COE 205, Term 991

Computer Organization & Assembly Programming

HW# 2

Due date: Wednesday, Oct. 6

Q.1. Show the content of the memory allocated based on the following directives, assuming that the first byte is allocated in address 0000H in the data segment.

```
I    DB    0,1,2
     DW    0,1,2
     DD    0,1,2
J    DB    `A`, `B`
     DB    `AB`
     DW    `AB`
     DW    offset J
     DB    2 dup (`AB`)
```

Q.2. Suppose that you have the following initial content of the registers and memory locations:

AX=0312H	BX=0012H	CX=0010H	DX=0210H
SI =0010H	DI =0012H	BP=0200H	SP =0300H
DS =2000H	ES =3000H	CS=1000H	SS =4000H
IP =1220H			

Memory Address		Contents (hex)
2000:	0010	12
	0011	34
	0012	56
	0013	78
	0014	9A
	0015	BC
	0016	DE
	0017	F0

(i) Show the contents of the registers and memory locations modified after the execution of each of the following instructions. Use the modified contents of the registers and memory locations as initial values for the subsequent instructions. Furthermore, specify the addressing modes of the source and destination operands in each instruction.

1. ADD BX, 0001H
2. MOV DL, [0011]
3. ADD CX, [SI]
4. MOV [DI], BL
5. MOV AX, 0001H[BX]
6. MOV [BX][DI], CL

(ii) Determine the starting and ending addresses of the code segment. What is the physical address of the next instruction to be fetched from memory.

(iii) Show the contents of AX, BX, and the flags (O,S,Z,A,P, and C) at the end of executing the following instructions

```
MOX AX, 7FDBH
MOV BX,  8799H
ADD AX, BX
```

(iv) Show the contents of AX, BX, and the flags (O,S,Z,A,P, and C) at the end of executing the following instructions

```
MOX AX, 7FDBH
MOV BX,  8799H
SUB  AX, BX
```

Q.3. Write an 8086 assembly program to prompt the user to enter a lower-case letter, read the letter entered and display the corresponding upper-case letter. The program should convert the letter to its upper-case equivalent and display it on a new line. The C version of the program is given below. Use the INT 21H routine, for character input, character output, and string output.

C version:

```
Main()
{
    int c;
    printf(``Enter a lower-case letter:``);
    c=getchar();
    c= c - 32; /* convert to lower case */
    printf(``\n The upper-case equivalent is: %c ``,c);
}
```