PLEASE ANSWER 4 OF THE FOLLOWING 5 QUESTIONS. NOTE: DO NOT ANSWER 5 QUESTIONS. IF YOU ANSWER 5, I WILL GRADE YOUR FIRST 4 AND IGNORE THE 5TH.

1. Imagine that you were testing the AutoCorrect features in Microsoft Word.

- Explain how you would develop a set of scenario tests that test this group of features.

- Describe a specific scenario test that you would use to test this feature. It is not necessary to describe your test step-by-step, but it is necessary to provide enough detail that I understand your test.

- Explain why this is a particularly good scenario test. In your explanation, be explicit about what you consider to be the criteria under which a scenario test is evaluated. (That is, what are the attributes of a good scenario test and how does this test relate to those attributes?)

2. Suppose you use a state model to create a long series of tests. You can decide whether or not the series order or test data selection is random. What should you use as a stopping rule? Compare three alternatives, one of them being that you stop after the program runs without error for a sequence of 200 computer-hours. What assurance do you have of sub-sequence coverage in these cases?

3. You are testing the group of functions that let you create and format a table in a word processor (your choice of MS Word or Open Office).

Suppose that a critical requirement for this release is **scalability of the product.** What scalability issues might be present in the table? List three. For each issue, list 2 types of failures that could involve scalability. For each type of failure, describe a good test for it and explain why that is a good test for that type of failure. (There are 6 failures, and 6 tests, in total). (NOTE: When you explain why a test is a good test, make reference to some attribute(s) of good tests, and explain why you think it has those attributes. For example, if you think the test is powerful, say so. But don't stop there, explain what about the test justifies your assertion that the test is powerful.)

4. Why is it important to design maintainability into automated regression tests? Describe some design (of the test code) choices that will usually make automated regression tests more maintainable.

5. Define statement coverage, branch coverage, and multicondition coverage. Now describe a small program, describe a set of tests that would completely test it (in terms of statement, branch and multicondition coverage), and describe some bugs that these tests would miss. Your description of the program can be pseudocode, a flowchart, or other simplifying diagram.