COSC 1336
Spring 2023

Assignment 7: List Sorting and Searching

[1] **Objectives**: This assignment aims to practice using lists and arrays, including a list of lists. We will also be sorting lists based on several values. The operations can be done using the list sort() method or sorted() functions. Do not define a sorting function in this assignment. We will also be searching and selecting values from a list. You are expected to use several methods of the list object.

[2] **Description**: The assignment is to develop a program that helps the customers of a used car dealer select from an inventory list. The list can be visualized as a table of rows and columns. It should be implemented as a list of lists. The inner lists (rows of the table) are records of the cars with the following attributes (columns).

- Make (string)
- Model (string)
- Year (int)
- Mileage (int)
- Price (int)

The table will be generated with the help of a random number generator. The code and some constants will be provided for you below. Use the code without making changes. You may use the constants anywhere in the program, i.e., no need to pass these parameters.

```
import random
makes = [
    'Toyota', 'Honda', 'Ford', 'Chevrolet', 'Nissan',
    'BMW', 'Mercedes', 'Audi', 'Kia', 'Hyundai'
]
models = ['Sedan', 'SUV', 'Truck', 'Van']
plus_minus = [0, 0, 1, 10000, 2000]
size = 20

def get_data():
    random.seed(77)
    cars = []
    for i in range(size):
        make = random.choice(makes)
        model = random.choice(models)
        year = random.randint(2010, 2024)
        mileage = random.randint(100, 100000)
        exotic = 5000 if make == 'BMW' or make == 'Audi' else 0
        price = round((20000-(mileage * 0.01)-(500*(2023-year)))+exotic)
        car = [make, model, year, mileage, price]
        cars.append(car)
    return cars
```

The list returned from the code above is the inventory for the customer to select. The list will get shorter as the customer makes decisions. The program provides two functionalities to help customers find the cars of their choice. The first function is the ability to view the table in different ways. For example,

viewing the table sorted by make. The second function is to select some rows from the table based on values, such as a price in the $10,000 +/- $2,000 or a model equal to 'SUV.' The program will let the user use these two functions in any order. In the end, a shortlist will be printed.

A sample execution is provided at the end of this file.

[3] **Requirements:** You are required to write the following functions as specified.

- Print_table(table): Input a table of cars and print the table nicely, as shown below. No return value. (Approximately 5 lines of code)
- Print_menu(word): Print a menu for the user to choose. A parameter is used to print two slightly different menus in one function. The function returns an integer. (Approximately 8 lines of code)
- Sort_table(table): Ask the user to sort the table and call the sorting function. No return value. (Approximately 3 lines of code)
- Choose_car(table): The function takes the table and selects some rows based on the user's interactive input. It returns a new table of the selected rows. The main function should take the new table and update the original table. (Approximately 15 lines of code)

The last function is slightly longer than the others. The reason is due to the complexity of dealing with strings and numbers. For string comparisons, the comparison must be exact except for the cases. For integers, we allow a range specified in the list called plus_minus. The main function is about 12 lines long (not a requirement, just FYI).

[4] **Output**: A sample output is given below. You should test for all possible input.

