Experiment 3

Name:Pratik Chavan

Div/Batch:A/A1   Roll No.07

```
.MODEL SMALL
.STACK 100H
.DATA
    NUM DB 5        ; Number for factorial (change this value as needed)
    FACT DW 1       ; Variable to store factorial result
    MSG DB 'Factorial: $'  ; Message to display before result

.CODE
MAIN PROC
    MOV AX, @DATA
    MOV DS, AX

    MOV AL, NUM      ; Load number in AL
    CBW            ; Convert AL to AX (sign-extend)
    CALL FACTORIAL   ; Call factorial procedure

    MOV DX, OFFSET MSG
    MOV AH, 09H
    INT 21H         ; Print message

    CALL PRINT_NUM   ; Print the factorial result

    MOV AH, 4CH
```

```
   INT 21H         ; Exit program
MAIN ENDP

; Factorial Procedure
FACTORIAL PROC
   MOV CX, AX      ; Move number to CX for loop counter
   MOV AX, 1       ; Initialize AX = 1 (Factorial starts at 1)

FACTORIAL_LOOP:
   MUL CX          ; AX = AX * CX
   LOOP FACTORIAL_LOOP

   MOV FACT, AX    ; Store the result in FACT
   RET
FACTORIAL ENDP

; Print Number Procedure
PRINT_NUM PROC
   MOV AX, FACT    ; Load factorial result
   MOV CX, 0       ; Clear CX (digit counter)

NEXT_DIGIT:
   MOV DX, 0
   MOV BX, 10
   DIV BX          ; AX / 10 → Quotient in AX, Remainder in DX
   PUSH DX         ; Push remainder (digit) onto stack
   INC CX          ; Increment digit counter
   TEST AX, AX     ; Check if AX is zero
```

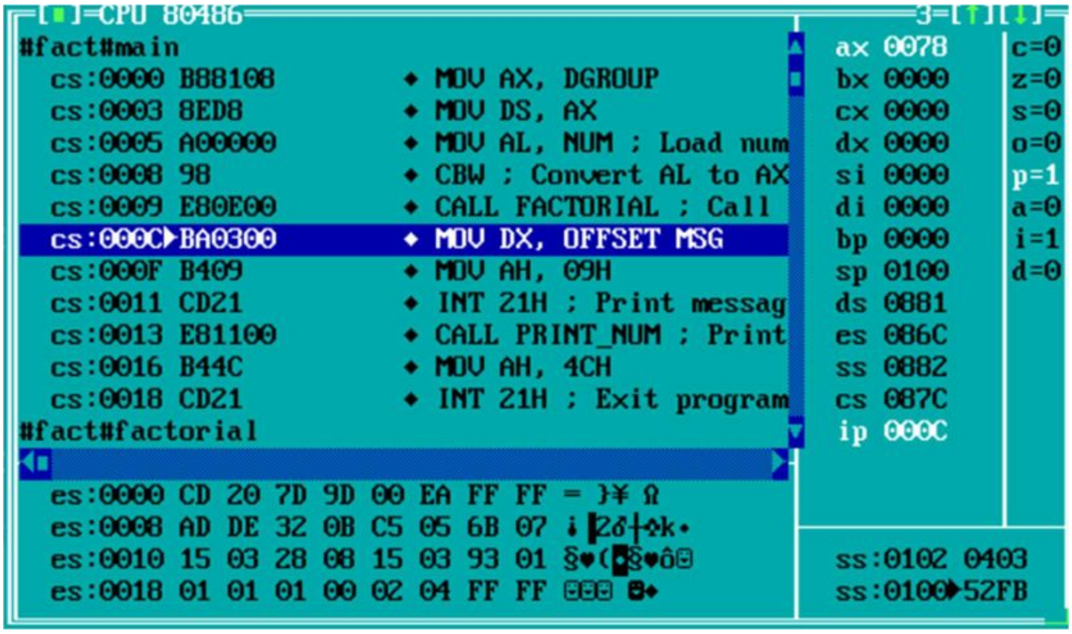
```
    JNZ NEXT_DIGIT   ; If not, continue extracting digits


PRINT_LOOP:
    POP DX          ; Get digit from stack
    ADD DL, '0'     ; Convert to ASCII
    MOV AH, 02H
    INT 21H         ; Print digit
    LOOP PRINT_LOOP  ; Repeat for remaining digits


    RET
PRINT_NUM ENDP


END MAIN
```

OUTPUT:



```
#fact#main
  cs:0000 B88108        ◆ MOV AX, DGROUP
  cs:0003 8ED8          ◆ MOV DS, AX
  cs:0005 A00000        ◆ MOV AL, NUM ; Load num
  cs:0008 98            ◆ CBW ; Convert AL to AX
  cs:0009 E80E00        ◆ CALL FACTORIAL ; Call
  cs:000C▶BA0300        ◆ MOV DX, OFFSET MSG
  cs:000F B409          ◆ MOV AH, 09H
  cs:0011 CD21          ◆ INT 21H ; Print messag
  cs:0013 E81100        ◆ CALL PRINT_NUM ; Print
  cs:0016 B44C          ◆ MOV AH, 4CH
  cs:0018 CD21          ◆ INT 21H ; Exit program
#fact#factorial

  es:0000 CD 20 7D 9D 00 EA FF FF = }¥ ß
  es:0008 AD DE 32 0B C5 05 6B 07 ¡▐2δ┼♧k•
  es:0010 15 03 28 08 15 03 93 01 §♥(◘§♥ô☺
  es:0018 01 01 01 00 02 04 FF FF ☺☺☺ ☻♦
```

```
ax 0078    c=0
bx 0000    z=0
cx 0000    s=0
dx 0000    o=0
si 0000    p=1
di 0000    a=0
bp 0000    i=1
sp 0100    d=0
ds 0881
es 086C
ss 0882
cs 087C
ip 000C

ss:0102 0403
ss:0100▶52FB
```