LAB #09

Implementation of Assembly Language programs using MACROS

Macros are just like procedures, but not really. Macros look like procedures, but they exist only until your code is compiled. After compilation all macros are replaced with real instructions. If you declared a macro and never used it in your code, compiler will simply ignore it.

Macros Syntax:

name MACRO [parameters...]

<instructio>

ENDM

When you want to use a procedure you should use CALL instruction, for example:

CALL MyProc

When you want to use a macro, you can just type its name. For example:

MyMacro [parameters...]

You should use **stack** or any general purpose registers to pass parameters to procedure. To pass parameters to macro, you can just type them after the macro name. For example:

MyMacro 1, 2, 3

To mark the end of the macro **ENDM** directive is enough.

Example:

org 100h

```
MyMacro MACRO p1, p2, p3
MOV AX, p1
MOV BX, p2
MOV CX, p3
ENDM
MyMacro 1, 2, 3
MyMacro 4, 5, DX
Ret
```

Lab Tasks

Execute the following tasks

Task.1:

Define the Macro to calculate the Cube of Number Present in Register.

SOURCE CODE:

```
cube MACRO
MOV AL,2
MUL CL
MUL CL
ENDM
cube
```

OUTPUT:

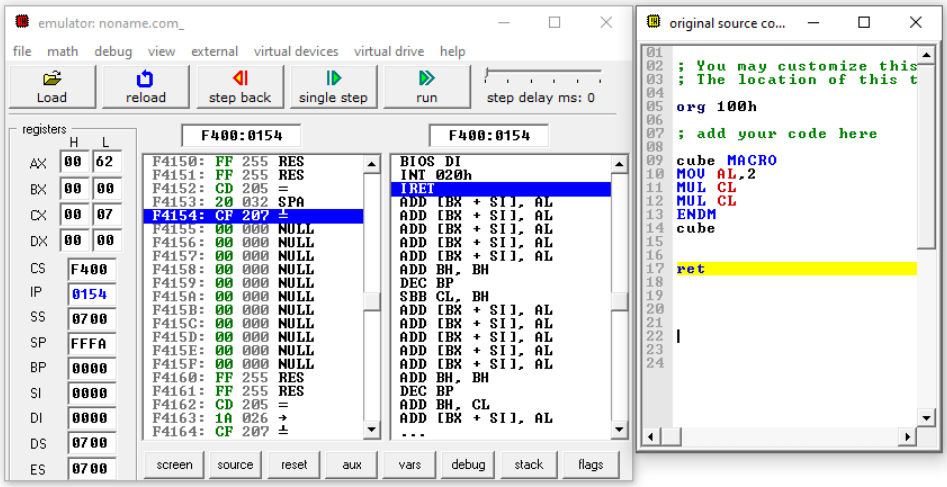
```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

; add your code here

   cube MACRO
   MOV AL,2
   MUL CL
   MUL CL
   ENDM
   cube

ret
```



Task.2:

Define the macro that will compare the number

SOURCE CODE:

```
Cmpr MACRO P1,P2
MOV AL,P1
MOV BL,P2
CMP AL,BL
ENDM
Cmpr  1,2
Cmpr  4,5
Cmpr  3,3
```

OUTPUT:

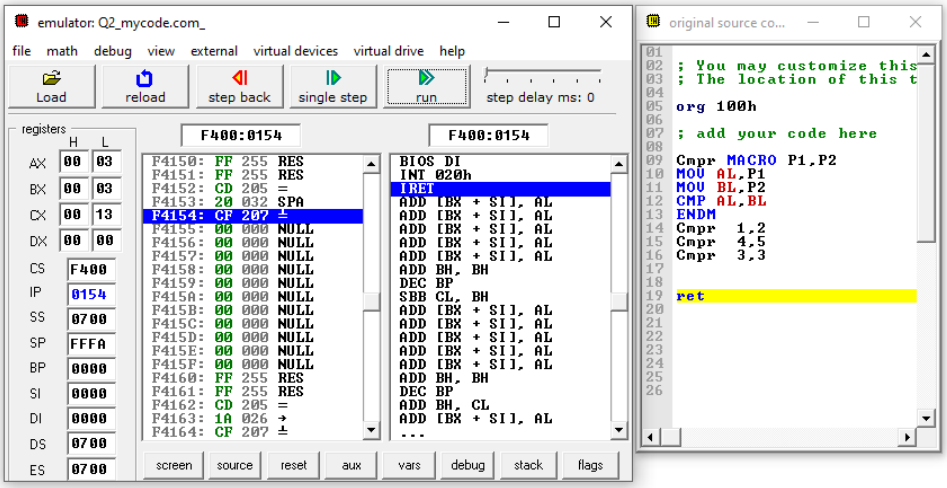
```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

; add your code here

    Cmpr MACRO P1,P2
    MOV AL,P1
    MOV BL,P2
    CMP AL,BL
    ENDM
    Cmpr  1,2
    Cmpr  4,5
    Cmpr  3,3

ret
```



