Kyrgyz-German Institute of Applied Informatics

Faculty of Web Informatics

**Program for car sales accounting**

(Coursework)

Completed by students of 1st grade

from group WIN-1-20

Chermashev Aman

Seitbek uluu Salavat

Temirov Mirlan

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lektor

Kibekbaev Azamat Zhumanazarovich

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Table of contents**

1.Introduction

2.Team roles

3.Where can you download the program and run it

4.Functionality of the program

5.Program code

6.User’s guide

7.List of the future features

8. Conclusion

**Introduction**

The subject of our project is “Car sales accounting”. We were tasked to make a program for a State Registration Service.

This is a team project and every person of the team is responsible for their tasks and roles.

**Team Roles**

* **Chermashev Aman** - Teamlead, documentary, analytics
* **Seitbek uluu Salavat** - Developer, documentary
* **Temirov Mirlan** - Developer, tester

**How to download our project**

Our team decided to create a GitHub repository.

Here is the link:

<https://github.com/WIN-1-20-5/coursework-1>

If you want to download the project, you need to do next steps.

Attention, you need to have a bash console or GitHub Desktop.

If you are using bash console:

1. Go the directory where you want to download repository
2. Type command git clone [git@github.com](mailto:git@github.com):WIN-1-20-5/coursework-1.git

If you are using GitHub Desktop:

1. Click button “File”
2. Click “Clone repository”
3. Click button “URL”
4. Paste link in input for URL <https://github.com/WIN-1-20-5/coursework-1>
5. Click button “Clone”

**How to run the program**

You can run the program using IDLE or VS Code.

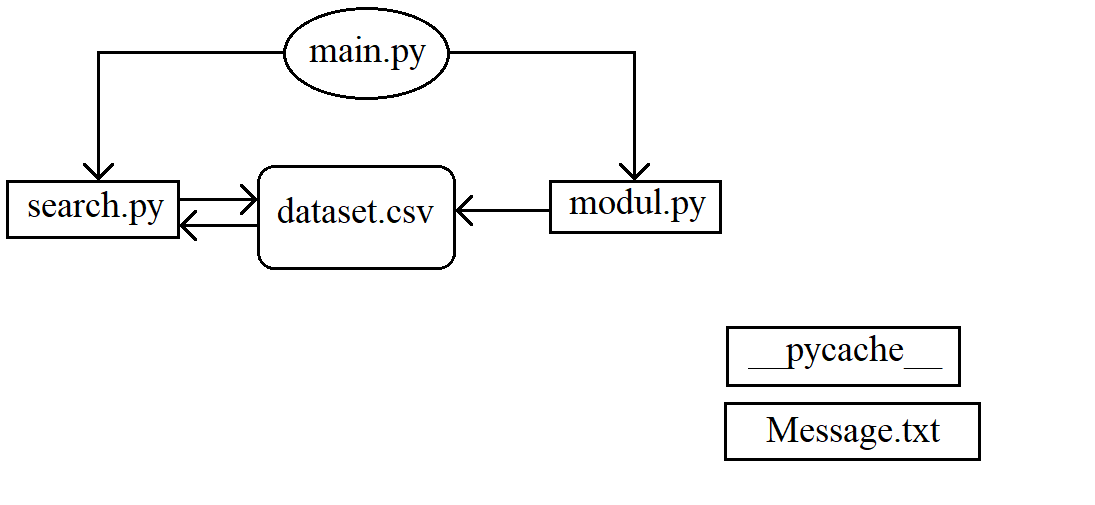
If you use IDLE:

1. Open IDLE
2. Open file main.py
3. Click F5

If you use VS Code:

1. Open VS Code
2. Open file main.py
3. Run file main.py in terminal

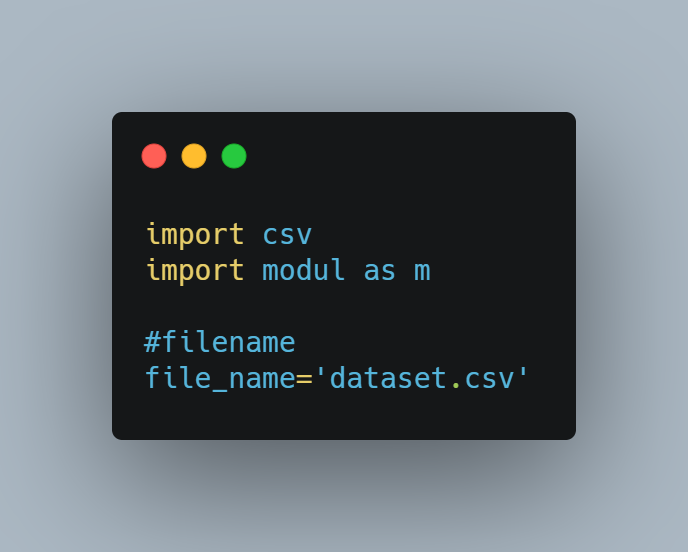
**Functionality of the program**

****

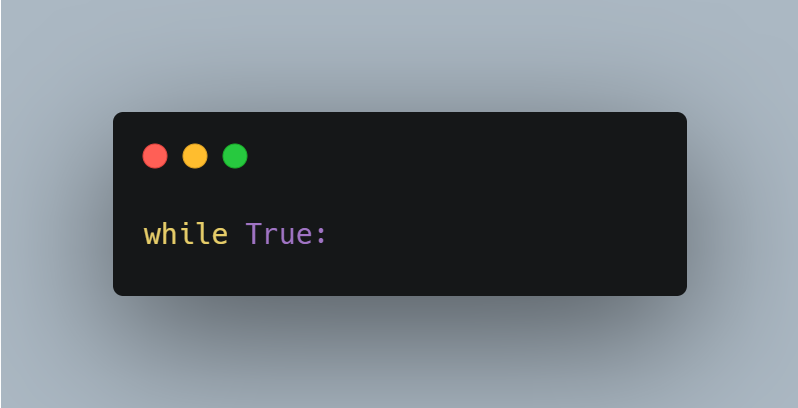
* “main.py” - the main program, all project functions are performed in it. It picks up functions from "modul.py" and links to search. This is where almost all interactions with the "dataset.csv".
* “modul.py” - a subordinate module that contains most of the functions that "main.py" uses. It is not a standalone program, when you activate it, you will only see a dialog box asking you to switch to the main program.
* “search.py” - a subordinate program responsible for search functionality. Unlike “modul.py” it has independence, due to which it can be launched in general from “main.py”, and from it it does not affect “dataset.csv” in any way, only taking the necessary data from it.  
  The "search" program has a different structure, therefore it needs a more detailed explanation. Abstractly speaking, it has 2 switches: "filter" and "sorting", where the first switch, changing its position, saves the changes (off / on for each brand / color) for further use, while the second can have only one position , without intermediate divisions; at the same time, both of them are used together (analogous to the temperature of the air conditioner and the direction of heating the machine)  
  Filter:
* • None - standard status, only it is exciting. To switch to it you need to enter "0". Does not accept text and empty lines. To exit the program, re-enter "0".
* • Mark - a status that allows you to select the necessary brands of cars of all available by simple input (case is not taken into account, write at least in lowercase, at least in capital letters). To switch to it, you need to enter "1"; re-entering reverses the filter values.
* • Color - a status that allows you to select the required colors of all available cars by simple input (case is also not taken into account). To switch to it, you need to enter "2"; re-entering reverses the filter values.  
  Sorting:
* • Mutual No - sorting is not applied.
* • Cost: Down - sorting from the lowest price is applied
* • Cost: Up - sorting from the maximum price is applied
* • Old: Down - sorting from the minimum age is applied
* • Old: Up - sorting from the maximum age is applieв  
  Note: by entering the same sorting number ("3" / "4"), you alternately change its status: No> Down> Up> No, but at the same time entering a different number, the first one changes to "No", while time as the second the next.
* “dataset.csv” - the project database, it contains all the information that changes when "main.py" is executed, may change, but "search.py" is only gleaned.
* “\_\_pycache\_\_” - a directory with necessary files to execute the program.

