```
; Assembly főprogram, amely adott szöveget ír a képernyőre
; ================================================
KOD  SEGMENT PARA  PUBLIC  'CODE' ; Szegmens kezdet
; KOD: a szegmens neve
; align-type (igazítás típusa): BYTE, WORD, PARA, PAGE
; combine-type: PUBLIC, COMMON, AT <kifejezés>, STACK
; class: 'CODE', 'DATA', ('CONSTANT',) 'STACK', 'MEMORY'
;              ajánlott értelemszerűen
      ASSUME  CS:KOD, DS:ADAT, SS:VEREM, ES:NOTHING
; feltételezett szegmens regiszter értékek.
; A beállításról ez az utasítás nem gondoskodik!
```



```
KIIR     PROC     FAR  ; A fő eljárás mindig FAR
                       ; FAR: távoli, NEAR: közeli eljárás
; Az operációs rendszer úgy hívja meg a főprogramokat, hogy
; a CS és IP a program végén lévő END utasításban megadott
; címke szegmens és OFFSET címét tartalmazza, SS és SP a
; a STACK kombinációs típusú szegmens végét mutatja,
; a visszatérés szegmens címe DS-ben van, OFFSET-je pedig 0
         PUSH     DS        ; DS-ben van a visszatérési cím
                            ; SEGMENT része
         XOR      AX, AX    ; AX⇐0, az OFFSET rész = 0
         PUSH     AX        ; Veremben a (FAR) visszatérési cím
         MOV      AX, ADAT  ; AX⇐ az ADAT SEGMENT címe
         MOV      DS, AX
; Most már teljesül, amit az ASSUME utasításban írtunk
; Eddig tartott a főprogram előkészületi része
```



```
         MOV      SI, OFFSET SZOVEG
                            ; SI⇐SZÖVEG OFFSET címe
         CLD                ; a SZÖVEGet növekvő címek
                            ; szerint kell olvasni
         CALL     KIIRO     ; Eljárás hívás

         RET                ; Visszatérés az op. rendszerhez
                            ; a veremből visszaolvasott
                            ; szegmens és OFFSET címre

KIIR     ENDP               ; A KIIR eljárás vége
```



```
KIIRO    PROC               ; NEAR eljárás,
                            ; megadása nem kötelező
CIKLUS:  LODSB              ; AL⇐a következő karakter
         CMP      AL, 0     ; AL =? 0
         JE       VEGE      ; ugrás a VEGE címkéhez,
                            ; ha AL=0
         MOV      AH, 14    ; BIOS rutin paraméterezése
         INT      10H       ; a 10-es interrupt hívása:
                            ; az AL-ben lévő karaktert kiírja
                            ; a képernyőre
         JMP      CIKLUS    ; ugrás a CIKLUS címkéhez,
                            ; a kiírás folytatása
VEGE:    RET                ; Visszatérés a hívó programhoz
KIIRO    ENDP               ; A KIIRO eljárás vége
KOD      ENDS               ; A KOD szegmens vége
```



```
ADAT     SEGMENT       PARA   PUBLIC   'DATA'
SZOVEG   DB        'Ezt a szöveget kiírja a képernyőre'
         DB        13, 10, 0  ; 13: a kocsi vissza,
                              ; 10: a soremelés kódja,
                              ;  0: a szöveg vége jel
ADAT     ENDS                ; Az ADAT szegmens vége
; ================================================
VEREM    SEGMENT       PARA   STACK
         DW        100  DUP (?)     ; Helyfoglalás 100 db
                              ; inicializálatlan szó számára
VEREM    ENDS                ; A VEREM szegmens vége
; ================================================
         END       KIIR      ; Modul vége,
                              ; a program kezdőcíme: KIIR
```


Az I8086/8088 utasítás rendszere

Jelölések

⇐ : értékadás

⇔ : felcserélés

op, op1, op2: tetszőlegesen választható operandus (közvetlen, memória vagy regiszter).

op1 és **op2** közül az egyik regiszter kell legyen!

reg: általános, bázis vagy index regiszter

mem: memória operandus

ipr: (8 bites) **IP** relatív cím

port: port cím (8 bites eltolás vagy **DX**)

[op]: az **op** által mutatott cím tartalma

Adat mozgató utasítások

Nem módosítják a flag-eket (kivéve **POPF** és **SAHF**)

```
MOV   op1, op2  ; op1 ⇐ op2 (MOVe)
XCHG  op1, op2  ; op1 ⇔ op2 (eXCHanGe), op2 sem
                ; lehet közvetlen operandus
XLAT            ; AL ⇐ [BX+AL] (trans(X)LATe), a
                ; BX által címzett maximum 256 byte-
                ; os tartomány AL-edik byte-jának
                ; tartalma lesz AL új tartalma
LDS   reg, mem  ; reg ⇐ mem, mem+1
                ; DS ⇐ mem+2, mem+3 (Load DS)
LES   reg, mem  ; reg ⇐ mem, mem+1
                ; ES ⇐ mem+2, mem+3 (Load ES)
LEA   reg, mem  ; reg ⇐ mem effektív (logikai) címe
                ; (Load Effective Address)
```


A veremmel (stack-kel) kapcsolatos adat mozgató utasítások:

```
PUSH    op  ; SP ⇐ SP-2; (SS:SP) ⇐ op
PUSHF       ; (PUSH Flags)
            ; SP ⇐ SP-2; (SS:SP) ⇐ STATUS
POP     op  ; op ⇐ (SS:SP); SP ⇐ SP+2
POPF        ; (POP Flags)
            ; STATUS ⇐ (SS:SP); SP ⇐ SP+2
```

Az Intel 8080-nal való kompatibilitást célozza az alábbi két utasítás:

```
SAHF        ; STATUS alsó 8 bitje ⇐ AH
LAHF        ; AH ⇐ STATUS alsó 8 bitje
```