Computer science (code 30424) a.y. 2020-2021

Exercises lesson 11

EXERCISES ON LECTURE TOPICS

All the exercises are provided with the script of a possible solution, considered the most efficient to illustrate the tools introduced in the theoretical lesson.
To verify if the code is working properly, test different logical flows and use different data types. Use the course textbook[1] as a support.

EXERCISE 11.1

In summer 2019, some famous groups and singers held one or more concerts at the San Siro stadium in Milan. Create in Python the program **Exercise11.1.py** that must:

a. ask the user for his/her favourite singer or band
b. if it belongs to the ones listed below, display the sentence:
"Do you know that in June/July 2019 there was a concert in San Siro?!!!"

Concerts at San Siro June/July 2019
• Vasco Rossi
• Ed Sheeran
• Muse

EXERCISE 11.2

The new electronic meter which measures the electricity consumption in Italian houses allows, for an unlimited time, the withdrawal of a power up to 10% higher than the maximum value signed in the contract. Create in Python the program **Exercise11.2.py** which must:

a. ask the user for the maximum electric power signed in the contract: 3, 4.5 or 6 kW
b. ask the user for the total power consumption (expressed in kW) of the electrical devices, assuming that they are all turned on at the same time
c. show the sentence *"Don't turn everything on at the same time, otherwise the electricity supply will be disconnected"* if the consumption exceeds the maximum power and *"Your consumption is below the limit of your* **XXX** *kW meter!"* (replacing XXX with the maximum power) otherwise

[1] Learning Python, A. Clerici, M. De Pra, M.C. Debernardi, D. Tosi, Egea, 2020

EXERCISE 11.3

Create in Python the program **Exercise11.3.py** that must:

a. generate and display an integer between 0 and 1000
b. display a message to say if the number is a multiple of 7

EXERCISE 11.4

Create in Python the program **Exercise11.4.py** that must:

a. ask the user how many modules of the ECDL Full Standard certification he/she has already passed
b. check if the number is an admissible value (0-7), otherwise it must print the message *"Invalid data - Run the program once again"*
c. if the answer is acceptable, the following messages should appear (adapt them when needed):

How many exams?	Text to display
7	Great, you passed them all :-)
5-6	Go, go, go, you are almost there!
1-4	Come on, only **XXX** missing modules, finish them quickly
0	What are you waiting for? You cannot do the second intermediate exam or the general exam without the prerequisite.

EXERCISE 11.5

Create in Python the program **Exercise11.5.py** that contains a variable (created by the user) and prints on the screen if it is an integer, a floating point, a Boolean, a string or a generic type of data (not included in the previous ones).

Test the program with different types of data.

EXERCISE 11.6

Create in Python the program **Exercise11.6.py** that must:

a. ask the user to choose heads or tails
b. generate a random number between 1 and 2, where 1 means heads and 2 tails
c. compare the choice of the user with the random number and print the following sentence (properly adapted):
"The outcome of the virtual coin is HEADS/TAILS so you WON/LOST!"

EXERCISE 11.7

We want to create the program **Exercise11.7.py** to support the cashiers of a clothing chain of stores. The program must:

a. ask for the total purchase value, expressed in euros (cents are allowed; max value 9999.99 euros)
b. calculate the discounted price, rounded down to euros, based on the following discount table:

Amount (€)	Discount
Up to 30	-
30,01 – 70	3%
70,01 – 120	5%
120,01 – 240	8%
More than 240	12%

c. display the following output (showing decimal and thousands separators), with the currency values aligned on the decimal separator:

```
Full price:              #,###.## EUR
Applied discount: x%
Discount value:            ###.## EUR
\-----------------------------------\
Discounted price:        #,###.## EUR
```

EXERCISE 11.8

Create in Python the program **Exercise11.8.py** to ask the user for his/her favourite foods. The program must use an infinite loop and ask for a new favourite dish at each step of the loop. The loop must be interrupted when the user writes *"ENOUGH"* and the program must finally print the sentence *"You like* **XXX** *dishes"*, where XXX contains the number of foods entered by the user.

