Introduction to Computer Science
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: CH-232-A
Date: 2019-10-11
Due: 2019-10-18

ICS 2019 Problem Sheet #5

Problem 5.1: *b-complement* (1+1 = 2 points)

Consider a b-complement number system with the base $b = 5$ and $n = 4$ digits.

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

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

Problem 5.2: *IEEE 754 floating point numbers* (3+1 = 4 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 S (set to 1 if the number is negative), followed by the 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). The exponent itself is represented in binary form (note we have a positive integer after adding the bias).

a) The absolute zero, 0 Kelvin, is at -273.15 degree Celsius. Explain step by step in your own words how the decimal fraction -273.15_{10} 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?

Problem 5.3: *base n numbers (haskell)* (1+1+1+1 = 4 points)

a) Implement a function `toBase :: Int -> Int -> [Int]`, which takes a base b and a decimal number n and returns the number n as a list of digit in the base b number system.

```
Prelude> map (\b -> toBase b 16) [2..12]
[[1,0,0,0,0],[1,2,1],[1,0,0],[3,1],[2,4],[2,2],[2,0],[1,7],[1,6],[1,5],[1,4]]
```

b) Implement a function `fromBase :: Int -> [Int] -> Int`, which takes a base b and a list of digit in the base b number system and returns the decimal integer.

```
Prelude> map (\b -> fromBase b [1,1,1]) [2..16]
[7,13,21,31,43,57,73,91,111,133,157,183,211,241,273]
```

c) Implement a function `showBase :: Int -> Int -> String`, which takes a base b and a decimal number n and returns the number n as a string using digits appropriate for the base. Implement the convenience functions `showBin :: Int -> String`, `showOct :: Int -> String`, and `showHex :: Int -> String`.

```
Prelude> map (\b -> showBase b 16) [2..16]
["10000","121","100","31","24","22","20","17","16","15","14","13","12","11","10"]
```

d) Implement a function `readBase :: Int -> String -> Int`, which takes a base b and a string containing a base b numberw and returns the decimal integer. Implement the convenience functions `readBin :: String -> Int`, `readOct :: String -> Int`, and `readHex :: String -> Int`.

```
Prelude> map (\b -> readBase b "111") [2..16]
[7,13,21,31,43,57,73,91,111,133,157,183,211,241,273]
```

Submit your Haskell code as a plain text file.