



1. Elaborate the following directive:

(4 Points)

`.model COMPACT, C, FARSTACK`

Answer: The directive creates one code segment and multiple data segments where stack segment is maintained outside the data segments. C calling convention is used to clean up the stack.

2. Encode the given instructions, provide hex-decimal values only

(8 Points):

`ADD CX, 13h`

`03 + 001 ← 13 00`

**= 04 13 00 h**

`AND DL, CL`

`0010 0100 11 001 010`

**= 24 CA**

`POP BX`

`58 h + 03 h`

**= 5B h**

`PUSH 4321 h`

`50 h ← 21 43`

**= 50 2143 h**

MOD=11			Instruction	Opcode
R/M	W = 0	W = 1	MOV	1000 10dw
000	AL	AX	ADD	0000 00dw
001	CL	CX	SUB	0010 10dw
010	DL	DX	OR	0000 10dw
011	BL	BX	XOR	0011 00dw
100	AH	SP	AND	0010 01dw
101	CH	BP	PUSH (16 bits)	50h
110	DH	SI	POP (16 bits)	58h
111	BH	DI		

3. Elaborate through an example, how does **SCASW** differ from **MOVSW**? (4 Points)

Answer: **SCASW** instruction compares a value in **AX** to a word addressed by **EDI** whereas **MOVSW** copies a WORD operand pointed to by **ESI** to the memory location pointed to by **EDI**.

e.g. **FREE RESPONSE**

4. Using string primitive instructions, replace each element of given array by its mathematical square, assume any valid type for array1. (4 Points)

array1 =

01	02	03	04	05	06	07	08	09	10
----	----	----	----	----	----	----	----	----	----

**Solution:**

```
.data
    array BYTE 11,12,13,14,15,16,17,18,19,20
.code
main PROC
    cld
    mov esi,OFFSET array
    mov edi,esi
    mov ecx,LENGTHOF array
L1: lodsb                ; load [ESI] into AL
    mul AL                ; AL2
    stosb                ; store AL into [EDI]
    loop L1
```