Task.3:

Define the Macro that will calculate factorial of a given number?

SOURCE CODE:

```
Fact MACRO P1
mov cx,P1
mov ax,1h
 L1:
 mul cx
 loop L1
 ENDM
 Fact 5
```

OUTPUT:

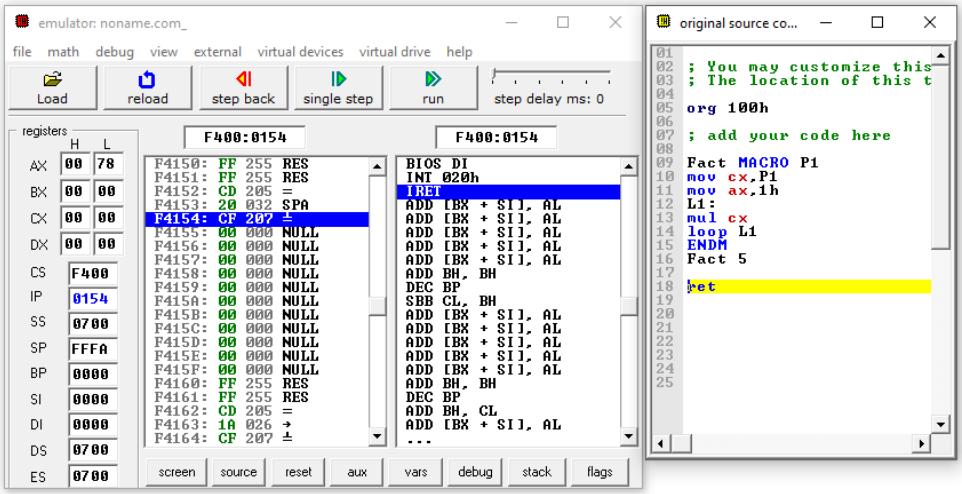
```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

; add your code here

    Fact MACRO P1
    mov cx,P1
    mov ax,1h
    L1:
    mul cx
    loop L1
    ENDM
    Fact 5

ret
```



Task.4:

Calculate the sum of numbers using the macroSUM. Once the sum is calculated, calculate its factorial using the procedure factorial

SOURCE CODE:

```
sum MACRO m1,m2

mov ax,m1

mov bx,m2

add ax,bx

endm

sum 4,7

factorial PROC

mov cx,ax

mov ax,1h

T2:
```

mul cx

loop T2

ret

factorial ENDP

ret

OUTPUT:

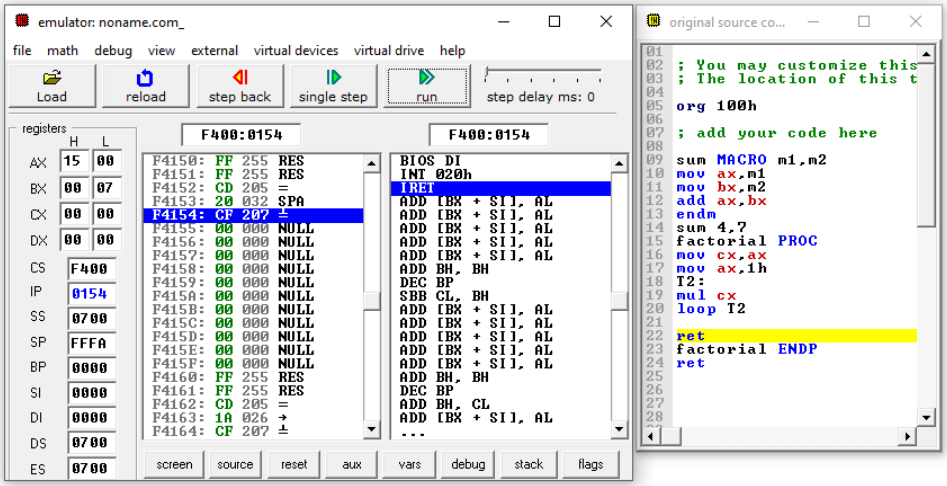
```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com_template.txt

org 100h

; add your code here

    sum MACRO m1,m2
    mov ax,m1
    mov bx,m2
    add ax,bx
    endm
    sum 4,7
    factorial PROC
    mov cx,ax
    mov ax,1h
    T2:
    mul cx
    loop T2

    ret
    factorial ENDP
    ret
```



