Python Exercises – Series 3

Iterations

1. Write the for loop below with a while loop:

```
for k in range(9,0,-1):
        print("k=", k)
```

2. Write a function that returns the smallest and largest element in a list (without using the built-in Python functions and methods such as sort, min and max!). Note that the function returns two values!

Nested lists

When we introduced lists, we saw that they can be heterogeneous (meaning that they can contain elements of different types). Since lists are also a type in Python, it is possible to store lists in lists, which we refer to as “nested lists”.

1. Write a function that takes a nested list as argument, and returns the length of the longest list. For example: [[1,2,3],[4,5],[6,7,8,9],[10]] → 4 since the longest list is [6,7,8,9] and its length is 4.
2. Write a function that has a nested list as an argument and calculates the averages of the j^{th} elements in each sub-list. Beware: the length of the list of averages should be as long as the longest sub-list! For example [[1,2,3], [4,5], [6,7,8,9], [10]] should return [5.25, 4.67, 5.5, 9.0] (calculated as follows [(1+4+6+10)/4, (2+5+7)/3, (3+8)/2 ,9/1]).

A practical example

We will try to solve a practical problem by using Python. An object is thrown from earth with a specific velocity and at a certain angle. From our high school physics class, we recall computing quantities such as the distance travelled of the thrown object or what the maximum height will be of the object. In this exercise, we will use Python to simulate the object's trajectory.

We start with writing a function to which we provide the following information:

- The angle **α** with the surface of the earth
- Starting speed $\mathbf{v_0}$
- The calculation frequency **f** with which we can execute the simulation

The function returns the following:

- Two lists of coordinates **x** and **y** with the location of the object at subsequent moments in time. The time between two calculations is determined by **f**. Note that the function returns two lists!

Here is the equation that you can use to determine a location with coordinates x and y at time t:

$$\begin{cases} x = v_0 \cos\alpha \ t \\ y = v_0 \sin\alpha \ t - \dfrac{gt^2}{2} \end{cases}$$

Your function should determine the subsequent values **[x,y]** by increasing **t** in steps, where the number of steps per second is given by means of the calculation frequency **f**. Calculations need to stop when the object touches the earth again.

Tip: the "math" module can be used to calculate sin() and cos() (using radians!). You will have to import this library first.

For a graphical representation of the simulation you can plot the values **[x,y]** in Python by using the functions provided in the matplotlib/pylab library. You can display your calculated values nicely with the following code:

```
from pylab import figure,plot,title,xlabel,ylabel,show

x,y = projectile_motion(alfa, v0, 0.01)


figure()
plot(x,y)
title('Projectile motion')
xlabel('Distance (m)')
ylabel('Height (m)')
show()
```

File I/O

You are given a file in which the grades for the ‘scripting languages’ exam are stored. Download the file ‘data.csv’ from the Pointcarre website. Each line in this file contains the information of 1 student in the following format: NAME;firstname;classname;grade, sorted on NAME.

1. Write a function that reads the file and returns the data as a nested list. Each element in that list is a list that contains the information of a student. Tip: you can use the function method split(*separator*) to break up a string into a list of substrings (the parameter *separator* determines how to split the string).

   Reading files is quite easy in Python. You can read and process all the lines in a text file one by one with a **for loop**:

   ```
   f = open(‘data.csv’,’r’)
   for line in f:
      # do something with that line, e.g. print
      print(line)
   f.close()
   ```

2. Write a function that looks up the grades of a student in the list created in question 1 and returns it. Besides the list itself, this function is given a name and a first name and returns a number (i.e. the grade of the student).
3. Write a function that can alter the grades of a certain student in the list and write a second function that can store this list in a file.
   Writing to a file is done as follows:

   ```
   f = open(‘output.csv’,’w’)
   for …:
      # do something for each sublist
      f.write(variable)
   f.close()
   ```

4. Write a function that takes the list of question 1 as an argument and returns 6 different lists, one per group (IA-A, IA-B, IR-A, IR-B, IR-C, IR-D).
5. Write a function that takes a student list as an argument and returns a list of grades as **float**s.
6. Write a function that creates a histogram of the exam results: i.e. count the number of times the number 0 occurs, how many times 1 occurs, etc. until 20. Normalize this histogram by dividing each counted value with the total number of elements in the list. Print the histogram to the shell by putting the percentage of students that received a particular grade in front of that grade.

   Note: You can also nicely plot histograms by using the pylab library in the following way:

```
from pylab import hist,show,figure,title

figure()
n, bins, patches = hist(gradelist, 20, range=(0,20), normed=0, histtype='stepfilled')
title('Group x')
show()
```