NYU, Tandon School of Engineering

CS-UY 1114 Introduction to Programming and Problem Solving — Spring 2017

Homework #8

Submission instructions:
1. You should **<u>not submit</u>** this assignment.
2. Use the questions here for practicing dictionaries.

Problem: In-N-Out of the MTA System

In this problem, we will have a brief introduction to how data structures in computer science work. If you've ever used any transit app to get directions in the NYC subway system, you probably were given either direct subway routes or directions to transfer at specific stops to other subway lines. These apps use data listing all the subway stations on each subway line and data about which stations have transfers from which line to which other line(s). Don't worry about details of how these apps work. In this homework, you'll write a program that deals with one aspect of such apps -- looking into the subway system with a Python program that reads and analyzes data from a given data file similar to those supplied by the MTA.

Your task is to design and write a program that will repeatedly ask the user for a train line until he/she enters 'done'. For each train line input the program should print out the stops of that train.

Follow this format (This is sample data and your actual data may not represent as such):

Please enter a train line, or 'done' to stop: 1
1 line: Van Cortlandt Park – 242 St, 238 St, 231 St, Marble Hill – 225 St, 215 St, ...

Please enter a train line, or 'done' to stop: A
A line: Inwood - 207 St, Dyckman St, 190 St, 181 St, 175 St, 168 St, 145 St, 125 St, 59 St - Columbus Circle, 42 St - Port Authority Bus Terminal, ...

Please enter a train line, or 'done' to stop: done

You are supplied with a file in '.csv' format, where the first line in the file outlines the columns. **One important thing to know about this data file is that the first character in each stop_id field indicates the train line**. For example, if the stop_id is 221S it means that this is a stop of the 2 train (since 2 is the first character in **2**21S).

Notes:

1. Before you start coding, take a look at this file and make sure you understand which columns are relevant and how subway stops are represented. You might want to open the file with a spreadsheet program, such as Excel.
2. It is up to you to decide which functions you'll need to write, how to read in the data and what data structures to use.
3. Make sure that when you print the stations, you are NOT printing out a list. That is there should be **no** brackets in the output; follow the output example above closely.
4. Make sure that there are no duplicates in your data structure.

Hints:

1. You may want to maintain a dictionary, in which the train lines are the keys. Each train line is mapped to a list of its stops. I.e. The value of a train line key would be a list of stops.
2. In order to avoid duplicates, you may want to use the **in** and **not in** operators.
   When applied on a dictionary, these predicates check if the dictionary contains a key.

   ```
   For example:
   >>> dictionary = {'Jan':1, 'Feb':2, 'Mar':3}
   >>> 'Feb' in dictionary
   True
   >>> 'Apr' in dictionary
   False
   >>> 2 in dictionary
   False
   >>> 'Feb' not in dictionary
   False
   >>> 'Apr' not in dictionary
   True
   ```