CS 115 - Introduction to Programming in Python

Lab Guide 08

Lab Objectives: Searching and Sorting.

a) Download the class `Stock` and the data file from Moodle. Data file contains the stock name, count, price and minimum stock level.

b) Update the class to include the following:
   a. Add the `__lt__` method according to the following: a stock is less than another if the `count` is less than the other's `count`. If their counts are equal, their stock names will be checked.

c) Create a class, `Stocklist`, with the following data members and methods. Note all data members and class variables should be private (__).

Data Members:

- `name`: stores the name of the Stock List
- `city`: stores the city of the stocklist
- `stock_list`: stores a list of Stocks.

Methods:

- `__init()__:` initializes the name and city passed as parameters. Initialize an empty stock_list.
- `get_stock():` returns the Stock in the list with the index passed as a parameter.
- `get_num_stocks():` returns the number of Stocks in the stock_list.
- `add_stock():` takes a stock as a parameter and adds the Stock to the stock_list. `bubbleSort():`sorts the stock_list by ascending (lowest to highest) count. If counts are equal names will be checked (__lt__ method will be used).
- `linear_search():` searches all stocks in the list and displays the quantities and names of stocks necessary to satisfy the minimum stock level. Uses a ***recursive linear search*** algorithm. See sample run for the format.
- `repr()__:` returns a string representation of a Stocklist object. See the sample run for the format.

d) Create a script which includes the following:

- **create_stocklist():** takes a filename as a parameter, and adds all Stocks in the file to a StockList and returns it. The first line of this file has the name and city of the Stocklist.

- **Your script should do the following:**
  - Create a Stocklist using **create_stocklist()** function.
  - Sort the list using **bubbleSort** and display the sorted list.
  - Display the quantities and names of all Stock items to be ordered whose count is less than their minimum stock level using **linear_search** method.

Sample Run :

```
Bubble sorted stocklist:
Biscuits            Ankara

Name: gofret
Count: 4
Price: 3.0
Minimum Stock Level: 5

Name: benimo
Count: 5
Price: 4.0
Minimum Stock Level: 3

Name: hayley
Count: 5
Price: 3.0
Minimum Stock Level: 12

Name: simit
Count: 5
Price: 5.25
Minimum Stock Level: 10

Name: metro
Count: 7
Price: 2.75
Minimum Stock Level: 10

Name: toblerone
Count: 7
Price: 16.75
Minimum Stock Level: 15

Name: burcak
Count: 11
Price: 6.75
Minimum Stock Level: 15

Name: sokokrem
Count: 12
Price: 2.5
Minimum Stock Level: 10

Name: albeni
Count: 13
Price: 2.5
Minimum Stock Level: 12
```

```
Name: damak
Count: 13
Price: 6.75
Minimum Stock Level: 10



Name: kraker
Count: 13
Price: 5.25
Minimum Stock Level: 10

Name: twix
Count: 13
Price: 3.75
Minimum Stock Level: 15

Name: biskrem
Count: 15
Price: 4.0
Minimum Stock Level: 21

Name: nutella
Count: 23
Price: 21.75
Minimum Stock Level: 20

Name: kakaolu
Count: 33
Price: 21.75
Minimum Stock Level: 15


Stocks that must be ordered:
1 quantities of gofret must be ordered
7 quantities of hayley must be ordered
5 quantities of simit must be ordered
3 quantities of metro must be ordered
8 quantities of toblerone must be ordered
4 quantities of burcak must be ordered
2 quantities of twix must be ordered
6 quantities of biskrem must be ordered
```