

Az Országos Commodore Egyesület lapja

újság

1989 / 7-8

PROFI DESIGN  
GEOS-HÍREK  
ÖTLETBÖRZE  
AMIGÁRÓL  
KOMOLYAN KELL VENNII!  
FEKETÉN-FEHÉREN  
SZÖVEGSZERKESZTŐ „80”  
ÉKEZETES C—16







## **FIZIKA ÉS SZÁMÍTÁSTECHNIKA**

### **Rezgések, Hőtan, Elektromosság**

A könyv a Fizika és számítástechnika — Mechanika folytatásaként, annak szellemében a gimnáziumok harmadik osztályának anyagát dolgozza fel.

Ez a kötet a rezgő- és hullámmozgással lezárja a mechanika témakörét, és tárgyalja a hőtan, az elektrosztatika és az egyenáramok fejezeteit.

Ebben a könyvben is jó néhány számítógépes program (demonstrációs, mérő, illetve feladatmegoldó) segíti az ismeretek elmélyítését.

**Ára: 149 Ft**

**Megjelent: 1989**



## MIT, HOGYAN, HOL, MIKOR?

**EGYESÜLETI ÜGYEK:** Egyesületünk tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az Egyesület irodájában (1133 Budapest, Kárpát u. 7/a. I. em. 11., tel: 497-559), vagy átutalással az MNB 217-98292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

Minden tagunk rendelkezésére áll PÖTYÖGŐSZOLGÁLTATUNK, a szervizkedvezmény és az apróhirdetési lehetőség. A Deákpáholy tagjai minden hónapban megkapják a C-újságot, a tagsági díj fél évre 366 forint.

A Pluszpáholy tagjai minden hónapban megkapják a C-újságot, és minden hónapban 2x60 forint vásárlási utalványt is kapnak. A tagsági díj fél évre 911 forint.

A Szuperpáholy tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 1800 forint vásárlási utalványt. A féléves tagsági díj 10 000 forint.

**ÜGYFÉLFOGADÁS:** minden kedden és csütörtökön 14—18 óra között várjuk tagjainkat és az érdeklődőket.

**PÖTYÖGŐSZOLGÁLTAT:** Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az Egyesület irodájában, vagy postai utánvétellel.

A pöttyögőszolgálat minden páros héten csütörtökön az egyesületi irodában működik, 16—19 óra között.

**APRÓHIRDETÉS:** Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 80 forint. A hirdetés

### A kedvezmények a következő vidéki könyvesboltok 2C sarkaiban válthatók be.

**PÉCS:** Zrínyi Miklós Könyvesbolt. 7621 Jókai u. 25. Tel.: 72-12835

**DEBRECEN:** Szak- és ismeretterjesztő Könyvúrház. 4024 Hunyadi u. 8. Tel.: 52-23237

**SZOMBATHELY:** Savaria Könyvesbolt. 9700 Mártírok tere 1. Tel.: 94-12341

**VESZPRÉM:** Kölcsey Ferenc Könyvesbolt. 8200 Cserhát út 7.

**BÉKÉSCSABA:** Radnóti M. Könyvesbolt. 5600 Tanácsköztársaság út 2. Tel.: 25-207

**GYŐR:** Pattantyús Á. Géza Szakkönyvesbolt. 9021 Molnár Ferenc u. 9.

**SZEGED:** Tömörkény Könyvesbolt. 6720 Lenin krt. 48. Tel.: 62-21453

**SZOLNOK:** Szigligeti Könyvesbolt. 5000 Ságvári krt. 35. Tel.: 56-11133

**MISKOLC:** Chip-kuckó. 3530 Tanácsház tér 14.

módja: az újságban megjelenő nyomtatvány kitöltésével.

A C-újság régebbi számai megvásárolhatók az Egyesület irodájában, vagy megrendelhetők utánvétellel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk a 497-559-es telefonszámon, vagy levélben!

Vidéki Pluszpáholy-tagjaink három havi tikkett összegyűjtésekor igénybe vehetik a NOVOTRADE 2C áruház ingyenes csomagküldő szolgálatát.

Vidéken további információk kaphatók:

Jászberényi Városi Könyvtár

Győri Bartók Béla Művelődési Ház

Zalaegerszegi Ságvári Endre Gimnázium

Pécsi Apáczai Csere János Gimnázium

### Budapesten havonta klubdélelőtt a Petőfi Csarnokban.

**Egyesületünk és az újság szerkesztősege várja az Önök programjait, hardver leírásait, cikkeiket. Szerkesszük együtt az új-ságot!**

### Tisztelt Szerkesztőség!

Mindenekelőtt egy-két észrevétel a lappal kapcsolatban:

A Pöttyögőszolgálat listája sokkal használhatóbb lenne, ha géptípusok szerint lennének csoportosítva a programok, hiszen a más géppel rendelkezőket általában nem érdeklik a többi gépen futó programok, vagy ha mégis, akkor így is megtalálnák azokat. Ezek mellett a nem gépfüggő (poke és sys nélküli) BASIC programokat is külön csoportokba sorolnám a verziószámok szerint (2.0, 3.5, 7.0). Természetesen egy csoporton belül az eredeti sorszám szerint állnának a programok. Ezt a listát játszva elő lehetne állítani és folyamatosan karbantartani pl. a Multiplan (C-64) programmal — a lista végére beírva az új adatokat, utána a „sort” funkcióval a gép berendezi őket a saját helyükre.

A programlista „hosszabbítására” fel kellene használni az apróhirdetési űrlap jelenlegi helyét, azt könnyen be lehetne szerkeszteni máshova.

Kicsit „kiakadtam” a programot bevezető kis rajzocskán, amely (az áprilisi számban például háromszor) azt ábrázolja, amint két fiatal nagy érdeklődéssel dolgozik egy ZX-SPECTRUM számítógéppel!!! Persze lehet, hogy ami nekem cikis, az másnap a manapság olyan divatos pluralizmus megnyilvánulása...

Üdvözlettel:

További jó munkát kívánok!

**Bognár Attila**

Végül ide listázok egy rövidke rutint, amely olyan programokba építhető be, amelyek személyi szám inputtal dolgoznak. Ez az algoritmus nem a 13. (stb.) hónap és a 32. (stb.) nap „kiszűrésével” dolgozik, de azért elég hatásosan ellenőrzi a bevitt adat helyességét.

```

○ 10 REM SZEME'LYISZA'M ELL.
    15 INPUT"NYELVSZEM. SZA'M";SZ$
    20 :
○ 25 S=0:J=0
    30 FORI=1TO10:J=J+1
    35 S$=MID$(SZ$,J,1)
○ 40 IFS$=" "ÜRS$=" " THENJ=J+1:GOTO35
    45 S=S+VAL(S$)*I
○ 50 NEXT I
    55 IFVAL(RIGHT$(SZ$,1))=
        INT((S/11-INT(S/11))*11+.001)THEN65
○ 60 PRINT"NYELVHELYTELEN !!!":STOP
    65 :
    70 :
○ 75 PRINT"NYELVREND BEN !"
    80 END
    
```

Az Országos Commodore Egyesület módszertani kiadványa.

Egyesületi iroda és szerkesztőség: 1133 Budapest, Kárpát u. 7/a I. em. 11. Tel.: 497-559

Felelős kiadó: Horváth Judit, az egyesület elnöke

Főszerkesztő: Rados Péter

Felelős szerkesztő: Dr. Horváth András

Művészeti szerkesztő: Szulyovszky József

Egyesületi szervező: Winter Júlia

Lapmenedzser: Kovács Gábor

Levélcím: Commodore Újság, 1388 Bp. 62. Pf. 86.

Index: ISSN 0237-756 X

Terjeszti a Magyar Posta.

Megvásárolható a hírlapárusoknál.

MSZH Nyomda



## Tisztelt Tagtársaim!

Két éve írogatok C64-en, illetve PLUS/4-en, néhány trükköt és rutint szeretnék átadni, amit mások is használhatnak.

### "GEN-GET-INPUT' 64"

Ez egy adatbeviteli rutin C64-re. A "gen" arra utal, hogy generálható a várható válasz, a "get-input" azt jelzi, hogy a karakterek bevitelére 'get'-tel történik.

A rutint egy adatbekérő programrészlettel mutatom be, a személyi szám ellenőrző rutin (gosub10210) nem az én művem, így jól működik.

Futtatáskor az előírt helyre generálja a megadott sztringet és kijelöli a mező hosszát. Az első két mezőnél nem generál semmit, a születési időnél egy dátumot generál, íráskor a számok után álló pontokat átlépi.

A személyi szám mezőnél az első számjegyek "1"-et, a további hat számjegyek a születési időből kapott év, hó, nap karaktereket generálja. A mezőből csak helyes személyi számmal lehet kilépni.

Ha a "Rendben van?" kérdésre Nemmel válaszolunk, előlőről kezdődik az adatok bevitelére, de minden mezőben megjelenik az előzőleg bevitt adat. A hibátlan mezőket 'RETURN'-nel átléphetjük, a hibásat pedig javíthatjuk.

Itt van jelentősége a generálásnak és olyan jellegű adatbevitelnek, ahol sok hasonló adatot kell egymás után bevinni, ekkor az új tétel bevitelénél mindig generálhatjuk az előző tételt, és csak a változást kell beírni.

Az adatmezőben jobbra, balra a kurzor billentyűkkel lehet mozogni (a kurzor-le is balra mozgat), a 'HOME' az első karakterre viszi a kurzort, a 'DEL' törli a mezőt. A bevitt 'RETURN'-nel bármikor befejezhetjük, de ha végigírtuk a mezőt, nem kell 'RETURN'-özni.

A működésről röviden:

A rutin a 650–810-es sorokban található, meghívása 'GOSUB 650'-nel vagy 'GOSUB 660'-nal történhet. 650 esetén a kurzor aktuális helyén kezdődik az adatmező (pl. kérdés után), 660-nál meg kell adni a sor (s) és az oszlop (o) számát is.

Bemenő adatok:

h: adatmező hossza

i\$: generálandó sztring

nu=1: csak számjegyet és tizedespontot fogad el és valós számot ír ki

zk=1: csak számjegyet fogad el (120)

dt=1: ha dátumot kér be (100)

Kijövő adatok:

nr\$: a mezőbe bevitt sztring

n: a mezőbe bevitt valós szám

A rutin által használt további változók:

e\$, g\$, is\$, m\$, p\$, r\$, sp\$, vi\$

g, k, ok, q

A rutin még további három rutint használ:

GOSUB 550: a megadott sor, oszlop pozícióitól kiírja a\$-t (PRINT AT)

GOSUB 860: az n valós számból generálható nr\$ sztringet állít elő

GOSUB 1250: villogtatja a keretet (pl. tiltott billentyű lenyomásakor)

A működés lényege:

675: képernyőre viszi az adatmezőt, ha i\$ < > "", akkor azt is generálja, a 'kurzor' az első karakteren áll (fehér színű, ha van karakter)

690: karakter bevitelére (get g\$)

770: a bevitt karaktereket összefűzi (nr\$)

— ha kurzorral mozgunk, az átlépett karaktereket "hozzáadja", illetve "levonja" nr\$-ből

— RETURN-re az egész mező tartalma átmegy nr\$-ba

A rutin meghívására a 80–150 sorokban van példa.

A továbbiakban minden PLUS/4-re vonatkozik.

```

1 REM *****
2 REM *      C= UJSAG      SORSZAM:168 *
3 REM *      GEN-GET-INPUT'64 *
4 REM *      PROGRAM: BAGDACS LAJOS *
5 REM *****
20 GOSUB4000:PRINT#
30 DATA NEV,SZULETESI HELY,SZULETESI IDO,ANYJA NEVE,SZEMELYI SZAM,LAJCIM
35 POKE53280,6:POKE53281,12:PRINT"Q":S=8:O=6:GOSUB550:PRINT"GENERALT GET-ES INPU
T RUTIN"
40 S=10:O=2:GOSUB550:PRINT"ADATBEKERO DEMOVAL, ES SZEMELYI SZAM"
45 S=12:O=10:GOSUB550:PRINT"ELLENORZO RUTINNAL"
47 S=18:O=13:GOSUB550:PRINT"BAGDACS LAJOS"SPC(71)"1989"
48 S=23:O=30:GOSUB550:PRINT"   SPACE   "
50 GETV$:IFV$=""THEN S0
60 FOR I=1 TO 6:READ A$(I):NEXT I
70 PRINT"Q":GOSUB 10160
80 S=3:O=20:H=10:I#=B$(1):GOSUB660:B$(1)=NR#
90 S=6:O=20:H=10:I#=B$(2):GOSUB660:B$(2)=NR#
100 S=9:O=20:H=11:DT=1:I#=B$(3):GOSUB660:B$(3)=NR#
105 B$(7)=MID$(B$(3),3,2)+MID$(B$(3),6,2)+MID$(B$(3),9,2)
110 S=12:H=10:I#=B$(4):GOSUB660:B$(4)=NR#
115 B$(8)=LEFT$(B$(5),1):IFB$(5)=""THENB$(8)="1"
118 I#=B$(8)+B$(7)+RIGHT$(B$(5),4)
120 S=15:H=11:ZK=1:GOSUB660:B$(5)=NR#
130 IF B$(7)>MID$(B$(5),2,6) THENB$(5)=""GOTO100
140 GOSUB 10210:IF HI=1 THEN HI=0:I#=NR#:GOTO 120
150 S=18:O=11:H=20:I#=B$(6):GOSUB660:B$(6)=NR#
180 S=22:O=9:A$="RENDBEN VAN A(I/N) ?":GOSUB550
190 GETV$:IFV$=""THEN190
200 IF V$="N"THEN 70
210 IF V$<"1"THEN GOSUB2150:GOTO190
240 END
550 POKE214,S:POKE211,0:SYS58732:PRINTA$:A$="" :RETURN
648 :
649 REM "GEN-GET 0-POZ NELKUL"
650 S=PEEK(214):O=PEEK(211)+2:O=1
658 :
    
```



```

O 659 REM "GEN-GET 0-POZICIONALASSAL"
660 M$="" : A$=I$ : NR$=""
665 I$#="" : I$# = LEFT$(SP$, H) : A$ = "#####" + I$# + "###" : GOSUB550
O 670 R$ = "###" : FORK = 1 TO H : OK = .
675 A$ = NR$ + R$ + MID$(I$, K, 1) + P$ + MID$(I$, K + 1) + R$ + P$ : GOSUB550
680 IFDT = 1 AND OK = . THEN IFK = 50RK = 80RK = 11 THEN G$ = "###" : GOTO730
O 690 GETG$ : IFG$ = " " THEN 690
692 IFG$ = CHR$(34) THEN GOSUB2150 : GOTO690
695 G = ASC(G$)
700 IFG = 13 THEN M$ = MID$(I$, K) : K = H : GOTO730
O 720 IFDT = . AND G = 32 THEN G$ = " " : GOTO770
730 IFG$ = "##" THEN VI$ = G$ : G$ = MID$(I$, K, 1) : GOTO770
740 IFK > 1 THEN IFG$ = "##" OR G$ = "##" THEN G$ = " " : K = K - 1 : NR$ = LEFT$(NR$, K - 1) : OK = 1 : GOTO675
O 750 IFG = 19 THEN NR$ = " " : A$ = LEFT$(SP$, H) : GOSUB550 : GOTO660
755 IFDT = . AND G = 148 OR G = 28 THEN NR$ = " " : I$ = " " : A$ = LEFT$(SP$, H) : GOSUB550 : GOTO660
760 IFNU = 1 OR DT = 1 THEN IF(G$ < "0" OR G$ < "9") AND G$ < " " THEN GOSUB2150 : GOTO690
762 IFZK = 1 THEN IF(G$ < "0" OR G$ < "9") THEN GOSUB2150 : GOTO690
O 765 IFNU = . THEN IF G < 32 OR G > 127 AND G < 160 THEN GOSUB2150 : GOTO690
770 NR$ = NR$ + G$ : I$ = NR$ + MID$(I$, K + 1)
775 IFVI$ = "##" AND K > LEN(NR$ + M$) THEN K = K - 1 : VI$ = " "
O 780 NEXTK : NR$ = NR$ + M$ : N = VAL(NR$) : E$ = NR$
790 IFQ = 1 THEN A$ = " " : GOSUB550
792 IFNU = . THEN 800
O 795 A$ = I$ : GOSUB550 : GOSUB860
800 REM NR$ = LEFT$(NR$ + SP$, H)
802 REM : A 800-AS SOB RELATIV FILE-NAL KELL ('H' HOSSZIG 'SPACE'-SZEL FELTOLTVE)
805 PRINTP$ : A$ = LEFT$(NR$ + SP$, H) : GOSUB550 : G$ = . : I$ = " " : PRINT " " : I$# = " " : H = . : NU = .
O 810 ZK = . : DT = . : OK = . : POKE198, . : RETURN
820 NR$ = MID$(STR$(H), 2) : RETURN
2150 C0 = PEEK(C4) : FORI = 1 TO 32 : POKEC4, I : NEXTI : POKEC4, C0 : RETURN
O 4000 SP$ = " " : FORI = 1 TO 30 : SP$ = SP$ + " " : NEXTI : P$ = "##"
4010 C4 = 53280 : B$(3) = "1910.01.01." : RETURN
O 10160 REM *** KEPERNYO MASZK KIIRAS ***
10170 PRINT "3" : O = 2
10180 FOR S = 3 TO 18 STEP 3
10190 GOSUB 550 : PRINT A$(S/3)
O 10200 NEXT S : RETURN
10210 REM *SZEMELYISZAM SZAM VIZSGALO*
10220 HI = 0 : S(0) = 0
O 10230 S# = B$(S)
10240 IF LEN(S#) > 11 THEN 10450
10250 S1 = VAL(LEFT$(S#, 1))
10260 S2 = VAL(MID$(S#, 2, 2))
O 10270 S3 = VAL(MID$(S#, 4, 2))
10280 S4 = VAL(MID$(S#, 6, 2))
10290 S5 = VAL(RIGHT$(S#, 1))
O 10300 IF S1 < 1 OR S1 > 2 THEN 10450
10320 IF S3 > 12 OR S3 < 1 THEN 10450
10330 IF S4 > 31 OR S4 < 1 THEN 10450
10340 IF S4 > 30 AND (S3 = 4 OR S3 = 6 OR S3 = 9 OR S3 = 11) THEN 10450
O 10350 IF S4 > 29 AND S3 = 2 THEN 10450
10360 FOR I = 1 TO 10
10370 S(I) = VAL(MID$(S#, I, 1))
O 10380 S(0) = S(0) + S(I) * I
10390 NEXT I
10400 SX = S(0) / 11
O 10410 E = S(0) - SX * 11
10420 IF E = 10 THEN E = 0
10430 IF E > 55 THEN 10450
O 10440 RETURN
10450 HI = 1 : RETURN
O
READY.
    
```

## "HUNTBÓL SORSZÁM"

A C-16 (PLUS/4) TEDMON programja segítségével, ha MONITOR-ba lépünk, a H (hunt) parancsra a gép megkeresi a kívánt karakterek vagy byte-ok összes előfordulási helyét a tárban. Pontosabban megadja a kezdőcímekeket hexadecimális alakban.

Ez a rutin megmondja, hogy a hexa. cím, amit a H parancsra kaptunk, melyik BASIC sorban található. Ez főleg hosszú programoknál jelenthet segítséget. A rutint célszerű a program végéhez fűzni, így GOTO 9000-rel bármikor használhatjuk.

Használata:

GOTO 9000 vagy önállóan RUN hatására programozza, majd kiírja a funkcióbillentyűk tartalmát és leáll a 9040-es sorban.

— 'F 1'-re MONITOR-ba lépünk, kiírja a >2b sort és felírja a HUNT parancsot. Itt a keresendő terület határait a 2b—2e címeknek megfelelően (fordított sorrendben) átírhatjuk pl. ha csak a rutin van a tárban: H 1001 1218'

Ha karaktereket, sorszámot vagy változót keresünk, azt ' (apoztrof=SHIFT+7) után kell írni. A BASIC alapszavakat hexa alakban kell megadni aposztrof nélkül.

Egy trükk: keressük az SCNCLR-t, de nem tudjuk a hexa. alakját. Írjuk be: O SCNCLR (RETURN), 'F 1'-el menjünk át MONITOR-ba és írjuk be: >1005 (RETURN), ezt látjuk: 1005 e8 00. . . , az e8 az SCNCLR hexa. alakja a tárban.

Most már írhatjuk: h 1001 1218 e8 (RETURN), a keresés eredménye: 1005 10a1

— 'F 2'-vel visszatérünk BASIC-be és ablakot készítünk.

— 'F 3'-mal indul a sorszámkeresés, a hexa. címet úgy kell megadni, ahogy a H után kaptuk, pl. 1005. A keresés eredménye: LIST 0 (ezt most írtuk be).

— 'F 3'-mal folytathatjuk a keresést, \$10a1-re LIST 9040-et kapunk.

FIGYELEM! Keresés közben ne változtassunk a BASIC sorok hosszán, mert eltolódnak a címek a tárban.

— 'F 4' törli az ablakot és a képernyőt a keresés végén.

A rutin működése:

A 9040 sor a funkcióbillentyűket programozza, a 9060—9075 sorok vizsgálják és jelzik, ha a cím a BASIC területén kívül esik. (Ekkor vad sorszámokat kapnánk.) A 9080-as sor a megadott címtől visszafelé haladva '00' byte-ot keres, ami a BASIC sor végét jelzi, ha az utána következő két byte által mutatott cím





előtt is '00' byte áll (9090). A sorszámot az igazi '00' byte utáni 3. és 4. byte adja (9090). Hamis '00' byte-okat adnak pl. a 257-nél kisebb BASIC sorszámok.

Ha a BASIC terület felső határa \$8000 felett van, ki kell adni egy POKE 2040,128 parancsot, ekkor MONITOR-ból elérjük a \$8000 feletti RAM-ot.

```

0 9000 REM "HUNTBOL SORSZAM"
0 9010 KEY1,"MI"+CHR$(13)+CHR$(20)+CHR$(13)+"MH 1001 4000 ^"
0 9020 KEY2,"X"+CHR$(13)+CHR$(27)+CHR$(13)
0 9030 KEY3,CHR$(19)+CHR$(13)+CHR$(9050)+CHR$(13):KEY8,"X"
0 9040 SCNCLR:KEY:PRINT"MI F1, F2, F3, HELP":STOP
0 9050 POKE19,1:INPUT"MI KEREM A HEXA CIMET: #":A#:PRINT
0 9060 POKE19,1:I=DEC(A#):AC=256*PEEK(44)+PEEK(43)+1:FC=256*PEEK(46)+PEEK(45)-4
0 9070 IFI>FCORICACTHENPRINT"MI * CIM: # "HEX$(AC)" ES # "HEX$(FC)" KÖZÖTT LEHET
1 "
0 9075 IFI>FCTHENPRINT"MI * * SPACE * ":GETKEYV#:GOTO9050
0 9080 FORJ=ITDI-300STEP-1:P=PEEK(J):IFPTHENNEXTJ
0 9090 CI=256*PEEK(J+2)+PEEK(J+1):IFPEEK(CI-1)THENNEXTJ
0 9100 SZ=256*PEEK(J+4)+PEEK(J+3):PRINT"MI LIST"SZ
0 9110 PRINT"MI LIST"SZ:TTI"
0 9120 A#="" :END

0 READY.

```

## "SZÍNBEÁLLÍTÁS"

A menü '1' pontja inkább színbemutató vagy színtervező lehetne. Itt a gép bekapcsolási színével kezdve minden szín és árnyalat beállítható, és ugyanakkor leolvasható a hozzátartozó COLOR illetve POKE számérték.

A POKE 1339 után álló számértéket kell adni a színtár POKE 2048—3047,x formájú színmegadás 'x' értékének (villogtatva x+128-at).

A '2' pont egy rutin, mely programon belüli színbeállításra alkalmas. A 22. sor adja az induló szint és GOSUB 6000-rel hívható a rutin, ami a 6170-es sorig tart. A beállított színértékeket a 220 és 224 tárcímen tárolja, így újabb RUN esetén is a beállított színnel indul a program.

```

0 1 REM *****
0 2 REM * C= UJSAG SORSZAM:169 *
0 3 REM * SZINBEALLITO PLUS/4 *
0 4 REM * PROGRAM: BAGDACS LAJOS *
0 5 REM *****
15 REM 1001 1990
20 PRINTCHR$(14):VOL8
0 22 IFPEEK(224)=. THEN COLOR4,1:COLOR.,2,3:POKE224,PEEK(65305):POKE220,PEEK(65301)
0 25 SCNCLR:PRINTSPC(12)"MI SZINBEALLITAS"
0 26 PRINTSPC(13)" "
0 28 POKE1339,1:PRINTSPC(12)"MI BAGDACS LAJOS"SPC(71)"1989":POKE205,10:PRINT
0 30 PRINTTAB(6)"MI 1 MI MINDEN SZIN BEALLITHATO"
0 40 PRINTTAB(6)"MI 2 MI PROGRAMON BELUL AZ ADOTT"
0 45 PRINTTAB(11)"MIKARAKTERSZINEKHEZ KERESHETO"SPC(53)"A KERET ES A HATTER SZINE"
0 50 PRINTTAB(6)"MI 3 MI VESE"
0 60 GETKEYV#:IFV#<"1"ORV#>"3"THENSOUND3,999,10:GOTO60
0 70 ONVAL(V#)GOSUB 80,90,100
0 80 GOSUB7000
0 82 PRINT"MI A POKE 2048-3047,X SZINRAM X ERTEKENEK MI IS A POKE 1339 UTAN ALLO";
0 84 PRINT"ERTEKET KELL MI ADNI.":GETKEYV#:GOTO25
0 90 GOSUB6000:GOTO25
100 END
6000 REM SZINBEALLITO SZUBRUTIN PRG-BAN
0 6020 SCNCLR:PRINTSPC(11)"MI SZINBEALLITAS":PRINT"MI A PROGRAMBAN A KEPERNYON"
;
0 6030 PRINT" LATHATO MIKARAKTERSZINEK FORDULNAK ELO,"
0 6040 PRINT"MI MOST BEALLITHATOD KEDVENC SZINEIDET:"
0 6050 POKE1339,1:PRINT"MI 1 MI KERET SZIN MI 2 MI KERET TONUS"
0 6060 PRINT"MI 3 MI HATTER SZIN MI 4 MI HATTER TONUS"
0 6070 PRINT"MI 5 MI EREDETI SZIN MI 6 MI KESZ"
0 6080 I=1:J=.:K=2:T=3:PRINT"MI"
0 6090 POKE239,1:GETKEYV#1#:G=VAL(G1#):IFG<10R>6THENSOUND3,912,10:GOTO6090
0 6100 IFG=6THENPOKE220,PEEK(65301):POKE224,PEEK(65305):RETURN
0 6110 IFG=5THENPOKE65301,PEEK(220):POKE65305,PEEK(224):GOTO6090
0 6120 IFG=1THENI=I+1:IFI=17THENI=1
0 6130 IFG=2THENJ=J+1:IFJ=9THENJ=
0 6140 IFG=3THENK=K+1:IFK=17THENK=1
0 6150 IFG=4THENT=T+1:IFT=8THENT=.
0 6160 COLOR4,I,J
0 6170 COLOR.,K,T:GOTO6090
0 7000 I=15:J=6:K=2:T=7:M=1:N=1:REM BEKAPCSOLASI SZINEK
0 7090 POKE239,1:GOSUB8010:GETKEYV#1#:G=VAL(G1#):IFG<10R>8THENSOUND3,912,10:GOTO70
90
0 7100 IFG=7THEN7000
0 7105 IFG=8THENRETURN
0 7120 IFG=1THENI=I+1:IFI=17THENI=1
0 7130 IFG=2THENJ=J+1:IFJ=9THENJ=
0 7140 IFG=3THENK=K+1:IFK=17THENK=1
0 7150 IFG=4THENT=T+1:IFT=8THENT=.
0 7160 IFG=5THENM=M+1:IFM=17THENM=1
0 7170 IFG=6THENN=N+1:IFN=9THENN=.
0 7190 GOTO7090

```



```

7900 RETURN
8000 PRINT "
8010 COLOR 4, I, J: COLOR 0, K, T: COLOR 1, M, N: SCNCLR: PRINT TAB(19) "MSZINBEALLITAS"
8015 GOSUB 9000: PRINT " 1 KERET SZIN "POKE(65305), "PEEK(65305): GOSUB 9000
8020 PRINT " 2 KERET TONUS "POKE(65305), "PEEK(65305): GOSUB 9000
8030 PRINT " 3 HATTER SZIN "POKE(65301), "PEEK(65301): GOSUB 9000
8040 PRINT " 4 HATTER TONUS "POKE(65301), "PEEK(65301): GOSUB 9000
8050 PRINT " 5 BETU SZIN "POKE(1339), "PEEK(1339): GOSUB 9000
8060 PRINT " 6 BETU TONUS "POKE(1339), "PEEK(1339): GOSUB 9000
8070 PRINT " 7 BEKAPCSOLASI SZIN "SPC(56) " 8 KESZ " "POKE(1339), "PEEK(1339): GOSUB 9000
8080 PRINT " FICYELD A SZAMERTEKEK VALTOZASAT " "POKE(1339), "PEEK(1339): GOSUB 9000
    
```

READY.

### Még néhány apróság PLUS/4-re:

— POKE 1351,128-cal letilthatjuk a SHIFT+C billentyűket, vagyis a nagybetű—kisbetű váltást. Ez hasznos lehet pl. ékezetes karakterkészlet használatánál.

— a funkcióbillentyűk tartalmát MONITOR-ból elmenthetjük S"key",1,55f,5e7 paranccsal, és 'L' paranccsal akkor is vissza tudjuk tölteni, ha program van a tárban. Ha LOAD"key",1,1 paranccsal töltjük be, úgy töltés után a NEW parancsot is ki kell adni.

— memóriatarakékossgát jelent, ha hosszú BASIC sorokat írunk, főleg ha pl. TURBOBASIC-kel hatsorosakat. Van egy egyszerű trükk a rövid sorok összefűzésére.

Példa: 10 print, „ez egy rövid sor”  
20 print, „ez is rövid...”

a) LIST 20 (RETURN)

b) töröljük le a 20-as számot, írjunk a PRINT elé kettőspontot és húzzuk balra 'DEL'-lel a képernyő széléig a kettőspontot

c) a LIST 20 feliratot javítsuk LIST 10-re

d) nyomjuk meg az 'ESC', majd az 'A' billentyűket, aztán a 'RETURN'-t

e) vigyük a kurzort az egybeírt sorra és 'RETURN'

f) töröljük a 20-as sort és 'ESC'+ 'C'-vel a beszúrásos üzemmódot (leírni hosszabb, mint megcsinálni).

Mostanában sokat dolgoztam hosszú ékezetes (f000—f700) programmal TURBOBASIC alatt. A sok jó mellett voltak kellemtelenségeim is, aminek nem tudom az okát:

— mindig ,7-tel mentettem, és gyakori volt a VERIFY ERROR, ami többszöri próbálkozás, esetleg magnófejlesztítés (naponta többször!) után OK lett;

— olykor-olykor a legkülönbözőbb helyeken szemes lett a program, először a DELETE parancsra gyanakodtam, de ha nem használtam, akkor is előfordult;

— egyszer úgy futottam ki a memóriából, hogy az új program-sor nem fért el (a változók is éltek), és lemerevedett a gép, ki kellett kapcsolni.

Ha a fentiekre tudnak magyarázatot, vagy netán van egy újabb TURBOBASIC, szíveskedjenek értesíteni. Az én TURBOBASIC-em a C=-ben megjelent javított változat a PÖTYÖGŐ-SZOLGÁLAT-tól.

Bagdács Lajos

## Nagyító C+4

Várkonyi István tatabányai tagtársunk átírta C+4-re a C-újság 1987/10-es számában megjelent NAGYÍTÓ programot. A használatáról ott leírtakat nem kívánjuk megismételni.

Két megjegyzés a C+4-es változathoz:

1. A program magnóról csak a normál módon kimentett képeket tudja beolvasni. Ezért a TEDPAINT-tal készült vagy TURBO-val kimentett képeket nem tudja beolvasni.

2. A program lemezről be tudja hívni a BOTTICELLI-vel, a TEDPAINT-tal, továbbá a C64-es rajzolóprogramokkal készült képeket is, de ebben az esetben a 8k-nyi bittérképes területhez hozzá kell fűzni még 2k-nyi területet. Az összefűzést úgy kell megcsinálni, mintha a kép C16-os vagy C+4-es gépen készült volna (először a 2k-nyi akármilyen 8k-nyi képinformáció). Ezt azért lehet megtenni, mivel a program csak arra használja a színekre és a fényességre vonatkozó adatokat, hogy a képpontok a \$2000—\$3FFF közötti helyre kerüljenek a tárban.

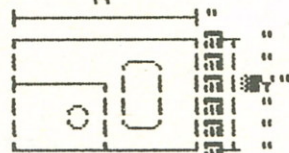
```

0 REM *****
1 REM * C= UJSAG SORSZAM:170 *
2 REM * NAGYITO C+4 *
3 REM * PROGRAM: VARKONYI ISTVAN *
4 REM * (NAGY BALINT UTAN) *
5 REM *****
10 IF XC=0 THEN 50
20 SYSDEC("D63C")
30 X=1:PRINT "KIS TURELMET..."
40 GOSUB 1100
50 COLOR 0,1:COLOR 4,1
60 PS=0
70 HS=0
80 PRINT " ";
90 X=10:Y=5:GOSUB 940:PRINT " "
100 X=10:Y=6:GOSUB 940:PRINT "1. KÉP BETÖLTÉS"
110 X=10:Y=7:GOSUB 940:PRINT " "
120 X=10:Y=8:GOSUB 940:PRINT "2. KÉP NYOMTATÁS"
130 X=10:Y=9:GOSUB 940:PRINT " "
140 X=10:Y=10:GOSUB 940:PRINT "3. KÉP BEKAPCSOLÁS"
150 X=10:Y=12:GOSUB 940:PRINT "4. DIRECTORY"
160 X=10:Y=13:GOSUB 940:PRINT " "
170 X=10:Y=14:GOSUB 940:PRINT "5. KILÉPÉS A PROGRAMBÓL"
180 X=14:Y=16:GOSUB 940:PRINT " "
190 X=14:Y=17:GOSUB 940:PRINT "VÁLASSZ !"
200 GETKEY A#
210 IF A#="1" THEN 740:REM KEP BETOLTES
220 IF A#="2" THEN 340:REM KEP NYOMTATAS
    
```



```

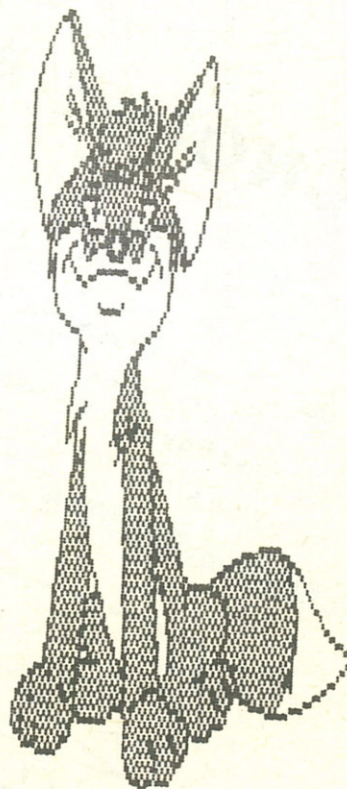
0 230 IF A$="3" THEN 960:REM KEP BEKAPCSOLAS
0 240 IF A$="4" THEN 270:REM DIRECTORY
0 250 IF A$="5" THEN END:REM KILEPES A PROGRAMBOL
0 260 REM *****
0 270 REM *** DIRECTORY ***
0 280 REM *****
0 290 PRINT "J"
0 300 SYSDEC("11D0")
0 310 GETKEY A$
0 320 GOTO 50
0 330 REM *****
0 340 REM *** KEP NYOMTATAS ***
0 350 REM *****
0 360 PRINT "J"CHR$(14+128);
0 370 PRINT "X"
0 380 PRINT " "
0 390 PRINT " "
0 400 PRINT " "
0 410 PRINT " "
0 420 PRINT " "
0 430 PRINT " "
0 440 PRINT " "
0 450 X=20:Y=0:GOSUB 940:PRINT " "
0 460 X=20:Y=1:GOSUB 940:PRINT "MEKKORA NAGYITÁST"
0 470 X=20:Y=2:GOSUB 940:PRINT "AKARSZ ?"
0 480 X=20:Y=4:GOSUB 940:PRINT "X=? 1":REM NX
0 490 X=20:Y=5:GOSUB 940:PRINT "Y=? 1":REM NY
0 500 X=1:Y=9:GOSUB 940:PRINT " "
0 510 X=1:Y=10:GOSUB 940:PRINT "VIGYÁZZ ! 2 X 2 -NÉL NAGYOBB NAGYÍTÁST"
0 520 X=3:Y=11:GOSUB 940:PRINT " "
0 530 X=3:Y=12:GOSUB 940:PRINT "CSAK SZALAGPAPÍRRA TUDOK CSINÁLNI !"
0 540 X=0:Y=13:GOSUB 940:PRINT "-----"
0 550 X=0:Y=14:GOSUB 940:PRINT " "
0 560 X=0:Y=15:GOSUB 940:PRINT "HANY PONT LEGYEN EGY SORBAN ?"
0 570 X=3:Y=16:GOSUB 940:PRINT " "
0 580 X=3:Y=17:GOSUB 940:PRINT "(GÉPPAPÍR-SZÉLESSÉG KB. 400 PONT.)"
0 590 X=3:Y=19:GOSUB 940:PRINT "MENNYI LEGYEN ? 400":REM SZ
0 600 X=22:Y=4:GOSUB 940:INPUT " ";NX
0 610 X=22:Y=5:GOSUB 940:INPUT NY
0 620 X=17:Y=19:GOSUB 940:INPUT SZ
0 630 POKEDEC("12EA"),NY:POKEDEC("12EB"),NX
0 640 S=SZ/NY:IF S>200 THEN S=200
0 650 POKEDEC("12EC"),S
0 660 X=0:Y=21:GOSUB 940:PRINT " "
0 670 X=0:Y=22:GOSUB 940:PRINT "ALLITSD BE A NYOMTATÓT ÉS NYOMJ [SPACET]!"
0 680 GETKEY A$:IF A$<>" " THEN 680
0 690 GOSUB 1020
0 700 SYSDEC("1070"):REM NAGYITAS
0 710 GOSUB 1070
0 720 GOTO 50
0 730 REM *****
0 740 REM *** KEP BETOLTES ***
0 750 REM *****
0 760 X=5:Y=21:GOSUB 940:INPUT "MELYIK RAJZOT ";N$
0 770 N$=LEFT$(N$,16):IF LEN(N$)=0 THEN 50
0 780 FOR I=1 TO LEN(N$):POKEDEC("12D8")+I,ASC(MID$(N$,I,1)):NEXT
0 790 X=9:Y=23:GOSUB 940:PRINT "[LOPPY VAGY [KAZETTA ?"
0 800 GETKEY A$
0 810 IF A$="K" THEN EG=1:GOTO 860
0 820 IF A$="F" THEN EG=0:GOTO 840
0 830 GOTO 800
0 840 GOSUB 1020
0 850 GOSUB 1040
0 860 POKEDEC("D8"),LEN(N$):POKEDEC("D9"),EG
0 870 SYSDEC("11B0"):REM KEP BETOLTES
0 880 IF EG=1 THEN 960
0 890 GOSUB 1070
0 900 GOTO 50
0 910 REM *****
0 920 REM *** KURZOR POZICIONALAS ***
0 930 REM *****
0 940 POKE202,X:POKE205,Y:SYS55464:RETURN
0 950 REM *****
0 960 REM *** KEP BEKAPCSOLAS ***
0 970 REM *****
0 980 GOSUB 1020
0 990 GETKEY A$
1000 GOSUB 1070
1010 GOTO 50
1020 SYSDEC("1270"):REM BITTERKEPES UZEMMOD BEKAPCSOLASA
1030 RETURN
1040 POKEDEC("D8"),PS:POKEDEC("D9"),HS
1050 SYSDEC("1298"):REM SZINMEMORIA TORLESE
1060 RETURN
    
```





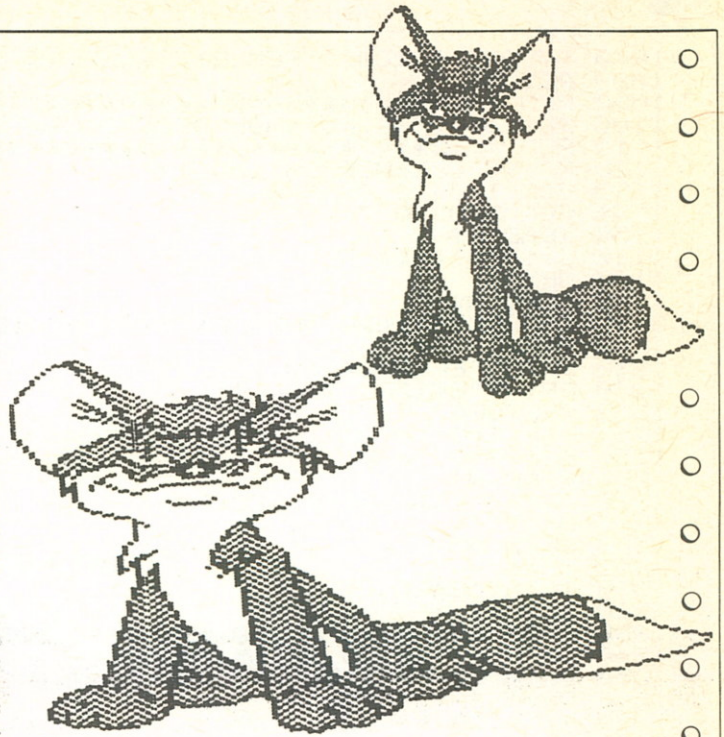
```

0 1070 SYSDEC("12C0"):REM BITTERKEPES UZEMMOD KIKAPCSOLASA
0 1080 RETURN
0 1090 REM *****
0 1100 REM *** ASS.RUTINOK BEOLVASASA ***
0 1110 REM *****
0 1120 PRINT"ELLENORZES INDUL!"
0 1130 FOR I=0 TO 88:C=0
0 1140 FOR B=0 TO 7
0 1150 READ A#
0 1160 C=C+DEC(A#)
0 1170 NEXT
0 1180 READ A#
0 1190 IF DEC(A#)<>C THEN PRINT"ADATHIBA A"I". SORBAN !!!":END
0 1200 NEXT
0 1210 PRINT"ELLENORZES KESZ!"
0 1220 PRINT"BEIRAS INDUL!"
0 1230 RESTORE
0 1240 FOR I=0 TO 88
0 1250 FOR B=0 TO 7
0 1260 READ A#
0 1270 POKE 4112+I*8+B,DEC(A#)
0 1280 NEXT
0 1290 READ A#
0 1300 NEXT
0 1310 PRINT"OK!"
0 1320 RETURN
0 1330 DATA AD,F8,12,29,F8,85,D8,AD,04E2
0 1340 DATA F7,12,29,07,05,D8,85,D8,0373
0 1350 DATA AD,F7,12,29,F8,48,0A,0A,0333
0 1360 DATA 0A,18,65,D8,85,D8,A9,20,0385
0 1370 DATA 6D,F9,12,85,D9,68,4A,4A,03D2
0 1380 DATA 4A,48,65,D9,85,D9,68,4A,03E0
0 1390 DATA 4A,18,65,D9,85,D9,AD,F8,04A3
0 1400 DATA 12,29,07,AA,BD,5B,10,A0,02B4
0 1410 DATA 00,31,D8,D0,03,A9,00,60,02E5
0 1420 DATA 4C,51,12,80,40,20,10,08,01A7
0 1430 DATA 04,02,01,01,02,04,08,10,0026
0 1440 DATA 20,40,80,00,00,00,00,00,00E0
0 1450 DATA A9,30,8D,E9,12,A9,00,8D,0397
0 1460 DATA F5,12,8D,F6,12,8D,EE,12,0429
0 1470 DATA 8D,EF,12,A9,C7,8D,F4,12,0491
0 1480 DATA 8D,F1,12,A9,80,8D,ED,12,0445
0 1490 DATA A2,04,A9,04,A0,00,20,BA,02CD
0 1500 DATA FF,A9,00,20,8D,FF,20,C0,0464
0 1510 DATA FF,A2,04,20,C9,FF,A9,0F,0445
0 1520 DATA 20,D2,FF,AD,E9,12,20,D2,0498
0 1530 DATA FF,A9,0D,20,D2,FF,A9,08,0457
0 1540 DATA 20,D2,FF,A9,0D,20,D2,FF,0498
0 1550 DATA A9,00,8D,F8,12,A9,00,8D,0376
0 1560 DATA F9,12,AD,F1,12,8D,F4,12,044E
0 1570 DATA 8D,F7,12,38,ED,EC,12,80,0469
0 1580 DATA 02,A9,FF,8D,F1,12,AD,EF,04D6
0 1590 DATA 12,8D,F0,12,20,10,10,F0,02D1
0 1600 DATA 06,AE,EE,12,8D,63,10,0D,02F1
0 1610 DATA ED,12,8D,ED,12,EE,F0,12,0478
0 1620 DATA AD,F0,12,CD,EB,12,D0,12,0458
0 1630 DATA EE,F8,12,AD,F8,12,C9,00,0478
0 1640 DATA D0,03,EE,F9,12,A9,00,8D,0482
0 1650 DATA F0,12,EE,EE,12,AD,EE,12,0490
0 1660 DATA C9,07,D0,C8,A9,00,8D,EE,048C
0 1670 DATA 12,AE,EA,12,AD,ED,12,20,0388
0 1680 DATA D2,FF,CA,D0,F7,A9,80,8D,0618
0 1690 DATA ED,12,CE,F7,12,AD,F7,12,048C
0 1700 DATA CD,F1,12,F0,15,AD,F5,12,0489
0 1710 DATA 8D,F8,12,AD,F6,12,8D,F9,04D2
0 1720 DATA 12,AD,EF,12,8D,F0,12,4C,0398
0 1730 DATA E4,10,AD,F8,12,8D,F5,12,043F
0 1740 DATA AD,F9,12,8D,F6,12,AD,F0,04EA
0 1750 DATA 12,8D,EF,12,AD,F4,12,8D,03E0
0 1760 DATA F7,12,A9,0D,20,D2,FF,AD,045D
0 1770 DATA F9,12,C9,01,D0,2F,A9,3F,038C
0 1780 DATA CD,F8,12,80,28,A9,0D,20,0385
0 1790 DATA D2,FF,AD,F1,12,C9,FF,F0,0639
0 1800 DATA 14,EE,E9,12,A9,00,8D,F9,042C
0 1810 DATA 12,8D,F8,12,8D,F5,12,8D,03CA
0 1820 DATA F6,12,4C,A6,10,A9,04,20,02D7
0 1830 DATA C3,FF,4C,CC,FF,4C,E4,10,0519
0 1840 DATA 00,00,00,00,00,00,00,00,0000
0 1850 DATA A6,D9,A5,D9,A0,00,20,BA,0477
0 1860 DATA FF,A5,D8,A2,D9,A0,12,20,04C9
0 1870 DATA 8D,FF,A9,00,A2,00,A0,18,03BF
0 1880 DATA 20,D5,FF,60,00,00,00,00,0254
0 1890 DATA A9,08,20,81,FF,A9,F0,20,043A
0 1900 DATA 93,FF,A9,24,20,A8,FF,20,0446
    
```





○ 1910 DATA AE,FF,A9,08,20,B4,FF,A9,04DA  
 1920 DATA 60,20,96,FF,A9,00,85,90,03D3  
 1930 DATA A0,03,84,AB,20,A5,FF,85,041B  
 ○ 1940 DATA DC,20,A5,FF,85,DD,A4,90,0536  
 1950 DATA D0,13,A4,AB,88,D0,EB,A6,051B  
 1960 DATA DC,A5,DD,20,5F,A4,A9,20,044A  
 ○ 1970 DATA 20,D2,FF,A9,00,D0,13,20,039D  
 1980 DATA A5,FF,F0,05,20,D2,FF,D0,055A  
 1990 DATA F6,A9,0D,20,D2,FF,A0,02,043F  
 ○ 2000 DATA D0,C8,20,AB,FF,A9,08,20,0433  
 2010 DATA B1,FF,A9,E0,20,93,FF,20,050B  
 ○ 2020 DATA AE,FF,20,AB,FF,A5,AE,20,04EA  
 2030 DATA B1,FF,A9,E0,20,93,FF,20,050B  
 ○ 2040 DATA AE,FF,A5,C6,C9,40,F0,FA,060B  
 2050 DATA 60,AD,F9,12,C9,00,F0,0A,03DB  
 ○ 2060 DATA AD,F8,12,C9,40,90,03,A9,03FC  
 2070 DATA 00,60,A9,FF,60,00,00,00,0268  
 ○ 2080 DATA 00,00,00,00,00,00,00,00,0000  
 2090 DATA AD,06,FF,09,20,8D,06,FF,036D  
 ○ 2100 DATA AD,12,FF,29,FB,09,08,8D,038D  
 2110 DATA 12,FF,AD,14,FF,85,DE,A9,04DD  
 ○ 2120 DATA 18,8D,14,FF,60,00,00,00,0218  
 2130 DATA 00,00,00,00,00,00,00,00,0000  
 ○ 2140 DATA A5,D9,8D,19,FF,85,DA,A5,0527  
 2150 DATA D8,0A,0A,0A,0A,05,DA,A2,0281  
 ○ 2160 DATA 00,9D,00,1C,9D,00,1D,9D,0210  
 2170 DATA 00,1E,9D,00,1F,CA,D0,F1,0365  
 ○ 2180 DATA 60,00,00,00,00,00,00,00,0060  
 2190 DATA AD,06,FF,29,1F,8D,06,FF,038C  
 ○ 2200 DATA AD,12,FF,29,F7,09,04,8D,0378  
 2210 DATA 12,FF,A5,DE,8D,14,FF,60,0494



# ECONOMIX-AKCIÓ

10% kedvezmény

A Marx Károly  
 Közgazdaságtudományi Egyetemen működő  
**ECONOMIX**  
 Közgazdász Egyetemi Kiszövetkezet  
 az alábbi szoftvereket forgalmazza:



**Inveszt-PC+**  
 (Szoftver '88 oklevéllel díjazott!)  
 Beruházások pénzügyi kezelése és  
 nyilvántartása

**ECONOMIX-ÁFA**

Általános forgalmiadó és -számítás  
 és vevő-szállító-nyilvántartás

**ECONOMIX-VARITERV**

Éves és középtávú gazdasági gördülőtervezés  
**ECONOMIX-SZJA**  
 Személyi jövedelemadó-számítás  
 és -nyilvántartás

**MAD**

Több tényezős döntéselőkészítő rendszer  
**FŐKÖNYVI RENDSZEREK**  
 széles választékban

Megrendelhető jelen hirdetés  
 egyidejű megküldésével.  
 1053 Budapest, Veres Pálné u. 36.  
 Tel.: 182-433, 174-188/17



Az

# AMIGA

operációs rendszere

(II. rész)



Az első részben a Workbench-ről és röviden a CLI-ről volt szó. Most a CLI részletesebb taglalása következik.

