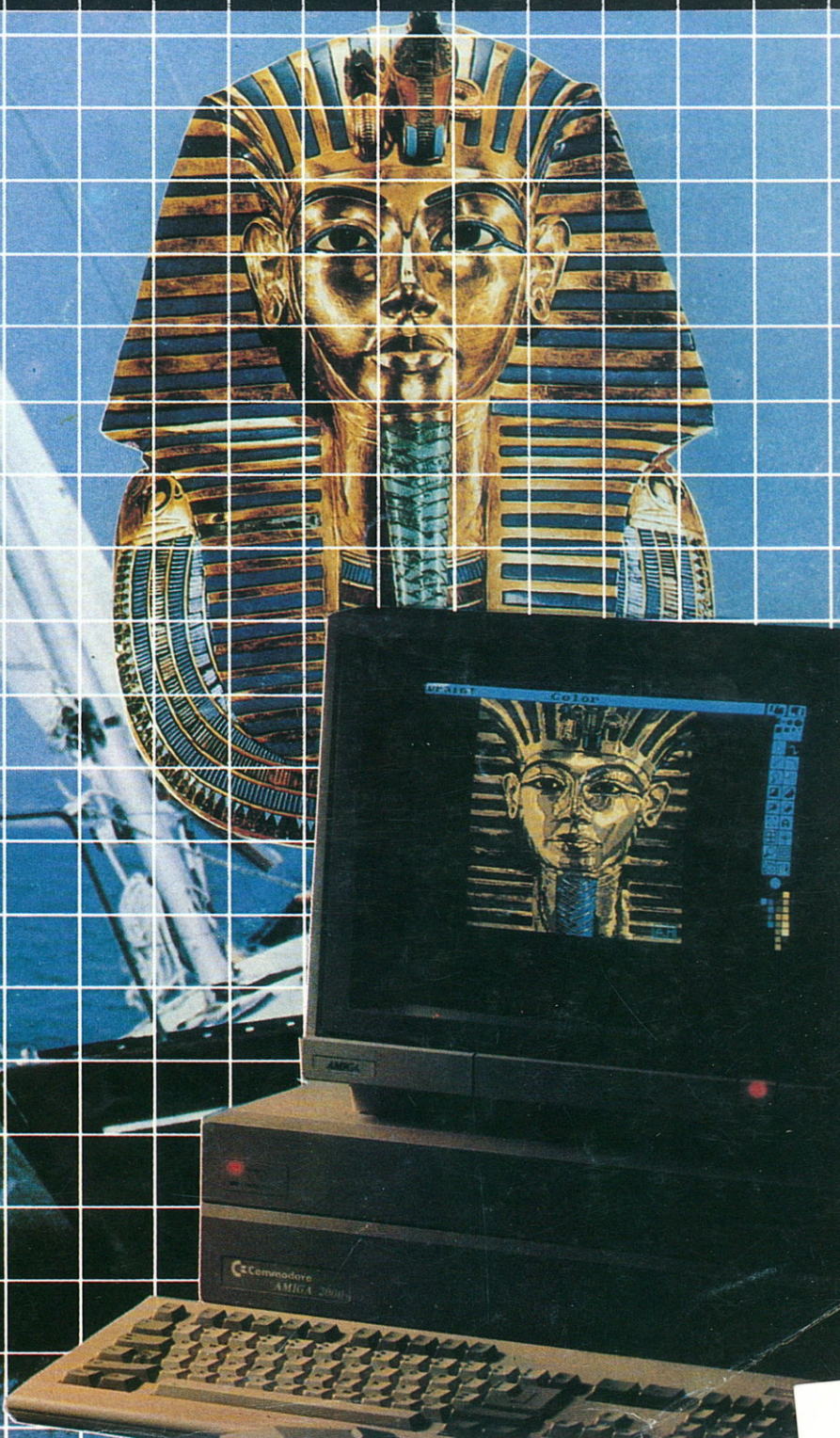


Az Országos Commodore Egyesület lapja

újság

1990/6

A LEGKISEBB PC
SZÍNEK A FALON
AUTOSTART
TIPPEK TRÜKKÖK
PROGRAMOK
KÖNYVEK



NOVOTRADE

commodore

WELTEXPORT

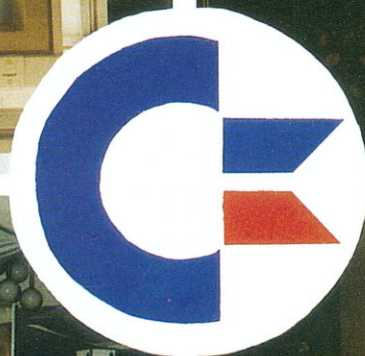
AMIGA

A tavaszi Commodore termékbemutatók után

AMIGA

a 2C-ben

Budapest XIII., Balzac utca 35.



C-64 alapgép
VC 1541/II. drive
MPS 1230
Datasette 1530
Monochrom monitor
Joystick

17 000,- Ft
18 500,- Ft
23 000,- Ft
3 000,- Ft
14 000,- Ft
1 300,- Ft

Amiga 500
Amiga harddisk (20 MB)
Amiga color monitor (1400)
Amiga távbővítő (1 MB)
Amiga RF modulátor
C-64 mouse

54 600,- Ft
5 500,- Ft
35 000,- Ft
17 000,- Ft
2 990,- Ft
2 900,- Ft

MIT, HOGYAN, HOL, MIKOR?

EGYESÜLETI ÜGYEK: Egyesületünk tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az Egyesület irodájában (**1132 Budapest, Visegrádi utca 38/A. II. em. Tel.: 129-5048**), vagy átutalással az MNB 217-98292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

Minden tagunk rendelkezésére áll **PÖTYÖGŐSZOLGÁLTATÁS**, a szervizkedvezmény és az apróhirdetési lehetőség.

A **DEÁKPÁHOLY** tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy évre 666 forint.

A **PLUSZPÁHOLY** tagjai minden hónapban megkapják a C-újságot, és minden hónapban 2 x 60 forint vásárlási utalványt is kapnak. A tagsági díj egy évre 1777 forint.

A **SZUPERPÁHOLY** tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 1800 forint vásárlási utalványt. Az éves tagsági díj 19 100 forint.

ÜGYFÉLFOGADÁS: minden kedden és csütörtökön 14.30—18 óra között várjuk tagjainkat és az érdeklődőket.

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

1989 novemberétől a pöttyögőszolgálat időpontja az alábbiak szerint változik: minden harmadik héten kedden és csütörtökön 15 órától 20 óráig az Egyesület irodájában

1132 Budapest, Visegrádi utca 38/A. II. em. Tel.: 12-95-048

PÖTYÖGŐSZOLGÁLTATÁS JÚNIUS: 5., 7., 26., 28.

A felsorolt napokon 15—20 óra között várjuk az érdeklődőket.

APRÓHIRDETÉS: Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 80 forint. A hirdetés 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 **12-95-048-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 Andre Gimnázium

Pécsi Apáczai Csere János Gimnázium

Az Országos Commodore Egyesület módszertani kiadványa.

Egyesületi iroda és szerkesztőség:

1132 Budapest, Visegrádi utca 38/A. II. em. Tel.: 12-95-048

Felelős kiadó: Horváth Judit, az egyesület elnöke

Főszerkesztő: Rados Péter, az OCE főtítkára

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

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

PÉCS: Zrínyi Miklós Könyvesbolt. 7621 Jókai u. 25. Tel.: 72-14988
DEBRECEN: Szak- és Ismeretterjesztő Könyvárúház. 4024 Hunyadi u. 8. Tel.: 52-16091

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.: 66-25207

GYŐR: Pattantyús Á. Géza Szakkönyvesbolt. 9022 Molnár Ferenc u. 9. Tel.: 96-22533

SZEGED: Tömörkény Könyvesbolt. 6720 Lenin krt. 48. Tel.: 62-12418

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

MISKOLD: 3515 Hunyadi út 15. Tel.: 46-18330

KECSKEMÉT: Művelt Nép Könyvesbolt. 6000 Március 15. u. 3—5. Tel.: 76-28157

Tisztelt Commodore Egyesület!

Több kérdéssel és kéréssel fordulok önökhöz. 15 éves vagyok, s a biológián kívül nagyon érdekel a számítástechnika. Örülök annak, hogy már Magyarországon is egyre többen foglalkoznak vele.

Hogy tollat — azaz printert — ragadtam, annak az az oka, hogy sokat hallottam az Országos Commodore Egyesületről. Rendszeresen veszem a Commodore újságot, s találok is benne magamnak való cikkeket, programokat.

Ezért úgy gondoltam, hogy jó lenne belépni az Egyesületbe. Igaz, hogy a tagdíj nekem meglehetősen sok, ám amit nyújtanak, azért — úgy gondolom — megéri.

Néhány kételyem azért akad.

— A Pluszpáholy tagjainak jár havonta 120 Ft kedvezmény. A 'cetli' szerint ez budapesti üzletekben határidőig váltható be. Mivel vidéki vagyok, elég nagy gonddal és költséggel járna havonta Budapestre utazni. Az is lehet, hogy a keresett áru éppen abban a boltban nincs, de máshol (v. más-kor) van. Ezt a problémát kellene megoldani.

— Nem tudom, hogy a tagság január 1-jétől december 31-ig tart-e, hiszen akkor ebben az évben már csak féléves tagságra érdemes belépni.

Nagyon szeretném, ha levelemre mielőbb válaszolnának, s — ha minden rendben — csekket is küldenének. Szívességüket előre is köszönöm:

Nagy Zoltán Tamás

Nagykanizsa

Rákóczi F.u.45.

HUNGARY 8800

1. Első kérdésével kapcsolatban olvassa el lapunk 3. oldalán a „Mit, hogyan, hol, mikor?” rovatot.

2. Egyesületünk tagjai sorába bármikor beléphet és akkor kezdődik az „időszámítás”.

Tisztelt Szerkesztőség!

A mellékelt lemezen található **VIDEOKATALOGUS** program az alábbi funkciókat látja el (ez CMB 64 BASIC V2-ben íródott): — A program azoknak a felhasználóknak íródott, akik rendelkeznek videomagnóval, és videokazetta-állományukat szeretnék hatékonyan (naprakészen) vezetni, illetve használni.

A program maximum 200 kazetta és kazettánként 30 db film könyvelésére alkalmas, de ez tetszés szerint megváltoztatható a DIM argumentumában.

A program az adatokat lemezen tárolja szekvenciális file-ban. A hibalekérdézés időmegtakarítás céljából elhagytam.

A program a következő adatokat kezeli:

1. Kazettanév (maximum 28 karakter hosszú sztring)
2. Kazettahossz percben (60—300 intervallumon belül)
3. Kazettahossz órában (1,2,3 vagy 4)

— megjegyzés: ez a program egy Grundig vs 540 márkájú képmagnó alapján lett tesztelve. Ezek a videón a szabványhosszak: 60 perc, 120 perc, 180 perc és 240 perc. Létezőnek egyéb hosszak is, a magnó ezt nem tudja felismerni, és más időket használ. Pl: 195 perces kazettát vagy 3 vagy 4 órának tüntet fel, és az eltelt időt (filmhosszakat) ennek ARÁNYÁBAN (fontos!) kezeli.

4. Maradék idő percben (ilyen hosszú vagy ennél rövidebb film elfér még a kazettán)

5. Filmek neve (maximum 28 karakter hosszú sztring)

6. Filmek hossza percben, a videószámláló kiírása alapján (ebből számolja ki a program a kazettahossz percben és órában számok segítségével a film tényleges hosszát—(arányszámítás)).

A program szolgáltatásai:

1. Adatbevitel és javítás (a kazetta sorszámának ismeretében).

2. Adattörlés (a teljes kazettatartalom törlése)

— megjegyzés: adatmentéskor a mentési folyamat véget ér, ha a gép üres tömbelemet talál.

3. Egy kazetta tartalmának kiírása (a sorszám ismeretében).

4. Összes kazetta tartalma (lineáris felsorolás).

5. Maradék idők listája (a kazettaszámok feltüntetésével).

6. Helykeresés filmnek (bekéri a felvenni kívánt film hosszát percben és kiírja az összes olyan kazettát, amire a film ráfér, valamint az utána maradó szabad helyet percben. Ebből a felhasználó már kiválaszthatja a számára legideálisabb kazettát).

7. Adatmentés lemezre

8. Adattöltés lemezről

9. Kilépés a programból

Mintthogy rengeteg videomagnó típus van a háztartásokban, és mivelhogy csak a saját (Grundig vs 540 Monolith) típusunkat tudtam a program alapjául venni, így csak remélni tudom, hogy a program alkalmas más (e módon működő) képmagnóval rendelkező számítógép-tulajdonos munkáját segíteni.

Utolsó megjegyzés: amennyiben az idők feldolgozásában furcsaságot észlelnék, meg kell jegyezni, hogy sokszor a videokazetták nem olyan hosszúak, mint ami rájuk van írva. Ez sok „hiba” forrása lehet.

Remélem, hogy programom elnyerte tetszésüket (és a lapban való közlést), őszinte tisztelettel:

