Assessent

App	Scenarios	Examples
Integer Devider		
	Validate app returns correct value when entering 2 integer values	./integer_devider 2 7
	Validate app returns a value with in format of x.xxxx (4 decimals)	
	Validate app returns a correct value when entering 2 very large integer values (32 bits allow up to 2147483647)	./integer_devider 2147483647 13
	Validate app returns a correct value when entering 2 very large negative integer values (32 bits allow from -2147483648)	./integer_devider -2147483648 13
	Validate the calculation is rounding up (i.e if result is 0.28576 it will show 0.2858)	
	Validate app returns an error when user enters 1 value only	./integer_devider 2
	Validate app returns an error when user enters no values	./integer_devider
	Validate app returns an error when users enters non integer values (i.e. string, decimals)	./integer_devider test me
	Validate app returns an error when user enters more than 2 values (or ignores the extra arguments)	./integer_devider 2 7 1345 test
	Validate app returns error when user enters a 64 bit Integer	./integer_devider 2147483649 13
Letter replacer		
	Validate app returns correct value when entering string, value to replace, value to replace with	./letter_replacer "MaMa mama" M T
	Validate app return value is a string	
	Validate the app returns an error when user doesn't enter all required arguments	./letter_replacer "Test" T ./letter_replacer ./letter_replacer "Test"
	Validate app returns an error when user enters a non-string value for string argument(i.e 2 M T)	./letter_replacer 2 T M
	Validate app returns an error when any of the character arguments are not ASCII characters (i.e. :谷新道)	./letter_replacer "Test Test" T :谷新道 ./letter_replacer "Test Test" :谷新道 T
	Validate app returns an error when user enters more than the required arguments (or ignores the extra arguments)	./letter_replacer "MaMa mama" M T
Reserve Table		
	Validate the app returns RESERVED! when user enters arguments that finds an empty table(i.e. restaurant is opened, tables are available for the number of people, reserved time is between min and max)	./reserve_table USER_10 2021-12-1 2021-12T15:25 5
	Validate the app returns NO TABLES AVAILABLE! when user enters data and table matching the user count is busy for specidied date/time	./reserve_table USER_10 2021-12-1 2021-12-12T15:25 6 ./reserve_table USER_10 2021-12-12 2021-12-12T15:25 3 (presuming the is already reserved)
	Validate app returns RESERVER! if the 2 people table are booked but the 3 people table is available (presuming this is a requirement) when booking for 2 people (also available other tests like give 5 people table to 3 people if 3 people table is busy)	./reserve_table USER_10 2021-12-12 2021-12-12T15:25 2
	Validate app returns No TABLES AVAILABLE! if user tries to book a table for more than 5 people	./reserve_table USER_10 2021-12-12 2021-12-12T15:25 7
	Validate app returns ERROR ENCOUNTERED! if user tries to book a table in the past	./reserve_table USER_10 2020-12-12 2021-12-12T15:25 2
	Validate app returns ERROR ENCOUNTERED! if user tries to book a table outside working hours	./reserve_table USER_10 2021-12-12 2021-12-12T12:25 2 ./reserve_table USER_10 2021-12-12 12T01:25 2
	Validate app returns ERROR ENCOUNTERED! if user tries to book a table for less than minimum time or more than maximum time	./reserve_table USER_10 2020-12-12 2021-12-12T14:25 2 ./reserve_table USER_10 2020-12-12 2021-12-12T17:25 2
	Validate app returns ERROR ENCOUNTERED! if userid is in wrong fromat (i.e. USER10, 10, USER) or missing	./reserve_table USER10 2021-12-12 2021-12-12T15:25 2 ./reserve_table 2021-12-12T14:20 20
	Validate app returns ERROR ENCOUNTERED! if date is in wrong format or missing	./reserve_table USER_10 12-12-202 12-12-2021:T15-20 2 ./reserve_table USER_10 2020-12-12 ./reserve_table USER_10 2
	Validate app returns ERROR ENCOUNTERED! if user count is in invalid format or missing	./reserve_table USER_10 2021-12-12 2021-12-12T15:25 23423 ./reserve_table USER_10 2021-12-12 2021-12-12T15:25
	Run performance tests to validate app responds in acceptable times (Based on SLAs) when multiple users try to book at the same time (100, 1000, 5000 users)	

			2
			2