



1. Given the following array write a recursive procedure to replace each of the following elements in the array with its mathematical twice value. [4 Points]

array WORD 1, -2, -3, 4, -5, 6, -7, 8, -9, -10

2. In the following instructions sequence, show the resulting value of **AL/AX** where indicated, in hexadecimal. [4 Points]

```
mov     al, 0F9h
not      al
rol      al, 2                      ; al = _____
```

```
mov     al, 77h
and      al, 74h
clc
rcr      al, 3                      ; al = _____
```

```
mov     al, 90h
test     al, 0ffh
xor      al, 0ffh
sar      al, 4                      ; al = _____
```

```
mov     ax, 7079h
mov     cx, 13f1h
shrd     ax, cx, cl                 ; ax = _____
```

3. Write the equivalent assembly code for following procedure and draw out the stack frame. Do not use ENTER/LEAVE, USES, and LOCAL directives. (Assume ESP = 00FF 3232h, and EBP = 1000 1000h, initially). [4 Points]

```
void main(){
    char a[]={100,81,64,49,36,25,16,9,4,1};
    roots(a);
}
```

```
void roots(char arr[]){
    char count = 10;
    for(int i=0;i<10;i++){
        arr[i] = arr[i]/count;
        count--;
    }
}
```

```
main    proc
        push    ebp
        mov     ebp,esp
        sub     esp, 13
        mov     [ebp-4], 100
        mov     [ebp-5], 81
        .
        .
        mov     [ebp-13], 1
        INVOKE  roots, offset [ebp-4]
        add     esp,4           ;cleaning the past arguments
        mov     esp, ebp       ;cleaning local data
        pop     ebp
        ret
main    endp

roots   proc, p:ptr byte

        push    ebp
        mov     ebp,esp
        mov     esi, p         ;pointer to x[]
        sub     esp, 4
        mov     [ebp-4], 10    ;count value
        mov     cx,10

        L1:     mov     ax,0
                Mov     al, [esi]
                div     [ebp-4]
                mov     [esi], al
                sub     esi, 1
                sub     [ebp-4], 1
        loop    L1

        mov     esp,ebp
        pop     ebp
        ret
roots   ENDP
```

00FF 322E	Ret address(system)	MAIN'S STACK FRAME
00FF 322A	1000 1000h (ebp pushed)	
00FF 3226	100	
00FF 3225	81	
00FF 3224	64	
00FF 3223	49	
00FF 3222	36	
00FF 3221	25	
00FF 3220	16	
00FF 321F	9	
00FF 321E	4	ROOTS' STACK FRAME
00FF 321D	1	
00FF 321C	00FF 3226 (Argument)	
00FF 3218	Return Address(main)	
00FF 3214	00FF 322Ah (ebp pushed)	
00FF 3210	10 (count)	