Nagy Róbert

A szerkesztőség megjegyzése: a program ellenőrzése során azt tapasztaltuk, hogy ha a kazetták számozása nem folyamatos (pl. 1-es van, 2-es nincs, 3-astól újra van) a 4-es menüpont hibásan működik. Vállalkozó szellemű olvasóink talán megkeresik és kijavítják a hibát.

```

0 REM *****
1 REM * C= UJSAG          SORSZAM:222
2 REM * C64              VIDEOKATALOGUS
3 REM * PROGRAM : NAGY ROBERT
4 REM *****
10 CLR:DIM K$(200,30)
20 POKE 53280,0:POKE 53281,0:POKE 646,1
30 PRINT CHR$(147);CHR$(142);CHR$(8)
40 PRINT "#####VIDEOFILM-KATALOGUS PROGRAM"
50 PRINT " "
60 PRINT "#####MENU"
80 PRINT"###(##1##) - ADATBEVITEL ES JAVITAS"
90 PRINT"###(##2##) - EGY KAZETTA TORLESE"
100 PRINT"###(##3##) - EGY KAZETTA TARTALMA"
110 PRINT"###(##4##) - OSSZES KAZETTA TARTALMA"
120 PRINT"###(##5##) - MARADEK IDOK LISTAJA"
130 PRINT"###(##6##) - HELYKERESESE FILMNEK"
140 PRINT"###(##7##) - ADATMENTES LEMEZRE"
150 PRINT"###(##8##) - ADATTOLTES LEMEZROL"
160 PRINT"###(##9##) - PROGRAM VEGE"
170 GET R$:IF R#="" THEN 170
180 IF R#<"1" OR R#>"9" THEN 170
190 R=VAL(R#):ON R GOTO 210,1490,600,750,930,1020,1150,1300,1430
210 PRINT"###ADATBEVITEL ES JAVITAS"
220 INPUT"###KAZETTASZAM (1-200):##";SZ
230 IF SZ<1 OR SZ>200 THEN 220
240 SZ#=STR$(SZ):SZ#=LEFT$(SZ#+ " ",3)
250 INPUT"###KAZETTANEV:##";KA#
270 IF KA#="" THEN 30
280 IF LEN(KA#)>28 THEN 260
290 KA#=LEFT$(KA#+ " ",28)
300 INPUT"###KAZETTAHOSSZ PERCBEN (60-300):##";KH
302 IF KH<60 OR KH>300 THEN 300
303 SH=KH
305 INPUT"###VIDEO SZERINT ORABAN:##";VH
306 IF VH<1 AND VH<2 AND VH<3 AND VH<4 AND VH<5 THEN 305
310 VP=VH*60
320 KH#=STR$(KH):KH#=RIGHT$( " "+KH#,3)
330 INPUT"###HANY FILM VAN A KAZETTAN (1-30):##";FS
340 IF FS<1 OR FS>30 THEN 330
350 FOR I=1 TO FS
360 PRINT"###";I;" FILM NEVE:##";INPUT FI#
370 IF FI#="" THEN 360
380 IF LEN(FI#)>28 THEN 360
390 FI#=LEFT$(FI#+ " ",28)
400 INPUT"###HOSSZA PERCBEN (VIDEO ALAPJAN):##";FH
410 IF FH<0 THEN 400
420 FH=INT(((KH/VP)*FH+.5)
430 FH#=STR$(FH):FH#=RIGHT$( " "+FH#,3)
440 MI=SH-FH:SH=SH-FH
    
```



```

O 460 MI#=STR$(MI):MI#=RIGHT$( " "+MI#,3)
470 K$(SZ,I)=FI#+"/"+FH#
480 NEXT I
O 490 K$(SZ,0)=SZ#+"/"+KH#+"/"+KH#+"/"+MI#
500 GOTO 30
O 600 PRINT"KÖRÖK IIRAS KAZETTANKENTIS"
610 INPUT"KÖRÖK KAZETTASZAM (1-200):#";SZ
O 620 IF SZ<1 OR SZ>200 THEN 610
630 PRINT" ";
650 PRINT" ";K$(SZ,0)
O 670 FOR I=1 TO 30
680 PRINT" ";K$(SZ,I)
690 IF K$(SZ,I)=" " THEN 710
700 NEXT I
O 710 PRINT" "SPACE' - MENU"
720 GET R#:IF R#=" " THEN 30
730 GOTO 720
O 750 PRINT" ";SZ=0
760 SZ=SZ+1:IF SZ>200 THEN 890
765 IF K$(SZ,0)=" " THEN 890
O 790 PRINT" ";K$(SZ,0)
820 FOR I=1 TO 30
830 PRINT" ";K$(SZ,I)
840 IF K$(SZ,I)=" " THEN 860
850 NEXT I
O 860 PRINT" "F7' - TOVABB"
870 GET R#:IF R#=" " THEN GOTO 760
880 GOTO 870
O 890 PRINT" "SPACE' - MENU"
900 GET R#:IF R#=" " THEN 30
910 GOTO 900
O 930 PRINT"KÖRÖK MARADEK IDOK LISTAJA";SZ=0
940 SZ=SZ+1:IF SZ>200 THEN 990
945 IF K$(SZ,0)=" " THEN 990
O 950 PRINT" ";SZ;" KAZETTA:#";RIGHT$(K$(SZ,0),3)
970 POKE198,0:WAIT198,255:GOTO 940
980 PRINT" "SPACE' - MENU"
O 990 GET R#:IF R#=" " THEN 30
1000 GOTO 990
O 1020 PRINT"KÖRÖK HELYKERESÉS FILMNEKIS"
1030 INPUT"KÖRÖK FELVEENDO FILM HOSSZA PERCBEN:#";FH:SZ=0
1040 IF FH<0 OR FH>900 THEN 1030
1050 SZ=SZ+1:IF SZ>200 THEN 1120
O 1060 IF K$(SZ,0)=" " THEN 1120
1070 KH=VAL(RIGHT$(K$(SZ,0),3))
1080 IF FH<KH THEN MI=KH-FH:GOSUB 1100
1090 GOTO 1050
O 1100 PRINT" ";SZ;" MARAD:#";MI;" PERC." :POKE198,0:WAIT198,255
1110 RETURN
O 1120 PRINT" "SPACE' - MENU"
1130 GET R#:IF R#=" " THEN 30
1140 GOTO 1130
O 1150 PRINT"KÖRÖK ADATMENTES LEMEZREIS"
1160 PRINT"ADOLGOZOM"
1165 PRINT" " :SZ=0
O 1170 OPEN 8,8,8,"@:VIDEODATAS,S,W"
1180 SZ=SZ+1:PRINT" ";SZ;" ";:IF SZ>200 THEN 1270
1190 IF K$(SZ,0)=" " THEN 1270
O 1200 PRINT#8,K$(SZ,0)
1210 FOR I=1 TO 30:IF I>1 THEN PRINT" ";
O 1220 IF K$(SZ,I)=" " THEN 1250
1230 PRINT#8,K$(SZ,I)
1240 NEXT I
O 1250 PRINT#8,"FV"
1260 GOTO 1180
O 1270 PRINT#8,"KV"
1280 CLOSE#8:PRINTCHR$(13);" (PRESS) " :POKE198,0:WAIT198,255:GOTO 30
O 1300 PRINT"KÖRÖK ADATTOLETES LEMEZROLIS"
1310 PRINT"ADOLGOZOM"
1315 PRINT" " :SZ=0
O 1320 OPEN 8,8,8,"@:VIDEODATAS,S,R"
1330 SZ=SZ+1:PRINT" ";SZ;" ";:IF SZ>200 THEN 1410
1340 INPUT#8,K$(SZ,0)
O 1350 IF K$(SZ,0)="KV" THEN 1410
1360 FOR I=1 TO 30:IF I>1 THEN PRINT" ";
O 1370 INPUT#8,K$(SZ,I)
1380 IF K$(SZ,I)="FV" THEN 1400
1390 NEXT I
O 1400 K$(SZ,I)=" ":GOTO 1330
1410 K$(SZ,0)=" ":CLOSE#8:PRINTCHR$(13);" (PRESS) " :POKE198,0:WAIT198,255:GOTO 30
O 1430 PRINT"KÖRÖK PROGRAM VEGE"
1440 PRINT"KÖRÖK KILEP A PROGRAMBOL (I/N)"
1450 GET R#
O 1460 IFR#="N" THEN 30
1470 IFR#="I" THEN SYS64738
1480 GOTO 1450
O 1490 PRINT"KÖRÖK EGY KAZETTA TORLESEIS"
1500 INPUT"KÖRÖK TORLEENDO KAZETTA SZAMA:#";SZ
O 1510 IF SZ<1 OR SZ>200 THEN 1500
1530 FOR I=0 TO 30
O 1540 K$(SZ,I)=" "
1550 NEXT
O 1560 GOTO 30

```

READY.



Az **AMIGA** lemezei

Befejező rész

A mostani résszel befejeződik az AMIGA lemezei című cikksorozat. A mai alkalommal az utolsónak maradt két blokktypussal (DIRECTORY-blokk, DATA-blokk) fogunk megismerkedni. Ahhoz, hogy lemeziünkön meg tudjuk valósítani a fastruktúrárs tárolási módot, mindenképpen szükségünk van olyan lemezbejegyzésekre, amelyek a szükséges szintek kezelését biztosítják (megkönnyítik). Ezért fejlesztették ki az AmigaDos tervezői a DIRECTORY-blokk (könyvtárblokk) típusot. A DIRECTORY blokk típus szervezése néhány hasonlóságot mutat a ROOT, illetve a FILE-HEADER blokkokkal (azonosak a következő mezők pozíciói és funkciói: Checksum, Hash Table, Protect-Flags, nap-óra-perc-másodperc mezők, névmező, komment mező). A másik aktuális blokk típus, a DATA-blokk. A feladatára igen könnyen rá tudunk jönni, ugyanis ez a blokk típus tartalmazza a file-okat alkotó adatokat 488 byte-os részekben, valamint az adatok adminisztratív jellemzőit (hányadik adatblokkról van szó, hány byte adat tartozik a file-hoz az aktuális DATA-blokkban...stb.). Mindezek után pedig lássuk e két blokk típus részletes szervezését!

A DIRECTORY-blokk felépítése:

1. duplaszó (offset 000) — Type — a blokk típus első duplaszáva — értéke 00000002
2. duplaszó (offset 004) — Header Key — a blokk logikai sorszámát tartalmazza
3. duplaszó (offset 008) — High Sequence — értéke 00000000
4. duplaszó (offset 00c) — Hash Table Size — értéke 00000000
5. duplaszó (offset 010) — Reserved — értéke 00000000
6. duplaszó (offset 014) — Checksum — a DIRECTORY-blokk ellenőrző összege
7. duplaszó (offset 018) — Hash Table Fore Part — a Hash Table első eleme
- . . .
78. duplaszó (offset 134) — Hash Table End — a Hash Table utolsó eleme
79. duplaszó (offset 138) — Reserved — értéke 00000000
80. duplaszó (offset 13c) — Reserved — értéke 00000000
81. duplaszó (offset 140) — Protect Flags — védelmi mező — lásd FILE-HEADER blokk
82. duplaszó (offset 144) — Reserved — értéke 00000000
83. duplaszó (offset 148) — Directory Comment Fore Part — a kommentmező első eleme
- . . .
104. duplaszó (offset 19c) — Directory Comment End — a kommentmező utolsó eleme
105. duplaszó (offset 1a0) — Reserved — értéke 00000000
106. duplaszó (offset 1a4) — Days of Last Modification — az utolsó lemezművelet napja

107. duplaszó (offset 1a8) — Hours/Minutes of Last Modification — az utolsó lemezművelet órája/perce
108. duplaszó (offset 1ac) — Second of Last Modification — az utolsó lemezművelet másodperce
109. duplaszó (offset 1b0) — Directory Name Fore Part — a directory névmezőjének az első eleme
- . . .
115. duplaszó (offset 1c8) — Directory Name End — a directory névmezőjének az utolsó eleme
116. duplaszó (offset 1cc) — Reserved Field Fore Part — a lefoglalt rész első eleme
- . . .
- . . . (00000000 értékkel feltöltve)
124. duplaszó (offset 1ec) — Reserved Field End — a lefoglalt rész utolsó eleme
125. duplaszó (offset 1f0) — Next Hash Pointer — értéke 00000000
126. duplaszó (offset 1f4) — Parent Dir Pointer — a szülő (parent) directory logikai sorszámára mutat
127. duplaszó (offset 1f8) — Extension Block Pointer — értéke 00000000
128. duplaszó (offset 1fc) — Secondary Type — értéke 00000002.

A DATA-blokk felépítése:

1. duplaszó (offset 000) — Type — a blokk típus első duplaszáva — értéke 00000008
2. duplaszó (offset 004) — Header Key — pointer a FILE-HEADER blokkra
3. duplaszó (offset 008) — Sequence Number — az adatblokk sorszámát tartalmazza (a blokkok 1-től vannak számozva, növekvő sorrendben)
4. duplaszó (offset 00c) — Data Size — az értékes adatok száma (hány byte adat tartozik a DATA-blokk Data Field részéből a file-hoz — maximális értéke \$1e8=488)
5. duplaszó (offset 010) — Next Data — pointer a következő DATA-blokkra
6. duplaszó (offset 014) — Checksum — a DATA-blokk ellenőrző összege
7. duplaszó (offset 018) — Data Field Fore Part — az adatmező első eleme
- . . . (itt helyezkednek el a file-t alkotó adatok)
128. duplaszó (offset 1fc) — Data Field End — az adatmező utolsó eleme.

Farkas András

ELINDÍTÓ

Sorozatunkat két újabb alapelem ismertetésével folytatjuk, a ciklusokkal és az elágazásokkal. Az előző részben láthattuk, hogy ha egy programrészt többször, különböző értékekkel akarunk használni, alkalmazhatjuk a szubrutinokat. Ciklust akkor használunk, ha egy utasításcsoportot bizonyos feltétel szerint kell többször (egymás után!) végrehajtani.

A ciklus részei:

Előkészítés: a kezdőértékek megadása,
ciklusmag: azok az utasítások, amiket többször kell végrehajtaniunk,
vizsgálat: a feltétel teljesülésének vizsgálata. Ez esetleg meg is előzheti a ciklusmagot.
 A vizsgálat helyétől függően találkozhatunk a szakirodalomban 'elő(vagy elől) tesztelő és utó(hátul) tesztelő ciklus' elnevezéssel.

Nézzünk néhány példát a ciklus megírására!

Legyen a feladat mondjuk az, hogy 10 számot kell bevinni billentyűzetről egy 100 elemű tömbbe. Az első bevitt szám a tömb 15. eleme legyen, innen folyamatosan töltünk a tömb 24. eleméig. (Természetesen ezek index-számok, azaz a 15-ös számú elem valójában a 16.) A listában található REM utasításokat nem kell beírni!

A program első sora lehet a tömb dimenzionálása:

```
99 REM előkészítés
100 DIM a(99)
110 b=0
119 REM ciklusmag
120 INPUT a(15+b) : REM egy szám bevitele a
    billentyűzetről az 'a' tömb '15+b'-edik elemébe
130 b=b+1
139 REM vizsgálat
140 IF b<10 THEN GOTO 120
150 REM eddig tartott a ciklus
```

Ez egy egyszerű ciklus, ami a ciklusmag végrehajtása után végzi a vizsgálatot. Lehetőségünk van a BASIC ciklusszerező utasításainak a használatára is. Ezek a FOR és a NEXT.

Ezzel az utasításokkal az előző példaprogram így alakul:

```
100 DIM a(99)
110 FOR b=0 TO 9 STEP 1
120 INPUT a(15+b)
130 NEXT b
```

A 110-es sor 'magyarra fordítva': Hajtsd végre a ciklusmagot b=0 kezdőértéktől, míg a b kilenc nem lesz. A 130-as sor a ciklusmag lezárása, a b változó értékének változtatása a STEP szerint. Lehet, hogy ez így most a 'kezdőknek' furcsán hangzik, de érdemes így gondolkodni a ciklusírásnál.

Mit szabad és mit nem?

Szabad a FOR utasítás végéről a STEP-et leahagyni, ha +1 a lépésköz. Szabad a NEXT utasítás után a változót elhagyni, de nem ajánlatos, különösen akkor nem, ha több ciklusunk is van! Nem igazán korrekt viszont a ciklusváltozót úgy megadni, hogy más utasításszó jelenjen meg a listában:

```
FOR I=STOP
```

A ciklusok szabadon egymásba ágyazhatók:

```
10 FOR A=1 TO 10
20 FOR B=2 TO -20 STEP -3
```

```
30 PRINT A,B
```

```
40 NEXT B
```

```
50 NEXT A
```

Ügyeljünk arra, hogy a NEXT-utasításban helyesen adjuk meg a ciklusváltozót! Arra is fordítsunk figyelmet, hogy ne ugorjunk (pl. GOTO-val) a ciklusmagba, és ne ugorjunk ki onnan! Ciklusváltozóként nem használható egész, szöveges (sztring) illetve tömbváltozó! Előfordul, hogy nem tudjuk, a ciklusváltozó mely értékénél lépünk ki a ciklusból. Ebben az esetben a BASIC 3.5-ös, vagy annál fejlettebb gépek tulajdonosai a DO—LOOP ciklust használhatják. Ennek működését nem írom le, a gépkönyvben megtalálható, szeretném azonban megmutatni, hogyan lehet megvalósítani az elvet akár a C 64-esen is. Legyen a feladat az alábbi!

