```
; Factorial program
; Eliav Gnessin, Fall 2002
; ========================================
; This is an example program in 8086 Assembly
; ========================================

TITLE FACT

; This instruction defines the memory model that MASM or TASM use
.model small

; Define the stack size. This instruction initializes the SP.
.stack 100h

; Variables & other definitions section
.data
N       dw 7            ; this is the input parameter
arg     dw 10           ; this is the output base for printax
zero    dw 0

; This is the program itself
.code
start:  mov ax,@data        ; Since the .data instruction doesn't initialize
        mov ds,ax           ; the ds register we have to do it manually

        mov bx,[N]          ; move our ds parameter to the input register
        call factorial      ; compute
        call printax        ; print the result

        mov ax,4c00h        ; This is the program terminator
        int 21h             ; just like putting "return 0" in C

; ================================================
; Procedure definitions
; ================================================

; ========================================================
; Procedure name: factorial - recursively compute BX!
; Input:          BX - the number to compute factorial of
; Output:         DXAX - BX!
; ========================================================
factorial proc near
          cmp bx,1              ; halt condition
          jg re_call            ; if not - do recursion
          mov ax,1              ; initialization
          jmp done
re_call:  push bx               ; put parameter in stack
          dec bx                ; N=N-1
          call factorial        ; recursive call
          pop bx                ; get parameter from stack
          mul bx                ; compute F(N)=F(N-1)*N
done:     ret
factorial endp


; ========================================================
; Procedure name: printax - print AX register in base arg
; Input:          AX - the number to be printed
;                 arg - output base [2-10]
; Output:         None
; ========================================================
printax proc near
```

```
        mov si,0            ; si will count the num of digits
again4: mov dx,0
        div arg             ; AX/arg-> reminder is in DX
        add dx,30h          ; convert value to ASCII: 0-9 => "0"-"9"
        push dx             ; Store in stack
        inc si
        cmp zero,ax         ; if the quotient is 0, we are finished
        mov cx,2            ; make sure the loop doesn't finish because
                            ;  CX=0
        loopnz again4

        ; Move down to next line - Carriage Return + Line Feed
        mov cx,si           ; CX will count the result's digits

        mov al,10           ; Print CR + LF
        call printch
        mov al,13
        call printch

again5: pop ax              ; get result from stack and print it
        call printch
        loop again5

        ret
printax endp

; ======================================================
; Procedure name: printch - Print a char to console
; Input:          AL - the char's ASCII code
; Output:         None
; ======================================================
printch proc near
        mov bx,0            ; No color definitions
        mov ah,0Eh          ; Print char to TTY function code
        int 10h             ; Call
        ret
printch endp

; End of program
end start
```