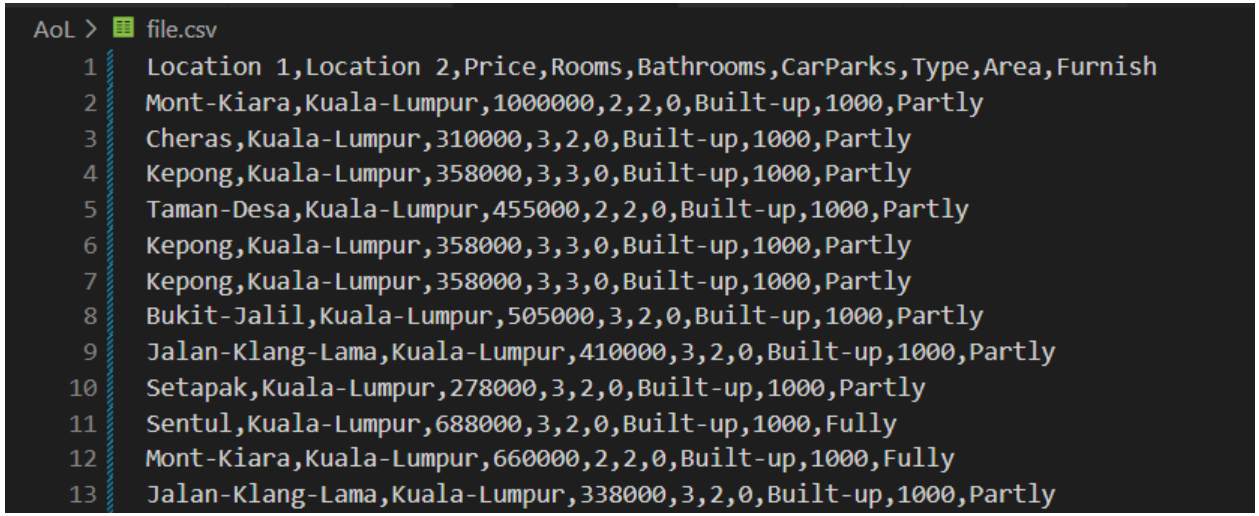
Case Study

You are required to write a program that can read a file and perform several functions. The file needed can be downloaded from link: https://1drv.ms/u/s!AhuAx03LAKWtnOM9O1wlXSAR84Z67g?e=lVmH5x .

File description:

This file is a csv file; therefore, each column is separated with comma. This file consists of 3939 rows of Housing Data in Malaysia with no missing value for each row. This file also has a **header. Therefore, when your program loads the data, your program should be able to skip this header before passing the data into your record variable**. A glimpse of the data:



You are required to perform 3 functions as follows:

1. **Describe**.

   This function explains the information from each column. When running this function, **give a prompt to get input from the user** as the name of the column you want to describe. Then, display:

a. For column loc1, loc2, room, bathrooms, carparks, type, or furnish, display:
    i. Frequency for each unique value
    ii. Maximum frequency
    iii. Minimum frequency

For example, if we call describe function followed with loc1 as column name, the program should display like this:

```
Batu-Caves : 19
Bangsar-South : 40
Kuchai-Lama : 37
Jinjang : 9
Bandar-Tasik-Selatan : 10
OUG : 40
Setiawangsa : 21
Sri-Hartamas : 63
Ampang-Hilir : 29
Seputeh : 21
Pandan-Indah : 10
Mid-Valley-City : 9
Brickfields : 19
Damansara : 5
Gombak : 3
Pandan-Jaya : 5
Alam-Damai : 1
Sunway-SPK : 8
Pandan-Perdana : 7
Other : 2
Happy-Garden : 1
Taman-Sri-Keramat : 1
TAMAN-MELATI : 1
Jalan-Sultan-Ismail : 12
Maximum value: Kepong with frequency: 450
Minimum value: TAMAN-MELATI with frequency: 1
```

b. For area and price column, display only:
    i. Minimum value
    ii. Maximum value
    iii. Average value

Because area and price value are not discrete, therefore there isn't a need to perform frequency check for each of it.