Vizsgáljuk a számokat 50 és 100 között, és írjuk ki az első olyan számot, aminek négyzete 5000-nél nagyobb!

```
100 FOR i=50 TO 100 STEP 1
110 a=i*i
120 IF a>5000 THEN a=i : i=100
130 NEXT i
140 PRINT a
150 END
```

Láthatjuk, hogy a kívánt kilépést azzal érjük el, hogy a ciklusváltozót a ciklus közben átírjuk. VIGYÁZAT! Ez egy veszélyes fegyver, óvatosan bánjunk vele!

ELÁGAZÁSOK:

A programok írása közben gyakran előfordul, hogy egy feltétel szerint a programot több különböző módon folytatjuk. Eddigi példáinkban is használtunk már ilyet, ezt nevezzük elágazásnak. Legegyszerűbb formája az IF utasítással összekapcsolt ugrás:

```
100 INPUT A
110 REM az INT függvény a számok egész részét adja meg
120 IF A=2*INT(A/2) THEN GOTO 150
130 PRINT „ A szám páratlan! ”: END
150 PRINT „ A szám páros! ”: END
```

A példaprogramban egy számról eldöntjük, osztható-e kettővel. Ennél a szerkezetnél, ha a THEN részt közvetlenül GOTO utasítás követi, a GOTO elhagyható, elég a sorszám:

```
120 IF A=2*INT(A/2) THEN 150
```

Az ugró utasítás lehet GOSUB is, ekkor a szubrutin végrehajtása után a gép a GOSUB után következő utasításra tér vissza, függetlenül attól, hogy ez új sorban van-e. Ha több feltétel egyidejű teljesülését kell vizsgálni, akkor használhatjuk a logikai függvényeket.

Az előző program módosítva:

```
120 IF A=2*INT(A/2) AND A=3*INT(A/3) THEN 150
150 PRINT „ A szám osztható hattal! ”
```

A BASIC 3.5-ös gépek ismerik az ELSE utasítást is, amely szintén az IF-es szerkezetben használható.

Ha több irányban is folytatható a program, és ezeket valami módon számozni tudjuk, akkor az ON A GOTO, illetve ON A GOSUB szerkezetet is használhatjuk. Ekkor egy változó értékétől függően kerül sor az ugrásra, felsorolt programsorszámok szerint. Például:

```
10 INPUT „ A hét hányadik napja van ”:A
20 ON A GOTO 100,200,300,400,500,600,700
```

```

30 PRINT „Ennyiedik napja nincs is a hétnek!”
100 PRINT „Akkor ma hétfő van!”
110 END
200 PRINT „Akkor ma kedd van!”
210 END
.
.
700 PRINT „Akkor ma vasárnap van!”
710 END

```



PROGRAMOK

Nyelvi áttekintés

Bár Commodore gépeinket általában BASIC nyelven vagy gépi kódban programozzuk, amelyek a gépbe vannak építve, más nyelvek is rendelkezésünkre állnak. Ha ezeken a nyelveken programozunk, általában a Commodore képernyőeditorját használjuk a program megírására, amely a BASIC-hez hasonlóan sorszámozza a programot, de az adott nyelv utasításkészletét és szabályait használja. A megírt programot aztán más program vagy programok, amelyek a memóriában vagy lemezen vannak, végrehajtható és az adott nyelven tárolható alakra hozzák. A begépelte program forráskódban, az előkészített változat futtatható kódban van.

A LEGNÉPSZERŰBB NYELVEK

Assembly nyelv — Ez a nyelv a gépi nyelv rokona, mert mindkettő ugyanolyan kódolt változatot állít elő. Az assembly nyelvvvel azonban számos olyan lehetőség áll a programozó rendelkezésére, amely a gépi nyelv használatakor hiányzik. A legegyszerűbb gépi nyelv kivételével valamennyi gépi nyelvű programozás valójában assembly nyelven történik egy assembler-program segítségével, ezért sokan a két kifejezést egyenértékűként használják.

BASIC (fordító) — A Commodore beépített BASIC nyelve úgynevezett „értelmező” nyelv. Ez azt jelenti, hogy a programsorokat egyenként alakítja át gépi kódúvá minden futtatáskor. A különböző „fordított” BASIC nyelvek az ismert utasításkészlettel és szabályokkal rendelkeznek, de a program a megírás után további feldolgozásra kerül: lefordítják. Ez a lépés a BASIC-et egy kvázi-gépi-nyelvű programmá alakítja, ez lesz a ténylegesen futtatott program. Amikor a lefordított program fut, a számítógépnek nem kell időt töltenie minden BASIC utasítás értelmezésével. A BASIC fordító gyorsabb, mint az értelmező változat, de a lefordított program nem listázható.

C — A C nyelv a legdivatosabb manapság a szakemberek körében. Hatékony nyelv, amelynek talán legfőbb előnye az, hogy a programok bármely gépen futtathatók. A C nyelvet talán kicsit nehéz megtanulni, de sokan úgy gondolják, hogy érdemes. Az Amiga programozói különösen kedvelik.

COBOL — A COBOL nyelv az üzleti világ kedvelt nyelve, mert kiválóan alkalmas az üzleti életben gyakran ismétlődő feladatok ellátására.

Az utasítás végrehajtásakor kiértékeli a változót. Ha nem egész, veszi az egész részét. Az értéknek 0—255 tartományba kell esnie. Ha van annyiadik sorszám a GOTO vagy a GOSUB után, azt végrehajlja, ha nincs, a következő sorban folytatja a programot. A GO TO alak itt NEM használható! A változó helyett természetesen kifejezés is állhat.

LENGYEL ISTVÁN

COMAL — Ez az érdekes nyelv nagyon vonzza a Commodore rajongóit. Hasonló a Pascalhoz, a C-hez és a Modula-2-höz. Az egyik gépen megírt programok használhatók bármely másikon is, ha nem tartalmaznak kizárólagosan a gépre érvényes kiterjesztéseket.

FORTRAN — A BASIC „unokatestvére”, ahhoz hasonló utasításkészlettel és szabályokkal. A FORTRAN nagy erőssége az, hogy komplikált matematikai számításokat is könnyen tud kezelni. A tudományos munkával foglalkozó szakemberek kedvelt nyelve.

FORTH — A sok változatban megtalálható FORTH azok kedvence, akik szeretik a szokatlan dolgokat. A felhasználó igényei szerint alakítható, a programozó maga definiálhatja saját utasításait.

LOGO — Könnyen megtanulható nyelv, amelyet az elemi iskolákban is érdemes tanítani. Úgynevezett „teknősbéka-grafikája”, amelyben egy teknősbéka mászik és így írja fel a sorokat, igen népszerű a gyerekek körében.

Pascal — Erősen strukturált nyelv, amely azt követeli a programozótól, hogy programját megtervezze annak megírása előtt. A strukturáltság lehetővé teszi, hogy mások is könnyen megértsék a programot. A tanárok ezt a tulajdonságot nagyra értékelik.

Modula-2 — A Pascal nyelv továbbfejlesztése, de könnyebb annál. Lehetővé teszi, hogy a programot részenként (modulokban) írjuk, amelyeket aztán összefűzhetünk.

A GÉPI NYELVEK TANULÁSA

A legtöbb ember egy gépi nyelvet nehezebben tanul meg, mint a BASIC-et. Ennek nem az az oka, hogy a gépi nyelv nehéz, hanem az, hogy a BASIC könnyű. Ez természetes, hiszen a BASIC nyelv azért íródott, hogy könnyen tanulhassunk meg programozni.

A gépi nyelv megtanulásának első lépése a bináris és hexadecimális számrendszer megismerése. A következő lépés a 6502 típusú mikroprocesszor regisztereinek, címzési módjainak és utasításkészletének megtanulása. Ha a 6502 típusú mikroprocesszorról nem áll rendelkezésünkre semmiféle leírás, akkor bármilyen más mikroprocesszorral is megismerkedhetünk, hiszen annak „gondolkodásmódját” kell elsajátítanunk. Ezután következhet a programozás.

RENEW C128

A program C128-ra egy renew, ami a NEW és a RESET után visszahozza a BASIC programot. A BASIC betöltő lefuttatása után a lemezen lesz egy RENEW program, ami SYS 2850-nel indítható

Gyárfás Richárd



```

1 rem"*****"
2 rem"*          Renew C128          *
3 rem"*          *
4 rem"*By:Dark Vision Software an AGLOSOFT*"
5 rem"*****"
6 rem"

19 print"REN"
20 for j=2850 to 2923:readk:cs=cs+k:pokej,k:next
30 if cs<>8443 then print"hiba!":stop
40 bsave"renew sys2850",b0,p2850 to p2924
50 data 160,3,200,177,45,201,0,208
60 data 249,152,105,1,160,0,145,45
70 data 165,46,160,1,145,45,234,165
80 data 45,133,65,165,46,133,66,160
90 data 0,162,0,177,65,201,0,240
100 data 10,200,192,0,208,243,230,66
110 data 76,67,11,232,200,192,0,208
120 data 2,230,66,224,3,208,228,200
130 data 140,16,18,165,66,141,17,18
140 data 96,0

```

CARTRIDGE SIMULATOR C128—C64

A program egy Cartridge-ot szimulál, ami C128-on töltődik be és onnan tér át, C64 üzemmódba. A program előnyét az jelenti, hogy 1571-es Floppy 128-as módban 10-szer gyorsabb, mint 64-es módban. Ezért a program csak 1571-es és 1570-es Floppyn fut gyorsan. 1541-esen normál megszokott sebességgel működik.

PROGRAM ELVE:

Amikor 128-as üzemmódból áttérünk 64-esbe, akkor a gép végrehajtja a C64-es teljes RESET rutinját, ami először \$8004-on keres egy kódot (CBM80), ha ezt nem találja meg, akkor befejezi a RESET-et és a gép alapállapotba kerül. Ha megtalálta akkor \$8000-al indirekt ugrást hajt végre. A

CARTRIDGE SIMULATOR (CT) átírja ezt a vektort \$CO-

00-ra ahol a RESET.MOD nevű program helyezkedik el, ami helyére másolja és elindítja a programot.

Program beléptetése a CT-be: A BASIC listában 80—115 sorokban van tárolva a CS számára a program adatai ezeket az adatokat kell megváltoztatnunk a következőképpen:

DATA A menüben-ben megjelenő név, indítási cím, program neve, kezdőcím, végcím.

Az indítási-, kezdő-, végcímet hexadecimális alakban kell megadni.

A CS 8 C64-es és 1 C128-as programot képes kezelni a C128-as program a SPACE billentyű megnyomására töltődik be. A program menüben megjelenő neve a 270-es sorban, a betöltési rutin pedig 4000-tól helyezkedhet el.

A CT által kezelt maximális program 96 Blokk. A CT-hoz érdemes autostartossá átalakítani a lemezt, erre a célra megfelel a 1571-es TEST/DEMO lemezen található AUTOBOOT MAKER program.

A CT-nek szüksége van egy RESET.MOD nevű programra, amit a kettes listán található program tartalmaz DATA sorokban.

Gyárfás Richárd

```

1 rem"*****"
2 rem"*C= ujsag          sorszam :      *"
3 rem"*          Cartridge Simulator 128-->64      *"
4 rem"*By:Dark Vision Software (DVS)          *"
5 rem"*          an AGLOSOFT USC.          *"
6 rem"*****"
7 rem"

8 goto10
9 fast:a#=n$(wq):bload(a#),b0,p8192:open15,8,15,"u0>m0":close15:scnc1r:slow:go64
10 graphic0:fori=1to8:keyi,right$(str$(i),1):next
20 color4,1:color0,1:print"REN"chr$(14)chr$(11);
30 print"          CARTRIDGE SIMULATOR C-128          "
31 print"┌──────────────────────────────────────────┐"
40 fori=1to19
50 print"┌──────────────────────────────────────────┐"
60 next
70 print"└──────────────────────────────────────────┘"
80 print"          :a#="COMMODORE 128 --> 64":gosu
90 b10000

```





```

71 print"⚡ The Dark Vision Software an AGLOSOFT.⚡"
80 data"gigaload",c800,"gl.ass",c7f0,cf80
85 data"turbonibbler 4.4",0a84,"turbonibbler 4.4",0801,281d
90 data"super-dos",080d,"",0801,0de2
95 data"fast validate",0826,"fast validate",0801,29e4
100 data"superkitcopy",080d,"superkit copy",0801,2d00
105 data"disk d-mon",080d,"disk-dmon 0-40.o",0801,2a53
110 data"abcload II",0d40,"",0801,0d68
115 data"profi-ass",fce2,"profi-ass",9000,a001
160 fori=1to8
170 reada$(i),c$(i),n$(i),s$(i),e$(i)
180 c(i)=dec(c$(i))
190 next:print"████"
200 fori=1to8:ifa$(i)="-"theni=9:goto260
210 printtab(11)"███-right$(str$(i),1)"█ "a$(i)"█"
260 next
270 printtab(9)"█SPACE█-Fast Hack'em"
290 geta$:a=val(a$)
300 if(a<1ora>8)anda$<>" "then290:elseifa$=" "then4000
310 wq=a:gosub20000
320 c=dec("8014")
330 by=dec(c$(wq))
340 gosub30000
350 c=dec("8010")
360 by=dec(s$(wq))
370 gosub30000
380 c=dec("8012")
390 by=dec(e$(wq))-dec(s$(wq))+8192
400 gosub30000
410 goto9
4000 bload"fast"
10000 printtab(20-len(a$)/2)a$:return
20000 restore20010:fori=.to8:reada:poke32768+i,a:next
20010 data68,229,53,164,195,194,205,56,48
20020 poke32768,28:poke32769,192
20030 poke32770,0:poke32771,192
20040 bload"reset.mod"
20050 return
30000 hi=int(by/256)
30010 lo=by-hi*256
30020 pokec,lo
30030 pokec+1,hi
30040 return

```

```

1 rem *****
2 rem"      Cartridge Simulator      *
3 rem"      RESET.bas --> RESET.mod  *
4 rem"      *
5 rem" By:Dark Vision Software an AGLOSOFT *
6 rem *****
7 rem
10 print"█████Reading data : "
20 i=49152
30 bank0
40 do
50 reada:ifa=-1thenexit
60 pokei+u,a
70 u=u+1
75 print"██████████████████"u
80 v=v+a
90 loop
100 ifv=20374thenbegin
110 print"SD.K"
120 print"█████Saving 'RESET.MOD' ...."
130 bsave"@reset.mod",on b0,p49152top49320
140 bend:else begin

```

○ 150 print"?Data error"	○ 1090 data 173,99,192,133,46,169,1,141	○
160 stop	1100 data 2,8,169,0,141,33,208,32	
○ 170 bend	1110 data 51,165,169,1,141,5,128,76	○
999 end	○ 1120 data 100,192,0,255,169,0,133,34	
○ 1000 data 32,188,246,32,225,255,208,17	1130 data 169,32,133,35,160,0,173,16	
1010 data 32,21,253,32,163,253,32,24	○ 1140 data 128,133,36,173,17,128,133,37	○
1020 data 229,169,1,141,33,208,108,2	1150 data 177,34,145,36,230,34,208,2	
○ 1030 data 160,76,114,254,165,45,141,98	1160 data 230,35,230,36,208,2,230,37	○
1040 data 192,165,46,141,99,192,162,5	○ 1170 data 165,35,205,19,128,208,233,173	
○ 1050 data 142,22,208,32,163,253,32,80	1180 data 18,128,197,34,208,226,169,192	○
1060 data 253,32,21,253,32,91,255,32	○ 1190 data 72,169,160,72,108,20,128,234	
1070 data 83,228,32,191,227,32,34,228	1200 data 234,76,134,227,64,224,224,224	○
○ 1080 data 162,251,154,173,98,192,133,45	1210 data -1	○

KERESZTREJTVÉNY

VÍZSZINTES: 1. Adorján Noémi írta e könyvet a mikroszámítógép-felhasználók körében. 12. A héliumcsillagok közé tartozik. 13. Szlovákiai helység. 14. Tetejére. 16. Foghús. 18. Tölt. 19. Folyadék. 20. Komárom megyei helység. 22. Sereg. 24. Titokban figyel. 25. Van ilyen mézes is. 27. Zala megyei helység. 28. Francia cég. 29. Kosár. 31. Kacat. 32. Minőségellenőr röviden. 34. Ez elszáll de az írás megmarad. 36. ... Jóska, a Talpatnyi föld főhőse volt. 38. Microsoft Word 3,0 könyv szerzőjének neve. 41. Van ilyen mag is. 42. Nem utolsó. 45. Dátumrag. 47. Ékes növény. 50. Egy bizonyos műveletet végez. 53. Kérdőszó. 55. Nagyon fázik. 58. Nóta. 59. Madárfajta. 61. -ből párja. 62. Szövő állat. 63. E nap. 64. Idegen kötőszó. 66. Fogoly. 68. Rédei László névjele. 69. Bortermő vidék. 71. Névelős testrész. 73. Délelőtt röviden.

