Problem 1:

Describe a test plan for the various implementations of calc. Given that the program cannot be tested on every possible input, what are three examples of tests that are implied by the spec but not checked by basic arithmetic.sh?

- ★ Test 1: I will be checking to see if the program outputs “BAD INPUT” and returns something other than zero. When I test for ***./calc a 4***.
- ★ Test 2: Second thing that I will be checking for is that the program outputs “NOT ENOUGHT INPUT” and returns non-zero using the test ***./calc 4***.
- ★ Test 3: In the test I will be checking for the program to output “TOO BIG” using the test ***./calc 512 512***.

Problem 2:

You may have noticed that the spec doesn’t say anything about what to do if the user supplies too many arguments at the command line. Unfortunately, ambiguity in specifications is very common in practice and can be very confusing for programmers. If you were asked to implement calc given this spec, you would be forced to choose from among at least three interpretations of what calc should do when called with, say, three arguments 3, 4, and 5:

- Calc should sum *all* of the arguments, and print 12.
- Calc should ignore the third argument, and print the sum of the first two: 7.
- Calc should print an error message and return non-zero.

Luckily for you, you are not the programmer (for now). Should your test scripts check what the program does when given more than two arguments? Why or why not?

- Yes I believe that the test script should check for what happens when more than two arugments are given because it is possible that the programmer does one of the three interpretations. Could be possible that cats programs have one of the three when looking at all of them.