

# Arquitetura de Computadores 2019/20

## TPC 1

**Deadline: 23:59, April 1<sup>st</sup>, 2020**

This homework consists of two individual programming exercises. You can discuss general doubts with colleagues, but the solution and the code writing should be strictly individual. All solutions will be automatically compared, and plagiarism cases will be punished in accordance with the regulations.

Your solutions should be submitted to DI's Mooshak (<http://mooshak.di.fct.unl.pt/~mooshak/>). The OS is a Linux and your program is compiled with the following command (in case of errors or warnings your program will fail):

```
cc -m32 -Wall -std=c11 -o prog prog.c
```

Note that 32bits architecture will be used and your program's output must be exactly as in the examples.

### Problem 1 (50%)

---

Write a C program that reads two integer numbers from standard input and then, if possible, writes to the standard output its sum and its product. In case one of the operations overflows the architecture capacity supported by your C compiler, the message "operation overflow" should be printed.

Example 1:

in:

**2  
30**

out:

**32  
60**

Example 2:

in:

**10000000000  
40000000000**

out:

**50000000000  
operation overflow**

The input is always integer numbers from -9000000000000000000 to 9000000000000000000.

See documentation for C functions `scanf` and `printf`.

## Problem 2 (50%)

---

Write a C program that reads a float number from standard input and then writes to the standard output the binary representation of its mantissa, including sign, the hidden bit and decimal point.

Example 1:

in:

**2.5**

out:

**+1.010000000000000000000000**

Example 2:

in:

**0**

out:

**+0.000000000000000000000000**

For this problem you should use the bitwise operators (&, |, ^, >>, <<, ~) to get the sign and mantissa bits.

Note that you can put the *float* representation into an *unsigned int* using pointers and cast:

```
unsigned int x = * (unsigned int*) &f;
```