

Testing Strategy

I used test driven development to design and implement my program, this means that every piece of code written was in response to or alongside a test that was written.

The results of all the automated tests are written to a text file each time the test command is run. I have included a copy of this file at the end of this report, along with the images produced when each of the full test files are run.

Because I wrote the code with testing in mind, the vast majority of the functions written in my parser/interpreter return an integer and take a pointer to the program structure as an argument. This enabled me to write my tester function which is passed a pointer to the function being tested and a test condition(usually TRUE or FALSE) it then prints the results of these tests to the test_results file.

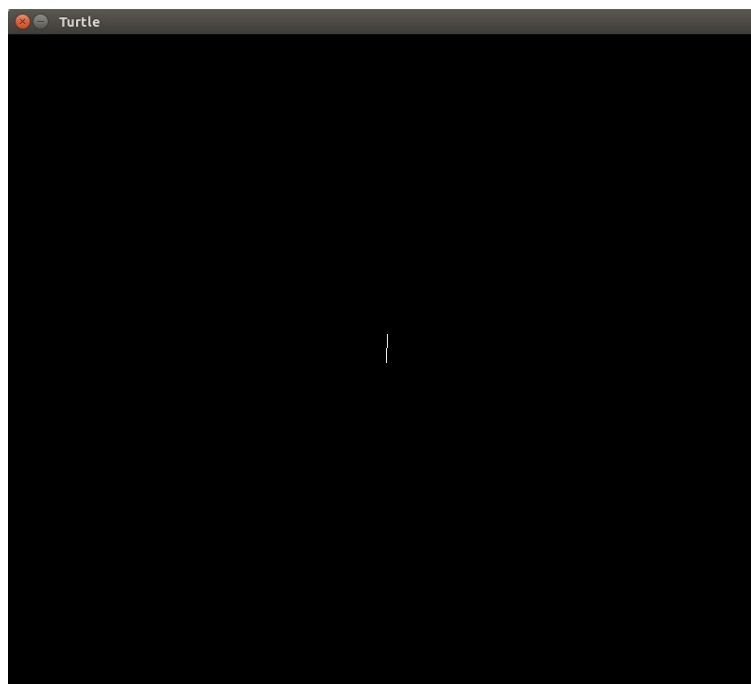
Because I had a large number of test function to run, I created an array of file pointers(pointing to each of my test functions) and an array of corresponding test names. I wrote another function which took a pointer to a test function and a test name as arguments. I then looped through each test function in tern with my run_test function calling and printing results from each of the tests.

Some of my test functions are bigger than I would have liked and given more time I would have looked to write more functions that could have cut down on this. My plan was to demonstrate in the two functions I did write I was able to demonstrate a bullet proof testing strategy.

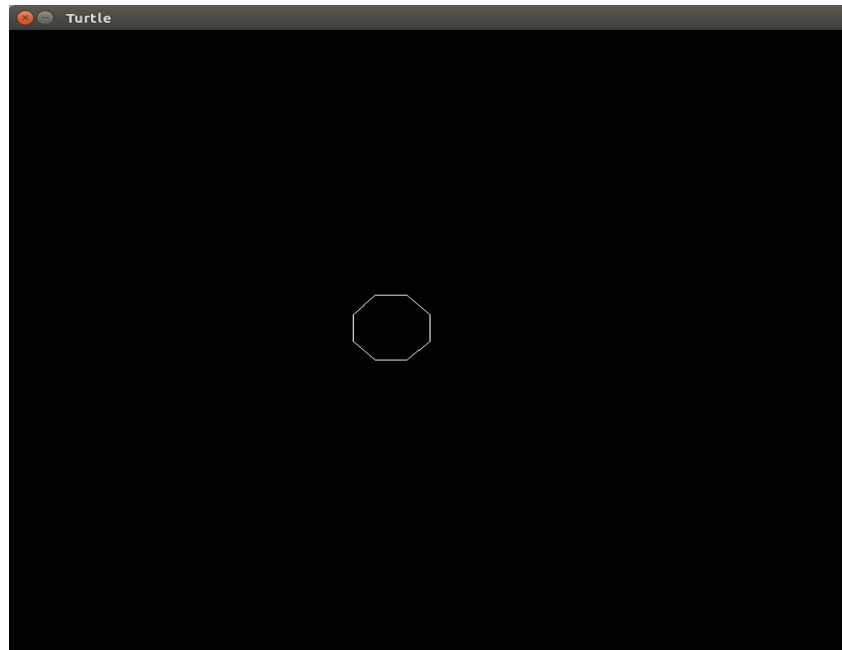
The black box tests I implemented sent a full user test file into the parse function. The purpose of these files was to show all aspects of the language(to some degree). Might be good to know at this point that I am not very creative, I am confident that somebody more gifted in this area could produce a beautiful image, mine simply show the output of arbitrary statements that include certain functionality.

