MA305 – Lab #5.
Reading (from a file) and Printing Lists
Date: 10/22/2018

In this lab you will learn how to use Python functions ***`readline()`*** *and* ***`readlines()`*** *to read data from a file, do some calculations, and print the results neatly using* ***`format`*** *statement.*

---

1. First, let us read a "self-terminating" input file. Create a directory ∼`/MA305/Lab5`, and in this directory create a data file (`dat5.txt`) containing the following six lines.

```
  date    index    x        y
20050212    8    15.30     5.40
20060212   12    14.50     9.52
20070212    8    16.30     6.40
20080212   12    13.50     8.52
20090212   12    12.50    -7.52
0    0     0.00     0.00     : date=0 to terminate input
```

Such a file is a typical example of how measurements are recorded. Each line represents a measurement of two quantities x and y, and we may be interested in their averages, say. The first value of our data is an integer (date), recording the date of the measurement. The second integer (index) may encode some other information. The next two (reals) are the measured values of x and y, which we view as components of two arrays x_i and y_i, whose averages we want to find.

2. Write a code in Python `lab5.py`, which reads this file, and stores the values in appropriate variables. Your code should print out how many lines it read from the data file, then the x_i values in one column, and the y_i values in a second column, neatly lined up. Label the columns appropriately.

a. First open a datafile to read data from.

```
f = open('dat5.txt','r')
```

b. The first line of the `dat5.txt` file is just a label, but it still must be read! A plain `readline()` statement will do it. Read the line as a string (`line`) and print it.

```
line=f.readline()
print(line,end='')  # end='', forces not to print an extra blank line
```

c. To read the remaining lines, use a `while` loop. The last line in `dat5.txt` has 0's, as signal to terminate further input. So the value of 'date' should be checked after each `readline()`. For this, you need to `split` the string `line` read into a list `Line`, and `break` the `while` loop if `int(Line[0])==0`. Once you are done reading the date file, close it.

```
while True:
    line = f.readline()          # line is a string
    Line = line.split()          # Line is a list
    if int(Line[0])==0:
        break
    print(Line)
f.close()
```

d. To print out all the x_i values in one column, and all the y_i values in another, side-by-side, neatly lined up, place the following statement inside the `while` loop.

```
print('\t {0:-5.2f} \t {1:-5.2f}'.format(float(Line[2]),float(Line[3])))
```

e. Use a counter `n +=1` inside the `while` loop to get the number of lines read and print it after the loop is terminated.

```
print(n,"lines from 'dat5.txt' are read for calculations!")
```

3. Now find (and print out!) the sum of x_i and y_i.

a. You can calculate sum of x_i and y_i, within the `while` loop also, but we may need these for later computations. So, we store `float(Line[2])` and `float(Line[3])` values in the lists `x` and `y`. For this, define two empty lists `x` and `y` before the `while` loop.

```
x=[]; y=[]
```

and `append` the values to the lists `x` and `y`, inside the `while` loop, as they are read.

```
x.append(float(Line[2]))
y.append(float(Line[3]))
```

Note that 'date' (`int(Line[0])`) and 'index' (`int(Line[1])`) have no bearing on our calculations so we don't need to keep their values.

b. Now, calculate the sum of the values in the list `sx` and `sy` using a `for` loop.

```
print('==========================')
sx = sy = 0
for i in range (n):
    sx += x[i]
    sy += y[i]
print('  Sum : {0:-5.2f} \t {1:-5.2f}'.format(sx,sy))
print()
```

4. Now, you will explore another method of reading the data file. You can also read the whole file with a single command `readlines()`. Do the following steps:

a. Copy the data file `dat5.txt` to `dat5a.txt` and delete the last line of the file `dat5a.txt`. That is we will work on the following data file.

```
  date    index    X        Y
20050212   8    15.30     5.40
20060212  12    14.50     9.52
20070212   8    16.30     6.40
20080212  12    13.50     8.52
20090212  12    12.50    -7.52
```

b. Create a new file `lab5a.py` and type the following line and run it.

```
f = open('dat5a.txt','r')
f.readline()
data=f.readlines()
print(data)

for line in data:
    Line = line.split()
    print(Line)
f.close()
```

You should get the same value in the list `Line` as in your `lab5.py` code?

5. Adapt your code `lab5a.py` to compute (and print!) the average values and the dot product of x and y.

$$\bar{x} = \frac{1}{n}\sum_{i=0}^{n-1} x_i, \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n-1} y_i, \qquad x.y = \sum_{i=0}^{n-1} x_i y_i$$

6. I/O Redirection. Import the `sys` module at the beginning of the code `lab5a.py`, comment the line `f = open('dat5a.txt','r')` and define `f = sys.stdin`,

```
import sys

#f = open('dat5a.txt','r')
f = sys.stdin
```

and run it with the following command.

```
$ ./lab5a.py < dat5.txt > out.txt
```

7. Make a typescript file showing your code (*lab5a.py*), its execution, and the output (*out.txt*), and as usual make it readable (clean it up) and *submit your work through Canvas. No Email!*