**Program code**

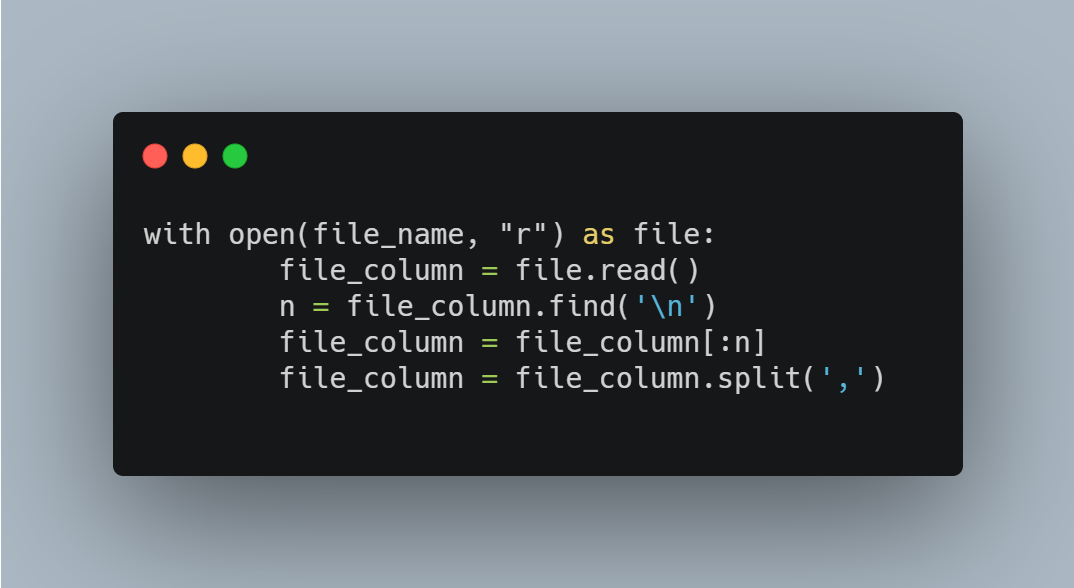
“main.py”



declaration of libraries and variables



while loop, the main loop of the program



writing down the names of the columns for future use

****

do the same for the number of rows in the database (zero does not count)

****

we arrange our rows and columns by adding missing columns or getting rid of unnecessary ones



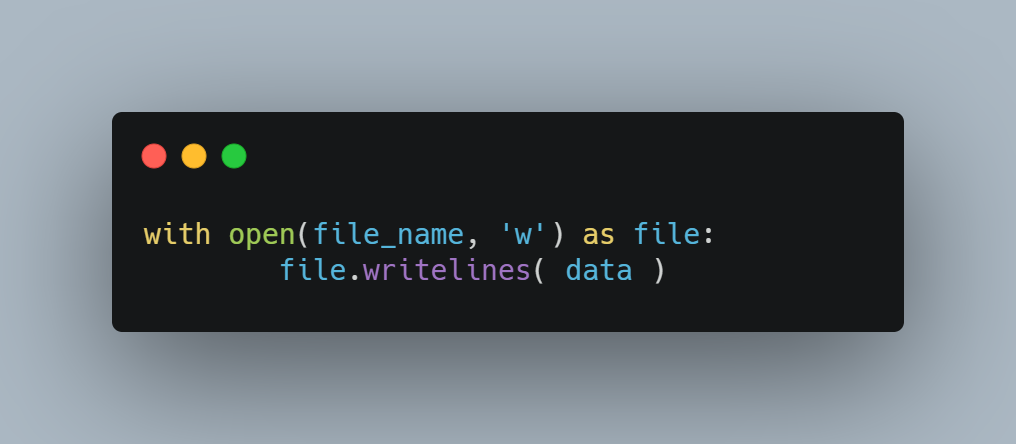
substitutes 2 dots instead of incorrect data



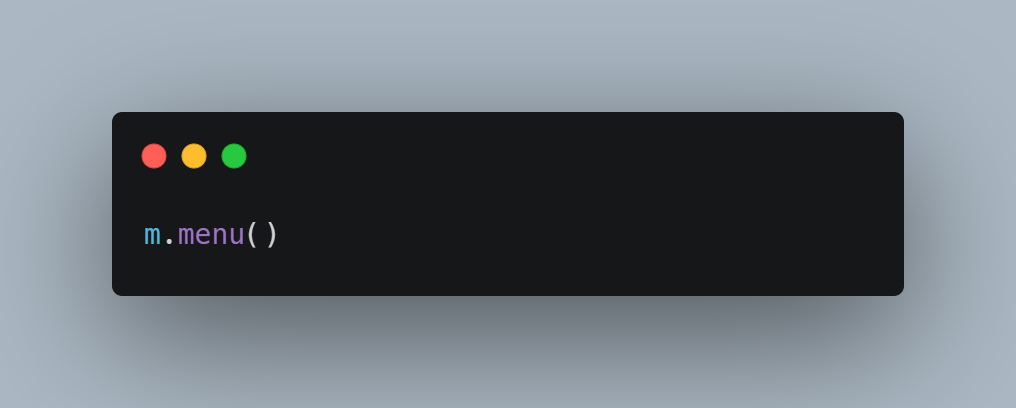
replaces any text in the carmasters column (number of previous owners) with 0



