Mira Saini
Cruz ID: misaini
ID #: 2075469

Problem 1:

Test Filename	Arguments	Expected output	Expected exit code
test_dec_arg	3.14, 4	BAD INPUT	non-zero
test_one_arg	4	NOT ENOUGH INPUT	non-zero
test_512_arg	512 512	1024	0
test_n512arg	-512 -512	-1024	0
test_non_arg	(no arguments)	NOT ENOUGH INPUT	non-zero
test_nbigarg	-513 -513	TOO BIG	non-zero
test_big_arg	513 512	TOO BIG	non-zero
test_511_arg	511 511	1022	0
test_2bigarg	513 513	TOO BIG	non-zero
test_n511arg	-511 -511	-1022	0
test_1bigarg	-512 -513	TOO BIG	non-zero
test_ran_arg	! $	BAD INPUT	non-zero
test_mix_arg	511 512	1023	0
test_2ranarg	e w	BAD INPUT	non-zero
test_nmixarg	-511 -512	-1023	0
test_-511512	-511 512	1	0
test_-512511	-512 511	-1	0
test_-512512	-512 512	0	0
test_-512513	-512 513	TOO BIG	non-zero
test_-513512	-513 512	TOO BIG	non-zero

test_-513513	-513 513	TOO BIG	non-zero
test_-511511	-511 511	0	0
test_514_512	514 512	TOO BIG	non-zero
test_-514512	-514 512	TOO BIG	non-zero

Problem 2: Section 5.2 mentions that the specification doesn't say what to do if a "calculator" program is run with more than two arguments. Take a couple paragraphs to discuss the consequences of writing a test to check what the program does when given three arguments. Support whether your test plan should or should not include such a test.

As only the person writing the tests for a program that I did not myself build, I do not know what the developers intended for the program, except for whatever the specification listed. A few consequences would arise if I were to create a test script for a part of the program that the specification did not specifically outline because I would be making assumptions as to what the program should do in that circumstance.

For this situation, if I were to write a test to check if the program correctly handles more than two inputs, I wouldn't know what would be assumed as the correct interpretation versus the incorrect interpretation. I could assume that the program would add them all up normally and check to make sure that the program does, but if that is an incorrect assumption where the program should instead be printing out an error message, then I've not contributed to the improvement of the program. If I don't know what the program is supposed to do, then I cannot correctly write scripts to test to make sure that it does the right computation. My assumption/interpretation could be very incorrect and mark correct interpretations as wrong.

On the other hand, if I don't test this specific situation, I could potentially leave bugs when the user enters three or more tiny integer arguments as the program could mistakenly have more than one interpretation. Though, since the specification does not say to test for this circumstance, there is too much ambiguity for any sort of test to improve the program as I would be unsure of what the program should do in that situation. I, as someone who is not the developer, do not know the intentions for the program so I shouldn't write tests that assume facts if they are not specified in the specification. Therefore, for reasons of too much ambiguity, I will not be implementing this test into my test plan for this assignment.