Minden Amigához jár legalább 3 lemez. A Workbench, az Extras és egy oktatólemez. A CLI szempontjából a Workbench-lemez — a rendszerlemez — a legérdekesebb. Mit tartalmaz ez a lemez? Ezt az AmigaDOS 'List' vagy 'Dir' parancsával lehet megnézni. A lemez főkatalógusa többek között tartalmazza a következő alkatalógusokat: 'S', 'C', 'L', 'Libs', 'Devs', 'Fonts', 'T', 'System', 'Utilities', 'Expansion', 'Empty'. A mostani tárgyalás szempontjából az első 5 a fontos.

Az 'S' alkatalógusban lévő 'Startup-Sequence' nevű batch-vagy szövegfile, funkcióját tekintve megegyezik az MS-DOS 'Autoexec.bat' állományával, tehát rendszertöltéskor az abban lévő DOS parancsok hajtódnak végre. Ez tetszőlegesen módosítható, így mindenki ízlése szerint állíthatja össze saját boot-programját.

A 'C'-ben található meg az összes AmigaDOS parancs (mint az első részben szerepelt, az 1.2-es AmigaDOS-ban minden parancs külső parancs). Itt rögtön felvetődik a kérdés, hogyan lehet az, hogy soha nem kell elérési utat megadni, ha egy AmigaDOS parancsot akarunk használni? (Pl. a 'Devs/printers' alkatalógusban vagyunk és szeretnénk megnézni a tartalomjegyzéket. Nem kell a 'DF0:C/List' vagy ':C/List' vagy 'C:List' vagy 'SYS:C/List' vagy a '//C/List' alakot használni, egyszerűen elég csak azt begépelni, hogy 'List'.) Ez úgy lehetséges, hogy a 'Path' parancsnak nem az az alapértelmezése, hogy nincs elérési út, hanem — mint ez a 'Path show' vagy paraméter nélküli 'Path' parancssal ellenőrizhető — 'Current directory', majd 'C:'. Tehát a beírt parancsot a gép egy program nevének tekinti, amelyet először az aktuális katalógusban, majd — ha ott nem találta — a 'C:' logikai katalógushoz rendelt fizikai alkatalógusban keresi. (Az alapértelmezésnél kevesebb elérési út nem adható meg.)



Az 'L' az operációs rendszer különféle kezelő- és segédprogramjait tartalmazza. Itt található pl. a 'Ram-handler', a RAM DISK kezelőprogramja, a Port-Handler stb. Ezek a programok a CLI-ből nem futtathatók — nem is arra valók —, ezeket igény esetén a rendszer tölti be automatikusan.

A rendszer tranzienis bináris könyvtárai a 'Libs' alkatalógusban vannak. Ilyenek pl. a matematikai könyvtárak, amelyek — többek között — a lebegőpontos rutinokat tartalmazzák. Ha egy felhasználói program ezeket a rutinokat akarja használni, a rendszer — ha előzőleg nem volt betöltve — betölti a megfelelő rutint tartalmazó könyvtárat és a hívó program rendelkezésére bocsátja. A betöltés mindig arról a lemezzel történik, amelyről a rendszert hívtuk.

A 'Devs' alkatalógusban különböző egységek meghajtóprogramjai vannak. Pl.: 'parallel.device' a párhuzamos port kezelésére, 'serial.device' a soros port kezelésére, 'printer.device' a nyomtató meghajtója stb. Mivel az Amigának szabványos soros és párhuzamos interfésze van, ezért nagyon sokféle nyomtató kapcsolható hozzá. A különböző nyomtatók — ha tudnak is néhány szabványosnak tekinthető más nyomtatót emulálni — éppen eléggé eltérnek egymástól ahhoz, hogy teljesen más nyomtatási képük legyen. Az Amiga operációs rendszerében van egy lehetőség, amely segíthet ezen a problémán. Éppen a 'Devs'-ből nyíló 'printers' alkatalógusban a rendszer által ismert nyomtatók meghajtóprogramjai találhatóak. A 'Preferences' segédprogrammal ezek közül lehet kiválasztani a kívántat. — A 'Preferences' egy általános konfiguráló program, amellyel többek között be lehet állítani a Workbench képernyő színeit/helyzetét, az egér által mozgatott mutató színeit/alakját, a soros csatorna paramétereit, a nyomtató típusát, a nyomtatás paramétereit, a billentyűzet és az egér paramétereit. A 'Preferences' a kiválasztott opciókat a 'Devs/system-configuration' fileba menti ki. — A 'Olvs'-ből egy másik lényeges alkatalógus is nyílik, a 'keymaps'. Mint ahogy a nyomtatók is különbözhetnek egymástól, úgy az egyes országok billentyűzetszabványai is eltérőek. A 'keymaps'-ban a nyugat-európai országok és az

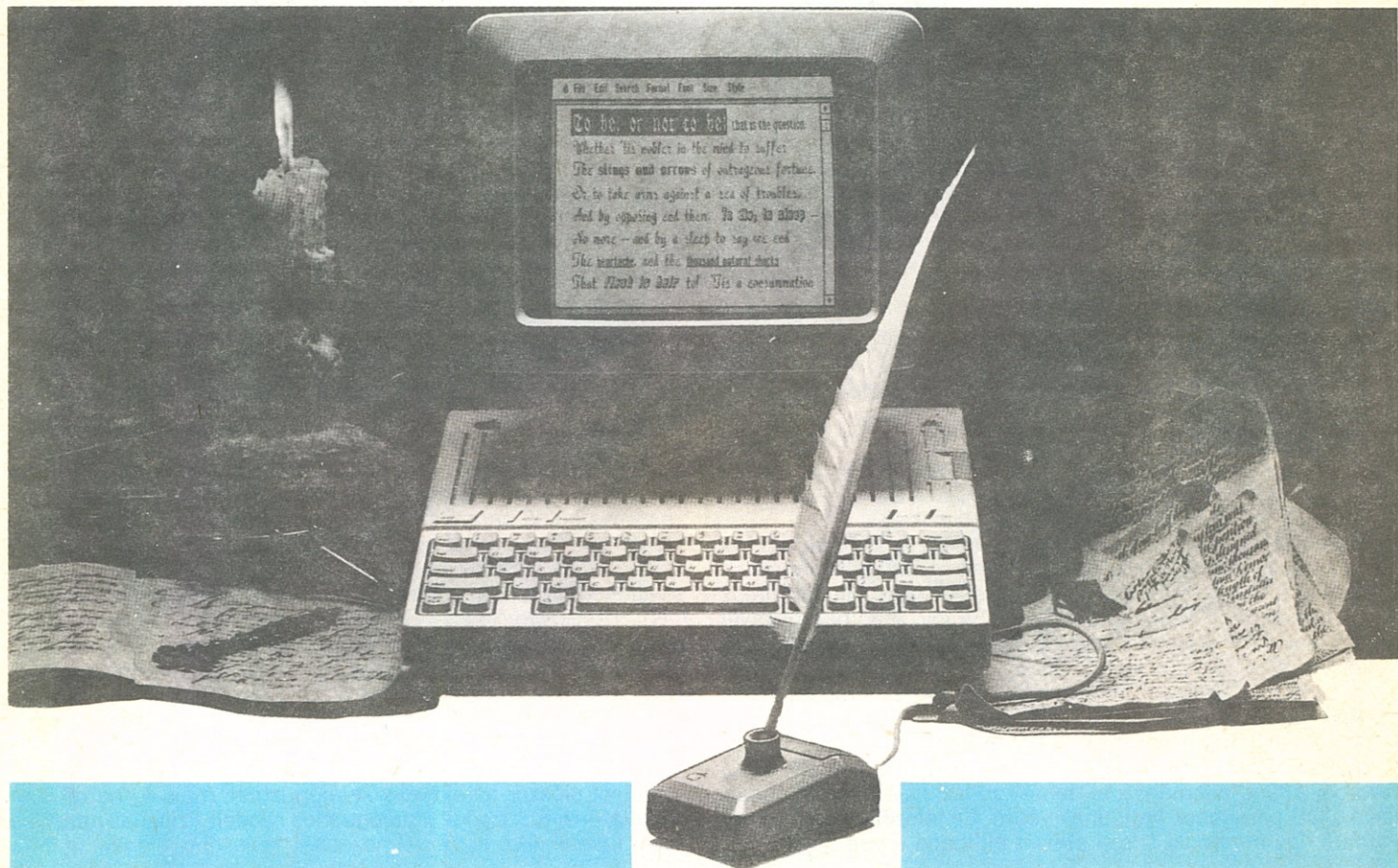
USA szabványai szerinti billentyűzet dekódolási táblázatok találhatóak. Ezek a 'SetMap' programmal aktivizálhatók. (A 'SetMap' program a 'System' alkatalógusban található.) Ha pl. a német szabvány szerint szeretnénk használni a billentyűzetet — és a főkatalógusban vagyunk —, elég csak annyit beírni, hogy 'System/SetMap d'. (A billentyűzet dekódolási táblázat neve elé nem kell elérési utat megadni, a 'SetMap' automatikusan a 'Devs:keymaps'-ban keresi.)

Ezek a legfontosabb szabvány alkatalógusok, amelyeket az operációs rendszer használ.

Érdekes kérdés, hogy a rendszer hogyan és mikor állapítja meg, hogy miből gazdálkodhat. (Előfordulhat ugyanis, hogy egy lemezen nincs meg egy vagy több szabványos alkatalógus.) Ennek megértéséhez érdemes részletesen megvizsgálni egy rendszertöltés fontosabb mozzanatait.

Bekapcsolás vagy reset után a ROM-ban lévő indítóprogram ellenőrzi, hogy a 0-ás lemezmeghajtóban lévő lemezzel lehet-e rendszertölteni. (Ilyen lemezt az 'Install' AmigaDOS parancssal lehet készíteni.) Ha lehet, akkor megvizsgálja a főkatalógusból nyíló szabványos alkatalógusokat — ha vannak. A memóriában elkészíti azt a hozzárendelési táblázatot, amely tartalmazza, hogy melyik szabványos logikai katalógushoz melyik fizikai alkatalógus tartozik. (Ezt a táblázatot az 'Assign list' vagy paraméter nélküli 'Assign' AmigaDOS parancssal lehet megnézni.) Ezután — ha létezik — betölti a 'Devs:system-configuration' file-t, amely a rendszer általunk beállított — l. 'Preferences' — paramétereit tartalmazza. (Ha nincs meg ez a file, akkor a ROM-ban lévő alapértelmezés érvényes.) Ezek után az alapparamétereknek megfelelő Workbench képernyőn egy AmigaDOS nevű CLI-ablakot nyit, majd megnézi, hogy van-e 'S:Startup-Sequence' file. Ha nincs, a rendszer töltése befejeződik és a CLI-ből máris lehet dolgozni; ha van, akkor a már ismert módon elkezdődik ennek a szövegfile-nek a feldolgozása.

Ezek ismeretében már érthető, hogy néha miért kéri vissza a gép a rendszerlemezt, noha a felhasználó egyáltalán nem





hivatkozott közvetlenül rá. Ilyen fordulhat elő nyomtatáskor, a 'Say' program használatakor, az első RAM DISK-re való hivatkozásakor, a Workbench 'Info' menüpontjának használatakor stb. Ilyen esetekben a rendszer azokat a tranzien rendszer-programokat hívja be a szabványos logikai katalógusokon keresztül, amelyek az adott funkció kezelését) vezérlését végzik. Az a tény, hogy a rendszer ezeket a funkciókat a logikai katalógusokon — ill. a hozzárendelési táblázaton — keresztül bonyolítja le, ad lehetőséget arra, hogy szükség esetén — a táblázatot

## 500 ... 2000 ... 2500 ... 3000 ...

1989 első negyedében több, Amigához kapcsolódó újdonsággal jelentkezett a Commodore cég.

A már korábban kifejlesztett A2090A jelű nagy sebességű merevlemezvezérlő kártya (amely egyszerre 4 SCSI és 2 ST506 csatlóójú egységet tud meghajtani) után az A2620/2 jelű processzorkártya is kapható. Az A2620/2 egy 14,32 MHz órafrekvenciával — pontosan az eredeti kétszeresével — működő 32 bites 68020-as processzort, egy 68881-es lebegőpontos matematikai társprocesszort (opcionálisan 20 vagy 25 MHz-es 68881-es vagy 68882-es FPU), egy 68851-es memóriakezelő egységet és 2MB 32 bites RAM-ot tartalmaz.

Az A590 egy 20MB-os winchester rendszer, amelyben egy csatlakozóaljzat is van RAM-bővítés számára 2 MB-ig. Feltehetőleg a cég ezzel kívánja általánossá tenni a winchester használatát.

A nem Amiga vonal is erősödött. Az ismert A2088 jelű XT-kártya után megjelent az AT-kártya is (A2286). Az A2286-ban egy 8 MHz-es Intel 80286-os processzor (opcionálisan 80287 is), 1MB RAM, 128 KB dual-port RAM (ezen keresztül kommunikál az AT és az Amiga), 16 KB BIOS, egy 1,2 MB-os 5 1/4 collos lemez meghajtó (opcionálisan 360 KB/720 KB 3 1/2 coll is) és akkumulátorral védett óra van. A kártyához MS-DOS 3.3-as operációs rendszert és GW-BASIC-et szállítanak.

megváltoztatva — módosítsunk a rendszer működésén. Így érhető el pl., hogy egy prog°am ugyanúgy működtethető legyen winchesterről, RAM DISK-ről, mint hajlékony mágneslemezzel, vagy hogy az összes AmigaDOS parancsot — megfelelően nagy szabad memória esetén — RAM-rezidenssé tegyünk. Ez utóbbit az alábbi két paranccsal lehet megvalósítani: 'Copy C:RAM:C', majd 'Assign C:RAM:C'.

Újhelyi Gyula

Az eddig ismertett újdonságok — az A590 kivételével — mind kártya kivitelűek, így csak a 2000-es alapú rendszerekben alkalmazhatók.

Két új géppel gazdagodott az Amiga-család. Az Amiga 2000HD-vel és az Amiga 2500-zal. A 2000HD egy Amiga 2000-es az A2090A vezérlővel és egy 40MB-os winchesterrel. A 2500-as szintén egy Amiga 2000-ből mint alapgépből áll, kiegészítve A2620/2 + A2090A + 40MB winchesterrel. Az Amiga 2500-hoz az Amiga operációs rendszerén kívül a UNIX is használható (UNIX System V, Release 3.1), amelyet itt AMIX-nek neveznek. A 2500UX konfigurációhoz a 40MB-os helyett egy 80 vagy 100MB-os winchester és egy 150MB-os streamer is tartozik.

Vajon mi jöhet a 2500-as után? Természetesen a 3000-es! Főprocesszora Motorola 68030-as 4MB RAM-mal. Tudni fog egy új — nem váltott soros — grafikus üzemmódot is: 1024 x 1024 képpont, több mint 16 millió színből egyszerre 256 lehet a képernyőn. A jobb grafikán túl egy továbbfejlesztett hang chip (!?) is javítja a gép lehetőségeit. A chip RAM 2MB (chip RAM-nak nevezik azt a memóriaterületet, amelyet a társprocesszorok is elérnek), de kell is ez, mivel az új képernyő memóriafoglalása kerekén 1MB! Az alapkiépítéshez 80MB winchester, 5 Amiga és 5 IBM AT bővítőhely is tartozik. A torony kivitelű gép operációs rendszere is érdekes: 2.0-ás verziójú Workbench (!), amely csak a 3000-en fog futni. Mindez nagyon jól hangzik, de '89 utolsó negyedéve előtt nem valószínű a megjelenése, így addig az Amiga 2500 a sorozat csúcsmo­dellje.

Ú. Gy.

## Tisztelt Szerkesztőség!

A csákvári Kossuth Művelődési Ház és a TIT Fejér Megyei Szervezete ismét találkozót szervez a számítógép-felhasználók és a számítógép-barátok részére. Rendezvényünkre a Csákvári Napok keretében kerül sor.

A találkozóra mindazokat várjuk, akik Commodore, Sinclair és Videoton Tv Computer típusú gépekkel dolgoznak, és mindazokat, akik még csak ismerkednek a számítógépekkel. Várjuk a lelkes amatőröket, és a szakmát magas szinten művelő profikat.

A program keretében tág lehetőséget biztosítunk a mikrogép-tulajdonosoknak a szoftverek csereberéjére is. Amennyiben programot akar cserélni, kérjük, szá-

mítógépet és monitort hozzon magával. Asztalt és elektromos áramot térítés ellenében biztosítunk.

A találkozó színhelye a csákvári Sportcsarnok lesz. Időpontja: 1989. augusztus 27. (vasárnap) A program és a cserebere reggel 10 órától délután 16 óráig tart.

Kérjük az Önök segítségét, hogy ez az információ minél több érdeklődőhöz eljusson.

Segítségét előre is köszönjük!

Tisztelettel:

**Hujber Antal**  
műv. ház ig.



## Operációk BASIC-ben

# Így számol a komputerünk

(II. rész)

Az egyes bitek tudatos be- és kikapcsolása mellett a második rész az igazságértékekkel is foglalkozik. A számítógépünk az igaz/hamis logika tudományág területére lép.

Miután az első részben a számunkra közvetlenül rendelkezésre álló BASIC műveletekkel foglalkoztunk, ma egy új, érdekes és igényes téma felé fordulunk. A téma pedig az igazságértékekkel való programozás, illetve a WAIT parancs leírása és használata.

## BITEK BE- ÉS KIKAPCSOLÁSA

Térjünk rá mindjárt az első gyakorlati tippre a BASIC programozók számára (az ÉS illetve a VAGY logika az előző részből bizonyonnan elöttünk van még). A logikai műveletek a POKE utasítás használatakor jelentenek sokat. Például, ha arra van szükség, hogy a gép tárolójában egyes bitek értékét határozzuk meg egy kívánatos minta szerint. A C64-es esetében ez a zene és a grafika programozásánál elkerülhetetlen, de alkalmazást a C128-as és a Plus4-es esetében is találunk. A megoldás a POKE, valamint az OR/AND funkciók kombinált használata.

Ha szeretnénk egy adott bitet az eredeti értékétől függetlenül bekapcsolni (1-es érték), vagy bekapcsolva hagyni, az alábbi egyszerű képletet kell megjegyezni:

```
POKE X,PEEK(X) OR 2↑Y
```

Az X az a tárolócella cím, ahol az Y-dik bitet (0–7!) akarjuk befolyásolni.

Ha például a 600-as címen a 7. bitet akarjuk bekapcsolni, az alábbi sort kell a föntiek szerint beadni:

```
POKE 600,PEEK(600) OR 2↑7
```

A C64-esnél a sprite-ok aktiválása gyakori feladat. Itt az Y annak a sprite-nak a száma, amelyet aktiválni kívánunk:

```
POKE 53269,PEEK(53269) OR 2↑Y
```

A működés megértése nem lehet nehéz. A fönti példóban legyen a 600-as cella 7. bitje nulla. A második operandusnál azonban a hetedik bit értéke 1 (be van kapcsolva), hiszen a  $2 \uparrow 7 = 128$ , azaz binárisan  $\%10000000$ . Az OR kapcsolás értelmezése szerint a két bitérték kombinációja az 1-es érték lesz. A 7. bitet tehát bekapcsoltuk. A többihez nem nyúltunk.

Ha most az egyes bitek törlését (kikapcsolás, lenullázás) akarjuk megvalósítani, akkor az AND utasításra van szükség:

```
POKE X,PEEK(X) AND (255-2↑Y)
```

Az X az a tárolócella cím, ahol az Y-odik bitet (0–7!) akarjuk befolyásolni. Például:

```
POKE 53269,PEEK(53269) AND (255-2↑13)
```

A C64-esnél a 3. sprite-ot kikapcsoljuk.

Itt sem nehéz a működés megértése. Bármely értékű is az adott tárolócellában a kiszemelt bit, a kívánt helyen a  $255-2 \uparrow Y$  formula a nulla értéket hozza létre a kettes számrendszerben felírt tárolócella érték esetében. A nulla érték ÉS kapcsolása (AND) hatására azután a kiszemelt bitet töröljük. Minden más bit értéke változatlan marad. A sprite kikapcsolási példában a  $255-2 \uparrow 3 = 247$ , ami binárisan  $\%11110111$  lesz. A harmadik bit (a számlálás 0-tól 7-ig!) értéke tehát nulla. Próbálkozzunk meg a számításokkal úgy, hogy magunk írjuk föl a bináris értékeket. Ne feledjük el, hogy csak 1 byte-os számokkal (0–255) dolgozunk, amikor a bitek ki- és bekapcsolása a feladat.

Térjünk rá most a másik témánkra. Ehhez előbb hadd állapodjunk meg egy fontos kérdésben. Az igaz/hamis kiértékelés a helyes kifejezés esetében a  $-1$ , a hamisnál pedig a nulla értéket adja eredményül.

## IGAZSÁGÉRTÉKEK PROGRAMOZÁSA

A fontú tulajdonságot kihasználhatjuk a programjainkban számítások végrehajtására (az igaz/hamis kiértékelés ugyanis a BASIC-ben egész számokkal dolgozik). Mutatnánk néhány olyan megoldást, amelyet a BASIC programozásnál használhatunk.

REPEAT...UNTIL

A repeat-until hurokban a kilépés akkor történik meg, ha egy bizonyos feltételt teljesítettünk. A hurkot legalább egyszer lefuttatjuk.

A C16-oson és a C128-ason ez a fajta hurok megtalálható a DO LOOP UNTIL alatt. Sajnos, ezt a BASIC V2.0 nélküli. De a szimuláció megtörténhet:

```
10 FOR FUT=-1 TO 0
```

```
20 REM a hurok akciómagja
```

```
30 FUT = NOT (a feltétel)
```

```
40 NEXT FUT
```

Egy konkrét példán a megoldás jobban szemléltethető:

```
10 FOR FUT=-1 TO 0
```

```
20 READ B$
```

```
30 FUT = NOT (B$="")
```

```
40 NEXT FUT
```

Ez a kis program csak a példa megvilágítására készült. Önmagában nem futtatható, hiszen nem állnak rendelkezésre DATA sorok. A működés viszont megérthető. A FOR—NEXT hurkon belül a változót (itt: FUT) megváltoztathatjuk. Ha a hurok  $-1$ -től nulláig fut, akkor a változót mindig  $-1$  értékre kell állítani, ha folyamatosan ismételni szeretnénk azt. Ha a változó értéke nulla lesz, megszakítjuk a hurkot. Az egyetlen szokatlan dolog a programunkban a FUT=NOT (feltétel) sor. De gondoljuk csak át. Ha a feltétel a mi esetünkben hamis lesz (ennek értékeredménye nulla), akkor a FUT változó értéke  $-1$  lesz (NOT 0= $-1$ ). Ha azonban a feltétel teljesül, akkor a változónk értéke azonnal nulla lesz, és a hurokból kiszállunk.

WHILE...DO HUOK

A while-do hurok esetében, amely a repeat-untilhoz hasonlóan a BASIC V2-ben nincs meg, az a feltétel, amelynek biztosítania kell a hurok lefutását, elől áll. Ha a feltétel nem teljesül, átugorjuk a hurkot, azaz itt nincs szükség még az egyszeri lefutásra sem. A C16-nál és a C128-nál ezt a hurkot a DO LOOP WHILE szintaktikával tudjuk programozni.

Egy lehetséges BASIC V2 struktúra az alábbi lehet:

```
10 ON -(NOT(feltétel)) GOTO 40
```

```
20 REM a hurok akciómagja
```

```
30 GOTO 10
```

```
40 REM a hurok vége
```

Itt is adnánk egy konkrét megoldást:

```
10 ON -(NOT(LEN(B$) = 20)) GOTO 40
```

```
20 B$=B$+" "
```



30 GOTO 10

40 REM a hurok vége

A B\$ változóhoz mindaddig szóközöket láncolunk hozzá, amíg az el nem éri a húsz karaktert.

A funkciót az alábbiak szerint érthetjük meg. Az ON... GOTO (illetve az ON...GOSUB) utasítás a többszörös ugrási hozzárendelés eszköze (pl. ON A GOTO 100,200,300, ahol ha az A értéke 1, 2 vagy 3, akkor a 100-as, a 200-as vagy a 300-as sorra ugrunk). De mód van arra is, hogy itt csupán egy ugrási célt adjunk meg. Ha a változó értéke itt (plusz) 1, akkor a megfelelő sorra ugrunk. Ha az érték egytől különböző, a sorban a következő utasítást hajtjuk végre, amely itt a következő BASIC sorban található. Az elemzés szerint a mínusz NOT... kifejezés értéke akkor lesz +1, ha a feltétel teljesül. A while-do hurkot máris elkészítettük.

## AZ ISMERETLEN WAIT PARANCS

A normál BASIC programozó igen ritkán használja a WAIT parancsot. Ennek oka az, hogy az alkalmazás erősen korlátozott, és a szintaktika igen komplex.

Ez a parancs mindaddig leállítja a futó BASIC programot, amíg egy meghatározott tárolócellában található érték (adott bináris minta) elő nem áll. A WAIT utasítás BASIC alkalmazása azért nehézkes, mivel szinte kizárólag az adott tárolócellákkal összefüggő külső akciók lekérdezése történhet meg (a BASIC programot leállítjuk, semmilyen más tárolócella értéke nem változtatható meg).

Ráadásul a szintaktika is áttekinthetetlen: WAIT X,Y[,Z]. A WAIT parancs után egy címet kell megadni (X), illetve egy (Y), esetleg választhatóan még egy (Z) értéket.

A funkció magyarázata sem egyszerűbb. Nézzük azt a példát, hogy a cím után két értéket adtunk meg. Akkor a megadott címen (X) található értéket (bitmintát) a logikai KIZÁRÓ VAGY funkcióval (exkluzív vagy-nak EOR-nak is nevezik), összekapcsoljuk a másodikkal megadott értékkel (Z), amelyet emiatt EOR maszknak neveznek. Az így megkapott értéket az elsőnek föltüntetett értékkel (Y) a logikai ÉS (AND) funkcióval kapcsoljuk össze. Az Y-t ezért AND maszknak is nevezik. Amennyiben a WAIT parancsnál a címet csak egy érték követi, úgy csakis az ÉS kapcsolás (AND) történik meg. Mivel a logikai műveletek megértése nehezebb (lásd ehhez feltétlenül az első részt), ezért logikus a ritka alkalmazás.

Ha a logikai műveletek eredménye nulla, akkor a megadott címet a WAIT parancs továbbra is megfigyelés alatt tartja, a BASIC program továbbra is áll. Ha azonban a művelet a nullától különböző értéket eredményezett, máris tovább megyünk a sorban következő BASIC paranccsal. Egyébként a régebbi 64-es, Plus4-es kézikönyvekben a WAIT parancs leírás hibás.

Amennyiben a WAIT parancsot egy értékkel használjuk, akkor a programunkat mindaddig leállítjuk, amíg a megadott tárolócellában egy meghatározott bit értéke 1 nem lesz. WAIT 653,1

Most a C64-es addig vár, amíg meg nem nyomjuk a SHIFT gombot. Ugyanis a 653-as tárolócella figyeli a SHIFT (0. bit), a Commodore (1. bit) és a CTRL (2. bit) gombokat. A fenti példában a 653-as címen található értéket az 1-es értékkel az AND funkció szerint kombináljuk:

A 653-as cím	érték	bit	7	6	5	4	3	2	1	0
nincs gombnyomás: AND	0		%0	0	0	0	0	0	0	0
AND maszk:	1		%0	0	0	0	0	0	0	1
az eredmény:	0		%0	0	0	0	0	0	0	0

Ha nem nyomtuk meg a <SHIFT>-et, úgy a címen a nulladik bit értéke nulla (a bit ki van kapcsolva). De így az AND maszk szerinti logikai művelet eredménye is nulla. Hiszen (lásd az előző részt) az ÉS kapcsolás igazságtáblázata szerint a bitek

összekapcsolásánál akkor és csak akkor kapunk 1-et, ha mindkét bit be van kapcsolva (azaz értékük 1). Amennyiben azonban megnyomjuk a <SHIFT>-et, az operációs rendszer azonnal bekapcsolja a C64-esnél a 653-as cellában a 0. bitet. Most az AND művelet eredménye a 0. bitben az 1 lesz, aminek hatására a WAIT parancs befejeződik, a program továbbfut.

Még egy példa ehhez:

WAIT 653,6

Ebben az esetben az AND maszk értéke két bekapcsolt bitet ad, amelyet mi decimálisan a 6-os értéknek értelmezünk. Azaz  $2 \cdot 1 + 2 \cdot 1 = 6$ , vagyis a lekérdezésnél az 1-es és a 2-es bitet figyeltetjük. Az elmondottak szerint ez a két bit a Commodore és a CTRL gombokhoz tartozik. Vagyis a WAIT utasítás befejezéséhez nekünk vagy a <C=>, vagy a <CTRL> gombot kell megnyomni. Az első esetben a logikai ÉS kapcsolás eredménye a decimális 2 érték lesz, a másikkal a decimális 4. Mindkettő nullától különböző, vagyis a megszakítási kritérium előáll.

## WAIT KÉT PARAMÉTERREL

Ha a WAIT X,Y,Z szintaktika szerint használjuk az utasítást, úgy a programot mindaddig leállítjuk, amíg egy bit kikapcsolódik (Y=Z), vagy egy bit kikapcsolódik, vagy egy másik bekapcsolódik (Y ≠ Z).

A WAIT parancsnál a logikai kizáró vagy (EOR) műveletet használjuk. Ez a C64-esnél és a Plus4-esnél nincs meg, a C128-as tulajdonosok a XOR alatt találják meg ezt a BASIC parancskészletben. A kizáró vagy kapcsolásnál (lásd az előző részt, illetve a Boolean algebra szakirodalmat) az eredmény abban az esetben lesz 1, ha az összevetett két bit értéke egymástól különbözik (0/1 vagy 1/0). Amennyiben mindkét összevetett bit kikapcsolt vagy bekapcsolt állapotban van, a logikai kapcsolás eredménye nulla lesz. Például várjon arra a C64-es, míg a 2-es port joystick tűzgombja megnyomott állapotba kerül.

WAIT 56320,16,16

Nézzük meg alaposan az 56320-as tárolócellát:

Az 56320-as cím	érték	bit	7	6	5	4	3	2	1	0
EOR	127		%0	1	1	1	1	1	1	1
EOR maszk (Z=16)	16		%0	0	0	1	0	0	0	0
a köztes eredmény: AND	111		%0	1	1	0	1	1	1	1
AND maszk (Y=16)	16		%0	0	0	1	0	0	0	0
az eredmény:	0		%0	0	0	0	0	0	0	0

Így a WAIT utasítás folyamatosan ismételi, mert a végeredmény nulla. Ha azonban a tűzgombot megnyomjuk, akkor az 56320-as címen álló érték bitmintája %01101111 lesz, mi'e a WAIT parancs befejezi a munkáját.

Elméletileg lehetséges a számítógépet két különböző Y és Z érték megadásával arra várakoztatni, hogy egy bit kikapcsoljon, ugyanakkor az azonos cím egy másik bitje bekapcsoljon. Értelmes alkalmazást azonban erre nem tudunk keresni.

Mára ennyi elég is lesz. Most már ismerjük a WAIT parancsot, a POKE-kal pedig komolyabban tudunk bánni. Lentebb még egyszer összefoglaljuk ezeknek a parancsoknak az alkalmazását. A következő részben kizárólag az igazságkeresés logikai műveleteivel fogunk foglalkozni, illetve megoldunk néhány logikai feladványt is.

1. várakozni, míg egy cím adott bitje bekapcsol:

WAIT cím, 2↑A

2. várakozni, míg egy cím adott bitje kikapcsol:

WAIT cím, 2↑B, 2↑B

3. várakozni, míg egy cím adott bitje be-, egy másik pedig kikapcsol:

WAIT cím, 2↑B+2↑A, 2↑B

A — annak a bitnek a pozíciója (0—7), amelyiket bekapcsoljuk.

B — annak a bitnek a pozíciója (0—7), amelyiket kikapcsoljuk.



Feketén-fehéren

# A NYOMTATÓ- PROGRAMOK FEJLŐDÉSE



Gutenberg, a könyvnyomtatás feltalálója erről legfeljebb csak álmodhatott. A modern nyomtatóprogramok segítségével a dokumentumok előállítására oly egyszerű, mint a krumplipeccételés.

## GUTENBERG NYOMÁBAN

A történet néhány évvel ezelőtt kezdődött. Két amerikai programozó kifejlesztett egy programot, melynek segítségével minden számítógép-tulajdonos képes volt elektronikus üdvözlőket küldözgetni, feltéve, ha a címzett is rendelkezett azonos típusú géppel. De épp ez volt a dologban a bökkenő. Hisz hogyan lehetett a barátnőt vagy a nagypapát meggyőzni arról, hogy ő is vegyen egy alkalmas házi számítógépet.

Amikor a fejlesztők fölismerték ezt a problémát, nekiláttak átírni programjukat. El is készült minden nyomtatóprogramok atyja, a Print Shop. Ez a program azzal az eleddig szokatlan ötlettel kápráztatta el a számítógépes világot, hogy megadta a szöveg és a grafika keverésének lehetőségét. Ezzel aztán lehetővé vált — teljesen függetlenül a számítógép jelkészletétől —, hogy szinte bármiféle írásképet/betűtípust használjunk.

A rendelkezésre álló funkciókat (üdvözlőlapok, feliratok, méteres transzparens, levélpari készítés) kis idő múlva a Printmaster program is a rendelkezésünkre bocsátotta, amelyet direkt konkurenciaként készítettek el. Buta módon ez olyan jól sikerült, hogy a Print Shop készítője, a Broderbund Software másolásért perelt és nyert is.

## AZ UTÓDOK

Az ítélet után azonnal megbízatásba adott Printmaster Plus program ugyan elkészült, de az soha nem jutott el Európába C64-es változatként. S amint azt az IBM változat forgalmazója mondja, ez nem is fog megtörténni. De nem is volna fair dolog ezt a programot a következőkben említettekkel összehasonlítani, mert ezek sokkal többet tudnak (igaz, a kezelésük viszont nehezebb).

Van egy fontos tulajdonság, ami innen indult, és azután egyre inkább elterjedt, ma pedig már mint szabvány él. Ezzel már a Printmaster is szolgált. A technika neve WYSIWYG. Emögött a rövidítés mögött a What You See IS What You Get angol kifejezés áll, ami gyakorlatban annyit tesz, hogy amit beadunk a képernyőre (s ott megjelenik), azt fogjuk viszontlátni a papíron is. Azaz a képernyőn szemrevételezhetjük a kész munkát. Egy modern pontmátrix nyomtató igen nagy grafikus fölbontása (grafikus pont per adott felület) miatt a képernyőn általában vagy csak egy részletet látunk egyszerre, vagy a nagykép kicsinyített mását.

Mialatt a Print Shop és a Printmaster csupán csak kevés nyomtatási képváltozat alkalmazását tette lehetővé, a későbbi programok jóval nagyobb játékteret adtak a fantáziának.

A 89 márkába kerülő Newsroom-ot kimondottan az iskolai, egyesületi és egyéb újságok készítésére találták ki. Ez a program kevésbé való üdvözlőlapok készítésére, bár ez is lehetséges. A betöltés után egy grafikus igazán igényes menüben találjuk magunkat, amelyből föl hívhatjuk az egyes almenüket. A „Photo Lab”-ban kikereshetjük a mellékelt 600 grafikából (fotóból) azt, amelyekre szükségünk van. Az újság fejlécét a „Banner”, az egyes cikkeket a „Copy Desk”-ben készíthetjük el. A „Layout” segítségével az egyes elemeket összekeverhetjük. A „Wire Service” segítségével a teljes munkát vagy annak egy részletét más Newsroom tulajdonoshoz küldhetjük telefonon keresztül.

Azaz egy sokoldalú, sikeres programról lenne szó? Majdnem. Mert a legeslegutolsó és leglényegesebb menüpontban, a „Press” (azaz nyomda) menüben kapott eredmény nem néz ki jól. Bár a programot illesztették a német ékezetekhez (és német a kézikönyv is), de nagy gondot okoz a nyomtatóillesztés. A megadott interfész típusok ugyanis Európában ismeretlenek. De mégis, ha sikerrel járnánk, nagyon szegényes a kapott „nyomat”. Gyenge teljesítmény ugyanis a csekély, 480 pont/soros fölbontás. Ha jól meggondoljuk, némely nyomtató beépített jelkészlete szebb karaktereket ad.



A Newsroom kezelése egyszerű, de az említett nyomtatási gondok és a soványka minimal-editor nem igazán meggyőző. A programmal azonban, főleg a mellékelt rengeteg grafika miatt nagy öröm dolgozni.

## A SPECIALISTÁK

A Printfox program már sokkal hatékonyabb, mint a Newsroom, igaz a kezelése is sokkal komplikáltabb, s az ára is több (98 márka). A Printfox esetében külön van választva a szöveges és a grafikus editor. Mialatt az előbbi erősen emlékeztet a legismertebb szövegszerkesztőre, a Vizawriterre, a grafikus editor egy Hi-Eddi Plus-nak felel meg. Két igen jó program egyesítve egy nyomtatóprogramban! S ráadásul ez a szintézis kiváló eredményt ad. A szöveges editorban megírhatjuk, amit akarunk, vagy behívhatjuk azt a lemezről. Ezután megspékeljük azt megfelelő utasításokkal, amelyekkel meghatározhatjuk mi (szöveges részlet), hogyan (jelkészlet, íráskép modifikáció) és hol (pozíció) jelenjen meg az a papíron. Itt számolgatni is kell, hogy a Printfox-szal közölhessük, hova szeretnénk tenni a grafikákat. Öt jelkészletet mellékelnek a programhoz. Ajánlható a 78 márkába kerülő Karakterfox programcsomag megvétele, nemcsak amiatt, mert ez kiváló jelkészlet editor, hanem mert húsz újabb jelkészletet még mellékelnek.

A szöveges editorból a betűket a grafikus tárolóba „nyomtatjuk”. Ezt úgy képzelhetjük el, mint egy fél A4-es lap méretű oldalt. Ez 640x400 képpontnak felel meg, amelyből 320x200 képpontnyi kivágás látszik. A grafikus editor tartalmazza egy festőprogram minden fontos funkcióját. A tulajdonképpeni nyomtatás előtt a grafikává konvertált szövegbe be kell még tölteni a grafikákat. S erre az alapra építve egy kilenc tűs printerből fantasztikus minőséget vagyunk képesek kihozni! A rendszerhez mellékelt Setup programmal még a legegzenetikusabb nyomtató illesztése is elvégezhető. Kétféle nyomtatási minőségben készíthetjük el anyagainkat standard szerint. Létezik azonban PFOX+ néven egy program, amely egy harmadik nyomtatási minőséget is rendelkezésre bocsát. A gyártó kínál 48 márkáért egy 24 tűs gépekhez való rutincsomagot is PIN24 néven.

A Printfox nyitott rendszer, még magunk is írhatunk hozzá saját bővítéseket, ha megvan a kellő ismeretanyagunk. S ezek nemcsak nyomtatókezelők lehetnek, hanem egérmeghajtó vagy idegen grafikus formátumokat kezelő rutinok is. A gyártó Pagefox néven maga is kínál bővítést. Itt 100 kbytenyi plusz tároló segít minket a profi kiadványszerkesztésben.

## C64 180 KBYTE-TAL

Hadd szóljunk konkrétan a Pagefoxról. Ezt a programot, hasonlóan a Printfoxhoz, Hans Haberl készítette. Egyszerű kezelés, verhetetlen szolgáltatási spektrum került bele egy szerény modulháza. Ellentétben az egyéb nyomtatóprogramokkal, a Pagefox ugyanis modulként került forgalomba. Mindennek persze megvan a maga oka. A modulban ugyanis a 64 kbyte program és jelkészletek mellett plusz 32 kbyte is a rendelkezésünkre áll. Ennyi munka természetesen az árban is tükröződik, 248 márka nem kis érvágás. Ezért azonban kapunk is valamit cserébe, mert a Pagefox nemcsak hogy mindent tud, amit a Printfox és a később ismertetendő Publish 64, hanem néhány olyan tulajdonsággal rendelkezik, amelyek igencsak figyelemre méltók. Megemlíthetjük például a kontúrszedést, ami annyit tesz, hogy a szöveget a grafika kontúrja mentén jeleníti meg, nem úgy, mint például a Newsroom, ahol a grafika mindig négy-szöget „harap” ki a szövegből. De a meglévő és igen jól dolgozó automatikus (német) elválasztás is klassz, mivelhogy igen helyesen dolgozik, a komplex elválasztási szabályok ellenére is. Egyébként a kontúrszedés nem egy drága PC-kre írt DTP programban hiánycikk. Mindezeket túl a Pagefox kezelése igen egyszerű (egér vagy joystick). A felhasználó rendelkezésére itt mindig egy komplett A4-es oldal áll, azaz a C64-esünk a meg-hökkentő 640x800 képpont kezelésére alkalmas. Ez pedig ebben

a géposztályban egyszerűen fantasztikus.

A nyomtatást tekintve a Printfox és a Pagefox azonosak, a 9 tűs mátrixnyomtatókkal elért telített, harmonikus képek már-már műremeknek minősülhetnek.

