NAME : Soham Manoj Pawar **Class : SE- IT**

ROLL NO : 71 **MPL LAB**

EXPERIMENT 11

AIM :-

Compute the factorial of a positive integer ‘n’ using recursive procedure.

LO No :- LO4

LO : Write programs based on string and procedure for 8086 microprocessor.

SOFTWARE :- Tasm Software

Theory :-

Assume CS : code , DS : data

- ASSUME statement can assign up to 4 segment registers in any sequences.
- DS: Data means that the assembler is to associate the name of data segment with DS register.

Similarly CS: Code tells the assembler to associate the name of code segment with CS register and so on.

MOV Ax,data MOV Ds,Ax :-

Initialize Ds to point start of memory, set alongside to store data.

PROC :-

Procedure is a part of code that can be called from your program in order to make some specific task. Procedures make program more structural and easier to understand. Generally procedure returns to the same point from where it was called.

MUL :-

The MUL (Multiply) instruction used in 8086 microcontroller to handles unsigned data. There are types of multiplication depending on the number of bits:

- Byte with byte
- Word with Word
- Byte with word

CALL :-

The **CALL instruction** is used whenever we need to make a call to some procedure or a subprogram. Whenever a **CALL** is made, the following process takes place inside the microprocessor:

- The address of the next instruction that exists in the caller program (after the program CALL instruction) is stored in the stack.
- The instruction queue is emptied for accommodating the instructions of the procedure.
- Then, the contents of the instruction pointer (IP) is changed with the address of the first instruction of the procedure.
- The subsequent instructions of the procedure are stored in the instruction queue for execution.

The Syntax for the CALL instruction is as follows:

```
    CALL subprogram_name
```

RET :-

The **RET instruction** stands for return. This instruction is used at the end of the procedures or the subprograms. This instruction transfers the execution to the caller program. Whenever the **RET instruction** is called, the following process takes place inside the microprocessor:

- The address of the next instruction in the mainline program which was previously stored inside the stack is now again fetched and is placed inside the instruction pointer (IP).
- The instruction queue will now again be filled with the subsequent instructions of the mainline program.

The Syntax for the RET instruction is as follows:

```
      RET
```

NAME : Soham Manoj Pawar **Class : SE- IT**

ROLL NO : 71 **MPL LAB**

INT 21H :-

int 21h means, call the interrupt handler 0x21 which is the DOS Function dispatcher. the "mov ah,01h" is setting AH with 0x01, which is the Keyboard Input with Echo handler in the interrupt.

Syntax: int 21H
Example: int 21

Code :-

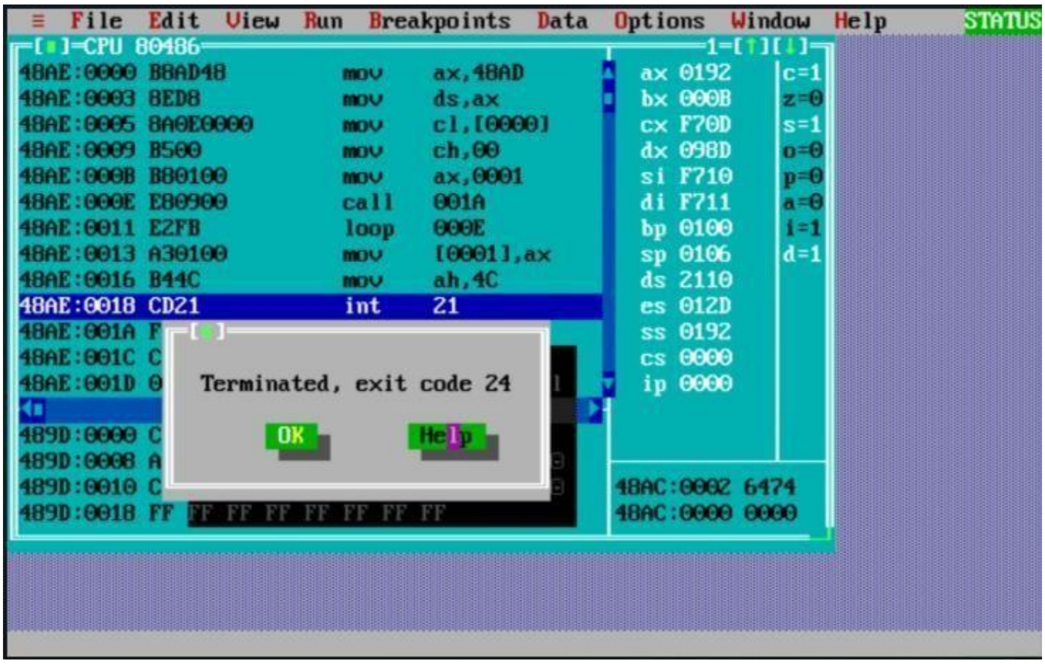
```
Assume    CS:Code,    DS:Data
DataSegment
       num  db  04H
ans dw 01 dup(?)
Data Ends
Code          Segment
Start:    MOV    AX,
Data
      MOV DS, AX
      MOV CL, num
      MOV CH, 00H
      MOV AX, 0001H
      Back: Call
      PROFACLOOP
      Back
      MOV ans,
      AX MOV
      AH, 4CH
      INT 21H


      PROFAC proc
      nearMUL CL
      RET
      PROFAC
    EndpCode
    Ends
    End Start
```

NAME : Soham Manoj Pawar **Class : SE-**

ITROLL NO : 71 **MPL LAB**

Output :-



```
48AE:0000 B8AD48        mov    ax,48AD          ax 0192   c=1
48AE:0003 8ED8          mov    ds,ax            bx 000B   z=0
48AE:0005 8A0E0000      mov    cl,[0000]        cx F70D   s=1
48AE:0009 B500          mov    ch,00            dx 098D   o=0
48AE:000B B80100        mov    ax,0001          si F710   p=0
48AE:000E E80900        call   001A             di F711   a=0
48AE:0011 E2FB          loop   000E             bp 0100   i=1
48AE:0013 A30100        mov    [0001],ax        sp 0106   d=1
48AE:0016 B44C          mov    ah,4C            ds 2110
48AE:0018 CD21          int    21               es 012D
48AE:001A F                                     ss 0192
48AE:001C C                                     cs 0000
48AE:001D 0                                     ip 0000

489D:0000 C
489D:0008 A
489D:0010 C                                     48AC:0002 6474
489D:0018 FF FF FF FF FF FF FF FF               48AC:0000 0000
```

```
Terminated, exit code 24
```

Conclusion :

Thus, in this experiment we learnt to compute factorial of a positive integer ‘n’ using recursive procedure.

NAME : Soham Manoj Pawar **Class : SE- IT**

ROLL NO : 71 **MPL LAB**