Question 1 of 1

Method 1 1 pt

Suppose that the following test fails in the buggy version:

```
// IllegalArgumentException is expected to be thrown
@Test(expected = IllegalArgumentException.class)
public void testBadEndpoints() throws Exception {
    // f refers to the sin function
    UnivariateRealFunction f = new SinFunction();
    BrentSolver brentSolver = new BrentSolver();

    // Executing the following line should cause an exception.
    brentSolver.solve(/* function */ f, /* min */ 1, /* max */ 1.5, /* initial */ 1.2);
}
```

In the above test, the solve method (i.e., brentSolver.solve) takes the following 4 parameters:

1. f: a function to solve
2. min: the lower bound for the interval
3. max: the upper bound for the interval
4. initial: an initial value

The solve method finds the root of function f (in Korean: 함수 f의 해) between the interval [min, max], using the given initial value as the starting point.

Note that brentSolver.solve(f, min, max, initial) should throw IllegalArgumentException, when f(min), f(max), and f(initial) have the same sign (i.e., either all three values are positive or all three values are negative). Meanwhile, if f(min), f(max), and f(initial) do not have the same sign, IllegalArgumentException is not thrown.

In the above test, function f refers to the sin function, and sin(1) = 0.8414709848, sign(1.5) = 0.9974949866, and sin(1.2) = 0.93203908596. Since all three values are positive, IllegalArgumentException should be thrown. However, IllegalArgumentException is not thrown in the current buggy implementation, and the test fails.

Now suppose that we have a patch for this bug, and we are going to use the following test to validate this patch. Complete the test by filling in the two underlined blanks. **In your answer, separate the two boolean conditions with a comma (e.g., x > 0, x < 0 if two boolean conditions are x > 0 and x< 0).**

Note that given a function f of the UnivariateRealFunction type and an arbitrary input x to function f, f(x) can be computed through f.value(x).

```
public void testBadEndpoints(int min,int max, int initial) throws Exception {
    UnivariateRealFunction f = new SinFunction();
    BrentSolver solver = new BrentSolver();
    try {
       double root = solver.solve(f, min, max, initial);

        // Fill in the following blank with a boolean expression.
        // Note that the following condition will become false, if the patch is incorrect.
        assertTrue(_____);
    } catch (IllegalArgumentException e) {
       // Fill in the following blank with a boolean expression.
       // Note that the following condition will become false, if the patch is incorrect.
        assertTrue(_____);
    }
```