Week 4

1. Answer the following questions:
    A. Explain how the following shift and rotate instructions work:
        a. SHL, SAL, SHR, SAR
        b. ROR, ROL, RCR, RCL
    B. Explain how the following jump instructions work:
        a. CALL, RET
        b. JMP
        c. LOOP
        d. JC, JE, JZ, JS
        e. JA, JB
        f. JG, JL
    C. Explain how the following I/O instructions work: IN, OUT
    D. Explain how the following string instructions work:
        a. MOVSB, MOVSW
        b. LODSB, LODSW, STOSB ki STOSW
        c. REP
    E. Explain the following addressing mechanisms work (including advantages and disadvantages) and give examples:
        a. Immediate addressing
        b. Register addressing
        c. Direct addressing
        d. Indexed addressing
        e. Based addressing
        f. Mixed (indexed-based) addressing

Shift instructions

2. Solved exercise s3ex1.asm
   Implement in assembly language the following expression, using shift instructions:
   AX = 7*AXï 2*BXï BX/8
    a. Compile s3ex1.asm
    b. Execute in OllyDbg
    c. Execute each instruction and follow the changes in the registers and flags.

3. Implement in assembly language the previous expression (exercise 2), using arithmetic instructions.

Memory addressing

4. Solved exercise s3ex2.asm

Write a program that copies an array of values found in the memory, to another memory location, in reverse order, with the use of the stack.

a. Compile s3ex2.asm
b. Execute in OllyDbg
c. Execute each instruction and follow the changes in the registers and flags.

5. Write a program that calculates the average value of an integer (on BYTE) array found in the memory. The average (integer value) will be memorized in a BYTE variable.

6. Solved exercise s3ex3.asm

Write a program that computes the sum of two bi-dimensional matrices.

a. Compile s3ex3.asm
b. Execute in OllyDbg
c. Execute each instruction and follow the changes in the registers and flags.

Jumps

7. Implementing IF-THEN-ELSE in assembly language:

Pseudocode:

```
if (AX > BX) then AX = AX+1
             else  BX = BX+1
```

Assembly language (for signed numbers):

```
CMP AX, BX          ;compare registers
JG et_then
INC BX              ;if BX >=AX
JMP et_iesire       ;if  òelseò is executed, jump over òthenò
et_then:
INC AX              ;if AX>BX
et_iesire:
```

8. Implementing FOR in assembly language:

Pseudocode:

```
for (i = 0 ; i < n ; i ++)
{
```

```
        AX = AX+1
}
```

Assembly language:

Version 1

```
MOV DI, n
MOV SI, 0      ; correspondent of òiò
et_for:
INC AX
INC SI         ;increment SI to get to the next value
CMP SI, DI
JBE et_for
```

Version 2

```
MOV CX, n          ;CX is implicit counter for LOOP
et_loop:
INC AX
LOOP et_loop       ;CX = CX-1, (CX == 0)?
```

9. Solved exercise s3ex4.asm

Determine the minimum and maximum values in a positive integer array represented on BYTE. Write the minimum and maximum in the memory.

a. Compile s3ex4.asm
b. Execute in OllyDbg
c. Execute each instruction and follow the changes in the registers and flags.

10. Determine the minimum and maximum values in a signed (positive and negative) integer array represented on WORD. Write the minimum and maximum in the memory.