

page ,132
; Toth Tamas - TOTPABT.SZE
; A N-edik hatvanya

```
code          segment      para public 'code'
              assume       cs:code,ds:data,ss:stack,es:nothing

hatvany      proc  far
              push  ds          ; visszatérési cím a verembe
              xor   ax,ax       ; ax = 0
              push  ax          ; visszatérés offset címe
              mov   ax,data     ; ds a data szegmensre mutasson
              mov   ds,ax       ; sajnos "mov ds,data"
                                ; nem megengedett

              MOV   WORD PTR E, 1   ;E=00000001H
              MOV   WORD PTR E+2, 0
              MOV   AX, A           ;A2k=A
              MOV   A2k, AX
              XOR   CH, CH         ;CH=00
              MOV   CL, N         ;CL=N, tehát CX=N

              JCXZ  kesz           ;0. hatvany eseten rogtan vegeztunk

szoroz:      SHR   CX,1           ;CX utolso bitje Carrybe kerul
              JNC  next          ;ha 0, akkor ugrunk
              MOV  AX, WORD PTR E ;ha nem 0, akkor E*A2k
              MUL  A2k

              MOV  WORD PTR E, AX ;a szorzatot E-ben taroljuk
              MOV  WORD PTR E+2, DX

next:        JCXZ  kesz           ;ha cx nulla, keszen vagyunk
              MOV  AX,A2k        ;kulonben, A kovetkezo hatvanya A2k*A2k
              MUL  A2k
              MOV  A2k, AX
              JMP  szoroz        ;vissza az elejere, CX-et ujbol leptetjuk

kesz:        MOV  AX, WORD PTR E+2 ;vegeredmeny kiirasa
              call hexa
              MOV  AX, WORD PTR E
              call hexa
              MOV  SI, offset kvse
              call kiiro

              ret                ; vissza az Op. rendszerhez
hatvany      endp                ; a skalár eljárás vége
; -----
```

```

hexa      proc                ; ax kiírása hexadeci málisan
          xchg ah,al          ; ah és al felcserélése
          call hexa_b         ; al (az eredeti ah) kiírása
          xchg ah,al          ; ah és al visszacserélése
          call hexa_b         ; al kiírása
          ret                 ; visszatérés
hexa      endp                ; a hexa eljárás vége
; -----
hexa_b    proc                ; al kiírása hexadecimálisan
          push cx             ; mentés a verembe
          mov cl,4            ; 4 bit-es rotálás előkészítése
          ROR al,CL           ; az első jegy az alsó 4 biten
          call h_jegy         ; az első jegy kiírása
          ROR al,CL           ; a második jegy az alsó 4 biten
          call h_jegy         ; a második jegy kiírása
          pop cx              ; visszamentés a veremből
          ret                 ; visszatérés
hexa_b    endp                ; a hexa_b eljárás vége
; -----
h_jegy    proc                ; hexadecimális jegy kiírása
          push ax             ; mentés a verembe
          AND al,0FH          ; a felső 4 bit 0 lesz,
                                ; a többi változatlan
          add al,'0'          ; + 0 kódja
          cmp al,'9'         ; <= 9 ?
          JLE h_jegy1        ; ugrás h_jegy1 -hez, ha igen
          add al,'A'-'0'-0AH ; A-F hexadecimális jegyek
                                ; kialakítása
h_jegy1:  mov ah,14           ; BIOS szolgáltatás előkészítése
          int 10H             ; BIOS hívás: karakter kiírás
          pop ax              ; visszamentés a veremből
          ret                 ; visszatérés
h_jegy    endp                ; a hexa_b eljárás vége
; -----
kiiro     proc                ; szöveg kiírás (DS:SI)-től
          push ax
          cld
ki1:      lodsb               ; al=a következő karakter
          cmp al,0            ; al=? 0
          je ki2              ; ugrás a ki2 címkéhez, ha al=0
          mov ah,14           ; BIOS rutin paraméterezése
          int 10H             ; az AL-ben lévő karaktert
; kiírja a képernyőre
          jmp ki1             ; a kiírás folytatása
ki2:      pop ax
          ret                 ; visszatérés a hívó programhoz
kiiro     endp                ; a kiiro eljárás vége
; -----
code      ends                ; a code szegmens vége
; =====

```

