Experiment 02

Learning Objective: Student should be able to develop a calculator (Addition and Subtraction) for a 16 bits number using macros and procedure. (Menu Based).

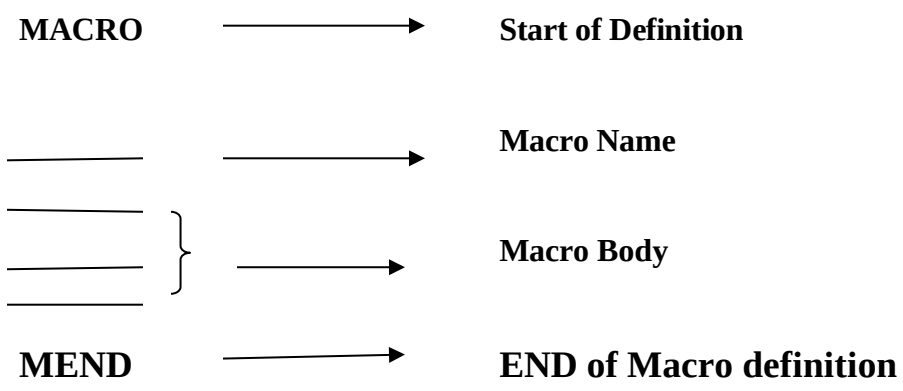
Tools: TASM/MASM

Theory:

Definition of Macro:

The assembly language programmer often finds certain statements being repeated in the program. The programmer can take the advantage of 'MACRO' facility where MACRO is defined to be –**Single line abbreviation for group of instructions.**

The template to be followed for defining a MACRO is as follows:



Definition & function of Macro processor:

- Macro processor is a program which is responsible for processing the macro.
- There are four basic tasks/ functions that any macro instruction processor must perform.

1. **Recognize macro definition:**

   A macro instruction processor must recognize macro definitions identified by the MACRO and MEND pseudo-ops.

1. **Save the definitions**:

   The processor must store the macro instruction definitions, which it will need for expanding macro calls.

2. **Recognize calls:**

   The processor must recognize macro calls that appear as operation mnemonics. This suggests that macro names be handled as a type of op-code.

3. **Expand calls and substitute arguments***:*

   The processor must substitute for dummy or macro definition arguments the corresponding arguments from a macro call; the resulting symbolic (in this case, assembly language) text is then substituted for the macro call. This text, of course, may contain additional macro definitions or calls.

   In summary: the macro processor must recognize and process macro definitions and macro calls.

The template to be followed for defining a **Procedure** is as follows:

PROC Proc_name ⟶ **Start of Definition**

_____

_____

_____

_____

RET

Proc_name ENDP **END of procedure**

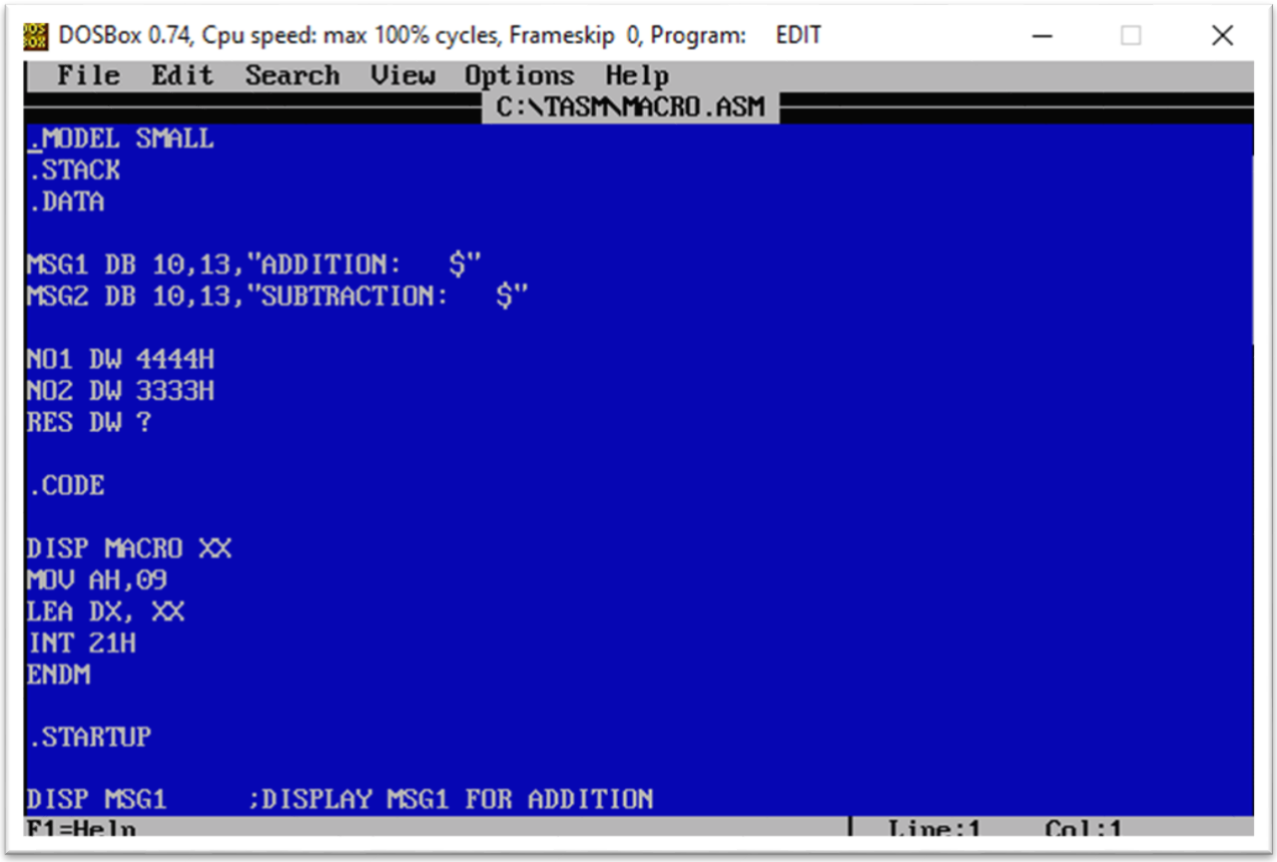
MACROS		PROCEDURE / Subroutine	
1	The corresponding machine code is written every time a macro is called in a program.	1	The Corresponding m/c code is written only once in memory
2	Program takes up more memory space.	2	Program takes up comparatively less memory space.
3	No transfer of program counter.	3	Transferring of program counter is required.

