Assignment 1: Data Representation

Starting point 2

LUONG TRAC DUC ANH - 103488117

Name: Luong Trac Duc Anh – ID: 103488117

Answers

Question 1:

a) The hexadecimal representation of the decimal value **675**

Divide the number by base and take the remainder

675/16 = 42 (remainder: 3)

42/16 = 2 (remainder: 10)

2/16 (remainder: 2)

$base^{position}$	16^2	16^1	16^0
0x	2	A	3

⇨ **0x2A3**

b) The 8-bit two's complement representation of the negative decimal value **-78**

Divide the positive counterpart of the number by base of 2^n and continue with the remainder

78/64 = 1 (remainder: 14)

14/8 = 1 (remainder 6)

6/4 = 1 (remainder 2)

2/2 = 1 (remainder: 0)

$base^{position}$	128	64	32	16	8	4	2	1
0b	0	1	0	0	1	1	1	0

- ⇨ Positive counterpart in binary: 0b01001110
- ⇨ One complement (flipping all the bits to make a negative number): 0b10110001
- ⇨ Two complement (add 1): **0b10110010**

c) The IEEE 754 single precision floating point representation of the decimal value **114.5**

114.5 = 114 + 0.5

Positive number => Sign = 0

Convert the decimal number of 114 into binary:

114/2 = 57 (remainder: 0)

57/2 = 28 (remainder: 1)

28/2 = 14 (remainder: 0)

14/2 = 7 (remainder: 0)

7/2 = 3 (remainder: 1)

3/2 = 1 (remainder: 1)

1/2 (remainder: 1)

$base^{position}$	128	64	32	16	8	4	2	1
0b	0	1	1	1	0	0	1	0

⇨ 114 (dec) = 0b01110010

Convert the fraction number of 0.5 into binary:

0.5*2 = 1

0.0*2 = 0

=> 0.5 (dec) = 0b0.1

=> 114.5 decimal in binary is 0b01110010.1000000

0b01110010.1000000 * 2^6 (move 6 decimal places) = 1.11001010000000

mantissa = 11001010000000

exponent = 127 + 6 = 133

Convert 133 into binary:

133/2 = 66 (remainder: 1)

66/2 = 33 (remainder: 0)

33/2 = 16 (remainder: 1)

16/2 = 8 (remainder: 0)

8/2 = 4 (remainder: 0)

4/2 = 2 (remainder: 0)

2/2 = 1 (remainder: 0)

1/2 (remainder: 1)

$base^{position}$	128	64	32	16	8	4	2	1
0b	1	0	0	0	0	1	0	1

- ⇨ 133 (dec) = 0b10000101

Answer in IEEE 754 single precision floating point: **0 10000101 11001010000000**

Question 2:

Given the 16-bit binary string: **0101 0111 1010 0110**

a) Convert to a hexadecimal value

Hexadecimal is 16-based, while binary is 2-based. With 2^4 = 6, we can group every 4 bits of the binary string to represent that value in hexadecimal.

Binary	0b	0101	0111	1010	0110
Hexadecimal	signed	5	7	10 = A	6

- ⇨ 0b 0101 0111 1010 0110 = **0x57A6**

b) Convert to two 8-bit unsigned integers as decimal values

- First number: 0b 0101 0111

$$(1 * 2^0) + (1 * 2^1) + (1 * 2^2) + (0 * 2^3) + (1 * 2^4) + (0 * 2^5) + (1 * 2^6) + (0 * 2^7) = \mathbf{87}$$

- Second number: 0b 1010 0110
  $$(0 * 2^0) + (1 * 2^1) + (1 * 2^2) + (0 * 2^3) + (0 * 2^4) + (1 * 2^5) + (0 * 2^6) + (1 * 2^7) = \mathbf{166}$$

c) Convert to two 8-bit unsigned integers (two's complement) as decimal values

- First number: 0b 0101 0111

The first digit is 0 => this is a positive number => does not change for two's complement

0b 0101 0111 = **87** (dec)