FÜGGŐLEGES: 1. Nem vasból készült. 2. Vés. 3. Idegen három. 4. Szór. 5. ... Árpád (1906—1987) Kossuth-díjas vajúró volt. 6. Magam. 7. Kötőszó. 8. Szálakat egyesít. 9. Ez a lelke mindennek. 10. Azonos a 18. vízszintessel. 11. Elcé(!). 15. Azonos a 20. vízszintessel. 17. Női név. 19. Ritka férfinév. 21. Kevés ellentéte. 22. Víziállat. 23. Székesegyház. 24. Kávé keverék. 26. Szlovákiai nagyváros. 28. E név jelentése latinul: vágódó, sóvárgó. 30. Porszerűen esik a hó. 32. Volt ilyen ágyú is. 33. Halkan mond. 35. Titokban megy. 36. Ilyen áramlat is van. 37. Város az NSZK-ban. 39. Zrínyi Miklós névjele. 40. Irányszó. 43. Juttat. 44. Közél-keleti állam. 46. Radó Sándor álneve volt a II. világháború idején. 48. Szerelem kis nyílása. 49. Lángol. 51. Ma már ritka férfinév. 52. Hamis. 53. Hélium vegyjele. 54. Azonos a 19. vízszintessel. 56. Kutya. 57. Megvehető. 59. Becézett leánynév. 60. A telefon feltalálója. 62. Orbán Katalin írta

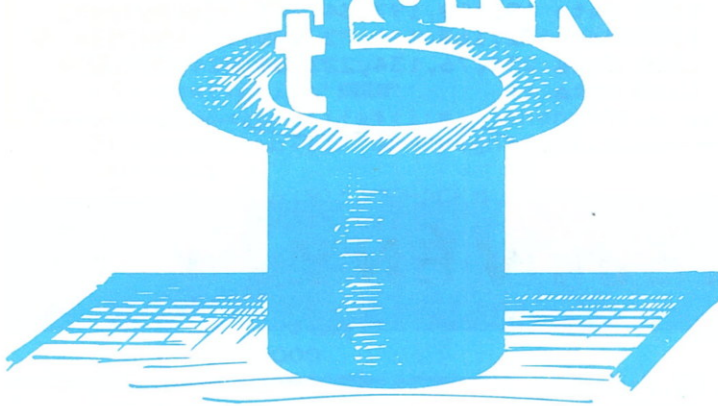
e könyvet. 63. Tisztít. 65. Fejér megyei helység. 67. Kórházi osztály. 69. Kétes(!). 70. Állóvíz. 71. Névelők. 72. Csak félig bors.

A rejtvényt nem kell beküldeni, mert szórakoztató jellegű!

Mokos István

1		2	3	4	5	6		7	8	9	10	11		↘
	○	12					○	13					○	
14	15	○	16			○	17	○	18			○	19	
20		21	○		○	22		23	○		○	24		
25			26	○	27					○	28			
	○	29		30	○	31			○	32			○	
○	33	○	34		35	○		○	36			○	37	○
38						39		40						
○		○	41				○	42				○		○
43	○	44	○	45		○	46	○	47		○	48	○	49
50	51		52		○	53		54	○	55	56		57	
○	58			○	59				60	○	61			○
62			○	63		○		○	64	65	○	66		67
68		○	69			70	○	71			72	○	73	
↙														

Tippek Trükkök



A VDC A GRAFIKAPROFI

A „VDC programozás” témában van néhány új hírünk. Például hogy elérhető a 488 grafikus soros felbontás is. Erre még az Amiga sem képes!

Hogy ez hogy lehetséges? Ehhez előbb meg kell állapítani, milyen VDC is van a gépünkben. Ehhez nem kell fölnyitni a gépet, elegendő az alábbi utasítássort is:

```
BANK 15:POKE 54784,25:PRINT PEEK (54785)
```

A képernyőn ezután vagy a 61-es, vagy a 71-es szám jelenik meg. A következő parancsok okán meglátjuk, miért is oly fontos ez a lekérdezés:

```
BANK 15
```

```
POKE 54784, 1:POKE 54785,50
```

```
POKE 54784, 2:POKE 54785,61
```

```
POKE 54784, 0:POKE 54785,71
```

```
POKE 54784,27:POKE 54785,50
```

```
POKE 54784,25:POKE 54785,80
```

Némely VDC esetében itt a kereten remegést, villódzást tapasztalunk. Ha ez nem így volna, akkor egy tökéletes 40 karakteres képernyő áll előttünk. Különbösen az alábbi parancsot adjuk be: POKE 54784,25:POKE 54785,80+6

A keret villódzása leáll. Ha az első öt POKE esetén ezzel a jelenséggel nem találkozunk, most ennek a parancsnak a beadásakor (ott) az előkerül. Erre az ismeretre azért van szükség, hogy a remegést minden VDC chip esetében kézben tudjuk tartani. Adott esetben változtassuk meg a VDC 25-ös regiszterét azzal, hogy kiolvassuk és újra fölülírjuk:

```
POKE 54784,25:X=PEEK(54785)
```

```
POKE 54784,25:POKE 54785,X OR 6
```

Első pillantásra nem sok értelmét láthatjuk annak hogy megállapítjuk, milyen videochip is van a gépben. Pedig ha grafikus témára váltunk, és egy képet be akarunk kapcsolni, vagy mást csinálunk, úgy a beépített chip minden különlegesét jól kell ismernünk.

A mi célunk nem az, hogy a képernyőt 40 karakter/sor méretre csökkentjük, hanem hogy azt 80 karakter x 50 sorra nagyítsuk. Ehhez a VDC-t az interlace üzemmódba kell kapcsolni:

```
POKE 54784,8:POKE 54785,240
```

Most a sorok össz-számát 98-ra növeljük. Hogy az interlace mód által kiváltott remegést elkerüljük, írjuk át a 36-os regiszter értékét: POKE 54784,36:POKE 54785,240

— a C 128-ashoz

Az összenyomott képernyőt az alsó szélén most az alábbi utasítással toljuk el: POKE 54784,7:POKE 54785,76

Az attribút tárolót a VDC RAM-ban állítsuk 2048-ról 4096-ra:

```
POKE 2607,16
```

```
POKE 54784,20:POKE 54785,16
```

Akár ötven sort is használhatunk, ha a 6-os regiszterbe írunk:

```
POKE 54784,6:POKE 54785,50
```

Most egy program segítségével gondoskodni kell arról, hogy a képernyő eme részébe kerüljön is valami. Ezt a legegyszerűbben úgy érhetjük el, hogy fölírjuk a felső részt, majd azt átmásoljuk az alsóba. A parancsok beadására most a 40 karakteres képernyőt használjuk, mivel különben a C128-as operációs rendszerre a nagy fáradsággal beállított VDC regisztereket fölülírná. Tehát a 40 karakteres képernyőt használjuk:

```
POKE 54784,24:X=PEEK(54785)
```

```
POKE 54784,24:POKE 54785,X OR 128
```

Ezzel bekapcsoltuk a COPY jelzőt. Most csak a kezdő- és végcímek megadni:

```
QU=0:REM startsor
```

```
POKE 54784,18:POKE 54785,QU/256
```

```
POKE 54784,19:POKE 54785,QU AND 255
```

```
ZI=2000:REM célcím
```

```
POKE 54784,32:POKE 54785,QU/256
```

```
POKE 54784,33:POKE 54785,QU AND 255
```

Most már csak az indítólövés hiányzik a másoláshoz. Nos ez akkor történik meg, ha a 30-as regiszterbe írunk valamit. Ez adja meg, hány byte-ot akarunk eltolni a VDC tárolójában. Ha egy sorról van szó, a 80-ast kell odaírni:

```
POKE 54784,30:POKE 54785,80
```

Ezzel egy szövegsort másoltunk át. Mielőtt ezután bármi-vel is hivatkoznánk a VDC-re, ki kell kapcsolni a COPY bitet:

```
POKE 54784,24:POKE 54785,X
```

Az attribútok (színek, villogás stb.) viszont még nem stimelnek. Azt a tárolót is hasonló parancsokkal kell kimásolni:

```
POKE 54784,24:X=PEEK(54785)
```

```
POKE 54784,24:POKE 54785,X OR 128
```

```
QU=4096:REM startsor
```

```
POKE 54784,18:POKE 54785,QU/256
```

```
POKE 54784,19:POKE 54785,QU AND 255
```

```
ZI=6096:REM célcím
```

```
POKE 54784,32:POKE 54785,QU/256
```

```
POKE 54784,33:POKE 54785,QU AND 255
```

```
POKE 54784,30:POKE 54785,80
```

```
POKE 54784,24:POKE 54785,X
```

Arra figyeljünk, hogy csak a másolandó tartomány címei változnak. Most csak egy sort másoltunk. A képernyőnek azonban 25 sora van. A rutint tehát 25-ször kell alkalmazni. Erre a célra a legjobb egy program.

NÉGY JELKÉSZLET EGYSZERRE

A C128-as 80 karakteres képernyőjét kezelő video chip, a VDC lehetővé teszi a számunkra, hogy a két karakterkészletet (kisbetű/nagybetű és nagybetű/grafikus karakterek) egyszerre

jelenítsük meg a képernyőn. A dupla karakterkészlet program lehetővé teszi, hogy egyszerre négy jelkészletet szerepeltessünk a képernyőn. Ez azért lehetséges, mert a VDC nemcsak a szín, aláhúzás, villogtatás attribútókat engedi meg, hanem az inverz kijelzést is. Ebben az esetben azonban nincsen szükség a meglévő két karakterkészlet inverz jelkészlet részére! A dupla karakterkészlet kihasználja ezt az attribútót, és így helyet csinál egy harmadik és egy negyedik jelkészletnek. Adjuk be a programunkat. A használatnál az inverz karakterekre váltást ne a <CTRL> <0>-val végezzük, hanem a PRINT CHR\$(11)-gyel. A kikapcsolás a CHR\$(12)-vel történhet. A <CTRL> <0> és a <CTRL> <9> billentyűkkel pedig az új jelkészletre válthatunk. A programot a SYS 3072 utasítással lehet aktiválni és a SYS 3083-mal kikapcsolni. Hogy egy jelkészletet a VDC-be tölthessük, egy kis trükköt kell alkalmazni.

A használni kívánt karakterkészletet az alábbi utasítással hívjuk be a C128-as grafikus tárolójába:

```
BLOAD „név”,ON B0,P12288
```

Az előfordulhat, hogy egy hosszabb BASIC-program igénybe veszi ezt a helyet. Akkor a betöltés előtt adjuk be az alábbi utasításokat:

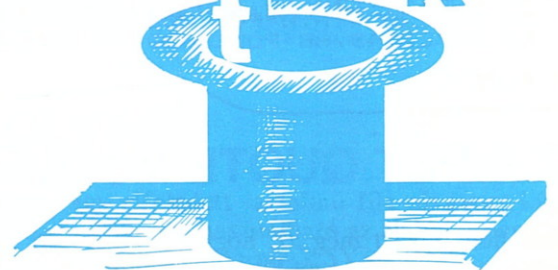
```
GRAPHIC 1,1:GRAPHIC 5
```

Ezzel eltoljuk a BASIC-terület kezdetét jelző mutatót, és így levédjük a területünket. A SYS 3143-mal a kisbetű/nagybetű karakterkészletet másolhatjuk be a videochip RAM-jába. A nagybetű/grafikus jeleket pedig a SYS 3146-tal. Az ehavi szám programszervíz lemezén további jelkészleteket találunk, de felhasználhatunk bármely sajtót, akár a 64-eshez kifejlesztetteket is.

```

1 rem *****
2 rem"* C= Ujsag          Szsz.: *
3 rem"*  DOUBLE-CHARSET.BAS C-128 *
4 rem"* Program:        Szabo Istvan *
5 rem *****
6 :
7 :
8 print chr$(147)"data beolvasas es ellenorzes ...";:j=03072:ve=03223:p=j
9 for b=0 to 7:read a$
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 a9,11,a0,0c,8d,34,03,8c,0694
21 data 35,03,60,a9,b9,a0,c7,d0,1073
22 data f3,24,d7,30,03,4c,b9,c7,1005
23 data c9,1b,f0,ee,a6,f5,f0,03,1360
24 data 4c,c9,c7,c9,14,d0,03,4c,0984
25 data d0,c7,a6,f4,d0,f2,c9,0b,1479
26 data d0,07,a5,f1,09,40,85,f1,1068
27 data 60,c9,0c,f0,03,4c,d0,c7,1035
28 data a5,f1,29,bf,85,f1,60,a9,1277
29 data 28,2c,a9,38,a2,12,20,cc,0725
30 data cd,a9,00,e8,20,cc,cd,ad,1220
31 data 90,0c,85,fd,ad,91,0c,85,1005
32 data fe,a0,00,a2,01,a9,fd,20,1031
33 data 74,ff,20,ca,cd,c8,c0,08,1210
34 data 90,f1,a9,00,20,ca,cd,88,1129
35 data d0,fa,18,a5,fd,69,08,85,1146
36 data fd,90,e0,e6,fe,18,ad,91,1447
37 data 0c,69,04,c5,fe,90,d4,60,1024
38 data 00,30,20,d0,8c,40,00,00,0492
ready.
```

Tippek
trükkök



— a C 64-eshez

REM-RUTIN

Sokszor problémát okoz, hogy egy BASIC-programhoz kapcsolódó rövidebb gépi kódú rutint hol is helyezünk el.

