Department of Computer Science
Faculty of Computing
UNIVERSITI TEKNOLOGI MALAYSIA

SUBJECT NAME:	**COMPUTER ORGANIZATION AND ARCHITECTURE**
SUBJECT CODE:	**SECR 1033**
SEMESTER:	**2 – 2023/2024**
LAB TITLE:	**Lab 2: Arithmetic Equations & Operations**
STUDENT INFO :	Execute the lab in group of two. **Student 1**: *No. 1, 3, 5* **Student 2**: *No 2, 4, 6* **Name 1:** _____ **Metric No:** _____ **Link for Video Demo:** _____ **Name 2:** _____ **Metric No:** _____ **Link for Video Demo:** _____
SUBMISSION DATE & ITEMS:	**Duration of submission :-** 2 weeks **Submission items in elearning:-** 1. Lab 2 exercise sheet/file (in .pdf), with the links for demo video 5 - 10min, on the cover page. 3. The assembly programs (in .asm).

MARKS: _____

Arithmetic Equation Coding in Assembly Language

Q1. Execute the program below. Determine output of the program by inspecting the content of the related registers.

a) Fill in Table 1 with the content of each register or variable on every LINE, in **Hexadecimal** (as per the output). Please complete the comments for every LINE.

b) Paste the screenshot of all registers' content after each LINE is executed.

```
INCLUDE Irvine32.inc
.data
var1 word 1
var2 word 9

.code
main PROC
        mov ax, var1     ; LINE1
        mov bx, var2     ; LINE2
        xchg ax, bx      ; LINE3
        mov var1, ax     ; LINE4
        mov var2, bx     ; LINE5
        call DumpRegs
        exit
main ENDP
END main
```

Answer Q1

a) Fill (Write) in the contents for the related register in each line:

Table 1

LINE1	**AX = 0001h** **var1 = 0001h**	Move the value of var1 (1d) into register AX
LINE2	**BX =** **var2 =**	
LINE3	**AX =** **BX =**	
LINE4	**AX =** **var1 =**	
LINE5	**BX =** **var2 =**	

b) Paste here screenshot of all registers' content after each LINE is executed:

LINE1:

LINE2:

LINE3:

LINE4:

LINE5:

Q2. Execute the program below. Determine output of the program by inspecting the content of the related registers and watches.

a) Fill in Table 2 with the content of each register or variable on every LINE, in **Hexadecimal** (as per the output). Please complete the comments for every LINE.

b) Paste the screenshot of all registers' content after each LINE is executed.

Arithmetic expression: Rval = (-Xval + (Yval – Zval)) + 1

```
include irvine32.inc

.data
Rval DWORD ?
Xval DWORD 26
Yval DWORD 30
Zval DWORD 40

.code
main proc
      mov eax,Xval       ; LINE1
      neg eax            ; LINE2
      mov ebx,Yval       ; LINE3
      sub ebx,Zval       ; LINE4
      add eax,ebx        ; LINE5
      inc eax            ; LINE6
      mov Rval,eax       ; LINE7
      exit
main endp
end main
```

Answer Q2

a) Fill (Write) in the contents for the related register in each line:

Table 2

LINE1	**EAX = 0000001Ah** **Xval = 0000001Ah**	Move the value of Xval (26d) into register EAX
LINE2	**EAX =**	
LINE3	**EBX =** **Yval =**	
LINE4	**EBX =** **Zval =**	
LINE5	**EAX =** **EBX =**	
LINE6	**EAX =**	
LINE7	**EAX =** **Rval =**	

b) Paste here screenshot of all registers' content after each LINE is executed:

LINE1:
LINE2:
LINE3:
LINE4:
LINE5:

LINE6:
LINE7:

Q3. Execute the program below. Determine output of the program by inspecting the content of the related registers.

a) Fill in Table 3 with the content of each register or variable on every LINE, in **Hexadecimal** (as per the output). Please complete the comments for every LINE.

b) Paste the screenshot of all registers' content after each LINE is executed.

