Question 1 of 1

Method 2 1 pt

Suppose that the following test fails in the buggy version:

```
public void testMath1021() {
    final int N = 43130568;
    final int m = 42976365;
    final int n = 50;
    double p = 0.8955874687296737;

    final HypergeometricDistribution dist = new HypergeometricDistribution(N, m, n);
    final int sample = dist.inverseCumulativeProbability(p);
    Assert.assertTrue(0 <= sample);
}
```

It turns out that the test fails when dist.getNumericalMean() returns a negative number. Otherwise (i.e., when dist.getNumericalMean() does not return a negative number), dist.inverseCumulativeProbability(p) returns a correct output.

Now suppose that we have a patch for this bug, and we are going to use the following test to validate this patch. Complete this test by filling in the underlined blank.

```
public void testMath1021(int sampleSize, int n, int popSize, double p) {
    // We make sure that sampleSize is not greater than popSize.
    assumeTrue(sampleSize <= popSize);
    // We make sure that p > 0 and p <= 1.
    assumeTrue(p > 0 && p <= 1);

    HypergeometricDistribution dist =
        new HypergeometricDistribution(popSize, sampleSize, n);
    int sample = dist.inverseCumulativeProbability(p);

     // Fill in the following blank with a boolean expression.
     // Note that under the given condition, the original version is assumed to run correctly,
     // and we expect that the same output will be produced from the patched version.
     assertSameOutIf(____________________,  /* output */ sample);
}
```