Assigned: October 27, 2009
Due: November 3, 2009

At the beginning of class on the due date, submit your neatly presented solution with
this page stapled to the front (50 pts). As with the previous assignments, you are to do
this assignment individually.

Part 1: Random Test-data Generation

   a. Rewrite the **tritype** program so that it contains only single conditions (no multiple
      conditions) but has the same functionality as the original program. Call this new
      program **tritype_single_conditions.**
   b. Write (implement) a **random_test_data_generator** that will generate sets of
      three integers (the inputs to **tritype_single_conditions**).
   c. Using your **random_test_data_generator,** generate 15 test cases for **tritype-
      _single_conditions.**   Present these test cases in the same format you used for
      Problem Set 6—e.g., if you have two test cases
            Test Case 1: isosceles 2 2 3 isosceles
            Test Case 2: equilateral 4 4 4 equilateral
      the file should contain
            "isosceles" 2 2 3 isosceles
            "equilateral" 4 4 4 equilateral
   d. Determine the branch coverage adequacy of the test suite you generated in (c).

For Part 1, submit
   • the listing (typed) of **tritype_single_conditions** (from a)
   • the listing of your **random_test_data_generator** (from b)
   • your test suite (typed) (from c)
   • branch coverage adequacy of your test suite, along with how you got it (from d).

Part 2: Goal-oriented or Path-oriented Test-data Generation

Select one of the test-data generation methods we discussed (symbolic execution,
concolic execution, search-based (genetic algorithms) generation)—call it
my_test_data_generation, and use it for the following:

   a. Select one of the uncovered branches in **tritype_single_conditions** that
      requires the traversal of at least three branches to reach it—this will be the
      target_branch for your test-data generation.
   b. Apply **my_test_data_generation** to generate test input to cover **target_branch.**
      Although there may be tools available for **my_test_data_generation,** you are
      not to use them. Instead do the work yourself, and show all steps, so that it will
      be easy for the reader to understand your approach.

For Part 2, submit
- The **target_branch** location in the program, along with the branches required to reach it.
- The description of how you used **my_test_data_generation** to generate test input for **target_branch**