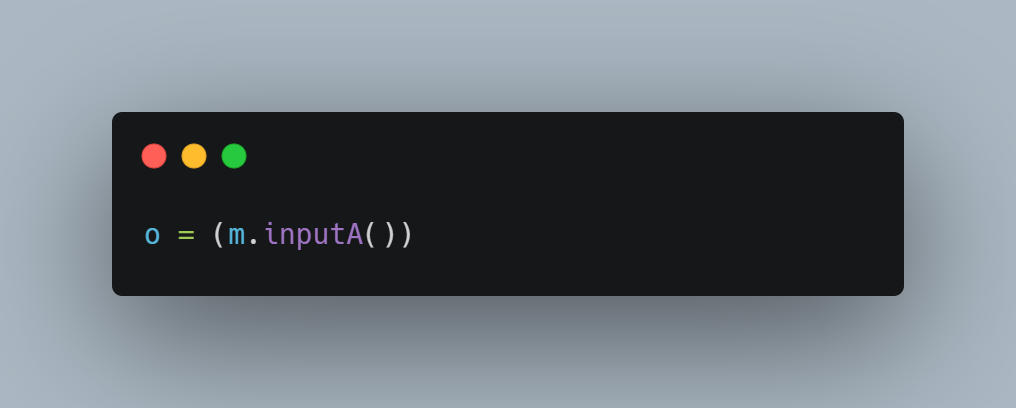
convert the price to a python number



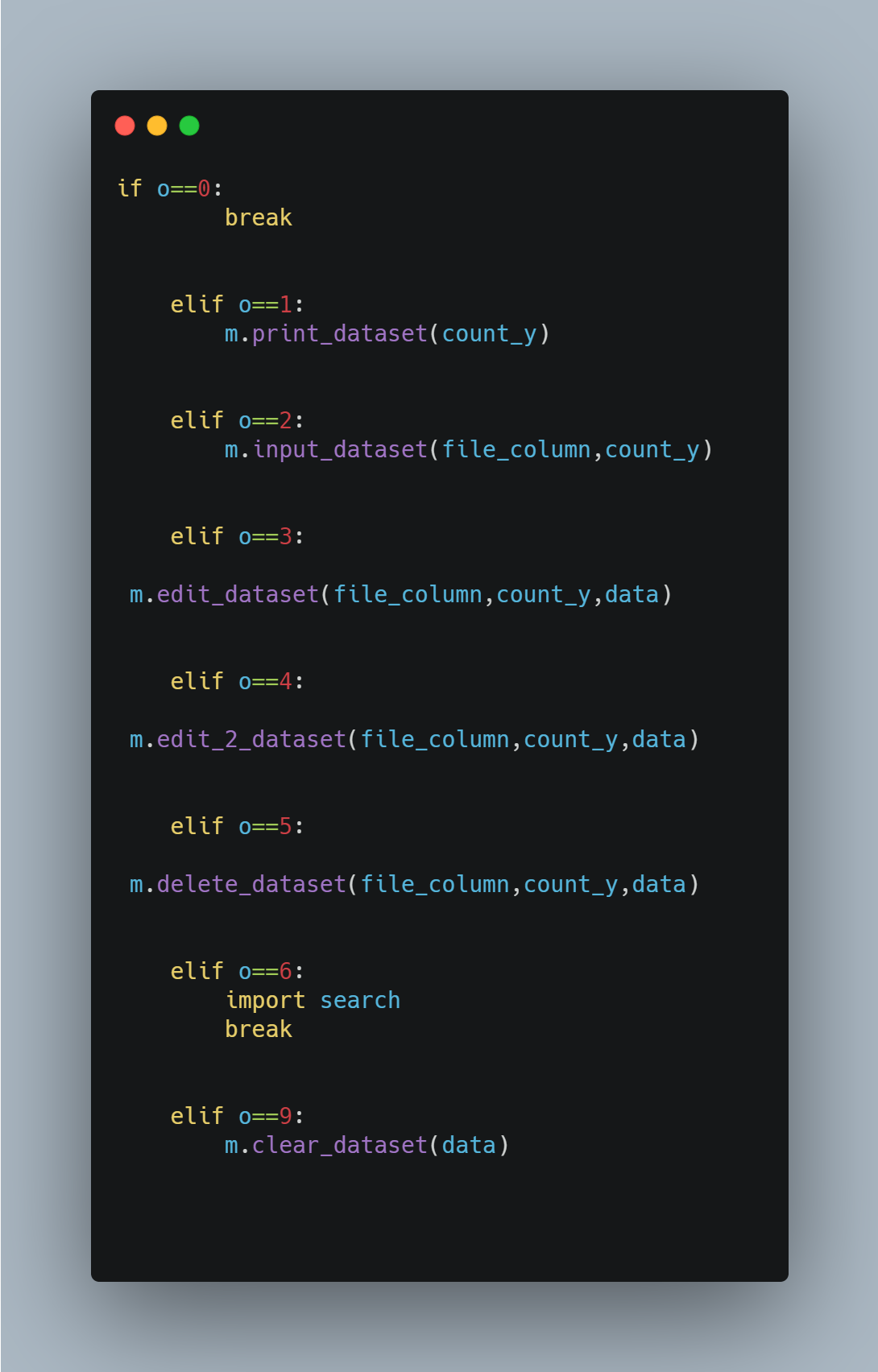
save the data in the file to a Python list of strings for further work

