Internet Programming 2

Learning Mini-Project 3: Vehicle Registration Number (VRN) Sequencer (Ver 1)

Project Specifications - February 2025

Notes:

At the last lecture, I briefly discussed the New Vehicle Registration Number (VRN) System that was implemented last year, in Kwa-Zulu Natal. As a pre-cursor to that project, here is a simpler mini project for you to explore, problem-solving and implement in JavaScript. This can be implemented using basic structured programming concepts (remember the structured programming theorem from first year?).

Project Outline

Below is an outline for an introductory version of the Vehicle Registration Number (VRN) Sequencer Project **specifically for the Eastern Cape province (EC)**. The format is **AAA NNN-EC**, where:

- **AAA** is a three-letter sequence (for the first version of your solution, use all 26 letters of the alphabet)).
- **NNN** is a numeric sequence ranging from **000** to **999**.
- The increment order is *right to left*, meaning **NNN** increments first. After 999, it rolls over to 000 and **AAA** increments.

---

1. Detailed Problem Statement

Project Description

You will implement a JavaScript program that, given a valid **EC** vehicle registration number in the format **AAA NNN-EC**, will produce the next registration number in the sequence. The sequence rules are:

1. **AAA**: Three letters.
2. **NNN**: A three-digit number that goes from **000** up to **999** (inclusive).
3. **-EC**: A fixed suffix identifying the Eastern Cape province.

Incrementing:

- Start by incrementing the **NNN** part by 1 (000 → 001 → 002 → … → 999).
- If **NNN** reaches **999** and needs to increment again, it rolls over to **000**, and **AAA** is incremented to the next valid three-letter sequence.
- If **AAA** also reaches its maximum (e.g., ZZZ) and needs further increment, you must decide whether to display an error or wrap around to the earliest possible value (BBB, if you skip vowels).

Examples of Incrementing

1. **Simple Increment (NNN)**
    - **Input**: BBB 000-EC
    - **Next**: BBB 001-EC
2. **Rollover from 999**
    - **Input**: BBB 999-EC
    - **Next**: BBC 000-EC (assuming the next valid three-letter sequence after BBB is BBC; NNN resets to 000).
3. **Overflow of Three Letters**
    - If you reach ZZZ 999-EC and need to increment, you may return an error message: "Maximum registration reached"

Test Cases

Here are some registration numbers and their expected next values:

1. **BBB 000-EC → BBB 001-EC**
   (Simple numeric increment)
2. **BBB 123-EC → BBB 124-EC**
   (Increment numeric from 123 to 124)
3. **BBB 999-EC → BBC 000-EC**
   (Rollover numeric from 999 → 000, increment letters from BBB → BBC.)
4. **BBZ 999-EC → BCB 000-EC**
   (If we assume the sequence after BBZ is BCA, but A is not allowed, so it might be BCB. This is an example of careful letter skipping logic.)
5. **ZZZ 999-EC → Error**
6. **Z12 D99-EC → Error**
7. Use these tests to confirm your solution.

PK Ramdeyal, 14 February 2025.