

```

;*****
;*          T E S T C O M . A S M          *
;*-----*
;* Task      : Simple COM program: can be assembled using *
;*            either Turbo Assembler (TASM) or Microsoft's *
;*            Macro Assembler (MASM) *
;*-----*
;* Author    : MICHAEL TISCHER *
;* Developed on : 06/07/1987 *
;* Last update : 12/20/1991 *
;*-----*
;* Assembly  : MASM:      masm testcom; *
;*            link testcom; *
;*            exe2bin testcom.exe testcom.com *
;*            *
;*            TASM:      tasm testcom *
;*            tlink /t testcom *
;*****

com      segment para 'CODE'      ;Definition of COM segment
                                   ;(freely selectable name)

        org 100h                  ;Code begins at address 100H
                                   ;immediately following the PSP

        assume cs:com, ds:com, es:com, ss:com

                                   ;All segments point to the
                                   ;COM segment

start:   jmp  init                ;Program starts here
                                   ;Jump to initialization

;== Data =====

        ;-- Data, buffer and variables -----
        ;-- can be stored here -----

        i...
        i...
        i...

;== Program =====

prog     proc near                ;This procedure is the actual
                                   ;main program and is executed
                                   ;after initialization

        ;-- Main program code -----
        ;-- can be inserted here -----

        i...
        i...
        i...

        ;--- Call DOS function 4CH to end program -----

        mov  ax,4C00h             ;Load function number 4CH, error code 0
        int  21h                 ;DOS interrupt call

        ;--- DOS interrupt 21H ends program, so no --
        ;--- more program cannot be added here -----

prog     endp                    ;End program

;-- Other procedures -----
;-- Provisions for subroutines

a_proc   proc near

        i...
        i...
        i...
        ret

a_proc   endp

```

```

b_proc    proc near

            ;...
            ;...
            ;...
            ret

b_proc    endp

;-- Initialization -----
;-- ENDE releases all memory and releases the stack

init:      mov  ah,4Ah                ;Function number - 'change memory size'
            mov  bx,offset ende       ;Length of program in memory
            add  bx,15                ;Round off to next paragraph
            mov  cl,4                 ;Compute offset in
            shr  bx,cl                ;paragraphs
            inc  bx
            int  21h                 ;Call DOS interrupt 21H

            mov  sp,offset ende       ;Remove stack
            jmp  prog                 ;Return to main program

init_ende label near

;== Stack =====
            dw (256-((init_ende-init) shr 1)) dup (?)

                                ;The stack comprises 256 words and
                                ;includes the INIT routine code
                                ;(INIT code no longer needed after
                                ;initial routine call)

ende       equ this byte          ;End of allocated memory (no
                                ;code after this)

;== Ende =====

com        ends                   ;End of COM segment
            end  start             ;End assembler programs - call
                                ;START to re-execute

```