After executing the following instruction sequence, what is the value of AL, in hexadecimal

mov al,86h  
or  al,42h

**Correct!**



**Correct Answers**

0xC6

C6h

0C6h

xC6

C6

After executing the following instruction sequence, what is the value of AL, in binary:

mov al,11001111b  
and al,00101011b

**Correct!**



**Correct Answers**

00001011b

0000 1011

0000 1011b

00001011

After the following instruction sequence, show the values of the Carry  , Zero  , and Sign  flags

mov al,6  
cmp al,5

What is a single instruction that clears bits 0, 3, and 4 in the AL register?

or al,11110010b

or al,11100110b

and al,11100110b

and al,11001110b

or al,00011001b

and al,11100101b

What will be the hexadecimal value of AL after these instructions execute?

mov al,94h  
xor al,37h

 3Fh

not listed

A3h

B7h

What will be the hexadecimal value of AL after these instructions execute?

mov al,3Ch  
or  al,82h

  BCh

not listed

3Eh

BEh

After executing the following instruction sequence, what is the value of AL, in binary:

mov al,94h  
xor al,37h

**Correct!**



**Correct Answers**

10100011b

1010 0011b

10100011

1010 0011

After executing the following instruction sequence, what is the value of AL, in binary:

mov al,00111100b  
or  al,82h

**Correct!**



**Correct Answers**

10111110b

10111110

1011 1110b

1011 1110

After executing the following instruction sequence, what is the value of AL, in binary:

mov al,4Bh  
and al,6Ch

**Correct!**



**Correct Answers**

01001000b

0100 1000b

01001000

0100 1000

After executing the following instruction sequence, what is the value of AL, in hexadecimal

mov al,4Bh  
and al,74h

**Correct!**



In what register will the remainder of the following instruction be found?

DIV BX

**Correct!**



What is the value of the Carry flag after the following instructions?

.data  
val1 WORD 2000h  
val2 WORD 0100h  
.code  
mov ax,val1  
mul val2

**Correct!**



The Irvine32 library call GetMseconds returns

number of system microseconds that have elapsed since midnight

number of system microsoftseconds that have elapsed since midnight

number of system midnight seconds that have elapsed since the epoch

number of system milliseconds that have elapsed since midnight

number of system milliseconds that have elapsed since the epoch

number of system microseconds that have elapsed since the epoch

What is the value of the Carry flag after the following instructions?

mov al,5h  
mov bl,10h  
mul bl

**Correct!**



What is the value of the Overflow flag after the following instructions?

mov eax,+4823424  
mov ebx,-423  
imul ebx

**Correct!**



In what register will the remainder of the following instruction be found?

DIV EBX

**Correct!**



Unlike the MUL instruction, IMUL preserves the sign of the product.

**Correct!**

True

False

Where is the result of the following operation stored?

MUL EBX

**Correct!**

  EDX:EAX

  DX:AX

  EDX

  EBX

  EAX

  EBX:EAX

What is the value of the Carry flag after the following instructions?

mov eax,12345h  
mov ebx,1000h  
mul eb

**Correct!**



In what register will the quotient of the following instruction be found?

DIV EBX

**Correct!**



Identify the sizes of the sign  , exponent  , and significand  for a Single Precision x86 floating point number.

In the x86 Floating-Point, a decimal number contains three components: a sign, a significand, and an exponent.

**Correct!**

  True

False

Select the correct **EVEN** parity 12 bit Hamming code value for the unsigned integer value 202

  1110  1001  1010

1010  1001  1010

1011  1000  1010

0011  1000  1010

1110  1011  1010

0011  1001  1011

 Select the correct **ODD** parity 12 bit Hamming code value for the unsigned integer value 235

0111  1101  1011

1111  1101  1011

 1010  1100  1011

0010  1100  1011

0111  1100  1011

1010  1101  1011

1100  1101  0110 is an **EVEN** parity 12 bit Hamming code that contains a single-bit error.  
What is the corresponding **uncorrupted** Hamming code?

  1100  1001  0110

  0100  1101  0110

  1100  1101  1110

1101  1101  0110

1010  1111  0011 is an **EVEN** parity 12 bit Hamming code that contains a single-bit error.  
What is the corresponding **uncorrupted** Hamming code?

