CS221 Assembly Language

Lab2

Prof. Mohamed Ben Othman

Lab instructor: Mr. Musa Alzakan

1- **Practice**:

Data Transfer:

use the given program template to declare the variety of data transfer we studied in lecture 3.

```
TITLE Flat Memory Program Template   (Template.asm)

; Program Description:
; Author:                       Creation Date:
; Modified by:                  Modification Date:

.686
.MODEL FLAT, STDCALL
.STACK

INCLUDE Irvine32.inc
.DATA
    ; (insert variables here)
.CODE
main PROC
    ; (insert executable instructions here)
    exit
main ENDP
    ; (insert additional procedures here)
END main
```

Use your slides to:

A- declare a byte, word and dword variables

B- use mov instruction to move data from:

a. register to register (try all registers types: general purpose and segment registers)

b. from memory to register and vice-versa
c. from register to memory
d. mov an immediate value to register and to memory

Note that you should know the restrictions (same operands' sizes, no memory to memory, no immediate in destination).

C- Use movzx and movsx to practice the mov with zero and sign extension. Note that the destination should be bigger than the source.

D- Use xchg instruction to exchange values between two operands. Note that there are some exceptions.

Exercise:

instructions	data declaration	macros	Registers
mov add inc loop	byte, word	`WriteString` `WriteInt`	al, ax, eax, edi,
		`WriteDec`	

E- declare an array (a1) of bytes of 10 integer elements (a value should between 0 and 255).

F- declare an array (a2) of bytes of 10 elements that is no initialized.

G- declare a word variable (Total) initialized to 0.

H- use this loop:

I-

```
mov ecx, 9
mov edi, 0 ; edi is the index register
mov eax,0; this to initialize the 32 bits to 0 even if you will use smaller register size.
xx:
; print a1[edi] the element number edi of a1
; Note that you should get the value in eax to use WriteInt
; mov a1[edi] in a2[edi] ; don't forget the mov restrictions
; add the a1[edi] with total in total ; don't forget the add restrictions
inc edi
loop xx ; note that loop subtracts 1 from ecx and jump to xx if ecx is not 0.
print the (message) "the total of the array a1 is: " and then the value of total (in the same line)
```

Note:

1- It is better to design your output in an attractive one: put spaces between integers and print the message of the total in a new line. Also use `WriteDec` to not print the sign.

2- To print a string (buffer):

```
mov  edx,OFFSET buffer    ; display the buffer
call WriteString
```

3- To print an Integer:

```
the integer you would like to print should be in eax
call WriteInt
```

4- You should practice and do the exercise home and, in the Lab time: the first half the instructor will check your homework, take notes, and answer your questions if you have any. The second half will be a quiz. quiz.

2- Quiz:

You will have a Quiz.