Laboratory 1

Objective
The topic of this laboratory is related to a gentle introduction in Python programming, focusing on the following themes:

- Familiarizing with the Python interpreter
- Familiarizing with a Python IDE, and we will use PyCharm Community Edition for writing our programs, but any other IDE is accepted.
- Writing small pieces of code based on the information received during the Lecture 1.

In order to be able to solve the proposed problems, some additional information about printing results or Python decision statement is needed, so it will be presented below.

print statement

When using Python code written inside of a file, for printing the results of the executed statements, a special statement called `print` is used. As we saw in the first lecture, we can obtain information about a Python method by invoking the `help` directive:

```
>>> help(print)
Help on built-in function print in module builtins:

print(...)
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

    Prints the values to a stream, or to sys.stdout by default.
    Optional keyword arguments:
    file:  a file-like object (stream); defaults to the current sys.stdout.
    sep:   string inserted between values, default a space.
    end:   string appended after the last value, default a newline.
    flush: whether to forcibly flush the stream.
```

At this moment we will use the simplified form of the print method, considering all the default settings.

if/else statements

The `if` statement executes if and only if the expression is True, otherwise the `else` expression is executed.

```
if expression:
   statement(s)
```

```
else:
   statement(s)
```

elif statement

This allows multiple statements to be checked and execute a block of code as soon an expression is evaluated to True

```
if expression1:
   statement(s)
elif expression2:
   statement(s)
elif expression3:
   statement(s)
else:
   statement(s)
```

Modulus %

Yields the remainder of division of the first argument to the second.

```
>>> 4 % 3
1
```

Retrieving input from the command line

```
input([prompt])
```

This function reads from the command line, converts into a string and returns it. If an int is expected, for example, the string needs to be converted using

```
int(s)
```

Assignment 1

Launch the Python shell and identify the version of the interpreter.
Issue the commands that were discussed during the lecture and see what happens.
Even if we didn't talk about Python modules yet, try the command below and see what happens:

```
>>>import this
```

Problem 1
Given three numbers, represented by three variables, write a piece of code which displays the maximum of them, at `stdout`. Use only `if/else/elif` and `print` statements.

Problem 2
Determine if a number is odd or even, using only the instructions learned by now. The number will be read from the command line.

Problem 3
Read a temperature (in Celsius degrees) from the command line, transform it to Fahrenheit using the formula below, then display the result:

```
Tf=(9/5)*Tc+32
```

Problem 4
Read a temperature (in Fahrenheit degrees) from the command line, transform it to Celsius using the formula below, then display the result:

```
Tc=(5/9)*(Tf-32)
```

Problem 5
Leap years occur according to the following formula: a leap year is divisible by four, but not by one hundred, unless it is divisible by four hundred. For example, 1992, 1996, and 2000 are leap years, but 1993 and 1900 are not. The next leap year that falls on a century will be 2400. Write a program which takes a year from the input console and prints to `stdout` if it is leap or not.

Problem 6
Read two numbers from the input console, then swap them and print the result to the `stdout`.

Problem 7
Read a number from the input console and check if it is negative, zero or positive. For each case print the result to `stdout`.

Problem 8
Read a 10 digit telephone number from the input console and format it, using different formatters like:
(XXXX)XXXXXX
XXXX-XXXXXX
XXX-XXX-XXXX

Problem 9
Given a string, write a program that will delete any '*' from it, and all the first characters located at the left and the right of the '*'. Examples:

'st*r' -> 's'
'pyt*on' -> 'pyn'

Problem 10

Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.

Sample String : 'restart'

Expected Result : 'resta$t'