Introduction to Computer Science
Jacobs University Bremen
Dr. Jürgen Schönwälder

Module: CH-232
Date: 2022-10-07
Due: 2022-10-14

ICS 2022 Problem Sheet #5

Problem 5.1: *IEEE 754 floating point numbers* (4+1 = 5 points)

IEEE 754 floating point numbers (single precision) use the following format (the numbers on the top of the box indicate bit positions, the fields in the box indicate what the various bits mean).

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S|    exponent   |         mantissa (23 bits)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The encoding starts with a sign bit, followed by the biased exponent (8 bits), followed by the mantissa (23 bits). For single-precision floating-point numbers, the exponents in the range of -126 to +127 are biased by adding 127 to get a value in the range 1 to 254 (0 and 255 have special meanings).

a) The standard acceleration of gravity is roughly 9.80665 meters per square seeconds. Explain step by step (in your own words) how the decimal fraction 9.80665 is converted into a single precision floating point number.

b) What is the decimal fraction that is actually stored in the single precision floating point number and what is the absolute error?

Problem 5.2: *unicode and utf-8 encoding* (1+2+1 = 4 points)

The content of a file containing UTF-8 Unicode encoded text is given by the following sequence of bytes in hexadecimal notation:

```
f0 9f 94 a8 20 e2 88 a7 20 f0 9f 8e 93 0a
```

a) Write each byte in binary notation.

b) Identify the unicode code points of the characters. What is the text stored in the file?

c) Which line end convention is used? What are other popular line end conventions?

Problem 5.3: *large Schröder numbers (haskell)* (1 point)

The large Schröder numbers S_n with $n \in \mathbb{N}$ are defined using the following equation:

$$S_n = \begin{cases} 1 & n = 0 \\ S_{n-1} + \sum_{k=0}^{n-1} S_k \cdot S_{n-k-1} & \text{otherwise} \end{cases}$$

For more information about large Schröder numbers, see the documentation of the number sequence A006318 in the On-Line Encyclopedia of Integer Sequences.

Implement a function `s :: Integral a => a -> a` to compute the nths Schröder number. Write unit test cases using the HUnit test framework.

Submit your Haskell code as a plain text file.