PRACTICAL 1: Machine Execution

Part I

1. How to open the debug program?

2. Which DEBUG command performs the following operations?

   a) Display the contents of registers after the execution.

   b) Begin assembling statements that will be converted to machine language.

   c) Display machine code for assembly instructions entered.

   d) Enter machine instructions into memory.

   e) To display the content of all the registers.

   f) To quit the DEBUG session.

3. Use DEBUG to enter the following commands:

```
a 100
mov cl, 42
mov dl,2a
add cl,dl
jmp 100
```

   What you can see when the following command is typed?

   a) u 100,107

   b) d cs:100

   c) e cs:100 a1 00 02 03 06 02 02

   d) u 100,106

4. What you can see when the following command is typed?

a) abc0 0fff

b) 0000 1111

5. Use DEBUG to enter the following commands:

```
a 100
mov ax,0123
add ax,0025
mov bx,ax
add bx,ax
mov cx,bx
sub cx,ax
sub ax,ax
jmp 100
```

What you can see when the following command is typed?

a) r

b) t (repeat 7 times)

What is the command to run 7 lines starting from the address 100?

6. Use DEBUG to enter the following command:

E CS:100 B8 45 01 05 25 00 The hexadecimal value 45 was supposed to be 54. Code another E command to correct only the one byte that is incorrect, that is change 45 to 54 directly.

7. Assume that you have used DEBUG to enter the following E command:

E CS:100 B8 05 1B 00 2C EB F8

What are the three symbolic instruction represented here?

Part II

1. Consider the machine language instructions

B0 1C D0 E0 B3 12 F6 E3 EB F6

Which instruction performs the following operations?

a) Move hex value 1C to the AL register.

b) Shift the contents of AL one bit to the left.

c) Move the hex value 12 to BL.

d) Multiply AL by BL.

e) Jump back to 100

Use DEBUG's E command to enter the program beginning at CS:100, then trace through the program until reaching JMP.

What is the final product in AX?

Confirm the result by manual calculation.

2. What is the output in AX?

```
A 100
MOV AL, 5            ; AL = multiplicand
MOV BL, 10           ; BL = multiplier (operand)
MUL BL
JMP 100
```

3. What is the output in AX and DX?

```
A 100
MOV AX, 0083         ; dividend
MOV BL, 2            ; divisor (8 bits)
DIV BL
JMP 100
```

4. What is the output in AX and DX?

```
A 100
MOV DX, 0          ; clear
MOV AX, 8003       ; dividend
MOV CX, 100        ; divisor (16 bits)
DIV CX             ;
JMP 10
```

5. Assume that AL contains 10101010 and that an item named BL contains 11110000. Determine the effect on AL for the following unrelated operations by using debug program:

   a) AND AL, BL

   b) OR AL, BL

   c) XOR AL, BL

   d) NOT AL

6. What is the output in AX and BX?

```
A 100
MOV AL, 8
SHR AL, 1          ; shift right →
MOV BL, 8
SHL BL, 1          ; shift left ←
 JMP 100
```

AX	BX

Part III

1. Enter the following instructions into DEBUG program. The CS should be 116E.

```
-A 100
116E:0100 MOV AX, 0010
116E:0103 MOV BX, 0020
116E:0106 MOV CX, 0030
116E:0109 ADD AX, BX
116E:010B INC BX
116E:010C SUB CX, AX
116E:010E DEC CX
116E:010F JMP 0100
116E:0111 <enter>
```

What is the content of register AX, BX, CX and IP?

AX	BX	CX	DX

2. Trace the content of the registers used in the following program segment:

```
    MOV AX, 1
    MOV BX, 1
    MOV CX, 5
    MOV DX, 0
A10:
    ADD AX, BX
    MOV DX, AX
    MOV AX, BX
    MOV BX, DX
    LOOP A10
```

	AX	BX	CX	DX
After 1st loop				
After 2nd loop				
After 3rd loop				
After 4th loop				
After 5th loop				

3. Trace the execution of the following instructions and record the values of the register.

```
MOV AX,010
MOV BX,020
MOV CX,030
ADD AX,BX
INC BX
SUB CX,AX
DEC CX
JMP 100
```

AX	BX	CX

4. What is the output?

```
a 100
xxxx:0100 jmp 126
xxxx:0102 db 0d,0a, "This is my first DEBUG program!"
xxxx:0123 db 0d,0a,"$"
xxxx:0126 mov ah,9
xxxx:0128 mov dx,102
xxxx:012B int 21
xxxx:012D mov ah,0
```

xxxx:012F int 21
xxxx:0131

5. What is the final value of AX and BX?

```
        MOV AX,00
        MOV BX,00
        MOV CX,3            ;Initialize for 3 loops
A20:
        INC AX
        ADD BX,AX
        LOOP A20            ;Decrement CX ;Repeat if nonzero
```

AX	BX	CX

6. What is the final value of AX and BX?

```
        MOV AX,0            ;Initialize AX and
        MOV BX,0            ;BX to zero,
        MOV CX,4            ;CX for 4 loops
A20:
        INC AX               ;Add 01 to AX
        ADD BX,AX           ;Add AX to BX
        LOOP A20            ;Decrement CX, loop if nonzero
```

AX	BX

PRACTICAL 2: Assembly Language Fundamental I

★1. Arithmetic Expression

Write a program that implements the following arithmetic expression:

result = val2 + 5 – val1 + val3

Using the following data definitions:

```
val1 DB 6
val2 DB 3
val3 DB 4
result DB ?
```

In comments next to each instruction, write the hexadecimal value of AL. Print the final result on screen.