2. **Search Data.**

To be able to handle search function, ask user to give input with format:

```
DataX in ColumnName
```

Your program should be able to parse above input. **It is prohibited to use 3 string input**. Then, display all data that has that DataX.

For example:

```
Partly in furnish
```

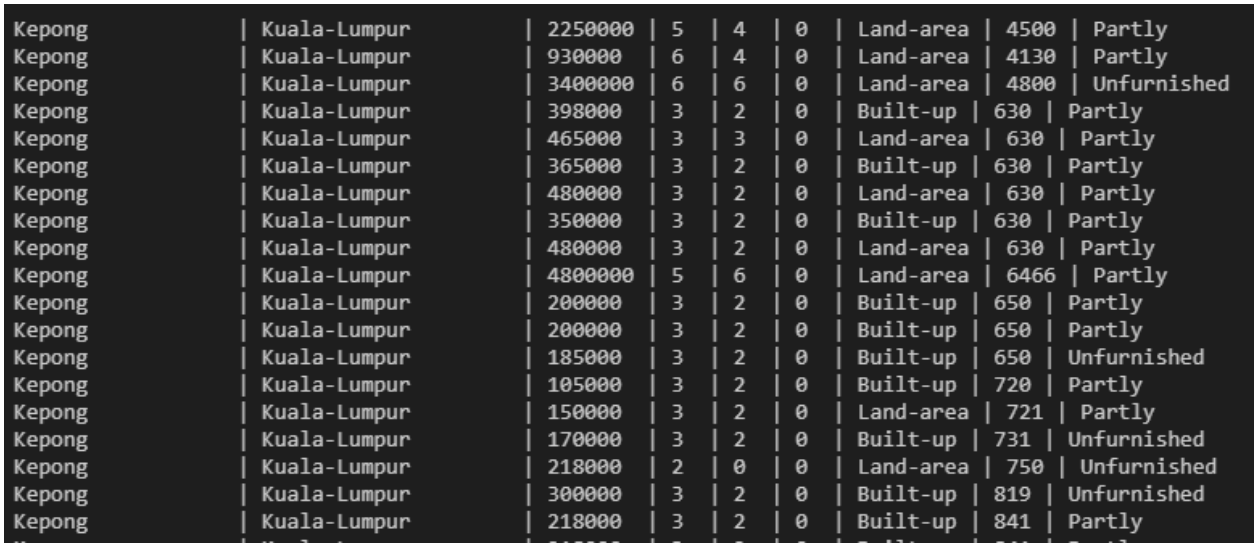
Above command will result in:

```
Kepong          | Kuala-Lumpur | 357000  | 3 | 2 | 0 | Built-up | 973 | Partly
Salak-Selatan   | Kuala-Lumpur | 248000  | 3 | 2 | 0 | Built-up | 973 | Partly
Mont-Kiara      | Kuala-Lumpur | 1160000 | 2 | 2 | 0 | Built-up | 973 | Partly
City-Centre     | Kuala-Lumpur | 800000  | 3 | 2 | 0 | Built-up | 975 | Partly
Sri-Petaling    | Kuala-Lumpur | 390000  | 3 | 2 | 0 | Built-up | 975 | Partly
KLCC            | Kuala-Lumpur | 1400000 | 2 | 2 | 0 | Built-up | 976 | Partly
KLCC            | Kuala-Lumpur | 1400000 | 2 | 2 | 0 | Built-up | 976 | Partly
KLCC            | Kuala-Lumpur | 1400000 | 2 | 2 | 0 | Built-up | 976 | Partly
KLCC            | Kuala-Lumpur | 1400000 | 2 | 2 | 0 | Built-up | 976 | Partly
KLCC            | Kuala-Lumpur | 1400000 | 2 | 2 | 0 | Built-up | 976 | Partly
Ampang          | Kuala-Lumpur | 690822  | 2 | 2 | 0 | Built-up | 977 | Partly
Ampang          | Kuala-Lumpur | 690822  | 2 | 2 | 0 | Built-up | 977 | Partly
Ampang          | Kuala-Lumpur | 690822  | 2 | 2 | 0 | Built-up | 977 | Partly
Ampang          | Kuala-Lumpur | 690822  | 2 | 2 | 0 | Built-up | 977 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 638000  | 3 | 2 | 0 | Built-up | 977 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 486000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 486000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 486000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 540000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 540000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 486000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 486000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 486000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Ipoh      | Kuala-Lumpur | 486000  | 3 | 2 | 0 | Built-up | 978 | Partly
Jalan-Klang-Lama | Kuala-Lumpur | 755000 | 3 | 2 | 0 | Built-up | 978 | Partly
KLCC            | Kuala-Lumpur | 760000  | 2 | 1 | 0 | Built-up | 980 | Partly
```

