Laboratory 03

Topics

1. Elaborations and comparisons between values of type `int`, `float` and `str`
2. Boolean expressions
3. Logic decisions
4. Conditional constructs (`if`, `else`, and `elif`)
5. Nested blocks and instructions

Discussion

A. What are the possible syntax forms of a conditional construct?

B. Explain the similarities and differences between:
   a. conditional operators and Boolean operators;
   b. assignments and equality test;
   c. `and`, `or`, and `not`.

C. How do you define compound statements?

Exercises

Part 1 – Comparisons, relational and Boolean operators

***Task**: For each of the following exercises, write a program in Python that meets the given requirements. Complete at least two exercises during the lab, and the remaining ones at home.*

03.1.1 True or False. For each of the following pairs of values, perform an *equality* test, assign the result to a variable, and display it. Try to predict what the result of each test will be.

I. `1`, `1`;
II. `1`, `1.0`;
III. `2.0`, `sqrt(4)`;
IV. `'1'`, `1`;
V. `'hello'`, `'Hello'`.

03.1.2 Identikit of the string. Write a program that reads a string and displays the appropriate messages, after checking if:

I. it contains only letters;
II. it contains only capital letters;
III. it contains only lowercase letters;
IV. it contains only digits;
V. it contains only letters and digits;
VI. it starts with a lowercase letter;
VII. it ends with a point.

[P3.17]

03.1.3 Strings and substrings. DNA sequences are like long strings consisting of only four letters: `'A'`, `'C'`, `'T'` and `'G'`. Write a program that takes as input a "long sequence" of DNA of twenty characters and a "short sequence" of three characters and displays:

I. if the "long sequence" contains the "short sequence";
II. if yes:
   a. from which position of the "long sequence" the "short sequence" starts;
   b. how many times the "long sequence” contains the shorter substring.

[Chapter 3.8]

03.1.4 It’s equal. Describe the pseudocode corresponding to the following Python program:

```
x = 3.0

s == 'seven point five'
if x = 3.0:
    s == 'three point zero'
print(s)
```

What is the output of this program?

03.1.5 Matter of logic. Write a program that takes as input an integer number x and displays the following expressions on the screen, along with their truth values. Test the program with numerous values of x.

```
I.   x > 0 and x < 100
II.  x > 0 or x < 100
III. x > 0 or 100 < x
IV.  x > 0 and x < 100 or x == -1
```

[Chapter 3.7]

03.1.6 De Morgan. The De Morgan’s law makes it easy to apply the not to expressions containing and/or operators. In particular, this law has two expressions, one for the negation of expressions in and, and one for the negation of expressions in or:

`not (A and B)`	is equal to	`not A or not B`
`not (A or B)`	is equal to	`not A and not B`

Consider the following expressions, and for each of them apply De Morgan's law. Try to describe "in words" the intuitive algebraic meaning of each of the expressions.

Then write a program that takes as input an integer x and for each of the following expressions (which correspond to the negation of the expressions in exercise **03.1.5**) prints: the starting expression, the expression after applying De Morgan's law, and their truth value:

```
I.   not (x > 0 and x < 100)
II.  not (x > 0 or x < 100)
III. not (x > 0 or 100 < x)
IV.  not (x > 0 and x < 100 or x == -1)
```

[Special Topic 3.5]

Part 2 – Decisions

***Task**: For each of the following exercises, write a program in Python that meets the given requirements. Complete at least two exercises during the lab, and the remaining ones at home.*

03.2.1 Trends. Write a program that reads three numbers and displays the message "increasing" if they are in strictly increasing order, "decreasing" if they are in strictly decreasing order, and "neither" if they are in neither increasing nor decreasing order. "Strictly increasing" means that each value must be greater than the previous one (similar meaning has the term decreasing): the sequence `3 4 4`, thus, it is not "increasing". [P3.5]

03.2.2 Grades. Write a program that translates a letter grade entered by the user into the corresponding numerical grade and prints it. The letter grades are `'A'`, `'B'`, `'C'`, `'D'` and `'F'`, possibly followed by a `+` or `–` sign. Their numerical values are, in order, `4.0`, `3.0`, `2.0`, `1.0` and `0.0`. `'F+'` and `'F-'` grades do not exist. A `+` sign increases the numerical grade by 0.3, while a `–` sign decreases it by the same amount. The `'A+'` grade is equal to `4.0`. [P3.12]

03.2.3 Seasonal cycles. The following algorithm (already seen in exercise **01.1.2**) identifies the season (Spring, Summer, Fall or Winter, i.e., spring, summer, fall or winter, respectively) to which a date belongs, given as month and day.

```
If month is 1, 2 or 3
    season = "Winter"
Otherwise if month is 4, 5 or 6
    season = "Spring"
Otherwise if month is 7, 8 or 9
    season = "Summer"
Otherwise if month is 10, 11 or 12
    season = "Fall"
If month is divisible by 3 and day >= 21
    If season is "Winter"
        season = "Spring"
    Otherwise if season is "Spring"
        season = "Summer"
    Otherwise if season is "Summer"
        season = "Fall"
    Otherwise
        season = "Winter"
```

