Hardware - Tutorial 1
Two's Complement and Floating point representation

Exercice 1: Signed Numbers

Encode the following numbers into 8 bits signed binary.

- -1

- -29

- -42

- -127

- -128

- -175

Exercice 2: Signed Operations

Perform the following 8 bits binary operations. Give the result in 8 bits binary and convert it to decimal given the context is signed or unsigned. If an overflow occurs, write down 'ERROR' in the corresponding cell.

Operation	Binary Result	Decimal Value	
		Unsigned	Signed
1111 0101 + 1111 1010			
1110 1000 - 1100 0110			
0101 1110 - 1001 1110			
0111 1110 + 0000 0101			
1100 1011 - 0001 1010			
1000 0000 + 1111 1010			
1000 0011 - 0000 1010			

Exercice 3: Decimal to float

Convert the following decimal numbers to their binary **single-precision** floating point representation:

- 128

- -32.75

- 18.125

- 0.0625

Exercice 4: Float to decimal

Convert the following **single-precision** floating point numbers to decimal representation:

- 1011 1101 0100 0000 0000 0000 0000 0000

- 0101 0101 0110 0000 0000 0000 0000 0000

- 1100 0001 1111 0000 0000 0000 0000 0000

- 1111 1111 1000 0000 0000 0000 0000 0000

- 0000 0000 0100 0000 0000 0000 0000 0000

Exercice 5: Decimal to double

Convert the following decimal numbers into their binary **double-precision** floating point representation:

- 1

- -64

- 12.06640625

- 0.2734375

Exercice 6: Double to decimal

Convert the following **double-precision** floating point numbers to decimal representation:

- 403D 4800 0000 0000

- C040 0000 0000 0000

- BFC0 0000 0000 0000

- 8000 0000 0000 0000

- FFF0 0001 0000 0000

Exercice 7: Float danger

Let us consider the following C program:

```c
void main() {
  float f1, f2, f3, r;

  f1 = 1E25; // f1 = 10^25
  f2 = 16;

  f3 = f1 + f2;
  r = f3 - f1;

  printf("r = %f\n", r);
}
```

Indication: $10^{25} \approx 2^{83}$

1) Once the program has run through, what will the value of r be ? Explain your reasoning.

2) Assuming that $f1 = 10^n$ where n is a natural number, what is the largest value of n that still gives a correct value of r ?

3) Assuming that $f1$, $f2$, $f3$ and r are declared as double, what is the largest value of n that still gives a correct value of r ?