****

here we display our menu



input that only accepts numbers, used to select further operation



the main functionality of the program:

database output

input ...

“modul.py”



declaring libraries and variables



menu output function



data output



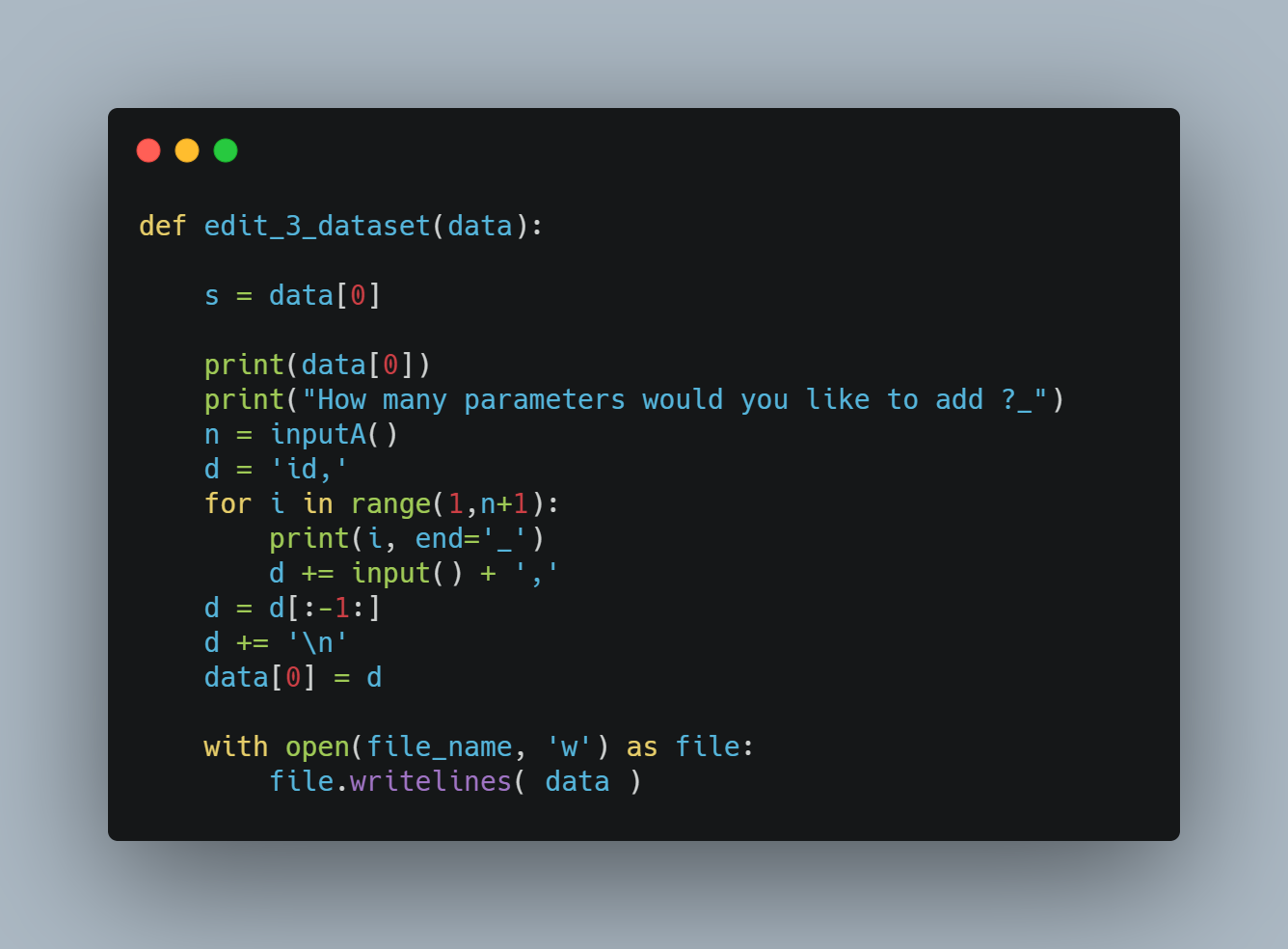
data input



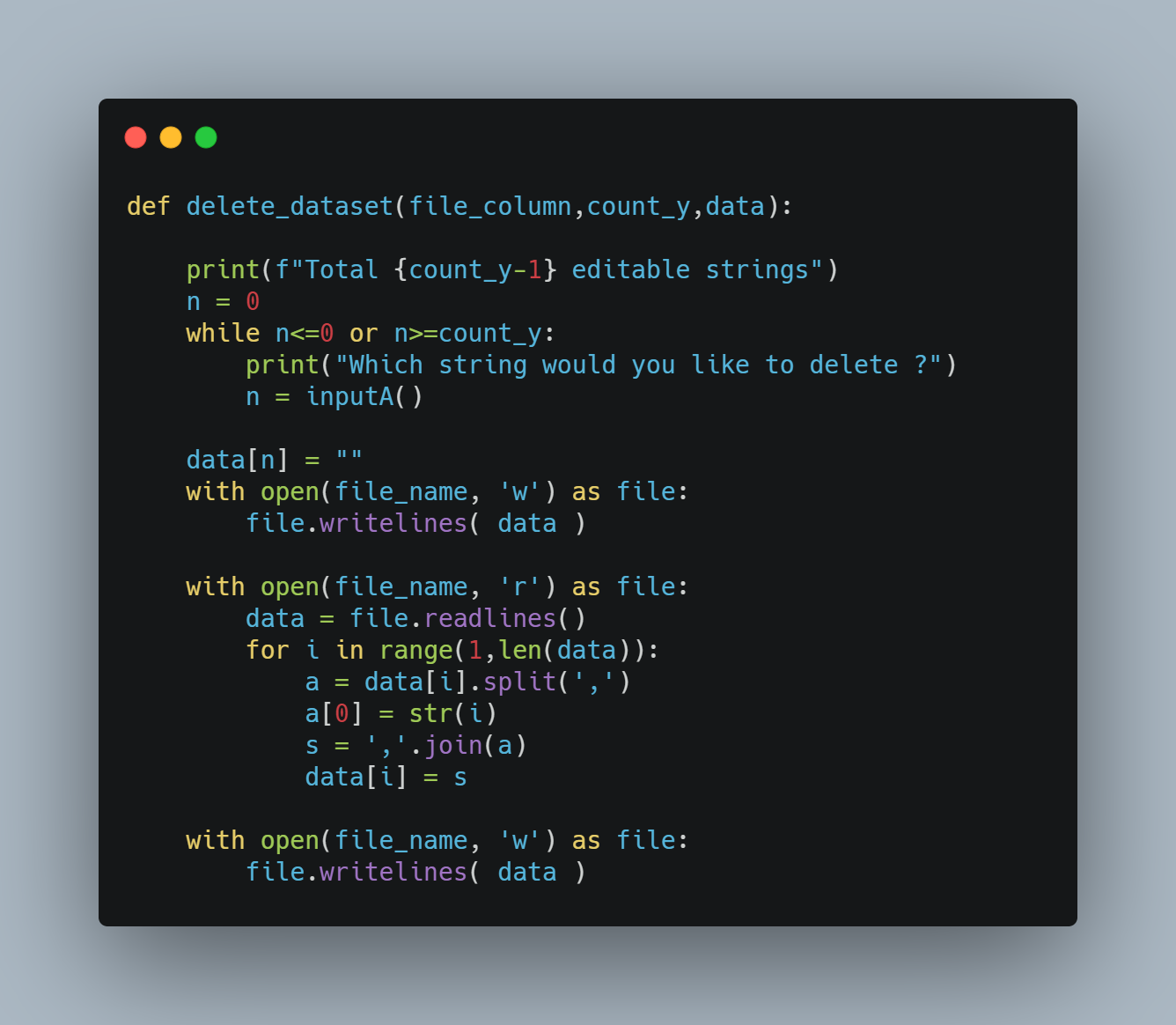
correction of "non-zero line"



