Image Processing I – Exercises Week #2

Notes:

Try to solve the following questions (except for Exercise 2a/b) without the help of a computer. Enter your name, your student ID and your answers in this document and submit it as a Microsoft Word doc or pdf file to the following e-mail address: image.processing.jena@gmail.com.

Deadline for the submission is: Tuesday, November 17th, 2020, 8.00 a.m. (CET).

Name: .....

Student-ID: .....

Exercise 1: Conversion of decimal, binary, octal and hexadecimal numbers

a) Convert the following decimal numbers into the binary, octal and hexadecimal system:

Decimal	Binary	Octal	Hexadecimal
10			
20			
30			
40			
127			
256			
1984			
2020			

b) Convert the following (unsigned) binary numbers into the decimal, octal and hexadecimal system:

Binary	Decimal	Octal	Hexadecimal
0000 0010			
0000 0100			
0000 1100			
0001 0000			
0001 1000			
0011 1000			
0111 1111			
1000 0000			
1010 1010			
1111 1111			

c) Convert the following **signed** binary numbers into the decimal system assuming that negative numbers are represented by the two's complement

Binary (2's compl)	Decimal
0000 0010	
0000 0100	
1000 0000	
1010 1010	
1111 1111	

d) Fill the gaps in the following text by converting the binary and decimal numbers into the hexadecimal system:

Hi _____ (3501),

that's so _____ (1011 1010 1101). I am so tired, I am almost

_____ (57005). I need to go to _____ (1011 1110 1101) right now.

e) Most people can only count to 10 on their fingers. You, however, can now do better. If you regard each finger as one binary bit, with finger extended as 1 and finger touching the palm as 0, how high can you count using both hands?

Exercise 2: Data classes

a) Which is the lowest and highest value that can be stored in a Python float variable? How can you determine these values?

b) Which is the smallest positive value that can be stored in a variable of type of float? How can you determine this value?

Exercise 3: Arithmetic operations

a) Perform the following additions on (unsigned) binary numbers.

```
          00011101
      +   01101111
  ----------------
  Carry
    Sum
```

```
          01000001
      +   11111111
  ----------------
  Carry
    Sum
```

b) Perform the following additions on signed binary numbers. (Note that in the two's complement notation the most significant bit represents the number -128. Negative numbers in the two's complement notation can be added like positive numbers and yield the correct result, when the overflow is ignored.)

```
          00011101
      +   01101111
  ----------------
  Carry
    Sum
```

```
          01000001
      +   11111111
  ----------------
  Carry
    Sum
```

c) Perform the following additions on hexadecimal numbers.

```
        1234                 FFFF
      + ABCD               + 1111
Carry                Carry
  Sum                  Sum
```

d) A binary number can be subtracted by adding its two's complement, which is obtained by inverting all bits and adding 1. For example, using 1 byte (ie. 8 bits) the decimal number 6 is represented by 0000 0110. To find the signed binary number representing the decimal value -6, we first invert the bits: 1111 1001 (one's complement) and then add 1: 111 1010 (two's complement).

Now, subtract B = 0101 0101 and C = 0111 0101 from A = 0111 0000 by adding the two's complement of B and C to A.

```
   A     01110000           A     01110000
  -B                       -C
Carry                    Carry
  Sum                      Sum
```