Liang, section 14.6, pp. 487-492

Dictionaries are an incredibly powerful tool in Python. With dictionaries you can quickly and easily organize massive data sets and perform deep analysis. As a reminder: "*A dictionary is a container object that stores a collection of key/value pairs. In enables fast retrieval, deletion, and updating of the value using the key.*" [1]

Update your repo if necessary (using `git pull`) and copy all the files in the following directory to a location of your choosing: `~/repo201/classwork/cw34/`. This directory contains a template file called (`midLookup.py`) to get you started.

Write a Python program that takes one input from the command line: A file of alpha codes and names, separated by semicolons. The program should open the file and create a dictionary with all the alpha codes as keys, and for each alpha code the associated name will be the value. The program should then prompt the user for an alpha code. If the alpha code exists, print the associated name. If the alpha code does not exist, print "`Alpha Code Not Found.`" Repeat this until the user enters an empty string (return by itself) when prompted for an alpha code.

There is a file in `cw34` called `mids.txt` you can use for testing. It contains 50 entries, which are printed on the back page of this handout (*displayed in two columns to fit on the page*). To help you visualize what the dictionary will look like, here's an example with two entries:

```
{ '216617' : 'Kennith K. Lintag', '217647' : 'Jody Y. Knoke' }
```

You may assume that the entered file exists and that alpha codes are unique (no duplicates). You don't need to perform exception handling now, but you can add it later if you wish.

A compiled version of this program is available as:

`~/repo201/classwork/cw32/lookupMid <-` to distinguish it from `midLookup`, which you're writing.

[1] Liang p. 487

```
216617;Kennith K. Lintag
217647;Jody Y. Knoke
216322;Jamison S. Yenzer
212588;Tana S. Helsabeck
217865;Jonna K. Cooperwood
212174;Sammie Y. Pollara
213228;Florene P. Stiller
214540;Torie M. McTiernan
219071;Garfield W. Lozzi
211688;Murray S. Vandriel
218206;Beatris N. Senge
217485;Foster M. Heraty
218446;Aldo H. Saluga
218544;Rodger Z. Romane
212202;Blaine M. Klefstad
211378;Lizbeth R. Markworth
215985;Burl Z. Harger
215039;Rocco K. Romigh
216101;Theodore X. Jeleniewski
216770;Matha K. Derus
212085;Vance Q. Thorsell
219152;Gregory T. Castaneira
211462;Melodee M. Florez
215236;Rolanda F. Baisa
214585;Cornell S. Zaucha
214036;Bernie B. Kamrad
213243;Bettyann L. Maginness
219953;Sheila U. Quang
211720;Diego C. Gandolfo
215571;Lorena F. Paese
219738;Manual F. Rejman
211667;Madelyn G. Ruscin
216245;Gerard V. Garred
213409;Tanika E. Almeida
218069;Karl R. Jeweii
213996;Cara U. Shaak
210084;Harrison P. Hyter
219584;Erica O. Broaddrick
210918;Lai Z. Anseth
218860;Bud K. Dumitru
210228;Joesph Y. Petrash
219603;Floy A. Quemada
213873;Kim D. Zul
215441;Aurea W. Pruszynski
215385;Jean U. Morgera
213303;Lois D. Sellars
219326;Suzie E. Lalima
219212;Lenore D. Faulman
213671;Tonita B. Chevez
216688;Leia R. Chiazza
```