Summary Exercise - Week 2

Due Oct 13 at 11:59pm   **Points** 16   **Questions** 16

Available after Oct 6 at 12am   **Time Limit** 360 Minutes   **Allowed Attempts** 2

Attempt History

	Attempt	Time	Score
KEPT	**Attempt 2**	70 minutes	16 out of 16
LATEST	**Attempt 2**	70 minutes	16 out of 16
	Attempt 1	85 minutes	14 out of 16

Score for this attempt: **16** out of 16
Submitted Oct 9 at 3:57am
This attempt took 70 minutes.

Question 1   1 / 1 pts

Convert the following string into its ASCII hex representation.
Don't use 0x or h with the hex values. The hex for "1+z" is 31 2B 7A

Studio

Correct!

> 53 74 75 64 69 6F

Correct Answers   53 74 75 64 69 6F

Question 2   1 / 1 pts
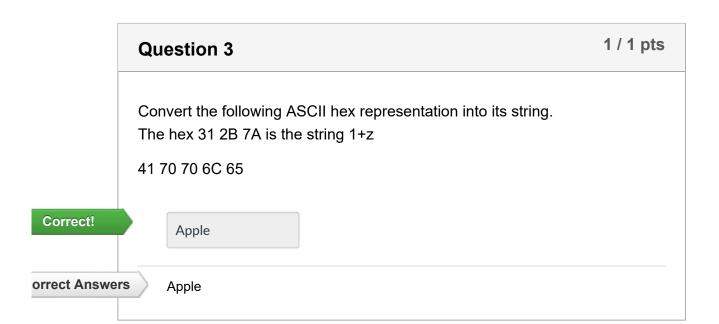
Convert the following string into its ASCII hex representation.
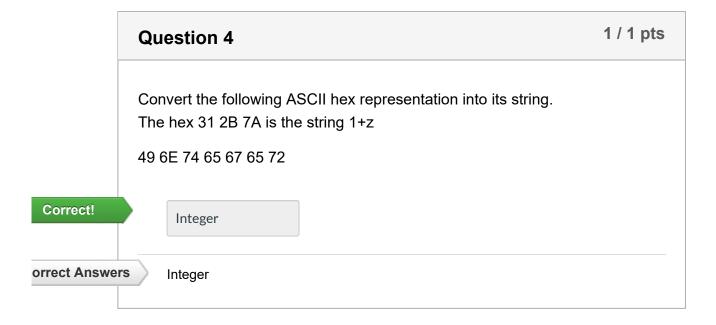Don't use 0x or h with the hex values. The hex for "1+z" is 31 2B 7A

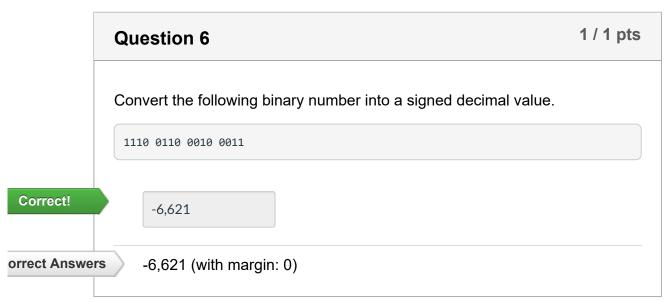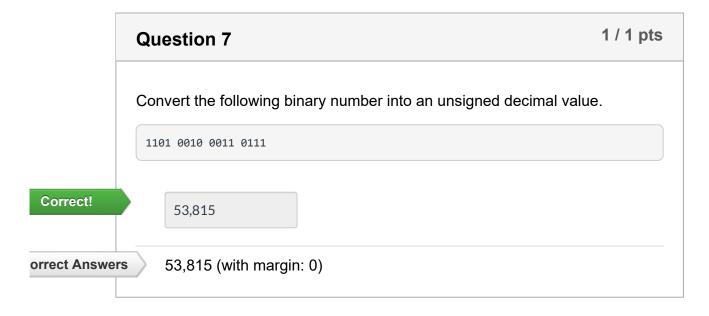Computer

43 6F 6D 70 75 74 65 72

43 6F 6D 70 75 74 65 72

Question 3

1 / 1 pts

Convert the following ASCII hex representation into its string.
The hex 31 2B 7A is the string 1+z

41 70 70 6C 65

Apple

Apple

Question 4

1 / 1 pts

Convert the following ASCII hex representation into its string.
The hex 31 2B 7A is the string 1+z

49 6E 74 65 67 65 72

Integer

Integer

Question 5

1 / 1 pts

Convert the following binary number into a signed decimal value.

```
1011 1001 0101 0000
```

Question 6

1 / 1 pts

Convert the following binary number into a signed decimal value.

```
1110 0110 0010 0011
```

Correct!	-6,621

orrect Answers	-6,621 (with margin: 0)

Question 7

1 / 1 pts

Convert the following binary number into an unsigned decimal value.

```
1101 0010 0011 0111
```

Correct!	53,815

orrect Answers	53,815 (with margin: 0)

Question 8

1 / 1 pts

Convert the following binary number into an unsigned decimal value.

```
1110 0110 0010 0011
```

58,915

58,915 (with margin: 0)

Question 9

1 / 1 pts

The following data segment starts at memory address 0x2300 (hexadecimal)
```
.data
printString BYTE "Assembly is fun",0
moreBytes BYTE 48 DUP(0)
dateIssued DWORD ?
dueDate DWORD ?
elapsedTime WORD ?
```

What is the hexadecimal address of dueDate?

