

HW#4Q1

1. NOT DX

DX = 1E0Ch; flags are unchanged

2. AND AL, 45h

AL = 41h; O=0, S=0, Z=0, A=?, P=1, C=0

3. OR BX, CX

BX = C7F7h; O=0, S=1, Z=0, A=?, P=0, C=0

4. XOR AX, CX

AX = 2982h; O=0, S=0, Z=0, A=?, P=1, C=0

5. TEST CX, 8820h

CX is not changed. C6F1 AND 8820 = 8020h

O=0, S=1, Z=0, A=?, P=0, C=0

6. SAL DX, CL

DX = 0000h; O=unch, S=0, Z=1, A=?, P=1, C=0

7. ROL DX, 1

DX = C3E7h; O=0, C=1, other flags are unchanged

8. ROL DX, CL

Since CL = Fh, this is equivalent to 24h

Note that rotating a register with a value that is multiple of 16 in CL does not change the value in DX. So, this will produce the same value on DX as ROL DX, 1.

The only difference is the effect on the flags.

$DX = C3E7h$ ;  $O = \text{unchanged}$ ,  $C = 1$ , other flags are unchanged

9. ROR  $DX, CL$

$DX = F0F9h$ ;  $O = \text{unchanged}$ ,  $C = 1$ , other flags are unchanged

10. RCR  $DX, 1$

$DX = F0F9h$ ;  $O = 0$ ,  $C = 1$ , other flags are unchanged

11. SAR  $DX, CL$

$DX = FFFFh$ ;  $O = \text{unchanged}$ ,  $S = 1$ ,  $Z = 0$ ,  $P = 1$ ,  $C = 1$

12. SHR  $DX, 1$

$DX = 70F9h$ ;  $O = 1$ ,  $S = 0$ ,  $Z = 0$ ,  $P = 1$ ,  $C = 1$

Q2

1. Count = 5

Number of times the loop instruction is executed is 5.

2. Count = -5

Number of times the loop instruction is executed is  $FFFFh = 2^{16} - 5 = 65531$  times

Q3

(i)

```
Mov AH, AL ; AH = AL
Mov CL, 2
SAL AL, CL ; AL = 4AL
Mov BL, AL ; BL = 4AL
SAL AL, 1 ; AL = 8AL
ADD BL, AL ; BL = 12AL
SAL AL, CL ; AL = 32AL
ADD AL, BL ; AL = 44AL
SUB AL, AH ; AL = 43AL
```

(ii)

```
Mov BL, AL
Mov CX, 42
```

Again:

```
Add AL, BL
Loop Again
```

Q4

(i)

The code swaps or exchanges the content of the first 100 bytes between Table1 and

Table2

(ii)

Without using the loop instruction

Next:

```
Mov SI, offset Table1
Mov DI, offset Table2
Mov BX, 0
Mov AL, [SI+BX]
Mov AH, [DI+BX]
Mov [SI+BX], AH
Mov [DI+BX], AL
INC BX
CMP BX, 100
JNE Next
```

25

```
.Model SMALL
.STACK 100
.DATA
I db ?
J db ?
F db ?
A dw ?
B dw ?
C dw ?
```

• CODE

• STARTUP

```
Mov I, 10
MOV J, -10
MOV F, 1
MOV A, 100
MOV B, 8
MOV C, 1000
MOV AX, A
MOV BX, B
MOV CX, C
MOV DL, J
Next? CMP DL, J
      JL Ewhile
      CMP F, 1
      JNE Ewhile
      SUB DL, 2
      SAL BX, 1
      ADD AX, BX
      CMP AX, 800
      JL Next
```

Ewhile?

```
Mov F, 0
JMP Next
SUB CX, AX
MOV C, CX
MOV A, AX
MOV B, BX
MOV I, DL
• EXIT
END
```