**Unit Testing of the is\_valid Function**

The objective of the is\_valid function is to validate whether a password meets specific criteria required for registration. It checks for minimum length, the presence of at least one uppercase letter, and at least one special character.

The is\_valid(password) function takes a single input, password, which is expected to be a string. Thus, out of {int, float, string, list}, the first three are invalid types. We create one test case for each invalid type and assert that the function raises a TypeError or returns False.

We now turn to valid input types. The input domain for valid strings can be divided into the following equivalence classes (EC):

1. **EC1**: Password is shorter than 8 characters.
2. **EC2**: Password is at least 8 characters but contains no uppercase letters.
3. **EC3**: Password is at least 8 characters and contains uppercase letters but no special characters.
4. **EC4**: Password is at least 8 characters and satisfies all conditions.

The coverage criteria we will use are: For each equivalence class, we need to test at least one input. This leads to the following test requirements:

* **R1**: If input is in EC1, is\_valid should return False and print an error message about the minimum length requirement.
* **R2**: If input is in EC2, is\_valid should return False and print an error message about missing uppercase letters.
* **R3**: If input is in EC3, is\_valid should return False and print an error message about missing special characters.
* **R4**: If input is in EC4, is\_valid should return True.

These lead to the corresponding test cases:

* **TC1**: Input = "V1!" (EC1)
* **TC2**: Input = "valid123!" (EC2)
* **TC3**: Input = "Valid123" (EC3)
* **TC4**: Input = "Valid123!" (EC4)

For invalid input types, the following test cases are added:

* **TC5**: Input = 12345678 (integer)
* **TC6**: Input = 12.34 (float)
* **TC7**: Input = ['Abcdefgh'] (list)
* **TC8**: Input = “” (empty string)

In order to execute the tests, the function is\_valid is called with each test input, and the results are compared with the expected output to ensure correct behaviour.