# A: DOCUMENTATION

**Name and Title: Car Sales Data Processing**

**Introduction:**

Car dealers maintain records of sales of various models of passenger cars sold by them for data analytics purposes. A CPP program can be used to analyze this data by loading data records into an STL container such as an array or a Map.

This problem statement in C++ language focuses on data of car models and the analysis and computations that can be performed on this data.

This program stores data values in a Map with the following string keys:

|  |  |
| --- | --- |
| **Struct Member Name** | **Description** |
| **SaleUnits** | It indicates the number of units sold for the particular vehicle model. |
| **Price** | It indicates the price for the particular vehicle model. |
| **Efficiency** | It indicates the efficiency(miles per litre of fuel) for the particular vehicle model. |
| **Horsepower** | It indicates the horsepower for the particular vehicle model's engine. |

**Problem Description:**

The following LoanApplication structure records can be considered for your reference. Update these record entries in the bitmap.c file as per instructions (comments) given.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Record No**. | **SaleUnits** | **Price** | **Efficiency** | **Horsepower** |
| 1 | 17000 | 45000 | 10 | 140 |
| 2 | 10000 | 65000 | 12 | 160 |
| 3 | 20000 | 35000 | 14 | 155 |
| 4 | 38 | 45000 | 10 | 140 |
| 5 | 27 | 21000 | -1 | 145 |

Store these values in map objects of string keys and int values. Then, add these to a standard vector which accepts values of this map definition.

**Note:**

* **-1 indicates invalid or missing value.**
* **Price will always be less than 100000.**
* **Efficiency will always be less than 100.**

In this problem statement, you are expected to complete the following functions, using the LoanApplication structure and data records given above for reference.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function Name** | **Arguments** | **Return Type** | **Description** |
| AverageCarPrice | Reference to a standard vector of map objects with string keys and int values. | float | It Calculates the average price of a car record(excluding invalid price values) and returns the answer rounded off to the nearest integer. |
| MaxHorsepower | Reference to a standard vector of map objects with string keys and int values. | int | It returns the maximum Horsepower observed in the records(excluding invalid values). |
| LowestEfficiency | Reference to a standard vector of map objects with string keys and int values. | int | It returns the lowest efficiency observed in the records(excluding invalid values). |

**Test cases:**

**\*\*PLEASE DO NOT EDIT THE TEST CASES**

The above-mentioned requirements have been considered for creating test cases. The below table gives details of test cases to be used for validating written code.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case No.** | **Test Case Description** | **Input Values** | **Expected Output** | **Explanation** |
| 1 | To validate *AverageCarPrice* functionality | Standard vector containing the above mentioned record values. | 41250 | This is the average price in the above mentioned car sales records. |
| 2 | To validate *MaxHorsepower* functionality | Standard vector containing the above mentioned record values. | 154 | This is the maximum horsepower observed excluding missing data values. |
| 3 | To validate the *FindLowestEfficiency* functionality. | Standard vector containing the above mentioned record values. | 8 | 8 is the lowest efficiency in the data. Note, -1 is considered as invalid value here too. |