- ○ 0x2310

Correct! ◉ 0x2344

- ○ 0x2340

- ○ 0x2368

Question 10

1 / 1 pts

The following data segment starts at memory address 0x1600 (hexadecimal)
```
.data
printString BYTE "MASM is fun",0
moreBytes BYTE 17 DUP(0)
dateIssued DWORD ?
dueDate DWORD ?
elapsedTime WORD ?
```

What is the hexadecimal address of dueDate?

- ◉ 0x1621

- ○ 0x161D

- ○ 0x1633

- ○ 0x160C

Question 11

1 / 1 pts

The following data segment starts at memory address 0x1000 (hexadecimal)

```
.data
printString BYTE "Do not add decimal to hex",0
someBytes WORD 27 DUP(0)
moreBytes BYTE 10, 20, 30, 40, 50, 60, 70, 80, 90
questionAddr DWORD ?
ignoreMe WORD ?
```

What is the hexadecimal address of questionAddr?

- ◉ 0x1059

- ○ 0x1050

- ○ 0x101A

- ○ 0x1089

Question 12

1 / 1 pts

After the following MASM code is executed:

```
mov    eax,19
mov    ebx,18
mov    ecx,17
add    eax,ebx
sub    eax,ecx
```

What is the value in the eax register (in decimal)? 20

What is the value in the ebx register (in decimal)? 18

What is the value in the ecx register (in decimal)? 17

Answer 1:

Correct!   20

Answer 2:

Correct!   18

Answer 3:

Correct!   17

Question 13

1 / 1 pts

After the following MASM code is executed:

```
mov    eax,212
mov    ebx,19
mov    edx,0
div    ebx
```

What is the value in the eax register (in decimal)? 11

What is the value in the ebx register (in decimal)? 19

What is the value in the edx register (in decimal)? 3

Answer 1:

11

Answer 2:

19

Answer 3:

3

Question 14

1 / 1 pts

Suppose that result is declared as DWORD, and the following MASM code is executed:

```
    mov eax,9
    mov ebx,6
    mov ecx,6
label5:
    add eax,ebx
    add ebx,2
    loop label5
    mov result,eax
```

What is the value stored in the memory location named result?

75

orrect Answers   75 (with margin: 0)

Question 15

1 / 1 pts

Please place the following steps of the instruction execution cycle in their proper order.

Step 1:

Fetch the instruction at th ▼

Step 2:

Increment the Instructior ▼

Step 3:

Decode the instruction in ▼

Step 4:

If the instruction requires ▼

Step 5:

Execute the instruction. ▼

Step 6:

If the output operand is in ▼

Question 16

1 / 1 pts

Select the pseudo-code that most closely corresponds to the following assembly code. Assume that the variables a, b, c, and d are initialized elsewhere in the program. Pay close attention to the conditional jumps, the corresponding pseudo code may surprise you!

```
.data
; General purpose variables
a       DWORD    ?
b       DWORD    ?
c       BYTE     ?
d       BYTE     ?
upperLevel    DWORD    18
lowerLevel    DWORD     3
```

```
; Strings
yes     BYTE        "Yes",0
no      BYTE        "No",0
maybe   BYTE        "Maybe",0

.code
    main PROC
    mov    eax, a
    cmp    eax, b
    jle    option1
    je     option2
    jmp    option3
option1:
    mov    edx, OFFSET yes
    call   WriteString
    jmp    endOfProgram
option2:
    mov    edx, OFFSET no
    call   WriteString
    jmp    endOfProgram
option3:
    mov    edx, OFFSET maybe
    call   WriteString
endOfProgram:
    exit
main ENDP
END main
```

```
    if (a < b)
        print (yes);
    else if (a == b)
        print (no);
    else
        print (maybe);
    ○
```

Correct!

```
    if (a < b) OR (a == b)
        print (yes);
    else
        print (maybe);
    ◉
```

```
if (a > b) OR (a == b)
    print (yes);
else
    print (maybe);
```

○

```
if (a > b)
    print (yes);
else if (a == b)
    print (no);
else
    print (maybe);
```

○

Note that the "jle" instruction jumps if the comparison is less than **or equal**, so the "je" (jump if equal) instruction never jumps.

Quiz Score: **16** out of 16