## HW 1-2

Due Aug 24 at 11:59pm Points 3	0 Questions 7	Available until Aug 24 at 11:59pm	Time Limit None	Allowed Attempts 2	
--------------------------------	---------------	-----------------------------------	-----------------	--------------------	--

## Instructions

Complete the following problems from chapter 1 of your text book. Solutions will be turned on after due date/time.

Average Time: 1 hour.

This quiz was locked Aug 24 at 11:59pm.

## **Attempt History**

	Attempt	Time	Score
KEPT	Attempt 2	1,338 minutes	29 out of 30
LATEST	Attempt 2	1,338 minutes	29 out of 30
	Attempt 1	7,364 minutes	22.2 out of 30

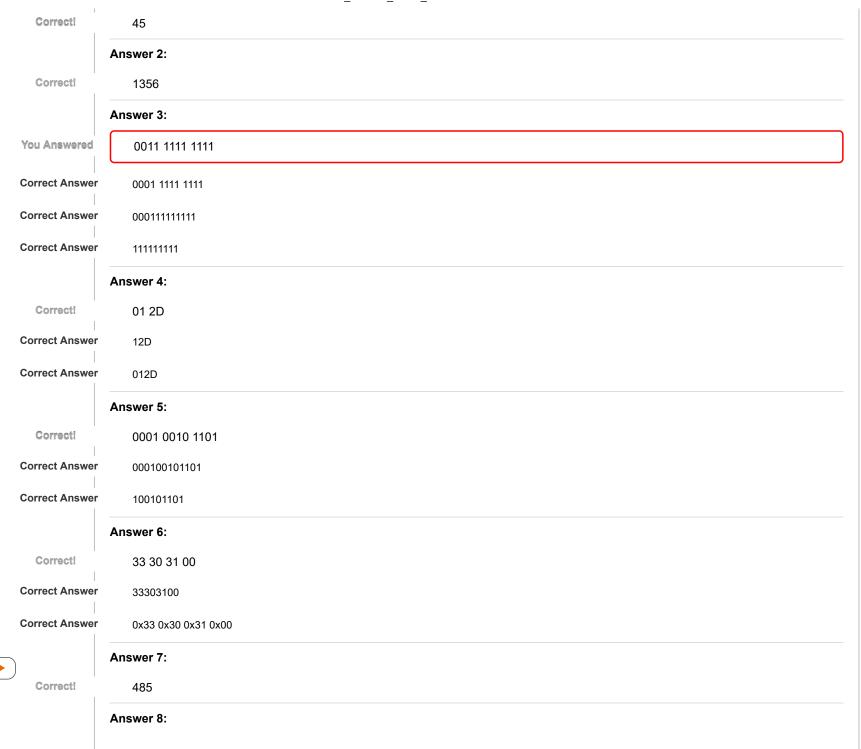
Score for this attempt: **29** out of 30 Submitted Aug 24 at 9:24pm This attempt took 1,338 minutes.

Question 1 9 / 9 pts

- 1.7 Perform the following:
  - (a) Convert  $101101_2$  to base ten.
  - (b) Convert  $1023_{10}$  to base nine.
  - (c) Convert  $1023_{10}$  to base two.



	(d) Convert 301 <sub>10</sub> to base 16.		
	(e) Convert 301 <sub>10</sub> to base 2.		
	(f) Represent 301 <sub>10</sub> as a null-terminated ASCII string		
	(write your answer in hexadecimal).		
	(g) Convert 3420 <sub>5</sub> to base ten.		
	(h) Convert 2314 <sub>5</sub> to base nine.		
	(i) Convert 116 <sub>7</sub> to base three.		
	(j) Convert 1294 <sub>11</sub> to base 5.		
(a) Base 10:	45		
(b) Base 9:	1356		
(c) Smallest	nibble (dont forget the spaces): 0011 1111 1111		
(d) Smallest	byte in hex (dont forget the space) 01 2D		
(e) smallest r	nibble boundary 0001 0010 1101		
(f) 33 30 31 (			
(g) 485			
(h) 411			
(i) 2022			
(j) 23201			
Answer 1:			



