Problem 1:

- Test bounds by ensuring a non 0 exit code and “TOO BIG” is printed when running
    - ./calc 513 2
    - ./calc 2 513
    - ./calc 513 513
    - ./calc -513 2
    - ./calc 2 -513
    - ./calc -513 -513
- Test valid inputs, making sure an 0 exit code is returned and the correct output is printed
    - ./calc 512 2
        - 514
    - ./calc 2 512
        - 514
    - ./calc 512 512
        - 1024
    - ./calc -512 -512
        - -1024
    - ./calc 512 -512
        - 0
    - ./calc 0 0
        - 0
    - ./calc +10 +10
        - 20
- Test non integer inputs, ensuring a non 0 exit code and “BAD INPUT” is printed
    - ./calc A 1
    - ./calc 1 A
    - ./calc BB BB
    - ./calc 5.5 3
    - ./calc 3 5.5
    - ./calc 5.5 5.5
- Test when fewer than 2 arguments are provided, ensuring a non 0 exit code and “NOT ENOUGH INPUT” is printed:
    - ./calc 1
    - ./calc 0
- The tests will all check that the output has no leading 0s, trailing spaces or 0s and that each line ends in a newline

Problem 2:

- Since I am not the programmer and the spec didn't specify whether more than 2 inputs are supported, I would not write tests to check what the program does when presented with more than 2 inputs since so many implementations were written by so many different programmers, some might have written their calculators with more than 2 inputs in mind whereas others might have written with only 2 in mind, and therefore I can’t assume either implementation is right or wrong so I won't test for it.