Laboratory Exercise 02

Topics

1. Variables in Python
    a. Variable types
    b. The basic arithmetic operators
    c. String manipulation

Discussion

A. In what relationship are the assigned value, the variable and the data type?
B. Give an example of operations that can be performed on variables with assigned values corresponding to different types of data.
C. Consider the "+" operator. What happens if it is used between...
    a. Two variables with values of type `int`;
    b. One variable with value of type `int` and one with `value of type float`;
    c. Two variables with `str values`;
    d. One variable with a value of type int `and a variable with a value of type` str`.`

Exercises

Part 1 - Arithmetic operations

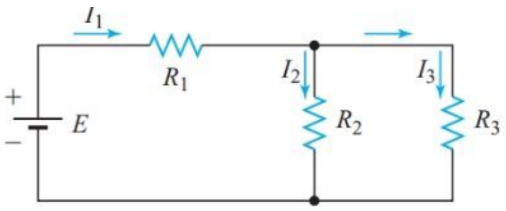
***Delivery**: For each of the following exercises, write a program in Python that responds to the requests indicated. Complete at least two exercises during the exercise, and the rest at home.*

02.1.1 Two numbers. Write a program that stores two integers in two constants defined in code, and then displays:

A. The sum;
B. The difference;
C. The product;
D. The average value;
E. The distance (i.e. the absolute value of the difference);
F. The maximum value (i.e. the greater of the two);
G. The minimum value (i.e. the lesser of the two).

Tip: Use the `max()` and `min()` functions defined in Python. They accept a comma-separated sequence of input values and return the maximum and minimum values of the sequence, respectively (for example, `max(10, 5)` returns `10`). [P2.4]

02.1.2 Resistances. Consider the following circuit.



Write a program that, starting from the resistances of the three resistors, input by the user, calculates the total resistance, using Ohm's law. [P2.38]

02.1.3 Figures. Write a program that stores in a constant a positive integer of five digits (therefore between 10000 and 99999), and displays the individual digits of which it is composed. For example, having the number `16384`, the program must display, on separate lines: `1 6 3 8 4`. [P2.16]

02.1.4 Hybrid car. Write pseudocode and related Python program that helps a person decide whether or not to buy a hybrid car. Program inputs should be:

I. the cost of a new car;
II. the estimate of the kilometers traveled in a year;
III. The estimation of the cost of fuel;
IV. Efficiency in kilometers per liter;
V. The estimate of the resale value of the used car after 5 years.

Calculate the total cost of ownership of the car for 5 years (for simplicity, do not take into account the cost of financing). To provide inputs to the program, search the web for realistic prices and consumption for two alternatives for buying a new car in the same price range: a hybrid model and a gasoline model. Run the program twice to compare the two alternatives, considering the current fuel price and the forecast of traveling 30,000 kilometers per year. [P2.10]

02.1.5 Electric Force. According to Coulomb's law, the electric force (expressed in Newtons) between two charged particles with charge, respectively, *Q1* and *Q2*, separated by a distance *r*, is

$$F = \frac{Q_1\,Q_2}{4\,\pi\,\varepsilon\,r^2}$$

where Farad / meter. Write a program that calculates and shows on screen the force relative to a pair of charged particles, allowing the user to choose the values $\varepsilon = 8.854 \times 10^{-12}$*Q1*, *Q2* (in Coulombs) and *r* (in meters). [P2.43]

Part 2 – String manipulation

***Delivery**: For each of the following exercises, write a program in Python that responds to the requests indicated. Complete at least two exercises during the exercise, and the rest at home.*

02.2.1 Characters. Write a program that stores a string in a variable and, starting from that variable, displays the first three characters of the string, followed by three periods, again followed by the last three characters. For example, if the string is initialized to the value "`Mississippi`", the program should display "`Mis... ppi`". What happens to the string is shorter than 6 characters? What if it's shorter than 3 characters? [P2.22]

02.2.2 Telephone number. The following pseudocode describes how to transform a string containing a ten-digit telephone number (such as "`4155551212`") into a more easily readable string, formatted in the U.S. style, such as "`(415) 555-1212`":

I. Take the string consisting of the first three characters and surround it with round brackets (this is the prefix, called "area code");
II. Concatenating the prefix with the string containing the next three characters, a hyphen, and the string consisting of the last four characters gives the number in the required format.

Translate this pseudocode into a Python program that stores a 10-digit phone number in a string, and then displays it in the format just described. [P2.33]

02.2.3 Alignment. Format the output of exercise **02.1.1** so that descriptions and numbers are aligned vertically, for example:

```
    Sum: 45

    Difference: -5

    ...         ...
```

[P2.5]

02.2.4 Emoji. According to data collected by the Unicode Consortium[1], the non-profit organization responsible for digitizing the world's languages, "tears of joy" () account for more than 5% of all emojis used (😂 the only other character that comes close is the ❤️). The top ten emojis used worldwide are: 🙏👍😂😘❤️😍🤣🥰😭😊 .
When exchanging messages on Telegram (or your favorite messaging app), what are the three emojis you personally use most frequently? Using the Unicode encoding information collected here[2], write a program that shows on the screen, for each of these three emojis:

I. the ranking position (rank);
II. the Unicode Number;
III. the Unicode Name;
IV. the emoji itself.

Format the output so that the information for each of the three emojis is collected on a line, and the different fields are aligned vertically.

02.2.5 Registrations. The matriculation of students of a University consists of two parts: a letter and a sequence of numbers. Write a program that, starting from two service tags, shows them on the screen in ascending order based only on the numerical part. Tip: Use the built-in functions of the Python language.

[1] https://home.unicode.org/emoji/emoji-frequency/

[2] https://unicode-table.com/