Correct!	411	
	Answer 9:	
Correct!	2022	
	Answer 10:	
Correct!	23201	

5 / 5 pts **Question 2** 1.8 Given the following binary string:  $01001001\ 01110011\ 01101110\ 00100111\ 01110100\ 00100000\ 01000001$ (a) Convert it to a hexadecimal string. (b) Convert the first four bytes to a string of base ten numbers. (c) Convert the first (little-endian) halfword to base ten. (d) Convert the first (big-endian) halfword to base ten. (e) If this string of bytes were sent to an ASCII printer or terminal, what would be printed? (a) Now you know the reason to put spaces in your answer. 49 73 6E 27 74 20 4 (b) w spaces after each decimal value (8 bits) (i.e. EF 10 = 239 16)  $\frac{73 \ 115 \ 110 \ 39}{115 \ 110 \ 39}$ 29513 (c) Remember a half-word is two bytes, first means leftmost. Give answer as base 10: 18803 (d) Remember a half-word is two bytes, first means leftmost. Give answer as base 10: Isn't Assembly Fun'

	Answer 1:
Correct!	49 73 6E 27 74 20 41 73 73 65 6D 62 6C 79 20 46 75 6E 3F 00
Correct Answer	49 73 6E 27 74 20 41 73 73 65 6D 62 6C 79 20 46 75 6E 3F
	Answer 2:
Correct!	73 115 110 39
	Answer 3:
Correct!	29513
	Answer 4:
Correct!	18803
	Answer 5:
Correct!	Isn't Assembly Fun?

1.9 The number 1,234,567 is stored as a 32-bit word starting at address
F043900016. Show the address and contents of each byte of the 32-bit
word on a

(a) little-endian system,
(b) big-endian system.

Each answer below requires all two digits (i.e. 0 = 00). Grading for this is strict!

The following fill in the blanks are for the little-endian system:

• What are the contents of address 0xF0439000?

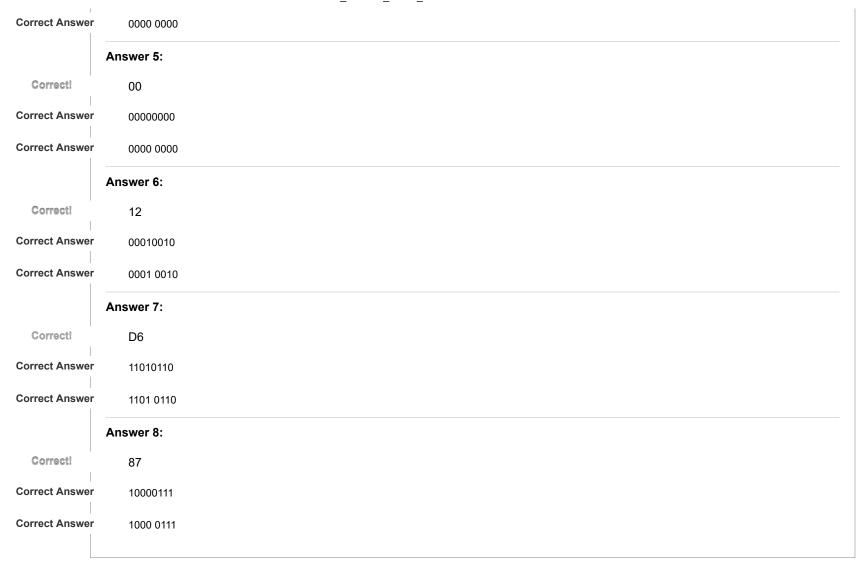
87

• What are the contents of address 0xF0439001?

D6

• What are the contents of address 0xF0439002?

	• What are the contents of address 0xF0439003?	00	
	The following fill in the blanks are for the big-endian	system:	
	• What are the contents of address 0xF0439000?	00	
	• What are the contents of address 0xF0439001?	12	
	• What are the contents of address 0xF0439002?	D6	
	What are the contents of address 0xF0439003?	87	
	Answer 1:		
Correct!	87		
Correct Answer	10000111		
Correct Answer	1000 0111		
	Answer 2:		
Correct!	D6		
Correct Answer	11010110		
Correct Answer	1101 0110		
	Answer 3:		
Correct!	12		
Correct Answer	00010010		
Correct Answer	0001 0010		
	Answer 4:		
Correct!	00		
Correct Answer	00000000		







1.10 The ISO/IEC 10646 standard defines 1,112,064 code points (glyphs). Each code point could be encoded using 24 bits, or three bytes. The UTF-8 encoding uses up to four bytes to encode a code point. Give three reasons why UTF-8 is preferred over a simple 3-byte per code point encoding.

