

```

; Linked List 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
X1 dw -1,-1      ; |
X2 dw 3,X1       ; | Linked list nodes
X3 dw 0,X2       ; |
mylist dw 0,X3   ; |

; When using DOS's print string routine, $ is the string terminator
outstr db "The number of zeros is $"

sum db ?         ; check - output parameter for counting
printbase dw 10 ; printax - print base
zero dw 0        ; printax - zero

; 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 dx, offset outstr ; This time we're using DOS's print string
        mov ah, 9h       ; routine
        int 21h

        mov bx,offset mylist ; move the linked list address to bx
        call check        ; count number of zeros

        mov ah,0
        mov al,sum
        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: check - count number of zeros in a linked list
; Input:          BX - the address of the linked list
; Output:         sum - the number of zeros
; =====
check proc near
        cmp bx,-1        ; halt condition
        jne re_call
        mov sum,0        ; initialize counter
        jmp sof
re_call: push bx

```

```

        mov bx,[bx+2]          ; this goes through the list's
                                ; nodes
        call check            ; recursion
        pop si
        cmp [si],word ptr 0   ; found a zero
        jne sof
        inc sum                ; increase zero counter
sof:    ret
check endp

; =====
; Procedure name: printax - print AX register in base
; Input:         AX - the number to be printed
;               printbase - output base [2-10]
; Output:        None
; =====
printax proc near
        mov si,0              ; si will count the num of digits
again4: mov dx,0
        div printbase        ; AX/arg-> remainder 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

```