[5] **Deadline**: 11:59 pm, Wednesday, April 5, 2023

```
Choose from the list:

Make       Model    Year Mileage   Price
Nissan     Truck    2013   31621   14684
Chevrolet  Sedan    2014   62521   14875
Kia        SUV      2012   72961   13770
Kia        Sedan    2023   36792   19632
Toyota     Van      2012   25164   14248
Kia        Van      2012   43548   14065
BMW        Sedan    2011   72319   18277
Nissan     Van      2013   29669   14703
Chevrolet  Sedan    2011   26331   13737
Kia        Van      2019   92408   17076
Kia        Sedan    2011   92819   13072
Chevrolet  Truck    2023   11603   19884
Honda      Van      2023   10498   19895
Kia        Sedan    2014   58395   14916
Kia        Sedan    2015   69531   15305
Toyota     Van      2022   76451   18735
Hyundai    SUV      2013   75160   14248
Audi       Van      2021   18232   23818
Mercedes   Sedan    2010    6946   13431
Audi       Sedan    2021   47030   23530

Enter S to Sort, C to Choose, Q to End: s
1 - Sort based on Make
2 - Sort based on Model
3 - Sort based on Year
4 - Sort based on Mileage
5 - Sort based on Price
0 - Exit
       Select a field to Sort: 2

Choose from the list:

Make       Model    Year Mileage   Price
Kia        SUV      2012   72961   13770
Hyundai    SUV      2013   75160   14248
Chevrolet  Sedan    2014   62521   14875
Kia        Sedan    2023   36792   19632
BMW        Sedan    2011   72319   18277
Chevrolet  Sedan    2011   26331   13737
Kia        Sedan    2011   92819   13072
Kia        Sedan    2014   58395   14916
Kia        Sedan    2015   69531   15305
Mercedes   Sedan    2010    6946   13431
Audi       Sedan    2021   47030   23530
Nissan     Truck    2013   31621   14684
Chevrolet  Truck    2023   11603   19884
Toyota     Van      2012   25164   14248
Kia        Van      2012   43548   14065
Nissan     Van      2013   29669   14703
Kia        Van      2019   92408   17076
Honda      Van      2023   10498   19895
Toyota     Van      2022   76451   18735
Audi       Van      2021   18232   23818

Enter S to Sort, C to Choose, Q to End: s
1 - Sort based on Make
2 - Sort based on Model
3 - Sort based on Year
4 - Sort based on Mileage
5 - Sort based on Price
0 - Exit
       Select a field to Sort: 1
```

```
Choose from the list:

Make       Model    Year Mileage   Price
Audi       Sedan    2021   47030   23530
Audi       Van      2021   18232   23818
BMW        Sedan    2011   72319   18277
Chevrolet  Sedan    2014   62521   14875
Chevrolet  Sedan    2011   26331   13737
Chevrolet  Truck    2023   11603   19884
Honda      Van      2023   10498   19895
Hyundai    SUV      2013   75160   14248
Kia        SUV      2012   72961   13770
Kia        Sedan    2023   36792   19632
Kia        Sedan    2011   92819   13072
Kia        Sedan    2014   58395   14916
Kia        Sedan    2015   69531   15305
Kia        Van      2012   43548   14065
Kia        Van      2019   92408   17076
Mercedes   Sedan    2010    6946   13431
Nissan     Truck    2013   31621   14684
Nissan     Van      2013   29669   14703
Toyota     Van      2012   25164   14248
Toyota     Van      2022   76451   18735

Enter S to Sort, C to Choose, Q to End: c
1 - Choose based on Make
2 - Choose based on Model
3 - Choose based on Year
4 - Choose based on Mileage
5 - Choose based on Price
0 - Exit
       Select a field to Choose: 2
Enter the value to choose: sedan

Choose from the list:

Make       Model    Year Mileage   Price
Audi       Sedan    2021   47030   23530
BMW        Sedan    2011   72319   18277
Chevrolet  Sedan    2014   62521   14875
Chevrolet  Sedan    2011   26331   13737
Kia        Sedan    2023   36792   19632
Kia        Sedan    2011   92819   13072
Kia        Sedan    2014   58395   14916
Kia        Sedan    2015   69531   15305
Mercedes   Sedan    2010    6946   13431

Enter S to Sort, C to Choose, Q to End: c
1 - Choose based on Make
2 - Choose based on Model
3 - Choose based on Year
4 - Choose based on Mileage
5 - Choose based on Price
0 - Exit
       Select a field to Choose: 5
Enter the value to choose: 14000

Choose from the list:

Make       Model    Year Mileage   Price
Chevrolet  Sedan    2014   62521   14875
Chevrolet  Sedan    2011   26331   13737
Kia        Sedan    2011   92819   13072
Kia        Sedan    2014   58395   14916
Kia        Sedan    2015   69531   15305
Mercedes   Sedan    2010    6946   13431
```

```
Enter S to Sort, C to Choose, Q to End: s
1 - Sort based on Make
2 - Sort based on Model
3 - Sort based on Year
4 - Sort based on Mileage
5 - Sort based on Price
0 - Exit
      Select a field to Sort: 3

Choose from the list:

Make       Model    Year Mileage   Price
Mercedes   Sedan    2010    6946   13431
Chevrolet  Sedan    2011   26331   13737
Kia        Sedan    2011   92819   13072
Chevrolet  Sedan    2014   62521   14875
Kia        Sedan    2014   58395   14916
Kia        Sedan    2015   69531   15305
```

```
Enter S to Sort, C to Choose, Q to End: q

Choose from the list:

Make       Model    Year Mileage   Price
Mercedes   Sedan    2010    6946   13431
Chevrolet  Sedan    2011   26331   13737
Kia        Sedan    2011   92819   13072
Chevrolet  Sedan    2014   62521   14875
Kia        Sedan    2014   58395   14916
Kia        Sedan    2015   69531   15305
```