Name	Description	Use Case	Inputs	Expected Outputs	Fail or Success
test_read_success	Tests if the function reads input into a desired register	The user wants to input data into a certain register	,+010000	Register 0 will have the contents of ,+010000	If the mentioned register has the respective value then it is a success
test_write_success	Tests if the program can successfully write a value to the screen	The user wants a certain value read to the screen	,+010000	The value, +010000 is read out to the console	If the value is read to the console then it was a success
test_load_success	Test if function can load a word from a register to the accumulator	The user wants to load a value into the accumulator to preform an operation	,+010000	The value , +010000 is held in the accumulator	If the value is in the accumulator then it was a success
test_store_success	Tests if the function can store words from the accumulator to a location in memory	The user wants to move a value in the accumulator into memory to use it later	,+010000	Register 0 will have the contents of ,+010000	If the mentioned register has the respective value then it is a success
test_add_success	Tests the success case for the add function	The user wants to add two numbers together	,+001234, +004321	Accumulator will hold the value , +005555	If the accumulator holds the value aforementioned then it was a success
test_add_overflow	Tests the overflow case for the add function	The user wants to add two numbers together, but the sum is too large for the amount of storage available in the system	,+500000, +500000, -500000, -500000	Add function will return False	If False is returned then it was a success

Name	Description	Use Case	Inputs	Expected Outputs	Fail or Success
test_add_zero	Tests if the add fucntion can properly handle a zero sum	The user inputs values that add up to zero	, -100000, +100000	Accumulator will hold the value , +000000	If the accumulator holds the value aforementioned then it was a success
test_add_negative	Tests the negative case for the add function	The user inputs two negative numbers to add together	, -100000, -100000	Accumulator will hold the value , -200000	If the accumulator holds the value aforementioned then it was a success
test_subtract_success	Tests the success case for the subtract function	The user wants to subtract two numbers together	,+001234, +004321	Accumulator will hold the value , +003087	If the accumulator holds the value aforementioned then it was a success
test_subtract_overflow	Tests the overflow case for the subtract function	The user wants to subtract two numbers together, but the difference is too large for the amount of storage available in the system	,-500000, +500000, +500000, -500000	Subtract function will return False	If False is returned then it was a success
test_subtract_zero	Tests if the subtraction function can handle a zero difference	The user wants to subtract two numbers together that end in a zero difference	, +100000, +100000	Accumulator will hold the value , +000000	If the accumulator holds the value aforementioned then it was a success
test_subtract_negative	Tests the negative subtract case	The user inputs two numbers that end up having a negative difference	, +100000, -100000	Accumulator will hold the value , -200000	If the accumulator holds the value aforementioned then it was a success

Name	Description	Use Case	Inputs	Expected Outputs	Fail or Success
test_divide_success	Tests the success case for the division function	The user inputs two numbers to be divided	,+000002, +200000	Accumulator will hold the value , +100000	If the accumulator holds the value aforementioned then it was a success
test_divide_zero	Tests the zero quotient case for the divide function	The user inputs two numbers where the quotient is zero	, +00001, +000000	Accumulator will hold the value , +000000	If the accumulator holds the value aforementioned then it was a success
test_divide_negative	Tests the negative quotient case for the divide function	The user inputs two numbers where the quotient is negative	, +000001, -050000	Accumulator will hold the value , -050000	If the accumulator holds the value aforementioned then it was a success
test_multiply_success	Tests the success case for the multiply function	The user inputs two numbers to be multiplied	, +000002, +200000	Accumulator will hold the value , +400000	If the accumulator holds the value aforementioned then it was a success
test_multiply overflow	Tests the overflow case for the multiply function	The user inputs two numbers where the product is too large for the memory of the system	, +000002, +500000, +000002, -500000	Multiply function will return False	If False is returned then it was a success
test_multiply_zero	Tests the zero product case for the multiply function	The user inputs two numbers where the product is zero	, +100000, +000000	Accumulator will hold the value , +000000	If the accumulator holds the value aforementioned then it was a success

Name	Description	Use Case	Inputs	Expected Outputs	Fail or Success
test_multiply_negative	Tests the negative product case for the multiply function	The user inputs two numbers where the product is negative	, +000002, -100000	Accumulator will hold the value , -200000	If the accumulator holds the value aforementioned then it was a success
test_branch_success	Tests if the system can branch to a different register address	The user wants to branch to a different place in the program	090, the desired location in memory	Current address will return 89, once the counter is incremented then it will be at 90	If the assert returns true then it was a success
test_branch_neg_move	Tests if the system will branch when the user tests for a negative number in the accumulator	The user wants to branch to a different place in the program if the number in the accumulator is negative	, -100000, 090 - desired address location	Current address will return 89, once the counter is incremented then it will be at 90	If the assert returns true then it was a success
test_branch_neg_stay	Tests if the system won't branch when the user tests for a negative number in the accumulator and there isn't one	The user tests if the value in the accumulator is negative and wants to stay in the same location if it isn't	, +1000, 090- desired address location	Current address will return 0	If the assert returns true then it was a success
test_branch_zero_move	Test if the system will branch of the value in the accumulator is zero	The user wants to branch to a different place in the program if the number in the accumulator is zero	090, the desired location in memory	Current address will return 89, once the counter is incremented then it will be at 90	If the assert returns true then it was a success
test_branch_zero_stay	Tests if the system won't branch when the user tests for zero in the accumulator and there isn't one	The user tests if the value in the accumulator is zero and wants to stay in the same location if it isn't	, +010000, 090 - desired address location	Current address will return 0	If the assert returns true then it was a success

Name	Description	Use Case	Inputs	Expected Outputs	Fail or Success
test_halt_success	Tests if the function returns false to terminate the program	The user wants to halt the program	Halt instruction called	False	If the assert returns true then it was a success