The code I produced for testing is contained in the test.c and test.h files.

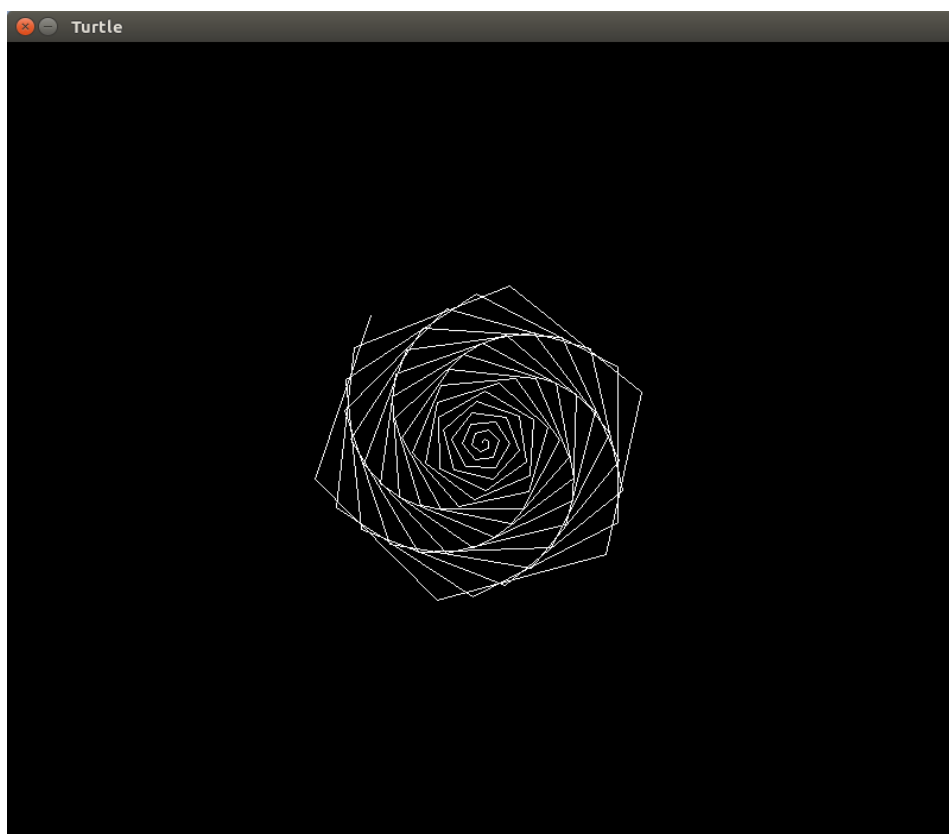
TEST1



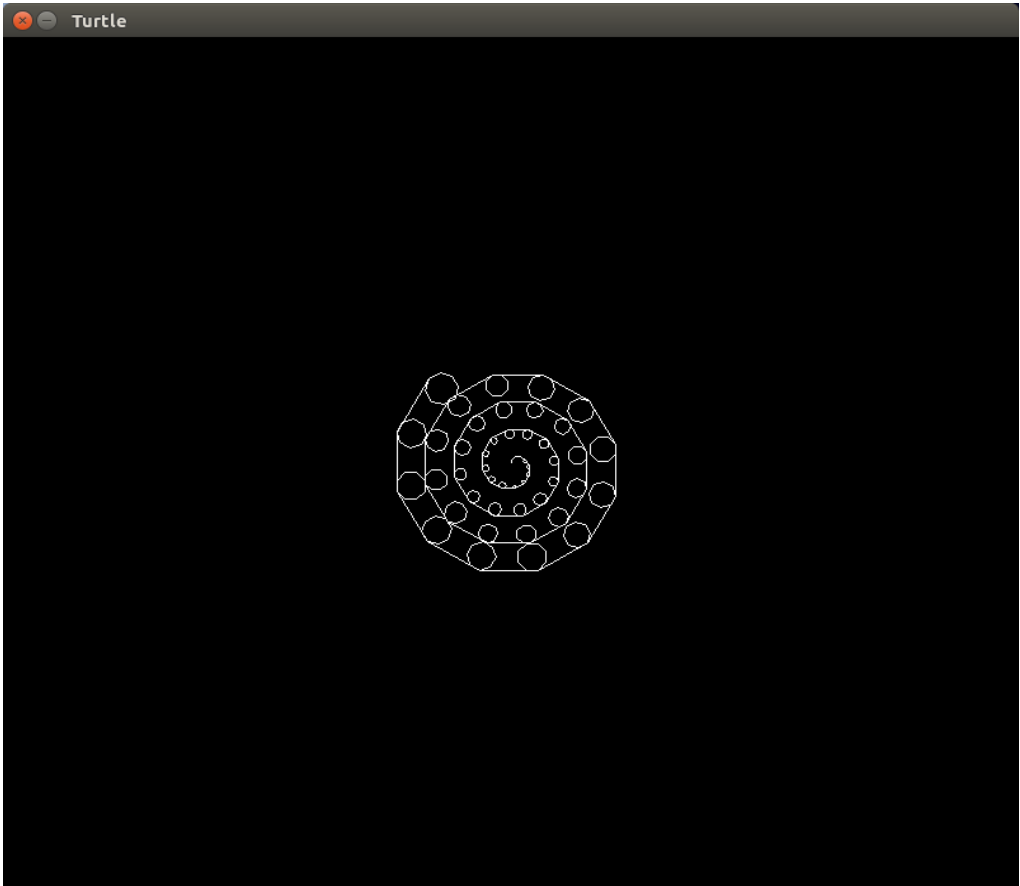
TEST 2



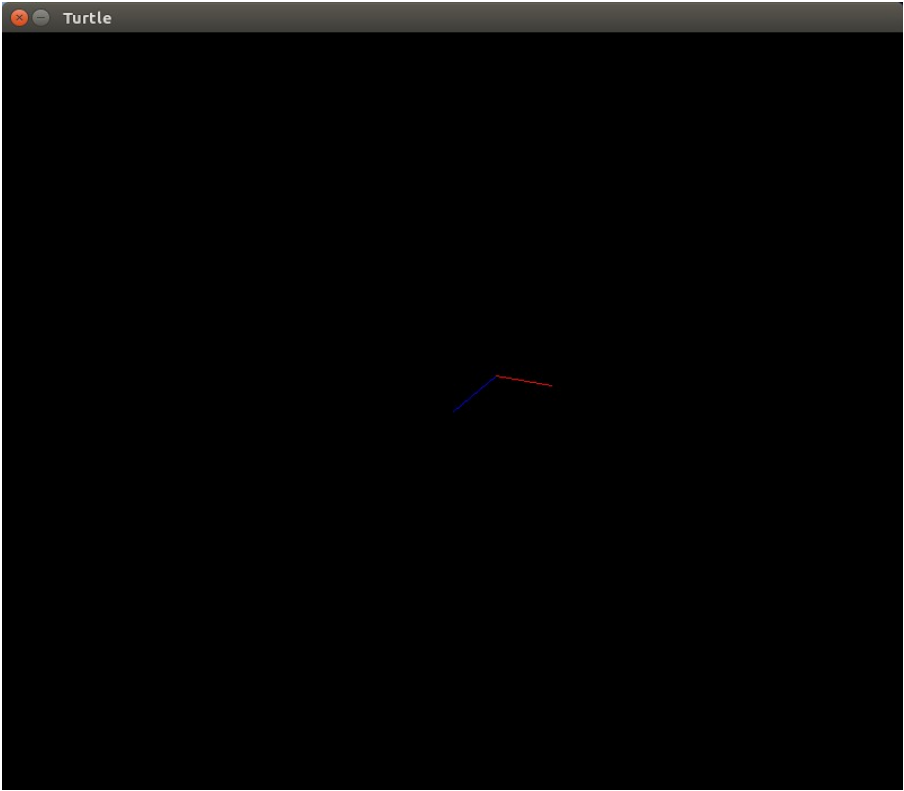