4	No overhead of using stack for transferring control.	4	Overhead of using stack for transferring control.
5	Execution is fast	5	Execution is comparatively slow.
6	Assembly time is more.	6	Assembly time is comparatively less.
7	More advantageous to the programs when repeated group of instruction is too short.	7	More advantageous to the programs when repeated group of instructions is quite large.

Application: Use of Macros and procedure in the Assembly Language programming to write modular program.

Design:

CODE:-



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: EDIT

File Edit Search View Options Help

C:\TASM\MACRO.ASM

```
.STARTUP

DISP MSG1      ;DISPLAY MSG1 FOR ADDITION

MOV AX,NO1
ADD AX,NO2
MOV RES,AX
CALL DISPLAY1

DISP MSG2      ;DISPLAY MSG2 FOR SUBTRACTION

MOV AX,NO1
SUB AX,NO2
MOV RES,AX
CALL DISPLAY1

JMP LAST


DISPLAY1 PROC
MOV BX,RES
AND BH,0F0H
```

F1=Help Line:41 Col:1

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: EDIT

File Edit Search View Options Help

C:\TASM\MACRO.ASM

```
DISPLAY1 PROC
MOV BX,RES
AND BH,0F0H
MOV CL,4
SHR BH,CL
ADD BH,30H
MOV DL,BH
MOV AH,02
INT 21H

MOV BX,RES
AND BH,0FH
ADD BH,30H
MOV DL,BH
MOV AH,02
INT 21H


MOV BX,RES
AND BL,0F0H
MOV CL,4
```

F1=Help | Line:58 Col:31

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: EDIT

File Edit Search View Options Help

C:\TASM\MACRO.ASM

```
MOV BX,RES
AND BL,0F0H
MOV CL,4
SHR BL,CL
ADD BL,30H
MOV DL,BL
MOV AH,02
INT 21H

MOV BX,RES
AND BL,0FH
ADD BL,30H
MOV DL,BL
MOV AH,02
INT 21H
RET
DISPLAY1 ENDP

LAST:
.EXIT
END
```

F1=Help | Line:55 Col:31

OUTPUT:-

```
C:\TASM>edit MACRO.asm

C:\TASM>tasm MACRO.ASM
Turbo Assembler  Version 3.0  Copyright (c) 1988, 1991 Borland International

Assembling file:   MACRO.ASM
Error messages:    None
Warning messages:  None
Passes:            1
Remaining memory:  475k


C:\TASM>tlink MACRO.OBJ
Turbo Link  Version 2.0  Copyright (c) 1987, 1988 Borland International

C:\TASM>MACRO.EXE

ADDITION:   7777
SUBTRACTION:   1111
C:\TASM>
```

Result and Discussion:-

1. **We develop a calculator for Addition and Subtraction for a 16 bits number using macros and procedure.**
2. **We understand and use macro processor to display message.**
3. **We also understand and use the procedure to display result.**

Learning Outcomes: The student should have the ability to

LO1: Explain how to use macros and procedure in the program.
LO2: Compare Macro and procedure.
LO3:Apply macros and procedure to implement the program.

Course Outcomes: Upon completion of the course students will be able to make use of instructions of 8086 to build assembly and Mixed language programs.

Conclusion:- **we learn about MACRO and Procedure and in make a calculator using it.**

For Faculty Use

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				