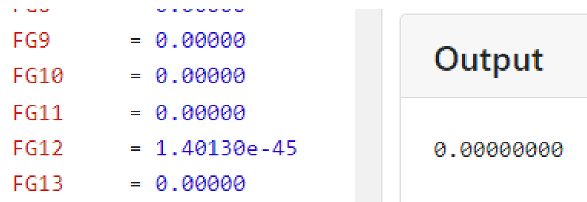
Assignment 3-4 Floating Point Representation: Diff between SNN and LDN. Lab 3

1. **Make a assembly program that has the hexadecimal data to represent the**
    a. Smallest normalized positive number
       val1: .word 0x00800000 # Smallest Normalized Number
    b. Largest denormalized positive number
       val2: .word 0x007FFFFF # Largest Denormalized Number
2. **And then make assembly code to subtract val1 from val2**

```
lwc1    $f5, val1
lwc1    $f7, val2
sub.s   $f12, $f5, $f7
```

3. **Print the result of subtraction, the printed value of $f12**
4. **Show the program result as a screenshot**

3 and 4 > 

5. **Elaborate on the value with IEEE 754 format**
    a. **.word 0x007FFFFF**

0x007FFFFF = 0000 0000 0111 1111 1111 1111 1111 1111
0 = sign bit = **positive**
0000 0000 = **denormalized** according to IEEE 754 standards.
111 1111 1111 1111 1111 1111 = Mantissa = largest possible value for the mantissa in a denormalized number, so it's 1 - 2^-23, which comes out to approximately **0.9999- and more** which is **very close to 1.0**

    b. **.word 0x00800000**

0x00800000 = 0000 0000 1000 0000 0000 0000 0000 0000
0 = sign bit = **positive**
0000 0001 = Exponent = **1**
Biased exponent = 1, actual exponent = -126 because **1 - 127 is -126.**
000 0000 0000 0000 0000 0000 = Mantissa = still exactly **1.0**

    c. **The register value $t0 after subtraction $f5 - $f7**

0x00000001 = 0000 0000 0000 0000 0000 0000 0000 0001
0 = sign bit = **positive**
0000 0000 = **denormalized**
000 0000 0000 0000 0000 0001 = Mantissa = **smallest possible positive denormalized number**, so it's **0.00000000000000000000001 or 1.40130e-45 or 1.401298 x 10^-45**
It's very close to zero but it's still a positive number.
However, moving the value from the floating point register to the general purpose register may results in the loss of precision, so the value may be interpreted as **1**.