★2. Uppercase to lowercase conversion

Defines a symbolic constant for uppercase letter 'A' and create a variable that uses the symbol as initialize. Write a program that converts this uppercase letter to lowercase. Print the output as the format below:

Modify the constant value and check the result.

A , a

★3. Lowercase to uppercase conversion

Modify the program in question 2 to allow a conversion from a lowercase letter to an uppercase letter.

★4. Exchanging two character

Write a program that defines two initialized character and exchanges their contents. Print the output as the format below:

(a , k) ► (k , a)

★5. Multiplication (product in single digit)

Write a program that prompts the user for a decimal digit and display the digit and its self multiplication result with an appropriate message.

```
Please enter a digit: 2
2 times 2 returns: 4
```

★★6. Multiplication (products in 2 digits)

Modify the program from Question 5, prompts the user for a decimal digit (4-9) and display the digit and its self multiplication result (2 digits) with an appropriate message.

```
Please enter a digit: 6
6 times 6 returns: 36
```

★6. Division

Write a program that calculates and displays the quotient and remainder of a division operation. For this exercise, use single digit dividend and divisor.

```
Dividend : 8
Divisor : 5
Quotient : 1
Remainder : 3
```

PRACTICAL 3: Assembly Language Fundamental II

★1. Copy the contents

Given the following declarations, write a program that replaces the content of *data2* with *data1* and displays the content of *data1* and *data2*.

```
data1 byte   "MILK"
data2 byte   4   dup   ('*')
```

Initial content
data1: MILK
data2: **

After replacement
data1: MILK
data2: MILK

★2. Reverse a string

Modify the program in question 1 to store the reverse of *data1* in *data2*. Then display *data2*.

Initial content
data1: MILK
data2: **

After replacement
data1: MILK
data2: MILK

After reversed
data1: MILK
data2: KLIM

★3. Uppercase to lowercase

Modify the program in question 1 to store the lowercase of *data1* into *data2*. Then display *data2*.

```
Initial content
data1: MILK
data2: ****

After replacement
data1: MILK
data2: MILK

After reversed
data1: MILK
data2: KLIM

After changed case
data1: MILK
data2: milk
```

★★4. Sum an array of integers

Declare a list of integers and write a program that adds each of the values in the list. Then display the sum. Use the following code to declare the list.

```
byteList    byte    2, 4, 6, 8, 10, 12
```

```
Sum: 42
```

★★5. Fibonacci Numbers

Write a program that uses a loop to calculate the first six values of the Fibonacci number sequence, described by the following formula:

Fib(1) = 1, Fib(2) = 1, Fib(n) = Fib(n -1) + Fib(n - 2)

Place each value in the array and display it.

```
1, 1, 2, 3, 5, 8,
```

PRACTICAL 4: Conditional Processing

★1. Counting negative values in array

Write a program to determine the number of positive and negative numbers in a list. The number zero is used to end the list, which means that the size of the integer list is not fixed.

E.g.
If the following list is used in the program,

```
list    byte   12, 29, -9, 5, -48, 20, 0
```

then the output should look like this:

There are 4 positive and 2 negative values in the list.

★2. Yes or No

Write a complete assembly language program to produce the result as below:

a. When the program is executed, an "A" will automatically be printed.
b. Next, a message is prompted for decision to continue program execution.
c. If the user keys in a „y", the program continue printing the next character.
d. If the user keys in a „"n", the program terminated and control return to DOS.
e. Otherwise, display an error message and allow the user to enter again.

A
Do you want to continue printing (y/n)? y
B
Do you want to continue printing (y/n)? s
Please enter y or n only
Do you want to continue printing (y/n)? y
C
Do you want to continue printing (y/n)? n

★★★3. Eggward's Eggnglish

Eggward from Eggland speaks Eggnglish.

Write a program to convert a normal English sentence held in memory into an Eggnglish by detecting every word starting with the letter "e" and replace the first letter "e" by the sequence "egg".

The program replaces the first letter "e", of any word begins with an "e", by "egg".

Sample output :

In english : You like english and espresso, excellent !
In eggnglish : You like eggnglish and eggspresso, eggxcellent!

PRACTICAL 5: Keyboard & Screen Processing

★1. String input

Write an assembly language program that requires the user to key in a string of characters. Display only the second character on the screen.

```
Enter a word: qwerty
The second character is w
```

★★2. Encryption

Encryption is the process of transforming information using an algorithm and a key to make it unreadable.

Write a complete program named *encrypt.asm* that allows users to enter a secret word and an encryption key (a digit from 1 to 9). A word can be encrypted by adding the encryption key to the ASCII value of each character in the word. Display the encrypted word.

```
Enter secret word: three333
Enter encryption key (1-9): 4
The encrypted word is xlvii777
```

★★3. Decryption

Decryption is the reverse process of encryption. Modify the program in question 2 to decrypt a secret word. The decryption key is the same as the encryption key. A sample output is given below.

```
Enter encrypted word: xlvii777
Enter decryption key (1-9): 4
The secret word is three333
```

★★4.Colors

Write a program that requires the user to key in the color for text display on the screen. A sample output is given below.

```
Green : 'G'
Red : 'R'
Blue : 'B'
Please enter G, B, or R for font's color : G
You choose green color !
```

★★★5. Boolean Calculator

Create a program that functions as a simple Boolean calculator. It should display a menu that asks the user to make a selection from the following list:

1. x AND y
2. x OR y
3. NOT x
4. x XOR y
5. Exit program

When the user makes a choice, prompt the user for the input and perform the corresponding operation. Display the result.