Introduction to Computer Science
Jacobs University Bremen
Dr. Jürgen Schönwälder

Module: CH-232
Date: 2021-10-01
Due: 2021-10-08

ICS 2021 Problem Sheet #4

Problem 4.1: *b-complement* (1+1+1 = 3 points)

We plan to use a fixed size b-complement number system with the base $b = 7$ and $n = 4$ digits.

a) What are the smallest and the largest numbers that can be represented and why?

b) What is the representation of -1 and -8 in b-complement notation?

c) Add the numbers -1 and -8 in b-complement notation. What is the result in b-complement representation? Convert the result from b-complement representation back into the decimal number system.

Problem 4.2: *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) 20.5 degree Celsius corresponds to 293.65 Kelvin. Explain step by step (and in your own words) how the decimal fraction 293.65 is converted into a single precision floating point number.

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

Problem 4.3: *munged passwords (haskell)* (1+1 = 2 points)

Some people try to create stronger passwords through character substitution. The substitutions can be anything the user finds easy to remember. We use the following substitution:

character	a	b	c	d	e	f	g	h	i	l	o	q	s	x	y
substitution	@	8	(	6	3	#	9	#	1	1	0	9	$	%	?

Using this table, the string `hello world` is munged into the string `#3110 w0r16`.

a) Using pattern matching, implement a function `sub` that takes a character and returns either the character or a substitution of it. Write down the type signature of your function followed by its definition.

b) Using pattern matching, implement a function `munge` that takes a string and returns a string with all character substitutions applied. Write down the type signature of your function followed by its definition.

Submit your Haskell code plus an explanation (in Haskell comments) as a plain text file.