Cross-correction: Tests of Function 1

Corrected by group: T3
Group we are correcting: T02

We recommend the following improvements:

- Datatype int suits better the variable “number of days”. Otherwise, you cannot define the boundary values as they are right now. For strings that take a limited and small number of values, you have to define an equivalence class for every possible value the string can take (in this case, you would have to do 10 tests: one for “1”, one for “2”, one for “3”...). The same goes for “phone number”, although for this one it is possible that datatype string is better in order to allow for example the phone number “012345678” (if it was int, the beginning 0 would be lost).
- You are using invalid ID_CARDs for valid test cases (for example: "12345678A" in TC1, "12345678B" in TC2, "12345678C" in TC3...), so the expected result will be an exception instead of a valid localizer. In all of them it should be for example "12345678Z", and the only place where the ID_CARD should be invalid is in the invalid test cases for ID_CARD.
- “OUTPUTS”, “OUTPUT LOCALIZER” and “OUTPUT FILE PATH” are not variables, they should be in the “criteria” column instead (and the “variable” column should be empty).
- You should put in column “result” either something like “valid localizer” or the specific exception that those inputs raise.
- TC1 should not have “all variables” as the field because, actually, we could say that for every valid test case. It should have a specific variable (choose whatever variable you want).
- TC35-TC41 (output test cases) are impossible to implement, because there is no way to assert that you will obtain those outputs with the given inputs. You should only write output test cases when you can find inputs that give that output.
- You should write an example for every EC and BV defined (for example: BVNV1 only says “17 digits” and does not have a specific example of a credit card with 17 digits).
- In ECV4-ECV7 and BVV2 you are putting as an example an invalid ID (it should be 12345678Z instead of 12345678A).
- You have ECNV for datatypes that give the wrong examples (for example: “name and surname” should be a string but the example given in ECNV14 is a string “THIS IS NOT A STRING”).
- In variable “ID”, you should define a ECNV for less than 1 letter at the end and another one for 2 letters at the end (and their respective BV). Also, you should define a ECV of valid letter and a ECNV of invalid letter.

- In variable “name and surname”, you should define ECNV for more than 1 whitespace separating (and its respective BV). Also, a ECV for a string with 3 substrings.
- You should define more EC to check the valid format of “arrival”, in a similar fashion as “ID” (for example: that DD is actually 2 digits, YYYY actually 4 digits, that 1 “/” separates DD and MM…). You should also check that the date that is given corresponds to a date that exists in real life (for example: “40/12/2024” does not exist).