change a specific parameter



change parameters (zero-string)



delete line



clean dataset

“search.py”



declaring libraries and variables



menu call



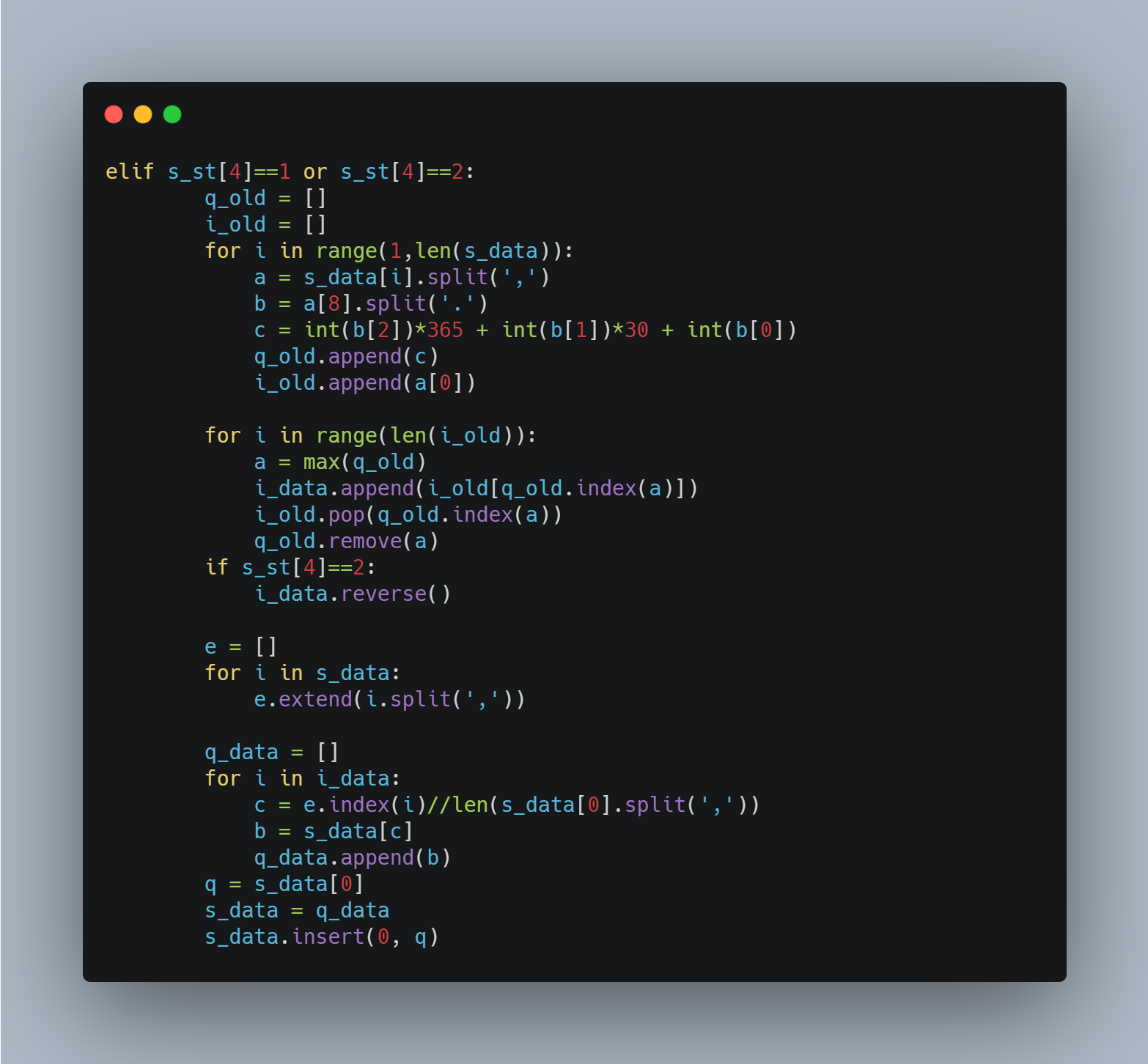
filter selection



filter application



sorting by cost



sorting by age



output



functions for filter



checking database compliance

writing down the column names

measure the number of lines

write all data

remove extra columns

add commas if there are less than required



write down stamps and color





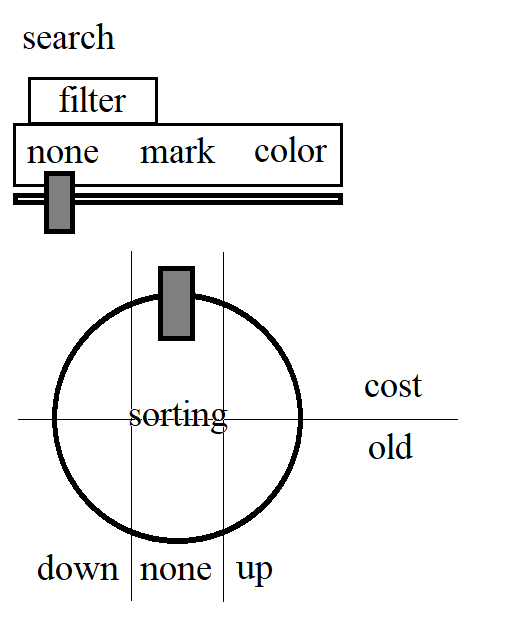


head program of search

**User's guide**

When the program starts user will meet the menu with 8 options:

1. Show dataset
2. Write new data
3. Edit data
4. Edit template
5. Delete data
6. Search in dataset
7. Clean dataset
8. Exit

* Show dataset - outputs data in the form of a table with sizes expandable from the data. If the data size exceeds the "IDLE" window, the font of the lines does not decrease, but continue from a new line, completely damaging the entire design of the program. Therefore, there is a mental limitation on the size of the data.
* Write new data - adds a new line with user data to "dataset.csv". At the same time, there is protection against typos, which allows you to enter invalid, but not valid data. So, in the column "car\_cost" and "car\_masters" (cost and number of previous owners, respectively), you can enter text, but the program will replace it with 0; you can write an impossible date in "car\_old" (or simply change the format from the built-in DD.MM.YYYY), in this case the program will not do anything, but if you enter it incorrectly (not to be confused with an incorrect one!) it will replace the entire text with “. . ”, Which means the smallest date for the program.
* Edit data - complete change of row values (the row is selected by index, which requires a preview of them, and if you do not know the index, you will have to collapse "IDLE"). The previous values are shown just above.
* Edit template - the only difference from the full one is that it replaces a specific parameter, and not the entire row.
* Delete data - deletes a row of data by index (similar situation with indexes). Lost data cannot be recovered.
* Search in dataset - "search.pyh", which allows filtering by car brands and their color, as well as sorting by cost and age of the car (the program turns a blind eye to incorrect data and looks at the date of the car's creation, and not at its mileage).  
  
* Clear dataset - clears all "dataset.csv" except for "null string" - parameters. When restoring data, it is advised to look at the "database format template", which is a "dataset" layout, but not a fallback, although it can be used by the user in this way.
* Exit - stops program.

**List of the future features**

* UI instead of console.
* Authentification.
* Code refactoring and optimisation.
* Adding more functions.
* Changing DB

**Conclusion**

Now it’s a small program for working with cars on a base level and it doesn’t have enough functionality to work fully with them. But our team will find more information about it and try to make this program even better. Now you just need to wait for the next update.