Többnyire két megoldás kínálkozik: vagy a BASIC-programban, DATA-sorokba írva — elég sok helyet foglal, a beol-

vasása pedig nagyon időigényes —, vagy egy külön file-ban, amit a programból hívunk be.

Am ennél sokkal praktikusabb azt magában a programban, a 0 BASIC-sorban egy REM után írva tárolni! Az így elhelyezett gépi kódú szubrutint szokás REM-rutinnak nevezni.

```

1 REM *****
2 REM * C= UJSAG SORSZAM: *
3 REM * REM-RUTIN KESZITO *
4 REM * C=64 *
5 REM * (C) SUKWARE 1990.02.06. *
6 REM *****
9 ...
10 DEF FN A(X)=ASC(MID$(A$,X,1))-48+7*(MID$(A$,X,1)>"@")
20 KC=53000:FOR I=KC TO KC+98:READ A$
30 A=16*FN A(1)+FN A(2):POKE I,A:S=S+A
40 NEXT I:IF S>9831 THEN PRINT "HIBA!"
50 END
99 ...
100 DATA 20,73,00,20,8A,AD,20,F7,B7,84
110 DATA F9,85,FA,20,73,00,20,8A,AD,20
120 DATA F7,B7,38,A5,14,E5,F9,85,02,C9
130 DATA F9,00,41,E6,02,A0,00,B1,F9,99
140 DATA 06,08,C8,C4,02,98,F6,A9,00,99
150 DATA 06,08,99,07,08,99,08,08,0D,03
160 DATA 08,0D,04,08,A9,0F,0D,05,08,A2
170 DATA 08,18,A9,07,65,02,90,01,E8,0D
180 DATA 01,08,0E,02,08,18,69,02,90,01
190 DATA E8,85,2D,06,2E,20,59,A6,60

READY.
    
```

Hívása pedig SYS 2054-gyel lehetséges majd. Van azonban néhány feltétel, korlátozás, amit figyelembe kell venni a REM-rutin alkalmazásánál:

- A tárolni kívánt gépi kódú program:
 - max.249 byte-ből állhat
 - nem tartalmazhat önmagára hivatkozó JMP vagy JSR utasítást (illetve át kell írni azokat a megváltozott kezdőcímhez /2054/viszonyítva)

- nem tartalmazhat 0 értékű kódot /pl. LDA #0/ mert a 0 a listában a BASIC sor végét jelzi, s ez elrontaná a rutint.

Előfordulhat, hogy listázásnál az interpreter egy 204-es kóddal találkozik, aminek az értelmezésével nem képes megbirkózni /poloska/, ekkor „SYNTAX ERROR”-t jelez és a listázás leáll. A további programsorok előcsalogatásában ilyenkor a LISTA utasítás segít.

A REM-rutin elkészítő program hívása /futtatás után/:SYS 53000, KC, VC ahol KC és VC az áthelyezendő gépi kódú program kezdő- és végcíme. A két cím közti terület lesz elhelyezve a 0 sorba, egy REM után.

Listázáskor furcsa szöveg jelenik meg a 0 sorban, akár több soron át, /persze az is megeshet, hogy éppen semmi sem látszik, vagy csak a kurzor színe változik meg../, de ez így természetes!

Első próbálkozásnak — tesztelésnek ajánlom a következő parancssor kiadását:

```
FOR 17= 49152 TO 49400:POKE I, 140:NEXT:SYS 53000, 49152, 49400 'RETURN'/LIST
```

Ekkor mintegy 2 képernyőre elegendő „RESTORE” felirat képezi a 0. sort! /a RESTORE kódja= 140/

De hasznos pl. REM-rutinná átalakítani és így elhelyezni BASIC-programban a következő programot: a „FILE-OLVASÓT!”

Egyébként a REM-rutin készítő a „K6” változó átírásával bármely más tárterületre is betölthető!

Ismertetem a REM-rutin készítő BASIC változatát is, tökéletesen működik ez is. Az első két sor /0 és 1/ feltétlenül szükséges, enélkül a program nem működik jól! /A REM után bármilyen karakter állhat, de a sor összesen legyen 80 karakter hosszú/.

Szűcs László

```

0 REM *****
1 REM *****
10 INPUT "A GEPI KODU RUTIN KEZDOCIME:";KC
20 INPUT "M VEGCIME:";VC
30 IF VC-KC>249 THEN RUN
40 FOR I=KC TO VC:POKE 2054+N,PEEK(I):N=N+1:NEXT
50 POKE2054+N,0:POKE2055+N,0:POKE2056+N,0:A=2055+N:FB=INT(A/256):AB=A-FB*256
60 POKE2049,AB:POKE2050,FB:POKE2051,0:POKE2052,0:POKE2053,143

READY.
    
```

AZ INPUT QUOTE MÓDUSA

Kit nem zavart még az, hogy az INPUT-os bevitelnél a képernyős maszkokat szét lehet rombolni, ha véletlenül megnyomjuk a kurzorvezérlő gombok valamelyikét. Az értelmes bevitelnek ezzel „vége”. Mindezt azonban elkerülhetjük azaz, ha a bevétel alatt bekapcsoljuk az idézőjel üzemmódot (quote módus).

Az egyik lehetőség abban áll, hogy a billentyűzet pufferba írunk egy idézőjelet, majd azt a CHR<20>-szal letöröljük. Így a trükk láthatatlan lesz. De ezt az üzemmódot aktiválhatjuk a POKE 212,1 utasítással is. Az eddigi INPUT „név”;A\$ helyett tehát az alábbi sort használjuk:

```
10 POKE 212,1:POKE 19,1:INPUT „név”;A$:POKE 19,0
```

A plusz POKE utasításokra (19,1 és 19,0) azért van szükség, mert különben a DEL vezérlőkód megjelenne az INPUT kérdőjele után. Kikapcsoljuk a kérdőjelet, így a zavaró vezérlő jel is eltűnik. Ennek a rutinnak az előnye még, hogy üres bevitel nem lehetségesek. Ez tehát egy ilyen irányú kis biztonsági rutint is helyettesít.

A rendszer INPUT rutinját a gépi kódú programozók is gyakran használják. Azt a \$FFGF-nél lehet aktiválni. Itt is szétrombolhatja azonban a kurzormozgatás a beviteli maszkot. Itt is a 212-es cellát kell segítségül hívni.

Induljunk ki abból, hogy szeretnénk Assemblerben egy öt-karakteres szöveget beolvasni és a \$C000-tól lerakni:

```
LDY #$00
LOOP JSR $FFCF
STA $C000,Y
```

```

    INY
    CPY #\$04
    BNE LOOP
    
```

Ha szeretnénk a quote módust használni, az alábbi sorokat kell még beszúrunk:

```

    LDY #\$00
    LDA #\$01
    STA \$D4
    STA \$13
    LOOP JSR \$FFCF
    STA \$C000,Y
    INY
    CPY #\$04
    BNE LOOP
    LDA #\$00
    STA \$13
    
```

Ha ezt az új rutint hívjuk föl, a számítógép egyetlen vezérlő funkciót sem hajt végre, hanem a megfelelő kódokat írja ki a képernyőre. Ezeket azután mint szöveget át is vesszük. Ha tehát a meghívás után ötször megnyomjuk a <CTRL/HOME> gombot, nem töröljük a képernyőt, hanem ellenkezőleg, át vesszük a 147-es kódot ötször, s azt a 49152-től letároljuk.

A VÁLTOZTATHATÓ BETÖLTÉSI CÍM

Egy PRG állomány, amely gépi kódú programot tartalmaz, rendelkezik egy meghatározott betöltési címmel. Néhányat ezek közül LOAD, „név”, 8,1-gyel hívhatunk be, majd SYS utasítással indíthatunk. Mód van azonban arra is, hogy ezeket a programokat a BASIC területre töltsük, s egy BASIC sort írjunk oda. Ebben az esetben a betöltés LOAD, „név”, 8-cal is megtörténhet, s az indításra a RUN-t használhatjuk. A program ekkor egy SYS sorból áll, amely fölhívja a szükséges eltolótutint, amely a szóban forgó programot a megadott címre másolja, majd ott elindítja. A második módszer alkalmazása sokkal egyszerűbb. Ennek azonban az a hátránya, hogy egy, a tárolóban álló BASIC-program elvész (márpedig sokszor töltnünk be BASIC-ből gépi kódú programokat). Ha a két módszer-kombinációt használnánk egyszerre, az volna a megoldás. Példa: behívhatunk egy gyorstöltőt parancsmódban, 8-cal és elindíthatjuk RUN-nal. Az erre beilleszti magát az operációs rendszerbe. Ezt követően mondjuk egy hosszú játékot sokkal gyorsabban tölthetünk be parancsmódban. Ugyanakkor az adott gyorstöltőt (mint alprogramot) utántölthetnénk is, majd egy ugrási paranccsal aktiválhatnánk, hogy további programokat tölthessünk még be. Az ilyen kombiállományok készítése egy trükkel igen egyszerű.

Bemutatásra vegyünk egy gépi kódú programot, melynek indítása SYS 50000-rel (\$C350) történik. Ehhez programkezdetnek a \$C000 címet vesszük, s oda egy JMP \$C350 parancsot írunk (\$4C, \$50 \$C0 byte-ok). Most a programot SYS 49152-vel is indítani lehet. A végcímnek a \$C471-et választjuk (199 byte hossz). Ha most a \$C000-\$C417 területet kimentjük, majd azt abszolút módon betöltjük, a programot SYS 49152-vel is elindíthatjuk. Ha viszont mint BASIC-programot hívjuk be, a rendszer biztosan lemerevedik.

A tulajdonképpeni gép kódú program a \$C350-\$C417 címen áll, egy 3 byte-nyi JMP utasítást pedig a \$C000-ra írtunk. Ha ezt a programot relatív módon töltjük be (a betöltési cím akkor \$0801 lesz), úgy a JMP utasítás a \$0801, \$0802, és a \$0803-as cellákban áll. Az első és a második címre az a cím-mutató kerül, amelyet az operációs rendszer a betöltéskor automatikusan korrigál. A \$4C és a \$50-es byte tehát nem zavar,

mert azokat a föntiek szerint az interpreter egy használható linkpointerre váltja.

A JMP utasításból így már csak egy byte marad a \$0803-ban. Ez a sorszám low byte-ja, amely a továbbiakban semmi szerepet nem játszik. Állítsuk a sorszám high byte-ját nullára. Ezt POKE-oljuk be tehát a \$C003-as címre. A relatív töltéskor tehát ez a \$0804-re kerül. Ezután egy BASIC-sort kell konstruálni. Ehhez a \$0805 címet vesszük elő. A SYS utasítás tokenje \$9E, vagyis POKE 49156,158 kell nekünk. A 2061-es ASCII kódot a \$C005-\$C008 területre rakjuk. Végül 3 nullabyte jelzi a program végét: POKE 49161,0:POKE49162,0:POKE49163,0

Ezt a három nullát a BASIC interpreter igényli, mert ez alapján ismeri föl a BASIC-programok végét. Ha ez nincs, LIST esetén a teljes tárolótartalmat kezdenék kijelezni egészen addig, amíg három nullára nem bukkanunk, s ez rosszabb esetben lemerevedést is okozhat.

Ha ezek után a gépi kódú programot relatív módon hívjuk be, a LIST után máris megjelenik a SYS utasítás. Ami még hiányzik, az egy másolórutin, ami a \$080D-nél kezdődik. Írjunk tehát egy rövid Assembler-programot a \$C00C-től (ez lesz a \$080D), amely SYS 2061-gyel indítható, és a szükséges területet a \$C350-re másolja. Mint tudjuk, a példaprogramunk 199 byte-ja abszolút töltés esetén ezen a helyen áll. Átszámolva a címeket, az abszolút töltés esetén a \$0B51-\$0C18 terület felel meg a \$C350-\$C471 eredetinek, így azt kell átmásolni oda. A másolórutint pedig egy JMP \$C350-nek kell követnie.

Az így „preparált” \$C000-C417 területet mentsük lemezre. Sajnos kombi-állományt kazettával nem használhatunk, mivel ott mindig azt a másodlagos címet kell használnunk, amivel a kimentés történt.

Ha az elkészített kombiállományt abszolút módon hívjuk be, az a \$C000-nál kezdődik. SYS 49152-re az ott álló JMP \$C350 a kívánt (eredeti) rutint máris aktiválja. Ha viszont a ,8-as töltést választjuk, a JMP utasítást a \$080B-re irányuló linkpointerré alakítja az operációs rendszer, a \$C1 (low) és \$00 (high) byte-értékek a 193-as BASIC sort „szimulálják”, s ezt egy „használható” SYS 2061 BASIC parancs követi. A RUN-nal való indítás után az interpreter elindítja a 2061-es címen kezdődő másolórutint, ami az eredeti rutinunkat a \$C350-es címre írja. A másolórutint követő JMP \$C350 pedig máris aktiválja ezt — a kívánságunk szerint!

Kombi-állományokkal más alkalmazásokat is megvalósíthatunk. Jó példa lehet erre a floppystátusz-lekérdezés, amelyet egy programból (is) utántölthetünk, ha floppyműveletek vannak. De ha csak néhány adatot akarunk — a számítógép parancsmódjában — megváltoztatni a lemezen, elég egy sima ,8-as töltés és a RUN parancs az aktiváláshoz anélkül, hogy a betöltési cím kereséssel kellene vacakolni.

FILE-OLVASÓ

Ismeretlen vagy elfeledett tartalmú file-ok gyors képernyőre listázására kitűnően használható az alábbi program:

Indítása: OPEN 2,8,2, „File-név”: SYS 49152

A kiírás a „CTRL” gombbal lassítható, bármely más billentyű lenyomásával pedig megállítható (a billentyűt fölengevedve a kiírás folytatódik).

Az OPEN 15, 8, 15: SYS 49152 kiadásával a floppy hibacsatornája olvasható és az esetleges hibaüzenet megjelenik a képernyőn.

A 'KC' /kezdőcím/ változó átírásával a program bármely más szabad tárterületre is betölthető! /Hívása ekkor értelem-szerűen 'SYS KC' lesz.../

Szűcs László

```

1 REM *****
2 REM * C= UJSAG   SORSZAM:   *
3 REM *           FILE-OLVASO *
4 REM *           C=64        *
5 REM * (C) SUKWARE 1990.02.06. *
6 REM *****
9 ...
10 KC=49152
20 FOR I=KC TO KC+33:READ A:POKE I,A
30 S=S+A:NEXT
40 IF S<>5254 THEN PRINT "HIBA!"
50 END
99 ...
100 DATA 166,184, 32,198,255, 32,207
110 DATA 255,164,144,192, 64,240, 11
120 DATA 32, 22,231,165,203, 41, 64
130 DATA 208,238,240,248,165,184, 32
140 DATA 195,255, 32,204,255, 96

READY.
    
```

IF... THEN NÉLKÜL

Sokszor van szükség arra, hogy a BASIC programokban egy (cél) változó értékét egy másik (teszt) változó értékének függvényében alakítsuk. Ezt általában az IF-TEHEN lekérdezések végtelen sorával szokták megoldani, valahogy így:

```

IF A<6 THEN B=9
IF A=6 THEN B=1
IF A>6 THEN B=-4
    
```

Ebben a példában az A a teszt-, a B a célváltozó. A C 64-es BASIC azonban lehetővé teszi az ilyen lekérdezések egyetlen sorban történő összefoglalását. A varázsszó a „logikai kapcsolat”. Ennek segítségével például a fenti három összevetést is egy sorba pakolhatjuk.

