Assignment #5

Student name: *Ivan Kabadzhov*

Course: *Introduction to Computer Science*
Date: *October, 2018*

Problem 5.1: utopia number system

$$\alpha^2 = \alpha \Rightarrow \alpha = 0 \vee \alpha = 1$$
$$\alpha + \alpha = \beta \Rightarrow \alpha = 1 \wedge \beta = 2$$
$$\gamma^2 = \gamma \Rightarrow \gamma = 0 \vee \gamma = 1$$
$$\gamma + \gamma = \gamma \Rightarrow \gamma = 0$$
$$\delta = \beta\beta \Rightarrow \delta^2 = 22$$
$$\delta + \delta = \alpha\alpha \Rightarrow \delta + \delta = 11$$

Considering irregular base:

$$(\delta + \delta)^2 = \delta^2 + 2\delta^2 + \delta^2 = 4\delta^2$$
$$\Rightarrow 11^2 = 4 \cdot 22 \Rightarrow 121 = 88$$

Note that in any base $[d_n..d_0] = d'_n \cdot b^n + ... + d'_0 \cdot b^0$.

$$121 = 88$$
$$1x^2 + 2x^1 + 1x^0 = 8x^1 + 8x^0$$
$$x^2 - 6x - 7 = 0$$
$$(x+1)(x-7) = 0$$

The number system must be in $\mathbb{N}$. Therefore the calculator uses 7 base system.
To represent 99:

$$99 \div 7 = 14 \rightarrow remainder = 1$$
$$14 \div 7 = 2 \rightarrow remainder = 0$$
$$2 \div 7 = 0 \rightarrow remainder = 2$$
$$\Rightarrow 99_{10} = 201_7$$

Problem 5.2: b-complement

a)

$$1 \div 5 = 0 \rightarrow remainder = 1$$
$$\Rightarrow 1_{10} = 0001$$
$$\Rightarrow -1_{10} = (4444 - 0001) + 1 = 4444$$

$$8 \div 5 = 1 \rightarrow remainder = 3$$
$$1 \div 5 = 0 \rightarrow remainder = 1$$
$$\Rightarrow 8_{10} = 0013$$
$$\Rightarrow -8_{10} = (4444 - 0013) + 1 = 4432$$

b)

$$(-1) + (-8) = 4444 + 4432 = 4431$$

To convert back to decimal:

$$4444 - (4431 - 1) = 0014 = 0 \cdot 5^3 + 0 \cdot 5^2 + 1 \cdot 5^1 + 4 \cdot 5^0 = 9_{10}$$

Problem 5.3: IEEE 754 floating point numbers

a)

$$\begin{aligned}
273 \div 2 &\rightarrow remainder = 1\\
136 \div 2 &\rightarrow remainder = 0\\
68 \div 2 &\rightarrow remainder = 0\\
34 \div 2 &\rightarrow remainder = 0\\
17 \div 2 &\rightarrow remainder = 1\\
8 \div 2 &\rightarrow remainder = 0\\
4 \div 2 &\rightarrow remainder = 0\\
2 \div 2 &\rightarrow remainder = 0\\
1 \div 2 &\rightarrow remainder = 1\\
\Rightarrow 273_{10} &= 100010001_2 = 1,00010001 \cdot 2^8
\end{aligned}$$

$$\begin{aligned}
0,15 \cdot 2 = 0,3 &\rightarrow int = 0\\
0,3 \cdot 2 = 0,6 &\rightarrow int = 0\\
0,6 \cdot 2 = 1,2 &\rightarrow int = 1\\
0,2 \cdot 2 = 0,4 &\rightarrow int = 0\\
0,4 \cdot 2 = 0,8 &\rightarrow int = 0\\
0,8 \cdot 2 = 1,6 &\rightarrow int = 1\\
\Rightarrow 0,15_{10} &= 0{,}00(1001)_2
\end{aligned}$$

Adding the power to find the exponent $\Rightarrow 127 + 8 = 135$.

$$\begin{aligned}
135 \div 2 &\rightarrow remainder = 1\\
67 \div 2 &\rightarrow remainder = 1\\
33 \div 2 &\rightarrow remainder = 1\\
16 \div 2 &\rightarrow remainder = 0\\
8 \div 2 &\rightarrow remainder = 0\\
4 \div 2 &\rightarrow remainder = 0\\
2 \div 2 &\rightarrow remainder = 0\\
1 \div 2 &\rightarrow remainder = 1\\
\Rightarrow 135_{10} &= 10000111_2
\end{aligned}$$

As the number is negative denote the sign with $S = 1$.

$$|1|10000111|00010001001001100110011|$$

b)

$$\begin{aligned}
&-100010001,00100110011001100110011001_2 =\\
&= -(2^7 + 2^3 + 2^0 + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-14} + ...2^{-23}) =\\
&= -273,149993896484375_{10}
\end{aligned}$$

Problem 5.4: decimal to binary (haskell)

```
bin :: Int -> [Int]
bin 0 = [0] --base case
bin 1 = [1] --base case
bin x = bin (x `div` 2) ++ [(mod x 2)] -- recursion

binf :: Double -> [Int]
binf 0 = [] --base case
binf n = [fromIntegral(truncate n)] ++  -- should be on same line
        take 23 (binf(n*2 - fromIntegral(truncate n*2))) -- recursion
-- truncate return the nearest argument between 0 and 1
-- manually stop the recursion
```