20	HW 1-2: SC_2020FA_CS3B_16840 : COMPUTER ORGANIZATION AND ASSEMBLY LANGUAGE	
Correct!	Backwards compatible with ASCII	
Correct!	☑ Efficient code structure	
Correct!	Easily extended to include new languages	
	☐ Easier to memorize the character sets	
	Every possible character is currently represented	
	Question 5	2 / 2 pts
	1.11 UTF-8 is often referred to as Unicode. Why is this not correct?  Mark all that apply.	
	☐ Unicode is backwards compatible to ASCII where as UTF-8 is not backwards compatible.	
Correct!	Unicode is not backwards compatible to ASCII where as UTF-8 is backwards compatible.	
	UTF-8 was designed as a 16-bit encoding and Unicode uses variable width encoding.	

Question 6 5 / 6 pts

1.12 Skilled assembly programmers can convert small numbers between binary, hexadecimal, and decimal in their heads. Without referring to any tables or using a calculator or pencil, fill in the blanks in the following table:

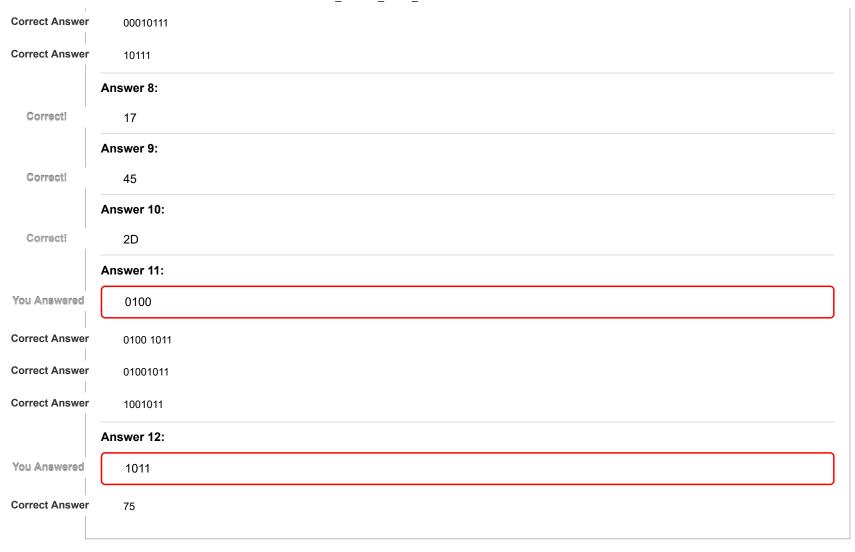
Unicode was designed as a 16-bit encoding and UTF-8 uses variable width encoding.

Binary	Decimal	Hexadecimal	
0101	5	5	



Correct!

/2020	HW 1-2: S	C_2020FA_CS3B_16840 : COMPUTER ORGANIZATION	ON AND ASSEMBLY LANGUAGE
	1010	10	А
	1100	12	С
	0001 0111	23	17
	0010 1101	45	2D
	0100	1011	4B
	Answer 1:		
Correct!	0101		
Correct Answer			
	Answer 2:		
Correct!	5		
Correct Answer	05		
	Answer 3:		
Correct!	10		
	Answer 4:		
Correct!	A		
Correct Answer	0A		
	Answer 5:		
Correct!	1100		
	Answer 6:		
Correct!	12		
	Answer 7:		
Correct!	0001 0111		



2 / 2 pts **Question 7** 1.13 What are the differences between a CPU register and a memory location? Choose all that apply.

https://canvas.ivc.edu/courses/44008/quizzes/190739?module\_item\_id=1742484

Correct!	Data in CPU registers can be used directly for computation.
	Computer memory consist of number of registers, each of which has a unique address.
Correct!	There are a small number of CPU registers.
	☐ The most common size of a register is 1 byte.
	☐ The ARM 32-bit processor family registers have variable size registers.
Correct!	Registers are used for storing temporary values, or temporary copies of data in memory.
	Registers are accessed via their memory addres.

Quiz Score: 29 out of 30

