TEST CASES EXPLANATION

After loading the machine with *config* file, the machine will look like -

***Outlets* = 4**	**INGREDIENTS**					
total_quantity	**500**	**500**	**100**	**100**	**100**	**0(Not Avail.)**
	hot_water	hot_milk	ginger_syrup	sugar_syrup	tea_leaves_syrup	green_mixture
hot_tea	200	100	10	10	30	-
hot_coffee	100	400	30	50	30	-
black_tea	300	-	30	50	30	-
green_tea	100	-	30	50	-	30

Following are the various test cases present in the *testcases.json* file. Every test case tests one of the functionality provided by the Coffee Machine / Exceptions / Any other scenario possible.

To run the test cases, enter the test case name you want to run in the following format:

test_case_x

here, 1 <= x <= 8 as eight different test cases have been added in the testcases.json file.

1. ***serveBeverage()***
    a. test_case_1 - A request is made to the Machine for the beverage and **IT IS SERVED** since all the ingredients are present in the Machine in sufficient quantities.
    input - [serve hot_tea, serve hot_coffee]
    output - serves both the requests

    b. test_case_2 - A request is made to the Machine for the beverage and is **NOT SERVED** *because currently the machine does not serve the requested beverage (Exception)*
    input - [serve cold_coffee]
    output - throws exception

    c. test_case_3 - A request is made to the Machine for the beverage and is **NOT SERVED** *because ingredients are present but are not sufficient in the inventory*

input - [serve hot_tea, serve hot_coffee, serve black_tea]
output - black_tea not served because after serving hot_tea and hot_coffee, sugar_syrup is left only 40ml but black_tea required 50ml

d. test_case_4 - A request is made to the Machine for the beverage and is **NOT SERVED** *because one or more ingredients required by the beverage is not present in the inventory (Exception)*

input - [serve green_tea]
output - green mixture not available, throws exception

2. ***refillAll()***
   a. test_case_5 - Extended version of TestCase3, where we refill the ingredients and place the same request again and the request **IS SERVED**

input - [serve hot_tea, serve hot_coffee, serve black_tea, refill_all, serve back_tea]
output - this time black_tea will be served

3. ***showIngredientsRunningLow()***
   a. test_case_6 - Will check if any ingredient is running low initially and then after serving a few beverages, will check if now any ingredient is running low.
Note: *An ingredient is said to be running low if its current availability is less than 25% of the maximum possible.*

input - [serve hot_coffee, serve black_tea, show_ingredients_running_low]
output - hot_water, hot_milk, sugar_syrup are left < 25% of the maximum value, so they will be indicated as running low

4. ***refill()***
   a. test_case_7 - Will refill only the required ingredient, and then serve the beverage which could not be served earlier

input - [serve hot_tea, serve hot_coffee, serve black_tea, refill sugar_syrup, refill hot_water, serve black_tea]
output - first time back_tea will not be served but the next time it will be served as the required ingredients are filled

   b. test_case_8 - refill a ingredient which is not present
input - [refill sugar]
output - exception throw