A Scanntonik cég Printfox és a Pagefox terméke tehát a kiváló nyomtatási minőséggel hívja föl magára a figyelmet. Azonban a programok ár/szolgáltatás viszonya is meggyőző. A Pagefox ezenkívül az egyszerűbb kezeléssel, az automatikus szövegválasztással és a betűk tetszőleges modifikálásának lehetőségével (kövér szedés, outline, duplaszedés/magas, árnyék stb.) is kitűnik. A Pagefox a „kistestvérének” minden elemét (jelkészletek, grafikák) föl tudja használni. Ehhez a két programhoz számtalan plusz grafika és jelkészlet kapható már.

Mindazok számára, akiknek az eddig említett kommersz szoftverek drágának minősültek volna, alternatívaként rendelkezésükre áll a Publish 64. Ezt a programot a 64'er Magazin 1988. novemberi száma közölte le a Hónap Listájaként, azaz aki hajlandó bepötyögni, a füzet árában (6,50 márka) kaphat egy jó nyomtatóprogramot.

A Publish 64 program komfortos szöveges editorja erősen emlékeztet a „Mastertext” szövegszerkesztőre. A „gyárilag” beépített grafikus program inkább egy minimális megoldás, a legfontosabb funkciók azonban (pl. vonalhúzás, zoom) megvannak. Bár az alap gondolat (szövegírás a szöveges editorban, ennek a grafikus tárolóba való „nyomtatása”, majd a grafikus tároló papírra vetése, valamint egyéb tulajdonságok is, mint a két 640x400 képpontos féloldal kezelése) emlékeztet a Printfoxra, hamar észrevehetőek a különbségek is. Például csak zoom (nagyítás) üzemmódban van mód egy féloldal négy grafikus képernyőjét görgetni, cserébe viszont nincs szükség bonyolult számításokat végezni a szöveges layouthoz. A Publish 64 esetében „szöveges boxokat” helyezünk el a képernyőn, ahová elosztjuk a megírtakat. A nyomtatás meggyőző lehet, de az erősen függ a használt jelkészletektől. A GEOS-ból „lopott” karakterek úgy-ahogy megfelelnek, a Printfox jelkészletekhez írandó konvertálóprogram bizonyos megoldaná a problémát.

A nyomtatórutin maga hasonlít a Printfoxéhoz. Kétszer nyomtatunk egy sort, miközben a második előtt egy aprócska soromelés történik. Jelenleg a Publish 64 csak Epson és kompatibilis gépekre írt nyomtatórutinokkal rendelkezik.

Amikor a nyomtatóprogramokról beszélünk, meg kell említeni egy-két szövegszerkesztő programot, amelyek ugyan régebbiek mint a Pagefox, de megvannak áldva néhány figyelemre méltó tulajdonsággal. A nevezett két program a Fontmaster II és a Fontmaster 128. A Fontmaster II inkább egy jó szövegszerkesztő program HiRes grafika nyomtatással, a Fontmaster 128 pedig inkább egy nyomtatóprogram. Bár a felhasználónak lehetősége van mintegy 128 nyomtatómeghajtó és mintegy 20 interfész típus közül válogatni, a nyomtatóillesztés mégis sok fejfájást okoz. (Ez a Newsroom szindróma: a megajánlott interfészek Európában ismeretlenek.) Mindkét Fontmaster program preview („megtekintés”) funkcióval dolgozó szövegszerkesztő, ezt azonban szokatlan módszerrel valósítják meg. Tulajdonképpen a megtekintés során a munkánkat HiRes grafikaként „nyomtatjuk” sorról sorra a képernyőre. A Fontmaster II a végeredményt csak sejteti a képernyőn, míg a C128-as változat a későbbi munkáról egy meglehetősen egzakt képet ad.

Ha szöveg és grafika keveréséről van szó, a Fontmaster II elbújhat. A ráfordítás mértéke nem áll arányban az eredménynyel. A Fontmaster 128 azonban négy egyszerű parancsot bocsát a rendelkezésünkre, amelyekkel akár képernyő méretű grafikákat lehet befűzni a dokumentumba. A nyomtatás minősége összevethető a Printfoxéval, a program súlypontja azonban inkább a szövegszerkesztésben van, semmint a szöveggrafika keverésben.

A lapzártáig a Fontmaster utód még nem állt a rendelkezésére. A forgalmazó állítása szerint a „Phoenix, das Textwunder” (Phoenix, a szövegíró csoda) nevű program hamarosan kapható lesz. Ez a program átdolgozott Fontmaster változatok egyesítése. Egyébként igen elte°jedt, és a vevőket meglehetősen bosszantó rossz szokás, hogy a termékeket a megjelenés előtt hónapokkal már hirdetik.



**RÖVIDEN, TÖMÖREN**

Annak ellenére, hogy az „ősapa”, a Print Shop már csak az USA-ban kapható, a Printmaster pedig teljesen eltűnt, a C64-es nyomtatóprogram-helyzet elevebb, mint valaha. A Newsroom szép, átgondolt program borzalmas nyomtatási minőséggel. A Print- és a Pagefox az igen jó nyomtatási képpel győzik meg az embert a viszonylag komfortos kezelés mellett. A legdöntőbb pluszpont azonban a nyitott koncepció, amely úgy a gyártónak, mint a gépi programozásban jártas felhasználónak lehetővé teszi a program bővítését, saját igényekhez való illesztését. Ha a ráfordításokat, az eredményt és az árat figyelembe vesszük, a Printfox a legjobb választás. Még egyszerűbb a dolgunk a Publish 64-gyel, amely az alulkínálhatatlanul olcsó ára mellett nem is ad rossz nyomtatási képet. Itt problémát csak a nem Epson kompatibilis nyomtatókhoz, és a szokásos szöveg/grafika formátumokhoz való illesztés jelent. Ugyanez érvényes a Fontmaster II-re és a 128-ra, bár ezek közül utóbbi meggyőző.

A piac mozgásban van, de azért áttekinthető. Gyakorlatilag mindenféle nyomtatási kívánsághoz megtalálhatjuk a megfelelő programot. Ha a nyomtató és a program összhangban van, akkor a papírra vetett eredményből nem is gondolnánk, hogy „csak” egy C64-es rejtezik mögötte. A nyomtatóprogramok szép jövő előtt állnak, a C64-esről már nem is beszélve.

**VERSENYEN KÍVÜL**
**A példátlan:**

Egy termék több szempontból is kiesik a körből, és ez a „Technicus”. Ezen a néven egy nyomtatóorientált programcsomagot árulnak, ahol az egyik figyelemre méltó tulajdonság a meglepően jó NLQ írásminőség. Ennek elérése céljából egy gépi kódú programot és a kívánt jelkészletet a C64-es RAM-jába kell írni, amely azután — pl. szövegszerkesztővel elkészített — szövegeket grafikává konvertálja, s a csatlakoztatott nyomtató segítségével papírra veti. Itt (a használt szövegszerkesztőtől függően) kompatibilitási problémákra számíthatunk. Éppen ezért érdemes a Vizawrite-ot használni, mert ehhez a programhoz egy külön Technicus változat létezik. Utóbbival a szövegeket automatikus sorkizárással is elkészíthetjük.

Ha egy jó, mechanikusan is megfelelő 9 tűs printert feltételezünk, akkor az írás minőségét tekintve a Technicus program az összes itt megemlített nyomtatóprogramot lazán zsebrevágja, bár a kínosan lassú szövegnyomtatás békáját is le kell ekkor nyelnünk. A szépírás-meghajtókön kívül különböző nyomtató segédprogramok, hardcopyrutinok is rendelkezésre állnak, amelyek miatt a Technicus program megérne egy külön misét is.

**Nem konkurencia:**

Az utóbbi időben két további nyomtatóprogram gazdagítja a német piacot. Az egyik a „Designmaker”. Ez a program a maga 45 márkás árát tekintve jutányos, és emellett meglehetősen jó nyomtatást készít. A kezelés azonban annyira bonyolult, hogy egyelőre csak kevesen barátkoztak meg vele. Egy másik program az ár szempontjából veri az összes többi. A „Create Page!” program csupán 19 márka. A kezelés tekintetében ez a program egy ötlettel jobb, mint a Designmaker, bár előbbi valamivel jobb minőségű eredményt ad.

Sem a Designmaker, sem a Create Page! nem tudnak elég meggyőzőek lenni, az alacsony ár ellenére sem, főleg a bonyolult kezelés és a hiányzó WYSIWYG funkció miatt.

**INFORMÁCIÓK:**

Printshop (49,95\$): Csak az USA-ban kapható  
 Broderbund Software, 17 paul Drive, San Rafael, CA 94903  
 USA  
 Printmaster (Plus): Már nem, illetve nem kapható.  
 Newsroom (89 márka):  
 Rushware, Bruckweg 128—132, D—4044 Kaarst 2  
 Printfox (98 márka) és Pagefox (248 márka):

Scantronik, Parkstrasse 38, D—8011 Zorneding  
 Publish 64:

6,50 márka újságárban, 88/11-es 64'er Magazin  
 29,90 programszerviz lemezen, Markt & Technik Verlag AG,  
 Unternehmensbereich Buchverlag, Hans-Pinsel-Str. 2,  
 D—8013 Haar

Fontmaster II/128 (98—98 márka):

Raab Bürotechnik, Friedhofstrasse 36, D—8605 Hallstadt  
 Designmaker (44,90 márka):

Romain Hoffmann, Mondorfer Str. 9., L—5552 Remich, Luxemburg

Create Page! (19 márka):

Hard- und Software Vertrieb Thorsten Lavid, Webschulstrasse  
 44, D—4050 Mönchengladbach 1

Technicus (39 márka):

Thilo Hermann, Poststrasse 6, D—7321 Börtlingen

**KI VOLT GUTENBERG?**

Johannes Gutenberg 1397-ben született, ő a könyvnyomtatás, pontosabban a „könyvnyomtatás öntött, mozgatható betűkkel” feltalálója. Gutenberg Friel Gensfleisch mainzi patrícus fia volt, akit a „zum Gutenberg” nevű házaról neveztek el. Gutenberg 1434 és 1444 között Strassburgban élt, 1448 után ismét Mainóban.

Kölcsönnel finanszírozta a latin „Gutenberg Biblia” nyomtatását, amelyből ma még 47 darab létezik. A biblianyomtatás után Gutenbergnek 1445-től pénzügyi gondjai voltak, 1458-ban ez aktákba is került. 1465-ben a mainzi érsek udvarába vette Gutenberget, ezzel biztosítva jövőjét. Gutenberg 1468. február 3-án halt meg Mainzban.

1900-ban alapították Mainóban a Gutenberg múzeumot (a nyomtatás művészetének világmúzeuma).





# OLIVETTI M/15 IBM PC-XT kompatibilis hordozható számítógép!

## M ű s z a k i      j e l l e m z ő k :

- Beépített NiCd akkumulátor (6 órás folyamatos üzem)
- 2 db 3.5"-os, egyenként 720 Kbyte-os floppy drive
- 25\*80 karakteres LCD kijelző, CGA emulációval
- Külső 5.25"-os floppy csatlakozási lehetőség
- I8088 processzor (4.77 MHz órajel)
- RS-232 és CENTRONICS interface
- 512 Kbyte RAM
- Súly: 5.7 kg

## T a r t o z é k o k :

- MS - DOS version: 3.20
- diagnosztikai lemez
- Dokumentáció
- Válltáska

Ára egy év garanciával:

**150.000Ft + ÁFA**

1136 Budapest, Sallai Imre u. 6.

Tel.: 310-776, 315-136

Tx: 22-6986 novtr h.

Fax: 530-605

**NOVOTRADE**

# PC Szalon

A SZÁMÍTÁSTECHNIKA BELVÁROSÁ



Komolyan veendő fenyegetés!

# számítógépes vírusok

A vírusprogramok egyre inkább fenyegetik a számítógépeket. A tömegtájékoztató eszközökben az „ártalmatlan számsorok”-tól a „szisztematikus gyilkosok”-ig mindent hallhatunk. S a C64-es sem marad ki ebből.

Egy pénteki napon történt, 13-án. Amikor a naptárra pillantottam, mindjárt tudtam: „Ma vigyázat!” Hogy a várható szerencsétlenséget elkerüljem, elhatároztam, hogy a napot zárt helyen, a számítógépem mellett töltöm el. Minő balgaság! Mert miközben a géppel dolgoztam, egyszerre elkezdett a képernyő kerete vadul villogni, majd a következő szöveget kaptam:

DR.DR.STROBE PAPA  
HACKER WAS HERE!  
ALSO STARRING GARFIELD!  
HALLO DICKERCHEN, DIES  
IST EIN ECHTER VIRUS!  
SERIALNO.: 18  
(Azaz: Halló, kövérke, ez egy igazi vírus!)

Mi történhetett? Ez egy vírus volna? A gépem talán beteg? A kikapcsolás vajon segített volna-e a bajon?

Nos, valóban egy számítógépes vírus jelentkezett be az említett esetben. A számítógépes vírusok olyan különleges programok, amelyek gondoskodnak a maguk sokszorosításáról, és eközben észrevétlenül bejutnak a számítógépbe és az adathordozókra. Ezek azután akár hosszabb ideig is ott várakozhatnak. Mikor végül akcióba lépnek, a katasztrófát nem lehet elkerülni, hacsak a vírus nem jóindulatú.

Egy ilyen jóindulatú vírussal fertőzött programot a májusi számban Mini-Vírus néven adtunk közre. Ez a vírus csak BASIC programokhoz „ragad”, és csak a SAVE utasítás használatakor szaporodik. A vírusmentesítés ebben az esetben nem okoz gondot. Könnyen észrevehető a fertőzés is. A Mini-Vírus példáján azonban számos vírusprogramozási fogást lehet tanulmányozni.

Az ártalmatlan vírusok úgynevezett „akciómagja” veszélytelen, azaz nem történik adatvesztés, bár a vírus maga szaporodik (továbbterjed). Vannak olyan vírusokhoz hasonló programok is, amelyek nem sokszorozódnak. Egy ilyen az 89/2. számban adtunk közre „Szövpók a képernyőn” néven. A nevezett pókprogramban időről időre leereszkedik a képernyőn egy pók, ami eleinte talán vidám, de később egyre inkább bosszantó.

## AZ IDŐZÍTETT BOMBA

Mindezzel szemben áll a Bayrische Hackerpost (BHP) társaság vírusprogramja (a BHP vírus, lásd lentebb), amely a kellemetlenebb, és inkább a veszélyesebb kategóriába tartozik. Az ilyenfajta vírusok két részből állnak. Az első, a „szaporodási rész” gondoskodik arról, hogy a vírusprogram minden elérhető adathordozóra rákerüljön. Eme tevékenysége közben a vírusprogram a még nem fertőzött adatállományokat keresi. Ha talál egy ilyet, akkor letárolja magát a lemezre (vagy a PC-k esetében) a fixlemezre. Ott azután úgynevezett beilleszti magát a directory és a fertőzött program közé. A directory az adott állományokról „létfonosságú” adatokat tartalmaz. Például azt, hogy áll az adott file a lemezen. Ezt az úgynevezett directory mutatóban kódoljuk, amely mint egy nyíl irányul az adott állományra. A vírus a szaporodási ciklusában lemezre írja magát, és eközben a még „egészséges” program directory mutatóját úgy alakítja át, hogy az észrevétlenül előbb mindig a vírusra irányul. A most már fertőzött program esetében tehát előbb mindig a vírust, és csak azután az adott programot hívjuk be.

Egy veszélyes vírus második részét akciórésznek nevezzük. Az a felelős a számos változta-

tásért. Ez az a rész, amely mondjuk az adatok visszahozhatatlan törléséért felel, amelyik formátálja a lemezt vagy a fixlemez, amelyik a saját vagy más érdekében visszaél az adatállományokban rejtőző információkkal. Megváltoztathatja a programokat úgy, hogy azok az adatok kimentése helyett törlik azokat, stb. A vírusok akkor lépnek akcióba, ha bizonyos feltételek adóttak. Vagyis a számítógépes vírusok időzített bombaként működnek. Hogy a bomba valamely naptári időpontban ( dátumellenőrzés) vagy egy adott (relatív) idő eltelte után (a vírus x-edik fölhívása) robban, az mindegy. A „ketyegés” ideje alatt a vírusnak épp elegendő ideje van arra, hogy az adott rendszerben elterjedjen. Számos rendszer fut még rendesen, közben pedig már rég fertőződött. Egy nap azonban fölrobban a bomba.

Ezeket a gyilkos programokat vírusnak nevezték el, mert azok pontosan úgy viselkednek, mint a biológiai ekvivalens: Ellenőrzés nélkül szaporodnak, információkat változtatnak meg, és a „gazdaállat” (értsd: számítógép és program) nélkül nem életképesek.

## A BHP VÍRUS

A legismertebb C64-es vírus a BHP vírus, amelynek funkcion keresztül szeretnénk a működését ismertetni. A „Bayrische Hackerpost” (röviden BHP) egy müncheni számítógépes klubot mindenki kinevette 1985-ben, amikor a tagok azt állították, hogy „minden számítógépen lehetséges vírusokat programozni, a házi számítógépeken (C64, Atari 800 stb.) is”. Hogy a kételkedőknek a torkára fagyasszák a nevetést, a BHP tagjai nekiálltak egy olyan vírust fejleszteni, amelyik rendelkezik egy veszélyes vírus minden tulajdonságával, de kárt nem okoz. A vírusnak feladata, hogy ellenőrzés nélkül

szaporodjon, és rejtőzzön el a számítógépben és a lemezen. Mindezeket a célokat sikerült megvalósítani a BHP vírusban! Ezek mellett a program resetálló, azaz akkor is a gépben marad, ha azt hardvagy szoftveres módon alapállapotba visszük. A vírust a C64-esből csak a gép kikapcsolásával lehet eltávolítani. A lemezekben azonban megmarad.

A BHP vírus akciórésze nagy vonalakban abban áll, hogy a fent leírt szöveget megjelenítse. Ennél a fajta vírussal az utolsó sor is feltűnő:

SERIALNO.: a generációs szám

A generációs szám adja meg a vírus által okozott fertőzések számát. Ez azt jelenti, hogy minden egyes fertőzésnél (továbbterjedésnél, szaporodásnál) ezt a számot megnöveljük. Nálam tehát a vírus 17 utódot produkált már. A főnti szövegkiadás után azonban tovább lehet dolgozni, egészen a következő bejelentkezéséig.

Alapjában két BHP vírust különböztetünk meg:

1. változat: A vírus egyetlen alkalommal jelentkezik be, és azután többet nem lép akcióba.

2. változat: A vírus szabálytalan időközökben újra és újra bejelentkezik.

Mivel a második változat sokkal elterjedtebb, ajánlatos azt (és minden más vírust) azonnal a bejelentkezése után eltávolítani. Használjuk ehhez a 89/5. számban adott vírusgyilkos programot.

Nézzük meg közelebbről a BHP vírus szaporodási mechanizmusát. Az aktiválás után a vírus számos olyan tárolócella értékét változtatja meg, amelyek programrészekre irányulnak. Ezek úgynevezett operációs rendszer vektorok (LOAD, SAVE, Break, NMI, BASIC melegindítás).

Ezeknek a tárolócelláknak a megváltoztatásával az összes művelet ellenőrzik a vírusban lévő rutinok, s csak ez után hajtnak végre valójában. A vírus



tehát megszerezte magának a gépünket és titokban dolgozik is. Ha floppy LOAD és SAVE műveleteket hajtunk végre, akkor elindítjuk a vírus akciórészét és az szaporodni kezd.

A BHP vírus előbb kikeres magának 9 szabad blokkot a lemezen, bemásolja magát oda, s a korábban említett módon átalakítja a directory mutatókat úgy, hogy azok szépen órá irányuljanak. A vírus osztdott... A kilenc fölhasznált blokk a directoryban nem jelenik meg, az tehát egyszerűen „eltűnt”.

Egy olyan vírust programozni, amelyik eszerint a séma szerint dolgozik, igazából nem nehéz. Alfred Poschmann, a 64'er Magazin szerkesztője szerint „ezt bárki meg tudja tenni, aki két-három éve programoz Assemblerben a C64-esen.” E vírusban a gonoszabb kortársaink, akik maguk nem képesek ilyet készíteni, átírhatják az akciórészét, mondjuk egy lemezformátálásra. Egy ismert számítógépes magazin nemrég vírusprogramozási tanfolyamot hirdetett meg a C64-esre. Nagyon reméljük, hogy ezeket csak jó cél érdekében használják, és nem születnek emiatt új, veszélyes vírusfajták.

## SZÁMÍTÓGÉPES BŰNÖZÉS VÍRUSOKKAL

A PC-k világában még ennél is tovább mennek. Létezik egy úgynevezett „Vírus Konstruktív Készlet” Nightmare Software (Rémálom Szoftver) néven. Ennek a készletnek a célja megismerkedni a vírusokkal, hogy veszély esetén cselekedni lehessen. A saját fejlesztésű vírusokhoz készített leírásokkal — egy specifikus vírusgyilkos programmal — „bárki föl tud ismerni egy vírust és tudni fogja, mit tegyen az ellen”, mondta Ralf Burger adatbiztonsági szakspecialista az AP (Associated Press) hírügynökségnek adott interjújában 1988 márciusában. A Nightmare Software-rel kapcsolatban figyelmeztetés is él: „Ez a program nem játékszer! A vele való munka nem veszélytelen.” Ezenkívül nem vállalnak felelősséget. A jogi védelem érdekében a vevőnek alá kell írnia egy papírt, hogy tisztában van a német büntető törvénykönyv 303a és 303b paragrafusával.

Ugyanis az adatok jogellenes manipulációját és a számítógépes szabotázszt akár öt évig is terjedő szabadságvesztéssel sújthatják.

A vírusokat néha politikai célokra is használják. Néhány évvel ezelőtt az USA-beli Delaware Egyetemet egy vírussal zsarolták meg. A vírus a bejelentkezésekor 2000 dollárt kért, amelyet egy pakisztáni számlára kellett átutalni. Az ígéret szerint ebben az esetben az egyetem egy víruskiller programot kapott volna a fertőzött adatállomány „fertőtlenítésére” (Süddeutsche Zeitung 1988. június). Vajon a vírusok fenyegetése a 90-es évek bűnözését vetíti elének? A Wiesbadeni Bundeskriminalamt (BKA, Szövetségi Bünyügyi Hivatal) információi szerint 1987-ben és 1988-ban nem jelentettek „vírusos” számítógépes bűnyügyi esetet az NSZK-ban. De hogy a vírusok ennek ellenére nagy kárt okozhatnak, azt egy USA-beli eset bizonyítja: Egy egyetemista az ország hatezer igen fontos számítógépét bénította meg, köztük a NASA és a védelmi minisztérium gépeit is.

A vírusoknak az ipari kémkedésben is nagy szerep jut. De ezektől a manipulációktól a normál felhasználó — hála Istennek — távol van. A C64-es tulajdonosokat egyelőre csak a BHP vírus bejelentkezései zavarhatják. De nagyon kell vigyázni, elméletileg olyan vírusok is létezhetnek, amelyek törlik vagy meghamisítják az adatokat.

## HOGYAN SZABADULHATOK MEG A VÍRUSOKTÓL?

Némely vírushoz készült speciális vírusgyilkos program. A BHP vírust például a májusi számban közreadott vírusgyilkossal lehet megsemmisíteni. Sajnos, azonban nincs minden fertőzés esetére alkalmas „fertőtlenítő”, hiszen a vírusok saját jellemzőkkel és mintával rendelkeznek. És azok között az ismert vírusgyilkosok között is vannak olyanok, amelyek nem száz százalékosak.

Biztos módszer csak egy van: Az adott lemez formátálása. Ennek a hátránya persze az, hogy ezzel nemcsak a vírust, de a lemezen álló összes adatot és programot is megsemmisítjük.

## HOGYAN LEHET VÍRUSOKAT „BESZEREZNI”?

Főleg az úgynevezett kalózmásolatok fertőzöttek. További potenciális fertőzési lehetőség van a távközlésen keresztül (TAF) beszerzett és a Public Domain programokban. A nagy cserevásárok is jobb elkerülni, mert ott egy vírus szélsébeesen elterjedhet. De sajnos egy csak eredeti gyári gyári szoftvereket felhasználó sem lehet száz százalékos biztonságban. Így például 10 000 (tízezer) darab Atari ST-re írt „GFA BASIC 3.0” programlemez fertőződött egy vírussal, amelyek közül 1500-at el is adtak, adta hírül a C'T számítógépes magazin a 88/7. számában. Ebben az esetben a gyártó felel. Alapvetően el kell kerülni az idegen lemezek használatát. Gondoljunk arra, hogy azok vírusokat rejthetnek!

## VÉDEKEZHETEK? HOGYAN

A vírusok ellen általános védelem és tanács tulajdonképpen nincs. Minden lemez, amelyet nem a saját kezünkkel formáltunk, lehet fertőzött. Éppen ezért csak saját lemezekkel dolgozzunk. A legjobb, ha ezeket elzárjuk, ha nem használjuk őket vagy hosszabb ideig távol vagyunk.

A napi munkáról készített biztonsági másolatok csökkentik a fertőzés veszélyét. Soha ne foglalkozunk kalózmásolatokkal, mert ezek igen gyakran „fertőzöttek”.

Senki ne dolgozhasson felügyelet nélkül a számítógépünkkel. Számos ember talál élvezetet abban, hogy embertársai lemezei közé vírust csempész.

## A TÖRVÉNY VÉDI AZ ADATOKAT

Az NSZK parlamentje 1986-ban megszavazott egy törvényt a számítógépes rendszerek és adatállományok védelmére: Az idegen számítógépes rendszerbe való behatolás és az adatokkal való manipuláció büntethető.

A törvény vonatkozik mind a programozó által közvetlenül végrehajtott manipulációkra, mind a vírusokkal végrehajtott

közvetett szabotázsra. Ugyanakkor rendkívül nehéz egy gyilkos vírus szerzőjét megtalálni. A keresés hasonlatos a „tű és a szénakazal” ismert példájához.

Hogy a károkat megelőzzék, hat USA-beli szoftveres cég megalapította a „Software Development Council”-t (SDC, szoftveres fejlesztő egyesület). Michael Odawa, az alapítók egyike állást foglalt a vírusokkal kapcsolatban: „A számítógépes vírus nem tréfadolog. Egy számítógépes vírussal nincsen semmilyen jó oldala. Ha egy vírus behatol egy kórház számítógépes rendszerébe, akár egy páciens halálát is okozhatja, a szükséges gyógyszer megkeverésével, vagy hogy ő nem a megfelelő műtőasztalra kerül.” Az ilyesmiket meg kell akadályozni. Az első lépés a törvényalkotásban még nyitott kérdések lezárása. Az SDC három tervet javasol: 1. 1000 dollár büntetés, vagy három hónapi elzárás egy adatokat károsító vírus fejlesztéséért.

2. 10 000 dollár büntetés és egy év elzárás az olyan adatkárosító vírus kifejlesztéséért, amely a nyilvános segélyszolgálatokat és a biztonságot veszélyezteti. 3. Tíz évi elzárás egy számítógépes vírus bármilyen életveszélyes tevékenységéért. (A Commodore Computing 88/10-es számából.)

## VÍRUS ADTA MATT 6000 SZÁMÍTÓGÉPNEK

1988. november 2-án, szerdán, megrendült az amerikai számítógépes világ. Egy vírus 6000 számítógépet bénított meg. Ezek között megtalálhatók voltak az ország legfontosabb rendszerei is. Számos cég, egyetem és laboratórium termináljai merevedtek le, azok vagy nem reagáltak a bevitelre, vagy azokra értelmetlen válaszokat adtak. Még az US-Navy (haditengerészet) parancsnoki központját, a Lawrence Livermore Laboratories-t (ez az amerikai atom- és SDI fegyvereinek fejlesztőhelye) és a NASA-t sem hagyta érintetlenül a vírus.

A vírus fejlesztőjét azóta ismerjük már. Robert Morris Jr., aki doktori disszertációját írja a Cornell Egyetemen (New York állam), a számítógépes törté-



nelemben a hatását tekintve eddigi legnagyobb kárt okozó programjának szerzője. Ő rakta be a vírust az ARPANET hálózatba, amely mintegy 50 000 számítógépet foglal magába. Morris ezzel azt akarta bizonyítani, hogy egy vírus képes a legjobban védett számítógépes hálózatba is bejutni és ott észrevétlenül dolgozni.

Egy beviteli hiba okozta azt, hogy a vírus „ahelyett, hogy egy rendszerben csak egyszer szaporodjon, ott állandóan reprodukálta magát”. A felhasználók ezzel a vírust az egész országban szétszórták. Mivel a fertőzés exponenciálisan növekedett, a nagyszámítógépek kapacitása is hamar kimerült. Ez okozta a rendszerösszeomlást. Ennek ellenére a vírus nem volt szándékosan rosszindulatúra programozva, mert minden adat változatlan maradt. Dean Kraft, a Cornell Egyetem informatikusa megerősítette a Spiegelnek adott interjújában „Morris jó szándékait”. „Számára borzasztóan

egyszerű lett volna igazi károkat okozni. Akár egy-két plusz sor is gondoskodhatott volna arról, hogy mindenhol minden tároló törlődjön.”

## INFORMÁCIÓK A PROFIKNAK

Amikor egy BHP vírussal fertőzött programot betöltünk, a mellékelt tárolókonfiguráció áll elő. LIST kiadása után (ezt azonban nem nagyon szokták megtenni) a vírus hírt ad magáról:

```
1987 SYS PEEK(43)+256x
PEEK(44)+48: VÍRUS
```

Ez azonban akkor már nem igaz, ha a vírus már aktív állapotban van a tárolóban.

Ha most a programot RUN-nal elindítjuk, a SYS parancs kiszámítva relatív módon elindítja a vírust. (A 2097-es címen, ha a BASIC start a szokott 2048-as címnél áll.) A SYS utasítással elindít egy gépi kódú programot, azaz itt a BHP vírust. A főnti „relatív” szó azt



jelenti, hogy a program kezdőcíme független a BASIC terület kezdőcímétől.

A vírus további lépéseit lentebb folyamatábrán mutatjuk be. Eközben egy olyan tárolókonfiguráció áll elő, mint amit az 1-es ábra mutat:

A vírus az I/O tartomány alatti RAM-ba másolja magát az 53248-as (SD000) címre. Ez a tartomány „kívülről” igen nehezen elérhető, a BASIC 2.0 számára pedig egyáltalán nem hozzáférhető. Ezután a gonoszság eltolja a felhasználói programot annak eredeti helyére,

a 32768-as címen (S8000) létrehozva a resetvédelmet (CBM 80 jelzés). Ahhoz tehát, hogy a vírust eltávolítsuk a környezetünkől, meg kell vonni tőle az életelemet. Azaz a számítógépet mindig kapcsoljuk ki!

Az 1-es ábrán fönt láthatjuk a fertőzött program betöltésekor előálló tárolókonfigurációt. Az egyetlen különbség a normál állapothoz képest, hogy a felhasználói program kb. 2300 byte-tal arrébb lett tolvá. Az így keletkező „üres” területen áll a vírus, amelyet a RUN beadásával azonnal életre hívunk.

1. ábra. A BHP vírus által előidézett tárolófoglaltság

		programkód						
	1	2	7	3	4	6	6	
0	1024	2048	32768	40960	49152	53248	57344	65535

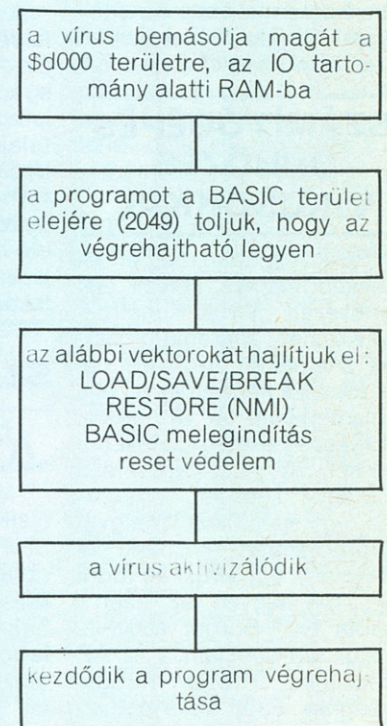
1. nulláslap
2. képernyő
3. BASIC interpreter
4. szabad tárolóterület
5. I/O tartomány; ez alatt a vírus
6. kernel/operációs rendszer
7. a reset védelem

		vírus programkód		I/O			
	1	2	3	4	tartomány	5	
0	1024	2048 kb.	4300	40960	49152	53248	57344 65535

1. nulláslap, stack, vektorok
2. képernyő
3. BASIC interpreter
4. szabad tárolóterület
5. kernel/operációs rendszer

Egy fertőzött program betöltésekor először minden rendben van: A programot kb. 2300 byte-tal eltoltuk, a még nem aktivizált vírus az eredeti BASIC terület elején áll. Egy aktív vírussal a program a normál területen áll, a vírus pedig az I/O tartomány alatt van. A reset védelmet a 32768-as címen (\$8000) találhatjuk.

2. ábra: A BHP vírus működésének folyamatábrája



A BHP vírus inicializálásakor az először magát helyezi biztonságba (bemásolja magát a \$D000 „alá”). Ezután megkísérli a normál állapotot szimulálni. A rendszer ellenőrzését azonban a vírus természetesen magánál tartja.





## 80 karakteres szövegszerkesztő

A program laponkénti levél megírására alkalmas. A képernyőn írás közben egy sor max. 80 karaktert tartalmazhat, úgy, mint nyomtatáskor. A sorhosszúság a program elején, és írás közben is állítható. Ha 80 karakternél kevesebbet kívánunk megjeleníteni (margó), akkor a kiírt szöveg ennek megfelelően középre rendeződik. Egy levéloldal max. 63 soros lehet. Ebből a képernyőn egyszerre csak 21 sor látható, így egy nyomtatási oldal három képernyőre oszlik. Bár a program a grafikus képernyőt használja, a gépi kódú átpakolás gyorsan mozgatja a képernyőket. A program sokoldalú, jól kezelhető, amint az az alábbiakból is kiténik.

### Kurzormozgatás

Oldalra léptetéskor a kurzor a lapszéleken nem ugrik át új sorba, de laptetőhöz, vagy a lap aljához érve a képernyő előrehátra lapozódik. Beírásnál a margónak megfelelően lép új sorba a kurzor.

- F2: a sor elejére állítja a kurzort
- F5: a sor végére állítja a kurzort
- F3: kurzor jobbra 10 karakterenként
- F6: kurzor balra 10 karakterenként
- F7: kurzor fel 5 karakterenként
- F8: kurzor le 5 karakterenként
- RETURN: ugrás új sor elejére
- Képernyő lapozása
- F1: lapozás előre
- F4: lapozás hátra
- Törlés — beszúrás:

Az eredeti funkció szerint [INST—DEL gomb]. (Csak az adott sort csúsztatja!)

### Szövegmozgatás

- ESC+kurzor jobbra: a teljes szöveg 2 karakterrel jobbra lép
- ESC+kurzor balra: a teljes szöveg 2 karakterrel balra lép
- ESC+ > vagy, bill.: a teljes szöveg 10 karakterrel jobbra lép
- ESC+ < vagy, bill.: a teljes szöveg 10 karakterrel balra lép
- ESC+ kurzor fel: a teljes szöveg az adott ponttól egy sorral felfelé tolódik
- ESC+ kurzor le: a teljes szöveg az adott ponttól egy sorral lefelé tolódik

### Egyéb ESC funkciók

- ESC+F8: a programkezelő segédtabla képernyőre íratása
- ESC+ s: a 63 soros lapoldal adatainak kimentése
- ESC+ 1: a szöveg adatainak visszatöltése
- ESC+ p: a szöveg kinyomtatása (A program mind a képernyő, mind a nyomtató esetén csak a magyar ékezetes kisbetűket használja normál, nem NLQ üzemmódban.)
- ESC+ a: a szöveg visszaírása a grafikus képernyőre (Ellenőrzésre használható. STOP esetén GRAPHICi:GOTO360 beütése után a program folytatható.)
- ESC+ r: a program újraindítása teljes törléssel
- ESC+ h: sorhosszúság újbóli beállítása

Megjegyzés: Ha valaki a programot magnóval kívánja használni, akkor a név után álló ,8-at át kell írni ,1-re a mentés-beolvasás programsoraiban.

### Programlisták

1. betöltő és indító segédprogram
2. BASIC program
3. MONITOR gépi kódú lista:

\$1000—\$13FF új karakterkészlet  
\$1400—\$1572 átpakoló rutinok

A programot kis képernyős tv-n vagy monitoron célszerű futtatni, mert a nagy méretű képernyőn a karakterek nehezebben kivehetőek.

A leírás is ezzel a szövegszerkesztővel készült.

Lőrentei János

```

1 REM *****
2 REM *      C= UJSAG          SORSZAM:171 *
3 REM *      SZOVEGSSZERK. INDITO PLUS/4 *
4 REM *      PROGRAM: LORENTEI JANOS *
5 REM *****
10 COLOR4,1:COLOR0,1:COLOR1,2
20 KEY1,"LOAD"+CHR$(34)+"SZ.SZ.GEPI"+CHR$(34)+" ,8,8"+CHR$(13)+"NEW"+CHR$(13)
21 KEY2,"LOAD"+CHR$(34)+"SZ.SZ.BAS."+CHR$(34)+" ,8"+CHR$(13)+"RUN"+CHR$(13)
25 PRINT"LOAD - RUN:  F1 + F2"
30 POKEDEC("2C"),DEC("B0")
40 POKEDEC("B000"),0
50 NEW
READY.
    
```





```

; ff00 00 00 ff 00 f1
>1000 00 00 00 00 00 00 00 00 : .....
>1008 00 00 e0 20 e0 a0 e0 00 : - - - -
>1010 80 80 e0 a0 a0 a0 e0 00 : - - - -
>1018 00 00 e0 80 80 80 e0 00 : - - - -
>1020 20 20 e0 a0 a0 a0 e0 00 : - - - -
>1028 00 00 e0 a0 e0 80 e0 00 : - - - -
>1030 00 c0 a0 80 c0 80 80 00 : @ @ . .
>1038 00 00 e0 a0 a0 e0 20 e0 : - - - -
>1040 80 80 e0 a0 a0 a0 a0 00 : - - - -
>1048 40 00 c0 40 40 40 e0 00 : @ @ @ @
>1050 20 00 60 20 20 20 20 e0 : - - - -
>1058 80 80 a0 a0 c0 a0 a0 00 : @ @ . .
>1060 40 40 40 40 40 40 60 00 : @ @ @ @ @
>1068 00 00 a0 e0 a0 a0 a0 00 : - - - -
>1070 00 00 c0 a0 a0 a0 a0 00 : @ . . .
>1078 00 00 e0 a0 a0 e0 00 00 : - - - -
>1080 00 00 e0 a0 a0 e0 80 80 : - - - -
>1088 00 00 e0 a0 a0 e0 20 20 : - - - -
>1090 00 00 e0 a0 80 80 80 00 : - - - -
>1098 00 00 e0 80 e0 20 e0 00 : - - - -
>10a0 40 40 e0 40 40 40 60 00 : @ @ - @ @
>10a8 00 00 a0 a0 a0 a0 e0 00 : - - - -
>10b0 00 00 a0 a0 a0 40 40 00 : . . @ @
>10b8 00 00 a0 a0 a0 e0 a0 00 : - - - -
>10c0 00 00 a0 a0 40 a0 a0 00 : @ . . .
>10c8 00 00 a0 a0 a0 e0 20 e0 : - - - -
>10d0 00 00 e0 20 40 80 e0 00 : - - @ .
>10d8 e0 80 80 80 80 80 e0 00 : - - - -
>10e0 00 00 00 00 00 00 00 00 : - - - -
>10e8 e0 20 20 20 20 20 e0 00 : - - - -
>10f0 00 00 00 00 00 00 00 00 : - - - -
>10f8 00 20 40 f0 40 20 00 00 : @ P @ .
>1100 00 00 00 00 00 00 00 00 : - - - -
>1108 40 40 40 40 00 00 40 00 : @ @ @ @ . @
>1110 a0 a0 a0 00 00 00 00 00 : - - - -
>1118 00 50 f0 50 50 f0 50 00 : p P p P p P
>1120 40 e0 80 e0 20 e0 40 00 : @ - - - @
>1128 a0 a0 20 40 40 80 a0 a0 : @ @ . .
>1130 00 30 50 60 b0 a0 d0 00 : O P - O P
>1138 40 40 80 00 00 00 00 00 : @ @ . . . .
>1140 40 40 80 80 80 40 40 00 : @ @ . . @ @
>1148 40 40 20 20 20 40 40 00 : @ @ @ @
>1150 00 00 00 00 40 00 00 00 : - - @ @
>1158 00 40 40 e0 40 40 00 00 : @ @ - @ @
>1160 00 00 00 00 40 40 80 00 : - - - @ @
>1168 00 00 00 e0 00 00 00 00 : - - - -
>1170 00 00 00 00 00 40 40 00 : - - - @ @
>1178 10 20 20 40 40 80 80 00 : @ @ . .
>1180 40 a0 a0 a0 a0 a0 40 00 : @ @ . .
>1188 40 c0 40 40 40 40 e0 00 : @ @ @ @ @ @
>1190 e0 20 20 e0 80 80 e0 00 : - - - -
>1198 e0 20 20 c0 20 20 e0 00 : @ . . .
>11a0 80 80 80 a0 e0 20 20 00 : - - - -
>11a8 e0 80 80 e0 20 20 e0 00 : - - - -
>11b0 e0 80 80 e0 a0 a0 e0 00 : - - - -
>11b8 e0 20 20 40 40 40 40 00 : - - @ @ @ @
>11c0 e0 a0 e0 a0 a0 a0 e0 00 : - - - -
>11c8 e0 a0 a0 e0 20 20 e0 00 : - - - -
>11d0 00 00 40 00 00 40 00 00 : @ . @ .
>11d8 00 00 40 00 00 40 80 00 : @ . @ .
>11e0 00 20 40 80 80 40 20 00 : @ . @ .
>11e8 00 00 e0 00 e0 00 00 00 : - - - -
>11f0 00 80 40 20 20 40 80 00 : @ @ . .
>11f8 80 90 10 20 40 00 40 00 : - - @ @
>1200 00 00 00 f0 00 00 00 00 : - - P . . .
>1208 e0 a0 a0 a0 e0 a0 a0 00 : - - - -
>1210 c0 a0 a0 c0 a0 a0 c0 00 : @ @ @ @
>1218 e0 80 80 80 80 80 e0 00 : - - - -
>1220 c0 a0 a0 a0 a0 c0 00 : @ @ . @
>1228 e0 80 80 e0 80 80 e0 00 : - - - -
>1230 e0 80 80 e0 80 80 80 00 : - - - -
>1238 e0 80 80 a0 a0 a0 e0 00 : - - - -
>1240 a0 a0 a0 e0 a0 a0 00 : - - - -
>1248 e0 40 40 40 40 40 e0 00 : @ @ @ @ @

```

```

>1250 60 20 20 20 20 20 c0 00 : - @ . .
>1258 a0 a0 a0 c0 a0 a0 a0 00 : @ . . .
>1260 80 80 80 80 80 80 e0 00 : - - - -
>1268 a0 e0 e0 a0 a0 a0 a0 00 : - - - -
>1270 90 90 d0 d0 b0 90 90 00 : . . p p 0 . .
>1278 e0 a0 a0 a0 a0 a0 e0 00 : - - - -
>1280 e0 a0 a0 e0 80 80 80 00 : - - - -
>1288 60 90 90 90 90 a0 50 00 : - . . . . p .
>1290 e0 a0 a0 c0 a0 a0 a0 00 : @ . . .
>1298 e0 80 80 e0 20 20 e0 00 : - - - -
>12a0 e0 40 40 40 40 40 40 00 : @ @ @ @ @ @
>12a8 a0 a0 a0 a0 a0 a0 d0 00 : - . . . p .
>12b0 a0 a0 a0 a0 a0 40 40 00 : @ @ . .
>12b8 a0 a0 a0 a0 e0 e0 a0 00 : - - - -
>12c0 a0 a0 40 40 40 40 a0 00 : @ @ @ @
>12c8 a0 a0 a0 40 40 40 40 00 : @ @ @ @
>12d0 e0 20 20 40 40 80 e0 00 : @ @ - .
>12d8 40 40 40 f0 40 40 40 40 : @ @ @ P @ @ @ @
>12e0 80 40 80 40 80 40 80 40 : @ @ . @ . @ . @
>12e8 40 40 40 40 40 40 40 40 : @ @ @ @ @ @ @ @
>12f0 00 70 90 50 50 50 90 00 : . P . p p p . .
>12f8 10 10 30 30 70 70 f0 f0 : . . O O P P P P
>1300 00 00 00 00 00 00 00 00 : - - - -
>1308 20 40 00 c0 40 40 60 00 : @ . @ @ @ .
>1310 00 00 00 00 f0 f0 f0 f0 : - . . . P P P P
>1318 f0 00 00 00 00 00 00 00 : P . . . . .
>1320 00 00 00 00 00 00 00 f0 : - - - - P
>1328 50 a0 00 e0 a0 a0 e0 00 : p . - - -
>1330 a0 50 a0 50 a0 50 a0 50 : p p p p p
>1338 80 80 40 40 20 20 10 10 : - . @ @ . .
>1340 00 00 00 00 00 00 00 00 : - - - -
>1348 f0 f0 e0 e0 c0 c0 80 80 : P P - @ @ .
>1350 10 10 20 20 40 40 80 80 : - . @ @ . .
>1358 c0 c0 c0 f0 f0 c0 c0 c0 : @ @ @ P P @ @ @
>1360 10 20 e0 a0 e0 80 e0 00 : - - - -
>1368 40 40 40 70 70 00 00 00 : @ @ @ P P .
>1370 00 a0 00 a0 a0 a0 e0 00 : - - - -
>1378 00 00 00 00 00 00 f0 f0 : - . . . . P P
>1380 30 40 e0 20 e0 a0 e0 00 : O @ - - -
>1388 60 60 60 f0 f0 00 00 00 : - - P P . .
>1390 00 00 00 f0 f0 60 60 60 : - . . P P .
>1398 30 30 30 f0 f0 30 30 30 : O O O P P O O O
>13a0 00 a0 00 e0 a0 a0 e0 00 : - - - -
>13a8 40 40 00 a0 a0 a0 e0 00 : @ @ . .
>13b0 20 20 40 e0 a0 a0 e0 00 : @ - - -
>13b8 f0 f0 00 00 00 00 00 00 : P P . . . .
>13c0 f0 f0 f0 00 00 00 00 00 : P P P . . .
>13c8 00 00 00 00 00 f0 f0 f0 : - . . . . P P P
>13d0 00 00 00 00 00 00 00 00 : - - - -
>13d8 50 a0 00 a0 a0 a0 e0 00 : p . - -
>13e0 30 30 30 00 00 00 00 00 : O O O . . .
>13e8 20 20 20 e0 e0 00 00 00 : - - - -
>13f0 c0 c0 c0 00 00 00 00 00 : @ @ . . . .
>13f8 00 00 00 00 00 00 00 00 : - - - -
>1400 78 8d 3f ff b1 5a 91 58 : A . ? 1 z . x
>1408 c8 d0 f9 e6 5b e6 59 b1 : h o y h n y l
>1410 5a 91 58 ca d0 f2 8d 3e : z . x j p k . >
>1418 ff 58 60 a5 5a 91 58 c8 : x - % z . x h
>1420 d0 f9 e6 59 a5 5a 91 58 : p y h z a . x
>1428 ca d0 f2 60 e6 25 a5 00 : j p k - h % %
>1430 a2 00 bd 00 11 9d 00 17 : - . - . . .
>1438 e8 e0 08 d0 f5 a2 00 bd : h - p u . =
>1440 00 17 4a 4a 4a 4a 9d 00 : - . . . .
>1448 17 e8 e0 08 d0 f1 a2 00 : h - p q .
>1450 bd 00 11 1d 00 17 9d 00 : - . . . .
>1458 17 e8 e0 08 d0 f2 60 00 : h - p k .
>1460 00 a9 10 8d 34 14 8d 52 : . ) . 4 . r
>1468 14 a9 00 8d 33 14 8d 51 : . ) . 3 . q
>1470 14 8d 3f ff a2 00 bd 00 : - . ? . = .
>1478 a4 8d da 14 20 d9 14 ad : $ . z . y .
>1480 05 15 8d 10 17 e8 bd 00 : - . . . . H =
>1488 a4 ca 8d da 14 20 d9 14 : $ j . z . y .
>1490 ad 05 15 8d 11 17 8d 3e : - . . . . >
>1498 ff ad 11 17 48 a9 00 85 : - . . . . h .
>14a0 7e 68 0a 26 7e 0a 0a 26 : H . & . & . &
>14a8 7e 8d 33 14 a5 7e 6d 34 : * 3 * * * 4
>14b0 14 8d 34 14 ad 10 17 48 : 4 . - . h

```





```

>14b8 a9 00 85 7e 68 0a 26 7e :).H.
>14c0 0a 0a 26 7e 8d 51 14 a5 :.g.
>14c8 7e 6d 52 14 8d 52 14 20 :Mr.r.
>14d0 30 14 20 08 15 60 00 00 :O.-.
>14d8 00 a9 20 c9 a1 b0 0d c9 :)i!0.1
>14e0 60 b0 10 c9 40 b0 13 c9 :.i@0.1
>14e8 20 b0 12 60 38 e9 00 8d :O.-81.
>14f0 05 15 60 38 e9 80 8d 05 :.-81.
>14f8 15 60 38 e9 40 8d 05 15 :.-81@.
>1500 60 20 d9 14 00 20 00 00 :.y.
>1508 a2 00 bd 08 17 8d 00 90 :".=.....
>1510 20 19 15 e8 e0 08 d0 f2 :.H.-.p
>1518 60 a9 01 18 6d 0e 15 8d :).M.
>1520 0e 15 90 03 ee 0f 15 60 :.N.
>1528 00 00 8d 3f ff a9 00 8d :.W.
>1530 0e 15 a9 40 8d 0f 15 a9 :.W.
>1538 00 8d 77 14 8d 87 14 a9 :.W.
>1540 90 8d 78 14 8d 88 14 78 :.X.X
>1548 20 61 14 a2 00 a9 01 18 :.A.
>1550 6d 77 14 8d 77 14 8d 87 :M.W.
>1558 14 90 06 ee 78 14 ee 88 :.N.N.
>1560 14 e8 e0 02 d0 e7 ae 78 :.H.-.p.g.X
>1568 14 e0 a4 d0 da 8d 3e ff :.p.p.
>1570 58 60 00 00 00 00 00 :X.....
>1578 00 00 00 00 00 00 00 :.....