Legelőször hozzá kell rendelni a célváltozóhoz a lehető legmagasabb értéket (itt B=9), majd ebből logikai összehasonlítások alapján levonjuk a 8-at (ha A=6), hogy így a B=1-hez jussunk. 13-at kell levonni, ha A>6 s ekkor a B=4 lesz a kívánt eredmény. A komplett sor így néz ki:

$$B=9+(A=6)\times 8+(A>6)\times 13$$

A definíciók szerint a két, zárójelben álló kifejezés értéke akkor és csak akkor lesz mínusz egy, ha az ott rögzített feltétel teljesül, azaz a kifejezés igaz. Ellenkező esetben a kifejezés a nulla értéket adja. Ha például :

A=6:

$$B=9+(-1)\times 8+(0)\times 13=1$$

Ha A>6 akkor:

$$B=9+(0)\times 8+(-1)\times 13=-4$$

Amennyiben a két kifejezés egyike sem teljesül, ezt kapjuk: $B=9+(0)\times 8+(0)\times 13=9$

Mindez csak egy a lehetséges programrövidítő próbálkozások közül. Kísérletezzünk nyugodtan tovább!

A 2000-es sorban kerekítjük a W változó értékét két helyre a tizedespont után, majd azt sztringgé alakítjuk. A 2005. sor adja ennek a sztringnek a hosszát. A 2010–2020. hurokban ellenőrizzük, szerepel-e ott az „E”, a kitevő jelzője. Ha igen, nem tudunk tovább dolgozni, mert a rutin ezeknek a számoknak a használatára nincs kiképezve.

A 2030–2050. hurokban a tizedespontot keressük. Ha nincs ilyen, pótoljuk azt és még két nullát illesztünk a számhoz. Ha a tizedespont az utolsóelőtti helyen van (mondjuk 32.5), akkor egy nullával toldjuk csak meg. A 2090. sor az esetleg fölösleges szeptet vágja le.

Ha a szám kisebb, mint 1, nulla egészszel pótoljuk ki. Végetül a 2110-es sor hat szóközzel bővíti ki a sztringet, de csak jobbról számolva nyolc jelet használ föl az eredményből, így számunk pont ennyi karakteres lesz legalább egy egészszel és pont két értékkel a tizedespont után.

Ha a negatív értékeket akarunk kezelni ugyanígy, a kerekítésnél nem hozzáadni, hanem kivonni kell a számból 0,005-öt.

```

2000 W$=STR$(W+0.005):L=LEN(W$)
2010 FOR I=1 TO L
2015 IF MID$(W$,I,1)="E" THEN STOP
2020 NEXT I
2030 FOR I=1 TO L
2040 IF MID$(W$,I,1)="." THEN 2070
2050 NEXT I
2060 W$=W$+"."00":GOTO 2110
2070 IF I=L-1 THEN W$=W$+"0":GOTO 2100
2090 W$=MID$(W$,1,I+2)
2100 IF I=2 THEN W$="0"+RIGHT$(W$,3)
2110 W$=" "+W$:W$=RIGHT$(W$,8)
2120 RETURN
    
```

ÉRTELMETLEN SZÁMÍTÁS

Általában úgy tudjuk, hogy a számítógépek tudnak számolni. Ezért van a számítógépben a „számít” szó. Hogy ez nem mindig van így, azt az alábbi példa mutatja:

```

5X=256-3E-8
10 FOR A=0 TO X:PRINT A:NEXT
    
```

Ha ezt a programot elindítjuk, a gép szépen sorban kiadja az általunk „várt” értékeket. Ha viszont spórolni akarunk a tárolóval és a sorszámokkal, akkor így adjuk be a programot: `10 FOR A=0 TO 256 -3E-8:PRINT A: NEXT`

De mit csinál a számítógép az indítás után? Elfelejtett volna számolni? Hiszen a megjelenő értékek semmiképp nem lehetnek helyesek!

Hogy megértsük, miért is kószált el a C 64 valami bizonytalan területre, egy ROM-listát kell elővinnünk. A TO parancsig minden rendben is megy. A BASIC ROM-ban ez a cím. A következő \$A775 címen olvassuk be a TO mögött álló számot. Ezt az értéket egy kis rutinnal (LDA \$66:ORA#\$7F: AND \$62: STA \$62) egy másik számformátumba (lebegőpontos, vagyis memory floating print) konvertáljuk. Egy ilyen szám egy exponensból és egy mantisszából áll. Integer alakban a számok csak 32 767 és — 32 768 közötti egész értékeket vehetnek föl. A lebegőpontos számok viszont „tetszőlegesekek” lehetnek, és a tizedespont utáni értékek is megengedettek.

A ROM rutin a \$AE43-nál a kapott számot a stackbe írja. Előbb azonban kerekítést is el kell végezni, amelyet a \$BC1B kerekítőrutin végez el. Az ide való ugráskor a FAC (lebegőpontos akkumulátor) tartalma a 265-3E-8 beadása esetén:

```
>$0061 $88 $7F $FF $FF $FF
```

A kerekítőbyte a \$0070-es pozícióban a \$80-as értékkel rendelkezik. Ez a C 64-esnek azt jelzi, hogy fölfelé kell kerekíteni. Ez a rutin azonban nem alkalmas lebegőpontos számok kerekítésére! Így itt a számot eggyel megnöveljük a \$B96F-nél. Az FAC tartalma ezután az alábbi lesz:

```
>$0061 $88 $89 $00 $00 $00
```

Ez azonban semmiképp nem a 256-nak, hanem a mínusz 128-nak felel meg. A FOR—NEXT hurokban tehát nem fölfelé, hanem lefelé számolunk. Bizonyítékként itt a modifikált FOR—NEXT hurok:

```
10 FOR A=0 TO 256 — 3E-8 STEP-1: PRINT A: NEXT A
A C 64-es 0-tól mínusz 128-ig számol visszafelé.
```

SPRITE-OK TISZTA KIKAPCSOLÁSA

Azokat a sprite-okat, amelyeket a VIC 29-es regiszterrel az X irányban fölnagyítottunk, rendes körülmények között nem lehet a bal szélen a keretből kiúztatni. Ugyanis ezek 48 pixel szélesek, viszont a bal szél „alatt” csak 24 pixel hely van. Ezért a nagyított sprite-ok félig beúsznak a keret „alá”, azután hirtelen teljesen eltűnnek. Mindez igen amatőr benyomást kelt a szemlélőben. Van azonban egy trükk, amivel a széles sprite-okat is lágyan a keret alá lehet „söpörni”:

Ha a sprite fele eltűnik (az X pozíció most nulla), kapcsoljuk be a sprite-hoz tartozó bitet a 16-os regiszterben, és az X pozíciót növeljük 247-re. Csökkentsük most lépésről lépésre az X értékét 224-re, mire a sprite lágyan eltűnik.

Ezt a rutint feltétlenül gépi kódban kell megírni, a 16-os regiszter megfelelő bitjének bekapcsolását pedig raszterinterrupttal kell a képernyői ablakon kívül végrehajtani, ellenkező esetben a kép kellemetlenül remegni fog.

HATÉKONY LOAD ÉS SAVE

Az a felismerés, miszerint a C 64-es BASIC nem épp a legkényelmesebb nyelv, oly régi mint maga a gép. Például a gépi kódú programok tárolása és betöltése kétségkívül azokhoz a munkálatokhoz tartozik, amelyek nehéz próbát jelentenek.

A LOAD „név”, 8,1 utasítás bizonyára okozott már idegösszeomlást némelyeknél, mivel parancsmódban kiadva a programvége mutatókat reménytelenül elhajlítjuk, az ezt helyreállító NEW utasítás viszont a tárolóban álló BASIC programot tünteti el.

Emellett az is előfordulhat, hogy egy gépi kódú programot a tárolt kezdőcímetől függetlenül szeretnénk valahová a tárolóba vinni, amit viszont csakis külső segítséggel lehet megoldani. Tétélezzük fel, hogy szeretnénk két olyan rutinot dolgozni, amelyek azonos területen állnak a tárolóban (\$ C000 effektus), mondjuk azért, hogy a kettőből egyet csináljunk. De mi a teendő, ha épp nem áll a rendelkezésünkre a SMON?

A másik probléma azután tetszőleges tartomány kimentése. Monitorprogram nélkül ez kész vesszőfutás. Az igaz, hogy olvashattunk erről itt meg ott néhány SYS-t és POKE-ot, de hol is?

Nem kell tovább bizonygatni, a LOAD és SAVE használata egy sor problémát jelent a napi gyakorlatban. De ennek mostantól vége! A „THAT’S IT!” program megoldást kínál.

Elismerjük, a program neve kissé szokatlan, csakhogy a kínált megoldás is az. A THAT’S IT a nevezett problémákat mind megoldja, ráadásul meglehetősen kis ráfordítással, és még az agyunkat sem kell megerőltetni mindenféle paraméter megjegyzésével. A rutin a VIC terület előtti 72 byte-ot foglalja el, és természetesen önmagával együtt (!) bármilyen más területre elvihető, miközben ott is bármilyen módosítás nélkül használható.

A készülékszámok és a másodlagos címek megadásától szándékosan eltekintettünk. A beállított (default) érték a 8-as (a floppyhoz), amelyet a kezdőcímet utáni negyedik pozícióban találunk. A POKE 53180,1 (az eredeti helyen) azonnal a kazettára állítja a rutint. A másodlagos cím a fölhívásból adódik, azt kívülről nem lehet manipulálni. Mivel alapvetően semmiféle mutatót nem állítunk át, ezért a segítséget ne használjuk „normál” BASIC programok betöltéséhez.

De akkor hogyan működik az egész egyetlen kezdőcímet megadásával? Egész egyszerűen. A paraméterek száma alapján döntünk arról, mi a teendő. A SYS fölhívást mindenestre elválasztójel nélkül kell beadni, ezt követően pedig az idézőjelben álló programnevet: SYS 53176 „név”.

Ha így a nevet nem követik paraméterek, akkor a „név” gépi kódú programot abszolút módon, azaz a letárolt kezdőcímet megfelelően (8,1) hívjuk be a tárolóba. De ha akarjuk, a nevet követheti egy vagy két, vesszővel elválasztott cím:

```
SYS 53176 „név”, cím
```

Ekkor a nevezett gépi kódú programot a normál kezdőcímetől függetlenül a „cím”-re töltjük be.

```
SYS 53176 “THAT’S IT!”, 828
```

Ez az utasítás például a rutint a kazettapufferbe tölti be. Ha a nevet két cím — pl. SYS 53176 „név”, cím1, cím2 — követi, akkor a „cím1” és a „cím2” által közrefogott területet a „név” alatti adathordozóra írjuk. Azt jegyezzük meg, hogy a „cím2” cellában álló értéket már nem visszük ki. A „cím2”-t tehát ennek megfelelően kell megválasztani. Példa:

```
SYS 53176 „név”, 49152,50000
```

Ez az utasítás a 49152 és a 49999 közötti területet menti ki.

Name :	that's it!	cfb3 d000						
cfb3 :	20 57 e2 a2 08 86 ba 20							
cfb0 :	79 00 f0 2c 20 fd ae 20							
cfb8 :	8a ad 20 f7 b7 48 20 79							
cfb4 :	00 f0 15 68 84 c1 85 c2							
cfb2 :	20 fd ae 20 8a ad 20 f7							
cfb6 :	b7 84 ae 85 af 4c ed f5							
cfb1 :	68 84 c3 85 c4 a0 00 2c							
cfb3 :	a0 01 84 b9 a9 00 4c a5							
cfb5 :	f4 3c 36 34 27 45 52 3e							

REND A SZÁMÍTÓGÉPBE

Rendszeresen szükségünk van arra, hogy a programjainkban neveket, zenecímeket vagy egyéb szövegeket ABC sorrendbe rendezzünk. Az egyik legegyszerűbb módszer erre az

ügynevezett buborék rendezés (Bubblesort) alkalmazása. Hogy ezt bemutathassuk, előbb egy szöveges táblázatra van szükség:

```
10 PRINT CHR$(147); DIM T$(15)
20 FOR T=1 TO 15: PRINT „ADATSORSZÁM”; T;: IN
PUT $(T): NEXT T.
```

A 10-es sorban T\$ néven egy táblázatot hozunk létre, amely 15 elemből áll. A 20-as sorban egy hurkot építünk föl, amely a táblázat feltöltéséről gondoskodik. A felhasználó itt tetszőleges szövegeket adhat be. Ha most rendezni akarjuk az egészet számítógéppel, az alábbi sorokat kell beadni:

```
50 FOR A=1 TO 14: FOR B=A+1 TO 15
70 IF T$(B)<T$(A) THEN P$=T$(A): T$(A)=T$(B):=P$:
PRINT“.”;
80 NEXT B,A
```

Az 50–80. sorok között a táblázat minden elemét összevetjük a mögöttes elemmel (az 1-est a 2–15-tel, a 2-est a 3–15-tel, stb.). Mivel a táblázat rendezett állapotában az összes összehasonlított elemnek nagyobbak kell lenni, mint a kiindulási elem, a 70. sorban megcseréljük a kettőt akkor, ha ez így volna. Ugyanott a PRINT parancs egy pont képernyőre írásával jelzi, ha ilyen csere történt. Ha még az alábbi sorokat is beszurjuk a programba, a készre rendezett táblázatot a képernyőn is megismerhetjük és ellenőrizhetjük.

```
100 PRINT CHR$(147): FOR T=1 TO 15: PRINT
“ELEM”; T;: „T$(T)
110 NEXT T
```

A következő kérdés most az, hogyan lehet ebből a rendezett táblázatból egy meghatározott elemet előkeresni? A legegyszerűbb módszer szerint a keresett elemet a táblázat minden egyes bejegyzésével összevetjük. Ehhez az alábbi sorokat kell a programhoz fűzni:

```
140 PRINT: INPUT „MELYIK ELEMET KERESSÜK”; S$
```

```
150 FOR T=1 TO 15
160 IF T$(T)=S$ THEN PRINT “A(Z)” CHR$(34); S$;
CHR$(34);” ELEM SORSZÁMA: “;T
170 NEXT T
```

DIRECTORY PROGRAMVESZTÉS NÉLKÜL

Sokan a cím elolvasásakor biztos azt mondják: „Már megingt egy szörnyen lassú BASIC directory beolvasó rutin...” Csak semmi pánik! Ezúttal valami egészen új dolog jön.

A lemezek tartalomjegyzékének betöltésénél az a nagy probléma, hogy a BASIC tárolóra van szükségünk, így egy esetleg ott álló program elvész. Eleddig tehát előbb az ott álló programot tárolni kellett, ezután behívhattuk a directoryt, majd a megtekintés után ismét az eredeti program visszahívásával folytathattuk a munkát. Ez bonyolult, körülményes, és fölöslegesen terheli a lemezt és a lemezegységet.

Miért nem csinálja azt a rutin, hogy kikeres magának egy szabad memóriaterületet anélkül, hogy bárkit is zavarna!

Nos, mi sem könnyebb ennél!

