Creating a brief Python program: "Distances in marathon training"

General directives, requirements

READ EVERYTHING

This assignment may be done in teams of two or individually. You will have to create a Python program. You will have to submit your program code (the .py file only) via the Canvas course website.

You will have to CREATE YOUR OWN GROUP in Canvas, (in the set "Groups for Assignment 1") of up to 2 people. GROUPS WITH ONE PERSON only NEED to be created as well in the case that you would prefer to work individually.

The submission deadline is: Tuesday May 31 noon (12:00 pm). You are encouraged to submit earlier as well.

OBJECTIVES: You are expected to practice and learn about:

Interpreting and understanding the characteristics of a brief problem description and requirements; implementing in Python a solution to a basic problem; and practicing submitting via Canvas.

Within the program:
Basic printing to the user; asking values to the user (without validation of correctness for now); creating and assigning values to variables (with reasonable names); doing basic arithmetic calculations; and including appropriate comments.

DESCRIPTION OF THE PROBLEM:

You are asked to write a Python program helpful for two runners who train in countries using different measure systems. The program will be given the information of where each of the two runners finished running, and the program will then inform the difference of distances covered by the two runners.

Both runners will be assumed to run from a 0 (zero) point. Distances will be measured in two systems: using the *Metric system* or the *Imperial system*. In the metric system, the locations (i.e. distance from 0) will be expressed as three numbers: kilometers (km), meters (m) and centimeters (cm) or as one unique number in centimeters. In the imperial system, the locations will be expressed as three numbers: miles(mi), yards (yd) and feet (ft) or as a unique number in feet.

You are asked to implement a basic version of a distances calculator only allowing calculations in specific units and in a very specific order as described next.

DETAILED DESCRIPTION

The program will first ask the user the location where the first runner arrived, expressed as three INTEGER numbers: kilometers, meters and centimeters, asking one number after the other [Note: the function ***input(…)*** followed by a conversion function to integer will be used to ask these numbers to the user, see Hints below].

Then the program should ask the user the information associated to the second runner, expressed as integer numbers representing miles, yards and feet (again asking this data as three numbers, one after the other).

Then the program should show to the user (print) the following results:

a) *First result*: The number of centimeters (cm) that the second runner covered
b) *Second result*: The number of centimeters (cm) representing the difference between what the two runners covered
c) *Third result*: the same result as in the previous part (i.e. difference between the two runners) but expressed with three INTEGER numbers as km, m, and cm so that there are less than 1000 m and less than 100 cm (since 1 km is equal to 1000 m and 1 m is equal to 100 cm)

The program as a whole will DO THIS PROCESS ONLY ONCE, that is, it will ask the user for the 6 numbers and then print the three results based on those numbers and then the program ends. To test the program with different values the user will have to run the program several times.

REQUIREMENTS AND ASSUMPTIONS

a. Assume that the user types exactly what he/she is asked to type with no mistakes and following the problem restrictions. That is, the program does NOT need to validate that the user types correctly.

b. The sample runs shown are just providing a couple examples. You may want to test with these numbers but also other cases. In particular, test easy and borderline cases to make it easy for you to double check correctness.

c. Your program should show analogous information as in the sample runs, although not necessarily identical messages. You will find it useful to add extra printing to best follow the execution of the program. This is referred to as “tracing” the program execution.

d. At the beginning of your program file you have to INCLUDE COMMENTS. (You may include more comments in the body of the program as well). The comments should indicate:

- the name/s of the author/s of the program
- the date of your last revision
- (continues in the next page)

- the approximate number of hours (you may include half hours also) that you spent working on this exercise. If you are working in a two-group team provide the total number of hours spent between the two.
- A very brief description (one sentence or a brief paragraph is ok) of the program description

HINTS

e. To ask the user for integer numbers use the function input(…) and then convert to integer using the function int(…). These functions will allow you to assign the numeric value that the user types into a variable.(Note that for this assignment you will need to define several variables). The string between brackets in the input function is the prompt that the user will see when asked to provide the value. We will discuss about this function in labs and class in more detail. You can also read about it in the course texts.

   For example:

```
integer_variable = int(25.48)  will store 25 in integer_variable

other_integer_variable = int(input("Please type in a number: "))
will store the integer number corresponding to what the user typed.
```

f. To calculate the third result you do NOT need to use any loops. Recall how you can get the quotient of an integer division using the integer division operator (//) and the remainder in a division using the remainder (or modulo) operator (%). For example, recall that 23 % 3 produces the result 2, because 2 is the remainder from dividing 23 by 3.

g. If you need to obtain the absolute value of a number you can use the function abs(…). If you want to round a number to a certain number of digits you can use the function round(…)

h. You can consult about the pre-existing (or built-in) Python functions on line. Links to the Python libraries are included in the LINKS section in the course website.

Responsibility, benefits, group submission

i. The submission needs to be made by the group of up to 2 people (one single .py file per group). You may discuss general ideas about this assignment with more people but no code should be shared between different groups.

j. Being a group activity it is your responsibility and to your benefit to do your share.

End of assignment #1 description.