```

```

10 key1, chr$(4): key4, chr$(5)
20 key2, chr$(6): key5, chr$(7)
30 key3, chr$(8): key6, chr$(9)
40 key7, chr$(10): key8, chr$(11)
50 poke117,255: goto200
60 sshapeis$,x,y,x +3,y +7: return
70 sshapejs$,xx,yy,xx +3,yy +7: return
80 gshapeis$,x,y,1: return
90 gshapejs$,xx,yy,1: return
100 hh =th *21:c1 =t1:c2 =c1 +hh:cc =(1
-2) *th *21 +t2: gosub1190: return
110 hh =th *21:c1 =(1 -2) *th *21 +t2:c2
=c1 +hh:cc =t1: return
120 cs =y /kh -4:c% =x /8:co =c% *8:
return
130 print chr$(147)
140 char1,3,4,"80 karakteres szovegszer
keszto"
150 char1,5,10,"adja meg a sorhosszusag
ot!"
160 char1,5,11,"30 - 80 karakter "
170 print: print: inputsh
180 ifsh <30 orsh >80 goto130
190 ko =(80 -sh) /2:kk% =ko:k% =79 -ko:
return
200 color0,1: color4,1: color1,2
210 gosub130: graphic1,1: char1,5,2,"80
karakteres sz[C= H]vegszerkeszt
[C= G]"
220 et =0:hh =20480:cc =16384: gosub1370
230 et =32:hh =8192:cc =36864: gosub1370
240 poke740,16
250 l =2:kh =8:ka =32
260 tr = dec("9000")
270 draw1,0,28 to320,28
280 forz =0 to320 step40
290 draw1,z,27 toz,29: next
300 draw1,319,27 to319,29
310 char1,37,2,"-" + right$( str$(1 -1)
,1) +""
320 t1 = dec("2500"):th =320
330 t2 = dec("4000")
340 b$ = " ":k =4:x =kk% *4:y =ka
350 box,x,32,x +3,39,1,1
360 ifaa$ = "e" then forz =1020 to60
step -100: vol8: sound1,z,1: next:a
a$ = ""

```

```

370 xx =x:yy =y:vs% =0: get keya$:a =
asc(a$)
380 vol3: ifx <k% *4 -32 then sound1,80
0,2: goto400
390 vol8: sound1,900,2
400 ifa >31 anda <128 goto600
410 ifa =29 thenu =1: goto700: remjobb
420 ifa =157 thenu =1: goto730: rembal
430 ifa =145 thenu =1: goto760: remle
440 ifa =17 thenu =1: goto790: remfel
450 ifa =20 goto1780
460 ifa =148 goto1890
470 ifa =13 thenx =kk% *4:y =y +kh:
goto1020
480 ifa =27 goto1040: remescfunkciok
490 a =a -3
500 on(a) goto540,550,520,530,560,570,5
80,590
510 goto600
520 goto1000: remsoareleje
530 goto1010: remsorvege
540 goto830: replapozelore
550 goto930: replapozvissza
560 u =10: goto700: remjobb
570 u =10: goto730: rembal
580 u =5: goto760: remle
590 u =5: goto790: remfel
600 char0,0,0,a$
610 sshapes$,0,0,3,7
620 gshapes$,x,y
630 sshapeis$,x +k,y,x +k +3,y +7
640 a =x /4:b =y /kh -4
650 x =x +k: ifx >4 *k% thenx =kk% *4:y
=y +kh
660 gshapeis$,x,y,1
670 poketr +a +1680 *(1 -2) +b *80, asc
(a$)
680 ify >200 -8 goto850
690 goto360
700 xx =x:x =x +k *u:yy =y
710 ifx >318 thenx =x -k *u: goto360
720 goto810
730 xx =x:x =x -k *u:yy =y
740 ifx <0 thenx =x +k *u: goto360
750 goto810
760 ify = >kh +ka thenxx =x:yy =y:y =y
-kh *u: goto810
770 ify = <ka thenxx =x:yy =y: ifl >1
goto930
780 goto810
790 ify <kh *20 +ka thenxx =x:yy =y:y =
y +kh *u: goto810
800 ify = >kh *19 +ka thenxx =x:yy =y:
ifl <5 goto830
810 gosub60: gosub70: gosub80: gosub90
820 goto360
830 remattoltes1
840 gosub70: gosub90
850 gosub100
860 ifvs% =1 then gosub60: gosub80:
return
870 l =l +1
880 ifl <5 thenhh =th *21:c1 =(1 -2) *t
h *21 +t2:cc =t1:c2 =c1 +hh: gosub11
90
890 ifl =5 thenl =4
900 char1,37,2,"-" + right$( str$(1 -1)
,1) +""
910 y =ka: gosub60: gosub80: goto360
920 vs% =1: goto830
930 remattoltes2
940 gosub70: gosub90
950 gosub100:1 =1 -1
960 ifl >1 thenhh =th *21:c1 =(1 -2) *t
h *21 +t2:c2 =c1 +hh:cc =t1: gosub11
90
970 ifl =1 thenl =2

```







```
○ 980 char1,37,2,"-" + right$( str$(1 -1)
,1) + "-" ○
990 y =ka: gosub60: gosub80: goto360
○ 1000 x =kk% *4: goto1020 ○
1010 x =4 *k%: goto1020 ○
1020 gosub60: gosub70: gosub80: gosub90
○ 1030 goto360 ○
1040 aa$ = "e": geta$: ifa$ = "" goto1040 ○
1050 remcsusztatas ○
○ 1060 ifa$ = chr$(157) theneg =8:ek =2: ○
goto1440 ○
○ 1070 ifa$ = chr$(29) theneg =8:ek =2: ○
goto1600 ○
1080 ifa$ = "<" ora$ = "," theneg =40:ek
=10: goto1440: rembalra ○
○ 1090 ifa$ = ">" ora$ = "." theneg =40:ek
=10: goto1600: remjobbra ○
○ 1100 ify >32 then ifa$ = chr$(145) then ○
eg =320:ek =80: goto1440 ○
○ 1110 ify <192 then ifa$ = chr$(17) then ○
eg =320:ek =80: goto1600 ○
1120 ifa$ = chr$(11) goto2230: remhelp
○ 1130 ifa$ = "l" goto2010: remload ○
1140 ifa$ = "s" goto2120: remsave ○
1150 ifa$ = "p" goto2550: remnyomtatas ○
○ 1160 ifa$ = "h" then gosub920: graphic0: ○
gosub130: graphic1: remsorhossz ○
1170 ifa$ = "a" then goto2520: remgraf.v ○
isszairas ○
1180 goto360 ○
1190 ll =c2 -c1:en =ll +cc ○
○ 1200 a% =ll /256:a =ll -256 *a% ○
1210 b% =(en -a) /256:b =en -256 *b% -a ○
○ 1220 c% =(c2 -a) /256:c =c2 -256 *c% -a ○
1230 poke2035,a% +1: poke2036,a ○
1240 poke91,c%: poke90,c ○
○ 1250 poke89,b%: poke88,b ○
1260 ifa =0 then sys35069: return ○
1270 sys35061: return ○
○ 1280 ll =c2 -c1 ○
1290 a% =ll /256:a =256 -ll +256 *a% ○
○ 1300 ifa =256 thena =0:a% =a% -1 ○
1310 b% =(cc -a) /256:b =cc -256 *b% -a ○
○ 1320 c% =(c1 -a) /256:c =c1 -256 *c% -a ○
○ 1330 poke2035,a% +1: poke2036,a ○
1340 poke91,c%: poke90,c ○
1350 poke89,b%: poke88,b ○
○ 1360 sys5120: return ○
○ 1370 ll =hh -1:a% =ll /256:a =256 -ll +2
56 *a% ○
○ 1380 ifa =256 thena =0:a% =a% -1 ○
1390 b% =(cc -a) /256:b =cc -256 *b% -a ○
○ 1400 poke2035,a% +1: poke2036,a ○
1410 poke91,et: poke90,et ○
1420 poke89,b%: poke88,b ○
○ 1430 sys5147: return ○
1440 remcsusztatasbalra ○
1450 gosub120 ○
○ 1460 ifx >315 andcs >21 goto360 ○
1470 sshapejs$,x,y,x +3,y +7 ○
1480 gshapejs$,x,y,1 ○
○ 1490 gosub100 ○
1500 c1 = dec("4000") +320 *cs +co +(1 -
2) *21 *320 ○
○ 1510 c2 = dec("8ec0") ○
1520 cc =c1 -eg: gosub1280 ○
1530 hh =eg:cc =36504:et =0: gosub1370 ○
○ 1540 gosub110: gosub1280 ○
1550 c1 = dec("9000") +80 *cs +80 *21 *(
1 -2) +co /4 ○
○ 1560 c2 = dec("a3b0") ○
1570 hh =c2 -c1:cc =c1 -ek: gosub1280 ○
○ 1580 hh =ek:cc = dec("a3b0"):et =32: ○
gosub1370 ○
1590 gosub60: gosub80: goto360 ○
○ 1600 remcsusztatasjobbra ○
1610 gosub120 ○
```

```
1620 ifx >311 andcs >21 goto360 ○
○ 1630 gosub60: gosub80: gosub100 ○
1640 c1 = dec("4000") +320 *cs +co +(1 -
2) *21 *320 ○
○ 1650 c2 = dec("8ec0") ○
1660 cc =c1 +eg: gosub1190 ○
○ 1670 hh =eg:cc =c1:et =0: gosub1370 ○
1680 gosub110: gosub1190 ○
1690 c1 = dec("9000") +80 *cs +80 *21 *(
1 -2) +co /4 ○
○ 1700 c2 = dec("a3b0") ○
1710 hh =c2 -c1 ○
○ 1720 cc =c1 +ek: gosub1190 ○
1730 hh =ek:cc =c1:et =32: gosub1370 ○
1740 gosub60 ○
○ 1750 gosub80 ○
1760 goto360 ○
○ 1770 rembeszuras ○
1780 ifx <4 goto360 ○
1790 sshapes$,x,y,x +320,y +3 ○
○ 1800 sshapeis$,x,y +3,x +320,y +7 ○
1810 gshapes$,x -k,y ○
1820 gshapeis$,x -k,y +3 ○
○ 1830 co =x /4:cs =y /kh -4:hh =80 -co -1 ○
1840 c1 = dec("9000") +80 *cs +80 *21 *(
1 -2) +co ○
○ 1850 c2 =c1 +hh:cc =c1 -1: gosub1280 ○
1860 pokec2,32 ○
1870 x =x -k: goto360 ○
○ 1880 rembeszuras ○
1890 sshapes$,x,y,x +320,y +3 ○
○ 1900 sshapeis$,x,y +3,x +320,y +7 ○
1910 sshapess$,16,0,19,7 ○
1920 gshapes$,x +k,y ○
○ 1930 gshapeis$,x +k,y +3 ○
1940 gshapess$,x,y ○
1950 co =x /4:cs =y /kh -4:hh =80 -co ○
○ 1960 c1 = dec("9000") +80 *cs +80 *21 *(
1 -2) +co ○
1970 c2 =c1 +hh:cc =c1 +1: gosub1190 ○
○ 1980 pokec1,32 ○
1990 x =x +k ○
○ 2000 goto360 ○
2010 gosub920: graphic0: print chr$(147
): remload ○
○ 2020 char1,5,1,"adatok beolvasasa" ○
2030 char1,5,3,"l: load " ○
2040 char1,5,4,"d: directory " ○
○ 2050 poke239,0: get key1$: print: print ○
2060 ifl$ = "d" then print chr$(147): ○
directory: poke239,0: get keya$: ○
goto2010 ○
2070 ifl$ < >"l" then graphic1: goto360 ○
○ 2080 input" nev";ne$:n$ = chr$(34):r$ = ○
chr$(13) ○
2090 key1,"m0" +r$ +"l" +n$ +ne$ +n$ +
,8" +r$ +"x" +r$ +"gR1:l=2:gO2520" ○
+r$ ○
2100 print: print" f1 : load" ○
○ 2110 end ○
2120 gosub920: graphic0: print chr$(147
): remsave ○
○ 2130 char1,5,1,"adatok kimentese" ○
2140 char1,5,3,"s: save " ○
○ 2150 char1,5,4,"d: directory" ○
2160 poke239,0: get key1$: print: print ○
2170 ifl$ = "d" then print chr$(147): ○
directory: poke239,0: get keya$: ○
goto2120 ○
2180 ifl$ < >"s" then graphic1: goto360 ○
○ 2190 input" nev";ne$:n$ = chr$(34):r$ = ○
chr$(13) ○
2200 key2,"m0" +r$ +"s" +n$ +ne$ +n$ +
,8,9000,a400" +r$ +"x" +r$ +"gR1:gO
2520" +r$ ○
○ 2210 print: print"f2: save" ○
2220 end ○
```



```

○ 2230 gosub920: graphic0: print chr$(147
) : poke239,0
2240 char1,0,1,"funkciobillentyuk
○ 2250 char1,0,3," f1 lapozas elore
2260 char1,0,4," f4 lapozas hatra
○ 2270 char1,0,5," f2 soreleje
2280 char1,0,6," f5 sorvege
○ 2290 char1,0,7," f3 ugras jobbra
2300 char1,0,8," f6 ugras balra
○ 2310 char1,0,9," f7 ugras le
2320 char1,0,10," f8 ugras fel
○ 2330 char1,23,1," esc uzemmodban:"
2340 char1,23,3," f1 load adatok
2350 char1,23,4," f4 save adatok
○ 2360 char1,23,5,"help segitseg
2370 char1,0,12,"_____esc kodbil
lentyuk:_____
○ 2380 char1,0,14,"s:save l:load d:dire
ctory p:nyomtatás"
○ 2390 char1,0,16," c.fel sortolas felfe
le
○ 2400 char1,0,17," c.le sortolas lefel
e
○ 2410 char1,0,18," c.bal keptolas balra
/2 kar.
○ 2420 char1,0,19," c.jobb keptolas jobbr
a/2 kar.
○ 2430 char1,0,20," > keptolas jobbr
a/10 kar.
○ 2440 char1,0,21," < keptolas balra
/10 kar.
○ 2450 char1,0,22," a adatujairas a
graf. kepernyore
○ 2460 char1,0,23," r ujraindítás
2470 char1,0,24," h sorhossz ujraa
llitasa
○ 2480 poke239,0: get keyhp$
2490 ifhp$ ="a" goto2520
2500 ifhp$ ="r" then run
○ 2510 gosub920: graphic1: goto360
2520 key1, chr$(4): key2, chr$(6)
○ 2530 graphic1: sys dec("152a"): goto880
2540 get keya$
2550 remkodok/magyarekezesbet.
○ 2560 ma$(1) = chr$(160) + chr$(212) +
chr$(214) + chr$(213) + chr$(184)
+ chr$(192)
○ 2570 ma$(2) = chr$(188) + chr$(193) +
chr$(192) + chr$(193) + chr$(188)
+ chr$(192)

```

```

○ 2580 ma$(3) = chr$(184) + chr$(212) +
chr$(214) + chr$(213) + chr$(216)
+ chr$(128)
○ 2590 ma$(4) = chr$(184) + chr$(194) +
chr$(193) + chr$(192) + chr$(186)
+ chr$(193)
○ 2600 ma$(5) = chr$(176) + chr$(202) +
chr$(201) + chr$(200) + chr$(178)
+ chr$(129)
○ 2610 ma$(6) = chr$(184) + chr$(197) +
chr$(196) + chr$(197) + chr$(184)
+ chr$(128)
○ 2620 ma$(7) = chr$(184) + chr$(192) +
chr$(194) + chr$(193) + chr$(184)
+ chr$(192)
○ 2630 ma$(8) = chr$(128) + chr$(200) +
chr$(250) + chr$(193) + chr$(128)
+ chr$(128)
○ 2640 ma$(9) = chr$(176) + chr$(200) +
chr$(202) + chr$(201) + chr$(176)
+ chr$(128)
○ 2650 close2: open2,4,7: trap360
○ 2660 forz =0 to5039:ip$ = chr$( peek(36
864 +z))
2670 ifip$ ="[C= A]" orip$ ="[C= S]" or
ip$ ="[C= D]" orip$ ="[C= F]" orip$
="[C= G]" then gosub2700
2680 print#2,ip$;
○ 2690 next: goto360
2700 k$ =ip$
○ 2710 ifk$ ="[C= A]" thenma$ =ma$(1):
goto2800
○ 2720 ifk$ ="[C= S]" thenma$ =ma$(2):
goto2800
○ 2730 ifk$ ="[C= D]" thenma$ =ma$(3):
goto2800
○ 2740 ifk$ ="[C= F]" thenma$ =ma$(4):
goto2800
○ 2750 ifk$ ="[C= G]" thenma$ =ma$(5):
goto2800
○ 2760 ifk$ ="[C= H]" thenma$ =ma$(6):
goto2800
○ 2770 ifk$ ="[C= J]" thenma$ =ma$(7):
goto2800
○ 2780 ifk$ ="[C= K]" thenma$ =ma$(8):
goto2800
○ 2790 ifk$ ="[C= L]" thenma$ =ma$(9):
goto2800
○ 2800 ip$ = chr$(8) +ma$ + chr$(15):
return

```

# HÍREK

## Összefogásra kényszerülnek a nyugat-európai számítógépgyárok

A számítástechnikai piacon kibontakozó új irányzatok hatására egyre sürgetőbb a nyugat-európai számítógépgyárok számára valamilyen együttműködés kialakítása. A gyors ütemű műszaki fejlődés miatt külön-külön szinte valamennyien stagnálásra, illetve rosszabb esetben veszteséges gazdálkodásra lennének kényszerítve. A piaci feltételek váratlan keményedését arra vezetik vissza, hogy mindinkább teret nyer a szabványosítás, így csökkennek a hardverárak.

A japán és az amerikai nagyvállalatok agresszív expanziója miatt számos nyugat-európai cég eredményei romlottak. A norvég Norsk Data fennállása óta először ez év januárjában jelentett veszteséget. A nyugatnémet Nixdorf profitja az egy évvel ezelőttinél a töredékére zsugorodott, és az Olivetti igazgatósága is rossz eredmények közzétételére készül. A finn Nokia cégnek szintén gondjai vannak az Ericssontól átvett számítástechnikai részleggel.

A felsorolt cégek kisebb vagy nagyobb mértékben jelen vannak abban a piaci szektorban, ahol a legnagyobb arányú változás figyelhető meg. Azokról a közepes és kisebb kategóriájú számítógépekről van szó, amelyeknek az operációs rendszerét egyre szabványosabbá teszik. Ennek eredményeként a felhasználók válogatni tudnak a szállítók ajánlataiból, kíméletlen árversenyre kényszerítve őket.

Konkrét együttműködési tervekről még nincs hír, a számítástechnikai iparral foglalkozó elemzők szerint azonban elkerülhetetlen, hogy az érintett cégek valamilyen formában kapcsolatot keressenek. Feltételezik például, hogy a Nixdorf partnerrá talál a brit ICL-ben, amely egyébként a nagy számítógépek gyártásában is érdekelt. Egy másik feltételezés szerint a brit cég az olasz Olivettivel, esetleg a nyugatnémet Siemens-szel tudna előnyös együttműködést kialakítani. (Reuter)



Bemutatkoznak a

# SYSTEM GEORGE

## nyelvoktató programok

A legutóbbi években a nyelvismeret döntő szerepet kapott gazdaságunk fejlődésében. Közismertek azonban mind a nyelvismerettel kapcsolatos szegényes statisztikai adataink, mind a nyelvoktatás, nyelvtanulás problémái. Szükség van tehát olyan módszerekre, melyek a hagyományos oktatás hibáit csökkentik, netán kiküszöbölik és ezzel a tanulást eredményesebbé teszik.

### A SZOFTVER FEJLŐDÉSE

Ez a felismerés vezetett ahhoz, hogy 1984 elején a mai, mintegy 15 szerzőből álló csoport magja, a mikroszámítógépek hazai elterjedésével egyidőben, elkezdte a nyelvoktató programok fejlesztését. Először az ANGOL—1 program első, technikailag kezdetlegesebb változata került kidolgozásra. Már ekkor, de később is célkitűzés volt, hogy teljes tanfolyami anyagot kell számítógépre vinni. Ez a program nagy sietséggel készült, mivel még a munka közben célszerű volt kipróbálni, hogy mennyire alkalmazható a gyakorlatban. Forgalmazása 1985 júliusában kezdődött. Egyrészt a visszajelzések, másrészt a szerzők közvetlen tapasztalatai alapján 1986 közepére elkészült a fejlesztett 2. változat, illetve az ANGOL—2 és a NÉMET—1, NÉMET—2 programok. Egyidejűleg megszűnt az ANGOL—1 1. változatának forgalmazása. Ennek felhasználói a 2. változatot 75% árengedménnyel kapták meg.

A további fejlesztési munkát a FRANCIA—1, majd 1987 közepén a SPANYOL—1, ill. 1987 végén az OROSZ—1 programok megjelenése jelezte. Még 1986-ban készült el az ANGOL és NÉMET, majd 1988 elején az OROSZ programok hanganyaga. Ezzel sikerült olyan komplett nyelvoktató rendszert kialakítani, mely mind a magántanulókat, mind a tanfolyami oktatást segíteni képes.

A szerzők 1987 óta foglalkoznak olyan nyelvoktató programok létrehozásával, melyek videoműsorral összekapcsolva működnek. Eddig nem sikerült olyan végleges megoldást találni, mely mind a technikai, mind a copyright problémákat elfogadható költségkereten belül megoldja. A kutatás a legolcsóbb, iskolák által is megfizethető megoldás irányában ma is folyik.

A programok nyelvi anyagát szaktanárok készítették és lektorálták. Az ellenőrzést ezenkívül anyanyelvi lektorok is elvégezték. A hanganyagok elkészültekor megtörtént egy második anyanyelvi lektorálás is. A gondos munka és ellenőrzés biztosította, hogy minimális mértékben fordult elő olyan észrevétel, mely a gyakorlatanyagra vonatkozott és javítást igényelt.

### A HARDVER

A fejlesztés másik vonalát a hardver-oldal jelentette. A programok minden esetben először Commodore—64 gépre, mágneslemezes, majd 1986-tól kazettás változatban is készültek. Az ANGOL—1 program 1. változata ezenkívül megjelent a következő gépeken is: IBM PC, PROPER—8, SYSTER, LABSYS, COMPUT—8, ROSYTEX, Sinclair Spectrum, Commodore—16. Az IBM PC kompatibilis gépekre 1987-ben elkészült az ANGOL és NÉMET programok második, majd 1988-ban a harmadik változata, mely már NOVELL hálózatban is működtethető.

Folyamatban van az összes többi program PC-re való alkalmazása is. Ez a fejlesztés 1990 elején az OROSZ—1 programmal fejeződik be.

### A PROGRAMOK MŰKÖDÉSE

A programok megegyeznek abban, hogy  
— az alkalmazásra vonatkozó részletes magyar nyelvű tudnivalók a képernyőn megjeleníthetők vagy printerrel kinyomtathatók,  
— két alapüzemmódban működtethetők: ezek a tanuló-gyakorló és az ismétlő-tesztelő üzemmód,  
— a különleges karakterek minden nyelvnél, mind a kis-, mind a nagybetűk esetében hitelesen jelennek meg.  
Az alkalmazott módszerek tekintetében a programok 3 csoportba sorolhatók. Ezek, és főbb jellemzőik a következők:

1. csoport (ANGOL—1, ANGOL—2, FRANCIA—1, OROSZ—1, SPANYOL—1).

Tanuló üzemmódban

— az összesen háromféle feladattípus gyakorlatait logikailag egymásra épített sorrendben kapjuk,  
— választható automatikus és kézi vezérlés, mely utóbbinál magunk állíthatjuk be a feltételeket,  
— meghatározott feladatcsoportonként a helytelenül megválaszolt kérdéseket visszaadja a program,  
— az adott körből nem jutunk ki, míg minden kérdésre helyesen nem válaszoltunk,  
— ha a megszabott minimális szintet nem értük el, előről kell kezdeni az adott szakasz feladatainak gyakorlását,  
— válaszadás előtt vissza lehet lépni az előző, már megoldott feladatokra a céllal, hogy segítséget kapjunk.

Tesztelő üzemmódban

— semmiféle segítség (hibavizsgálat, megoldott feladatok visszahívása stb.) nem vehető igénybe,  
— választani lehet, hogy mely fejezetekből, nyelvtani, ill. témakörből vagy az egész lemezről kapjuk véletlenszerűen a kérdéseket,  
— választható az is, hogy esetleg a tanulás sorrendjében legyen előírva a válaszadás.

Egyébként

— bármely kérdés félretehető a memóriába,  
— a félretett kérdésekkel a napi gyakorlás végén külön lehet foglalkozni,  
— ez a kérdéscsoport, valamint a napi eredménytábla külön lemezre is kimenthető és később behívható,  
— a különböző billentyűk használatát ismertető táblázat hívható be,  
— a különféle szolgáltatások (folyamatos eredményérés, időnyilvántartás, hangjelzés stb.) ki- és bekapcsolhatók,  
— a gyakorlatanyag bármely része vagy egésze végigolvasható, a megoldással együtt.

2. csoport (ANGOL—2—H,I,J).

Három lemezen nyelvtani, ill. lexikai gyakorlatokat tartalmaznak.

Működésük jellemzői:

— A gyakorlás szintválasztással kezdődik.  
— Az anyag megtanulása az első, kezdő szinten történhet.  
— Az ötödik, legmagasabb szinten alig 10 másodperc áll rendelkezésre a válaszadásra.  
— A gyakorlatanyag egy-egy lemezen 16 fejezetből áll.  
— A kezdő szinten a kiválasztott fejezet gyakorlatait sorban oldhatjuk meg, és segítség is igénybe vehető.  
— A többi szinten két, négy vagy több, esetleg az összes fejezetből véletlenszerűen kapjuk a kérdéseket.  
A játékos, versenyszerű programok különösen az ifjúság körében népszerűek.



**3. csoport (NÉMET—1, NÉMET—2).**

Három, ill. négy lemezen 12-12 fejezetben (blokkban) tartalmazzák a különféle gyakorlatokat. Jellemzőik:

- Minden fejezet azonos felépítésben 8 programból áll.
- A gyakorló üzemmódban a program a tanuló (tanár) által meghatározott módon adja vissza a kérdéseket, és csak akkor enged tovább, ha a bevitt feltételt teljesítettük.
- A gyakorlatok egyik alaptípusánál négy változat bármely variánsa helyes lehet.
- Ennél a kérdéscsoportnál egyszerre két hallgató is leülhet a géphez, és ilyenkor felváltva kapják a feladatokat, és az eredményük elszámolása is külön történik.
- A másik alaptípusnál egyszerre hat mondat jelenik meg, melynek hiányzó részeit egy adott készletből csak egyféleképpen lehet helyesen pótolni.
- A főnevek nemét gyakorló programban a színek szerinti megkülönböztetés és az elhelyezés módja nagy segítséget jelent a tanulásban.
- A főnevek ragozásával kapcsolatos feladatok megoldása a végződések begépelésével történik.
- Az igék alakjait is be kell gépelni a megfelelő program alkalmazásánál.
- A különleges karakterek igen egyszerűen gépelhetők be.
- A főnevek, igék és egyéb szavak szószedete külön programokban rendelkezésre áll.
- Az elért eredmény részben folyamatosan, részben külön táblázatban tekinthető meg, ill. behívható.

**MAGÁNTANULÁS**

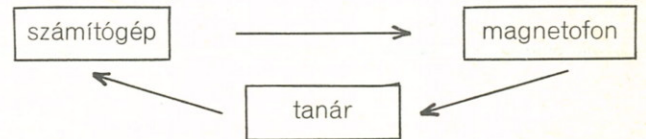
Tekintettel arra, hogy legtöbb programnak hanganyaga is van, a tanári segítség teljesen nélkülözhető azoknál, akik önállóan is tudnak tanulni. Mérsékelt tanári felügyelettel azonban mindenki eredményesebb, olcsóbban és kevesebb időráfordítással tanulhatja a nyelveket, mintha hagyományos tanfolyamra járna. Le kell azonban szögezni, hogy — mint mindent — a nyelvet is csak komoly munkával lehet megtanulni. A rendelkezésre álló eszközök, mint ezek a programok is csak megkönnyíteni és nem helyettesíteni tudják ezt a munkát.

A SYSTEM GEORGE nyelvoktató programok alkalmazására rengeteg lehetőség van. Legcélszerűbb először — a hibavizsgálat kiiktatásával — az előirányzott napi adagot egyszerűen átolvasni. Ezt követheti a hanganyaggal együtt történő olvasás, majd a gyakorlatok megoldása, szintén a hanganyaggal párhuzamosan. Az elért eredménytől függően ezután szigorított feltételekkel addig gyakoroljuk a kérdéseket, míg kielégítő eredményt érünk el. Ha ily módon megfelelő mennyiségű gyakorlatot átvettünk, tesztelő üzemmódban végezhetjük annak vizsgálatát, hogy mindent kellően elsajátítottunk-e. Munkánk alatt összegyűlt néhány probléma, melyeknek nyelv- és nyelvtankönyvben nézünk utána. Ha megoldhatatlan probléma merül fel, tanárral kell konzultálnunk.

**MAGÁNTANÁR**

A leghasznosabb mind a tanár, mind a tanulók számára, ha a tanár rendelkezik egy C—64 számítógéppel és a megfelelő SYSTEM GEORGE programmal. Ilyenkor az alábbi alapséma szerint folyhat az oktatás:

Egyszerre 3 tanuló vesz részt az oktatásban. Ezek akár a legkülönbözőbb tudásszintet képviselhetik, mivel az oktatás teljesen egyéni. Egy alkalommal 3 órát vesz igénybe a foglalkozás. A tanulók számára az alábbi 3 munkahelyet kell képezni:



A tanulók óránként váltanak munkahelyet. A számítógépnél a tanár által meghatározott módon gyakorolnak vagy beszámolnak. A magnetofonnál a beszédértés és helyesírás ellenőrzése folyik, egyrészt a programok hanganyagával, másrészt a tanár által előírt anyaggal. Ezen a munkahelyen időnként más feladatokat is kaphat a hallgató, pl. fordítást. A tanárnál a másik két munkahelyen, valamint az otthon végzett munka ellenőrzése, a problémák megbeszélése és a beszédképesség fokozása történik.

**TANFOLYAMI OKTATÁS**

Amennyiben a tanfolyamot vezető tanár úgy dönt, hogy a SYSTEM GEORGE programokat, mint elsődleges tananyagot használja az oktatásban, fenti sémát alkalmazhatja 2, 3 vagy 4 számítógéppel és magnetofonnal. Ilyenkor sem kell a hallgatóknak azonos szinten lenniük. Csak az a követelmény, hogy a tanárhoz egyszerre kerülők társalgási szempontból közel azonos szinten álljanak.

Egyébként sokféle megoldás lehetséges. Ezek közül itt kettőt említünk:

1. A tanfolyam a megszokott módon folyik, és a programokat csak háttér tanulásra használják. A tanfolyam szervezői ebben az esetben a számítógépes gyakorlási lehetőséget biztosítják a hallgatók számára. Mivel a számítógépes anyagban mindig megtalálható az éppen tanult nyelvtan, ill. lexika megfelelője, a hallgató minden órára ezzel is felkészülhet és így elmélyíti az ismeretanyagot.

2. A tanfolyamot a tanár úgy irányítja, hogy az órán az általa használt nyelvkönyv is és a számítógépes tananyag is szerepet kap. A tanulók tehát nemcsak az órán kívül találkoznak a számítógéppel, hanem az órán is. A tanár ebben az esetben a számítógépek és a magnetofonok számától függően a hallgatók egy részét folyamatosan beszámoltatja, vagy pl. a külön lemezre kimentett, nehezen megoldott feladatokból időnként probléma-órákat rendezhet stb.



**Dr. Vadász György**  
közgazdász,  
rendszerfejlesztő



**Vadász György**  
nyelvtanár,  
programfejlesztő, programozó



**Rozsnyói Árpád**  
mérnök,  
programozó





Adott esetben az oktatás, a tanulás optimális lehetőségeit a szerzők szívesen megtárgyalják a tanárokkal. Ez a szolgáltatásuk ingyenes.

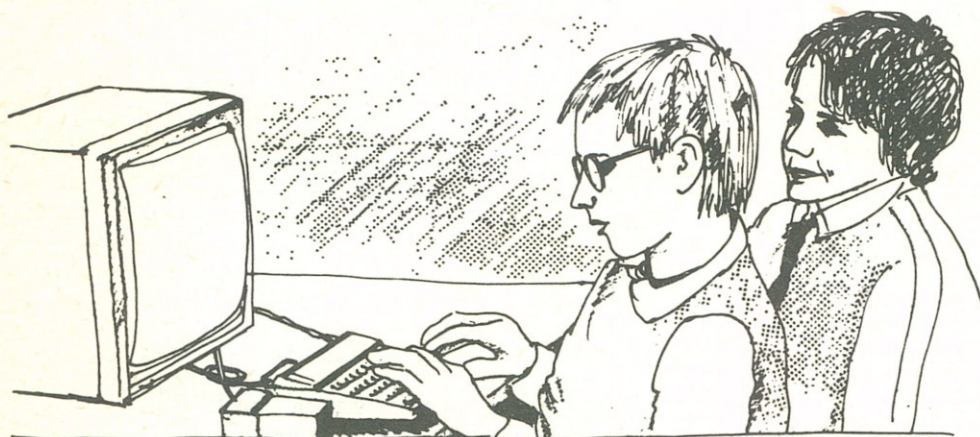
### AZ ÁRAK

Az alkalmazott árpolitika abból indul ki, hogy a vállalatok sok, esetleg több száz dolgozójuk számára és sok évre vásárolják a programokat, tehát már a 3. példánynál 75% engedményt kapnak a kiinduló árhoz képest (4000 Ft/prg). Amennyiben négy-nél több gépre vagy hálózatban pl. 10 munkahelyen kívánják a prog-

ramokat használni, akkor négy program árát (32 000 Ft) fizetik. Az átlagár így 3200 Ft/prg. 20 munkahelyre, ill. hálózatra 8 program árát kell fizetni, azaz 48 000 Ft-ot. Ebben az esetben az átlagár 2400 Ft/prg., előlött munkahelyenként 1000 Ft/prg fizetendő, tehát az átlagár tovább csökken.

Ennek alapján érthető, hogy az iskolák, művelődési intézmények, magánszemélyek stb. már az első két példány esetében is nagy engedményt kapnak.

Befejezésül csak annyit, hogy ez a bemutatkozás még oldalon át tarthatna. Az érdeklődők számára azonban mind az Országos Commodore Egyesület irodája, mind a szerzők szívesen állnak rendelkezésre.



## ÉKEZETES C-16

E cikk tekinthető a „PLUS/4 gép ékezetesítésének (és ezzel együtt a saját karakterek használatának) problémái” címen, a Commodore Újság 1988. decemberi számában megjelent cikk folytatásának.

### ELŐLJÁRÓBAN

Lehet, hogy mániákusságnak tűnik, de mi magyarok vagyunk, ezért fontosnak tartom még a legegyszerűbb magyar nyelvű játékprogramok esetén is, hogy azok tényleg magyarul, azaz ékezetes karakterekkel íródjanak. És itt elsősorban gyermekeinkre, a számítógépeken (is) felnövő új generációra gondolok — úgyszólván túl keveset olvasnak, sok televíziót néznek, általában gond van a helyesírásukkal stb. — ezért szinte bűnnek tartom, hogy egyik kedvenc játékuk rontja (tovább rontja) magyar nyelvű ismereteiket.

Az eddig megjelent szakirodalom nemigen foglalkozik részletesen azzal, hogyan ékezetesítsük gépünket, ha az nem ékezetes. És ha már ékezetesítettük, hogyan dolgozzunk vele. Jelen cikk azért íródott, hogy megkönnyítse a programírók, -fejlesztők széles táborának munkáját.

### HOGY A GÉP MAGYARUL ÍRJON

Szerencsére nem kell karaktereket terveznünk, erre kiválóan alkalmas pl. Mészlányi Zoltán „Ékezetes betűk a PLUS/4-

esen” programja, ha azon néhány sort módosítunk.

Az új ékezetes készlet a PLUS/4 gépen is a BASIC munkaterület végére kerül (ha nem használ grafikát). Ezt kell elérnünk C16-on is, ahol a BASIC munkaterület a \$4000 címig tart. Ehhez a már idézett programban az alábbi módosításokat kell végrehajtani:

```
0 REM SAVE"ékezetes C16/1" — 1988.
dec. 20.
```

```
10 POKE 56,56:clr
20 print" <fehér> <törlés> <le> <le> <le>
<le> <le> <le> td000 d7ff 3800
```

```
380 for i=16136 to 16351
```

```
410 keyl,"poke 65298,192:poke65299,56:
print chr$(14)" + chr$(13)
```

```
420 poke 65298,192:poke65299,56
```

```
430 print "Ékezetes karakterkészlet";
```

```
440 print" $3800-3FFF"
```

A program lefuttatása után karakterkészletünk a \$3800-\$3FFF címekre kerül.

#### Megjegyzések:

1. Mészlányi programjával ellentétben a karakterkészlet helyét megadó utasítást azért tettem az F1 billentyűre, mert ha valaki PLUS/4-en fejleszt C16-os programot és programírás közben véletlenül az F1 funkcióbillentyűt nyomja

meg, zavaros képernyővel a szövegszerkesztő jelentkezik be, és addigi munkánk eredménye kárba veszik.

2. Ha biztosak vagyunk abban, hogy csak a kisbetűs készletet akarjuk használni, akkor a következő utasításokat kell megváltoztatni:

```
10 POKE 56,60:CLR
```

```
20 ..... TD400 D7FF 3C00
```

```
410,420 sorok: .... POKE 65299,60
```

```
440. sor: PRINT utasításban: $3C00-3FFF
```

3. Ha a BASIC program mérete nem engedi meg, hogy mindkét karakterkészlet a RAM-ban legyen és szükségünk van mindkét készletre, akkor az ékezetesítő programban a 2. megjegyzésben leírt módosítások mellé még tegyük fel pl. az F2 billentyűre az eredeti karakterkészletet visszaváltó utasítást:
 

```
415 KEY2,"POKE54298,196:POKE65299,208"
```

Az így elkészített készleteket használó programokban ajánlatos két rutint készíteni a karakterkészletek átváltására, pl.

```
10000 POKE65298,192:POKE 65299,60:
PRINT CHR$(14): RETURN: nem új
10010 POKE65298,196:POKE 65299,208:
RETURN: nem eredeti
```

4. Célszerű kimenteni az így kapott karakterkészletet (monitor üzemmódban) — ekkor ugyanis az ékezetes készlet bármelyik BASIC program mellé utólag is betölthető (monitor üzemmódban!). Betöltés után azonban ne felejtjük el parancs módban kiadni:

— BASIC terület végét beállító utasításokat:

```
POKE 56,56 (teljes karakterkészlet esetén), illetve
```

```
POKE 56,60 (csak kisbetűs karakterkészlet esetén).
```

```
POKE 54,PEEK(56):POKE 52,PEEK(56):POKE 51,0: POKE 53,0: POKE 55,0
```

### GRAFIKUS KARAKTEREK HASZNÁLATA

Ékezetesített gépen a kisbetűs készlet grafikus karaktereinek száma 18-ra csökken. Ez nem jelent problémát akkor, ha csak ezeket akarjuk használni. Ha azonban ezeken kívül (helyett) más karakterekre van szükségünk, már gondban vagyunk. Helyzetünk azonban ekkor sem reménytelen, mert ha nincs szükségünk az összes gra-



fikus karakterre, előre gondosan megtervezhetjük, melyiket akarjuk használni és ekkor a karakterek csereberelésével célt érünk.

## ÉS HA KEVÉS A 18 GRAFIKUS KARAKTER?

Természetesen ekkor is van megoldás, csak kompromisszumra kényszerülünk: elégedjünk meg azzal, hogy programunk csak nagybetűket használ. Ekkor csak a 8 ékezetes nagybetűt kell elhelyeznünk a megfelelő billentyűkön, és így csak 8 grafikus karakterről kell lemondanunk. Egy ilyen karakterkészlet az „Ékezetek C16-ra” program 4. készlete.

Szebb írásképet kaphatunk, ha az összes írásjelet és nem ékezetes kyraktert hozzáillesztjük az ékezetekhez. Két ilyen karakterkészletet állít elő a „Kis ékezetes C16” program.

(A fenti trükkök természetesen PLUS/4 gépre is alkalmazható!)

## ÉS HA AZ ÖSSZES GRAFIKUS KARAKTER KELL?

Újabb kompromisszumokat kötve, erre is van megoldás: szöveget csak a képernyő minden második sorába írunk és az előlötti sorokban helyezük el az ékezeteket. Háromféle ékezet van a magyarban: a vessző, a két vessző és a két pont. Mivel az írásjelként használt vessző alkalmas az „egy vessző” ékezet írására, így csak két új karakterre van szükségünk. Ezeket célszerű két, ritkán használt írásjel helyére tenni, pl. a szögletes zárójelek ([ és ]) helyére. Egy ilyen karakterkészletet állít elő az „Ékezetek C16-ra” program is (6. készlet).

## ÉS HA PROGRAMUNK MÉG EZT SEM ENGEDI MEG?

(mert szükségünk van a saját karakterkészlet által elfoglalt helyre is!)

Ekkor írjunk az előző pont szerint, de lehetőleg csak olyan szöveget írjunk a képernyőre, amely az Á, É, Í, Ó, Ú magánhangzókon kívül más ékezetes magánhangzót nem tartalmaz.

## EGY UNIVERZÁLIS ÉKEZETESÍTŐ PROGRAM, MELY AZ ÉKEZETESÍTÉSEN KÍVÜL MÉG SOK EGYEBET IS TUD

Úgy gondoltam, felesleges az eddig leírt variációkat külön-külön kis programcskákban megvalósítani. Helyette célszerűbb egyetlen univerzális programot készíteni, melynek futtatásakor a felhasználó maga választja ki a számára szükséges ékezetes karakterkészletet. Ez a program az „Ékezet C16-ra” nevet viseli.

