7-9. Suppose AL, BL, and CL contain ASCII for some alphabetical character.

Note that ASCII for 'A' and 'B' are 41h and 42h, and ASCII for 'a' and 'b' are 61h and 62h.

7. Give a logical instruction to change the character in AL to lower-case if it is upper-case. (No change if it is already lower-case.)

Instruction: and AL, 1101 1111b If AL had 'C', after value in AL: 63h (in hex)

8. Give a logical instruction to change the character in BL to upper-case if it is lower-case. (No change if it is already upper-case.)

Instruction: or BL, 0010 0000 If BL had 'e', after value in BL: 45h (in hex)

9. Give a logical instruction to change the character in CL to upper-case if it is lower-case and change it to lower if it is upper.

Instruction: xor CL, 0010 0000 If CL had 'G', after value in CL: 67h (in hex)

43h

7. 'C' = 0100 0011

'c' = 0110 0011

45h

8. 'e' = 0110 0101

'E' = 0100 0101

47h

9. 'G' = 0100 0111

'g' = 0110 0111

10. Give an "and" instruction that replaces a number in DX with the remainder of the number when divided by 64.

Instruction: and dx, 0011 1111 If DX had 541d, after value in DX: 1Dh (in hex)

Show work.

11. Suppose the register CH contains an unsigned number whose value is between 0 and 9.

Note that ASCII for the characters, '0', '1' and '9' are 30h, 31h, and 39h.

Write a logical instruction to convert the value in CH to the ASCII for the decimal digit corresponding to the value.

Instruction: add ch, 0011 0000b If CH had 4d, after value in CL: 7Dh (in hex)

10. 541d = 0010 0001 1101

and 0011 1111

0001 1101

8/4/2/1

1 D

11. 4d = 0100 1101₂

add 0011 0000₂

0111 1101

7 D

Part I.

1-3. For each problem, give the value in **hex** after the instructions have been executed.

	Instructions	After	Show work

1	mov ax, 0AF75h
	mov bx, 00FFh
	and ax, bx
		AX: 0075h SF: 0	

2	mov bx, 0AF75h
	mov cx, 0FF0h
	or bx, cx
		BX: AFF5h SF: 1	

3	mov cx, 0AF75h
	mov dx, 0F00Fh
	xor cx, dx
		CX: 5F7Ah SF: 0	

1. 0AF75h = 0000 1010 1111 0111 0101

00FFh = and 0000 0000 1111 1111

SF: 0 → 0000 0000 0111 0101

0 0 7 5

8/4/2/1

2. 0AF75h = 0000 1010 1111 0111 0101

0FF0h = or 0000 1111 1111 0000

1011 1111 1111 0101

SF: 1 → A F F 5

3. 0AF75h = 0000 1010 1111 0111 0101

0F00Fh = xor 1111 0000 0000 1111

0101 1111 0111 1010

SF: 0 → 5 F 7 A

4-6. Suppose a label **Flags**, represents 8 true/false values:

```
section .data
Flags  db 0F0h
```

4. Give a logical instruction to store "true" in bits 1 and 4 without altering the other bits, and give the after value in hex.

Instruction: or byte [Flags], 00010010b After value in Flags: F2h

5. Give a logical instruction to store "false" in bits 2 and 5 without altering the other bits, and give the after value in hex.

Instruction: and byte [Flags], 11011011b After value in Flags: D0h

6. Give a logical instruction to change truth values in bits 3 and 6 without altering the other bits, and give the after value in hex.

Instruction: xor byte [Flags], 01001000b After value in Flags: B8h

4. 0F0h

7 6 5 4 3 2 1 0

1 1 1 1 0 0 0 0

↓ or

1 1 1 1 0 0 1 0

0F2h

5.

1 1 0 1 0 0 0 0

↓

1 1 0 1 0 0 0 0

D0h

6.

1 1 1 1 0 0 0 0

↓

1 0 1 1 1 0 0 0

B8h

A = 10

B = 11

C = 12

D = 13

E = 14

F = 15

8/4/2/1