WORK PACKAGE 0: SET-UP

EXERCISE 1: ENVIRONMENT

In this exercise, you should set-up your development environment[1]. You should make sure that you can compile the following program in your IDE:

```
// (C) __Your names here, group: 123 __ (2022)
// Work package 0
// Exercise 1
// Submission code: XXXXXX (provided by your TA-s)
#include <stdio>

// Main function in the program, no program arguments supported
void main (void) {

        \\ Print a string to the console
        printf("%s", "Hello World!\n");       \\ Note: double quotes
}
```

You should also execute the binary file from:

a) Your IDE
b) Your console/terminal

EXERCISE 2: PARAMETRIZED PROGRAMS

In this exercise, you should extend the program from Exercise 1 by adding a program argument. You should write the program, which will print the string "`Hello World! – I'm <argument>!`". The <argument> should be provided in the command line, when executing the program, e.g.

```
my_program.exe <argument>
```

Example:

> For the following program execution: `my_program.exe tom`, the result should be `Hello World! – I'm tom!`

EXERCISE 3: FAIL-SAFETY

You should modify your program from Exercise 2 to be fail-safe – adding the check of the argument.

- If the user provides no argument – the program should write an error message and information about the parameter "-h"
- If the user provides more than one argument – the same as above
- If the user provides the first argument "-h" – the program should write the information how to use it

[1] You can use any desktop programming IDE that you want if it supports C. **You can check your programs with online compilers (e.g. https://www.onlinegdb.com), but you cannot use them in the demonstration.** Lectures will show the use of Visual Studio 2019, Visual Studio Code and, occasionally, CLion. The same is true for the compilers – visual studio compiler and gcc are supported and used in the lectures.