



1. Given the following array write a recursive procedure to replace each of the array element with its mathematical square root value. [4 Points]

squares WORD 1, 4, 9, 16, 25, 36, 49, 64, 81, 100

2. Given that following code is some snippet from an encryption algorithm, where ax = 90h, dx = 07h, 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 Calling conventions, ENTER/LEAVE, USES, and LOCAL directives. (Assume ESP = C101 0100h, and EBP = 0000 F700h, initially). [4 Points]

```
void main()
{
    char x[5]= {2,3,4,5};
    squares(x);
}
```

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

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

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

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

C101 00FC	Ret address(system)	MAIN'S STACK FRAME
C101 00F8	0000 F700 (ebp)	
C101 00F4	2	
C101 00F3	3	
C101 00F2	4	
C101 00F1	5	
C101 00F0	6	
C101 00EC	C101 00F4 (Argument)	Squares' Stack Frame
C101 00E8	Ret Address (main)	
C101 00E4	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]
            mul   al
            mov   [esi], al
            sub   esi, 1

      loop  L1
      pop   ebp
      ret
squares  ENDP
```