1010  1110  0011

1110  1111  0011

1010  1111  1011

1010  0111  0011

 Which of the following selections contain instructions that jump to label L4 **only** if bits 1, 2, *and* 3 are all set in the DL register?

|  |  |
| --- | --- |
|  | and dl,0Eh cmp dl,0Eh je  L4 |
|  | test dl,0Eh jnz  L4 |

|  |  |
| --- | --- |
|  | and dl,07h cmp dl,07h je  L4 |
|  | and  dl,0Eh test dl,0Eh jnz  L4 |

What will be the hexadecimal value of AL after these instructions execute?

mov al,0CFh  
and al,2Bh

|  |  |
| --- | --- |
|  | 06h |
|  | not listed |

|  |  |
| --- | --- |
|  | 0Bh |
|  | EAh |

The MUL (unsigned multiply) instruction can have operands of different sizes.

|  |  |
| --- | --- |
|  | True |
|  | False |

The x86 instruction set supports three formats for the IMUL instruction: one operand, two operands, and three operands.

|  |  |
| --- | --- |
|  | True |
|  | False |

Identify which of the following are correct formats for the DIV instruction. (Check all that apply)

|  |  |
| --- | --- |
|  | DIV imm8 |
|  | DIV imm32 |

|  |  |
| --- | --- |
|  | DIV mem16 |
|  | DIV reg, mem |

|  |  |
| --- | --- |
|  | DIV reg |
|  | DIV mem8 |

|  |  |
| --- | --- |
|  | DIV reg, reg |
|  | DIV mem32 |

|  |  |
| --- | --- |
|  | DIV imm16 |

Even in older x86 processors, there was an insignificant difference in performance between multiplication by bit shifting versus multiplication using the MUL and IMUL instructions.

|  |  |
| --- | --- |
|  | True |
|  | False |

What will be the final values of CX and DX when the following code executes?

.data  
array  SWORD 8,2,3,5,-4,6,0,4  
.code  
    mov cx,1  
    mov esi,2  
    mov ax,array[esi]  
    mov bx,array[esi+4]  
    cmp ax,3  
    jae L2  
    cmp bx,4  
    jb  L1  
    jmp L3  
L1: mov cx,4  
L2: mov dx,5  
    jmp L4  
L3: mov dx,6  
L4:

  CX = 1, DX = 6

CX = 4, DX = 5

CX = 1, DX = 5

CX = 4, DX = 6

After the following instruction sequence, show the values of the Carry  , Zero  , and Sign  flags

mov al,00110011b  
test al,2

What is a single instruction that inverts bits 5 and 6 in BL without changing any other bits?

xor bl,1100000b

and bl,1100000b

xor bl,0011111b

or bl,1100000b

Which answer choice shows the correct values of the Carry, Zero, and Sign flags after the following instructions execute?

mov al,6  
cmp al,5

  CF = 1, ZF = 1, SF = 0

 CF = 1, ZF = 0, SF = 0

CF = 0, ZF = 0, SF = 0

CF = 1, ZF = 0, SF = 1

Suppose EAX, EBX, and ECX contained three unsigned integers. Which of the following code excerpts would display the largest of the three integers?

 cmp eax,ebx  
    jb  L1  
    mov eax,ebx  
L1: cmp eax,ecx  
    jb  L2  
    mov eax,ecx  
L2: call WriteInt



    cmp eax,ecx  
    jae L1  
    mov eax,ebx  
L1: cmp eax,ebx  
    jae L2  
    mov eax,ecx  
L2: call WriteInt



    cmp eax,ebx  
    jae L1  
    mov eax,ebx  
L1: cmp eax,ecx  
    jae L2  
    mov eax,ecx  
L2: call WriteInt



    cmp eax,ebx  
    jnae L1  
    mov eax,ebx  
L1: cmp ecx,eax  
    jnae L2  
    mov eax,ecx  
L2: call WriteInt

After the following instruction sequence, show the values of the Carry  , Zero  , and Sign  flags

mov al,5  
cmp al,7

Select the correct **ODD** parity 12 bit Hamming code value for the unsigned integer value 154

1010  0011  1010

1011  0011  1010

0101  0010  1010

0111  0010  1010

0111  0000  1010

1010  0010  1010

Select the correct **ODD** parity 12 bit Hamming code value for the unsigned integer value 55

  1100  0111  0111

0001  0110  0111

1001  0111  0111

0001  0111  0111

1100  0110  0111

0100  0110  0111

1010  0111  1010 is an **EVEN** parity 12 bit Hamming code that contains a single-bit error.  
What is the corresponding **uncorrupted** Hamming code?

1010  0111  0010

1011  0111  1010

1010  0101  1010

1110  0111  1010

 0011  1000  1011 is an **EVEN** parity 12 bit Hamming code that contains a single-bit error.  
What is the corresponding **uncorrupted** Hamming code?

0011  1000  1010

0011  1001  1011

0111  1000  1011

0011  0000  1011