Assembly and Machine Language - Fall 96 **Midterm Exam**	**Instructor:** **B. Nasihatkon**	دانشگاه صنعتی خواجه نصیرالدین طوسی K. N. TOOSI UNIVERSITY OF TECHNOLOGY
Name:	**ID:**	**Azar 1396 - Nov. 2017**

Part I: Questions

1- After running the assembly command: `mov AL, 0D7h`

a) What will be the **binary** representation of AL? Why? (4 points)

b) Looking at AL as an **unsigned integer**, what number (in decimal) is stored in it? Why? (3 points)

c) Looking at AL as a **2's complement signed integer**, what number (in decimal) is stored in it? Why? (4 points)

2- After running the assembly command: `mov DX, -300`

d) What will be the **hexademial** representation of **DX**? Why? (6 points)

e) Looking at **DL** as an **unsigned integer**, what number (in decimal) is stored in it? Why? (3 points)

f) Looking at **DH** as a **2's complement signed integer**, what number (in decimal) is stored in it? Why? (4 points)

3- Write a single assembly command equivalent to the the following code. Briefly, explain your answer. (4 points).

```
not EAX
inc EAX
```

4- What will be stored in AL after running the following code? Why? (6 points)

```
mov al, bl
xor bl, -1
and al, bl
```

5- Write three lines of assembly code that sets bit number 0 of AX to 1, sets bit number 12 of AX to 0, and flips the value of bit 7 (6 points).

Part II: Programming

- You do not need to (and must not) write full programs. Just write the required assembly code for doing each task.
- Write programs in the designated code area as follows:

label	command	arguments
`loop1:`	`mov`	`eax, ebx`
	`call`	`prog2`
	`jne`	`loop1`
`prog2:`		

- You can use functions from the book:

`call print_int`	prints the value EAX
`call print_nl`	prints out a new line character.
`call read_int`	reads an integer and stores it into EAX.

6- Write a program that counts and prints the number of "110" bit sequences in EAX. For example if EAX= 00101010**110**111**110**01010**110**01010111 it has to print **3 (15 points)**.

label	command	arguments

7- We want to write a program that simplifies fractions (e.g. simplifies 8/12 to 2/3). The program reads two **positive** integers as numerator and denominator, and prints two numbers as the numerator and denominator of the simplified fraction (in simplest form). Assume all inputs are positive. (20 points) Example:

input: **8** **12** output: **2** **3**	input: **24** **6** output: **4** **1**	input: **12** **7** output: **12** **7**

label	command	arguments

8- We want to implement a modified version of Fibonacci series that starts with two arbitrary numbers a,b. That is

F[1] = a, F[2] = b, F[k] = F[k-1] + F[k-2].

Your task is to write a subprogram **fibo(a,b,n)** which takes a,b and n as arguments and returns F[n] (the n-th term of the above series).

a) Code the function. Use C convention. Store the return value in EAX. (20 points)

label	command	arguments
`fibo:`	`push`	`ebp`
	`mov`	`ebp, esp`

b) Write few lines of assembly code for calling the function fibo(2,4, 5). Use the default C calling convention. (5 points)

label	command	arguments

EXTRA SPACE:

