Session no. 5

Learning goals: Using functions in python. Reinforcement of loops. More input/output.

1. Observe the following program. It uses two functions: one to encapsulate the input operation, and another one to calculate and print the grade.

```
def ask_for_grades():
    exam1 = float(input("Grade 1. exam: "))
    exam2 = float(input("Grade 2. exam: "))
    exercises = float(input("Grade exercises: "))
    return exam1, exam2, exercises

def show_grades(test1, test2, exercises):
    final = test1*0.35 + test2*0.35 + exercises*0.3
    print("Grade final: {:.2f}".format(final))

grade1, grade2, grade3 = ask_for_grades()
show_grades(grade1,grade2,grade3)
```

2. The following program asks for new input and visualizes the grades until the user wishes to finish the program.

```
def ask_for_grades():
    exam1 = float(input("Grade 1ª exam: "))
    exam2 = float(input("Grade 2ª exam: "))
    exercises = float(input("Grade exercises: "))
    return exam1, exam2, exercises

def show_grades(test1, test2, exercises):
    final = test1*0.35 + test2*0.35 + exercises*0.3
    print("Final grade: {:.2f}".format(final))


repeat = "yes"
while (repeat != "no"):
    grade1, grade2, grade3 = ask_for_grades()
    show_grades(grade1,grade2,grade3)
    repeat = input("keep going? (yes/no) ")
    repeat = repeat.lower()
else:
    print("\nEnd of program")
```

3. Modify the previous program such that any negative input terminates implicitly the input loop (clearly, this last entry should not participate in the average).

4. Write a program to compute the Collatz sequence of a positive integer number a_0, where for such a sequence we have $a_{n+1} = 1/2 \cdot a_n$, if a_n is even, and $a_{n+1} = 3 \cdot a_n + 1$, if a_n is odd. The sequence terminates whenever we arrive at the number 1. For instance, for $a_0 = 3$ we obtain the sequence: 3, 10, 5, 16, 8, 4, 2, 1.

   Note: the Collatz conjecture is that no matter what value of n, the sequence will always reach 1. This conjecture remains has not been proven yet[1].

5. We now that in Spain, height of young men follows a normal distribution (mean 172 cm and standard deviation 8.1 cm). We want to calculate the percentage of this population that is taller than 185 cm. To obtain this percentage we will use the Montecarlo method: we create data points following the given distribution, and we count how main points are above the specified height (185 cm). Write a function to compute this percentage.

   Tip1: we will consider that 10000 points is a big enough set of data to have an accurate estimation.

   Tip2: the `random` library provides a function called `gauss` that can be used to generate the data. See python documentation[2]

6. Modify the previous program so that the mean, standard deviation and height are asked to be entered by the users.

[1] https://en.wikipedia.org/wiki/Collatz_conjecture
[2] https://docs.python.org/3/library/random.html