SIMULATOR OF A PRIMITIVE COMPUTER SYSTEM
V0

Initial tasks

Establish a terminal connection (`putty` or `MovaXterm`) to the machine named `ritchie`. Then, make a copy of the directory `/var/asignaturas/ssoo/2024-2025/V0` in your private directory. You can download a compressed version of that directory from the eCampus as well.

The following exercises must be done working with the contents of the copied set of files.

You can use Linux Virtual Machines or "***Ubuntu on Windows***" instead of using `ritchie`.

Exercises

1. Modify the **call** to the `ComputerSystem_DebugMessage()` function, placed in the function `Processor_DecodeAndExecuteInstruction`, and what is necessary in the file "`messages.txt`" in order to show the contents of the PSW register, so the displayed message (number 3) looks like this (PSW should be shown in hexadecimal format and the numbers are in red colour):

   ```
   (PC: 233, Accumulator: 1 [00000001], PSW: 00000000)
   ```

2. Modify the `ComputerSystem_PowerOff()` function so it shows a message, section `SHUTDOWN`, that looks like:

   ```
   END of the simulation
   ```

3. Modify the function `Processor_DecodeAndExecuteInstruction()345` to add a new instruction called `MEMADD` to the processor set of instructions. Its syntax will be:

   ```
   MEMADD operand1 memAddress
   ```

   The instruction will add `operand1` and the contents of the memory cell addressed by `memAddress`. The result will be stored in the accumulator register.

   To create the necessary enumerated value for the new instruction, add this at the end of "`Instructions.def`":

   ```
   INST(MEMADD)   // decimal 10 = Hex (0x0a)
   ```

4. Write a user program (in a file called `prog-V0-E4`) that stores, successively, the **odd** integer values between 100 and 120 in memory cell at address 122. For this purpose, use jump instructions and the above `MEMADD` instruction.
5. Write a user program (in a file called `prog-V0-E5`) that stores, successively, the powers of 2, in ascending order, from 1 (2^0) to 1024 (2^{10}), in memory cell at address 64.
6. OPTIONAL (try this when having finished all previous exercises).

Modify `ComputerSystem.c` so the simulator executes the program whose name is stored in first line of the file "`nameOfTheProgramToBeExecuted`". If file "`nameOfTheProgramToBeExecuted`" or the one referenced by the name inside that file do not exist, the program to be executed would be `programToBeExecuted`.

For example, if first line of `nameOfTheProgramToBeExecuted` contains:

```
prog-V0-E4
```

Program of exercise 4 would be executed.