Summary Exercise - Week 2

Take the Quiz Again

Attempt History

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

Score for this attempt: **14** out of 16

Submitted Oct 8 at 6:49pm

This attempt took 85 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

Integer

Correct!

> 49 6E 74 65 67 65 72

Correct Answers   49 6E 74 65 67 65 72

Question 2   1 / 1 pts

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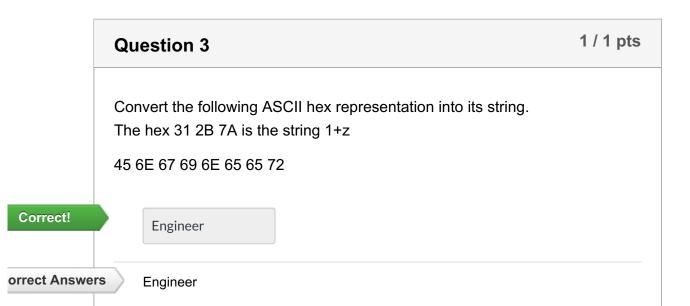Don't use 0x or h with the hex values. The hex for "1+z" is 31 2B 7A
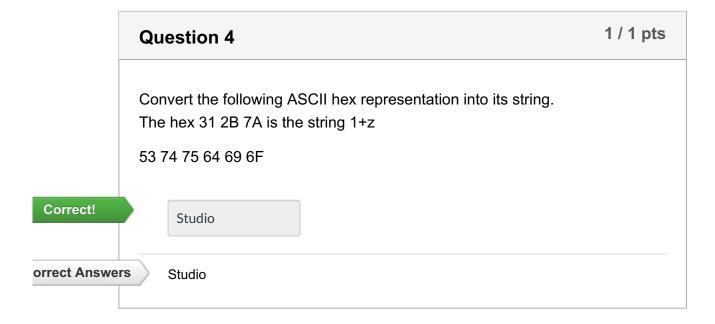
Complete

Question 3

1 / 1 pts

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

45 6E 67 69 6E 65 65 72

Engineer

Engineer

Question 4

1 / 1 pts

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

53 74 75 64 69 6F

Studio

Studio

Question 5

1 / 1 pts
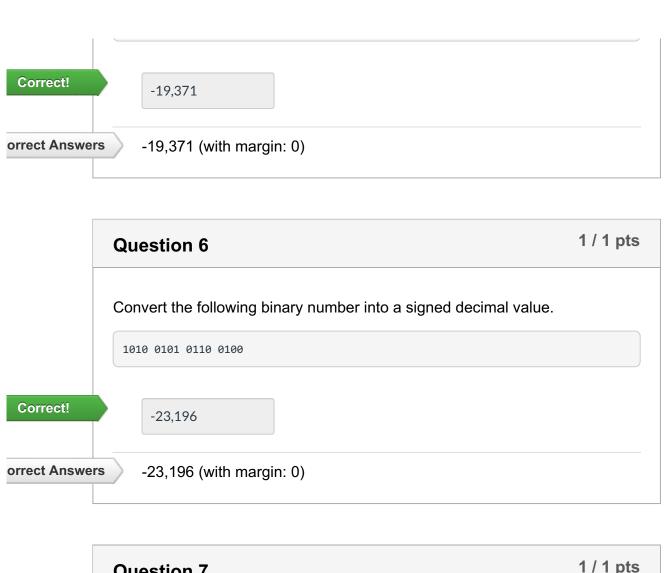
Convert the following binary number into a signed decimal value.

```
1011 0100 0101 0101
```

-19,371

-19,371 (with margin: 0)

Question 6   1 / 1 pts

Convert the following binary number into a signed decimal value.

```
1010 0101 0110 0100
```

-23,196

-23,196 (with margin: 0)

Question 7   1 / 1 pts

Convert the following binary number into an unsigned decimal value.

```
1110 1011 0001 1110
```

60,190

60,190 (with margin: 0)

Question 8   1 / 1 pts

Convert the following binary number into an unsigned decimal value.

```
1011 1110 0100 1011
```

Question 9

0 / 1 pts

The following data segment starts at memory address 0x1200 (hexadecimal)

```
.data
printString BYTE "Assembly is fun",0
moreBytes BYTE 25 DUP(0)
dateIssued DWORD ?
dueDate DWORD ?
elapsedTime WORD ?
```

What is the hexadecimal address of dueDate?

⚪ 0x1229

⚪ 0x1210

⚪ 0x122D

⦿ 0x1245

Question 10

0 / 1 pts

The following data segment starts at memory address 0x1100 (hexadecimal)

```
.data
printString BYTE "MASM is fun",0
moreBytes BYTE 48 DUP(0)
dateIssued DWORD ?
dueDate DWORD ?
```

```
elapsedTime WORD ?
```

What is the hexadecimal address of dueDate?

- ○ 0x1140

- ◉ 0x1164

- ○ 0x110C

- ○ 0x113C

Question 11

1 / 1 pts

The following data segment starts at memory address 0x2100 (hexadecimal)
```
.data
printString BYTE "Do not add decimal to hex",0
someBytes WORD 56 DUP(0)
moreBytes BYTE 10, 20, 30, 40, 50, 60, 70, 80, 90
questionAddr DWORD ?
ignoreMe WORD ?
```

What is the hexadecimal address of questionAddr?

Correct!

- ◉ 0x2193

- ○ 0x218A

- ○ 0x2247

- ○ 0x211A

Question 12

1 / 1 pts

After the following MASM code is executed:

```
mov     eax,127
mov     ebx,43
mov     ecx,27
add     eax,ebx
sub     eax,ecx
```

What is the value in the eax register (in decimal)?   143

What is the value in the ebx register (in decimal)?   43

What is the value in the ecx register (in decimal)?   27

Answer 1:

Correct!   143

Answer 2:

Correct!   43

Answer 3:

Correct!   27

Question 13   1 / 1 pts

After the following MASM code is executed:

```
mov     eax,96
mov     ebx,13
mov     edx,0
div     ebx
```

What is the value in the eax register (in decimal)?   7

What is the value in the ebx register (in decimal)? `13`

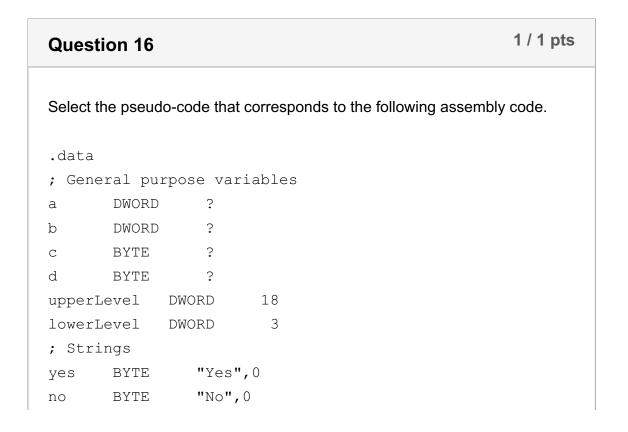What is the value in the edx register (in decimal)? `5`

Answer 1:

7

Answer 2:

13

Answer 3:

5

Question 14

1 / 1 pts

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

```
    mov eax,7
    mov ebx,5
    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?

67

---

67 (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 ir ▼

Question 16

1 / 1 pts

Select the pseudo-code that corresponds to the following assembly code.

```
.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, 0
    cmp    eax, lowerLevel
    jne    option1
    jmp    option2
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
```

print (no);

print (maybe);

Correct!

print (yes);

print (eax);