Take back the analysis of the algorithm and then write a program that, by implementing it, asks the user for a month and a day and, then, displays the season determined by this algorithm, verifying its correctness. [P3.20]

03.2.4 Leap years. A 366-day year is called a leap year and is used to keep the calendar synchronized with the Sun, since the Earth revolves around it once every 365.25 days or so. In reality, this number is not accurate, and for all dates after 1582 the Gregorian correction applies: usually years divisible by 4, such as 1996, are leap years, but years divisible by 100, such as 1900, are not; as an exception to the exception, years divisible by 400, such as 2000, are leap years. Write a program that asks the user for a year (greater than 1582) and determines whether it is a leap year using a single `if` construct and Boolean operators. [P3.27]

03.2.5 Grades again. Given the numerical values of the grades explained in exercise **03.2.2**, write a program that translates a number between `0.0` and `4.0` into the literal grade closest to it. For example, the number `2.8` (which could be the average of several grades) should be translated as `'B-'.` Resolve cases of equality in favor of the better grade: for example, `2.85` should be translated as `'B'`. [P3.13]

03.2.6 Taxes. Write a program that calculates taxes according to the following scheme. The program should acquire the value of the income from the user, and print the taxes due. It is not required to print intermediate steps. [P3.25]

for civil status "unmarried" and taxable income higher than	but not higher than	taxes are	of the sum in excess of
$ 0	$ 8000	10%	$ 0
$ 8000	$ 32 000	$ 800 + 15%	$ 8 000
$ 32 000		$ 4 400 + 25%	$ 32 000
for civil status "married" and taxable income higher than	**but not higher than**	**taxes are**	**of the sum in excess of**
$ 0	$ 16 000	10%	$ 0
$ 16 000	$ 64 000	$ 1 600 + 15%	$ 16 000
$ 64 000		$ 8 800 + 25%	$ 64 000

03.2.7 Conversions. Write a program for the conversion of units of measurement. It asks the user for: the starting unit of measurement (choosing from: `ml`, `l`, `g`, `kg`, `mm`, `cm`, `m`, `km`); the unit of measurement to which it wants to convert (choosing from: `fl`, `oz`, `gal`, `oz`, `lb`, `in`, `ft`, `mi`), rejecting incompatible conversions (such as, for example, `km` to `gal`); the value to be converted. The program is supposed to display the data entered and the value resulting from the conversion. [P3.26]

03.2.8 Shopping vouchers. A supermarket rewards its customers with shopping vouchers whose amount depends on the amount of money spent on food. For example, if you spend `$50`, you get a shopping voucher equal to `8%` of the amount you spent. The table below shows the percentage used to calculate the shopping voucher relative to different amounts. Write a program that calculates and displays the value of the shopping voucher given to the customer, based on the amount of money he or she spent on the purchase of groceries. [P3.40]

Money spent	Percentage of the voucher
Less than $ 10	No voucher
From $ 10 to $ 60	8%
From more than $ 60 to $ 150	10%
From more than $ 150 to $ 210	12%
More than $ 210	14%

03.2.9 Wavelengths. Write a program that asks the user as input a wavelength value (real number, which can be written in scientific notation, e.g., `1.23e-7`), and displays the description of the corresponding part of the electromagnetic spectrum, as shown in the table below. [P3.43]

Type	Wavelength (m)	Frequency (Hz)
Radio waves	$> 10^{-1}$	$< 3 \times 10^{9}$
Microwaves	$From\ 10^{-3}\ to\ 10^{-1}$	$From\ 3 \times 10^{9}\ to\ 3 \times 10^{11}$
Infrared	$From\ 7 \times 10^{-7}\ to\ 10^{-3}$	$From\ 3 \times 10^{11}\ to\ 4 \times 10^{14}$
Visible light	$From\ 4 \times 10^{-7}\ to\ 7 \times 10^{-7}$	$From\ 4 \times 10^{14}\ to\ 7.5 \times 10^{14}$
Ultraviolet	$From\ 10^{-8}\ to\ 4 \times 10^{-7}$	$From\ 7.5 \times 10^{14}\ to\ 3 \times 10^{16}$
X-Rays	$From\ 10^{-11}\ to\ 10^{-8}$	$From\ 3 \times 10^{16}\ to\ 3 \times 10^{19}$
Gamma rays	$< 10^{-11}$	$> 3 \times 10^{19}$

03.2.10 Back to the comet. A person on average can jump off the ground with a speed of 11 kilometers per hour without having to fear leaving the Earth's surface. Conversely, if a person jumped with the same speed while on Halley's Comet, would he or she be able to return to the surface? Create a program that allows the user to input a launch speed (in kilometers per hour) from the surface of Halley's Comet, and determine whether the person jumping will be able to return to the surface. If not, the program will need to calculate how much more mass the comet would have to have for that to happen.

Hint: the escape velocity is equal to

$$v_{escape} = \sqrt{2\frac{G\,M}{R}},$$

where $G = 6.67 \times 10^{-11}\,N\,m^2 / kg^2$ is the gravitational constant, M is the mass of the celestial body, and R is the radius. Halley’s Comet has a mass of $2.2 \times 10^{14}\,kg$ and a diameter of $9.4\,km$. [P3.52]