- Second number: 0b 1010 0110

The first digit is 1 => this is a negative number => subtract 1

1010 0110 – 1 = 1010 0101

Flip: 0101 1010

Convert to decimal: $(0 * 2^0) + (1 * 2^1) + (0 * 2^2) + (1 * 2^3) + (1 * 2^4) + (0 * 2^5) + (1 * 2^6) + (0 * 2^7) = 90$

Since this is a negative number, therefore it is **-90** (dec)

d) Convert to one half-precision (16 bit) floating point value as a decimal value

0 | 10101 | 1110100110

Sign = 0 => Positive number

Exponent bias = 2^{5-1} – 1 = 16 – 1 = 15

Exponent = 0b 10101 – Exponent bias = $(1 * 2^0) + (0 * 2^1) + (1 * 2^2) + (0 * 2^3) + (1 * 2^4) - 15 = 1 + 4 + 16 - 15 = 6$

Mantissa = 1.1110100110 = $(1 * 2^0) + (1 * 2^{-1}) + (1 * 2^{-2}) + (1 * 2^{-3}) + (0 * 2^{-4}) + (1 * 2^{-5}) + (0 * 2^{-6}) + (0 * 2^{-7}) + (1 * 2^{-8}) + (1 * 2^{-9}) + (0 * 2^{-10}) = \frac{979}{512} = 1.912109375$

Name: Luong Trac Duc Anh – ID: 103488117

Number = $(-1)^{sign} * Mantissa * 2^{exponent} = (-1)^0 * 1.912109375 * 2^6 =$ $\mathbf{122.375}$

Question 3:

c2 b9 f0 9d 9f ba 39 e2 a0 a1

Convert UTF-8 to binary

UTF-8	Hexadecimal	First digit to binary	Second digit to binary	Adding 2 digits together
U+00C2	C2	0xC = 0b1100	0x2 = 0b0010	1100 0010
U+00B9	B9	0xB = 0b1011	0x9 = 0b1001	1011 1001
U+00F0	F0	0xF = 0b1111	0x0 = 0b0000	1111 0000
U+009D	9D	0x9 = 0b1001	0xD = 0b1101	1001 1101
U+009F	9F	0x9 = 0b1001	0xF = 0b1111	1001 1111
U+00BA	BA	0xB = 0b1011	0xA = 0b1010	1011 1010
U+0039	39	0x3 = 0b0011	0x9 = 0b1001	0011 1001
U+00E2	E2	0xE = 0b1110	0x2 = 0b0010	1110 0010
U+00A0	A0	0xA = 0b1010	0x0 = 0b0000	1010 0000
U+00A1	A1	0xA = 0b1010	0x1 = 0b0001	1010 0001

UTF-8 number of bytes

From	To	# bytes	Bits used	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	1	7	0xxxxxxx			
U+0080	U+07FF	2	11	110xxxxx	10xxxxxx		
U+0800	U+FFFF	3	16	1110xxxx	10xxxxxx	10xxxxxx	
U+10000	U+10FFFF	4	21	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

- C2 and B9 are represented by a 2-byte UTF-8 single Unicode character as it has 110xxxxx format: <u>110</u> 00010 10 111001
- F0, 9D, 9F and BA are represented by a 4-byte UTF-8 single Unicode character as it has 11110xxx format: <u>11110</u> 000 10 011101 10 011111 10 111010
- 39 is represented by a 1-byte UTF-8 single Unicode character as it has 0xxxxxxx format: 0 0111001

- E2, A0, and A1 are represented by a 3-byte UTF-8 single Unicode character as it has 1110xxxx format: 1110 0010 10 100000 10 100001

- ⇨ First character: 0b0000 0000 1011 1001 => 00B9
- ⇨ Second character: 0b0001 1101 0111 1111 1010 => 1D7FA
- ⇨ Third character: 0b0000 0000 0011 1001 => 0039
- ⇨ Fourth character: 0b0010 1000 0010 0001 => 2821

Adding U+ to the beginning:

U+00B9, U+1D7FA, U+0039, U+2821