A program a kívánt karakterkészlet előállítását mellett az eddigi ékezetesítő prog-

ramok funkciói mellett még további szolgáltatásokat is nyújt, nevezetesen:

- megjeleníti az új(régi) billentyűzetet — ezáltal látszik, melyik karakter hová került;
- kívánságra kiírja a képernyőre az eredeti, ill. új megjeleníthető karaktereket ASCII ill. képernyőkódjuk növekvő sorrendjében;
- visszaállíthatja az eredeti karakterkészletet;
- kiírja a képernyőre az egyes karakterek adatait a megfelelő karaktergenerátorból, lehetővé téve azok tanulmányozását, pl. későbbi átszerkesztés vagy nyomtatáshoz illesztés céljából.

### Megjegyzések:

1. Hasznos tanulmányozni a program szerkezetét, a program rutinjait.
2. A program, bármennyire is szerettem volna, nem bővíthető tovább egyéb funkciókkal, mert különben nem fér el C16-on.
3. Gyekezzem a képernyőszövegeket úgy kialakítani, hogy azok mindig helyes magyarságúak legyenek. E célt szolgálják pl. a különböző karakterkészletek változóinak értéket adó rutinok.

## A PROGRAM MŰKÖDÉSE

A program RUN-nal indul, s rögtön az 1. menü jelentkezik, mely a karakterkészlet-választékot sorolja fel.

### 1. menü:

1. \$3800-\$3FFF: grafikus + ékezetes készlet
2. \$3C00-\$3FFF: ékezetes készlet
3. \$3C00-\$3FFF: ékezetes készlet, s emellett elérhető a \$D000-\$D3FF: grafikus készlet
4. \$3C00-\$3FFF: grafikus készlet, ékezetes nagybetűkkel
5. \$3C00-\$3FFF: grafikus készlet, kisebb ékezetes nagybetűkkel
6. \$3C00-\$3FFF: grafikus készlet, az ékezetesítéshez + két új ékezet
7. \$3C00-\$3FFF: egyéb készlet: ez a pont ad lehetőséget egyéb karakterkészlet vizsgálatára!
8. TÖVÁBB: át kell lépni a 2. menüre, pl. az aktuális készlet vizsgálatához.

Az 1—6. válasz esetén a program elkészíti a kért karakterkészletet, megjeleníti a billentyűzetet, majd a 2. menüre adódik a vezérlés.

7-es válasz esetén a program megkérdezi, bent van-e a vizsgálandó karakterkészlet a memóriában. Nemleges válasz esetén befejeződik a program futása, majd monitor üzemmódban be lehet tölteni az új készletet a \$3C00-\$3FFF területre; ezután a program újraindítható.

Mivel az 5. készlet a program betöltésekor a \$3800-\$3BFF memóriaterületen van, az 1. készlet előállítását után már nem áll rendelkezésre. Ilyenkor a programot újra be kell töltenünk. (Érdekes lehet az 5. készlet meglétének ellenőrzése is: a program betöltésekor a 0. utasításban a 759-es memóriacím 0 értéket kap. Az 1. készlet

betöltése után erre a címre 1-et írunk és a 0. utasításban a "poke" tokenjét a "REM" utasítás tokenjére írjuk át. Az 5. készlet indításkori helyének megléte egyszerűen ellenőrizhető a 759-es cím tartalma alapján. Ennek értéke egy új ékezetes készlet betöltése és a program újraindításakor is változatlan marad, hisz a 0. utasítás ekkor már megjegyzésként értelmeződik.)

**2. menü:** karakterkészletek vizsgálatához, újraindításhoz, programfutás végéhez.

Pontjai:

1. Eredeti ROM karakterkészlet beállítása.
2. Új karakterkészlet visszaállítása.
3. ASCII kódokat kiírja a képernyőre.
4. A képernyőkódokat írja ki a képernyőre.
5. A karakterkészletet az ASCII kódjakkal és képernyőkódjukkal együtt írja ki a képernyőre.
6. A karakterek adatai vizsgálhatók.
7. Megjeleníthető a billentyűzet.
8. Újraindítható a program.
9. Programfutás vége.

**Karakterek adatai:** a 2. menü pontjai közül ezt kell részletesen megmagyarázni. Ez a rész további menüpontokat tartalmaz.

**1. választás:** melyik készlet adatait kérjük:

1. új készlet
2. eredeti 1.set (grafikus)
3. eredeti 2.set (kisbetűs/nagybetűs)
4. vissza a 2. menühöz

**2. választás:** mely újakarakterek adataira vagyunk kíváncsiak:

1. az összes karakterre
2. egy adott karakterre: ezt a megfelelő billentyű lenyomásával lehet majd megadni; a karakterek megadását a <RETURN> billentyű lenyomásával kell befejezni.
3. intervallumban adjuk meg a karaktereket: ekkor a 3. választás után megkérdezi a program, hogy „honnan” és „meddig”; ezekre a karakterek képernyőkódjával kell válaszolni.

**3. választás:** milyen típusú adatokat szeretnénk a képernyőn látni:

az adatokat a program a megfelelő karaktergenerátorból veszi:

1. decimálisan kérjük az adatokat, vagy
2. hexadecimálisan, vagy
3. mindkettőt együtt szeretnénk látni, vagy

4. a decimális és hexadecimális adatok mellé a karaktergenerátorbeli képet felnagyítva, rajzban szeretnénk látni.

## A PROGRAM SZERKEZETE

### BASIC főprogram:

- 0— 10. sorok: kezdőértékdadások
- 20—350. sorok: 1. menü és vezérlés
- 360—430. sorok: 2. menü és vezérlés
- 450—7000. sorok: szubrutinok

### Gépi kódú részek:

\$37B2-\$37BA:

Monitor utasításhoz (a karakterkészletek másolás monitorban történik; ennek befejezése után az innen a billentyűzet-pufferbe másolt karaktereket olvasva hajtódik végre a visszatérés a BASIC főprogramba).



**\$37D5-\$37DF:**

ROM-olvasó rutin: az eredeti ROM karakterkészletből egy karakter adatait teszi hozzáférhetővé.

Hívása: SYS14293

**\$37E0-\$37F8:**

Indítórutin: a BASIC program végét állítja be. (Betöltéskor ugyanis a BASIC-program vége mutatók a \$3C00 címre mutatnak!)

Hívása: SYS14304

**\$36C0-\$3797:**

1—4. karakterkészlet új karaktereinek adatai

**\$3798-\$37AF:**

a 6. készlethez a két ékezet és a közte lévő 3. karakter adatai

**\$3800-\$3BFF:**

teljes 5. készlet karaktereinek adatai

**Subrutinok:**

450—452: eredeti ROM karakterkészlet beállítás

460—465: új karakterkészlet visszaállítás

470—471: nem létező kód helyett space-t írunk majd

475—477: mindegyik karakterkészletre közös értékadások

480—483: minden 2. sorba írjunk?

485: C=/Shift billentyűk tiltása, engedélyezése

490: teljes ékezetes készletet előállító rutin (1—3. készlethez)

500—570: ASCII kódokat írja a képernyőre: ezen belül: 500: vezérlés, 510—515: egy intervallum(sorozat) kiírása, 520—570: egy képernyő kiírása

600—675: képernyőkódokat írja a képernyőre; ezen belül: 600: vezérlés, 610—615: egy intervallum (egy sorozat) kiírása, 620—675: egy képernyő kiírása

680: „lapozó”-rutin: T billentyű lenyomására vár

690: „lapozó”-rutin: T vagy V billentyűre vár

700—780: ASCII és képernyőkódokat együtt írja a képernyőre; ezen belül: 700: vezérlés, 710—715: egy intervallum(sorozat) kiírása, 740—780: egy képernyő kiírása

800—895: karakterkészlet adatainak kiírását vezérli; ezen belül: 800—851: karakterkészlet kiválasztása, ill. visszatérés 855—860: vizsgálandó karakterek kiválasztása

865—871: adattípus kiválasztása

873: végrehajtás vezérlése

875—880: egyedi karakterek rész

890: a képernyő címsora

895: eredeti ROM-készlet olvasása (ehhez tartozó részek még: 1400—1650 sorok is)

900—906: 1. menükészítő rutin

910—913: 2. menükészítő rutin

950: egy karakter adatainak beírása az új karakterkészletbe

960: COLOR utasítások a képernyő színezéséhez

970—980: vízszintes scroll-rutin

1000—1040: változók: kisbetűs/grafikus készlethez

1050—1087: változók: nagybetűs ékezetes készlethez

1100—1130: változók: nagybetűs, grafikus,

2 új ékezetes készlethez

1150—1180: változók: eredeti készlethez

1200—1330: billentyűzetet írja a képernyőre; ezen belül: 1200—1290: vezérlés, 1300—1330: teljes fejléceket ír, 1330: csak fejsort ír

1400—1655: karakterkészlet adatait írja ki a képernyőre; ezen belül: 1400: teljes készlet határainak beállítása

1410—1440: egy intervallum kiírása,

1450—1480: egy karakter adatai,

1500—1520: intervallum megadása (felhasználótól kéri),

1550: csak karakteradatok kelljenek, vagy a rajz is?,

1600—1625: karakter rajzát írja a képernyőre,

1650—1655: egy byte 1-es bitjeinek helyére \* karaktert rajzol

2000—7000: karakterkészletek előállítása:

2000—2030: 1. készlet

3000—3030: 2. készlet

4000—4030: 3. készlet

5000—5010: 4. készlet

6000—6010: 5. készlet

7000: 6. készlet

**A program összeállítás** (nagy gondosságot igényel!)

1. Gépeljük be az „Ékezetek C16-ra” BASIC programot. Ügyeljünk arra, hogy a túl hosszú sorok csak az utasítások rövidítéseit használva írhatók be! Igyekezünk a programot pontosan úgy begépelni, ahogy az a listán szerepel (azaz felesleges szóközök nélkül és az 1. sor kivételével REM nélkül); ellenkező esetben a program futása „out of memory” üzenettel megszakadhat.

A SYS-rutinok hívását tegyük REM-be, majd próbáljuk ki mindent, amit ilyen félkész állapotban is lehet. Ezután mentjük ki a programot.

2. Gépeljük be a gép részeket előállító „Ék.C16 gépi” nevű programot. Mentés után futtassuk le, majd a futás eredményét monitorban mentsük el:

S"ÉK/1",1,36C0,3800

3. Gépeljük be a "KIS ÉKEZETES C16" programot. Mentés után futtassuk le valamelyik, nekünk jobban tetsző változatra, majd másoljuk át a kapott karakterkészletet:

MONITOR <RETURN>

T 3C00 3FFF 3800 <RETURN>

4. Másoljuk hozzá (még mindig monitor üzemmódban maradván) a 2. futás eredményét:

L"ÉK/1" <RETURN>

Ezután mentsük ki (még mindig monitorban):

S"ÉK/2",1,36C0,3C00 <RETURN>

5. Töltsük be az 1. pontban elkészített programunkat, majd töröljük a SYS előtti REM-eket. Nézzük meg és írjuk fel a főprogram végét jelző mutatókat:

MONITOR <RETURN>

M2B 2B <RETURN> — írjuk fel a visszaírt 8 értéket.

Ha a programunkat pontosan úgy gépeltük be, ahogy az a listán szerepel, akkor a \$2D-\$2E címeken a \$30D5 érték szerepel. Ha nem, remélhetőleg akkor

sincs túl nagy baj, csak majd javítanunk kell (l. 6. pont).

6. Még mindig monitor üzemmódban maradván betöltjük a 4. pontban elkészített gépi kódú részt.

Ha a BASIC program mérete más volt, akkor javítanunk kell az indítórutin:

A 37E4 LDA 2 D felírt értéke

A 37E6 LDA 2E felírt értéke

Most már kimenthetjük az így összeállított programot:

S"ÉKEZETEK C16-RA",1,1001,3C00

X

Ha jól dolgoztunk, akkor programunk hiba nélkül fog működni. Ha mégis javítanunk kell, javítás előtt felejtjük el monitorban beállítani a \$2D-\$2E címekre a BASIC főprogram végét; a javítás után pedig ennek alapján az indítórutin fenti címeit.

## HOGYAN MENTSÜNK KI ÉKEZETES KARAKTEREKET HASZNÁLÓ PROGRAMOT?

1. Ékezetes karakterkészlet nélkül, de ekkor a program minden betöltése előtt a gépet ékezetesíteni kell.

2. Az ékezetes készlettel együtt, monitorban:

a) \$1001—4000 címek között:

Ha BASIC programunk mérete ezt igényli, vagy nincs kedvünk „vacakolni” a karakterkészlet ide-oda másolásával. (De ekkor ne felejtjük el programunk elején beállítani a BASIC munkaterület vége mutatót a megfelelő POKE 56,xx utasítással!)

b) \$1001— . . . címek között:

Ha BASIC programunk jóval rövidebb, mint az ékezetes készlet helye, célszerű azt monitorban átmásolni programunk mögé, és az így kapott rövidebb programot kimenteni. De ekkor a programot úgy kell megírni, hogy futásának elején az áthelyezze a karakterkészletet a megfelelő helyre.

Egy rövid példa erre:

— a BASIC program a \$1001-\$2250 címek között van, s az 1. utasítás sorszáma 10

— módosítsuk BASIC programunk elejét:

4 IF PEEK(56)<57 THEN 8:REM program újraindításakor már nem kell átmásolni (és nem is lehet) a karakterkészletet

5 POKE 34,0:POKE 35,64:POKE 36,0:

POKE 37,44:POKE2035,0:POKE 2036,

8:SYS51192:REM átmásolás

6 POKE 55,0:POKE 56,56:POKE53,0:

POKE 54,56:POKE 51,0:POKE 52,56:

REM BASIC mutatók

7 POKE 65298,192:POKE65299,56:

REM karakterkészlet helye

8 KEY1,"POKE 65298,192:POKE 65

299,56" + CHR\$(13)

— másoljuk át monitorban a karakterkészletet: T 3800 3FFF 2400

— mentsük ki programunkat monitor-



- ban az 1001-2C00 címek között.
- visszatöltéskor BASIC programunk fogja „ékezetesíteni” a gépet.
- ha a program még fejlesztés alatt áll, egy újabb mentés előtt ne felejtsük el visszamásolni a karakterkészletet a

program mögé. Ha a program a javítás során olyan hosszú lett, hogy beletnyúlna a karakterkészlet helyébe, szükség lehet a karakterkészlet helyének mentéskori módosítására a program elejének módosításával együtt!

## Befejezésül

Csak biztatni tudom az olvasókat, saját (ill. gyermekeik) érdekében készítsék el már meglévő programjaik ékezetes változatát. És újabbakat már csak így készítsenek. A fenti útmutató elég ahhoz, hogy egy így ékezetesített gépen gyorsan „magyarítani” lehessen a programokat. Higygyék el, a ráfordított munka sokszorosan megtérül majd. Ehhez mindenkinek sok szerencsét kíván a szerző.

*Lugosi Antalné*

```

0 REM SAVE"EK. C16-GEPI"
1 REM *****
2 REM * C-UJSAG SORSZAM: 172 *
3 REM * EKEZETES BETUKESZLETEK C16-RA *
4 REM * - A PROGRAM GEPI KODU RESZEIT*
5 REM * ALLITJA ELO *
6 REM * PROGRAM: LUGOSI ANTALNE *
7 GOTO 70
8 REM POKE-DLO RUTIN (A=HONNAN; B=HANYAT)
9 J=A+B-1:FORI=ATOJ:READW#:POKEI,DEC(W#):NEXT
10 RETURN
67 :
68 :
69 REM MONITOR UTASITAS ADATAI
70 :
71 A=DEC("37B2"):B=9:GOSUB9
74 :
75 DATAD,50,D,47,CF,32,36,30,D
77 :
78 :
79 REM UJRAINDITO RUTIN
80 :
81 A=DEC("37E0"):B=25:GOSUB9
84 :
85 DATA70,8D,3F,FF,A9,D5,85,2D,85,2F,85,31
86 DATAA9,30,85,2E,85,30,85,32,8D,3E,FF,50,60
97 :
98 :
99 REM ROM-OLVASO RUTIN
100 A=DEC("37D5"):B=11:GOSUB9
101 :
102 DATA00,00,01,D8,99,DF,0,88,00,F0,60
104 :
105 :
106 REM 1-3. KARAKTERKESZLET ADATAI
107 :
108 A=DEC("36C0"):B=216:GOSUB9
109 :
110 DATA0C,10,00,00,30,10,30,00
120 DATAC,10,30,10,10,10,30,0,33,66,30,66,66,66,30,0,0,0,0,0,0,0,0,FF
150 DATA33,66,0,3C,66,66,30,0,CC,CC,33,33,CC,CC,33,33,1,1,1,1,1,1,1,1,0,0,0,0,CC
,CC,33,33
190 DATACC,99,33,66,CC,99,33,66,03,03,03,03,03,03,03,03
210 DATAC,10,30,66,7E,66,66,0,C,10,30,66,7E,60,30,0,10,10,1F,1F,0,0,0
240 DATA0,66,0,66,66,66,3E,0,0,0,0,0,0,0,FF,FF,C,10,30,6,3E,66,3E,0
270 DATA0C,10,7E,60,70,60,7E,00,33,66,0,66,66,66,30,0,66,0,66,66,66,30,0
300 DATA00,66,00,3C,66,66,30,00,C,10,0,66,66,66,3E,0,C,10,0,3C,66,66,30,0
330 DATA66,0,3C,66,66,66,30,0,C,10,66,66,66,66,30,0,C,10,3C,66,66,66,30,0
360 DATA01,03,06,6C,70,70,60,0,33,66,0,66,66,66,3E,0
447 :
448 :
449 REM 2 UJ EKEZET
450 :
455 A=DEC("3790"):B=24:GOSUB9
456 :
460 DATA0,0,0,0,66,66,66,0,C,12,30,7C,30,62,FC,0,0,0,0,0,0,66,66,0

```

READY.



```

0 POKE759,0:KEY2,"P65298,196:P65299,208:CHR$(142)+CHR$(13)
1 SYS14304:REM"EKAZETEK C16-RA"
4 POKE65298,196:POKE65299,208:PRINTCHR$(142)
9 POKE55,190:POKE56,54:POKE53,190:POKE54,54:POKE51,190:POKE52,54:CLR:COLOR0,15,7
:COLOR4,15,6
20 COLOR1,1:GOSUB452:GOSUB475:PRINT"    M    C 1 6    E       KEZETES I       TE       SE"
40 F$(1)="$3800-$3FFF: GRAFIKUS + E       KEZETES":G$="NAGYBETUK":X$="$3C00-$3FFF:
"
50 F$(2)=X$+"E       KEZETES KE       SZLET":F$(7)=X$+"EGYE       B":F$(8)="TOVA       B"
60 F$(3)=F$(2)+RIGHT$(S$,11)+"$D000-$D3FF: GRAFIKUS    "+CHR$(34):X$=X$+"GRAFIKU
S + "
70 F$(4)=X$+"E       KEZETES"+RIGHT$(S$,32)+G$:F$(5)=X$+"KISEBB E       K. "+RIGHT$(S$,
30)+G$
90 F$(6)=X$+"2 U       J E       KEZET":X$="    KE       REM, VA       LASSZON!":PRINTTAB(9)L
EFT$(E$,24)
100 F$(8)="* LUGOSI-SOFT *** COPYRIGHT 1989 ***"
115 W$="":Y$="":L=8:J1=15:J2=0:A=1:B=1:GOSUB900:G=M:GOSUB475:IFQ=8THEN350
145 IFQ=5ANDPEEK(759)>0THENCHAR,0,24," MA'R NEM LEHET!!!" + LEFT$(S$,19):GOSUB680
:RUN9
150 KA=192:KB=60:COLOR4,15,6:COLOR0,2,7:IFQ=7THEN300:ELSEIFQ=1THENX$="D000 D7FF
3800":GOTO190
170 IFQ<4THENX$="D400 D7FF 3C00"
175 IFQ>3THENX$="D000 D3FF 3C00":IFQ=5THENX$="T3800 3BFF 3C00"
180 PRINT"    ";X$:A=1319:B=8:L=14258:GOSUB950:POKE239,9:MONITOR
260 PRINT"    "+CHR$(14):ONOGOSUB2000,3000,4000,5000,6000,7000
280 KK=14:GOSUB485:GOSUB1200:IFQ=1THENPOKE759,1:POKE4101,143
285 GOTO360
300 PRINT"    BENT VAN MA'R A KARAKTERKE/SZLET $3C00- TOL? I/N":GETKEY0$
310 IFQ$="I"THENGOSUB462:X1$="U'J KE'SZLET"+P$:GOTO360:ELSEIFQ$<>"N"THEN300
320 PRINT"    TOLTSÉ MONITORBAN, MA'J INDÍTSÁ    U'JRA A PROGRAMOT!":STOP
350 GOSUB1150:IFQ=8ANDPEEK(65299)=208THENKA=196:KB=208:KK=142:"RE       GI "
360 COLOR0,1:COLOR4,1:COLOR1,8,7:PRINT"    M    ";M$ 2":PRINT"    =====":F$(1)=R1$
364 F$(6)=A6$+"SZLET ADATAI"
365 F$(2)=A2$:F$(7)=B$:F$(9)="STOP":I=9:IFQ=6ANDPEEK(65299)<208THENI=17
370 F$(3)=A$+" -->" + LEFT$(K$,I):F$(4)=K$+" -->" + LEFT$(K$,I):F$(5)=A$+" "+K$:F$(
8)=A6$
380 F$(8)="" : J1=3:J2=7:A=8:B=7:L=9:X$="" : W$="" : Y$="" : J=3:GOSUB910:W=M:IFW=9THEN
END
410 IFW=8THENRUN9
430 ONWGOSUB450,460,500,600,700,800,1210:GOTO360
450 PRINTCHR$(9):F$(10)=X1$:GOSUB1150:U$=" REGI":X1$=""
452 POKE65298,196:POKE65299,208:PRINTCHR$(142):RETURN
460 IFKB=208THENRETURN
462 GOSUB485:POKE65298,KA:POKE65299,KB:PRINTCHR$(KK)
465 ONOGOSUB1000,1000,1000,1050,1050,1100,1150:X1$=F$(10):RETURN
470 W$=CHR$(L):IFW$="ORL=128THENW$=""
471 RETURN
475 S$=""
476 C$="AKSH":G$="+<GOMB>":G$(1)="1 2 3 4 5 6 7 8 9 0 + - ="
477 G$(4)="Z X C V B N M , . /":E$="=====":RE
TURN
480 PRINT"    MINDEN 2.SORBA I'RJAK? I/N"
482 GETKEYW$:J1=1:IFW$="I"THENRETURN:ELSEIFW$<>"N"THEN482
483 J1=2:RETURN
485 PRINTCHR$(8-<Q=1)):RETURN
490 A=16136:B=215:L=14016:GOSUB950:RETURN
500 GOSUB480:C=7:GOSUB960:M=0:B=31:J2=128:GOSUB510:B=159:J2=255:GOSUB510:RETURN
510 DO:M=M+1:A=B+1:B=A+J1*50-1:IFB>J2THENB=J2
515 GOSUB520:LOOPUNTIL(B=J2ORW$="V"):RETURN
520 PRINT"    M    "A$"/";STR$(M):PRINT"    "LEFT$(E$,16):PRINTTAB(37)
/ "
540 J=INT((B-A)/5):FORI=0TOJ:FORK=0TO4:JJ=3:IFK=0THENJJ=5-J1
550 PRINTLEFT$( "    M",JJ):L=A+I+(K*(J+1)):IFL>BTHENPRINT"    ":GOTO570
560 GOSUB470:PRINTW$:PRINTUSING"####";ASC(CHR$(L)):IFL=34THENPRINTW$"";
570 NEXTK,I:GOSUB690:RETURN
600 GOSUB480:C=15:GOSUB960:M=0:B=-1:J2=127:GOSUB610:B=127:J2=255:GOSUB610:RETURN
610 DO:M=M+1:A=B+1:G=J1*50-1:G=G+(G>94)*10:B=A+G:IFB>J2THENB=J2
615 GOSUB620:LOOPUNTIL(B=J2ORW$="V"):RETURN
620 PRINT"    M    "U$K$"/";STR$(M):PRINT"    "LEFT$(E$,24);
630 PRINT"    M    ":PRINTTAB(37):"    ":J=INT((B-A)/5):G=3225
650 FORI=0TOJ:FORK=0TO4:G=G+8:L=A+I+(K*(J+1)):IFL>BTHENPRINT"    ";GOTO670
660 POKEG,L:PRINTUSING"#####";L;
670 NEXTK:IFJ1=1THENPRINT"    M    ":G=G+40
675 NEXTI:GOSUB690:RETURN
680 CHAR,23,24,"ITOVA       B":GETKEYW$:IFW$="T"THENRETURN:ELSE690
690 CHAR,9,24,"ITOVA       B    E       GE":GETKEYW$:IFW$="T"ORW$="V"THENR
ETURN:ELSE690
700 C=11:GOSUB960:B=31:M=0:J2=128:GOSUB710:B=159:J2=255:GOSUB710:RETURN
710 DO:M=M+1:A=B+1:B=A+19:IFB>J2THENB=J2
715 GOSUB740:LOOPUNTIL(B=J2ORW$="V"):RETURN
740 PRINT"    M    "A$ + "K$"/";STR$(M):PRINT"    "LEFT$(E$,36)
742 J=4:IFQ=6ANDPEEK(65299)<208THENJ=8
745 FORI=0TO1:PRINT" ASCII KAR.":LEFT$(K$,J) "    ":NEXT:PRINT"    ";J=INT((B-A)
/2):G=3261
750 FORI=0TOJ:PRINT"    M    ":FORK=0TO1:G=G+20:L=A+I+K*(J+1):IFL>BTHEN780
760 GOSUB470:PRINTUSING"    ";ASC(CHR$(L)):PRINTW$:IFL=34THENPRINTW$"";
770 PRINTUSING"#####";PEEK(G);
780 NEXTK:G=G+40:NEXTI:GOSUB690:RETURN
800 C=10:GOSUB960:GOSUB890
810 F$(2)="EREDETI 1.SET (GRAFIKUS)":F$(3)="EREDETI 2.SET":F$(4)="VISSZA"
820 F$(1)=U$:J1=2:J2=7:A=C:B=7:L=4:X$="" : J=5:W$="" : Y$="" :GOSUB910
830 W=M:IFW=1ANDKB=208THENPRINT"    NINCS    ";U$:GOSUB906:GOTO830
835 ONWGOTO840,845,847,850

```









```

1625 IFK=1THENPRINTTAB(23);"123456789":RETURN:ELSERETURN
1650 FORJ2=3TODSTEP-1:W#=" ":IF(J1AND(2↑J2))=2↑J2THENW#="*"
1655 PRINTW#:NEXT:RETURN
2000 GOSUB490:X1#=""KEZETES KARAKTERK SZLET: #3000-#3-----"
2030 KEY1,"P65298,192:P65299,56"+CHR$(13):KB=56:GOSUB1000:RETURN
3000 GOSUB490:X1#=""KEZETES KISBET+S K SZLET: #3-00-#3-----":GOSUB1000
3030 KEY1,"P65298,192:P65299,60"?CHR$(14);"+CHR$(13):RETURN
4000 GOSUB2000:KB=60:MID$(X1#,31,1)="-"
4030 GOSUB1000:X2#=""IRAFIKUS NAGYBET+S K SZLET: #-000-#3-----":RETURN
5000 A=16144:B=15:L=14024:GOSUB950:A=16216:B=7:L=14096:GOSUB950:A=16264:B=23:L=1
4144
5010 GOSUB950:A=16312:B=23:L=14192:GOSUB950:X1#=""KEZETES NAGY":GOTO6010
6000 X1#=""J KEZETES N.
6010 X1#="X1#+BET+S K SZLET"+P#:GOSUB1050:GOTO3030
7000 A=15576:B=23:L=14232:GOSUB950:X1#=""GRAF.KE SZLET,+2 E KEZETTEL"+P#:GOSUB110
0:RETURN

```

READY.

```

0 REM SAVE"KIS EKEZETES C16" + 1989.01.28
1 REM *****
2 REM * C= UJSAG SORSZAM: 173... *
3 REM *KISEBB MERETU EKEZETES NAGYBETUK*
4 REM * PROGRAM: LUGOSI ANTALNE *
5 REM *****
7 REM
10 POKE56,60:CLR
15 POKE65298,196:POKE65299,208
20 PRINT"XXXXXXXXXXXXXXXXX D3FF 3000"
30 FOR I=1319 TO 1326:READJ:POKEI,J:NEXT
40 POKE239,8
50 MONITOR
60 PRINT"X":GOTO110
70 C=B*8-1+A:FORI=ATOC:READD#:POKEI,DEC(D#):NEXT:RETURN
100 DATA13,88,13,71,207,54,48,13
110 PRINT"AZ ENNE NYGY A ENNE KE SZLETET KE R1?"
115 PRINT"XXXXXXXXX1.KE SZLET: 5*5 ME RET"
116 PRINT"XXXXXXXXX2.KE SZLET: 6*6 ME RET"
120 GETKEYQ#:IFQ#="2"THENRESTORE2000:ELSEIFQ#<"1"THEN120
130 PRINT"XXXXXXXXX D O L G O Z O M !!!"
210 A=DEC("3000"):B=30:GOSUB70:REM KEP.KOD: 0-29
220 A=DEC("3088"):B= 3:GOSUB70:REM KEP.KOD: 33-35
230 A=DEC("3038"):B= 5:GOSUB70:REM KEP.KOD: 39-43
240 A=DEC("3068"):B=19:GOSUB70:REM KEP.KOD: 45-63
255 IFQ#="2"THENRESTORE1090
260 A=16144: B= 2:GOSUB70:REM KEP.KOD: 98-99
270 A=16216: B= 1:GOSUB70:REM KEP.KOD:107
280 A=16264: B= 3:GOSUB70:REM KEP.KOD:113-115
290 A=16312: B= 3:GOSUB70:REM KEP.KOD:119-121
300 KEY1,"POKE65298,192:POKE65299,60"?CHR$(14);"+CHR$(13)
310 KEY2,"POKE65298,196:POKE65299,208"+CHR$(13)
320 POKE65298,192:POKE65299,60:PRINTCHR$(14)
330 PRINT"J.":Q#: ". KEZETES KARAKTERK SZLET;"
340 PRINT"KISEBB M RET BET KKEKEL: #3000-#3FFF"
350 END
999 REM 1.KESZLET ADATAI

```

```

1000 DATA0,30,66,6E,68,62,30,0:REM @
1001 DATA0,00,30,66,7E,66,66,0:REM A
1002 DATA0,00,70,66,70,66,70,0:REM B
1003 DATA0,00,30,68,68,68,30,0:REM C
1004 DATA0,00,70,66,66,66,70,0:REM D
1005 DATA0,00,70,68,70,68,7E,0:REM E
1006 DATA0,00,7E,68,70,68,68,0:REM F
1007 DATA0,00,3E,68,66,62,3E,0:REM G
1008 DATA0,00,66,66,7E,66,66,0:REM H
1009 DATA0,00,30,18,18,18,30,0:REM I
1010 DATA0,00,00,00,00,60,38,0:REM J
1011 DATA0,00,66,60,78,60,66,0:REM K
1012 DATA0,00,68,68,68,68,7E,0:REM L
1013 DATA0,00,63,77,68,63,63,0:REM M
1014 DATA0,00,66,76,7E,6E,66,0:REM N
1015 DATA0,00,30,66,66,66,30,0:REM O
1016 DATA0,00,70,66,70,68,68,0:REM P
1017 DATA0,00,30,66,66,30,66,0:REM Q
1018 DATA0,00,70,66,70,68,66,0:REM R
1019 DATA0,00,30,68,30,66,30,0:REM S
1020 DATA0,00,7E,18,18,18,18,0:REM T
1021 DATA0,00,66,66,66,66,30,0:REM U
1022 DATA0,00,66,66,24,30,18,0:REM V
1023 DATA0,00,63,63,68,7F,63,0:REM W
1024 DATA0,00,63,36,10,36,63,0:REM X
1025 DATA0,00,66,66,30,18,18,0:REM Y
1026 DATA0,00,7E,00,18,38,7E,0:REM Z
1027 DATA0,00,30,38,38,38,30,0:REM [
1028 DATA0,12,38,70,38,62,70,0:REM \
1029 DATA0,00,30,00,00,00,30,0:REM ]
1030 DATA0,00,18,18,18,00,18,0:REM !
1034 DATA0,00,66,66,66,00,00,0:REM "
1035 DATA0,00,66,FF,66,FF,66,0:REM #
1039 DATA0,00,06,0C,18,00,00,0:REM ^

```

```

1040 DATA0,0C,18,38,38,18,0C,0:REM <
1041 DATA0,38,18,0C,0C,18,38,0:REM >
1042 DATA0,00,66,3C,FF,3C,66,0:REM *
1043 DATA0,00,18,18,7E,18,18,0:REM +
1045 DATA0,00,00,00,7E,00,00,0:REM -
1046 DATA0,00,00,00,00,18,18,0:REM .
1047 DATA0,00,06,0C,18,38,68,0:REM /
1048 DATA0,00,30,66,7E,66,30,0:REM 0
1049 DATA0,00,18,38,18,18,30,0:REM 1
1050 DATA0,00,30,66,0C,18,7E,0:REM 2
1051 DATA0,00,30,66,0C,66,30,0:REM 3
1052 DATA0,00,10,30,60,7E,0C,0:REM 4
1053 DATA0,00,70,68,70,06,70,0:REM 5
1054 DATA0,00,30,68,70,66,30,0:REM 6
1055 DATA0,00,7E,46,0C,18,38,0:REM 7
1056 DATA0,00,30,66,30,66,30,0:REM 8
1057 DATA0,00,30,66,3E,06,30,0:REM 9
1058 DATA0,00,00,18,00,00,18,0:REM :
1059 DATA0,00,00,18,00,18,18,0:REM ;
1060 DATA0,00,0C,18,38,18,0C,0:REM <
1061 DATA0,00,00,7E,00,7E,00,0:REM =
1062 DATA0,00,38,18,0C,18,38,0:REM >
1063 DATA0,00,30,66,0C,00,18,0:REM ?
1068 DATA0,00,30,18,18,18,30,0:REM I'
1099 DATA36,24,30,66,66,66,30:REM O"
1107 DATA0C,08,30,66,7E,66,66,0:REM A'
1113 DATA0C,08,7E,68,70,66,66,0:REM E'
1114 DATA36,24,08,66,66,66,30:REM U"
1115 DATA24,08,66,66,66,66,30:REM U'
1119 DATA24,08,30,66,66,66,30:REM 0:
1120 DATA0C,08,66,66,66,66,30:REM U'

```



```

1121 DATA0,00,30,66,66,66,30,0:REM D
1999 REM 2.KESZLET ADAT1
2000 DATA0,30,66,6E,60,62,30,0:REM B
2001 DATA0,30,66,66,7E,66,66,0:REM A
2002 DATA0,70,66,70,66,66,70,0:REM B
2003 DATA0,30,66,60,60,60,30,0:REM C
2004 DATA0,78,60,66,66,70,78,0:REM D
2005 DATA0,7E,60,78,60,60,7E,0:REM E
2006 DATA0,7E,60,70,60,60,60,0:REM F
2007 DATA0,30,66,60,6E,66,30,0:REM G
2008 DATA0,66,66,66,7E,66,66,0:REM H
2009 DATA0,30,18,18,18,18,30,0:REM I
2010 DATA0,00,00,00,00,60,38,0:REM J
2011 DATA0,66,60,78,78,60,66,0:REM K
2012 DATA0,60,60,60,60,60,7E,0:REM L
2013 DATA0,63,77,60,63,63,63,0:REM M
2014 DATA0,66,7E,6E,6E,66,0:REM N
2015 DATA0,30,66,66,66,66,30,0:REM O
2016 DATA0,70,66,70,60,60,60,0:REM P
2017 DATA0,30,66,66,66,30,66,0:REM Q
2018 DATA0,70,66,70,78,60,66,0:REM R
2019 DATA0,30,60,30,66,66,30,0:REM S
2020 DATA0,7E,18,18,18,18,0:REM Y
2021 DATA0,66,66,66,66,66,30,0:REM U
2022 DATA0,66,66,66,24,30,18,0:REM V
2023 DATA0,63,63,63,60,7F,63,0:REM W
2024 DATA0,66,30,18,30,66,66,0:REM X
2025 DATA0,66,66,30,18,18,18,0:REM Y
2026 DATA0,7E,06,00,38,60,7E,0:REM Z
2027 DATA0,30,30,30,30,30,30,0:REM C
2028 DATA0C,12,30,70,30,62,70,0:REM E
    
```

```

2029 DATA0,30,00,00,00,00,30,0:REM J
2033 DATA0,18,18,18,18,00,18,0:REM I
2034 DATA0,66,66,66,00,00,00,0:REM "
2035 DATA0,66,66,FF,66,FF,66,0:REM #
2039 DATA0,06,00,18,00,00,00,0:REM ^
2040 DATA0,00,18,30,30,18,00,0:REM <
2041 DATA0,30,18,00,00,18,30,0:REM >
2042 DATA0,00,66,30,FF,30,66,0:REM *
2043 DATA0,00,18,18,7E,18,18,0:REM +
2045 DATA0,00,00,00,7E,00,00,0:REM -
2046 DATA0,00,00,00,00,18,18,0:REM /
2047 DATA0,03,06,00,18,30,60,0:REM \
2048 DATA0,30,66,6E,76,66,30,0:REM 0
2049 DATA0,18,38,18,18,18,30,0:REM 1
2050 DATA0,30,66,00,30,60,7E,0:REM 2
2051 DATA0,30,06,10,06,66,30,0:REM 3
2052 DATA0,10,30,60,7E,00,00,0:REM 4
2053 DATA0,7E,60,70,06,46,70,0:REM 5
2054 DATA0,30,60,60,70,66,30,0:REM 6
2055 DATA0,7E,66,00,18,18,18,0:REM 7
2056 DATA0,30,66,30,66,66,30,0:REM 8
2057 DATA0,30,66,3E,06,66,30,0:REM 9
2058 DATA0,00,00,18,00,00,18,0:REM :
2059 DATA0,00,00,18,00,18,18,30:REM ;
2060 DATA0,00,00,18,30,18,00,0:REM <
2061 DATA0,00,00,7E,00,7E,00,0:REM =
2062 DATA0,00,30,18,00,18,30,0:REM >
2063 DATA0,30,66,00,18,00,18,0:REM ?
READY.
    
```

# A DIRECTORY GYORSAN, ÁTTEKINTHETŐEN

A programjainkba írt, a lemez tartalomjegyzékét programvesztés nélkül kiolvasó rutin nem kell hogy bonyolult legyen. Gyakran azonban ezek a rutinok túl lassan dolgoznak, főleg, ha sok bejegyzést tartalmaz a Directory. Ekkor ugyanis az első feliratokat odébb kell görgetni. Ugyanez érvényes a lemez nevére vagy ID-jére is, ha azokra gyakran szükség van. A beolvasáskor mindjárt ez az a két információ, ami elsőre eltűnik. Ezek miatt írtam egy saját rutint erre a célra (1. lista).

A dolog elméletileg ugyanúgy működik, mint a számos, már korábban leközölt Directory beolvasó rutin. Az enyém azonban namcsak a gyors olvasásra, de az áttekinthető kijelzésre is figyel.

Utóbbit pedig úgy érjük el, hogy a tartalomjegyzéket oldalakra osztjuk föl 20-20 bejegyzéssel (1. oldal: fejléc és 29 bejegyzés stb.) A fejléc minden esetben tartalmazza a lemez nevét és az ID-t.

A kivitelt a ← gombbal lehet befejezni, vagy bármely más gombbal tovább lapozhatunk. A legutoljára megnyomott gombot a GET A\$-ral kérdezzük le.

De térjünk vissza a programhoz. A demot (2. lista) a következő utasítással tölthetjük be:  
LOAD "DEMO",8

Az indítás RUN <RETURN>-nel történhet. A program behívja a szükséges gépi kódú részt, s gombnyomásra kijelzi a meghajtóba behelyezett lemez tartalomjegyzékét. A meghajtó számát 8 és 15 között változtathatjuk az L változóban. A rutint az alábbi módon használhatjuk saját programjainkban (a demoprogram már tartalmazza összehasonlítási célokra a fontos lépéseket):

1. A rutint a programunkban lehetőleg minél korábban (az éiején) olvassuk be (30. sor).

2. A felhívás előtt (vagy a teljes programunkban) ne használjuk a kisbetű/nagybetű karakterkészletet (40. sor).

3. A rutin szintaxisa SYS 49152. Ez előtt módosítani lehet a meghajtó egység számát a POKE 49155,érték (8–15) paranccsal.

4. A felhívás után a legutoljára megnyomott billentyűt kérdezzük le GET A\$-ral, vagy azt elimináljuk POKE 198,0-val.

Még egy tipp a gyakorlottabb felhasználóknak: A felhívás után a lemez neve a 49423–49437 (\$C10F–\$C11E), az ID pedig a 49440–49444 (\$C121–\$C125) címeken áll.

```

1 rem *****
2 rem"* C= Ujsag          Sorszam:174 *
3 rem"* GYORS DIRECTORY *
4 rem"* Program:        Kovacs Zoltan *
5 rem *****
6 :
7 :
8 print chr$(147)"data beolvasas es ellenorzes ...";:j=49152:ve=49423:p=j
9 for b=0 to 7:read a#
10 l=asc(mid$(a#,2,1))
11 h=asc(mid$(a#,1,1))
12 l=l-48:if l>9 then l=l-7
13 h=h-48:if h>9 then h=h-7
14 print"#####"p;:p=p+1
15 if h>15 or l>15 then 17
16 a=h*16+l:poke j+b,a:t=t+a:next b:read a:if a=t then 18
17 print:print"data hiba ... sor:"peek(64)*256+peek(63):end
18 t=0:j=j+8:if j<ve then 9
19 print"###kesz":end
    
```





```

30 if a=0 then a=1:load"fast directory",8,1
40 printchr$(8);chr$(14)
50 l=8:poke 49155,1
60 rem *****
61 rem * ures file-ok felirasa *
62 rem *****
65 print"File-okat felirjam (i/n)?":wait 198,255:get a$:if a$="n" then 120
66 rem ::: clr ::: crsr le ::: crsr jobbra :::
70 print"Kis turelmet - felirom a file-okat..."
71 rem ::: clr ::: crsr le ::: crsr jobbra :::
80 for a=1 to 40:print" ";40-a;" "
81 rem ::: home ::: 3*crsr le ::: crsr jobbra :::: crsr balra :::
90 f$="file"+str$(a)+"w,p":open 1,8,2,f$:close 1:next
100 rem *****
101 rem * directory meghivas *
102 rem *****
120 print"Gombnyomasra - Directory"
121 rem ::: clr ::: 5*crsr le ::: 5*crsr jobbra :::
130 wait 198,255:get a$
140 sys 49152:get a$
150 rem a$=" ←" megszakitja a directory hivast
160 goto 120

ready.

```

20 data a9,01,a2,08,a0,00,20,ba,0718	38 data c0,4c,2b,c0,20,44,e5,a2,0994
21 data ff,a9,01,a2,0e,a0,c1,20,0986	39 data 07,bd,f8,c0,20,d2,ff,ca,1335
22 data bd,ff,20,c0,ff,a2,01,8e,1228	40 data d0,f7,ae,3d,03,a9,00,20,0894
23 data 3d,03,20,c6,ff,a9,00,8d,0859	41 data cd,bd,a2,08,bd,ff,c0,20,1232
24 data 3c,03,20,e3,c0,20,94,c0,0886	42 data d2,ff,ca,d0,f7,a2,00,bd,1473
25 data 4c,3f,c0,20,cf,ff,20,cf,1064	43 data 0f,c1,e8,20,d2,ff,e0,10,1177
26 data ff,a9,40,c5,90,f0,0e,a9,1252	44 data d0,f5,a2,06,bd,07,c1,20,1042
27 data 0d,20,d2,ff,20,5c,c0,20,0858	45 data d2,ff,ca,d0,f7,a2,00,bd,1473
28 data 51,c0,4c,67,c0,20,74,c0,0984	46 data 21,c1,e8,20,d2,ff,e0,05,1184
29 data 20,cc,ff,a9,01,20,c3,ff,1143	47 data d0,f5,ee,3d,03,a9,0d,20,0969
30 data 60,20,cf,ff,20,d2,ff,c9,1288	48 data d2,ff,60,a2,08,20,cf,ff,1225
31 data 00,d0,f6,60,20,cf,ff,aa,1214	49 data ca,d0,fa,a2,00,20,cf,ff,1316
32 data 20,cf,ff,20,cd,bd,60,ae,1190	50 data 9d,0f,c1,e8,e0,17,d0,f5,1297
33 data 3c,03,e0,13,f0,11,ee,3c,0861	51 data 60,20,12,3a,45,47,41,d0,0617
34 data 03,4c,2b,c0,a9,00,85,c6,0814	52 data 12,3a,4b,53,49,c4,20,92,0681
35 data a9,01,c5,c6,d0,fc,60,20,1153	53 data 12,3a,c4,c9,20,92,24,00,0687
36 data 74,c0,ad,77,02,c9,5f,f0,1138	
37 data bf,a9,00,8d,3c,03,20,94,0744	ready.

# AUTOSTART

Az Auto-Start.bas program egy autostart generátor, amely a lemezre ír egy mintegy 120 byte-nyi programot. Ez maximum két újabb állományt hív be és indít el automatikusan. Például az első program egy floppygyorsító lehet, vagy valami hasonló. Nincs szükség a kézzel történő betöltésre és indításra, mivel az a tulajdonképpeni főprogram indításával együtt megtörténik.

