Car sales' statistics

In this activity, you need to generate some statistics related to total sales of a car dealership. The sales file will be provided to your program as the first argument. The file contains five columns: type, price, year, color, and quantity in comma-separated format such that the value of any column will either be a non-negative integer (>=0) or string. You don’t have to deal with doubles or floats. Also note that there won’t be any pair of rows such that they have the same values in the first four columns i.e. no pair of rows differ “only” in the quantity column.

LabDirectory contains the following files:

- car\_sales.sh: This is the bash script that you need to complete.

- testcases: This folder contains the testcases for the bash script.

- sales.csv: This file contains the sales data.

- problem.pdf: The pdf file you are currently reading.

Your script needs to answer two types of queries [NOTE: name of the csv file is given as command line argument] : (Output to each of them is a single line that is to be printed on the terminal.)

1. The most sold color of car, based on either “type”, “year” or “price”. The“type”, “year” or “price” is provided as the third argument and the value is provided as the fourth argument to the bash script. If only two arguments are provided, then the most sold color of car is printed regardless of “type”, “year” or “price”.

Valid queries will be of following format:

- ./car\_sales.sh sales.csv 1 type sedan

- ./car\_sales.sh sales.csv 1 year 2023

- ./car\_sales.sh sales.csv 1 price 200000

- ./car\_sales.sh sales.csv 1

2. The total amount of sales from cars such that the price of each car is strictly greater than a given value and the car is of a particular color. The price and color are provided as the third and fourth argument respectively to the bash script. If the fourth argument is not provided, then the total amount of sales such that the price of each car is strictly greater than the given value is printed. If both third and fourth arguments are not provided, then the total amount of sales is printed regardless of “type”, “price”, “year”, “quantity,” or “color”.

Valid queries will be of following format:

- ./car\_sales.sh sales.csv 2 200000 red

- ./car\_sales.sh sales.csv 2 200000

- ./car\_sales.sh sales.csv 2

ERROR HANDLING (in order of precedence, if multiple conditions are met, then the first one should be executed):

- If no argument is provided, then the script should print without quotes "ERROR: No arguments provided" and exit with status 1.

- If only one argument is provided, then the script should print without quotes "ERROR: No query type provided" and exit with status 1.

- If the first argument is not the name of an existing file, then the script should print without quotes "ERROR: File does not exist" and exit with status 1.

- If the second argument is neither 1 nor 2, then the script should print without quotes "ERROR: Invalid query type" and exit with status 1.

All testcases assume the following:

- If the file provided as the first argument exists, then it will be in the correct format.

- The command line arguments will be provided in correct order and will be of the correct type.

- For both queries, if you do not find any car based on arguments provided, then you should print without quotes "NRF".

- The error handling mentioned above is complete i.e. you don’t need to handle any error that is not mentioned above.

- For query 1, there won’t be two colors that have the same most sold cars for any particular trait.

- Any argument will either be a non-negative integer (>=0) or string. You don’t have to deal with doubles or floats.

Important Points:

- Please run the autograder scripts to check if your program passes on public testcases.

- Timeout for each testcase is 5 seconds but don’t worry, if you write reasonably efficient code then you won’t face the issue of timeout.

Marks distribution (out of 6):

- 2.5 marks for each query type. All these testcases will be according to the above mentioned format of queries with no errors at all.

- 1 mark for error handling. All these testcases will have some error in the arguments provided.

Submission:

- You need to submit the bash script (car\_sales.sh) only.

DISCLAIMER:

- Marks provided by the autograder are provisional. Final marks will be awarded after evaluation on hidden testcases. So you should not hardcode the answers based on the provided testcases. Try to write your own testcases to test the correctness of the script.