Project Name:	LED globe	Test Designed by:	Jonathan
Module Name:	runDC.py	Tests last executed:	

Using a stub MotoStub.py to provide inputs (motorInputs class) and allow output (MotorStub class)

Test ID	Test Name	Test Description	Test Cases (inputs)	Test Steps	Expected Results	Acutal Results	Status (pass/fail)
1	test_userRPStoDutyNeg()	Input of a negative value	RPS = -50	Input RPS to userRPStoDuty and compare the result with expected duty	return 0		
2	test_userRPStoDutyZero()	Input of a value 0	RPS = 0	Input RPS to userRPStoDuty and compare the result with expected duty	return 0		
3	test_userRPStoDutyNorm()	Input of a value within range	RPS = 50	Input RPS to userRPStoDuty and compare the result with expected duty	return 35.46		
4	test_userRPStoDutyMax()	Input of a max value	RPS = 141	Input RPS to userRPStoDuty and compare the result with expected duty	return 100		
5	test_userRPStoDutyLarge()	Input of value larger than max	RPS = 200	Input RPS to userRPStoDuty and compare the result with expected duty	return 100		
6	test_userRPStoDutyInvalid()	Input a input type like char	RPS = aaa	Input RPS to userRPStoDuty and compare the result with expected duty	exception		
7	test_applyDutyNeg()	In motorInput class, set the input value to a negative value.	setInput(-50)	Call function applyDuty() so it enters its loop of reading inputs and sending it to the stub. Then call setInput(input) in motorInput class.	stubPWM print True. (comparison of duty passed and expected duty from RPS input)		
8	test_applyDutyNegOne()	In motorInput class, set the input value to -1, break condition for applyDuty loop.	setInput(-1)	Call function applyDuty() so it enters its loop of reading inputs and sending it to the stub. Then call setInput(input) in motorInput class.	stubPWM print True		
9	test_applyDutyZero()	In motorInput class, set the input value to 0.	setInput(0)	Call function applyDuty() so it enters its loop of reading inputs and sending it to the stub. Then call setInput(input) in motorInput class.	stubPWM print True		

10	test_applyDutyNorm()	In motorInput class, set the input value to a normal value.	setInput(75)	Call function applyDuty() so it enters its loop of reading inputs and sending it to the stub. Then call setInput(input) in motorInput class.	stubPWM print True		
11	test_applyDutyMax()	In motorInput class, set the input value to max value.	setInput(141)	Call function applyDuty() so it enters its loop of reading inputs and sending it to the stub. Then call setInput(input) in motorInput class.	stubPWM print True		
12	test_applyDutyLarge()	In motorInput class, set the input value to a value larger than max.	setInput(200)	Call function applyDuty() so it enters its loop of reading inputs and sending it to the stub. Then call setInput(input) in motorInput class.	stubPWM print True		