Adjuk be a listát és a letárolás után indítsuk el a RUN-nal. Azonnal megjelenik a kérdés, mi legyen az autostart program neve. Ezt később ,8,1-gyel kell behívni. Adjuk be tehát a kívánt nevet, például AUTO. Az ezt követő <RETURN> után az elsőként betöltendő program nevére kérdez a program. A kipróbálás kedvéért adjuk be az autostart generátor programunk AUTO-START.BAS nevét. A számítógép ezután arra a beugrási



címre vár, amelyre a második program elindítása előtt lépünk. Szoftveres floppygyorsító esetében például így kihasználhatjuk azt a második program betöltéséhez. Mivel a mi esetünkben egy egyszerű BASIC programról van szó, adjuk be itt a 43121-es címet, amely a RUN 0 BASIC parancs gépi kódú megfelelője. Ebből az következik, hogy a kezelendő BASIC programjainknak mindig rendelkezniük kell egy nulladik sorral, még ha az a 0 REM is.

A nulladik sorra akkor nincs szükség, ha a BASIC programot másodikként, azaz egy gépi kódú programként hívjuk be. A gépi kódú programok indító címeit legtöbbször az első és egyetlen kilistázható BASIC sorban találjuk egy SYS utasítás mögött. A rutinnak logikusan egy RTS-sel kell végződnie, hogy a második program utántöltése is megtörténhessen.

A beugrasi cím megadása után ugyanezt a procedúrát a névvel és a címmel meg kell ismételnünk a második programmal is. Az utolsó kérdésnél egy BASIC program kezdéséhez szükséges 43121-es értéket mindjárt meg is ajánlunk. A mi esetünkben nincs másik program, ezért csak az első kérdést kell <RETURN>-nel visszaigazolni, és máris kezdődhet a tárolási

munka az autostart állomány elkészítésére. Ha behívjuk a lemezünk tartalomjegyzékét ezek után, ott egy egyblokkos AUTO nevű állományt kell találnunk, amely mögött megtaláljuk a ,8,1-et. Így azután rettentően könnyű ezt a programot a Directoryból behívni (LOAD a név elé és <RETURN>, máris kész).

A működési elv nem különösebben komplex. Az autostart program a tárolóban a \$01E0-\$0247 területen áll, vagyis az a processzorstack közepén kezdődik. A hely korlátozott, ezért a két betöltendő program neve összesen nem lehet több, mint 17 karakter! Adott esetben nevezzük át programjainkat. Ha behívjuk az autostart állományt, azaz a mi esetünkben az AUTO-t, akkor ezzel fölülírjuk a stacket. A betöltést követően a főprogramba (ebben az esetben ez a BASIC interpreter) való ugráshoz igyekszünk megszerezni a stackből a megfelelő beugrasi címet, de ekkor gondosan elkészített programocskára bukkanunk. Az autostarter első része ugyanis néhány \$02-es byte-ból áll, amely arra ösztönzi a CPU-t, hogy a (\$0202+\$01)-es címre ugorjon. Itt áll azonban a mi kis betöltő rutinunk, és ekkor minden a maga előre kijelölt útján halad tovább.

```

0 rem
1 rem *****
2 rem"* C= Ujsag          Ssz.:175  *
3 rem"*          AUTO START GENERATOR      *
4 rem"* Program:          Kovacs Zoltan *
5 rem *****
6 :
7 :
10 a1=-1:a2=-1
20 print chr$(14)"Kerem megadni az autostart"
21 rem ::: clr ::: crsr le :::
40 print" program nevet:";
41 rem ::: crsr le :::
50 input fi$
60 if fi$="" then print"file-nev hiba !!":end
70 fi$=left$(fi$,10)+" #8#1#"
71 rem ::: comm d ::: comm d ::: shift @ :::
80 :
90 print" es most az elsokent betoltendo"
91 rem ::: 2*crsr le :::
100 print" file nevet:";
101 rem ::: crsr le :::
110 input f1$
120 if f1$="" then print"file-nev hiba !!":end
130 :
140 input" Beugrasi cim ";a1
141 rem ::: 2*crsr le :::
150 :
160 print" most a masodszorra betoltendo"
161 rem ::: 2*crsr le :::
170 print" program nevet:";
171 rem ::: crsr le :::
180 input f2$
190 if len(f2$)=0 and a1=-1 then print"file-nev hiba !!":end
200 :
210 if len(f2$)=0 then 250
220 input" Beugrasi cim (BASIC) 43121";a2
221 rem ::: 2*crsr le ::: 7*crsr balra :::
230 if len(f2$)>0 and a2=-1 then print"Cim hiba !!":end
240 :
250 open 1,8,2,fi$+","p,w"

```



```

○ 260 open 2,8,15 ○
270 : ○
○ 280 input#2,fe,fe$ ○
290 if fe=63 then print"Autostart-file mar letezik !":goto 680 ○
291 rem ::: 2*crsr le ::: ○
○ 300 if fe>0 then print fe,fe$:goto 680 ○
310 : ○
○ 320 rem *** regi file kicserelere ? *** ○
330 print#2,"r:"+f1$+"="+f1$ ○
○ 340 input#2,fe,fe$ ○
350 if fe<>63 then print fe,fe$:goto 680 ○
360 : ○
○ 370 if f2$="" then 430 ○
380 print#2,"r:"+f2$+"="+f2$ ○
○ 390 input#2,fe,fe$ ○
400 if fe<>63 then print fe,fe$:goto 680 ○
○ 410 : ○
420 rem *** adatok beolvasasa *** ○
○ 430 pr=0:t=0:restore ○
440 read a:if a>=0 then poke 828+t,a:t=t+1:pr=pr+a:goto 440 ○
○ 450 if pr<>7983 then print"Hibas ellenorzoszam !!!":goto 680 ○
460 : ○
○ 470 if a1=-1 then 510 ○
480 hi=int(a1/256):lo=a1-256*hi ○
490 poke 893,lo:poke 894,hi ○
○ 500 : ○
510 if a2=-1 then poke 892,44:a2=a1 ○
○ 520 : ○
530 hi=int(a2/256):lo=a2-256*hi ○
○ 540 poke 930,lo:poke 931,hi ○
550 : ○
○ 560 poke 901,72+len(f1$) ○
570 poke 860,len(f1$):poke 861,len(f2$) ○
580 : ○
○ 590 for a=1 to len(f1$) ○
600 poke 931+a,asc(mid$(f1$,a,1)):next ○
○ 610 : ○
620 if len(f2$)=0 then 650 ○
○ 630 for b=1 to len(f2$) ○
640 poke 930+a+b,asc(mid$(f2$,b,1)):next ○
○ 650 print#1,chr$(224);chr$(1); ○
660 for d=828 to 931+a+b ○
○ 670 print#1,chr$(peek(d));:next ○
680 close 1:close 2:end ○
690 : ○
○ 700 data 2,2,2,2,2,2,2,2 ○
710 data 2,2,2,2,2,2,2,2 ○
○ 720 data 2,2,2,2,2,2,2,2,2,2 ○
730 data 2,2,2,2,174,167,0,0,96 ○
○ 740 data 169,00,32,144,255,169,1 ○
750 data 162,8,160,1,32,186,255 ○
○ 760 data 173,0,2,162,72,160,2,32 ○
770 data 189,255,169,0,32,213,255 ○
○ 780 data 32,2,2,173,1,2,240,12,162 ○
790 data 80,160,2,32,189,255,169,0 ○
○ 800 data 32,213,255,32,51,165,165 ○
810 data 23,134,45,133,46,32,96,166 ○
○ 820 data 162,253,154,169,0,76 ○
830 data 113,168,-1 ○

○ ready. ○

```



# TIZENHAT IGAZI SPRITE

Miután találkoztam olyan programokkal, amelyek a képernyőre több száz sprite-ot varázsolnak, amelyeket sehogyan sem lehet mozgásra készíteni, én végre egy olyat akartam fejleszteni, amelyik használható. Ilyen program a Multi 16.

Bár csak tizenhat sprite áll a rendelkezésünkre, de ezek kezelése korlátozások nélkül megvalósítható. A program mind a 16 sprite számára lehetővé teszi a szín, a pozíció, a multicolor mód és az X/Y nagyítás használatát.

A Multi 16 a \$CE00 címtől kezdve áll a tárolóban. Az indításhoz választhatunk a SYS 52992 (csak az IRQ-t indítani) és a SYS

52995 (az „új” VIC-et is inicializálni) között. A program kikapcsolása a SYS 52998-cal történhet. Ekkor a VIC-et a kiindulási állapotba visszük. A \$CF00-tól állnak az 1–8. sprite-ok regiszterei, a \$CF30-tól pedig a 9–16. számúakéi. A foglaltság egy az egyben megfelel az eredeti VIC regiszterek foglaltságainak. De csak a \$D000–\$D010, a \$D015, a \$D017, a \$D01B–\$D01F és a \$D025–\$D02F cellákat vesszük figyelembe. A többi VIC regisztert továbbra is használni lehet az eredeti funkcióknak megfelelően. A \$CF60-tól és a \$CF70-tól kell állniuk a sprite mutatóknak, amelyeket mindig eltolunk a \$07F8-hoz. A képernyőt tehát nem szabad más helyre vinni. Különben minden más megengedett (például megváltoztatott jelkészletek). Felhasználjuk a raszterinterrupt trükköt. A VIC-et egyszer a \$CF00-nál, egyszer pedig a \$CF30-nál jelezzük. Emiatt a sprite-ok villódnak kissé, főleg világos háttér előtt.

```

1 rem *****
2 rem"* C= Ujsag          Sorszam:176 *
3 rem"*          MULTI 16          *
4 rem"* Program:      Kovacs Zoltan *
5 rem *****
6 :
7 :
8 print chr$(147)"data beolvasas es ellenorzes ...";:j=52736:ve=52983:p=j
9 for b=0 to 7:read a$
10 l=asc(mid$(a$,2,1))
11 h=asc(mid$(a$,1,1))
12 l=l-48:if l>9 then l=l-7
13 h=h-48:if h>9 then h=h-7
14 print"#####"p;:p=p+1
15 if h>15 or l>15 then 17
16 a=h*16+l:poke j+b,a:t=t+a:next b:read a:if a=t then 18
17 print:print"data hiba ... sor:"peek(64)*256+peek(63):end
18 t=0:j=j+8:if j<ve then 9
19 print"#####kesz":end
20 data 4c,29,ce,4c,09,ce,4c,d8,0906
21 data ce,a0,00,b9,00,d0,99,00,0912
22 data cf,99,30,cf,c8,c0,2f,d0,1262
23 data f2,a0,00,b9,f8,07,99,60,1091
24 data cf,99,70,cf,c8,c0,08,d0,1287
25 data f2,78,a0,00,b9,00,d0,99,1068
26 data 80,cf,c8,c0,30,d0,f5,a9,1397
27 data 5a,8d,14,03,a9,ce,8d,15,0791
28 data 03,a9,81,8d,1a,d0,a9,01,0846
29 data 8d,12,d0,ad,11,d0,29,7f,0933
30 data 8d,11,d0,a9,00,8d,d6,ce,1096
31 data 58,60,ad,19,d0,8d,19,d0,0964
32 data 30,07,ad,0d,dc,58,4c,31,0674
33 data ea,ad,d6,ce,f0,05,a2,30,1282
34 data 4c,75,ce,a2,00,8e,d7,ce,1124
35 data a0,00,bd,00,cf,99,00,d0,0917
36 data e8,c8,c0,11,d0,f4,ae,d7,1482
37 data ce,bd,15,cf,8d,15,d0,bd,1182
38 data 17,cf,8d,17,d0,a0,1b,bd,0978
39 data 1b,cf,99,00,d0,e8,c8,c0,1219
40 data 20,d0,f4,ae,d7,ce,a0,25,1276
41 data bd,25,cf,99,00,d0,e8,c8,1226
42 data c0,2f,d0,f4,a0,f8,ae,d7,1488
43 data ce,e0,00,f0,02,a2,10,bd,1039
44 data 60,cf,99,00,07,e8,c8,c0,1087
45 data 00,d0,f4,ad,d6,ce,49,01,1119
46 data 8d,d6,ce,4c,7e,ea,00,00,0997
47 data 78,a9,31,8d,14,03,a9,ea,0905
48 data 8d,15,03,a9,00,8d,1a,d0,0709
49 data a0,00,b9,80,cf,99,00,d0,1041
50 data c8,c0,30,d0,f5,58,60,79,1198
ready.
    
```



```

○ 10 rem *****
○ 11 rem"* C= Ujsag          Sorszam:177 *
○ 12 rem"*          MULTI  16  DEMO          *
○ 13 rem"* Program:      Kovacs Zoltan *
○ 14 rem *****
○ 15 :
○ 16 rem"          DEMO
○ 17 :
○ 18 if peek(52736)<>76 then load"multi 16",8,1
○ 19 sys 58648:sys 64789:sys 64931
○ 20 poke 53281,..:poke 53280,..
○ 23 for t=12864 to 13952:poke t,0:next
○ 25 ba=49152+15*256 : rem a vic-multi 16 baziscime
○ 30 v1=ba:v2=ba+3*16: rem uj vic cimek
○ 35 z1=ba+6*16:z2=ba+7*16: rem spritepointer cimek
○ 40 sa=49152+14*256:rem kezdocim
○ 50 print"Éz a programm 16 sprite-ot képes egy-"
○ 51 rem ::: clr ::: ctrl 2 :::
○ 55 print"szerre egy kepernyon megjeleníteni,"
○ 60 print"akar mind a 16-ot egymás mellett is."
○ 65 print"egy kis turelmet kerek!":poke 56334,0:poke 1,51:print"É"
○ 66 rem ::: 2*crsr le :::
○ 70 for t=0 to 15:po=201*64+t*64
○ 75 nu$=right$("0"+right$(str$(t+1),len(str$(t+1))-1),2)
○ 80 a1=asc(left$(nu$,1)):a2=asc(right$(nu$,1)):print"É"nu$
○ 81 rem ::: crsr fel :::
○ 85 for r=0 to 7
○ 90 poke po+r*3,peek(a1*8+53248+r)
○ 95 poke po+r*3+1,peek(a2*8+53248+r)
○ 100 next r
○ 105 next t
○ 110 poke 1,55:poke 56334,1
○ 113 sys sa+3:rem start (inicializalassal)
○ 115 print"Émost egymásutan bekapcsoljuk a 16 "
○ 116 rem ::: clr :::
○ 120 print"sprite-ot":for t=0 to 7:poke z1+t,201+t:next
○ 123 for t=0 to 7:poke z2+t,209+t:next
○ 125 for t=1 to 15 step 2:poke v1+t,100:next
○ 130 a=20:for t=0 to 14 step 2:poke v1+t,a:a=a+20:next
○ 135 for t=1 to 15 step 2:poke v2+t,100:next
○ 140 for t=0 to 14 step 2:poke v2+t,a:a=a+20:if a>255 then
○ 145 if a>255 then a=a-256:poke v2+16,240
○ 150 next
○ 155 l=0:for t=0 to 7:l=1+2~t:poke v1+21,l
○ 160 for r=1 to 1000:next r:next t
○ 165 l=0:for t=0 to 7:l=1+2~t:poke v2+21,l
○ 170 for r=1 to 1000:next r:next t
○ 175 for t=1 to 15 step 2:for r=100 to 200 step 1:poke v1+t,int(r):next r,t
○ 180 for t=1 to 15 step 2:for r=100 to 200 step 1:poke v2+t,int(r):next r,t
○ 190 for t=39 to 46:poke v1+t,1:for r=1 to 300:next r,t
○ 195 for t=39 to 46:poke v2+t,1:for r=1 to 300:next r,t
○ 200 for t=39 to 46:poke v1+t,2:for r=1 to 300:next r,t
○ 205 for t=39 to 46:poke v2+t,2:for r=1 to 300:next r,t
○ 210 for t=1 to 4000:next:run

```

ready.



# PROFI DESIGN

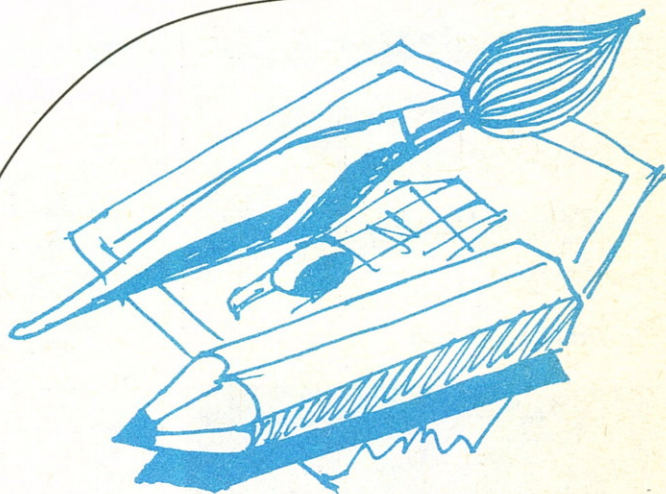
A C64-es esetében is egyre több a felhasználói felületeket használó program, amelyek fölöslegessé teszik a billentyűzetről történő bevitelt. Ezek helyett joystickkal vagy egy egérrel kell egy mutató nyilat a menüpontokra vezetni. Gombnyomásra azután a megfelelő alprogramot aktiváljuk.

Aki járatos az Assembler programozásban, annak ez nem probléma, a BASIC-ben programozók azonban törhetik a fejüket, ha ilyen komfortot akarnak beépíteni programjaikba. De nem kell megírni azt, ami már kész. A lista tartalmazza a szükséges alprogramokat egy, a 2-es portba csatlakoztatott joystick BASIC lekérdezéséhez. A nyíl mozgása azután teljesen automatikusan történik. A 4-es és az 5-ös listák mutatják szemléletesen a gépi kódú rutinok használatát. Most nézzük a szintaxist: SYS 49152,m,sz,s

```

1 rem *****
2 rem"* C= Ujsag          Sorszam:178  *
3 rem"*          EGER  64          *
4 rem"* Program:      Kovacs Zoltan *
5 rem *****
6 :
7 :
8 print chr$(147)"data beolvasas es ellenorzes ...";:j=49152:ve=49431:p=j
9 for b=0 to 7:read a#
10 l=asc(mid$(a$,2,1))
11 h=asc(mid$(a$,1,1))
12 l=l-48:if l>9 then l=l-7
13 h=h-48:if h>9 then h=h-7
14 print"#####"p;:p=p+1
15 if h>15 or l>15 then 17
16 a=h*16+l:poke j+b,a:t=t+a:next b:read a:if a=t then 18
17 print:print"data hiba ... sor:"peek(64)*256+peek(63):end
18 t=0:j=j+8:if j<ve then 9
19 print"#####kesz":end
20 data 20,9b,b7,e0,00,f0,05,e0,1063
21 data 01,f0,0e,60,78,a9,31,8d,0830
22 data 14,03,a9,ea,8d,15,03,58,0679
23 data 60,20,29,c0,78,a9,61,8d,0888
24 data 14,03,a9,c0,8d,15,03,58,0637
25 data 60,ea,20,9b,b7,8e,27,d0,1089
26 data 20,9b,b7,86,02,a9,0d,8d,0829
27 data f8,07,a9,28,8d,00,d0,a9,0982
28 data 37,8d,01,d0,a9,00,8d,10,0731
29 data d0,a2,00,bd,d3,c0,9d,40,1183
30 data 03,e8,e0,3f,d0,f5,ad,15,1169
31 data d0,09,01,8d,15,d0,60,07,0691
32 data 13,a4,02,d0,03,4c,31,ea,0755
33 data ad,00,dc,4a,b0,0a,ae,01,0828
34 data d0,e0,32,f0,03,ce,01,d0,1140
35 data 4a,b0,0a,ae,01,d0,e0,f9,1116
36 data f0,03,ee,01,d0,4a,b0,23,0975
37 data 85,03,ad,10,d0,29,01,f0,0815
38 data 0e,ae,00,d0,d0,03,ce,10,0829
39 data d0,ce,00,d0,4c,cf,c0,ae,1271
40 data 00,d0,e0,18,f0,29,ce,00,0943
41 data d0,a5,03,4a,b0,21,ad,10,0848
42 data d0,29,01,f0,0d,ae,00,d0,0885
43 data e0,57,f0,13,ee,00,d0,4c,1092
44 data cf,c0,ae,00,d0,e0,ff,d0,1468
45 data 03,ee,10,d0,ee,00,d0,88,1047
46 data 4c,63,c0,80,00,00,e0,00,0719
47 data 00,78,00,00,3f,00,00,1f,0214
48 data e0,00,1f,fc,00,0f,f8,00,0770
49 data 07,f0,00,07,f0,00,03,f8,0745
50 data 00,03,fc,00,01,8e,00,01,0399
51 data 06,00,00,00,00,00,00,00,0006
52 data 00,00,00,00,00,00,00,00,0000
53 data 00,00,00,00,00,00,00,00,0000
54 data 00,00,60,32,45,ff,00,ff,0725

ready.
    
```



## EGER-MUTATO DEMO 1

```

100 rem -----,-----,-----
105 rem"----- Egermutato-demo 01' -----
115 rem -----
120 :
125 :
130 if a=0 then a=1:load "eger 64",8,1
135 sys 49152,1,0,2
140 poke 53280,15:poke 53281,15
145 poke 646,11
150 print chr$(147);chr$(14)
155 :
160 x=peek(53248)-24+256*(peek(53264)and 1)
165 y=peek(53249)-50
170 t=peek(56320) and 16
175 if t=0 then a$="Tuz":goto 185
180 a$=""
185 printchr$(147);x,y,a$
190 goto 160
    
```

ready.



Ez a programban csak egyszer szerepeljen. Az „m” a módot jelenti, értéke 0 és 1 lehet. Előbbi az egér-, utóbbi a joystick-mutatóhoz való. Az „sz” paraméter rögzíti a mutató nyíl színét, az értelemszerűen 0 és 15 közé eshet. Az „s” a sebességet meghatározó adat 0 és 255 között. A kisebb érték a lassúbb sebességet jelenti. Túl gyorsra ne állítsuk a kurzort, mert akkor a nyílát nem fogjuk tudni pontosan pozicionálni. A koordináták lekérdezése sem nehéz:

X=PEEK(53248)—24+256\*(PEEK(53264) AND 1)

Y=PEEK(53249)—50

T=PEEK(56320) AND 16

Ezután az X és az Y változóba kerülnek a nyíl csúcsának koordinátái, a T-be pedig a tűzgömb állapota. A T értéke 0, ha a gömb igen, 1 pedig ha az nincs meányomva.

Az egérmutató kezelése sem nehéz, igaz, azt meg kell szokni. Kétséges esetben bizonyára segít a mellékelt két példaprogram elemzése.

Az elsőben a nyílhegy mindenkori pozícióját írjuk ki, a másodikban a nyíl által éppen kijelölt szó karaktereit a tűzgömbbal inverzbe és vissza állíthatjuk.

```

    )                EGER-MUTATO DEMO 2                ○
○ 100 rem -----,----- ○
  105 rem"----- Egermutato-demo 02 ----- ○
  115 rem -----,----- ○
○ 120 : ○
  125 : ○
  
```

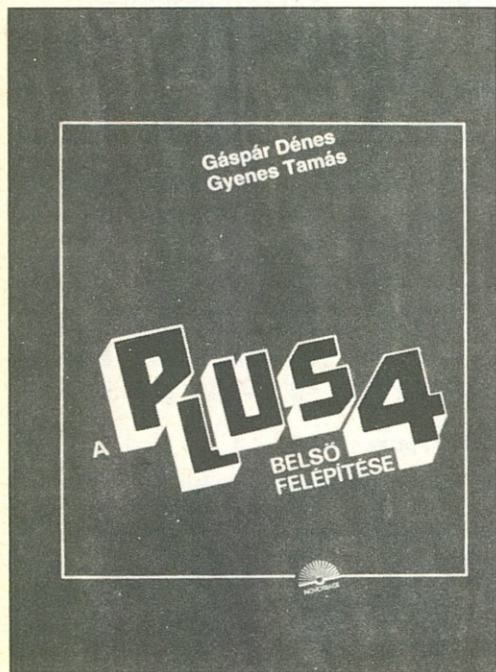
```

○ 130 if a=0 then a=1:load "eger 64",8,1 ○
  135 poke 53280,15:poke 53281,15 ○
  140 poke 646,11 ○
○ 145 print chr$(147);chr$(14) ○
  150 : ○
  155 print "Kerek 10 tetszoleges szot" ○
○ 160 print "beirni ":"print ○
  165 for t=0 to 9 ○
  170 input w$(t) ○
  175 next t ○
○ 180 : ○
  185 print chr$(147); ○
○ 190 for t=0 to 9 ○
  195 print w$(t) ○
  200 next t ○
○ 205 sys 49152,1,0,2 ○
  210 : ○
  215 t=peek(56320) and 16 ○
○ 220 if t<>0 then 215 ○
  225 y=peek(53249)-50 ○
  230 l=int(y/8) ○
  235 if l>10 then 215 ○
○ 240 x=peek(53248)-24+256*(peek(53264)and 1) ○
  245 s=int(x/8) ○
○ 250 if s> len(w$(1)) then 215 ○
  255 : ○
  260 ad=1024+l*40 ○
○ 265 for t=0 to len(w$(1)) ○
  270 poke ad+t,(peek(ad+t)+128) and 255 ○
  275 next t ○
○ 280 goto 215 ○

ready.
  
```

Gáspár D.—Gyenes T.:

## A Plus/4-es belső felépítése



Ára: 149 Ft  
Megjelent: 1988

A könyv részletesen ismerteti a Commodore Plus/4-es felépítését. Bemutatja az egyes részek kapcsolatait mind a hardver, mind a szoftver vonatkozásában. Részletesen megismerhetjük a különböző perifériákkal való kapcsolattartás lehetőségeit is.

A könyvhöz teljes, aprólékosan kommentált ROM lista tartozik. A kapcsolási rajzokat mindazok jól használhatják, akik maguk akarják gépüket bővíteni. A könyv szerzői különös gondot fordítottak arra, hogy a C 16-os géptől való eltéréseket nyilvánvalóvá tegyék, s igyekeztek minden olyan terület feltárására is, amelyre az eddigi könyvekben nem került sor (pl. ACIA, RS232, cartridge stb.).

Aki megunta már a lövöldözős játékokat, szereti a számítógépet és egy kicsit a fizikát, nagyon sok hasznosat talál majd a könyvben.

A mechanika tárgyköréből vett számos probléma (versenyfeladatok) megoldásának bemutatására vállalkoznak a szerzők.

A problémák, ill. a feladatmegoldások megközelítése a számítógépes megoldáshoz juttatja el az olvasót.

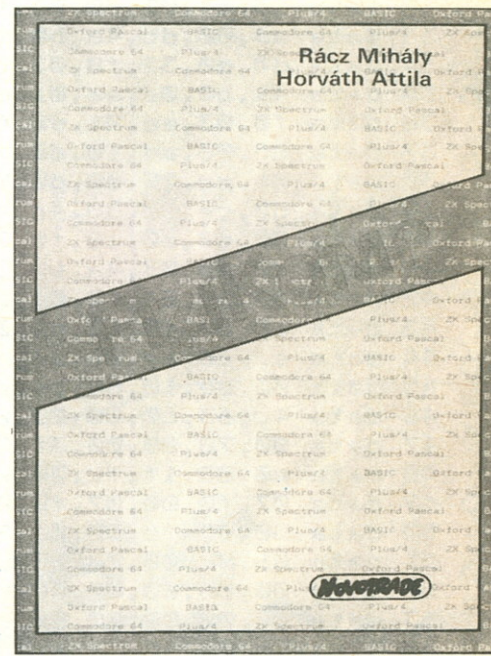
A megoldásokat PLUS/4-es, C 64-es és ZX Spectrum gépekre közlik a szerzők BAVIC és Pascal nyelvű programokban.

A Pascal nyelvvel egy külön fejezet foglalkozik. Az itt szerzett ismeretek elegendők a programok megértéséhez és futtatásához.



Rác M.—Horváth A.:

## Fizikomp



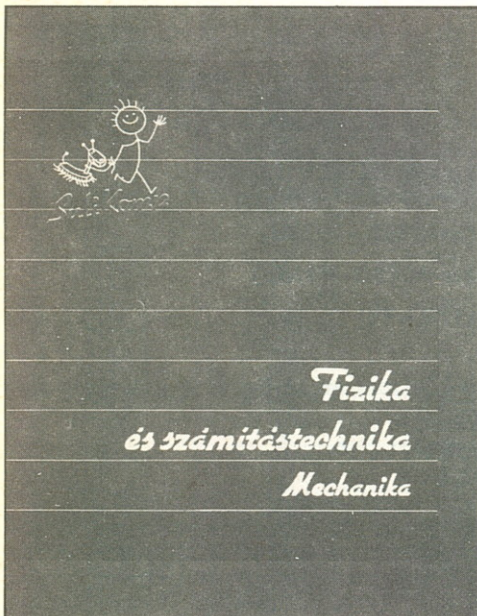
Ára: 129 Ft  
Megjelent: 1987



dr. Kovács I. (főszerkesztő):

## Fizika és számítástechnika — Mechanika

A könyv a gimnáziumok második osztályának fizika anyagát dolgozza fel. A feldolgozás és szemlélet újszerűsége abban mutatkozik, hogy mind az elméleti, mind a gyakorlati problémák tárgyalásánál a számítógépes feldolgozás lehetőségét is bemutatják a szerzők. A problémák megértése az algoritmikus gondolkodás következetes végigvitelét jelenti az Olvasó számára. A programlisták BASIC nyelven C64-es és PLUS/4-es gépekre készültek.



Ára: 149 Ft  
Megjelent: 1987


 Dedinszky F.—  
dr. Horányi I.:

## Számítástechnika a történelem tanításában

A szerzők bemutatják, hogy a történelemtanárok, ill. a történelem iránt érdeklődők hogyan használhatják munkájuk során a számítógépet. Nem titkolt célja a könyvnek, hogy kedvet és bátorságot adjon mindazoknak, akik egyémségüktől nagyon távol állónak érzik a számítástechnikát. A szerzők sem nyíltan, sem burkoltan nem akarnak senkit megtanítani programozni, azt akarják megmutatni, hogy a „mosógépet sem mosógépszerelő” kezeli.



Ára: 99 Ft  
Megjelent: 1987

Rácz J.—Rácz Zs.:

## Matematika és számítástechnika

A könyvek a gimnáziumi első és második osztályos matematika anyagot dolgozzák fel. Mind az elméleti, mind a gyakorlati rész integráltan tartalmazza a számítástechnikai megoldásokat, ill. tanítja a számítástechnikát. Messzemenő célja a szerzőknek a matematika élvezetes megtanításán túlmenően algoritmikus gondolkodás elsajátítása.

A programok Commodore 64-es és PLUS/4-es gépekre készültek.



I. kötet ára: 149 Ft  
Megjelent: 1987

SULIKOMP — tanároknak, diákoknak







# hírek a GEOS világából

Hosszú ideje nem hallottunk újdonságokat a GEOS-ról, hacsak a 2.0-t nem számítjuk ide. Most azonban egy sor alkalmazás megjelenése várható: Geochart, Geobasic, Geoterm, Megapack...

A GEOS rendszerhez hamarosan egy sor jó program lesz kapható. Elsőként a már régóta várt Geochart-ot (1. kép) említhetjük. A programot ugyanis időközben lefordították németre, és az áprilisban meg fog jelenni a német piacon. Kb. 59 márkába fog kerülni. Ezzel a programmal mód lesz nyolcféle formátumú diagramokat készíteni. Nem lesz gond a tortadiagram sem. Itt az egyes „szeleteket” különböző mintákkal tölthetjük ki. A diagramokat a Geochart segítségével mindenféle írásképpel, mindenféle méretben feliratozni lehet anélkül, hogy át kellene konvertálni a képet a Geopaint formátumra. A GEOS 2.0 alatt az értékek beadására még a jegyzetömböt is használni lehet.

Egy további újdonság neve Geobasic. Ezzel az applikációval játékos könnyedséggel lehet a GEOS alatt programokat írni. A programozáskor nem kell csak a Geobasic parancsaira hagyatkozni. Ezenkívül programozhatunk menüket a joystickkel vagy az egérrel (2. kép). Sajnos, az még nem tisztázott, mikor kerül ez a program az üzletkebe. A Markt & Technik Buchverlag-tól kapott értesülés szerint jelenleg folynak a tárgyalások a Berkeley Softworks-szel arról, hogy ezt a programot mint különálló terméket, vagy mint Bookware-t (könyv+program) forgalmazzák-e.

A Megapack a legkülönfélébb GEOS alkalmazások gyűjteménye. Ebben az esetben a forgalmazás előtt szabad az út. Ez a termék Bookware-ként kerül az üzletkebe. A csomag egy új, átdolgozott grafika-, egy jelkészlet- és egy soros nyomtatókat kezelő meghajtórutinok illesztésére szolgáló konvertert tartalmaz. A jelkészlet konverterrel a Printfox számára írt karaktereket lehet GEOS-jelkészlettel alakítani. Így a leveleinket a legeslegkülönfélébb kinézettel készíthetjük el. A jelenleg meglévő 190 Printfox jelkészletet 90 GEOS jelkészlettel kellett összefoglalni, emiatt problémákat okozhat, ha csak a méreteket szeretnénk megváltoztatni.

A grafika-barátoknak a Megapack nagy grafika-könyvtárat kínál. Ebben számtalan kis kép található. Ezt a komoly szoftver-

csomagot a Markt & Technik Buchverlag március végén jelentette meg. A könyvet és a három lemezt tartalmazó csomag ára kb. 59 márka.

A GEOS 128 V2.0 német változatát április elején várhatjuk. Egy teszt példány (egyelőre angolul) rendelkezésre áll, így mód van egy kis előzetes ismertetésére. A 40 karakteres képernyőn a GEOS 128-ast nem tudjuk megkülönböztetni a GEOS 64-től. Csak két plusz almenü tűnik fel: a Switch 40/80 és a Boot). A 80 karakteres képernyőn a GEOS 128 V2.0 már inkább a jelenleg kapható GEOS 128-ra emlékeztet. A színes piktogramok, amelyek a 40 karakteres képernyőn az adatállományok fölismerését könnyítik meg nagyban, sajnos a 80 karakteresen nem léteznek. Ennek hardveres oka van, a VDC chipben igen kevés a tárolóhely.

Az új GEOS 128-as változat időközben kiválóan együttműködik a 1581-essel. Írási/olvasási hibák nem jelentkeztek. A program ára lapzártáig még nem derült ki.

## A GEOS és a TAF

A távadat-feldolgozás szerelmesei a közeljövőben egy olyan program birtokába juthatnak, amely önmagáért beszél. Ez pedig a Geoterm (3. kép). Időközben a programot erősen átdolgozták, javítottak rajta. A program például képes a VT 52 emulációra, és ismeri az X-modem üzemmódot is. A javított változatba a biztonság kedvéért beépítették az 1200 baud-os átvitel lehetőségét is. Sajnos, a teszt példányunkban, amely egyelőre csak előzetes verzió, ez a rész még hiányzik. A programozó most gondolkodik a C128-as változaton. Az M&T Buchverlagtól kapott információk szerint a 128-as Geoterm változat mindkét képernyőt (40/80) kezelni fogja. A terminálprogram kb. 69 márkába fog kerülni.

Mindent egybevetve a GEOS applikációkról szóló hírek sokat ígérőek. Ezek a programok valóban mindenkinek kínálnak valami használhatót.



# Tippek és trükkök — a GEOS-hoz

A GEOS operációs rendszer a kezdőknek és a profiknak is kihívás. A hozzáadott tippekkel és trükkökkel, mint amilyen a GEOS és az NLQ, a kezdők is GEOS profikká válhatnak.

## A GEOS ÉS AZ NLQ

Számos olvasó panaszkodik a Geowrite dokumentumok nem éppen megdöbbentő minőségén. Egy apró trükk segítségével azonban egy NLQ nyomtatón a GEOS-szal is lehetséges NLQ írást létrehozni. Ehhez a Geomerge programra és a nem arányos „Commodore” jelkészletre van szükség. Először meg kell írni a szöveget a Geowrite segítségével. Ékezeteket és különleges jeleket azonban nem lehet használni. Ha kész a levél, hívjuk be a Geomerge programot, a dokumentumot mint formaszöveget deklaráljuk és az ezt követő kérdésre „nem”-mel válaszolunk. A nyomtatási arányok beállításánál csak annyit tegyünk, hogy az NLQ mezőt választjuk ki, és a nyomtatót az NLQ módba kapcsoljuk. A szövegben álló grafikákat és ékezeteket nem nyomtatjuk ki.

## A RAJZABLAK TÖRLÉSE

Amikor egy kép rajzolásának fogunk neki, gyakran gyorsabb, ha a teljes grafikát közvetlenül töröljük. Erre a célra nincs a GEOS-ban megfelelő funkció. Egy módszerre azonban, amellyel a rajzfelületet gyorsan törölhetjük, magam bukkantam rá. A leglogikusabb az, ha az új rajzot egy új névvel kezdjük. Más lehetőség szerint válasszuk ki a teljes rajzfelületet editálási felületnek. Ezt azután gond nélkül „szabaddá” tehetjük. Néha ilyenkor „keretrészek” maradhatnak vissza, amelyeket így nem lehet eltávolítani. Ezt a leggyorsabban és a legegyszerűbben úgy lehet törölni, ha a radírgumira gyorsan kétszer egymás után „rá-lövünk”.

## ÜRES OLDAL TÖRLÉSE

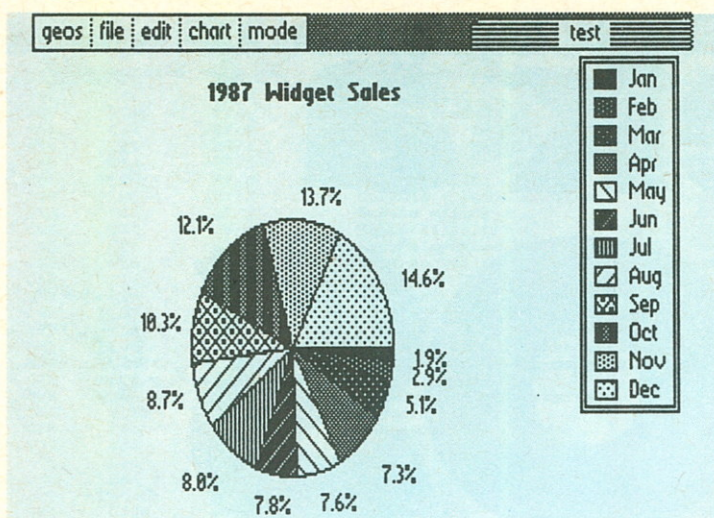
Gyakran előfordul a GEOS 1.3 és a GEOS 1.28 alatt az, hogy üres lapok állnak a tartalomjegyzékben. Ezek az üres lapok a lemez „kitakarítás” után is megmaradnak. Egy lemezes monitorral azonban ezek eltávolítása lehetséges. Mi az, amire figyelniünk kell?

Először másoljuk át az adatállományokat úgy, hogy az üres lap legyen az utolsó. Vegyünk elő ezután egy tetszőleges lemezes monitort. Olvassuk be a 18-as track 3-as szektorát. A 1581-es esetben természetesen a 40-es track 1-es szektorát hívjuk be. Az első két byte itt az, amely megmutatja a logikailag következő blokk track- és szektorszámát. Ha az utolsó elértük, akkor a két byte értéke \$00 és \$FF. Ez jelzi a tartalomjegyzék utolsó szektorát. Ezzel a következő blokkokról nem veszünk tudomást.

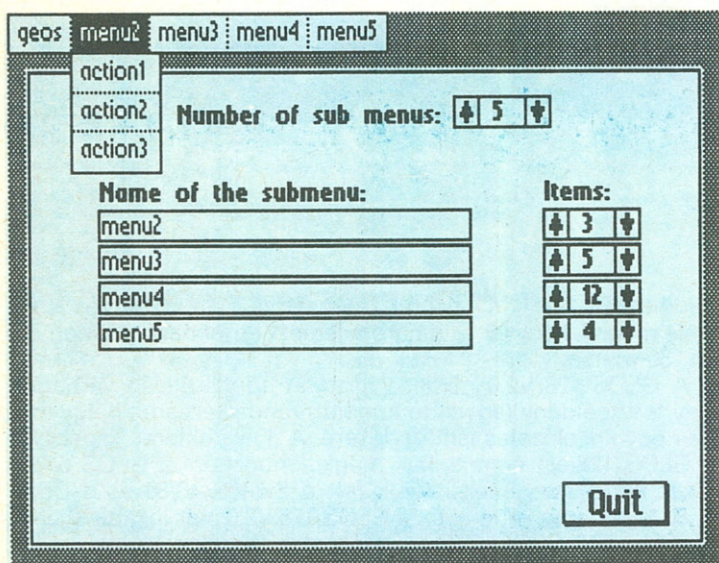
A GEOS 2.0 alatt a directory egyes lapjait a <CTRL S> és a <CTRL T> segítségével problémamentesen törölhetjük vagy szűrhetjük be. Olvassuk el ezzel kapcsolatban a GEOS 2.0 teszt-jét is a Magazin e számából.

## KONVERTÁLNI VAGY SEM?

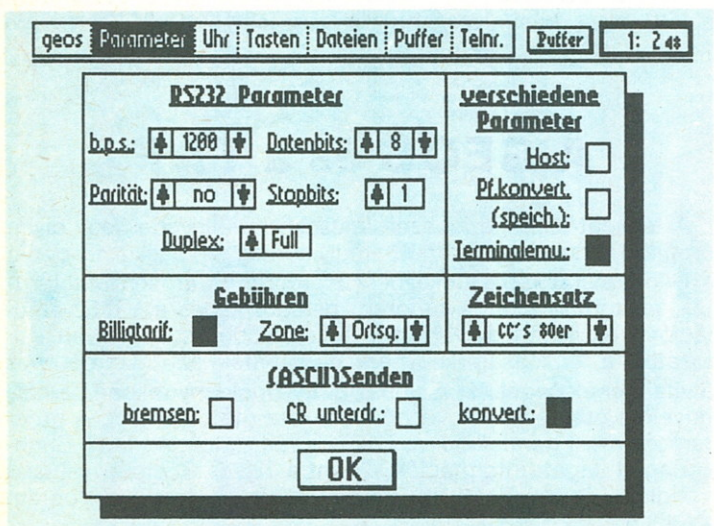
Gyakran kérdezik, konvertáljuk-e lemezeinket a GEOS formátumra vagy sem. A normál C64-es operációs rendszerrel föl-írt formátumok konvertálása semmi gondot nem okoz. Ha azonban a lemez blokkfoglaltsági térképe (BAM, Block Allocation Map) nem teljesen korrekt, mint az a másolás ellen védett lemezeknél szokásos, a konvertálásnál adatvesztés léphet föl. Azok-



1. kép: A Geochart program adatokat nyolcféle diagram képében tudja megjeleníteni.



2. kép: A Geobasic segítségével GEOS alkalmazásokat írhatunk BASIC parancsokkal vagy menüvel.



3. kép: A Geoterm még egy VT 52 emulációt is kínál a TAF játékok vezérléséhez.



nál a lemezeknél, amelyek 40 trackes formátumúak (például a Dolphin-DOS), valamint az originál diszkek esetében, alapvetően ne konvertáljunk. Egy programokat tartalmazó lemez esetében az alábbi módon járunk el:

1. Nyissuk meg a nem-Geos lemezt.
2. Nyomjuk meg a tűzgombot a „nein” (nem) mezőben.
3. Válasszuk ki az „Aufräumen” (takarítani) pontot a Diskette menüben.
4. Nyissuk meg a lemezt még egyszer.

Amíg a GEOS alatt dolgozunk, nagyobb problémákra nem kell számítani. Nehézségek csak akkor jelentkezhetnek, ha a GEOS lemezeket a GEOS-on kívül használjuk. Előfordulhat olyan helyzet, hogy a lemez még GEOS formátumú, a valóságban azonban a specifikus információk (pl. a keret helye a lemezen) rég szét van rombolva. Ebben az esetben csak az használ, ha a még jó állományokat átmásoljuk egy lemezre, és a hibásat újraformátáljuk.

### NYOMTATÓPROBLÉMÁK MEGOLDÁSA

Igen sok olvasó küszködik nyomtatóproblémával. Legyen az az MPS1000-es IBM módja, a Star LC10-es vezérlése stb., stb. Éppen ezért adunk közre nyomtatóillesztési tanácsokat.

Az MPS100-es az IBM módban csak akkor nyomtat, ha az 1—1 jelű DIP kapcsoló ON állásban van. Nyomtatómeghajtásnak ekkor az IBM 5152+ állományt használhatjuk.

A Praesident 6313C esetében a 6—1, 6—2, 7—1, 13—1, 13—2, 14—1, 14—2 és a 18—2 jelű DIP kapcsolókat kell ON állásba biliteni. Így az Epson LX800-as meghajtó használható.

Egy Epson LX800-as a userporton üzemeltetve az FX80(GC) meghajtóprogrammal viszonylag jó minőségben képes nyomtatni.

Ha bárkinek még problémája lenne a printerével, akkor az alábbi címre írjon:

Markt & Technik Buchverlag

Stichwort: Druckeranpassung zu GEOS

Hans-Pinsel Str. 2., D—8013 Haar bei München

Egy ingyenes lemezt várhatunk erről a címről, amelyik több nyomtatómeghajtót tartalmaz. Ha ezek közül egyik sem működne, akkor a Printer Driver Creator-ral el lehet készíteni egy megfelelő nyomtató meghajtót.

Még egy útmutatás a Geofile-hoz. Ebből a programból sajnos az első példányok hibásak voltak. A hiba abban jelentkezik, hogy a Geofile nem találja meg a nyomtatómeghajtót annak ellenére, hogy az a lemezen van. Ha valakinek ez a változata

lenne meg, ingyen kicserélhető a fönti címen (az eredeti lemez beküldésével).

### MIRE JÓ AZ „ANZEIGE”?

A menüsávban álló „Anzeige” (kijelzés) pontot számos felhasználó egyáltalán nem használja ki. Ha az állományok kijelzését a „piktogram” módon kívül másra állítjuk, (látszólag) nem lehet használni ezt a segédfunkciót. Mire jó akkor ez?

Nos, tételezzük fel, hogy a Geowrite program valamelyik lemezünkön található. Ha ezt akarjuk megkeresni, állandóan meg kell nyitnunk a vizsgálandó lemezeket, ami viszonylag sokáig tart. Minden esetben be kell ugyanis olvastatni a piktogramok rajzállományait. Ez azonban nem szükséges, ha a kijelzési módot a „piktogram”-ról a „nach Größe”, azaz méret szerintire állítjuk. A GEOS ekkor az adott lemezen található állományokat nagyság szerint rendezve jeleníti meg. Mivel a Geowrite meglehetősen hosszú, az mindjárt a lista elején lesz, így azt hamar meg lehet találni. Ha a program megvan, újra „piktogram” kijelzésre lehet kapcsolni.