Arithmetic expression: var4 = [(var1 * var2) + var3] - 1

```
include irvine32.inc

.data
var1 DWORD 5
var2 DWORD 10
var3 DWORD 20
var4 DWORD ?

.code
main proc
      mov eax, var1          ; LINE1
      mul var2               ; LINE2
      add eax, var3          ; LINE3
      dec eax                ; LINE4
      exit
main endp
end main
```

Answer Q3

a) Fill (Write) in the contents for the related register in each line:

Table 3

LINE1	**EAX = 00000005h** **var1 = 00000005h**	Move the value of var1 (5d) into register EAX
LINE2	**EAX =** **var2 =**	
LINE3	**EAX =** **var3 =**	
LINE4	**EAX =** **var4 =**	

b) Paste here screenshot of all registers' content after each LINE is executed:

LINE1:
LINE2:
LINE3:
LINE4:

Q4. Execute the program below. Determine output of the program by inspecting the content of the related registers.

a) Fill in Table 4 with the content of each register or variable on every LINE, in Hexadecimal (as per the output). Please complete the comments for every LINE.

b) Paste the screenshot of all registers' content after each LINE is executed.

Arithmetic expression: var4 = (var1 * 5) / (var2 – 3)

```
include irvine32.inc
.data
      var1 WORD 40
      var2 WORD 10
      var4 WORD ?
.code
main proc
      mov ax,var1       ; LINE1
      mov bx,5          ; LINE2
      mul bx            ; LINE3
      mov bx,var2       ; LINE4
      sub bx,3          ; LINE5
      div bx            ; LINE6
      mov var4,ax       ; LINE7
      exit
main endp
end main
```

Answer Q4

a) Fill (Write) in the contents for the related register in each line:

Table 4

LINE1	**AX = 0028h** **var1 = 0028h**	Move the value of var1 (40d) into register AX
LINE2	**BX =**	
LINE3	**AX =** **BX =**	
LINE4	**BX =** **var2 =**	
LINE5	**BX =**	
LINE6	**AX =** **BX =** **DX =**	
LINE7	**AX =** **var4 =**	

b) Paste here screenshot of all registers' content after each LINE is executed:

LINE1:
LINE2:
LINE3:

LINE4:
LINE5:
LINE6:
LINE7:

Short Notes for MUL CX and DIV BL:

MUL CX

a. MUL always uses AX (or its extended versions EAX or RAX) as the implicit destination register.

b. The operand size determines the size of the result:

   i. Byte-sized operand: Result in AX

   ii. Word-sized operand: Result in DX:AX

   iii. Doubleword-sized operand (32-bit mode): Result in EDX:EAX

   iv. Quadword-sized operand (64-bit mode): Result in RDX:RAX

c. The upper half of the result (DX or EDX or RDX) holds any overflow bits.

d. The Carry Flag (CF) is set if the upper half of the product is non-zero.

DIV BL

a. DIV always uses the DX:AX or EDX:EAX pair as the implicit dividend register.

b. The divisor is specified as the operand of the DIV instruction.

c. The quotient is stored in AX (for 16-bit division) or EAX (for 32-bit division).

d. The remainder is stored in DX.

e. Clear DX (or EDX for 32-bit division) before division to ensure a correct 16-bit or 32-bit dividend.

f. If the divisor is 0, a division error occurs.

g. The Overflow Flag (OF) is set if the quotient is too large to fit in the destination register.

Q5. Given the following instructions as is Code Snippet 1.

a) Write a full program to execute the Code Snippet 1.
b) What are the contents of the related registers after Code Snippet 1 is executed? Paste the screenshot of DumpReg.

; Code Snippet 1 (MUL CX)

```
MOV DX,0        ; Clear DX
MOV AX,1000h    ; Load 1000h into AX
MOV CX, 25h     ; Load 25h into CX
MUL CX          ; Multiply AX by CX, storing the result in DX:AX
```

Answer Q5

a) Screenshot of full program (.asm) :

b) Paste here the screenshot of the final registers' content (DumpReg):

Q6. Given the following instructions as is Code Snippet 2.

a) Write a full program to execute the Code Snippet 2.
b) What are the contents of the related registers after Code Snippet 2 is executed? Paste the screenshot of DumpReg.

```
; Code Snippet 2 (DIV BL)
MOV DX, 0       ; Clear DX to form the 16-bit dividend in DX:AX
MOV AX, 803h    ; Load the dividend (8003h) into AX
MOV BL, 10h     ; Load the divisor (10h) into BL
DIV BL          ; Divide DX:AX by BL, whereby AX=quotient & DX=remainder
```

Answer Q6

a) Screenshot of full program (.asm) :

b) Paste here the screenshot of the final registers' content (DumpReg):