**Software Development Engineer in Test (SDET)**

**Candidate Brief**

**Overview of the Exercise**

In this exercise you are required to present your solution to the panel.

You will be required to deliver your findings to the panel orally and by Screen Share to deliver your presentation.

**Background Information**

You have been given a CSV file containing data on a large volume of vehicles.

This data needs to be parsed, validated, and stored in the correct format.

**Validation rules:**

*All characters are case insensitive.*

**VRN**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| L | L | D | D |  | L | L | L |
| Letter | Letter | Digit | Digit | Optional space | Letter | Letter | Letter |

Examples

Valid: [ AB12XYZ, Ab12 XyZ ]

Invalid: [ ABC 1YZ ]

**Make**

One of: [ BMW, AUDI, VW, MERCEDES ]

Examples

Valid: [ BMW, bMw ]

Invalid: [ Skoda ]

**Colour**

One of: [ WHITE, BLACK, RED, BLUE ]

Examples

Valid: [ WHITE, WhiTE ]

Invalid: [ Green ]

**Date of Manufacture**

Date of manufacture cannot be in the future.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| D | D | - | M | M | - | Y | Y | Y | Y |
| Day  Optional leading zero  Range: 1-31 | | Hyphen | Month  Optional leading zero  Range: 1-12 | | Hyphen | Year  Range: 1900 – Present year | | | |

Examples

Valid: [ 01-01-2012, 1-01-2012 ]

Invalid: [ 01,01,2012, 01012012, 01-01-2024 ]

**Format:**

When the CSV data has been parsed and validated, it will need to be stored in the following format:

|  |  |  |  |
| --- | --- | --- | --- |
| VRN | Make | Colour | Date of Manufacture |
| LLDD LLL   * Space mandatory * All uppercase | * BMW * Audi * VW * Mercedes | * White * Black * Red * Blue | * Abbreviated day * Zero padded day value followed by the full month name * Full year   Example:  Tue, 01 January 1990 |

**Presentation Scenario**

You will need to author code which performs two separate functions:

1. Parse the CSV data file and determine how many valid and invalid vehicles the file contains.
   1. For invalid vehicles, the reasons why they are invalid should be present.
2. For the valid vehicles, output their data in the correct format i.e., terminal or file.

You may use **any** reasonable general-purpose programming language to implement your solution. (Your audience will be unlikely to understand deliberately esoteric and obfuscatory toy languages, e.g., Malbolge.)

The code is expected to compile and run successfully. You may use third-party libraries to assist you, but the implementation of task must be original.

You **do not** have to implement testing for this code, **but** you should consider how you might test it since you may be asked about it.

**You must submit your solution at least 24 hours before your interview.** Please provide a link to a public source code repository such as GitHub to [ITSRecruitment@dvla.gov.uk](mailto:ITSRecruitment@dvla.gov.uk)

**The Presentation Task**

You should prepare a short (10 minutes) presentation in advance. You will be asked to present this during the interview.

Your slides and talk **should** cover:

* A working demo of your code against the two criteria
* A general overview of how your code works
* Aspects of the task you enjoyed or disliked
* Aspects of the task you found difficult and why
* Aspects of your code you would improve or change if you refactored it or did the task again

Your slides and talk **should not** cover:

* A line-by-line explanation of the code