Az állományokat azonban nemcsak a méret szerint lehet rendezni. Az állománytípus (lásd a mellékelt táblázatot), a dátum és az ABC sorrend is választási kritérium lehet. Ezzel sok programot lehet viszonylag gyorsan megtalálni.

### AZ ÁLLOMÁNYOK TÍPUSA

- 0 nem
- 1 BASIC
- 2 Assembler
- 3 adatok
- 4 rendszer
- 5 segédprogram
- 6 alkalmazás
- 7 dokumentum
- 8 jelkészlet
- 9 nyomtatómeghajtó
- 10 beviteli meghajtó
- 11 lemezmeghajtó
- 12 startprogram
- 13 ideiglenes
- 14 önvégrehajtó
- 15 C128 bev. meghajtó

PÖTYÖGŐ  
AKCIÓ

# AKCIÓ!

OLCSÓBB  
PROGRAMOK

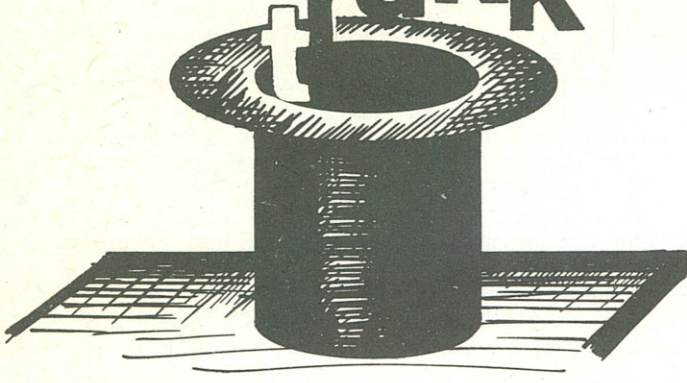
**A FLOPPY  
ÖNNEK  
INGYEN  
VAN!**

8 db program VC20-ra	220,—
12 db program Plus/4-re	320,—
16 db program C128-ra	420,—
20 db program C16-ra	520,—
24 db program C64-re	620,—
32 db program C64-re	820,—

— Megrendelhető  
utánvétellel!  
— Megvásárolható  
az egyesület  
irodájában kedden  
és csütörtökön!



# Tippek trükkök



## FLASH-LOAD

A közismerten lassú 1541-es floppy töltési sebességének növelésére több program létezik. Íme, egy újabb.

Sebessége az eredetihez képest kb. 6-szoros. Ez a gyorsított között átlagos. Tapasztalatom szerint a HYPRA ennél valamelyest lassabb, a MEGALOAD és M+ azonos sebességű, de lényegesen megbízhatatlanabb. A DISK TURBO+4 megbízható és ugyanilyen gyors, de a képernyő memóriát használja, ami néha hátrány.

A program RUN-nal indítva újra definiálja a funkcióbillentyűket, majd az ezeket követő \$0580–07d8 területre másolja magát. Ettől fogva a DLOAD és LOAD "...", "8 műveletek felgyorsulnak. Mint minden gyorsított soros buszt használó programnál, a soros buszra csak az egy darab lemezegység lehet kapcsolva. Más egységeket elég ez időre csak kikapcsolni.

Az F-billentyűk közül az 1-es, 4-es és 8-as törlődik, a többi funkcionálisan a bekapcsolási állapotnak megfelelő lesz (néhol rövidítve van a parancs).

Tetszőleges hosszúságú egyrésztes programot képes betölteni, amely nem használja az említett területet (a HYPRA-nál ez 233, a MEGALOAD-nál 246 blokk). Természetesen MONITOR-ból is működik, és használható a másodlagos cím (.8 vagy .8,1). A több résztes programoknál lényeges, hogy felülírják-e a területét és a további részeket a LOAD-vektoron keresztül töltik-e. Ha a terület nem változik és a részek a vektoron keresztül töltődnek, a FLASH-LOAD használható. Mivel ezeket csak a programok elemzésével tudhatjuk meg pontosan, egyszerűbb kipróbálni az eseteket.

A töltő floppy-programja a C64-re készült FLASH-COPY V2.0 rutinján alapszik.

Nagy Tamás

```

1000 REM *****
1001 REM * C= UJSAG SORSZAM: 179 *
1002 REM *****
1003 REM * FLASH-LOAD *
1004 REM *****
1005 REM * C-16, C-116, PLUS, 4 *
1006 REM *****
1007 REM * NAGY TAMAS 1989.01.20. *
1008 REM *****
3000 DATA 16, 10, C4, 07, 9E, 34, 31, 33, 0227
3001 DATA 30, 3A, 00, 3A, 54, 4F, 4D, 20, 0234
3002 DATA 48, 2E, 58, 2E, 00, 00, 00, 46, 0142
3003 DATA 4C, 41, 53, 48, 2D, 57, 41, 52, 023F
3004 DATA 45, EA, EA, EA, A9, 80, 8D, 2E, 04E7
3005 DATA 03, A9, 05, 8D, 2F, 03, A0, 00, 0210
    
```

```

03006 DATA 09, 4F, 10, 99, 5D, 05, C8, D0, 03A8
03007 DATA F7, B9, 4F, 11, 99, 5D, 06, C8, 03D4
03008 DATA D0, F7, B9, 4F, 12, 99, 5D, 07, 03DE
03009 DATA C8, C0, 7B, D0, F5, 60, 00, 00, 0428
03010 DATA 00, 06, 05, 00, 06, 04, 04, 00, 0019
03011 DATA 44, 4C, 4F, 41, 44, 22, 44, 49, 0213
03012 DATA 02, 3A, 00, 44, 53, 41, 56, 45, 028C
03013 DATA 22, 52, 05, 3A, 0D, 4C, C9, 3A, 02DF
03014 DATA 0D, 06, B4, 04, B5, A5, AE, C9, 049C
03015 DATA 08, F0, 03, 4C, 4A, F0, A6, AD, 03D4
03016 DATA 20, 60, F1, A9, 60, 85, AD, 20, 03CC
03017 DATA 05, F0, A5, AE, 20, FA, ED, A5, 04F4
03018 DATA AD, 20, 1A, EE, 20, 8B, EC, 85, 03F1
03019 DATA 9D, A5, 90, 4A, 4A, B0, 28, 20, 035E
03020 DATA 0B, EC, 85, 9E, 8A, D0, 08, A5, 04A1
03021 DATA B4, 85, 9D, A5, B5, 85, 9E, 20, 0473
03022 DATA AB, FF, 20, 11, F2, AD, 06, FF, 047F
03023 DATA 29, EF, 8D, 06, FF, A9, 02, 20, 0375
03024 DATA C3, FF, 20, 1B, 06, 90, 0B, AD, 034B
03025 DATA 06, FF, 09, 10, 8D, 06, FF, 4C, 02FC
03026 DATA 2B, A8, 20, 04, 06, A9, 08, 20, 029E
03027 DATA B1, FF, A9, 6F, 20, 93, FF, A0, 051A
03028 DATA 04, B9, 16, 06, 20, A8, FF, 88, 0328
03029 DATA 10, F7, 20, AE, FF, 20, 49, 06, 0343
03030 DATA AD, 06, FF, 09, 10, 8D, 06, FF, 035D
03031 DATA 00, CD, A6, 9D, D0, 02, C6, 9E, 04F6
03032 DATA C6, 9D, A6, 9D, A4, 9E, 60, 06, 044E
03033 DATA 07, 45, 2D, 4D, A9, 08, 20, B4, 024B
03034 DATA FF, A9, 6F, 20, 96, FF, 20, A5, 0491
03035 DATA FF, C9, 32, 08, 90, 17, 48, 20, 0311
03036 DATA D2, FF, 68, 20, D2, FF, 20, A5, 04EF
03037 DATA FF, C9, 0D, F0, 05, 20, D2, FF, 04BB
03038 DATA 90, F4, 20, 3A, FB, 20, AB, FF, 04A3
03039 DATA 20, 60, 78, AD, 19, FF, 48, 20, 032D
03040 DATA A2, 06, C9, 02, 80, 45, 20, A2, 032A
03041 DATA 06, 05, DE, 20, A2, 06, A4, DE, 03B3
03042 DATA F0, 02, A9, FE, 85, DF, A0, 00, 049D
03043 DATA 24, D0, 30, 0C, 20, A2, 06, 20, 0225
03044 DATA A2, 06, C6, DF, C6, DF, C6, D0, 0595
03045 DATA 20, A2, 06, 91, 9D, C8, C4, DF, 0461
03046 DATA D0, F6, 18, 98, 65, 9D, 85, 9D, 049A
03047 DATA 90, 02, E6, 9E, A5, DE, D0, 0F, 0528
03048 DATA C0, FE, F0, 07, 2D, A2, 06, C8, 0445
03049 DATA 18, 90, F5, 68, 8D, 19, FF, 58, 0402
03050 DATA 4C, 1B, 06, A9, 04, 85, 01, 24, 01C4
03051 DATA 01, 10, FC, A9, 00, 85, 01, A2, 02DE
03052 DATA 06, CA, EA, D0, FC, A2, 04, A5, 04D1
03053 DATA 01, 8D, 19, FF, 8D, 19, FF, 8E, 03D8
03054 DATA 19, FF, 0A, 08, 0A, 26, 06, 20, 0238
03055 DATA 26, B6, 2E, 19, FF, CA, D0, E7, 04A3
03056 DATA A5, B6, 49, FF, 60, A9, 36, A2, 0484
03057 DATA 07, 85, 03, 86, 04, A0, 00, A9, 0262
03058 DATA 06, 04, D0, 84, 05, 85, 06, A9, 0324
03059 DATA 08, 20, B1, FF, A9, 6F, 20, 93, 03A3
03060 DATA FF, A9, 4D, 20, A8, FF, A9, 2D, 0492
03061 DATA 20, A8, FF, A9, 57, 20, A8, FF, 048E
03062 DATA A0, 00, A5, 05, 20, A8, FF, A5, 03B6
03063 DATA 06, 20, A8, FF, A9, 20, 20, A8, 035E
03064 DATA FF, B1, 03, 20, A8, FF, C8, C0, 0502
03065 DATA 20, 90, F6, 20, AE, FF, 18, 98, 0423
03066 DATA 65, 03, 85, 03, 90, 03, E6, 04, 026D
03067 DATA 18, 98, 65, 05, 85, 05, 90, 02, 0236
03068 DATA E6, 06, C9, A2, 90, B1, 60, A9, 04A1
03069 DATA 03, 85, 31, 4C, D1, F4, 20, 18, 0302
03070 DATA C1, A5, 18, A6, 19, 85, 0C, 86, 0354
03071 DATA 0D, A9, E0, 58, 85, 03, A5, 03, 031E
03072 DATA 30, FC, C9, 02, 90, 20, A9, C0, 0410
03073 DATA 85, 03, A5, 03, 30, FC, A9, E0, 03E5
03074 DATA 85, 03, A5, 03, 30, FC, C9, 02, 0327
03075 DATA 90, 0C, 78, 48, 20, 78, 06, 68, 0262
03076 DATA A2, 03, C9, 02, 90, 20, A9, 00, 0351
03077 DATA 00, 20, 78, 06, B9, 00, 03, 20, 017A
03078 DATA 78, 06, C8, D0, F7, AD, 00, 03, 03BD
03079 DATA F0, 16, 48, C9, 36, 00, 15, 20, 0332
03080 DATA 4B, F2, CD, 01, 03, F0, 0D, 90, 039B
    
```



```

○ 3081 DATA 0B,AE,01,03,68,4C,0E,06,0185
3082 DATA 58,4C,9E,C1,A9,66,20,78,03AA ○
3083 DATA 06,A2,03,58,4C,4D,05,85,02F6
○ 3084 DATA 85,2C,00,18,10,FB,A9,10,028D
3085 DATA 8D,00,18,A2,04,A9,00,06,01FA ○
3086 DATA 85,2A,0A,06,85,2A,0A,8D,0205
3087 DATA 00,18,CA,D0,F0,A2,01,D0,0415 ○
3088 DATA 00,EA,EA,A9,0F,8D,00,18,0331
3089 DATA 60,00,00,00,00,00,00,00,0060 ○
4030 BR=4097
4031 PRINT"KONTROL INDUL !"
○ 4032 FORK=0T089
4033 C=0
4034 FORB=0T07
○ 4035 READA#
4036 C=C+DEC(A#)
○ 4037 NEXTB
4038 READA#
4039 IF C<>DEC(A#) THEN PRINT"ADATHIBA
A"3000+K"-ADIK SORBAN !":STOP
○ 4040 NEXTK
4041 PRINT"KONTROL KESZ !"
○ 4042 PRINT"BEIRAS INDUL !"
4043 RESTORE
○ 4044 FORK=0T089
4045 FORB=0T07
4046 READA#
○ 4047 POKEBA+K*8+B,DEC(A#)
4048 NEXTB
4049 READA#
○ 4050 NEXTK
4051 PRINT"OK."
4052 PRINT"MENTSOD KI !"
○ 4053 PRINT"RUN-NAL INDUL."
4054 POKE45,202:POKE46,18:CLR:END
○
READY.
○

```

## BASIC-LISTA-SCROLL

Valószínűleg sokakban felmerült a gondolat a JCL-ASSEMBLER láttán, hogy a lista kurzorral való fel-le mozgását jól lehetne BASIC programok szerkesztéséhez is használni. Ehhez azonban mindig be kell tölteni a JCL-t, ami tekintélyes méretű (lemezen 65 blokk). A problémát oldja meg ez a rövid program.

A BASIC töltő lefutása után mentsük ki a kész programot (3 blokk!).

Indítása: RUN. Ezután dolgozhatunk a gépen. A kurzor teljesen normálisan működik, de ha megnyomjuk a CTRL és HELP gombot együtt, a BASIC programunk attól a sortól fog a képernyőre írni, amelyiken a kurzor éppen állt. Most a fel és le nyilakkal mozgathatjuk a listát, a RETURN-nel előre lapozhatunk. A szerkesztőbe a RUN/STOP-pal léphetünk vissza. Természetesen a lista ott marad, ahová beállítottuk.

A CTRL-HELP, majd a RUN/STOP megnyomásakor a keret színe (oda-vissza) megváltozik, így mindig látható, melyik üzemmódban vagyunk.

A programnak két változata van, ezek csak a memóriabeli elhelyezkedésükben különböznek:

Az első változat \$0599–07AA-ig tart. Ez rendszerterület, így BASIC programoktól nem vesz el helyet, és nem zavarja a grafikus üzemmód sem. Ellenben futtatáskor előfordulhat, hogy a BASIC-stack beleír, és akkor RUN/STOP-RESET után újra kell tölteni.

A második változat \$0FE8–11F9-ig tart. Tehát 512 byte-ot foglal le a BASIC memóriából. A grafikus üzemmód tönkreteszi, de mindenféle programfuttatást, egyebet kibír.

Mindkét változat a RUN/STOP-RESET után a SYS 3048 [vagy SYS DEC ("OBEB)] utasítással újraindítható, kivéve persze, ha felül íródtak.

Az első változat forráskódját közöljük. Pötyögőszolgáltatunktól a második változat is megrendelhető.

```

○ 1000 REM *****
1001 REM * C= UJSAG SORSZAM:180 *
1002 REM *****
○ 1003 REM * BASIC-LISTA-SCROLL $0599 *
1004 REM *****
○ 1005 REM * C-16,C-116,PLUS/4 *
1006 REM *****
1007 REM * NAGY TAMAS 1989.01.20. *
○ 1008 REM *****
2000 DATA 1D,10,C5,07,9E,34,31,32,022E
2001 DATA 38,20,08,0E,22,14,04,4F,01C7
○ 2002 DATA 4D,20,46,52,4F,4D,20,C8,0289
2003 DATA 2E,D8,2E,00,00,00,00,20,0154
2004 DATA 9C,F3,A2,0D,8D,50,10,9D,03F8
○ 2005 DATA E7,0B,CA,D0,F7,8D,5E,10,04AE
2006 DATA 9D,99,05,E8,D8,F7,8D,5E,0305
2007 DATA 11,9D,99,06,E8,D8,F7,8D,04B9
○ 2008 DATA 5E,12,9D,99,07,E8,E0,11,0386
2009 DATA D0,F5,20,E8,0B,4C,7E,86,0428
2010 DATA 78,A9,99,8D,14,03,A9,05,030C
○ 2011 DATA 8D,15,03,58,68,A5,01,30,02B3
2012 DATA 12,A9,7F,20,78,0B,29,04,02D2
○ 2013 DATA D0,09,A9,FE,20,70,0B,29,0414
2014 DATA 08,F0,03,4C,0E,CE,AD,19,02E9
2015 DATA FF,49,FF,8D,19,FF,A9,0E,04A3
○ 2016 DATA 8D,14,03,A9,CE,8D,15,03,02C0
2017 DATA AD,09,FF,8D,09,FF,58,20,03C2
2018 DATA 82,DF,20,8D,07,90,0C,A9,02DA
○ 2019 DATA 2B,A0,03,20,94,04,AA,08,02B8
2020 DATA 20,97,04,85,14,86,15,20,020F
2021 DATA 88,08,20,71,07,A0,01,20,02B9
○ 2022 DATA D1,04,F0,1F,C8,20,A2,07,0375
2023 DATA 20,40,8B,20,8C,DC,20,FB,038E
○ 2024 DATA 06,A5,CD,C9,18,F0,0C,A0,03F5
2025 DATA 00,20,A2,07,86,5F,85,60,0293
○ 2026 DATA 4C,E9,05,78,A9,7F,20,70,036A
2027 DATA DB,0A,90,18,A9,DF,20,70,03A8
○ 2028 DATA DB,4A,80,03,4C,AF,06,29,0302
2029 DATA 04,F0,27,A9,FE,20,70,0B,042D
○ 2030 DATA 29,02,D0,DF,4C,7D,07,A9,0353
2031 DATA 7F,20,70,0B,0A,90,F8,58,03D4
○ 2032 DATA A9,00,85,EF,AD,19,FF,49,042B
2033 DATA FF,8D,19,FF,20,E8,0B,4C,0403
○ 2034 DATA BE,FC,58,A6,CD,F0,06,20,049B
2035 DATA 0D,DD,4C,F5,06,20,CF,DC,03FC
○ 2036 DATA 20,0D,07,85,14,86,15,90,01F8
2037 DATA 03,4C,F5,06,20,8B,DE,20,02F3
○ 2038 DATA 71,07,90,36,A6,68,E4,2C,0354
2039 DATA 90,30,06,E1,CA,86,60,A0,0477
○ 2040 DATA FF,20,01,04,C5,E1,F0,05,048F
2041 DATA 88,F0,1F,D0,F4,88,20,D1,04D4
○ 2042 DATA 04,C5,5F,F0,02,00,EA,EA,048E
2043 DATA 18,98,65,5F,85,5F,90,02,02EA
○ 2044 DATA E6,60,A0,02,20,A2,07,20,02D1
2045 DATA 40,8B,20,9B,DC,20,9A,D8,03F4
○ 2046 DATA 4C,F5,06,58,A6,CD,E8,18,040A
2047 DATA F0,06,20,00,DD,4C,F5,06,033A
○ 2048 DATA 20,82,DF,20,0D,07,00,27,028C
2049 DATA 85,14,86,15,20,8C,DC,20,02DC
○ 2050 DATA 71,07,90,25,A0,00,20,A2,028F
2051 DATA 07,85,60,86,5F,A0,01,20,0292
○ 2052 DATA D1,04,F0,15,C8,20,A2,07,036B
2053 DATA 86,14,85,15,4C,E9,05,A5,0313
○ 2054 DATA CD,F0,06,20,0D,DD,4C,8C,0305
2055 DATA 06,20,FB,06,4C,0F,06,A5,022D
○ 2056 DATA CA,18,65,C8,8D,8D,FF,A5,044D
2057 DATA C9,69,00,E9,0B,8D,0C,FF,03BE
○ 2058 DATA 60,A0,00,84,E6,84,E7,88,045D
2059 DATA C8,30,4C,01,C8,29,7F,C9,042E
○ 2060 DATA 20,F0,F5,C9,3A,80,40,30,0430
2061 DATA E9,30,90,3B,85,E3,06,E6,0438
○

```



```

○ 2062 DATA 26,E7,B0,3F,A5,E6,85,E4,04F0
○ 2063 DATA A5,E7,85,E5,06,E6,26,E7,04EF
○ 2064 DATA B0,31,06,E6,26,E7,B0,2B,03B5
○ 2065 DATA A5,E6,65,E4,85,E6,A5,E7,05CB
○ 2066 DATA 65,E5,85,E7,B0,1D,A5,E6,050E
○ 2067 DATA 65,E3,85,E6,A5,E7,69,00,04A8
○ 2068 DATA 85,E7,B0,0F,4C,14,07,A5,0337
○ 2069 DATA E6,05,E7,F0,06,A5,E6,A6,04F9
○ 2070 DATA E7,18,60,38,60,20,3D,8A,02DE
○ 2071 DATA A5,5F,85,E2,A6,60,86,E3,04DA
○ 2072 DATA 60,58,A6,CD,E0,18,F0,06,0419
○ 2073 DATA 20,8C,DC,4C,F5,06,20,82,0371
○ 2074 DATA DF,20,0D,07,B0,03,4C,DF,02F1
○ 2075 DATA 05,A5,CD,00,03,4C,F5,06,0391
○ 2076 DATA 20,0D,0D,4C,8A,07,20,01,02D8
○ 2077 DATA 04,C8,AA,4C,01,04,00,00,0297
○ 2078 BA=4097
○ 2079 PRINT"KONTROLL INDUL !"
○ 2080 FORI=0TO77:C=0
○ 2081 FORB=0TO7
○ 2082 READA#
○ 2083 C=C+DEC(A#)
○ 2084 NEXTB
○ 2085 READA#
○ 2086 IFDEC(A#)<>0THENPRINT"ADATHIBA
    A"2000+I"-EDIK SORBAN !!!":STOP
    
```

```

○ 2087 NEXTI
○ 2088 PRINT"KONTROLL KESZ !"
○ 2089 PRINT"BEIRAS INDUL !"
○ 2090 RESTORE
○ 2091 FORI=0TO77
○ 2092 FORB=0TO7
○ 2093 READA#
○ 2094 POKEBA+I*8+B,DEC(A#)
○ 2095 NEXTB
○ 2096 READA#
○ 2097 NEXTI
○ 2098 PRINT"OK."
○ 2099 PRINT"MENTSOD KI 'SAVE'-VEL !!!"
○ 2100 PRINT
○ 2101 PRINT"INDITASA 'RUN'-NAL ."
○ 2102 PRINT"AKTIVIZALASA CTRL+HELP ."
○ 2103 PRINT"FUNKCIOK: KURZOR FEL-LE,"
○ 2104 PRINT"                RETURN."
○ 2105 PRINT"KILEPES : RUN/STOP"
○ 2106 POKE45,111:POKE46,18:CLR:END
    
```

○ READY.

## 2×2-es karakterek C= Plus/4-en

A program segítségével a nagy felbontású képernyőre négy-szeres (2×2) nagytűs karaktereket írhatunk. A program hívásakor a bal felső karakternegyed koordinátáit kell megadni (X=0–38,Y=0–23). Ha a szöveg egy sorba nem fér ki, a sor végén „dupla” soremelés hajtodik végre. A program a szímemóriába nem ír, így a képernyő korábbi színei nem változnak meg a szöveg kiírásával.

A karakterkészlet kezdőcímét (a CHAR utasításhoz hasonlóan) a \$02e4 (740) címen állíthatjuk be. Alapértelmezése \$d0 (208), azaz a nagybetűk-grafikus jelek készletét íródik ki, \$d4 (212) beírásával a kisbetűk-nagybetűk készletre kapcsolhatunk, de természetesen saját készletet is használhatunk.

A program BASIC betöltője először végigellenőrzi az adatokat, majd az ellenőrzőösszegeket, s csak ha mindent rendben talál, ír bele a memóriába. Az 1. listában szereplő program a \$0332 (818)-as címen kezdődik, és nem helyezhető át más területre, csak a 2. lista módosításainak begépelésével. A program munka- és adatterületként használja a \$d0–\$d9 és \$02cc–\$02db címeket.

Hibátlan futás után a kiírás a SYS <kezdőcím>,X,Y,<szöveg>[,inverz?]  
utasítással indítható (a paraméterek a CHAR-éval egyeznek meg).

Hibalehetőségek: a koordináták nem esnek a 0–38, illetve 0–23 intervallumba, a szöveg üres string, vagy nincs grafikus terület.

A program jól használható képűságokban, oktató- és más, nagytűs írást igénylő programokban.

*Kámán Tibor*

```

○ 1 REM"*****"
○ 2 REM"*"
○ 3 REM"* C= UJSAG SORSZAM:181"
○ 4 REM"* 2 X 2 -ES KARAKTEREK C=PLUS4"
○ 5 REM"*"
○ 6 REM"* PROGRAM: KAMAN TIBOR 1988."
○ 7 REM"*"
○ 8 REM"*****"
○ 9 :
○ 10 PRINT"JELLENORZES INDUL!"
○ 20 H=0
○ 30 FOR I=0 TO 33:S=0
○ 40 FOR J=0 TO 7
○ 50 READ X#
○ 60 S=S+DEC(X#)
○ 70 NEXT J
○ 80 READ S#:OS=OS+DEC(S#)
○ 90 IF S<>DEC(S#) THEN PRINT"ADATHIBA AZ"1000+I". SORBAN !":H=H+1
○ 100 NEXT I
○ 110 IF OS<>33510 THEN PRINT"MHIBA A KONTROLLOSSZEGEKBEN !!!"
○ 120 IF H<>0 THEN PRINT:PRINT H" HIBAT TALALTAM !":END
○ 130 PRINT"JELLENORZES KESZ. A PROGRAM HIBATLAN..."
○ 140 KC=818
○ 150 PRINT"MININDUL A BEIRAS !"
○ 160 RESTORE
○ 170 FOR I=0 TO 33:FOR J=0 TO 7
○ 180 READ X#:POKE KC+I*8+J,DEC(X#)
○ 190 NEXT J:READ S#:NEXT I
○ 200 PRINT"MINBEIRAS KESZ..."
○ 210 PRINT"MININDITAS:"
○ 220 PRINT"MI SYS"KC"MI,OSZLOP,SOR,'SZOVEG'[,INVERZ]"
    
```



```

1000 DATA 20,8F,C7,20,D8,90,E0,27,442
1001 DATA B0,09,86,00,20,D8,90,E0,404
1002 DATA 18,90,03,40,10,99,86,D1,303
1003 DATA 20,91,94,20,48,9C,F0,F3,42C
1004 DATA 85,02,98,48,8A,48,20,A5,30E
1005 DATA C3,8A,6A,66,03,68,85,22,3FF
1006 DATA 68,85,23,A0,00,84,07,A9,304
1007 DATA 00,85,06,20,86,84,8A,26,25F
1008 DATA D6,8A,0A,26,06,85,04,A5,3E4
1009 DATA D6,80,E4,82,85,05,38,A5,460
1010 DATA D4,E9,01,85,04,68,02,C6,48F
1011 DATA D5,20,AF,03,E6,08,E6,08,513
1012 DATA A6,08,E0,27,36,8F,A2,08,35E
1013 DATA 86,08,E6,01,E6,01,A6,01,63B
1014 DATA E0,18,90,01,60,A4,07,C8,42C
1015 DATA C4,02,08,69,60,A0,08,A9,400
1016 DATA 00,85,08,85,09,B1,04,48,488
1017 DATA 29,8F,A2,84,4A,98,8A,48,28A
1018 DATA 18,8D,36,84,65,09,85,09,38B
1019 DATA 68,CA,08,F0,68,29,F0,6A,400
1020 DATA 6A,6A,6A,A2,84,4A,98,8A,2C8
1021 DATA 48,18,8D,36,84,65,08,85,319
1022 DATA D8,68,CA,08,F0,A5,08,99,5E0
1023 DATA C8,02,A5,09,99,03,02,08,441
1024 DATA D8,8D,A6,01,A4,08,20,91,529
1025 DATA C2,A0,00,A9,CC,20,10,84,318
1026 DATA A9,04,20,10,84,18,A9,48,2E7
1027 DATA 65,8C,85,8C,A5,01,65,8D,39E
1028 DATA 85,8D,98,08,A9,08,20,10,368
1029 DATA 34,A9,08,8D,23,84,A2,08,2DB
1030 DATA 8D,08,02,24,03,18,02,49,2E9
1031 DATA FF,91,8C,C8,91,8C,C8,E8,5B1
1032 DATA E8,84,08,EC,68,C8,38,8C,3FC
1033 DATA 03,00,00,00,00,00,00,00,003
    
```

READY.

```

1 REM"*****"
2 REM"
3 REM" C= UJSAG SORSZAM:181
4 REM" 2 X 2 -ES MODOSITAS C=PLUS4
5 REM"
6 REM" PROGRAM: KAMAN TIBOR 1988.
7 REM"
8 REM"*****"
9
140 INPUT"KEZDOCIM";KC
193 FOR I=1 TO 7:READ A,B:X#=HEX$(B+KC)
194 POKE KC+A,DEC(RIGHT$(X#,2))
195 POKE KC+A+1,DEC(LEFT$(X#,2))
196 NEXT I
2000 DATA 98,125,146,260,171,268,206
2001 DATA 235,211,235,231,235,236,241
    
```

READY.

```

1 REM"*****"
2 REM * C= UJSAG 183 *
3 REM * JAVITOTT KERES *
4 REM * PROGRAM: STEF *
5 REM"*****"
6 REM
10 FOR I=1536 TO 1549 : READA#
20 POKE I,DEC(A#) : NEXT I
30 DATA A5,01,69,38,8D,11,FF
40 DATA A5,C6,C9,3C,D8,F3,68
50 SYSDEC("FFF9")
    
```

READY.

#### MONITOR

PC	SR	AC	KR	YR	SP
: FFFF	00	FF	FF	FF	F9
. 2000	A5	01			LDA #01
. 2002	69	38			ADC ##38
. 2004	8D	11	FF		STA #FF11
. 2007	A5	C6			LDA #C6
. 2009	C9	3C			CMP ##3C
. 200B	D8	F3			BNE #2000
. 200D	68				RTS
. 200E	57				???
. 200F	57				???
. 2010	57				???
. 2011	57				???
. 2012	57				???
. 2013	57				???
. 2014	57				???

## Kereső C16

A program a 87/8–9. számban megjelent program javított, rövidebb és intelligensebb változata, mert a kereső funkcióból SPACE billentyűvel ki lehet lépni. A BASIC betöltő a programot Hexa 600-tól teszi el, így az resetbiztos és nem zavarja a basic programokat, sőt a játékok többségét sem. Betöltés után RUN. Erre a program kitörli magát! A keresés indítása: SYS 1536.

Ha valaki véletlenül mentés előtt elindította a programot, monitorból S" KERESŐ", 1, 0600, 060E utasítással mentheti ki.

STEF  
(Lengyel István)



# 2c áruház

\* 1136 Budapest XIII., Balzac u. 35. Tel.: 402-954 \*

## Fényceruza:

Az 1.3 GEOS által felhasználható, nyomógombos fényceruza segítségével könnyebben dolgozhatunk, készíthetünk szabadkézi vagy műszaki rajzokat.

Egyesületi tagoknak 20% kedvezmény.

**Ára: 2500 Ft**

## TV-COMPUTER átkapcsoló:

A kapcsoló segítségével megszüntethetjük a kábelek állandó ki-be húzkodásával járó kellemetlenségeket. Az öntapadó matricával könnyen, bárhova felerősíthető kapcsoló egy mozdulattal átváltható a megfelelő üzemmódba. A kapcsoló akár videóhoz is felhasználható.

**Ára: 490 Ft**

## C-64 Euroscart kábel:

Az Euroscart bemenettel rendelkező video vagy televízió és a Commodore számítógép összekötésére alkalmas. Az Euroscart bemenet segítségével tökéletesebb, szebb képet élvezhetünk.

**Ára: 675 Ft**

## CCX interface:

A CCX interface kábel alkalmassá teszi a C-64-est arra, hogy centronics bemenetű nyomtatókat (Citizen 1200, STAR, EPSON) kapcsolhassunk a soros vonalra. Segítségével kihasználhatjuk a nagyobb teljesítményű nyomtatók előnyeit.

**Ára: 9009 Ft**

## Basic SYNTAXER C-64 lemez:

A lemezen található programcsomag a Basic programozás legkellemetlenebb hibáit vizsgálja. Hiba esetén a téves sort azonnal módosíthatjuk.

**Ára: 1985 Ft**



## Boríték címzése nyomtatón

Sokféle adatbázis-kezelő program forog közkézen. Hogy most mégis egy ilyen programmal foglalkozunk, annak több oka van. Ezek közül egyik, hogy a profi adatkezelők sokoldalúak ugyan, de mégsem képesek minden igényt kielégíteni, különösen a perifériák sokféleségét, azok minden képességét nem tudják kihasználni. A programunk elve nagyon egyszerű, tipikusan hobby megoldást használ, melyet még a kezdők is könnyen felhasználhatnak arra, hogy saját igényeiknek és lehetőségeiknek megfelelően átalakítsák.

A program lemezen, soros állományban levelezéshez használható címtárat kezel. Új cím beírásakor ezt az állomány végére csatolja (APPEND). Levél írásakor a címzett adatait az állományból kikeresi és a nyomtatóba helyezett borítékot megcímezi. Külön szolgáltatás a címjegyzék listázása.

Látható, hogy a programból elhagyunk sok tipikus adatkezelő tevékenységet. Az adatmódosítást helyettesíti a keresési algoritmus szerkezete, ugyanis a többszörösen szereplő címek esetén a keresés folytatható. A törlésre a viszonylag kevés adatot tartalmazó házi címjegyzékben nincs feltétlenül szükség.

A program első sorában a lemez állomány nevét adjuk meg. Ezt az állományt a programtól függetlenül kell létrehozni:

```
OPEN 8,8,'CIMTAR,S,W' : CLOSE 8
```

A konstansok beállítása, mely a MASTER előtt található, az egyéni elképzeléseknek és a meglévő nyomtatónak megfelelően módosítható. A 0. indexű címmező az az emlékeztető, mellyel az állományban keressük a címzettet. A többi kerül a a borítékra. A címmezők közül néhányat (2., 5. és 7.) dupla szélességű jelekkel írunk a borítékra, ha 19-nél kevesebb jelből állnak. A címmezők tartalmi változtatása esetén a 2330 számú sort kell esetleg emiatt átírni. A program a különleges postai kezelésre vonatkozó feliratokat csak külön kívánságra írja a borítékra. A feladó adatainak végén az országot akkor nyomtatjuk, ha a címzett utolsó adata, az irányítószám x (kisbetűvel!). Erre a cím adatainak beírásakor kell ügyelni. A program alapvető működését nem befolyásolja, ha a katalógust készítő modult (3000—3230) és a 1110 számú sort elhagyjuk.

Dr. Hack Frigyes

```

100 rem *****
101 rem *   c= ujsag sorszam: 184   *
102 rem *   boritek cimezo         *
103 rem *   dr.hack frigyes       *
104 rem *****
105 rem *   cbm basic v3.5         *
106 rem *   c=16 - plus/4         *
107 rem *   diszk - 1551          *
108 rem *   sp-180 vc             *
109 rem *****
110 :
120 ct$="cimtar,s,"
130 rem nyomtato vezerlo
140 esc$=chr$(27) : rem esc
150 sl$ =es$+"3"+chr$(23): rem 8 lpi
160 sl$ =es$+"3"+chr$(36): rem 6 lpi
170 lf$ =chr$(13) : rem soremeles
180 ff$ =chr$(12) : rem lapemeles
190 tb$ =chr$(16) : rem tabulator
200 nb$ =chr$(145) : rem grafikus
210 nl$ =chr$(120) : rem nlq
220 nl$ =esc$+nl$+"1"
230 i$ =es$+"4" : rem italic
240 i0$ =es$+"5"
250 f$ =es$+"e" : rem fett
260 f0$ =es$+"f"
270 v$ =es$+"g" : rem kettos-
280 v0$ =es$+"h" : rem leutes
290 a$ =es$+"-1" : rem alahuzas
300 a0$ =es$+"-0"
310 r$ =chr$(18) : rem revers
320 r0$ =chr$(146)
330 d$ =es$+"w1" : rem dupla-
340 d0$ =es$+"w0" : rem szelesseg
    
```

```

350 rem a felado adatai:
360 fa$(1)=nl$+ "Feladó:"
370 fa$(2)=f$ + "Dr. Hack Frigyes"
380 fa$(3)=a$ + "Budapest XIV"
390 fa$(4)=i$ + "Szikszó park 12."
400 fa$(5)=f$ + " 1 1 4 2"
410 fa$(6)=f$+d$+"HUNGARY"+d0$
420 rem mezonevek / betutipusok
430 m$(0)="K u l c s" : fm$(0)=""
440 m$(1)="Titulus" : fm$(1)=i$
450 m$(2)="N é v" : fm$(2)=v$
460 m$(3)="Külf.cim" : fm$(3)=i$
470 m$(4)=" - - ? - -" : fm$(4)=i$
480 m$(5)="V á r o s" : fm$(5)=a$
490 m$(6)="Belf.cim" : fm$(6)=i$
500 m$(7)="Ország" : fm$(7)=i$+v$
510 m$(8)="Ir.sz./x!" : fm$(8)=v$
520 rem különleges jelzések
530 e$(1)=" Expressz "
540 e$(2)=" Ajánlott "
550 e$(3)=" AIR MAIL "
1000 rem ***** master *****
1010 scnclr : print chr$(142)
1020 char,10, 3,"U-----I
1030 char,10, 4," | menu |"
1040 char,10, 5,"J-----K
1050 char,12, 7," Boritek keszites
1060 char,12, 9," Katalogus
1070 char,12,11," Új cim felvetel
1080 char,12,13," Exit!
1090 getkey q$
1100 if q$="b" then gosub 2000
1110 if q$="k" then gosub 3000
1120 if q$="u" then gosub 4000
1130 if q$="e" then clr : end
1140 goto 1000
1150 :
2000 rem +++++ boritek +++++
2010 print chr$(14)" Boritek"
2020 gosub 5000 : rem kereses
2030 if k$="*" then return
2040 print " E x t r á k "
2050 for e=1 to 3
2060 : print " e$(e) " ← (i/n)";
2070 : getkey q$ : e(e)=(q$="i")
2080 : print " " q$ ""
2090 next e
2100 print " "
2110 print " Boritekot befuzni!"
2120 print " Nyomtatot bekapcsolni!"
2130 print " [RETURN]-t lenyomni!";
2140 print " ";
2150 getkey q$:if q$<>lf$ then 2150
2160 open 10,4,10 :print#10
2170 open 4,4,7.
2180 rem felado
2190 for j=1 to 5
2200 : print#4,fa$(j)
2210 : print#10
2220 next j
2230 if c$(8)="x" then print#4,fa$(6);
2240 print#4,
2250 rem ekstrak
2260 for e=1 to 3
2270 : gosub 6000
2280 next e
2290 print#4,lf$
2300 rem cimzes
2310 for j=1 to 8
2320 : if len(c$(j))>18 then 2340
2330 : if j=2 or j=5 or j=7 then print#4,d$;
2340 : print#4,tb$"25" f$ fm$(j) c$(j)
2350 : print#10
2360 next j
2370 print#4,ff$
2380 close 4: close 10
2390 goto 2010
2400 :
3000 rem +++++ katalogus +++++
3010 print chr$(14)" Katalogus"
3020 input " Dátum:";ans
3030 open 10,4,10 : print#10
3040 open 4,4,7
    
```





```

○ 3050 print#4,nl$ d$ an$ lf$lf$
3060 open 8,8,8,ct$+"r"
3070 i=1 : si=0
○ 3080 do while si=0
3090 : input#8,c$
3100 : print#4, d$ f$;i;c$ d0$ f0$
○ 3110 : for j=1 to 8
3120 : input#8,c$ : si=st
3130 : print#4,tb$"25" fm$(j) c$
○ 3140 : print#10
3150 : next j
3160 : print#4," _____"
○ 3170 : if i/6=int(i/6) then print#4,ff$
3180 : i=i+1
3190 loop
○ 3200 print#4,ff$
3210 close 8 :close 10 :close 4
3220 return
○ 3230 :
4000 rem <<<< adatbevitel >>>>
4010 print "☺"chr$(14)
○ 4020 bb$=" ##. #####☺"
4030 for j=0 to 8
4040 : print using bb$;j,m$(j);
○ 4050 : c$(j)=" " : input c$(j)
4060 : if len(c$(j))>35 then 4040
4070 next j
○ 4080 print "☺ ☺ Rendben? (i/n)☺"
4090 getkey q$
4100 if q$(">")="i" then 4010
○ 4110 open 9,8,9,ct$+"a"
4120 for j=0 to 8
4130 : print#9,c$(j)
4140 next j
○ 4150 close 9
4160 return
○ 4170 :

```

```

○ 5000 rem <<<< kereses >>>>
5010 print " * ☺☺☺ Vége☺☺"
○ 5020 input " ☺ K u l c s : *☺☺☺";k$
5030 if len(k$)=0 then 5020
5040 if k$="*" then return
5050 open 8,8,8,ct$+"r"
○ 5060 van=0
5070 do until van or st=64
5080 : input#8,c$(0) :printc$(0)
5090 : van=(k$=c$(0))
5100 : for i=1 to 8:input#8,c$(i):next i
○ 5110 loop
5120 if van then 5150
5130 print " ☺☺ Nincs ilyen!☺☺"
○ 5140 close 8 : goto 5020
5150 print"☺ ☺A címzett:☺☺"
5160 for j=0 to 8
○ 5170 : print " "c$(j)
5180 next j
5190 print "☺ ☺ Biztos? (i/n)☺"
○ 5200 getkey q$
5210 if st=64 then close 8 :return
5220 if q$="i" then close 8 :return
5230 van=0 : goto 5070
5240 :
○ 6000 rem <<<< ekstrak >>>>
6010 if note(e) then 6060
○ 6020 print#4,nb$ "U_____I"s1$
6030 print#4," |" r$f$ e$(e) r0$f0$ " |"
○ 6040 print#4,nb$ "J_____K"s2$
6050 print#10 : return
○ 6060 print#4,s1$lf$lf$s2$ : return

```

Feladó:  
**Dr. Hack Frigyes**  
Budapest XIV  
*Szikszó park 12.*  
 1 1 4 2  
**HUNGARY**

**AIR MAIL**

COMMODORE Business Machins;Inc.  
 1200 Wilson Drive

West Chester  
 Pen. 19380  
 U. S. A.

x



Feladó:  
 Dr. Hack Frigyes  
 Budapest XIV  
 Szikszó park 12.  
 1 1 4 2

Állomány

Tc.  
**COMMODORE ÚJSÁG**  
 Szerkesztőség

Budapest XIII.  
 Kárpát u. 7/a.

1133

# HÍREK

## A NEC-é a világ leggyorsabb számítógépe

### LEMARADNAK-E AZ AMERIKAI AK A SZUPERSZÁMÍTÓGÉP-GYÁRTÁSBAN?

A japán NEC konszern bejelentette, hogy megkonstruálta a világ jelenleg leggyorsabb szuperszámítógépét. Ezzel az eredménnyel a szigetország ismét egy nagy lépést tett, hogy a legkritikusabb műszaki területeken is behozza vagy túlszárnyalja az Egyesült Államokat. A korábbi leggyorsabb számítógépet az ugyancsak japán Fujitsu mondhatta magáénak a VP-2000 sorozatának tavaly decemberi bejelentésekor.

A NEC most bemutatott SX-3 sorozatának egyik legfontosabb jellemzője az, hogy követi a szóban forgó területen tevékenkedő amerikai gyártó, a Cray Research konstrukciós elveit. Ez azt jelenti, hogy több processzor egységet kapcsoltak egymáshoz, a műveleti sebesség megsokszorozása céljából. Az egymással párhuzamosan működő négy processzor lehetővé teszi, hogy másodpercenként 22 milliárd műveletet végezzenek el. Az amerikai felhasználók szemében az is vonzóvá teheti a NEC gépét, hogy az UNIX operációs rendszert (alaprogram rendszert) használja, amely egyre inkább ipari szabványvá változik az Egyesült Államokban.

A szuperszámítógépek a hagyományos elvű nagyszámítógépeknél is lényegesen drágábbak. Míg ez utóbbiak tipikus árai többnyire 1-5 millió dollár között mozognak, addig a NEC most bejelentett SX-3 gépe több mint 24 millió dollárba fog ke-

rülni. A szuperszámítógépek elsősorban abban különböznek a hagyományos berendezésektől, hogy képesek speciális matematikai műveletek elvégzésére. Jellemző, hogy nem egyszerű mennyiségekkel, hanem úgynevezett vektorokkal, illetve mátrixokkal dolgoznak. A szóban forgó adottságaik alapján elsősorban műszaki-tudományos számításokra, illetve a különösen számításigényes mérnöki—tervezői feladatokra használják. A legtipikusabb alkalmazási területek között szokták említeni a részecskekutatást, a repülőgép- és nukleárisfegyver-tervezést, az időjárás-előrejelzést. További fontos felhasználási lehetőség kínálkozik a vegyi elemzésben, a gyógyszergyártásban, a katonai hadműveletek tervezésében és lebonyolításuk szimulálásában, valamint a rejtjelkulcsok megfejtésében.

A NEC bejelentésével kapcsolatban külön felhívjuk a figyelmet arra a körülményre, amely megbéklyózza az amerikai konstruktőrök munkáját. Arról van szó, hogy az ilyen, rendkívül modern gépekhez különleges szempontok szerint konstruált félvezető alkatrészekre van szükség. A japán szuperszámítógép-gyártókat ezen a területen elsősorban az segíti, hogy egyszemélyben többnyire jelentős félvezetőgyártási és -fejlesztési kapacitásokat is sajátjuknak tudhatnak. Így lehetőségük nyílik arra, hogy a legújabb műszaki eredményeket már akkor elérjék, amikor amerikai



versenytársaik még a kritikus alkatrészek megjelenésére várnak. Ebben a helyzetben az Egyesült Államok említett gyártója többnyire csak abban bízhat, hogy még tart az az előny, amelyet a szoftver fejlesztésében korábban elért.

