ACIT2515

Licence plates in Manitchewan

In the fictional province of Manitchewan, all cars must have a licence plate. The format of the licence plate is: `NN-LL-NN`, where:

- `N` is an integer from 1 to 9 (included - no 0s !)
- `L` is an uppercase letter from A to Z (included)

`check_plate`

Using the `test_plates.py` file, write the `check_plate` function. It must return `True` if the argument provided is a valid license plate, and `False` otherwise.

`find_next_plate`

This function allows you to generate the licence plate numbers coming after a given licence plate. It takes two arguments:

- the current plate number
- how many plates should be generated (default value: 1)

It returns a string: the **last** licence plate number generated.

Examples

- `find_next_plate("11-AA-11")` is `11-AA-12`
- `find_next_plate("11-AA-19")` is `11-AA-21`
- `find_next_plate("11-AA-99")` is `11-AB-11`
- `find_next_plate("11-AZ-99")` is `11-BA-11`
- `find_next_plate("11-ZZ-99")` is `12-AA-11`
- `find_next_plate("11-AA-11", 2)` is `11-AA-13`
- `find_next_plate("11-AA-11", 8)` is `11-AA-19`

Hints

`chr` and `ord` are very useful functions to transform a character into its Unicode (ASCII) value and back. You may want to use them to manage the letters on the license plates.

For example:

```
>>> ord("A")
65
>>> chr(65)
```

```
'A'
>>> chr(ord("A") + 1)
'B'
```

Submission and grading

- Make sure all tests pass.
- Submit your file to D2L.
- `test_check_plate`: 1 mark
- other tests: 2 marks (6 total)
- **= 7 marks total**