



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

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

2. Given that following code is some snippet from an encryption algorithm, where ax = 09h, dx = 70h, and cx = 4. Write the encrypted value in AX, and correct decryption code for the following snippet. [4 Points]

```
not    ax
rol    dl, cl
rcl    al, cx
xor    cx, 2
ror    al, cx
stc
rcl    al, 5
xchg   al, ah
add    al, 7
add    al, dl
```

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 = 11FF CD40h, and EBP = 7FED 7FEDh, initially). [4 Points]

```
void main(){
    char a[]={1,2,3,4,5};
    cubes(a);
}
```

```
void cubes(char arr[]){
    for(int i=0;i<5;i++){
        arr[i] = arr[i] * arr[i] * arr[i];
    }
}
```

```
main  proc
      push  ebp
      mov   ebp,esp
      sub   esp, 8

      mov   [ebp-4], 1
      mov   [ebp-5], 2
      mov   [ebp-6], 3
      mov   [ebp-7], 4
      mov   [ebp-8], 5
      INVOKE cubes, offset [ebp-4]

      add   esp, 4      ;cleaning passed arguments
      mov   esp, ebp    ;cleaning local data
      pop   ebp
      ret

main  endp
```

11FF CD3C	Ret address(system)	MAIN'S STACK FRAME
11FF CD38	7FED 7FED (ebp)	
11FF CD34	1	
11FF CD33	2	
11FF CD32	3	
11FF CD31	4	
11FF CD30	5	Squares' Stack Frame
11FF CD2F	11FF CD34 (Argument)	
11FF CD2B	Ret Address (main)	
11FF CD27	C101 00F8 (ebp)	

```
squares PROC, p:ptr byte
      push  ebp
      mov   ebp,esp
      mov   esi, p      ;pointer to x[]
      mov   cx, 5

      L1:   mov   ax,0
            mov   al, [esi]
            movzx bx, al
            mul   al
            mul   bx
            mov   [esi], al
            sub   esi, 1

      loop  L1
      pop   ebp
      ret

squares  ENDP
```