[Question 1 (4 pts)] In your own words, explain the process of TDD.

Write Tests and ensure they pass; if not, THEN write the code to fix the bug. Every few tests, refactor your code to ensure clean, concise methods. This immensely increases understanding of how your code works in comparison to how it *should*, and hence, dramatically reduces debug-time.

[Question 2 (6 pts)] TDD proponents claim that it benefits software development. The advantages often associated with TDD are (a) it increases the confidence software developers have in their code, and (b) it improves overall code quality. Do you agree with these claims? Justify your answer.

I definitely agree! By writing tests prior to the main code, I know precisely what and where to look-at for errors in my code. (i.e. since my code worked for 1, 2, & 3 but not 4, the problem must have been with how I handled inputting multiple prime factors into the list)

[Question 3 (5 pts)] Based on your experience with this lab, please specify advantages and disadvantages of TDD.

TDD keeps each member of the team active and *in the loop*. Debugging is substantially easier since everything occurs literally one step at a time.

If using repos, the back-and-forth TDD style used in this lab can be extremely time-consuming due to the consistent need to update over and over.