Homework 11

Question 1:

Express the following base 10 number in IEEE 754 single-precision floating-point format. Express your answer in Hexidecimal.

```
-13.5625 ⇒ (?)
```

solution

Integer conversion:

```
⇒ (13)/2 ⇒ 6 r 1
⇒ (6)/2 ⇒ 3 r 0
⇒ (3)/2 ⇒ 1 r 1
⇒ (1)/2 ⇒ 0 r 1

Result: 1101
```

Decimal conversion:

```
⇒ (.5625) × 2 = 1.125
⇒ (.125) × 2 = 0.25
⇒ (.25) × 2 = 0.5
⇒ (.5) × 2 = 1.0

Result: .1001
```

Combining

$\Rightarrow$ 1101.1001

$$\Rightarrow 1101.1001 = 1.1011001 \times 2^3$$

We can create a Generalized Program:

```
(*if sign was not 1 | !(1) = 0.1011001*)
let Sign = 1;
let Bias = 127;
let Exponent = 3;
(*total = 130 = 10000010*)
let Total = Bias + Exponent;
(*Normalized = above 1.1011001 - sign = 1011001*)
let Normalized = 1011001;
let zeroPadding = 0000000000000000;
(*
  Mantissa consist of the combined binary
  normalization & the trailing 16 bits (zeroPadding)
*)
let Mantissa = Normalized + zeroPadding;

(*
now combine: {Sign & Total & Mantissa}
 ⇒ 1 10000010 10110010000000000000000
*)

let Result = Sign + Total + Mantissa;
```

$Q.E.D.$

$\Rightarrow$ we can convert the `Mantissa` $\Rightarrow$ `Hexidecimal`

$\Longrightarrow$

Result	Hexidecimal
1100	C
0001	1
0101	5
1001	9
`ZeroPadding`	0x16 = 000...

$\therefore -13.5625 \Longrightarrow$ `0xC1590000`

Question 2:

Convert the following IEEE 754 single-precision floating-point number to decimal format.

0x40980000

solution

Starting with:

`0x40980000` ⟹ `0100 0000 1001 1000 0000 0000 0000 0000`

hex	binary
4	0100
0	0000
9	1001
8	1000
ZeroPadding	0x16 = 000...

⟹

```
let Sign = 0;
let exponent = 10000001; (*129*)
let Mantissa = 00110000000000000000000;
```

⟹ exponent - 127 = 2;

Reconstructing Mantissa:
⟹ 1 + 0.5 + 0.25 +

Question 3:

Translate this C++ code into RISC-V assembly language with correct use of Floating-Point instructions where necessary. Submit your code and screenshot of the outputs.

```
int main() {
 float value1 = 3.5;
 float result = 0;

 if (value1 < 7)
     result = 7 + value1;
 else
     result = value1 / 7;

 cout << result << endl;
}
```

solution

something