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 9.

**General**

- Many students submitted code that did not run at all, hence they lost a great amount of correctness marks. You are advised to make sure your code runs.
- All the functions in this assignment warrant 2-3 distinct examples. These should appear after the purpose and contract and before the function definition.
- Overall, this assignment was extremely well done!

**Question 2**

- Some students did not use constants in their code. Although no marks were deducted this time, the use of constants is still warranted by the question.
- Part (a) was very well done!
- A vast majority of students were able to complete part (b) correctly, but some had very messy and hard-to-read solutions. Please try to make sure your code is clean and readable, and make use of the auto-indenting features available on DrRacket.
- Some students submitted an incomplete solution for part (b) that had syntax issues like missing parentheses that made the file un-runnable.

**Question 3**

- Many students only had one example for this question.
- Many students that used Sym in their contract forgot to include the requirement that the argument should be either 'arithmetic or 'geometric.
- Some students used Nat or Int in their contract to represent the first two numbers of the potential sequence, which is incorrect.
Question 4

- A vast majority of students used (listof Any) in their contract instead of using the (listof X) format. The (listof X) format conveys more information about how the types of the arguments and the type of the result of the function are related. For example, if a function foo manipulates a (listof Num) to form another (listof Num), a (listof Str) to another (listof Str) and so on, the contract foo: (listof X) -> (listof X) conveys more information than foo: (listof Any) -> (listof Any) since the first contract guarantees that (foo list-of-num) is a (listof Num), whereas the second one implies that (foo list-of-num) could be a list of other types, such as a (listof Str). A small penalty was applied for this.

- Many students had only one example for some of the functions they defined.