EXERCISE 11.9

Create in Python the program **Exercise11.9.py** which uses a *while* loop to increment a variable, called *var1* and initialized to 10, until it reaches the value of 100. At each step the variable must be incremented by 10 and the corresponding square root must be printed on the screen.

E.g., when *var1* is equal to 20, the following message must be displayed:
"The square root of 20 is 4.47213595499958"

EXERCISE 11.10

Create in Python the program **Exercise11.10.py** which uses a *for* loop to increment a variable, called *var1* and initialized to 25, until it reaches the value of 150. At each step the variable must be incremented by 5 and the corresponding square root must be printed on the screen.

E.g., when *var1* is equal to 30, the following message must be displayed:
"The square root of 30 is 5.477225575051661"

EXERCISE 11.11

Create in Python the program **Exercise11.11.py** which asks for an integer value and displays the sum of all numbers from 1 to the given value, without checking if the number entered by the user is actually an integer.

> E.g. if the value given by the user is 5, the following message must be displayed:
> *"The sum of numbers from 1 to 5 is 15"*

EXERCISE 11.12

Create in Python the program **Exercise11.12.py** to create a list of tags entered by the user, respecting the following requests:

a. the program must use a loop to ask the user for each single tag (a tag is simply a text string)
b. the loop must stop when the user enters an empty string
c. all the tags must be stored in a single string variable, adding each new tag at the end of the string, using a comma and a blank space as separator
d. at last, the program must print the list of tags

> E.g. if the tags entered by the user are "University", "Milan" and "Computer Science", the program must display the following message:
> *"Tag list: University, Milan, Computer Science, "*

EXERCISE 11.13

Create in Python the program **Exercise11.13.py** that must:

a. generate three random numbers between 0 (included) and 1 (not included) and print them in the same order in which they have been created
b. display the numbers sorted in descending order, with 4 decimal places

The following is an example of output:

```
0.9068460185504952
0.707501730324115
0.15628479510621263
1st value:  0.9068
2nd value:  0.7075
3rd value:  0.1563
```

EXERCISE 11.14

Create in Python the program **Exercise11.14.py** that must:

a. ask the user how many times (from 1 to 4) he/she wants to roll an eight-sided die
b. show the results of each roll
c. show the average score (rounded to the first decimal place) remembering that, if number of rolls is 1, the average corresponds to the roll's result
d. show the message *"You want me to work too much!"* if the user asks for more than 4 rolls

EXERCISE 11.15

Create in Python the program **Exercise11.15.py** which must ask the user for his/her preferences about colours, flavours, numbers, scents, sounds and tactile consistency, thus evaluating the psychological type.

NB: this program wants to be a game, it has no scientific claim!

In particular, the program must:

a. ask the user to choose:
   - a colour between red, yellow, green, black and blue
   - a flavour between sweet and salty
   - a positive integer from 1 to 20
   - a scent among citrus fruits, honey, coffee and sandalwood
   - the sound of the stormy sea or the sound of the drizzle
   - a tactile consistency between silk, fur, sandpaper and steel

b. classify the user into a psychological type (sensorial, sentimental, intuitive, logical-rational) according to the given answers:

TYPE	COLOUR	FLAVOUR	NUMBER	SCENT	SOUND	CONSISTENCY
Sensorial	red o yellow		multiple of 4	sandalwood	drizzle	
Sentimental	green o blue	sweet		honey o citrus fruits		NOT sandpaper
Intuitive		salty	less than 10			silk or steel
oppure…	blue				storm	
Logical-rational	black					
oppure…			odd	coffee		steel

c. show the message *"I'm sorry, you're really a strange guy ;-)"* if the user doesn't belong to the four types written above

d. provide some explanatory messages to describe the purpose of this program.