1. Did you find more test cases via black box or white box techniques? Do you believe adequate testing could be done with just one of the techniques? Was it easier to develop black box or white box tests? Why is one faster than the other?

Answer => White Box testing was more helpful for me and I found more cases with that technique as in white box testing we know how the code is supposed to work and that gives us more power and control over the testing and thinking about newer ways of testing the code. Testing the code with just one technique cannot be adequate as there are different methods involved in both of these techniques while the black box testing is best for user-oriented issues the white box testing reveals the internal problems with the code and also the edge cases. If we just rely on one technique then it will lead to incomplete testing and some issues may be missed. In general black box tests are easier to develop as it doesn’t require in depth analysis of the code and we can solely focus on the expected behaviors and inputs from a user’s point of view and because of this sole reason Black-Box testing is more faster to set up and execute since it does not require the tester to go in detail of the code.

1. Consider how you would set up integration tests for the functions above. What would be your general approach to creating integration tests? Do you need to create additional code to set up and run the test? How will you write code to compare the results to ensure they are correct? How much additional time do you think writing the additional code will take?

Answer => To set up integration tests for the functions, we would need to test their interactions and functionality when used together as a part of the overall system. We need to determine which functions need to be integrated, identify their dependencies, and identify specific test scenarios that involve the integration of these functions. Then start creating the necessary test environment, including any required data structures or variables. After the environment is set up, we can call the functions and simulate the integration by invoking them in the appropriate sequence and providing the necessary inputs based on the defined test scenarios. It takes different amounts of time depending on how familiar the developer is with the program but if we have to estimate then it would be around 1-2 additional hours of coding

1. Create one integration test for the combination of two functions above. Show the code you created to set up the test, execute the test and compare the result to the expected result. You only need to demonstrate one set of test data, but it should be obvious how to add more tests easily. Comment your code to point out the set up, execution, and comparison parts of the code.

A screen shot of a computer program

Description automatically generated