Black box testing techniques.

Phone Number – Black box testing techniques
General description:
User calls an API "…/phone-number/:total", where user inserts a value for total, the API calls a function that generates an array of phone-numbers and sends back in a JSON format.

Valid inputs include the integers 1 and 100, and all integers between the two.

Valid equivalence partitions:
- 1 to 100

Non-valid equivalence partitions:
- Negative integers
- Non-integer numbers
- Non-numeric characters
- 0

Output partitions
- An array with phone numbers in string format
- Array.length = input number

Boundaries:
Valid boundaries: 1, 99
Invalid boundaries: 0, 101

Lower boundary:
- Invalid: -1
- Invalid: 0
- Valid: 1

Upper boundary:
- Valid: 99
- Invalid: 100

Testing the boundaries, we have the following input values and the expected output

#	Input	Output
1	101	100
2	100	100
3	99	99
4	50	50
5	1	1
7	0	0
8	-1	0
9	@SymbolsAndLetters	0 / HTTP-ERROR

While test cases #1-8 are tested within the phone-number generation function itself, and strings, symbols, and non-integer inputs are tested in the API function.

Black Box Testing Documentation

Endpoint `/identity/:amount`

> Description: Takes in the `amount` path parameter and returns `amount` of identity objects.\ The `amount` can range from `2` to `100`.

Partitions:

- Valid Partitions:
  - `2 to 100`
- Invalid Partitions:
  - `<2`
  - `>100`
  - `Characters`

Boundary Value Analysis:

Partition	Invalid Lower Boundary	Valid Lower Boundary	Invalid Upper Boundary	Valid Upper Boundary
`2 to 100`	`1`	`2`	`101`	`100`

Test Cases: `10`, `2`, `100`, `101`, `1`, `thisIsNotNumber`