**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...]

<instructions> 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 CLO [1]**

1. Define the Macro to calculate the Cube of Number Present in Register
2. Define the macro that will compare the number
3. Define the Macro that will calculate factorial of a given number?
4. Calculate the sum of numbers using the macroSUM. Once the sum is calculated, calculate its factorial using the procedure factorial
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.