TEST 3



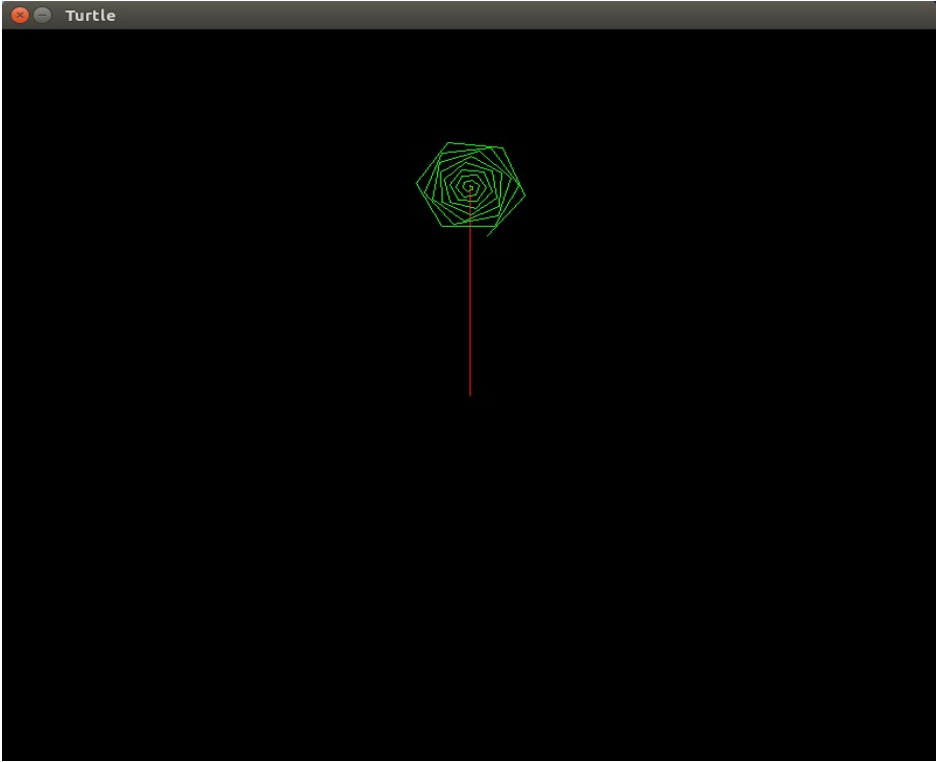
TEST 4



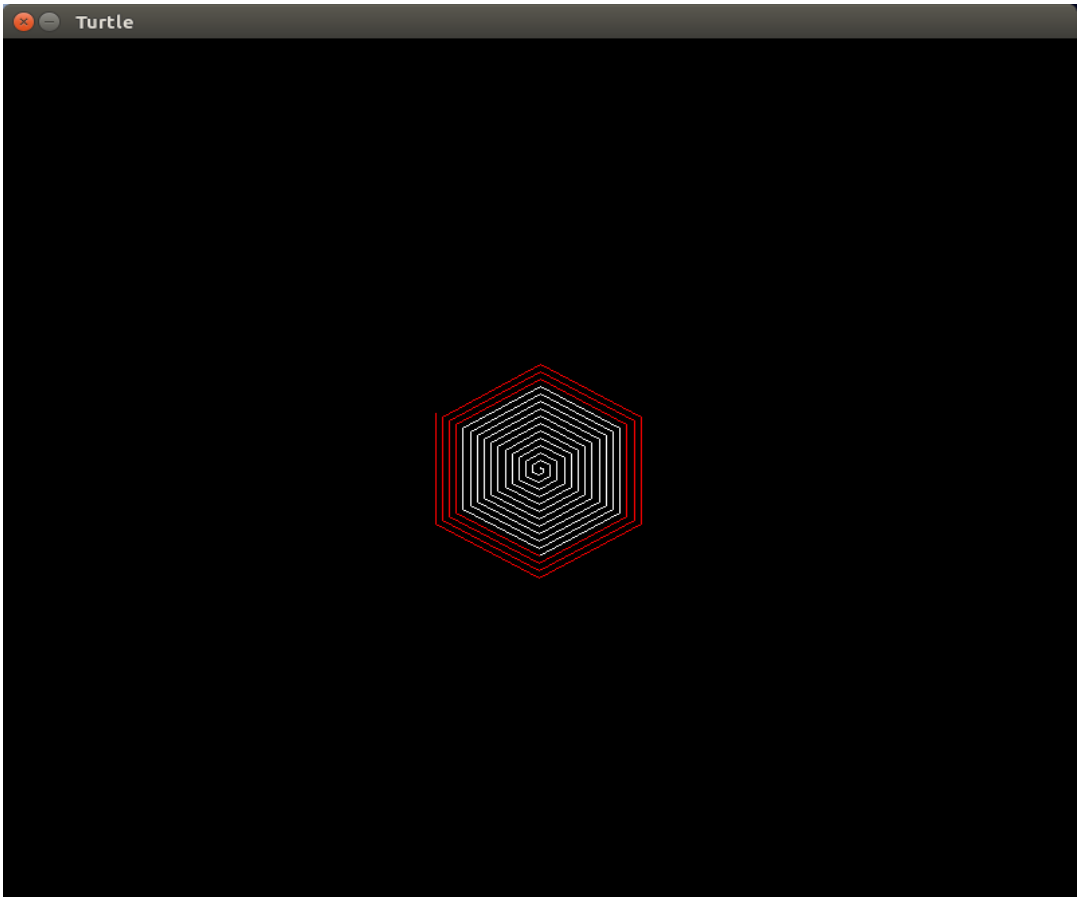
TEST 5



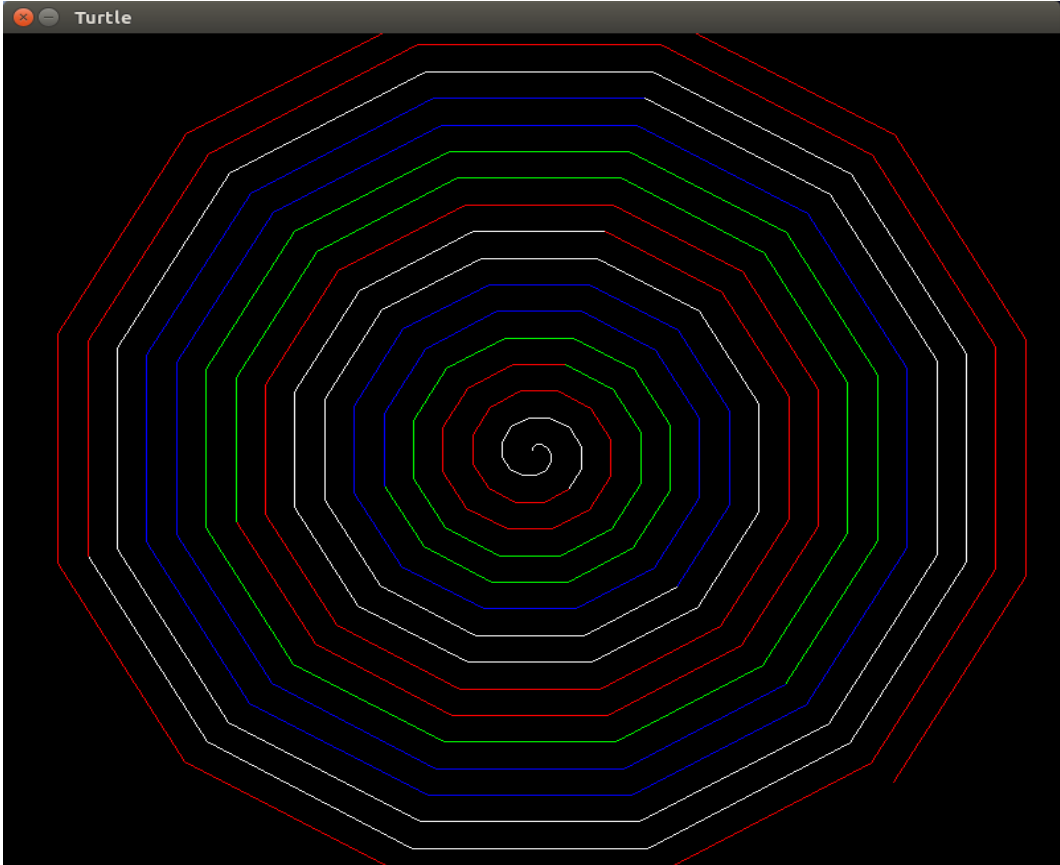
TEST 6



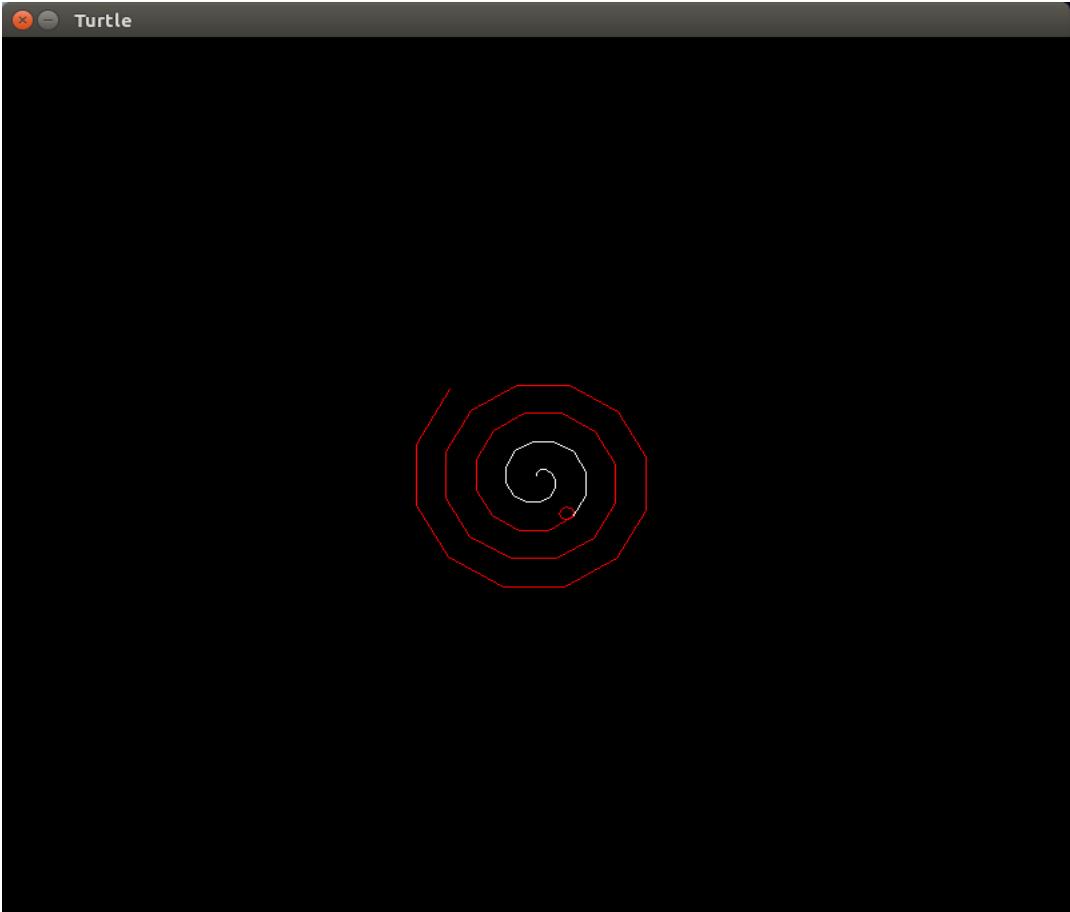
TEST 7



TEST 8



TEST 9



Below are the test results for each function:

parse test 1: Passed
parse test 2: Passed
parse test 3: Passed
parse test 4: Passed
parse test 5: Passed
parse test 6: Passed
parse test 7: Passed
parse test 8: Passed

Test parse function PASSED!

Input test 1: Passed
Input test 2: Passed
Input test 3: Passed
Input test 4: Passed
Input test 5: Passed

Test check input function PASSED!

Test initialise program function PASSED!

Validate test 1: Passed
Validate test 2: Passed

Test validate function PASSED!

Instrctlst test 1: Passed
Instrctlst test 2: Passed
Instrctlst test 3: Passed

Test instrctlst function PASSED!

