

 Fantastic-Four_deliverable4.md

Chapter 4

Executive Summary

We have completed the task of creating 20 additional test cases in addition to the test cases from the last deliverable. The new test cases are the following `testCase6.txt-testCase25.txt` . All the test cases test the following methods:

1. `add(x, y)` : addition of two numbers
2. `sub(x, y)` : subtraction of two numbers
3. `mul(x, y)` : multiplication of two numbers
4. `div(x, y)` : division of two numbers
5. `pow(x, y)` : the exponentiation of two numbers

Test Cases:

Test ID	Requirement	Component	Method	Test Inputs	Expected Outcomes	Text file
1	Addition of Two Numbers	<code>functions.py</code>	<code>add(x, y)</code>	(1,2)	No errors expected with natural numbers, result should be returned as 3	Test Case 1
2	Addition of Two Numbers	<code>functions.py</code>	<code>add(x, y)</code>	(-1,1)	No errors expected with integers, result should be returned as 0	Test Case 2
3	Addition of Two Numbers	<code>functions.py</code>	<code>add(x, y)</code>	(1.1,2.1)	No errors expected with rational numbers due to casting with the <code>_d</code> function, result should be returned as 3.2	Test Case 3
4	Addition of Two Numbers	<code>functions.py</code>	<code>add(x, y)</code>	(<code>pi</code> , <code>sqrt(2)</code>)	No errors should occur with irrational numbers, the result should be returned as <code>math.pi + math.sqrt(2)</code>	Test Case 4
5	Addition of Two Numbers	<code>functions.py</code>	<code>add(x, y)</code>	(<code>sys.maxsize</code> ,2)	Note: the included <code>sys</code> library is needed to get the maximum integer No errors occur when invoking the maximum size integer added with 2, the result will vary based on the system architecture and will add 2 with no overflow error	Test Case 5
6	Subtraction of Two Numbers	<code>functions.py</code>	<code>sub(x, y)</code>	(1,2)	No errors expected with natural numbers, result should be returned as -1	Test Case 6
7	Subtraction of Two Numbers	<code>functions.py</code>	<code>sub(x, y)</code>	(-1,1)	No errors expected with integers, result should be returned as -2	Test Case 7
8	Addition of Two Numbers	<code>functions.py</code>	<code>sub(x, y)</code>	(1.1,2.1)	No errors expected with rational numbers due to casting with the <code>_d</code> function, result should be returned as -1	Test Case 8

Test ID	Requirement	Component	Method	Test Inputs	Expected Outcomes	Text file
9	Subtraction of Two Numbers	functions.py	sub(x, y)	(pi,sqrt(2))	No errors should occur with irrational numbers, the result should be returned as $\text{math.pi} - \text{math.sqrt}(2)$	Test Case 9
10	Subtraction of Two Numbers	functions.py	sub(x, y)	(sys.maxsize,2 * (sys.maxsize + 1))	Note: the included sys library is needed to get the maximum integer No errors occur when invoking the maximum size integer subtracted by 2, the result will vary based on the system architecture and will subtract 2 with no overflow error	Test Case 10
11	Multiplication of Two Numbers	functions.py	mul(x, y)	(1,2)	No errors expected with natural numbers, result should be returned as 2	Test Case 11
12	Multiplication of Two Numbers	functions.py	mul(x, y)	(-1,1)	No errors expected with integers, result should be returned as -1	Test Case 12
13	Multiplication of Two Numbers	functions.py	mul(x, y)	(1.1,2.1)	No errors expected with rational numbers due to casting with the <code>_d</code> function, result should be returned as 2.3100000000000005	Test Case 13
14	Multiplication of Two Numbers	functions.py	mul(x, y)	(pi,sqrt(2))	No errors should occur with irrational numbers, the result should be returned as $\text{math.pi} * \text{math.sqrt}(2)$	Test Case 14
15	Multiplication of Two Numbers	functions.py	mul(x, y)	(sys.maxsize,2)	Note: the included sys library is needed to get the maximum integer No errors occur when invoking the maximum size integer multiplied with 2, the result will vary based on the system architecture and will multiply 2 with no overflow error	Test Case 15
16	Multiplication of Two Numbers	functions.py	div(x, y)	(1,2)	No errors expected with natural numbers, result should be returned as 0.5	Test Case 16
17	Division of Two Numbers	functions.py	div(x, y)	(-1,1)	No errors expected with integers, result should be returned as -1.0	Test Case 17
18	Division of Two Numbers	functions.py	div(x, y)	(1.1,2.1)	No errors expected with rational numbers due to casting with the <code>_d</code> function, result should be returned as 0.5238095238095238	Test Case 18
19	Division of Two Numbers	functions.py	div(x, y)	(pi,sqrt(2))	No errors should occur with irrational numbers, the result should be returned as $\text{math.pi} / \text{math.sqrt}(2)$	Test Case 14
20	Division of Two Numbers	functions.py	div(x, y)	(sys.maxsize,2)	Note: the included sys library is needed to get the maximum integer No errors occur when invoking the maximum size integer divided by 2, the result will vary	Test Case 20

Test ID	Requirement	Component	Method	Test Inputs	Expected Outcomes	Text file
					based on the system architecture and will divide 2 with no overflow error	
21	Multiplication of Two Numbers	functions.py	pow(x, y)	(1,2)	No errors expected with natural numbers, result should be returned as 2	Test Case 21
22	Multiplication of Two Numbers	functions.py	pow(x, y)	(-1,1)	No errors expected with integers, result should be returned as -1	Test Case 22
23	Multiplication of Two Numbers	functions.py	pow(x, y)	(1.1,2.1)	No errors expected with rational numbers due to casting with the <code>_d</code> function, result should be returned as 1.2215876651600335	Test Case 23
24	Multiplication of Two Numbers	functions.py	pow(x, y)	(pi,sqrt(2))	No errors should occur with irrational numbers, the result should be returned as <code>math.pi ** math.sqrt(2)</code>	Test Case 24
25	Exponentiation of Two Numbers	functions.py	pow(x, y)	(sys.maxsize,2)	Note: the included <code>sys</code> library is needed to get the maximum integer No errors occur when invoking the maximum size integer raised with 2, the result will vary based on the system architecture and will raise <code>sys.maxsize</code> to the power of 2 with no overflow error	Test Case 25

Failed Test Cases:

Test Case ID	Expected Oracle	Actual Oracle	Reason for Failure
16	0.5	1/2	Different output generated by the <code>div(x, y)</code> method
17	-1.0	-1	Precision is different for <code>div(x, y)</code> method
23	1.2215876651600335	1.221587665160033476	Rounding for the <code>pow(x, y)</code> method
24	5.047497267370911	5.047497267370911089	Rounding for the <code>pow(x, y)</code> method