CS1117: Exercise Sheet Semester 2 Week 5

This exercise sheet examines selection sort but also provides further practice on dictionaries and files. The dictionary question will take time and testing (either through print statements or debugging) to get right. <u>Please ensure that you take the time to figure this out by yourself if the time exceeds that taken in lab. Make use of the supports available to you if you need additional help.</u>

Question 1 (Dictionaries & files):

<u>Background:</u>

You are tasked with analysing some data on virus infections at different locations. You are given a data file (see canvas, **occurences.txt**), where on each line it states the name of a location followed by seven values (integers) that indicate the **cumulative** number of cases of a disease found at that location over the first seven days, respectively. The values on each line are separated by commas, but there can be an arbitrary number of spaces between each value and each comma.

```
Kerry , 1, 1, 1, 1, 1, 1, 1
Cork,1 ,2 ,3 ,4 ,5 ,6 ,7
Limerick   ,1,1, 2,   3,     5,      8,       13
```

This file has data for three locations and each location has seven values representing the cumulative (total) number of cases of the infection at that location over seven subsequent days. For example, in Kerry, there was 1 case found on the first day, and then no new cases for the next six days (so the cumulative number of cases remained 1 throughout all the days). On the other hand, in Cork, on the first day one new case was detected and, on each subsequent day (for the next six days), one new case was detected each day. As a result, the cumulative number of cases increases by one each day. You can assume that the location names in the file are all unique. In other words, you'll never get two lines that have the same location name at the beginning.

<u>Part A: Reading the file into a dictionary</u>

Write a function **`def load_data(filename).`** This function takes in the name of a datafile (string), which has the format of the data file described above. The function should return a dictionary in which the keys are the names of locations in the data file, and the value associated with each key is a list of the (integer) values presenting the cumulative number of infections at that location. For example, if you were passed the filename **'occurences.txt'**, your function should return the following dictionary:

```
{
   'Kerry': [1, 1, 1, 1, 1, 1, 1],
   'Cork': [1, 2, 3, 4, 5, 6, 7],
   'Limerick': [1, 1, 2, 3, 5, 8, 13]
}
```

Note that the function **strip()** applied to a string is useful both for removing the "newline" character (**\n**) at the end of a line in a file as well as removing extra spaces at the start/end of a string. So, if we had the string:

s1 = ' example of stripping spaces '

and we called:

s2 = s1.strip()

then **s2** would have the value **'example of stripping spaces'** (without the spaces at the start/end of the string).

Part B: Calculating the number of infections per day

Once you have the load_data function working, the second part of this problem requires you to write the function **def daily_cases(cumulative)**. This function takes in a dictionary of the type produced by the **load_data** function (i.e., keys are locations and values are lists of seven values representing cumulative infection numbers). The function should return a new dictionary in which the keys are the same locations as in the dictionary passed in, but the value associated with each key is a list of the seven values (integers) presenting the number of new infections each day at that location. So, given the dictionary shown above (produced from the file **'occurences.txt'**), your function should return the dictionary shown below.

```
{
  'Kerry': [1, 0, 0, 0, 0, 0, 0],
  'Cork': [1, 1, 1, 1, 1, 1, 1],
  'Limerick': [1, 0, 1, 1, 2, 3, 5]
}
```

Note that Kerry, for example, had 1 case the first day, but then no additional new cases on any subsequent days. Cork however has one new case every day.

Hint: For every day, except the first, you can determine the number of new cases by subtracting the cumulative number of cases on the day before from the cumulative number of cases on that day.

Question 2 (Selection Sort):

You are tasked with writing a function to conduct a selection sort on a list of numbers. You have been provided with a template/skeleton .py script on Canvas. Complete this to solve the program. Refer to your lecture notes for guidelines on how selection sort works. You should print out the list after every iteration/round/cycle of the list.