K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Batch: B1 **Roll No.: 1711072**

Experiment / assignment / tutorial No. 4

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Title: Program to calculate the factorial of a given number using FAR PROCEDURE or MACRO.

Objective: To understand types of procedure

Expected Outcome of Experiment:

CO 1: Explain the process of Compilation from Assembly language to machine language.

Books/ Journals/ Websites referred:

1.Microcomputer Systems: 8086/8088 family Architecture, Programming and Design: By Liu & Gibson (PHI Publication).

2.8086/8088 family: Design Programming and Interfacing: By John Uffenbeck(Pearson Education).

Pre Lab/ Prior Concepts:

Theory:

Procedure is basically group of instructions which can be used whenever we have to execute several times throughout the program

Syntax of procedure:

```
procedure_name PROC FAR
      ;STATEMENTS
ENDP procedure_name
```

Macros: When the repeated group of instruction is too short and not appropriate to be written as procedure, we use macro

Syntax of Macro:

```
macro_name MACRO
      ;STATEMENTS
ENDM macro_name
```

Instruction used:

1. Dec Instruction:
Syntax:
```
dec register_name
```

Eg: `dec CX`

2. Conditional jump instruction:

Syntax:
```
JNZ loop_name
```

Eg: `JNZ loop1`

3. Call instruction:
Syntax:

```
CALL function/macro name
```

Eg: `CALL factorial`

Return instruction:
Syntax:
```
RET
```
Eg: `RET`

Algorithm/Program for calculating the factorial:

Using far call procedure:

```
DATA SEGMENT
        n dw 05h
DATA ENDS

RANDOM SEGMENT
factorial PROC FAR
        MUL CX
        DEC CX
        RET
ENDP factorial
RANDOM ENDS

CODE SEGMENT
        ASSUME CS: CODE, DS: DATA
START:
        MOV AX,DATA
        MOV DS, AX
        MOV CX, 05H
        MOV AX, 01H
        ATHAVALE:
                CALL factorial
                JNZ ATHAVALE
        MOV AX, 4CH
        INT 21H
CODE ENDS
END START
ENDS
```

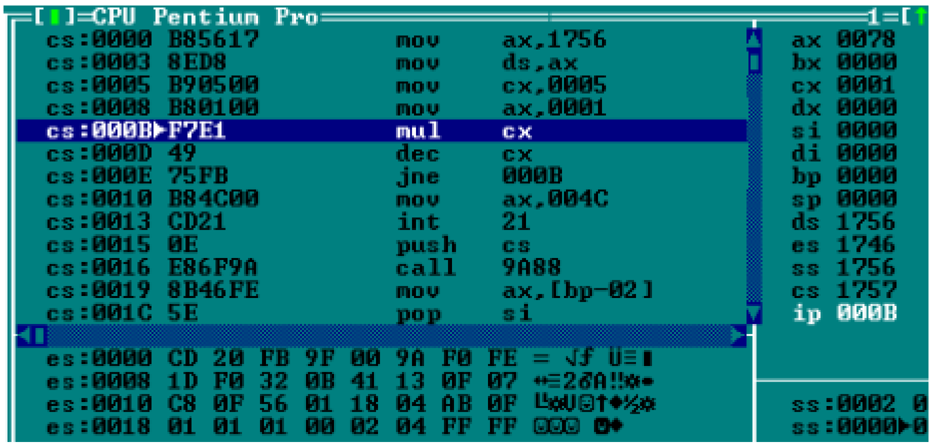
Using macro:

```
DATA SEGMENT
        n dw 05h
DATA ENDS

RANDOM SEGMENT
factorial MACRO
        MUL CX
        DEC CX
ENDM factorial
RANDOM ENDS
```

```
CODE SEGMENT
        ASSUME CS: CODE, DS: DATA
START:
        MOV AX,DATA
        MOV DS, AX
        MOV CX, 05H
        MOV AX, 01H
        ATHAVALE:
                factorial
                JNZ ATHAVALE
        MOV AX, 4CH
        INT 21H
CODE ENDS
END START
ENDS
```

Output:



```
=[ ]=CPU Pentium Pro===================================1=[↑]
 cs:0000 B85617        mov     ax,1756           ax 0078
 cs:0003 8ED8          mov     ds,ax             bx 0000
 cs:0005 B90500        mov     cx,0005           cx 0001
 cs:0008 B80100        mov     ax,0001           dx 0000
 cs:000B►F7E1          mul     cx                si 0000
 cs:000D 49            dec     cx                di 0000
 cs:000E 75FB          jne     000B              bp 0000
 cs:0010 B84C00        mov     ax,004C           sp 0000
 cs:0013 CD21          int     21                ds 1756
 cs:0015 0E            push    cs                es 1746
 cs:0016 E86F9A        call    9A88              ss 1756
 cs:0019 8B46FE        mov     ax,[bp-02]        cs 1757
 cs:001C 5E            pop     si                ip 000B

 es:0000 CD 20 FB 9F 00 9A F0 FE  = √ƒ Ü≡▮
 es:0008 1D F0 32 0B 41 13 0F 07  ↔≡2♂A‼☼•       ss:0002 0
 es:0010 C8 0F 56 01 18 04 AB 0F  ╚☼V☺↑♦½☼
 es:0018 01 01 01 00 02 04 FF FF  ☺☺☺ ☻♦        ss:0000►0
```

Conclusion: The program ran successfully as we were able to calculate factorial of a number using both far call procedure and macro.

Post Lab Descriptive Questions (Add questions from examination point view)
Comparison between Procedure & Macro:

Ans.

Macros	Procedures
Accessed during assembly when name given to macro is written as an instruction in the assembly program.	Accessed by CALL and RET instructions during program execution.
Machine code is generated for instructions each time a macro is called.	Machine code for instructions is put only once in the memory.
This due to repeated generation of machine code requires more memory.	This as all machine code is defined only once so less memory is required.
Parameters are passed as a part of the statement in which macro is called.	Parameters can be passed in register memory location or stack.

Date: 11/02/2019 **Signature of faculty in-charge**