

Review Question

Short Answer

1. A : AL = 64h / B : AL = EAh / C : AL = FDh / D : AL = A9h
2. A : AL = 9Ah / B : AL = 6Ah / C : A9h / D : 3Ah
3. DX = 0002h / AX = 2200h
4. AX = 0306h
5. EAX = 0001 2340h / EDX = 0000 0000h
6. 0으로 나눔 / 오버플로우 예외 발생
7. BX = 066h
8. 몫 : 0108 0000 0000 0033 3000h / 나머지 : 20h
9.
 1. Mov esi, val1 : 데이터값을 로드함 OFFSET 사용 필요
 2. Mov al, [rsi] -> sbb al, [rdi] -> mov [rbx], al처럼 al결과를 result에 저장하는 방식으로 수정
10. RAX = 0004 080C 1014 000h

Algorithm Workbench

1.

Shl eax, 16

Sar eax, 16

2.

mov ah, al

shr al, 1

test ah, 1

jz no_set

or al, 80h

no_set:

3.

Shl eax, 4

4.

Shr ebx, 2

5.

Rol dl, 4

6.

SHLD dx, ax, 1 = ax의 최상위비트를 dx최하위 비트로 옮겨가게 왼쪽으로 쉬프트 한칸

7.

mov al, byteArray

shr al, 1

mov byteArray, al

mov al, byteArray+1

rcr al, 1

mov byteArray+1, al

mov al, byteArray+2

rcr al, 1

mov byteArray+2, al

8.
clc
mov ax, wordArray+4
shl ax,1
mov wordArray+4,ax

mov ax, wordArray+2
rcl ax,1
mov wordArray+2,ax

mov ax, wordArray
rcl ax,1
mov wordArray,ax

9.
.data
val1 WORD ?
.code
mov ax,5
mov bx,3
mul bx
mov val1,ax

10.
.data
val1 WORD ?
.code
mov ax,276
mov bl,10
div bl
movzx ax,al
mov val1,ax

11.
mov eax,val2
mul val3
mov ebx,val4
sub ebx,3
div ebx
mov val1,eax

12.
mov eax,val2
cdq
idiv val3
mov ecx, eax
mov eax, val1
add eax, val2
imul ecx
mov val1, eax

13.

showDecimal8 PROC

```
xor ah,ah  
mov bl,10  
div bl  
add al,'0'  
call WriteChar  
mov al,ah  
add al,'0'  
call WriteChar  
ret
```

showDecimal8 ENDP

14.

AX = 0108h, AF = CF = 1

15.

```
mov eax,n      ; eax = n  
mov ebx,y      ; ebx = y  
mov ecx,ebx  
sub ecx,1      ; ecx = y - 1  
and eax,ecx    ; eax = n mod y  
mov x,eax
```

16.

```
; 입력: EAX = signed integer  
; 출력: EAX = |EAX|  
mov edx, eax  
sar edx,31      ; edx = 0 (양수) 또는 -1 (음수)  
mov ecx,edx  
xor ecx,0FFFFFFFh ; ecx = edx ^ -1 (0 -> -1, -1 -> 0)  
add ecx,1        ; ecx = -edx    (0 -> 0, -1 -> 1)  
xor eax,edx      ; x ^ s  
add eax,ecx      ; (x ^ s) - s 를 ADD로 구현
```