

Project 4 Overview

The Sage County Water Plant serves customers in Sage County.

Here are the county business rules and monthly pricing guidelines:

- All customers get their first 2,000 gallons of water for a flat rate of \$8.
- The next 3,000 gallons cost \$0.004/gallon.
- Any gallons above 5,000 are billed at \$0.007/gallon.
- Low-income households get a flat \$4 credit, but NOT if their total usage exceeds 8,000 gallons for the month.
- Billing is based on meter readings that are recorded in 1-gallon unit increments.
- There is a mandatory infrastructure surcharge charged as follows:
 - \$4 if usage \leq 2,000 gallons
 - \$12 if usage over 2,000 but \leq 8,000 gallons
 - \$20 for high-usage customers
- There is also a 2.5% county tax applied to the water amount, but not to the infrastructure surcharge. This tax is waived for low-income households.

You are to test the software that calculates the billing amount. This will be done in six steps, to be turned in as two assignments – HW 4 and Project 4 – using the following guidance:

STEP 1: Write the test plan for the water plant's billing calculator. Hint: All test cases will have two inputs and one output. You can use the following example boundary test case as your first test case and as a template for the remainder of your test cases.

Test Case 1: Boundary test for 2000 gallons used

Input:

- Gallons_Used = 2000;
- County_Aid = False

Expected Output: \$12.20

Calculation Used: $\$8 + \$4 + (\$8 \times 0.025)$

Actual Output:

Pass/Fail:

STEP 2: Feed the specs to ChatGPT (or your favorite AI tool) and have it generate the code for the billing calculator. Hint: The “specs” refer to the business rules on the previous page.

STEP 3: Test the code by running your previously-written test cases against the code and document the results. Hint: “Document the results” means you should record the actual output and indicate whether or not the test has passed or failed. You do not need to fix the code if the test does not pass; instead, document the test as failed.

IMPORTANT! Your test plan and the AI-generated code should be turned in as Homework 4. This is due on the first Tuesday after this assignment has been given.

STEP 4: Build a code graph for the code that was generated by the AI tool. Hint: You can ask the AI tool to do this for you, but, if you do, it is your responsibility to ensure the graph correctly depicts the control flow through the code. Adhere to the graphing conventions used in class: all contiguous code statements should be shown as a rectangular node in the graph, and all branch statements should be depicted as a diamond-shaped node.

STEP 5: Figure out how attain statement, branch, and path coverage withing the graph. Hint: Refer to the example in the lesson slides on how to do this. The slide is on the next page for your convenience.

STEP 6: Figure out if the test cases you wrote in Step 1 (and ran in Step 3) satisfy the coverage goals you determined in Step 5.

If you can complete the coverage goals using the test cases already defined, indicate which test case follows each path. Hint: There may be more than one test case that follows the same path. If that happens, you only need to provide ONE test case that follows the path.

If you cannot complete the coverage goals – in other words, if you find an untested path – then figure out a new test case that would cover the untested path, or execute the untested statement. Be sure these additional test cases are documented separately from your original test plan.

Project 4 is due one week after Homework 4. Be sure to include all six steps in Project 4. If you were given feedback on Homework 4 indicated something should have been done differently or could have been done better, be sure to make those corrections in the final project.

APPENDIX A: Sample Code Graph with Coverage Analysis (taken from the lesson slides)

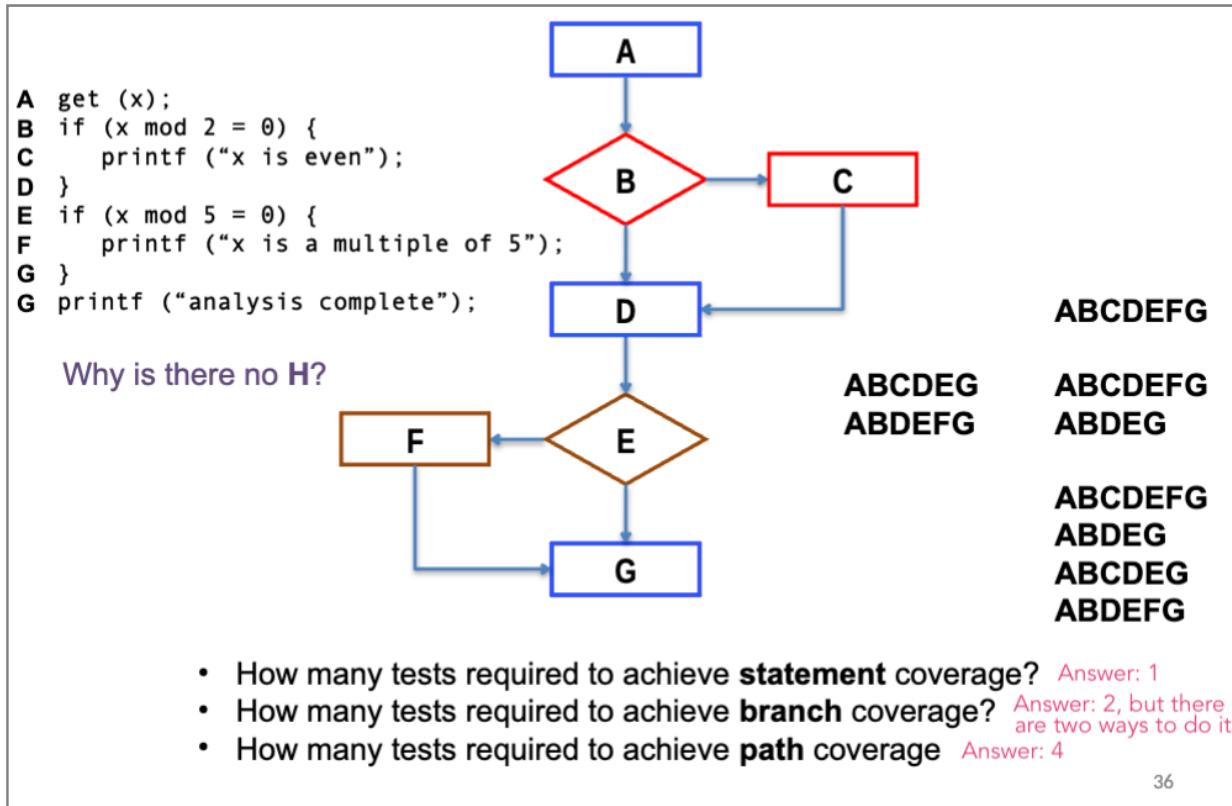


Figure 1: A snippet of code with its associated code graph, along with the answers to the three-part question found in Step 5 in this assignment. Your assignment should have a similar graph of your AI-generated code, along with a coverage analysis answering the questions shown here.