Course Title: Microprocessors and Assembly Language Lab (CSE-4504)

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT), Gazipur

Lab # 07

Understanding Macro using Assembly Language Program.

Objective:

To understand 8086 instructions related to Macro using Assembly Language Program.

Theory:

- **Macro**
  A macro is a symbolic name given to one or more assembly language statements. A macro may be used to generate instructions or data definitions.

  Syntax for Declaring MACRO

```
macro_name   MACRO d1, d2, . . . , dn
             statement1
             statement2
             .   .   .
             statementk
ENDM
```

  Where d1, d2, . . . ,dn is an optional list of dummy parameters. Dummy parameters are temporary variables; they are not declared by data definition directives (DB, DW). They can be used as input as well as output parameters.

  A macro definition can appear anywhere in an assembly language program before the END directive. It is usual to place all macro definitions at the beginning of a program before the segment definitions.

  The difference between *macro* and *procedure* is that *procedure* requires separate memory location to store, where as *macro* uses the same memory location of the code where the macro actually is being called. Furthermore, in *macro* parameter values can be passed.

Assembly Language Program Example for Macro and Procedure:

```
MDSPLY_STRING   MACRO  STRING    ; Declaration of MACRO
                MOV   DX, OFFSET  STRING
                CALL  DSPLY_STRING  ; Calling the Procedure
ENDM

ORG 0100H

.DATA

MESSAGE1   DB   'Microprocessors and', 0DH, 0AH, '$'
MESSAGE2   DB   'Assembly Language$'

.CODE

MAIN  PROC

MOV AX, @DATA
MOV DS, AX

MDSPLY_STRING  MESSAGE1 ; 1st Call of the MACRO
MDSPLY_STRING  MESSAGE2 ; 2nd Call of the MACRO

MOV AH, 4CH          ; Return to DOS
INT 21H

MAIN  ENDP

DSPLY_STRING PROC      ; Declaration of PROCEDURE
             MOV  AH, 09H
             INT  21H
             RET
DSPLY_STRING  ENDP

END  MAIN
```

Tasks to do:

1. Write an Assembly Language code that declares an ARRAY using decimal digits (i.e., **array DB 2, 0, 4, 7, 1, 9**) and pass the array value and address to a MACRO (i.e, **macarray MACRO digit**). Using the array elements, address and one ***procedure*** search a digit inputted by the user.

   Sample Input / Output:

   Input: 2
   Output: Digit 2 Found

   Input: 3
   Output: Digit 3 Not Found

2. Write an Assembly Language code that takes an input ARRAY of 6-decimal digits (i.e., **array DB 6 Dup (?)**) and passes the array values and address to a MACRO (i.e, **macarray MACRO oddeven**). Using the array, address and one ***procedure*** separate out the ODD digits and EVEN digits.

   Sample Input / Output:

   Input: 2, 0, 4, 7, 1, 9
   Output: ODD Digits: 7 1 9
   EVEN Digits: 2 0 4