Department of Computer Science
Faculty of Computing
UNIVERSITI TEKNOLOGI MALAYSIA

SUBJECT NAME: **COMPUTER ORGANIZATION AND ARCHITECTURE**

SUBJECT CODE: **SECR 2033**

SEMESTER: **2 - 2019/20**

LAB TITLE: **Programming 2: Arithmetic Equations & Operations**

STUDENT INFO :

Name: **SEE WEN XIANG**

Matric No: **A19EC0206**

Secction: **07**

SUBMITTED DATE: **14/4/2020 (Tuesday)**

COMMENTS:

MARKS:

_____

Part B: Arithmetic Equation Coding in Assembly Language (Equation to Code)

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

```
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:

Table 1

LINE 1	**AX = 001h** **var1 = 001h**	Move the value of var1(1d) into register AX
LINE 2	**BX = 009h** **var2 = 009h**	Move the value of var2(9d) into register BX
LINE 3	**AX = 009h** **BX= 001h**	Value in register AX exchange with register BX
LINE 4	**AX = 009h** **var1 = 009h**	Move the value of AX(9d) into var1
LINE 5	**BX = 001h** **var2 = 001h**	Move the value of BX(1d) into var2

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

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:

Table 2

LINE 1	**EAX = 0000001Ah** **Xva1 = 0000001Ah**	Move the value of Xvar1(26d) into register EAX
LINE 2	**EAX = FFFFFFE6h**	Reverse the sign of the number in register EAX converting the number to its two's complement
LINE 3	**EBX = 0000001Eh** **Yval= 0000001Eh**	Move the value of Yva1(30d) into register EBX
LINE 4	**EBX = FFFFFFF6h** **Zva1 = 00000028h**	Subtract the value of EBX with Zval then store in EBX
LINE 5	**EAX = FFFFFFDCh** **EBX = FFFFFFF6h**	Add the value of EAX with EBX then store in EAX
LINE 6	**EAX = FFFFFFDDh**	Add one from the contents of EAX
LINE 7	**EAX= FFFFFFDDh** **Rval = FFFFFFDDh**	Move the value of EAX into variable Rval

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

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
     mov var4, eax          ; LINE5
     exit
main endp
end main
```

Answer:

Table 3

LINE 1	**EAX = 00000005h** **var1 = 00000005h**	Move the value of var1(5d) into register EAX
LINE 2	**EAX = 00000032h** **Var2 = 0000000Ah**	Multiply var2 with the value in EAX then store into EAX
LINE 3	**EAX = 00000046h** **var3= 00000014h**	Adding var3 with the value in EAX then store into EAX
LINE 4	**EAX = 00000045h**	Subtract one from the contents of EAX
LINE 5	**var4 = 00000045h**	Move the value of EAX(69d) into var4

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

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:

Table 4

LINE 1	**AX = 0028h** **var1 = 0028h**	Move the value of var1(40d) into register AX
LINE 2	**BX = 0005h**	Move the value 5 (5d) into register BX
LINE 3	**AX = 00C8h** **BX = 0005h**	Multiply the with AX with BX then store into AX
LINE 4	**BX = 000Ah** **var2 = 000Ah**	Move the value of var2 into register BX
LINE 5	**BX = 0007h**	Subtract 3 with the value of BX then store into BX
LINE 6	**AX = 001Ch** **BX = 0007h** **DX = 0004h**	Divide AX (dividend) with the value of BX, the quotient is stored in the AX register; the remainder is stored in the DX register
LINE 7	**AX= 001Ch** **var4 = 001Ch**	Move value of AX into var4

Table 5 – Initial Registers Values

EAX	EBX	ECX	EDX
10H	20H	2H	0H

5. Refer to Table 5 above for the following TWO (2) tasks.

i. Write and execute the instructions below, then fill in the table with the correct value of the registers. Also note the changes.
******Note****: Always start with the initial value of the registers given in Table 5.*

a. MUL BX

EAX	EBX	ECX	EDX
00000200H	00000020H	00000002H	00000000H

b. MUL CX

EAX	EBX	ECX	EDX
00000020H	00000020H	00000002H	00000000H

c. MUL AX

EAX	EBX	ECX	EDX
00000100H	00000020H	00000002H	00000000H

d. DIV BX

EAX	EBX	ECX	EDX
00000000H	00000020H	00000002H	00000010H

e. DIV CX

EAX	EBX	ECX	EDX
00000008H	00000020H	00000002H	00000000H

f. DIV AX

EAX	EBX	ECX	EDX
00000001H	00000020H	00000002H	00000000H

ii. Write and execute the instructions below, then fill in the table with the correct value of the registers. Also note the changes.
******Note****: Always start with the initial value of the registers given in Table 5.*

a. MUL EBX

EAX	EBX	ECX	EDX
00000200H	00000020H	00000002H	00000000H

b. MUL ECX

EAX	EBX	ECX	EDX
00000020H	00000020H	00000002H	00000000H

c. MUL EAX

EAX	EBX	ECX	EDX
00000100H	00000020H	00000002H	00000000H

d. DIV EBX

EAX	EBX	ECX	EDX
00000000H	00000020H	00000002H	00000010H

e. DIV ECX

EAX	EBX	ECX	EDX
00000008H	00000020H	00000002H	00000000H

f. DIV EAX

EAX	EBX	ECX	EDX
00000001H	00000020H	00000002H	00000000H