empty instead.
• Some students forgot to include `empty` in their quoted lists.

Part e
• Some students forgot `(cons empty empty)` after `(cons "W" ...)`.
• Many students had too many `cons` or too few `emptys`.
• Many students had trouble with the nested lists and represented `(cons (cons "X" ...))` as `(cons "X" ...)`.

Question 4

Part a
• It is incorrect to replace `default` with its value in the function body in the second step.

Part b
• To do this question correctly, it is important to realize that 3 is a positive number.
• When we reach `(or true expr)`, the stepping rules indicate that `expr` is not evaluated and `true` is immediately produced.

Part c
• Some students did not substitute the value of `default` in the first step, and instead substituted the value of `my-list`.
• Some students wrote out the entire function body of `rec` for steps 1 and 2, which led to a lot of unnecessary and incorrect work.
• Some students indicated that `positive?` or `negative?` would produce an error, because it is given a list instead of a number. It is important to realize that the list given to the function `rec` is a one-dimensional list.

Part d
• This question was poorly done overall.
• Many students simply replaced with `(post-text my-post)` with "Text", skipping the step where the value of `my-post` needs to be substituted.
• For the students who correctly replaced `my-post` with its value, many students incorrectly wrote `(make-post "Text" 'Dan default)` in the first step. It is important to realize that the definitions have already been fully simplified.
• Though not part of the correct solution, many students who proceeded to convert "Text" into a list did not convert it to a list of characters.

Part e
• Some students skipped substituting the value of `my-list` in their first step.
• Some students changed the `cons` from their first step into `list` into the second step, instead of simply taking the first element. This does not result in the same list.
• This question was well done overall.
Question 5

- This question was well done overall.
- The solution required conditions for when the consumed list is empty, and when the consumed list only has 1 element. Many students were missing one of these base cases.
- Some students checked if the first element of the list was strictly less than (<) the second element, and proceeded to recurse if this condition was true. However, based on the first example given, the <= inequality should be used instead.
- Some students did not properly extract the second element of the list when comparing it with the first element.

Question 6

- Some students did not use string->list on s to turn it into a list of characters for substr/list, causing them to use string->list and list->string everywhere in their solution, resulting in extremely messy and unreadable code.
- Many forgot to use list->string on the output of substr/list.
- Many students forgot one or more of the base cases.
- Many students had trouble framing the appropriate recursive calls for each case (usually, with most implementations, when start > 0 and start = 0).

Question 7

Part a

- Many students forgot the restrictions on the price and type of the cellphone.
- Some students forgot to write the contract for the template.
- While conditional statements were usable in this question, they were usually not used correctly.

Part b

- Many students returned the price of the cell instead of a Cell.
- Many students forgot to floor the price of a cellphone, even though it was a hint in the question.
- Some students had difficulty correctly ensuring the price was always at least $25. For example, some students checked if the base price was below $25, rather than checking if the particular discount would lower the price to below $25.
- Some students had trouble calculating the discounts for Android phones.
- A few students did not correctly use the selector functions for the Cell structure.
- Many students did not define and use constants for the numerical values in this question - particularly for the $25 minimum price.
- Some students rounded down prices that should not be rounded, and some did not round down prices that should be rounded.
Part c

- This question was poorly done overall.
- Many students put the entire key-value pair into the produce list, instead of just Cells.
- Many students did not refer to their parameter names in the purpose. This is a recurring issue from the assignments that must be kept in mind when writing purpose statements in the future.
- A few students had trouble with the contract. In continuation, many students violated their contract in their recursive calls.
- Many students had trouble properly selecting the inventory level from the Cell Inventory using second.
- Many students handled the dumb phone logic incorrectly.
- Many students handled at least one recursive case properly but failed to handle all the recursive cases correctly.
- Some students wrote functions that produced true and false instead of a (listof Cell).