Task.5:

Calculate the sum of two consecutive numbers from 1-10. The numbers must be passed to Macro created for the operation of sum. Store the result in consecutive memory location.

SOURCE CODE:

macroSUM MACRO P1,P2

MOV SI,100H

MOV AX,P1

MOV BX,P2

MOV CX,9

L1:

ADD AX,BX

MOV [SI],AX

INC BX

INC SI

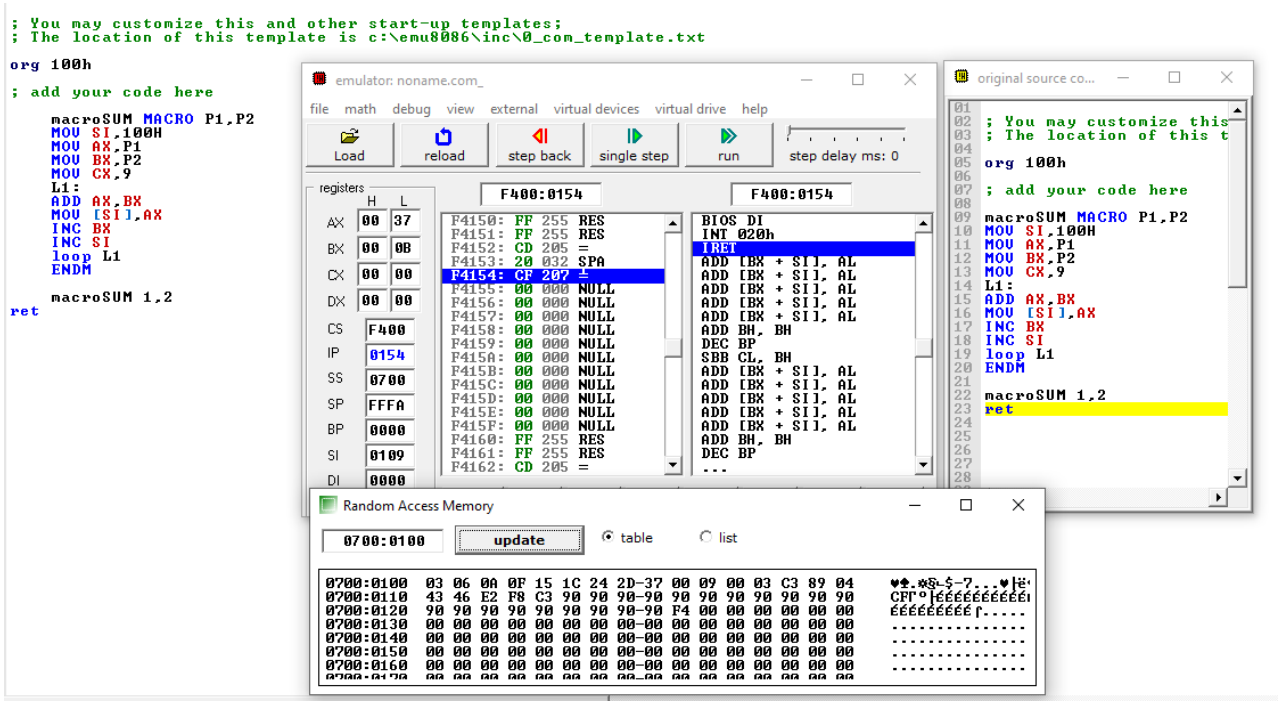
loop L1

ENDM

macroSUM 1,2

ret

OUTPUT:



----------THE END----------