A NEC közleményétől látszólag teljesen független az az amerikai bejelentés, amely szerint az USA egyik szuperszámítógépgyártója feladja a további küzdelmet és kivonul a szóban forgó területről. Az ETA Systems nevű cégről van szó, amelynek a legfőbb részvényese, a Control Data mintegy 800 millió dollár veszteséget könyvelt el az utóbbi négy év során. Az általa irányított ETA tavalyi veszteségei elérték a 100 millió dollárt, az 1983-as megalapítása óta számított összesített hiánya azonban ennél lényegesen nagyobb.

A jelenlegi helyzetben fennáll az a veszély, hogy a csúcstechnikán belül is csúcsszintet képviselő szuperszámítógépgyártás ugyanúgy japán ellenőrzés alá kerül, mint korábban a közfogyasztási elektronikai termékek, illetve a félvezető-alkatrészek többségének előállítására. Erről azonban a Control Data elnöke azt mondta, hogy ez már nem az ő problémája, hanem az Egyesült Államoké. A szóban forgó műszaki terület feladásának ugyanis nem az az igazi tétje, hogy elvesz egy bizonyos piaci részesedés, hanem az, hogy mi lesz az amerikai versenyképességgel a XXI. században.

(Reuters)

## Az IBM illeszthetővé teszi nagygépeit a japán versenytársaikéihoz

„Az International Business Machines (IBM) még sohasem ment ilyen messzire a más számítógépgyártókkal való békés együttélés szándékának nyilvánításában” — így kommentálta az amerikai óriáscég új döntését a The Wall Street Journal című napilap. Az IBM bejelentette, hogy japán alvállalata júniusban megkezdi egy olyan illesztő egység (szaknyelven emulátor) árusítását, amelynek segítségével az IBM központi számítógépein (mainframe) futtathatók lesznek majd a Hitachi és a Fujitsu programjai.

Jóllehet az óriáscég nyereségének 10 százaléka származott a szigetországi eladásokból 1987-ben — azaz a legnagyobb hányad az USA-n belüli forgalmat leszámítva —, van mit visszaszereznie ezen a piacon. Szakmai elemzők szerint 15 évvel ezelőtt az IBM gyakorlatilag a kezében tartotta a japán központigép-eladásokat, később azonban elhanyagolta az árpolitikát, így 1987-ben már 28 százalékot mondhatott magáénak ezen a területen. (Összehasonlításképpen: a világ nagyszámítógépi-piacán az IBM részesedése 70 százalék körül mozog.) A szigetországi riválisok — a Hitachi és a Fujitsu — ezzel szemben

50 százalékos hányadot szereztek a japán központigép-eladásokból.

Ezek a cégek nagy lendülettel fogtak a szoftverek kidolgozásához, ám a programok rövidesen nem csak saját modelljeiken, hanem az IBM-gépeken is futtathatók lesznek. Az amerikai vállalat már eddig is kínált speciálisan átalakított szoftvereket, amelyeket a Hitachi- és Fujitsu-programokhoz — no meg a Prism márkanévű, 15,7 millió jenne kerülő átalakító egységhez — kapcsolva, lehetővé tették azok használatát az IBM számítógépein. Ez a megoldás azonban számottevően lelassította a műveleti sebességet. Az új illesztő berendezésben az említett „fordító-szoftver” már beégetett áramköri egységként (firmware) fog szerepelni, és szükségtelemné válik egyéb kiegészítő, így a Pris is.

Iparági szakértők rámutatnak: az amerikai óriáscég látványos döntése nem jelenti azt, hogy már országok számítógépgyártóinak termékeihez is hamarosan megszűnik az IBM-átalakító. Jóllehet az IBM már többször is szabadalomvédelmi vitába keveredett az említett japán vállalatokkal, a 60-as évekre visszanyúló licenccsere egyezmények következtében a felek operációs rendszerei rendkívül hasonlóvá váltak egymáshoz. Ez azt jelenti, hogy lényegesen könnyebb volt megalkotni az IBM-nek a japán gépekhez való emulátort, mint az amerikai tervektől alaposan eltérő modellekhez.

Azt is megemlítik, hogy ezzel a megoldással sem járul el minden akadály a szigetországi forgalomnövelés útjából. A japán vásárlók nem szívesen térnek át ugyanis új szállítóra a megismert helyett, főként mivel a szigetországi gyártók az IBM-nél jóval nagyobb karbantartó mérnöki állományt foglalkoztatnak, és ezzel jobban tudják segíteni a rendszerek üzemképességét. (AP—DJ)

## Erőteljes forgalomnövekedés az osztrák személyszámítógépi piacon

A Dataquest piackutató cég felmérése szerint Ausztriában tavaly 37 százalékkal több személyi számítógépet adtak el, mint 1987-ben. Az erőteljes növekedés elsősorban a nagyobb felhasználók beszerzéseinek élénküléséből származott. Ez a csoport a teljes személyszámítógépi-forgalomból 30 százalékkal részesedett. További számottevő felhasználói kategóriának minősítik a kisvállalkozókat, illetve az oktatási intézményeket. Az olcsó otthoni számítógépek részaránya kismértékben viszszaesett. Ez az irányzat annak ellenére is megfigyelhető volt, hogy az oktatásra szánt olcsóbb gépeket is alkalmassá tették az MS-DOS kompatibilis üzemmódban való felhasználásra.

A legnagyobb piaci részesedést az IBM érte el 27,4 százalékkal. A nagyvállalat a tavalyi év során 80 százalékkal növelte eladásait. További népszerű márkák Ausztriában az Atari, az Olivetti és az Apple. Számottevő mennyiségben adtak el a távol-keleti térségből származó IBM kompatibilis személyi számítógépeket.

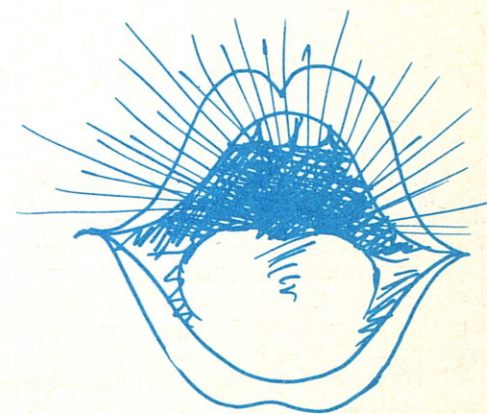
A Dataquest egyébként szövegezte, hogy a számítógépes szektorban az innovációs készséget erőteljesen fekezi a közületi szektor nagyarányú gazdasági jelenléte. Ugyancsak mérsékli az önálló kezdeményezést az utóbbi időben végrehajtott csökkentés ellenére is túlzottan minősített adóteher. A számítógépek gyors elterjedésének további akadályát jelenti a szakszervezetek részéről megnyilvánuló hagyományos ellenkezés, amely a munkahelyeket félti az új technika elterjedésétől (APA)

## Terjeszkedik Európában a Commodore

Az amerikai székhelyű Commodore cég azt tervezi, hogy tovább bővíti európai jelenlétét. A vállalat arra számít, hogy a Közös Piac 1992-re tervezett teljes integrációja után eladásai meghaladhatják az 1 milliárd dolláros küszöbértéket. A legutóbbi pénzügyi évben mintegy 600 millió dolláros forgalmat bonyolított le a cég a kontinensen.

Az igazgatóság azt tervezi, hogy tovább folytatja azt a sorozatot, amelyet 1982-ben a C-64-es típusal indított útjára. A legújabb modellek elsősorban a műveleti sebesség, illetve a színes megjelenítési technika területen múlják felül a korábbiakat, eközben azonban megfelelnek majd annak a követelménynek, hogy a kompatibilitás megmaradjon az utóbbiakkal.

Az európai forgalom bővítéséhez új üzemet szándékoznak építeni az NSZK-ban. Korábban Braunschweigben gyártott személyi számítógépeket a Commodore, a legutóbbi bejelentés szerint azonban külsőben áll a döntés egy új üzem felépítéséről. Ezzel párhuzamosan bővítik a franciaországi, az olaszországi és a spanyolországi szervezetüket is. (AP—DJ)







# PROGRAM

## DEC-HEX-BIN

A mellékelt program, a DEC-HEX-BIN akkor keletkezett, amikor gépi kódban kezdtem el programozni. Mint kezdő, igen megterhelőnek éreztem a különböző számrendszerek közti állandó átszámítgatást. Éppen ezért elhatároztam, hogy az első „hasznos” gépi kódú programom ilyen segédprogram lesz.

A program a decimális, hexadecimális és a kettes számrendszerek közti átszámításokat végzi el a másikakba.

Az indításhoz használjuk az alábbi utasítást:

```
LOAD "DEC-HEX-BIN",8,1
```

```
NEW:SYS 52000
```

Mivel a program az interruptban fut, azt egyszer inicializálni kell a SYS utasítással. Ezután minden interruptnál lekérdezzük az <F1>, <F3> és az <F5> billentyűket. Ezeknek a billentyűknek a megnyomásakor egy ezután a billentyűzeten beadott számot a másik két rendszerbe számítjuk át.

A kezelés igen egyszerű. Meg kell nyomni a funkciósbillentyűk egyikét. Hogy melyiket, az attól függ, hogy miről mire akarunk számítani. A gombnyomásra eltűnik a kurzor, töröljük a felső képernyői sort, és ott beadhatjuk a megfelelő számot. Pozitív egész értékek beadása lehetséges 0 és 65535 között. A bevétel korrigálása megtörténhet a <DEL> billentyűvel. A bevételt a <RETURN> gombbal kell lezárni. Ez azonban csak akkor lehetséges, ha a pontokkal ábrázolt bevételi mezőt kitöltöttük. A bevétel lezárása után a beadott érték mellett a felső sorban megjelenik a szám a másik két számrendszerben. Ezután a kurzor arra a helyre kerül, ahol az a funkciósbillentyű megnyomása előtt állt.

A funkciósbillentyűk hozzárendelése a következő:

<F1>: Decimális számok átszámítása. Ötjegyű decimális értéket kell beadni. A jelölés egy "d".

<F3>: Hexadecimális számok átszámítása. Négyjegyű hexadecimális értéket kell beadni. A jelölés a "\$".

<F5>: Bináris számok (bitminta) átszámítása. Tizenhat jegyes bevétel szükséges, a jelölés a "%".

Könnyű mintapéldát találunk. Például beadunk egy BASIC programot, és épp egy POKE utasításban van a sor. Szeretnénk tudni annak a hexadecimális számnak a decimális értékét, amelyiket ismerjük. Nyomjuk meg hát az <F3>-as gombot és adjuk be a hexa számjegyeket. A felső sorban a bevétel befejezése után megtaláljuk a számot mindhárom számrendszerben. Most azonnal beírhatjuk a program számára szükséges decimális értéket a POKE utasítás mögé, mert a kurzor pontosan oda kerül vissza, ahol az <F3> gomb megnyomása előtt állt.

Az indító cím 52000, így a segédprogram számos, a tárolóban a 49152-es címtől álló más programmal is együtt tud dolgozni. Természetesen a forrásprogramban megváltoztatott kezdőcím esetén a programunk a tároló más területeire is asszemblálható, azaz szabadon eltolható. A forráskód listát a Magazin e havi számához tartozó programszerviz lemezen találjuk.

```

1 rem *****
2 rem"* C= Ujsag          Sz.:185 *
3 rem"*          DEC - HEX - BIN *
4 rem"* Program:      Kovacs Zoltan *
5 rem *****
6 :
7 :
8 print chr$(147)"data beolvasas es ellenorzes ...";:j=52000:ve=53167:p=j
9 for b=0 to 7:read a$
10 l=asc(mid$(a$,2,1))
11 h=asc(mid$(a$,1,1))
12 l=l-48:if l>9 then l=l-7
13 h=h-48:if h>9 then h=h-7
14 print"#####|p;:p=p+1
15 if h>15 or l>15 then 17
16 a=h*16+l:poke j+b,a:t=t+a:next b:read a:if a=t then 18
17 print:print"data hiba ... sor:"peek(64)*256+peek(63):end
18 t=0:j=j+8:if j<ve then 9
19 print"#####kesz":end
20 data 78,a9,30,8d,14,03,a9,cb,0873
21 data 8d,15,03,58,20,7e,cb,60,0710
22 data a5,c5,c9,04,d0,0e,a2,80,1079
23 data 8e,0e,dc,20,71,cb,20,89,0893
24 data cb,4c,69,cb,a5,c5,c9,05,1155
25 data d0,0e,a2,80,8e,0e,dc,20,0920
26 data 71,cb,20,47,cc,4c,69,cb,1007
27 data a5,c5,c9,06,d0,10,a2,80,1083
28 data 8e,0e,dc,20,71,cb,20,ec,0992
29 data cc,a2,81,8e,0e,dc,4c,31,0996
30 data ea,a0,00,a9,20,99,00,04,0752
31 data c8,c0,28,90,f8,60,a9,0e,1103
32 data 20,16,e7,a9,80,8d,91,02,0870
33 data 60,d8,a2,04,8e,00,04,a2,0786
34 data 3a,8e,01,04,a0,00,a9,2e,0580
35 data 99,02,04,c8,c0,05,90,f8,0948

```



```

36 data a2,40,8e,8a,02,a0,00,8c,0808
37 data 3c,03,ee,3c,03,20,9f,ff,0810
38 data 20,e4,ff,c9,00,f0,f6,c9,1403
39 data 14,f0,19,c9,0d,f0,2a,ac,0953
40 data 3c,03,c0,06,b0,e7,c9,30,0917
41 data 90,e3,c9,3a,b0,df,99,01,1183
42 data 04,4c,aa,cb,ac,3c,03,c0,0880
43 data 02,90,d2,ce,3c,03,a9,2e,0840
44 data ac,3c,03,99,01,04,4c,ad,0642
45 data cb,ac,3c,03,c0,06,90,bd,0969
46 data a2,00,8e,8a,02,a2,36,8e,0802
47 data 40,03,a2,35,8e,41,03,8e,0634
48 data 42,03,8e,44,03,a2,33,8e,0637
49 data 43,03,a0,00,c8,b9,01,04,0620
50 data d9,3f,03,90,0f,d0,07,c0,0849
51 data 05,90,f1,4c,24,cc,20,71,0851
52 data cb,4c,89,cb,a2,02,86,22,0951
53 data a2,04,86,23,a9,05,20,b5,0722
54 data b7,a9,01,20,7e,bd,a2,50,0942
55 data a0,03,20,d4,bb,20,80,cd,0959
56 data 20,d1,cd,20,34,ce,60,d8,1048
57 data a2,24,8e,0a,04,a2,3a,8e,0716
58 data 0b,04,a0,00,a9,2e,99,0c,0555
59 data 04,c8,c0,04,90,f8,4c,68,0972
60 data cc,3a,3b,3c,3d,3e,3f,40,0631
61 data a2,40,8e,8a,02,a0,00,8c,0808
62 data 3c,03,ee,3c,03,20,9f,ff,0810
63 data 20,e4,ff,c9,00,f0,f6,c9,1403
64 data 14,f0,28,c9,0d,f0,39,ac,0983
65 data 3c,03,c0,05,b0,e7,a0,00,0827
66 data d9,61,cc,f0,e0,c8,c0,07,1381
67 data 90,f6,c9,30,90,d7,c9,47,1270
68 data b0,d3,ac,3c,03,99,0b,04,0790
69 data 4c,72,cc,ac,3c,03,c0,02,0823
70 data 90,c3,ce,3c,03,a9,2e,ac,0995
71 data 3c,03,99,0b,04,4c,75,cc,0628
72 data ac,3c,03,c0,05,90,ae,a2,0912
73 data 00,8e,8a,02,20,80,cd,20,0679
74 data 82,ce,a9,50,a0,03,20,a2,0942
75 data bb,a9,01,20,7e,bd,a2,50,0946
76 data a0,03,20,d4,bb,20,80,cd,0959
77 data 20,34,ce,60,d8,a9,25,8d,0949
78 data 13,04,a9,3a,8d,14,04,a0,0575
79 data 00,a9,2e,99,15,04,c8,c0,0785
80 data 10,90,f8,a2,40,8e,8a,02,0916
81 data a0,00,8c,3c,03,ee,3c,03,0664
82 data 20,9f,ff,20,e4,ff,c9,00,1162
83 data f0,f6,c9,14,f0,19,c9,0d,1186
84 data f0,2a,ac,3c,03,c0,11,b0,0902
85 data e7,c9,30,90,e3,c9,32,b0,1278
86 data df,99,14,04,4c,0d,cd,ac,0866
87 data 3c,03,c0,02,90,d2,ce,3c,0877
88 data 03,a9,2e,ac,3c,03,99,14,0626
89 data 04,4c,10,cd,ac,3c,03,c0,0728
90 data 11,90,bd,a2,00,8e,8a,02,0794
91 data 20,80,cd,20,1e,cf,20,d1,0875
92 data cd,60,30,31,32,33,34,35,0604
93 data 36,37,38,39,41,42,43,44,0488
94 data 45,46,2d,34,30,39,36,2d,0440
95 data 32,35,36,2d,31,36,2d,31,0399
96 data a2,72,86,22,a2,cd,86,23,0980
97 data a9,05,20,b5,b7,a2,60,a0,0988
98 data 03,20,d4,bb,a2,77,86,22,0883
99 data a2,cd,86,23,a9,04,20,b5,0922
100 data b7,a2,70,a0,03,20,d4,bb,1051
101 data a2,7b,86,22,a2,cd,86,23,0989
102 data a9,03,20,b5,b7,a2,80,a0,1018
103 data 03,20,d4,bb,a2,7e,86,22,0890
104 data a2,cd,86,23,a9,02,20,b5,0920
105 data b7,a2,90,a0,03,20,d4,bb,1083
    
```

```

106 data 60,a9,50,a0,03,20,a2,bb,0889
107 data a0,00,8c,3c,03,8c,3d,03,0567
108 data a0,60,8c,3e,03,a2,24,8e,0801
109 data 0a,04,a2,3a,8e,0b,04,ad,0564
110 data 3e,03,a0,03,20,67,b8,20,0579
111 data 2b,bc,c9,01,d0,06,ee,3c,0945
112 data 03,4c,ef,cd,ac,3c,03,ae,0932
113 data 3d,03,b9,62,cd,9d,0c,04,0725
114 data ad,3e,03,a0,03,20,50,b8,0697
115 data 20,b4,bf,18,ad,3e,03,69,0770
116 data 10,8d,3e,03,a0,00,8c,3c,0582
117 data 03,ee,3d,03,ac,3d,03,c0,0733
118 data 04,90,bc,60,a9,50,a0,03,0844
119 data 20,a2,bb,a9,90,a0,03,20,0889
120 data 67,b8,20,f7,b7,8d,aa,03,1063
121 data 8c,ab,03,a9,25,8d,13,04,0684
122 data a9,3a,8d,14,04,a0,00,0e,0566
123 data aa,03,b0,05,a9,30,4c,63,0746
124 data ce,a9,31,99,15,04,c8,c0,0994
125 data 08,90,ec,a0,00,0e,ab,03,0736
126 data b0,05,a9,30,4c,79,ce,a9,0970
127 data 31,99,1d,04,c8,c0,08,90,0779
128 data ec,60,a0,00,8c,3c,03,8c,0835
129 data 3d,03,a0,60,8c,3e,03,20,0557
130 data 0c,bc,20,53,b8,a2,50,a0,0901
131 data 03,20,d4,bb,ad,3e,03,a0,0832
132 data 03,20,a2,bb,20,b4,bf,ae,0961
133 data 3e,03,a0,03,20,d4,bb,ad,0832
134 data 3e,03,a0,03,20,50,b8,ac,0696
135 data 3c,03,b9,0c,04,ac,3d,03,0500
136 data d9,62,cd,f0,0e,ad,3e,03,1012
137 data a0,03,20,67,b8,ee,3d,03,0784
138 data 4c,b7,ce,a9,50,a0,03,20,0909
139 data 67,b8,a2,50,a0,03,20,d4,0936
140 data bb,a0,00,8c,3d,03,18,ad,0748
141 data 3e,03,69,10,8d,3e,03,ee,0630
142 data 3c,03,ac,3c,03,c0,04,90,0638
143 data a3,a9,50,a0,03,20,a2,bb,0956
144 data 20,dd,bd,a0,00,b9,00,01,0788
145 data c9,00,f0,07,99,01,04,c8,0806
146 data 4c,05,cf,a9,04,8d,00,04,0606
147 data a9,3a,8d,01,04,60,a0,00,0629
148 data a9,00,be,15,04,e0,31,f0,0897
149 data 05,18,2a,4c,30,cf,38,2a,0500
150 data c8,c0,08,90,ed,8d,ab,03,1096
151 data a0,00,a9,00,be,1d,04,e0,0776
152 data 31,f0,05,18,2a,4c,4a,cf,0717
153 data 38,2a,c8,c0,08,90,ed,8d,1020
154 data aa,03,a9,70,a0,03,20,a2,0811
155 data bb,20,b4,bf,20,0c,bc,ac,0994
156 data ab,03,20,a2,b3,20,30,ba,0813
157 data 20,0c,bc,ac,aa,03,20,a2,0771
158 data b3,20,6a,b8,a2,50,a0,03,0906
159 data 20,d4,bb,20,dd,bd,a0,00,1033
160 data b9,00,01,c9,00,f0,07,99,0787
161 data 01,04,c8,4c,80,cf,a9,04,0789
162 data 8d,00,04,a9,3a,8d,01,04,0518
163 data a9,50,a0,03,20,a2,bb,a9,0962
164 data 01,20,7e,bd,a2,50,a0,03,0753
165 data 20,d4,bb,60,ff,00,ff,00,1037
    
```

ready.

## Két apróság C+4-re

Az alábbi két kis program különösebb magyarázatot nem igényel, futtatáskor kiírják a képernyőre, amit tudni kell róluk.

Lörentei — Debre



```

0 20 PRINTCHR$(147)
30 FORZ=3592TO4071
40 POKEZ,64:NEXT
41 PRINT:PRINT:PRINT
50 PRINTSPC(7)"A PROGRAMOT FUTTATVA"
60 PRINTSPC(7)"A KEPERNYO ALSO ES FELSO"
70 PRINTSPC(7)"RESZE EGY GOMBNYOMASRA"
80 PRINTSPC(7)" FELCSERELODIK."
170 DATA A2,00,0D,08,0E,A9,0D,00
180 DATA 0C,9D,08,0E,98,9D,00,0C
190 DATA CA,00,0F,0D,08,0F,A9,0D
200 DATA 00,0D,9D,08,0F,98,9D,00
210 DATA 0D,E8,E8,E8,0D,ED,50,00
220 FORT=0TO39:READA#
230 POKE819+T,0EC(A#):NEXT
260 GETKEYA#:IFA#=" "THENEND
270 SYS819:GOTO260
    
```

○ READY.

```

0 10 PRINTCHR$(147);CHR$(14)
20 FORT=0TO35:READA:POKE819+T,A:NEXT
30 DATA 169,000,201,193,176,009
40 DATA 201,161,176,012,201,064
50 DATA 176,015,096,056,233,128
60 DATA 141,052,003,096,056,233
70 DATA 064,141,052,003,096,056
80 DATA 233,064,141,052,003,096
90 GOTO120
100 AK=ASC(A#):POKE20,AK:SYS819
110 A=PEEK(20):RETURN
120 PRINT" * LEUTOTT BILLENYU * KODJAT"
130 PRINT" KEPERNYOKODOR ALAKITVA IRATJUK"
140 PRINT" A KEPERNYORE"
150 GETKEYA#:GOSUB100
160 PRINTCHR$(147):PRINT
170 PRINT" KARAKTER : "A#:PRINT
180 PRINT" KARAKTERKOD : "AK:PRINT
190 PRINT" KEPERNYOKOD : "A :PRINT
200 GOTO150
    
```

○ READY.

## Object save

Gondot okoz a gépi kódú programok kimentése? Nem akarjuk a BASIC kezdet-, végcímeiket összevissza POKE-olni? Írjunk gépi kódú programot a feladat megoldására! Igen ám, de hol helyezzük el? Az ilyen program mindig pont ott van, ahol a kimentendő tárgyprogramoknak kellene lenni. Mindig „relakálni”, szintén némi kényelmetlenséggel jár. Egy kb. 60 byte-nyi program elhelyezése BASIC sorban mindig megoldja az át-helyezhetőség kérdését is. És még a felhívási cím kiszámolásával sem kell bajlódni, ha így építjük be a programunkba: SYS(PEEK(43)+256\*PEEK(44)+6), "NÉV", készülékszám, kezdő-cím, végcím+1.

(Előfordulhat ugyanis, hogy a gépi kódot generáló programunkat nem a normál BASIC területre — 2049-től kezdődően —, hanem „feljebb tolva” kell betöltenünk.)

A rutin tartalmaz egy 0-val való töltést. Ezt nem tehetjük a REM sorba, mert mint az köztudott, a 0 byte a BASIC sor végét jelöli. Egyetlen sor módosítása után igen furcsán festene programunk, s az már másodlagos, hogy használhatatlan is lenne. Ezért ezt a műveletet egy nulláslapi cím (2-es) tartalmának betöltésével pótoltuk, melynek értéke a legtöbb esetben 0. (Legtöbb esetben; itt azt jelenti, hogy egy más programmal szándékosan nem írtunk bele 0-tól eltérő számot.)

Akinek ez a néhány mondat kínaiul van, annak csak annyit, hogy a rutint csak akkor használhatjuk zavarmentesen, ha a PRINT PEEK (2) utasítás 0-t ad eredményül.

Ezt a kis programot be is építhetjük a C= Újság hasábjain gyakorta megjelenő gépi kódot generáló programunkba. (Mert ugye a fejrészt már mindenki külön kimentette, aki saját kezűleg pötyögi be a programokat?) Nos, a fejrészünk ezután így néz ki:

Folytatás a 65. oldalon

```

0 0 rem *.....*
1 rem *****
2 rem"* C= Ujsag Ssz.:186 *
3 rem"* SYS2055,"nev",ksz,kcim,vcim+1*
4 rem"* Program: Kovacs Zoltan *
5 rem *****
6 :
7 :
8 print chr$(147)"data beolvasas es ellenorzes ...";:j=02055:ve=02118:p=j
9 for b=0 to 7:read a$
10 l=asc(mid$(a$,2,1))
11 h=asc(mid$(a$,1,1))
12 l=l-48:if l>9 then l=l-7
13 h=h-48:if h>9 then h=h-7
14 print"#####"p;:p=p+1
15 if h>15 or l>15 then 17
16 a=h*16+l:poke j+b,a:t=t+a:next b:read a:if a=t then 18
17 print:print"data hiba ... sor:"peek(64)*256+peek(63):end
18 t=0:j=j+8:if j<ve then 9
19 print"#####kesz":stop:sys(peek(43)+256*peek(44)+6),"fejresz",8,2049,3019
20 data 20,fd,ae,20,57,e2,20,fd,1089
21 data ae,20,9e,b7,a4,02,20,ba,0931
22 data ff,20,fd,ae,20,8a,ad,20,1089
23 data f7,b7,48,98,48,20,fd,ae,1185
24 data 20,8a,ad,20,f7,b7,48,98,1029
25 data aa,68,a8,68,85,14,68,85,0936
26 data 15,a9,14,20,d8,ff,90,03,0860
27 data 4c,f9,e0,60,20,2a,2a,2a,0803
    
```



# Hardver kiegészítők

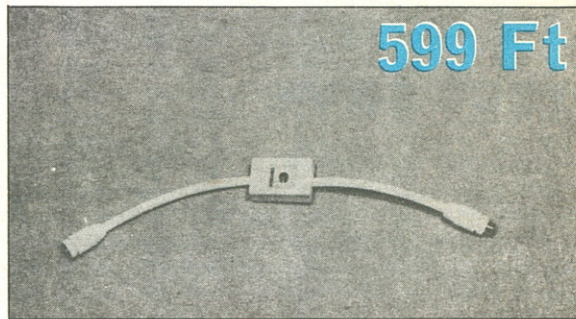
1. Enterprise 128 KByte-os számítógép és magnó

11 900 Ft



4. Enterprise impedancia-illesztő

599 Ft



2. Enterprise Euroscart monitorkábel

1 297 Ft



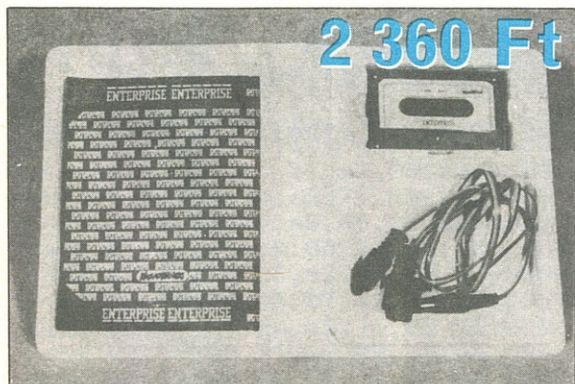
5. Enterprise ZX Emulátor és SYSTEM BUS BRIDGE

7 770 Ft



3. Commodore Multi File Transfer

2 360 Ft



6. Enterprise Joy-illesztő

456 Ft





# gészítők

A NOVOTRADE RT.  
2C Áruházában  
megvásárolhatók

7. Enterprise alapgép és  
magnó porvédő



10. Enterprise Speak EASY



8. Enterprise EX-DOS illesztőkártya



11. Enterprise DIN monitorkábel



9. Enterprise Mouse



Számítástechnikánk a jövő technikája

Örömmel látjuk a

# 2c

áruházban

1136 Budapest XIII., Balzac u. 35. Tel. 402-954  
Nyitva: hétfőtől péntekig 9-től 18 óráig



# EGYESÜLETI TAGOK FÓRUMA

## C 16, Plus/4

PLUS4 + 1551 floppy + Joystick + 5 lemez + 150 program + irodalom reklámáron, sürgősen eladó.  
Németh Géza, 5430 Tiszaföldvár, Kinizsi krt. 11/A.

C16 és PLUS4 játékprogramokat keresek! Címlistát és ármejelölést kérek!  
Zsöllei Attila, 8440 Herend, Rákóczi u. 5.

Programcsere! PLUS4 programokat cserélek vagy veszek. Csak felhasználói programok érdekelnek!  
Koleszár Zoltán, Kont út 11/A.

C16 + magnó és PLUS4 + magnó eladó.  
Kulcsár György, 2060 Bicske, Somogyi B. u. 20/A.

## Vegyes

C64, C128, PLUS4 programokat cserélek lemezen.  
Kővári Attila, 1193 Budapest, Derkovits u. 62. Telefon: 47-11-77.

Minden típusú Commodore személyi számítógéphez sok csatornás vezérlő hardwert, szoftwert adok olcsón.  
T. Nagy József, 6000 Kecskemét, Fehér u. 20.

48K SPECTRUM, KEMPSTON interfész, fényceruza, sok program és irodalom eladó vagy 1541-es floppyra cserélhető, értékegyeztetéssel.  
Illyés István, 2119 Pécel, Blaha Lujza u. 12.

PLUS4-hez és C64-hez kedvező áron kínáljuk az alábbi kiegészítőket, postai szállítással: FÉNYCERUZA 1300 Ft, JOYSTICK 675 Ft.  
Computeam GM, 7400 Kaposvár, Berzsenyi u. 32.

## C 64

Eladó új C64 alapgép, 1541 floppy, 180 darab játékprogram. Érdeklődni: 168-179 telefonon, Fock Lajosnál. Irányár: 40 000 Ft.

C64 programokat nagy választékból eladok, valamint igényes játék- és felhasználói programokat cserélek.  
TIGGER SOFTWARE, 3635 Dubicsány, Rákóczi u. 3.

A következő C64-es programokat keresem: THE LAST NINJA, ANTIRIAD, THE CURSE OF SHERWOOD, STRIKE FORCE COBRA.  
Bertók Antal, 1181 Budapest, Mátyás király u. 34.

C64 programokat cserélek lemezen. Listát kérek és küldök. Keresem: SUPERGRAPHIC, SYNTHIMAT programokat és leírásukat.  
Valuska Tamás, 5500 Gyomaendrőd, Mikszáth u. 50.

Régi, nem szögletes dobozos, C 64-est, javításra szoruló vagy javíthatatlant keresek megvételre, ajánlatokat levélben, ár és hiba megjelöléssel kérem.

Kerekes Géza, Törökbálint, Tulipán u. 6. 2045

C 64 programokat cserélek lemezen. Keresem a DEFENDER OF THE CROWN című játékot. Listát kérek és küldök.  
Matalik Krisztián, Gyöngyöspata, Szücsi út 11. 3035

Keresem a MAUS 4 című programot, megvételre vagy cserére. Lehetőleg leírással.  
Balló Attila, Budapest, Erzsébet Királyné útja 1/B. 1146

A VISASTAR 64 program magyar nyelvű leírását keresem.  
Miklya János, 6726 Szeged, Fésű u. 2/D. IV. 17.

Keresem a MERCENARY programot kazettán és más programokat is cserélek.  
Székely Krisztián, Győr, ifjúság kört. 41. I/4.

Keresem C64-re a következő programokat: IDŐRÉGÉSZ, BEACH HEAD 3, BARBARIAN 3, ALLIGATOR, DUN DARACH, EXCALIBUR, DARK SCEPTRE. Ezenkívül bármilyen ZOMBIE-program érdekel.  
Németh András, 2030 Érd, Kárpitos u. 41.

C64-es programokat cserélek lemezen. Kizárólag újdonságok érdekelnek. Cserealap: 1000 program. Keresem: GRAPHIC ADV. CREATOR, PROF. ADV. W. SYSTEM, GEOPUBLISH.  
Németi Ferenc, 1182 Budapest, Nagyenyed u. 8/A.



Fontos, hogy a 0. sorunk legalább 65 byte-ot tartalmazzon a REM után!!! Írjuk be a 20. sortól kezdődően a fenti 8 DATA sort! Futtassuk le a programot! A 0. sorunk elég furán fog kinézni, de ezzel ne törődjünk. Fontos, hogy ezt a 0-ás sort ezután már ne editáljuk!!!

Töröljük a DATA sorokat, és máris letárolhatjuk az új fejrész. Ha mindezt még kiegészítjük a Databevivő rutinnal (C= Újság 1988/9), akkor már egy majdnem komfortos rutinunk van a gépi kódú programok be- és kivitelére.

Ja, majd elfelejtettem. Az általam közölt gépi kódú programok bevivő része azért olyan kacifántos, mert hexadecimális szá-

mokat kell bevinni. Azokat könnyebb felismerni és megjegyezni. Ez pedig megér egy egyszeri fáradságos többletmunkát. Nem igaz?

Végül a SAVE rutin assembler listája:

A fenti „fejrész” programot a STOP után CONT-tal ki is menthetjük lemezre (vagy kazettára, de akkor a készülékszám = 1), ha a byte-ok száma a listán megadottal egyező. Ha nem (azaz egy betűvel is több, vagy kevesebb, akkor tessék számolni (PEEK(45) és PEEK(46) alapján), és módosítani a címeket!

*Kovács Zoltán*

```

0  .,0807 20 fd ae jsr $aeafd ; 1. vessző
    .,080a 20 57 e2 jsr $e257 ; filenév
0  .,080d 20 fd ae jsr $aeafd ; 2. vessző
    .,0810 20 9e b7 jsr $b79e ; készülékszám
0  .,0813 a4 02 ldy #02 ; az a bizonyos 0. byte, másodlagos cím
    .,0815 20 ba ff jsr $ffbfa ; fileparaméter beállítás
0  .,0818 20 fd ae jsr $aeafd ; 3. vessző
    .,081b 20 8a ad jsr $ad8a ; FRMNUM, kezdőcím
0  .,081e 20 f7 b7 jsr $b7f7 ; $14/15 és Y/A szerint
    .,0821 48 pha ; HIGH-byte
    .,0822 98 tya
0  .,0823 48 pha ; LOW-byte
    .,0824 20 fd ae jsr $aeafd ; 4. vessző
0  .,0827 20 8a ad jsr $ad8a ; végcím
    .,082a 20 f7 b7 jsr $b7f7 ; $14/15 és Y/A szerint
0  .,082d 48 pha
    .,082e 98 tya
0  .,082f aa tax
    .,0830 68 pla
0  .,0831 a8 tay ; végcím-LOW X-be, HIGH Y-ba
    .,0832 68 pla ; kezdőcím LOW-byte
0  .,0833 85 14 sta #14
    .,0835 68 pla ; kezdőcím HIGH-byte
0  .,0836 85 15 sta #15 ; kezdőcím $14/15-be
    .,0838 a9 14 lda ##14 ; a kezdőcím címe
0  .,083a 20 d8 ff jsr $ffd8 ; SAVE
0  .,083d 90 03 bcc #0842 ; nincs hiba
    .,083f 4c f9 e0 jmp $e0f9 ; hibaüzenet
0  .,0842 60 rts
    
```

## Amigások, figyelem!

AMIGA-szekció alakult a pécsi Commodore Klubban. Amigások jelentkezését várjuk az ország minden részéről.

Minden Amigás szerveződéssel szívesen felvesszük a kapcsolatot, együttműködés céljából.

Címünk: Pécsi Nevelési Központ, Commodore Klub  
7632 Pécs, Apáczai Csere János körtér.  
Makány György

## Tisztelt Szerkesztőség!

Van egy AMIGA 500-as számítógémem, amihez német nyelvű leírást és egy AMIGA Basic könyvet mellékeltek. Különböző könyvesboltokban érdeklődtem, de magyar nyelvű fordítása nincs, szakirodalom e típushoz nem található.

Örömmel értesültem az áprilisi COMMODORE Újságból, hogy egy cikket közöltek róla.

A magyar ékezetes betűk megjelenítése számomra is érdekes.

Szeretném, ha lapjuk hasábjain e géppel kapcsolatban több cikk is megjelenne.

Tisztelettel:

*Szabó Mihály*





# kedvezmények

A kedvezmény a megjelenéstől számított egy hónapig érvényes.



**KEDVEZMÉNYEK a PC-Szalontól a Szuperpáholy tagok részére!**  
**IMPOSSIBLE MISSION**

II. 15% **781,— ÁFÁVAL**  
játékprogram  
**AMBASSADOR PC szövegszerkesztő**  
10% **5950,— ÁFÁVAL**

A NOVOTRADE RT. 2C  
Áruházában az Egyesület  
PLUSZ- és SZUPER PÁHOLYÁNAK  
tagjai kedvezményrel  
vásárolhatják meg a következő programokat:

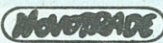
	régi ár	új ár
Sulikomp matek	149 Ft	134 Ft
fizika I.	149 Ft	134 Ft
történelem	149 Ft	134 Ft
fizika II.	149 Ft	134 Ft
	régi ár	új ár
Sulikomp matek + kazetta	429 Ft	386 Ft
fizika + kazetta	429 Ft	386 Ft
Fényceruza	2500 Ft	2250 Ft
Fizikomp	129 Ft	116 Ft
Charlie tesztkészítő program + kazetta	399 Ft	359 Ft
Nyuszi olvasni tanít sorozat I—VI.		
	régi ár	új ár
C+4 kazetta	1165 Ft	1048,50 Ft
C—64 kazetta	1165 Ft	1048,50 Ft
Szókincstár C—64 lemez	1828 Ft	1645,00 Ft
Nebuló C—64 oktatólemez	1165 Ft	1048,50 Ft



## júliusi—augusztusi 120 forintos vásárlási utalvány

Beváltható készpénzes  
vásárlás esetén az  
ÁPISZ szaküzleteiben  
XI., Budafoki út 7.  
VIII., Szigony u. 15.

Érvényes: 1989. november 30.



## júliusi—augusztusi 120 forintos vásárlási utalvány

Beváltható készpénzes  
vásárlás esetén a 2C  
áruházban XIII., Balzac u. 35.  
és a Művelt Nép  
vidéki boltjaiban működő 2C sarkokban.

Érvényes: 1989. szeptember 30.

A Newline számítástechnikai vállalkozás 10% kedvezményt ad az egyesület tagjainak:

C 16 beépíthető 64 KByte memóriabővítő

16—64-es átkapcsoló	1990,— Ft
beépítés munkadíja	150,— Ft
ROMTURBO 16	490,— Ft
együtt megrendelés esetén	770,— Ft
árengedménnyel:	3400,— Ft
	3060,— Ft

Jogosultak: a Plusz- és a Szuperpáholy tagjai  
Igazolás: ennek a tikkettnek postai elküldésével.  
Cím: Newline, 1014 Budapest, Tárnok u. 26. 1/5.



A Fotelelektronik-Novotrade-Alfa KFT az alább felsorolt szervezeiben mindenféle szervizszolgáltatás munkadíjából és az 1989. évben megkötött átalánydíjas szerződés összegéből 10% kedvezményt ad az egyesületi tagoknak.

Határidő: nincs, illetve átalánydíjas szerződésnél 1989. 12. 01.

A kedvezményt nyújtó szervezeink:

1053 Budapest, Magyar u. 12—14.	Telefon: 173-551
1083 Budapest, Szigony u. 9.	Telefon: 343-153
1191 Budapest, Gábor Á. sétány 3.	Telefon: 274-763
3100 Salgótarján, Arany J. u. 3.	Telefon: 32-14-007
3525 Miskolc, Fazekas u. 1—3.	Telefon: 46-17-011
4034 Debrecen, Holló L. u. 14.	Telefon: 52-32-863
5600 Békéscsaba, Bartók B. u. 37.	Telefon: 66-27-195
6726 Szeged, Csongrádi sugárút 76.	Telefon: 62-13-377
7400 Kaposvár, Füredi u. 24.	Telefon: 82-16-307
7624 Pécs, Jurisics M. u. 17.	Telefon: 72-11-812
8000 Székesfehérvár, Széchenyi u. 15/a.	Telefon: 22-12-711

9024 Győr, Bem J. tér 1.	Telefon: 96-12-802
9700 Szombathely, Szalonok u. 31.	Telefon: 94-13-419

Igazolás: a javítandó berendezés leadásakor egyesületi igazolvánnyal.  
A kedvezmény többször is igénybe vehető.







# KOGINFORM

MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET



**KOGINFORM** Levélcím: 1325 Újpest 1. Pf.: 159 Telefon: 894-642, 894-756 **KOGINFORM**

## Felhasználói rendszerek IBM XT és AT számítógépekre

Vezetésszolgáltatási Információs Rendszer	120 000 Ft + ÁFA
Anyag- és Félkésztermék Nyilvántartási Rendszer	200 000 Ft + ÁFA
Pénzügyi és Főkönyvi Könyvelési Rendszer	250 000 Ft + ÁFA
Teljesítményelszámolási Rendszer	90 000 Ft + ÁFA
Munkaügyi és Személyzeti Nyilvántartó Rendszer	150 000 Ft + ÁFA
Bérszámfejtési Rendszer	180 000 Ft + ÁFA
Termelésirányítási Rendszer	250 000 Ft + ÁFA
Értékesítési, Készáru-nyilvántartási és Számlázási Rendszer	150 000 Ft + ÁFA

### Szolgáltatásaink

- A rendszert a Megrendelő igényeihez adaptáljuk — egy szakemberünk egy havi munkájával térítésmentesen, azon felül meg egyezés szerint.
- Betanítjuk a rendszert ingyenesen.
- A rendszer beüzemelését szolgáló próbaüzem időtartama alatt — kívánságra — programozói, ill. operátori ügyeletet biztosítunk.
- A rendszer átadását követő 5 éven keresztül térítésmentesen elvégezzük a szabályozórendszer változásainak következtében felmerülő módosításokat évi egy ember/hónap munkaigényig.
- Számítógéppel nem rendelkező ügyfeleink részére programrendszereket az üzemeltetésükhöz szükséges számítógép-konfigurációval együtt biztosítjuk:
  - az eszközöket eladjuk,
  - az eszközöket bérbe adjuk,
  - hálózatot telepítünk.

A legtöbb helyen saját gépeinket kihelyezzük a felhasználóhoz, és azokon ún. gépi adatfeldolgozó kapacitásszolgáltatást nyújtunk.

Áraink munkahelyenként értendők, a megvásárolt darabszámtól függően árengedményt adunk.

Ez a mennyiségi árengedmény érvényes akkor is, ha több különböző Rendszerünket vásárolják meg.

Oktatási és egészségügyi intézmények részére további 50%-os árengedményt adunk.



# KOGINFORM

MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET  
SZÁMÍTÁSTECHNIKAI ÁGAZAT



Telefon: 898-279

**KOGINFORM**

**KOGINFORM**



Ára: 96 Ft



# KOGINFORM

MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET



KOGINFORM Levélcím: 1325 Újpest 1. Pf.: 159 Telefon: 894-642, 894-756 KOGINFORM

*Műszaki Fejlesztő  
Kisszövetkezet  
hardver árajánlata  
1989.*

### IBM PC XT

- 640 Kbyte RAM
  - 360 Kbyte floppy drive
  - 20 Mbyte Winchester
  - monochrom monitor
  - color monitor
- 180 000 Ft + ÁFA  
205 000 Ft + ÁFA

### IBM PC AT

- 640 Kbyte RAM
  - 1,2 Mbyte floppy drive
  - 20 Mbyte Winchester
  - monochrom monitor
  - color monitor
- 215 000 Ft + ÁFA  
240 000 Ft + ÁFA

### IBM AT/386

- 2 Mbyte RAM
  - 1,2 Mbyte floppy drive
  - 40 Mbyte Winchester
  - monochrom monitor
  - color monitor
  - EGA monitor
- 535 000 Ft + ÁFA  
560 000 Ft + ÁFA  
595 000 Ft + ÁFA

### FX 1050 printer

- Szünetmentes tápegység (300 W)
  - ArcNet kártya
  - Hálózatkiépítés
- 99 000 Ft + ÁFA  
62 000 Ft + ÁFA  
30 000 Ft + ÁFA  
500 Ft/m + ÁFA

**Bármilyen típusú (pl. C64),  
de működőképes  
számítógépet beszámítunk!**



# KOGINFORM

MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET

SZÁMÍTÁSTECHNIKAI ÁGAZAT

Telefon: 898-279



KOGINFORM

KOGINFORM