Instruction test 1: Passed
Instruction test 2: Passed
Instruction test 3: Passed
Instruction test 4: Passed

Test instruction function PASSED!

Fd test 1: Passed
Fd test 2: Passed
Fd test 3: Passed
Fd test 4: Passed

Fd test 5: Passed

Fd test 6: Passed

Test fd function PASSED!

Lt test 1: Passed
Lt test 2: Passed
Lt test 3: Passed
Lt test 4: Passed
Lt test 5: Passed
Lt test 6: Passed

Test lt function PASSED!

Rt test 1: Passed
Rt test 2: Passed
Rt test 3: Passed
Rt test 4: Passed
Rt test 5: Passed
Rt test 6: Passed

Test rt function PASSED!

make_positive test 1: Passed
make_positive test 2: Passed

Test make positive function PASSED!

Varnum test 1: Passed
Varnum test 2: Passed
Varnum test 3: Passed
Varnum test 4: Passed

Test varnum function PASSED!

Push test 1: Passed

Test push function PASSED!

Get_parameter test 1: Passed
Get_parameter test 2: Passed
Get_parameter test 3: Passed

Test get parameter function PASSED!

is_number test 1: Passed
is_number test 2: Passed
is_number test 3: Passed
is_number test 4: Passed
is_number test 5: Passed
is_number test 6: Passed
is_number test 7: Passed

****Test is number function PASSED!****

is_var test 1: Passed
is_var test 2: Passed
is_var test 3: Passed
is_var test 4: Passed

****Test is var function PASSED!****

Check_stack test 1: Passed
Check_stack test 2: Passed

****Test check stack function PASSED!****

Add test 1: Passed
Add test 2: Passed

****Test add function PASSED!****

Subtract test 1: Passed
Subtract test 2: Passed

****Test subtract function PASSED!****

Divide test 1: Passed
Divide test 2: Passed

****Test divide function PASSED!****

Multiply test 1: Passed
Multiply test 2: Passed

****Test multiply function PASSED!****

op test 1: Passed
op test 2: Passed
op test 3: Passed

op test 4: Passed
op test 5: Passed

****Test op function PASSED!****

pop test 1: Passed
pop test 2: Passed

****Test pop function PASSED!****

polish test 1: Passed
polish test 2: Passed
polish test 3: Passed
polish test 4: Passed
polish test 5: Passed

****Test polish function PASSED!****

set test 1: Passed
set test 2: Passed
set test 3: Passed
set test 4: Passed
set test 5: Passed
set test 6: Passed
set test 7: Passed

****Test set function PASSED!****

loop_condition test 1: Passed
loop_condition test 2: Passed
loop_condition test 3: Passed
loop_condition test 4: Passed
loop_condition test 5: Passed

****Test loop condition function PASSED!****

perform_loop test 1: Passed
perform_loop test 2: Passed
perform_loop test 3: Passed

****Test perform loop function PASSED!****

loop test 1: Passed
loop test 2: Passed
loop test 3: Passed

****Test loop function PASSED!****

Set_nex_xy test 1: Passed
Set_nex_xy test 2: Passed
Set_nex_xy test 3: Passed
Set_nex_xy test 4: Passed

****Test set new xy function PASSED!****

If_condition test 1: Passed
If_condition test 2: Passed
If_condition test 3: Passed
If_condition test 4: Passed
If_condition test 5: Passed

****Test if condition function PASSED!****

If_letter test 1: Passed
If_letter test 2: Passed
If_letter test 3: Passed

****Test if letter function PASSED!****

If_colour test 1: Passed
If_colour test 2: Passed

****Test if colour function PASSED!****

Assign_colour test 1: Passed
Assign_colour test 2: Passed
Assign_colour test 3: Passed
Assign_colour test 4: Passed

****Test assign colour function PASSED!****

Set_colour test 1: Passed
Set_colour test 2: Passed
Set_colour test 3: Passed
Set_colour test 4: Passed
Set_colour test 5: Passed

****Test set colour function PASSED!****

Set_letter test 1: Passed
Set_letter test 2: Passed
Set_letter test 3: Passed
Set_letter test 4: Passed

****Test set letter function PASSED!****

NUMBER OF TESTS PASSED: 34

NUMBER OF TESTS FAILED: 0