Creating a brief Python program: "Time goes by"

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 via the Coursys system.

Notice that to submit this exercise via Coursys you will need to CREATE A GROUP AND SUBMIT AS A GROUP, EVEN IF YOU WORK INDIVIDUALLY (in which case the group will only have one person, yourself).

The submission deadline is: Monday January 26, 11:59 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 submitting via Coursys.

Within the program:
Printing messages to the user, asking values to the user (without validation for now), creating and assigning values to variables (with reasonable names), doing basic calculations with integer numbers, and including appropriate comments.

DESCRIPTION OF THE PROBLEM:

You are asked to write a Python program which will calculate how much time passed between two dates in different formats.

Dates will be expressed as three numbers: Year, month, and day. The dates considered will be within this century and last century, hence years may have values between 1901 and 2099 inclusive. In this simplified version we will assume that months have 30 days (and hence all years have 360 days). For example: 1925,10,23 and 2015,1,17 are two valid dates. 2015,1,31 will not be valid. In any case we will assume that the user will provide data following these assumptions.

The program will first ask the user the first date as three numbers: year, month and day, asking one number after the other [Note: the function input will be used to ask these numbers to the user, see below]. Then the program will ask the user to provide the second date in the same format.

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

a) *First result*: The number of days that passed, for each date, since 1901,1,1
b) *Second result*: how many days have passed between the two dates.

c) *Third result*: the same result as in the previous part but expressed as number of years, months and days so that it is not more than 30 days and not more than 12 months.

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 results for those numbers and 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. Assume that the user will always type the earlier date first

c. The sample runs shown are just providing a couple examples. You may want to test these and other cases. In particular, test easy and borderline cases.

d. Your program should show analogous information as in the sample runs, although not necessarily identical messages.

e. At the beginning of your program code **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
- 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 is ok) of the program description

HINTS

f. To ask the user for integer numbers use the function input(…). This function will allow you to assign the value that the user types into a variable. The string between brackets provided to the function is the prompt that the user will see. We will discuss about this function in labs and class in detail. You can also read about it in the course texts. For example:

```
numeric_variable = input("Please type in a numeric value: ")
```

g. You do not need to use any loops for this assignment. Recall how in an integer division you obtain a quotient and a remainder. You may want to consider using the % (modulo or remainder) operator to help you calculate the third result. Recall that 23 % 3 produces the result 2, because 2 is the remainder from dividing 23 by 3.

SUBMITTING IN COURSYS

h. You will submit via Coursys. You will find links to the Coursys system from the course website, in the Links section. You may submit many times. The last version within the deadline is the one that will be marked.

i. **Before submitting**, one team member needs to first create the Coursys group and then invite the other member, whom in turn **needs to accept the invitation (also within Coursys)**.

j. To create a group in Coursys (which you will have to do even if you work individually), use the Actions menu, manage groups, to the left hand side. To accept an invitation, the invited member also has to do it through the action manage groups, joining the group.

k. Name the Coursys group including the id's (sfu email address up to @) of the group member/s and then the letter/number A1 (standing for Assignment 1).

   For example, if the two team members are Arthur Pendragon and Merlin Wizard, with email addresses apendra@sfu.ca and merlin@sfu.ca, the group should be named apendra_merlin_A1.

l. **Only one team member should submit the assignment** (after the group was created), which should be a .py file.

End of assignment #1 description.