However, your program also should be able to search using only sub-string, for example:

```
pong in loc1
```

will result in:



```
Kepong          | Kuala-Lumpur        | 2250000 | 5 | 4 | 0 | Land-area | 4500 | Partly
Kepong          | Kuala-Lumpur        | 930000  | 6 | 4 | 0 | Land-area | 4130 | Partly
Kepong          | Kuala-Lumpur        | 3400000 | 6 | 6 | 0 | Land-area | 4800 | Unfurnished
Kepong          | Kuala-Lumpur        | 398000  | 3 | 2 | 0 | Built-up  | 630  | Partly
Kepong          | Kuala-Lumpur        | 465000  | 3 | 3 | 0 | Land-area | 630  | Partly
Kepong          | Kuala-Lumpur        | 365000  | 3 | 2 | 0 | Built-up  | 630  | Partly
Kepong          | Kuala-Lumpur        | 480000  | 3 | 2 | 0 | Land-area | 630  | Partly
Kepong          | Kuala-Lumpur        | 350000  | 3 | 2 | 0 | Built-up  | 630  | Partly
Kepong          | Kuala-Lumpur        | 480000  | 3 | 2 | 0 | Land-area | 630  | Partly
Kepong          | Kuala-Lumpur        | 4800000 | 5 | 6 | 0 | Land-area | 6466 | Partly
Kepong          | Kuala-Lumpur        | 200000  | 3 | 2 | 0 | Built-up  | 650  | Partly
Kepong          | Kuala-Lumpur        | 200000  | 3 | 2 | 0 | Built-up  | 650  | Partly
Kepong          | Kuala-Lumpur        | 185000  | 3 | 2 | 0 | Built-up  | 650  | Unfurnished
Kepong          | Kuala-Lumpur        | 105000  | 3 | 2 | 0 | Built-up  | 720  | Partly
Kepong          | Kuala-Lumpur        | 150000  | 3 | 2 | 0 | Land-area | 721  | Partly
Kepong          | Kuala-Lumpur        | 170000  | 3 | 2 | 0 | Built-up  | 731  | Unfurnished
Kepong          | Kuala-Lumpur        | 218000  | 2 | 0 | 0 | Land-area | 750  | Unfurnished
Kepong          | Kuala-Lumpur        | 300000  | 3 | 2 | 0 | Built-up  | 819  | Unfurnished
Kepong          | Kuala-Lumpur        | 218000  | 3 | 2 | 0 | Built-up  | 841  | Partly
```

- **Give information if data searched doesn’t exist in the record.**
- **YOU ONLY HAVE TO DO SEARCH IN THE COLUMN OTHER THAN AREA AND PRICE.**

3. ***Alphabetizing a List of String.*** In the original question, you are asked to alphabetize list of string using 10 to 15 name of towns. **For this case, implement what is being asked using column Location 1.**