15.10.2020

PRESENTACIÓN LAB 1 : data lab

- No se pueden usar loops, condicionales, constantes de más de 8 bits, macros, funciones auxiliares → excepto en los de punto flotante

- El código hay que pushearlo a la rama lab1, generar un pull request y el otro compañero lo va a tener que aprobar para poder mershear a la rama lab1.

- Cuando el TP haya finalizado, para entregar, desde la rama lab1 hago un pull request a la rama master.

- allEvenBits { devuelve 1 si tengo un 1 en todos los bit pares

32   0

↳ más significativo   ↳ menos significativo

→ ejemplo:

allEvenBits (0xFFFFFFFE)   'F F F F F F F E'   '1111 1111 1111 1111 1111 1111 1111 1110'

→ return 0 : devuelve 0 porque en la posición 0 (par) tiene un 0 en vez de un 1 y debería haber un 1 en TODAS las posiciones pares

legal ops: ! ~ & ^ | + << >>

```
int allEvenBeats (int x) {

    int m8 = 0xAA;  (1)
    int m16 = m8 | (m8 << 8);  (2)
    int m32 = m16 | (m16 << 16);  (3)
    int fillx = x | m32;  (4)

    return !~fillx;  (5)
}
```

(1)   m8 = $\overbrace{1010}^{A}$ $\overbrace{1010}^{A}$

(2)   m8 << 8 = 1010101000000000

             10101010

OR ← ① 1010101000000000

        1010101010101010 = m16

                    16 0s

(3)   m16 << 16 = 10101010101010100000000000...0

supongo que x tiene 1s en sus posiciones pares

                      10101010101010

| 1010101010101010101010100000000000000000

m32 = 10101010101010101010101010101010

si x = 01010101010101010101010101010101

      11111111111111111111111111111111

↳ cuando lo niego me va a quedar == 0, entonces devuelvo !0 == 1 ✓