


Assembly and Machine Language - Spring 1397 (2018) Midterm Exam	Instructor: B. Nasihatkon	
Name:	ID:	Ordibehesht 1397 - May 2018

Question 1 Assume that the data section of your assembly code is as follows

```
1b11: dd 0
```

After running the following assembly commands

```
xor EAX, EAX
mov AX, -1
mov AL, 7Dh
mov [1b11], EAX
mov CL, [1b11+1]
```

- What will be the **binary** representation of AL? Why? (3 points)
- As an **unsigned integer**, what **decimal** number does AL represent? Why? (3 pts)
- As a **2's complement signed integer**, what decimal number does AL represent? Why? (3 points)
- What is the Hexadecimal representation of AX? Why? (3 pts)
- As an **unsigned integer**, what decimal number does **AX** represent? Why? (3 pts)
- As a **2's complement signed integer**, what decimal number does AX represent? Why? (3 points)
- As an **unsigned integer**, what decimal number does **CL** represent? Why? Assume a little-endian system. (3 points)

Functions from the book

<code>call print_int</code>	prints EAX as an unsigned integer
<code>call print_nl</code>	prints a new line character
<code>call read_int</code>	reads an integer and stores it in EAX.
<code>call print_string</code>	prints a null-terminated string whose starting address is in EAX

Programming

Write programs in the designated code area as follows:

label	command	arguments
<code>loop1:</code>	<code>call</code>	<code>prog2</code>
	<code>jne</code>	<code>loop1</code>
<code>prog2:</code>		

Question 2 What does the following assembly code compute? Explain the relationship between the user input and the program output. (18 points)

```
    call read_int
    mov  edi, eax

    mov  ecx, 0
    mov  ebx, 1
    mov  esi, 1
loop1:  cmp  ebx, edi
        jg  endloop1
        mov  eax, ebx
        imul eax
        imul esi
        add  ecx, eax
        neg  esi
        inc  ebx
        jmp  loop1
endloop1:
    mov  eax, ecx
    call print_int
    call print_nl
```

Question 3 Write an assembly program that checks the bits of the **AL** register. If bit 0 is equal to 1, bit 1 is equal to 0, and bit 6 is not equal to bit 4 it must print YES. Otherwise, it has to print NO. The **AL** register must remain unchanged (before printing the output). (16 points)

label	command	arguments	label	command	arguments
segment .data					
yes_msg:	db "YES", 10, 0				
no_msg:	db "NO", 10, 0				
segment .text					

Question 4 The function sum (below on the right) takes 3 arguments, call them a,b and k. It computes and returns the sum of numbers between a and b with a skip of k, that is $a + (a+k) + (a+2k) + \dots + (a+mk)$ such that $(a+mk) \leq b$.

The code on the left reads a,b and k, calls the sum function with these as arguments, and prints the return value. But there are 7 mistakes in the code (instructions to be modified, removed or added). Find and fix them. (20 points)

<pre> call read_int mov ebx, eax call read_int mov ecx, eax call read_int push eax push ecx push ebx call sum add esp, 12 call print_int call print_nl </pre>	<pre> sum: push ebp mov esp, ebp push eax mov eax, 0 mov ecx, [ebp+16] loop1: cmp ecx, [ebp+12] jg endloop1 add eax, ecx add ecx, [ebp+8] jmp loop1 endloop1: pop eax ret 12 </pre>
---	--