Adjuk be az alábbi parancsot: POKE 44, PEEK (46)+1

Ezzel a BASIC programok számára fönntartott hely kezdetét jelző címet a tárolóban álló BASIC program mögé irányítjuk. Ha most behívjuk a LOAD „\$”, 8-cal a directoryt, LIST-tel gond nélkül megtekinthetjük azt. Legvégül a POKE 44,8 utasítást kell beadni, és a BASIC programunk érintetlenül ismét a rendelkezésünkre áll. Azt editálhatjuk, tárolhatjuk, futtathatjuk.

JÚNIUSI AJÁNLATUNK:

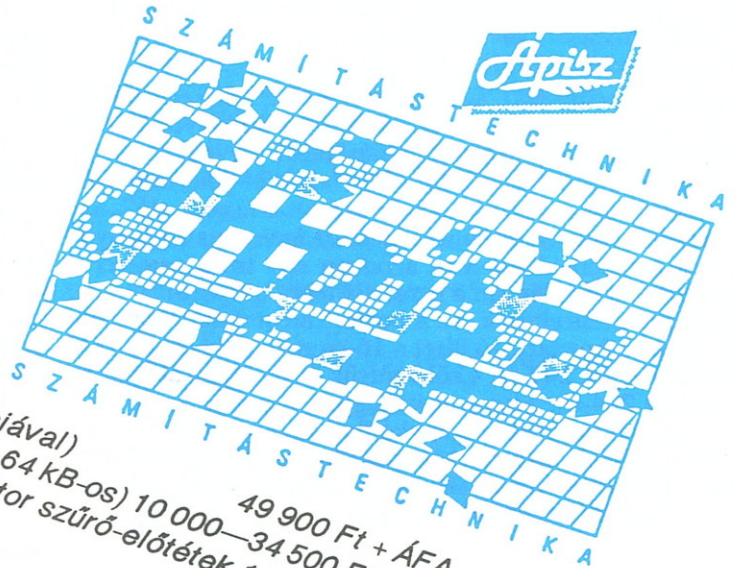
- Epson FX-1050 nyomtató (garanciával)
- Casio manager kalkulátorok (32, ill. 64 KB-os)
- 12" és 14" monochrom/színes monitor szűrő-előtétek 1 600—2 050 Ft között

49 900 Ft + ÁFA
34 500 Ft között

SAKÜZLETEINK:

Kínálunk továbbá festékszalagokat, tisztítószereket, különféle tartozékokat és szakkönyveket széles választékban, 1-, illetve 2-, 3- és 4- példányos IMPORT LEPORELLÓ-PAPÍROKAT szinte valamennyi méretben.

Budapest VIII., Szigony u. 15., tel.: 114-3446
Budapest, XI., Budafoki út 7., tel.: 181-1107



A MAGYAR OLIMPIAI BIZOTTSÁG HIVATALOS TÁMOGATÓJA

*Mi már tudomásul vettük, hogy
életünk szerves részévé vált a
számítástechnika.*



- számítógépek
- szoftverek
- szakkönyvek
- hardver kiegészítők
és különféle médiák széles
választékával várja Önt a

**2C áruház a mindennapok
számítástechnikájával**

NOVOTRADE

**2C ÁRUHÁZ, XIII. Balzac u. 35.
Telefon: 402-954**



KOGINFORM

KOGINFORM

MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET

Levélcím: 1325 Újpest 1. Pf.: 159

Telefon: 189-6142, 169-2989, 160-0611 Telefax: 169-2989, 189-6142



KOGINFORM

OKTATÓPROGRAM-KATALÓGUS

6. rész

A „KOGINFORM a számítástechnikával támogatott oktatásért” programja keretén belül most a kémia tanításához készített oktatóprogramokat ismertetem.

A katalógusban szereplő újabb rövidítések:

Osztály rovatban: T — tanárok

Programtípus: JJ — jutalomjáték

L — leírás

(magyarázat)

LJ — logikai játék

Tb — táblázat

Tk — találd ki!

To — totó

W — verseny

Továbbra is várjuk a kedves olvasók véleményét a katalógus bővítéséhez, karbantartásához.

PROGRAM NEVE	Komm. nyelv	TÉMAKÖR (megjegyzés)	Pr. típus	Osz- taly	Gép	A program készítője	Ter- jesztő	Ár (Ft/adat- horozó)	M. érték
Algoritmus KÉ/V—101	magy.	Telített szénhidrogének égési egyenletének helyes felírását segíti megtanulni.		8	TVC		TUD.	250/k 375/l	
Az anyag változásai (2 részes) *	magy.	— Anyagok csoportosítása, anyagváltozások és energiaviszonyok. — Feladatok: anyagok megkülönböztetése, kémiai reakciók csoportosítása, halmazállapot-változások energiaviszonyai, tűzoltás.	L F,T	7	C16	dr. Győri Erzsébet Gátfalvi Miklós	NOV.	306/K	+
Az atom szerkezete (2 részes) *	magy.	— Atommodellek, elemi részecskék, elektronhéj szerkezete. — Feladatok: elemi részecskék tulajdonságai, tömege, elektronhéjak felépítése, atomok elektronszerkezete.	L F,T	7	C16	dr. Győri Erzsébet	NOV.	306/K	+
Boszorkánykonyha	magy.	Kalandozás az ált. iskolában tanult vegyületek, elemek, kémiai folyamatok világában. Cél: egy előre kigondolt vegyület megtalálása. A tanulónak a gép kérdéseire kell válaszolnia, s a gép találja ki, mire gondolt a „játékos”.	OJ	7—8 (9—)	C+4 TVC	Gerlai Tiborné Papp György	NOV.	454/K 454/K	+
Cukrok térszerkezete	több	Előkészületben.			IBM		TUD.		
Egyenletek KÉ/33	magy.	Kémiai egyenletek írásának gyakoroltatása: a gép által kiírt hiányos egyenlet befejezésével (a hiányzó számokat, ill. teljes képletet kell pótolni.)			C+4		TUD.	375/K 563/l	
Elektrolízis KÉ/ISZ—104	magy.	Oldódást, vízbontást, ill. a réz rafinálását szemlélteti.	SZ		C64		TUD.	812/K	+
Elektronszerkezet I—II.	több	Előkészületben (2 program).			IBM		TUD.		
Elemek	magy.	A kémiai periódusos rendszer eddig felfedezett elemeinek tulajdonságait dolgozza fel. A különböző tulajdonságok között összefüggések grafikonon ábrázolhatók. Az elemek többféle szempontból rendezhetők is.	L,G Tb	7—11 T	C+4	Bálint György	NOV.	345/K	+
Galvánelemek	több	Előkészületben.			IBM		TUD.		
Hidratáció	magy.	1. Ionkristály oldódása vízben, ionok hidratációja. 2. Tetszés szerinti képződmény létrehozása; (néhány tipikus szerkezet bekereshető). A program ezen része tetszés szerint alakítható.	Sz G	7—11	C+4	dr. Deák László Szenási Lajos	VOR.	245/K	
Az ionok kialakulása (2 részes) *	magy.	— Kationok, anionok keletkezése. — Feladatok: ionok képződése, jelölése, elektronszerkezete, tömegük kiszámítása.	L F,T JJ	7	C16	dr. Győri Erzsébet Gátfalvi Miklós	NOV.	306/K	+
Ionos kémiai kötés KÉ/100	magy.	Az elemek periódusos rendszerére és az ionos kötésre vonatkozó, egyéni gyakorlásra alkalmas feladatsorokat tartalmaz.	F	7—8	C+4		TUD.	375/K	

PROGRAM NEVE	Komm. nyelv	TÉMAKÖR (megjegyzés)	Pr. típus	Osz-tály	Gép	A program készítője	Ter-jesztő	Ár (Ft/adat-horozó)	M. érték
Kémia 7. osztály *	magy.	10 részes programsorozat, a hozzá tartozó programokat jelöltem *-gal. Alt. 2 fő részből állnak: — oktató (elméleti), — feladatmegoldó; ugyanazon feladat többszöri megoldása esetén az adatok más-más variációban jelennek meg. Értékelés: jó/rossz válaszok száma, %-os. 8. osztályos ismétléshez is ajánlott. (Több részből állnak a C 16 memóriája miatt!)	L F,T	7 (8)	C16		NOV.		+
Kémiai egyenletek	magy.	Egy konkrét, tanított reakción apró lépésekkel haladva magyaráz, kitér minden elképzelhető hibára. Részletes értékelést ad; gyakoroltat.	L F	7—9	C+4	dr. Deák László	VOR	199/K	
Kémiai kötések azonos atomok között (2 részes) *	magy.	— Kovalens és fémes kötések kialakulása, jellemzői, energiaviszonyai. — Feladatok: atomok/molekulák jelölésének megkülönböztetése, vegyjel és képlet mennyiségi jelentése; gyémánt, grafit, fémek tulajdonságai.	L F,T TO TK	7	C16	dr. Győri Erzsébet Gátfalvi Miklós Laczi András	NOV.	306/K	+
Kémiai kötések különböző atomok között (2 részes) *	magy.	— Kovalens és ionos kötések. — Feladatok: vegyület- és elemmolekulák, poláris és apoláris kötés; képlet menny. jelentése; kémiai egyenlet; ionok töltése, ionarány, ionvegyületek képletének jelentése.	L F,T	7	C16	dr. Győri Erzsébet	NOV.	306/K	+
Kémiai reakciók	több	Előkészületben.			IBM		TUD.		
Kémiai reakciók I. (Egyesülés, bomlás) (3 részes) *	magy.	— Kémiai egyenlet írásának fontosabb lépései. — Feladatok: 20 egyenlet szerkesztése. — Egyesülés (egyszerűbb és bonyolultabb.) — Bomlás.	L F,T	7	C16	dr. Győri Erzsébet Gátfalvi Miklós	NOV.	306/K	+
Kémiai reakciók II. Redoxi, protolitikus (3 részes) *	magy.	— Elektronátmenettel és protonátmenettel foglalkozó reakciók; sav—bázis reakciók. — Feladatok: ionok/reakciók csoportosítása, oldatok kémhatása, egyenletek felírása.	L F,T	7	C16	dr. Győri Erzsébet Laczi András	NOV.	306/K	+
Kémiai reakciók sebessége KÉ/99	magy.	5 részből áll, ezek a kémiai folyamatok sebességét befolyásoló tényezőket demonstrálják.	Sz	7—8	C+4		TUD.	313/K	
Kémiai vizsgaprogramrendszer (20 pr.)	magy.	— Gyakoroltatás. — Tesztelés. — Vizsgáztatás.	Gy T É		C+4		BAZ.	2250/I	+
Kinetan programcsomag KÉ/ISZ—7	magy.	18 alapvető reakciótípust tartalmazó könyvtárból választott reakcióban számítja és a felhasználó által választott módon ábrázolja a komponensek koncentrációváltozásait, a teljes sebességeket és a reakciósebességeket (kémiai kinetika tanításához).	Sz Ab		C64		TUD.	1500/K	
LAVOISER KÉ/V—16	magy.	Szerves molekulák szabályos nevét tanítja: alkánok, alkének, alkanolok, alkoholok, éterek.		10	TVC		TUD	438/K	
MEND KÉ/V-04	magy.	Mengyelejev-féle periódusos rendszer törvényszerűségeit segíti felismerni, alkalmazni adatok becslése alapján.		9	TVC		TUD	313/K	
Mólok KÉ/20	magy.	Vegyjel, képlet, anyagmennyiség témaköréhez kapcsolódik.	F,T JJ		C+4		TUD	313/K	
Az oldatok (2 részes) *	magy.	— Oldószer és oldott anyag kölcsönhatása, oldhatóság. Oldódás energiaviszonyai, anyagok szétválástása. — Feladatok: oldódás energiaváltozásai, oldódás különböző oldószerekben, oldatok töménysége.	L F,T	7	C16	dr. Győri Erzsébet Gátfalvi Miklós	NOV.	306/K	+
Periodusos rendszer	több	Előkészületben.			IBM		TUD		
A periodusos rendszer (2 részes) *	magy.	— A periódusos rendszer kialakulása, felépítése. — Feladatok: atomok elektronszerkezete; periódusos rendszer felépítése, szabályai, elemei; a vegyjel melletti számok jelentése.	L F,T	7	C16	dr. Győri Erzsébet Gátfalvi Miklós	NOV.	306/K	+
A periodusos rendszer elemei	több	Előkészületben.			IBM		TUD		
Periódusos rendszer főcsoportjainak elemei	magy.	A 8. osztályban tanult főcsoportok elemeinek tulajdonságai, az elemek felhasználása (játékos formában).		8	C16		BAZ.	1875/I	+

PROGRAM NEVE	Komm. nyelv	TÉMAKÖR (megjegyzés)	Pr. típus	Osz-tály	Gép	A program készítője	Ter-jesztő	Ár (Ft/adathorozó)	M. érték
Sűrűség KÉ/18	magy.	Ált. iskolai fizikában tanult fogalmak alkalmazása a kémiában. 5 kérdést ad fel.	F		C+4		TUD	313/K	
Titrlás	magy.	Titrlás ismertetése; konkrét alkalmazást szimulál grafikusan és kvantitatív módon. A tanulóknak rá kell jönniük a titrlásnál ténylegesen alkalmazott közelítő módszere.	Sz Lj W	7—11	C+4	dr. Deák László	VOR.	188/K	
Titrlás	több	Előkészületben.			IBM		TUD		
Tömeg KÉ/20	magy.	Oldatok összetétele (tömegszázalékban). Kétféle kérdészi forma közül lehet választani.	F		C+4		TUD	313/K	
A vegyjel (2 részes) *	magy.	— A kémiai írásmód kialakulásának története az alkímistáktól napjainkig; elemek vegyjelei. — Feladatok: anyagmennyiség- és atomtömeg- számítások; vegyjelek gyakorlása.	L F,T JJ	7	C16	dr. Györi Erzsébet Gátfalvi Miklós	NOV.	306/K	+

Mindenkinek jó tanulást/tanítást kíván az összeállítás szerzője: LUGOSI ANTALNÉ



KOGINFORM

MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET

Levél cím: 1325 Újpest 1. Pf.: 159

Telefon: 189-6142, 169-2989, 160-0611 Telefax: 169-2989, 189-6142



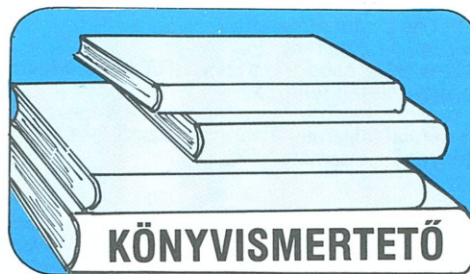
Bálint Á. — Tátrai F. :

Gyakorlati statisztikai számítások

(C 64-es számítógépre)

A mérnöki és közgazgatási, orvosi stb. gyakorlatban nagyon sok statisztikai jellegű adat feldolgozására van szükség.

A könyv első részében a gyakorlat szempontjából legfontosabb próbák és



legismertebb eljárásokat, hanem bemutatja az új módszereket is. A C 64-es és C 128-as gépekre teljes megoldásokat tárgyal, érthetően dokumentálva (programlisták, gépi nyelvű listák). Kitér a kazetta és mágneslemez másolásának védelmére, bemutat védett programokat assembly és BASIC nyelven.

Ára: 249,- Ft
Megjelenés: 1989.



elvégzésüket megkönnyítő programok ismertetésére kerül sor. A második részben a különböző alakú, egyszerű függvények illesztésére, ill. adatok közötti összefüggések vizsgálatára mutatnak be a szerzők a gyakorlatban is jól használható eljárásokat és programokat. A könyvhöz lemez is vásárolható.

Ára: 390,- Ft
Megjelenés: 1989.

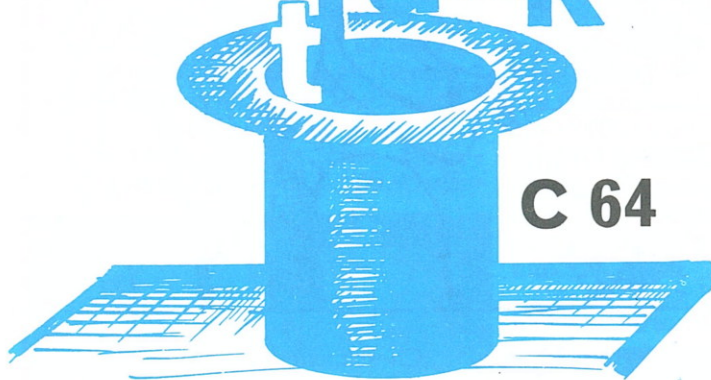
Lengyel J. — Varga A. :

Lakat alatt — védelmi módszerek C 64-esre

A könyv programvédelmi és másolásvédelmi eljárásokat tárgyal. Nemcsak összefoglalja az eddig alkalmazott



Tippek trükkök



TÁROLÓTÖRLÉS? SEMMI GOND!

Az Assemblerben programozó kezdők és profik számára is néha komoly gondot okoz, hogy valami okból a C 64-es tárolóját nullákkal kell feltölteni. Aki dolgozott már gépi kódú monitorral, az ismeri az F (Fill) parancsot. Másról (pl. az SMON-nál) a W (Write) szolgál erre. Viszont van egy jobb módszer, amely kicsit gyorsabban dolgozik, és a komplett BASIC tárolót a \$0801-től a \$9FFF-ig nullákkal tölti fel. Adjuk be sorszám nélkül az alábbi utasítást:

```
DIM A (7777): NEW <RETURN>
```

Ekkor azonnal „nullára” söpörjük a megadott területet.

AZ ÖSSZES BASIC PARANCS

Számos BASIC bővítés kínál a felhasználóknak olyan parancsot, amellyel a rendelkezésre álló utasításkészletet ki lehet írni a képernyőre. A C 64-es BASIC V2.0-ja ilyen sajnos nem ismer. Megoldás volna egy olyan program, amely az összes, a kézikönyvben szereplő parancsot átvinné, és ezeket a PRINT utasítással kijelezné. Sajnos azonban ez nagyon körülményes, és igen sok tárolóterületre volna szükség.

Van azonban egy sokkal egyszerűbb módszer is. Mivel logikus, hogy a gép a saját parancsait megérti, valahol föl kell hogy legyen sorolva egy táblázatban az összes információ ezekről. Ez a táblázat az úgynevezett BASIC ROM-ban áll, és a C 64-esnél a 41 118-as címen kezdődik. Itt állnak a parancsok, igaz furcsa rend szerint, és sajtóságos formában. Csupa kis betűt találunk, csak az utolsó karakter nagy betű. A táblázat eleje például így néz ki: enDfoRnexTdatA... Aki alaposabban odanéz, fölismerheti az END, a FOR, a NEXT és a DATA parancsot.

Az alábbi kis program semmi mást nem csinál, mint kiolvassa ezt a táblázatot, korigálja a sajtóságos kisbetű nagybetűs „keveredést”, és kilistázza a parancsokat a képernyőre.

A C64-ES BASIC parancsainak kijelzése.

```
10 PRINT "CLR": P=10: FOR I=0 TO 254: X=PEEK (41118+I)
```

```
20 IF X>128 THEN X=X-128: T=P:P=P+10+40*(P=40):
```

```
GOTO 40
```

```
30 T=0
```

```
40 PRINT CHR$(X) TAB(T); NEXT
```

Kérem, hogy a **Commodore Újság** legközelebbi számában jelenjen meg a következő szövegű apróhirdetés:

(minden oszlopba egy betűt írjon!)

Név és cím:.....

Feladandó az újság címére:

Commodore Újság
Budapest, 1388. Pf. 86.

Alulírott megrendelem a következő programokat a PÖTYÖGŐSZOLGÁLAT-tól:

PROGRAM	SORSZÁMA	ÁRA
1.	<input type="text"/>	<input type="text"/> , - Ft
2.	<input type="text"/>	<input type="text"/> , - Ft
3.	<input type="text"/>	<input type="text"/> , - Ft
4.	<input type="text"/>	<input type="text"/> , - Ft
5.	<input type="text"/>	<input type="text"/> , - Ft
6.	<input type="text"/>	<input type="text"/> , - Ft
7.	<input type="text"/>	<input type="text"/> , - Ft
8.	<input type="text"/>	<input type="text"/> , - Ft
9.	<input type="text"/>	<input type="text"/> , - Ft
Összesen:	<input type="text"/> db	<input type="text"/> , - Ft

A programokat a SZOLGÁLAT által biztosított

lemezre (99 Ft/db)

kazettára (40 Ft/db)

az általam küldött adathordozóra kérem.

(Kérjük a megfelelő szöveg aláhúzását!)

POSTAköltség (25 Ft)

A fizetés módja: személyesen — csekken — utánvéttel

(Kérjük a megfelelő szöveg aláhúzását!)

ÖSSZESEN:

A megrendelő neve:

Címe:

PÖTYÖGŐ SZOLGÁLAT

sorszám	tipus	megjelenés	a program neve	ára
210	C64	90/2	Supervisor	40,-
211	PLUS/4	90/2	Véletlen képcseré	40,-
212	PLUS/4	90/3	Karaktertervező	50,-
213	C64	90/4	Zsolnai átlagszámító	30,-
214	C64	90/4	Labdák	30,-
215	C64	90/4	Funkcióbillentyűk	40,-
216	C64	90/4	Futó felirat	40,-
217	C64	90/4	Jegyzetömb	20,-
218	C64	90/5	Videofelíratkozó	40,-
219	C64	90/5	DUMP	50,-
220	PLUS/4	90/5	Büvös négyzet	60,-
221	C64	90/5	BONGO	40,-

A teljes pötyögőlista az 1990 májusi számunkban található.

JÁTÉKPIACI ÚJDONSÁGOK

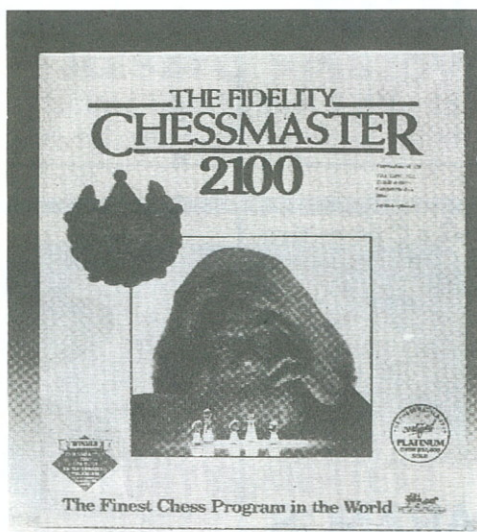


Copyright anno 1964...

Állhat-e egy 1990-es számítógépes Magazin „újdonságok” rovatában egy új játéknál a fönti copyright felirat? Igen, ha a „Batman”-ról, az ismert mozifilmmel azonos nevű játékról van szó. Az ötlet ugyanis már huszonhat évvel ezelőtt megszületett. A programban ugyanúgy, mint a filmben, Bruce Wayne, azaz Batman feladata, hogy egy egész várost megvédjen a bűnözők hordájától. Szembenáll vele természetesen fő ellensége, Joker.

A játékos feladata, hogy öt szinten (Axis, Chemical Plant, Gotham City, Batcave és Gotham Cathedral) küzdje át magát. Az ismert eszközök, úgymint a Batarang, a Batmobile, a Batwing és a Batrope persze a rendelkezésére állnak.

„Batman”, gyártó: Ocean, ár: 49,95 márka (lemez), 34,95 márka (kazetta), forgalmazó: Ariola Soft GmbH, Hauptstr. 70, D-4835 Rietberg 2.



Még a „legjobb” is jobb lett

A „legjobb”-nak a Chessmaster 2000-t nevezték, amelynek most megjelent az utóda Chessmaster 2100 néven. Ez a sakkprogram nemcsak magát a sakkjátékot sajátította el perfekt módon, de egy sor plusz trükkel is kitűnik. A megszokott funkciók, mint a játékerő beállítása, a pillanatnyi állapot kimenthetősége, betölthetősége, vagy a játékgurák tetszőleges elrendezése (feladványállások) mellett a Chessmaster 2100 lehetővé teszi a sakkjáték aktív elsajátítását. Erre olyan játékmódok szolgálnak, melyekben a program bemutatja a lehetséges lépéseket, konkrét javaslatokat tesz, betekintést ad a saját stratégiai terveibe, vagy elemzi és értékeli az emberi játékerőt.

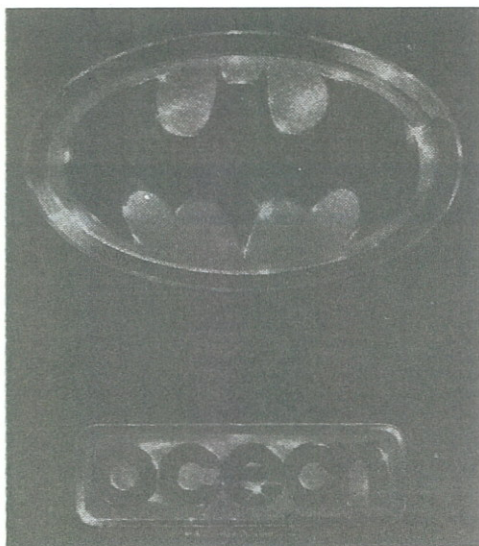
Természetesen klasszikus partik utánjátszása is lehetséges.

„Chessmaster 2100”, gyártó: The Software Toolworks, ár: 49,95 márka (lemez), forgalmazó: Rushware, Bruchweg 128—132, D—4044 Kaarst 2.

The Final Chesscard

Új dolgokról ad hírt a sakkprogramok világában a VTS Data szoftverház. „Final Chesscard” néven egy saját akkumulátorral ellátott sakkmodult kínálnak, amely állítólag minden eddigi, a C 64-es sakktemában közreadott programot zsebre vág. Mindezt a modulban található saját, ird és mondd 5 (öt!) Mhz-cel meghajtott 6502-es mikroprocesszorral indokolják. A modul tehát ötször olyan gyorsan dolgozik, mint a C 64 6510-ese. A „The Final Chesscard” így jóval több lépést tud „kikalkulálni” előre, mint az eddig közreadott sakkprogramok. A modult 179 márkáért árulják.

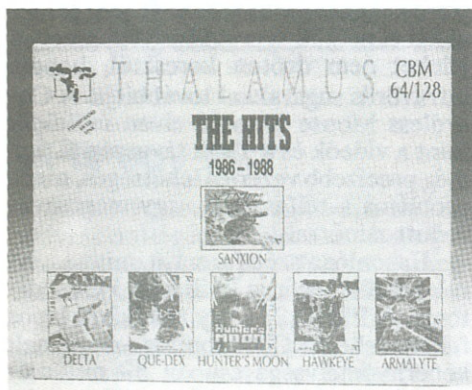
„The Final Chesscard”, gyártó: VTS Data, ár: 179 márka, forgalmazó: Rushware, Bruchweg 128—132, D—4044 Kaarst 2.



A Sanxion-tól az Armalyte-ig

„The Hits 1986—1988” címen egy gyűjtemény jelent meg, amely a Thalamus szoftverház legjobb eddigi programjait tartalmazza. A csomagban a Sanxion, a Delta, a Quedex, a Hunter's Moon, a Hawkeye és az Armalyte játékokat találjuk. Akcióból és lövöldözésből tehát szép a kínálat. A gyűjteményt a Rushware forgalmazza, 59,95 márkáért.

„Thalamus — The Hits 1986—1988”, gyártó: Thalamus, ár: 59,95 már-



ka (lemez), forgalmazó: Rushware, Bruchweg 128—132, D—4044 Kaarst 2.

Rövid teszt: A SHARP IQ—7100

A Sharp IQ—7100 néven egy sokoldalú elektronikus szervezőt kínál. A kijelzés egy LCD képernyőn történik, amely 64x96 pontot tartalmaz (választhatóan 8x16 vagy 4x12 karakter). Az alfanumerikus billentyűzet segítségével (amelyen az ismertebb különleges jeleket is megtaláljuk) a 32 kbyte-os tárolóból 26 és felet töltöttünk föl. Kívánságra jelszóvédelmet is beiktathatunk. Az elemek kímélése érdekében a készülék automatikusan lekapcsol, ha 6 perc elmúltával sem nyomunk meg egyetlenegy gombot sem. Az óra, amely a világ 212 városára mindig kijelzi a pontos időt (még a nyári időszámítást is) azonban nem tartozik ide. A beépített naptár 1901-től 2099-ig képes nagyítószzerűen kijelezni egy adott hetet vagy annak egy napját. Egy terminus tervezett időtartama akár grafikus formában is megjelenhet. Az adott időponthoz egyéb adatokat is beadhatunk, amelyeket persze visszakereshetünk (keresés kezdőbetűk, dátum, kulcsszó szerint). Még az évente rendszeresen visszatérő eseményekre (születésnap) is gondoltak. Beépítettek

egy akusztikus időpont-émlékeztetőt, és egy maximum 700 számot befogadó telefonregisztert is. A memo saját számítási funkciókkal rendelkezik, amellyel az adatok között szereplő mérték vagy mennyiségi értékeket föl lehet dolgozni.

Az IQ—7100 a számítási műveleteket kalkulátorként végzi el, ahol a megszokott zsebszámológép-képességek mellett a „paperless printer” (papír nélküli nyomtató) funkció is megvan. Itt maximum 50 (ebből kilenc látható) bevétel marad képernyőn, hasonlóan egy üzleti blokkhoz, amelyeket (mint az IQ—7100 minden adatát) editálni lehet. További lehetőségek még: felhasználói szótár, időpecsételő, clípfunkció, memorycheck, kijelző kontrasztszabályozó stb. A 208 oldalas német nyelvű leírásból minden kiderül.

A Sharp az IQ—7100 számára különleges tartozékként egyéb kiegészítő perifériákat is kínál. Vannak további számítási kártyák (idő/költség menedzser, nyolcnyelvű fordító, thesaurus), egy kazettás deck az adatarshiválásra, kábel két IQ—7100 összekapcsolására, de akad egy nyomtató is. Végezetül a készüléket össze lehet kötni egy PC-vel, s az adatokat azután a nagygép segítségével kezelhetjük tovább.

Nem sokkal a lapzárta előtt kaptuk a hírt, hogy a legújabb modell az elődjével ellentétben gombnyomásra olaszra, an-



golra, németre, franciára, spanyolra, svédre vagy finnre vált.

Ha a számos funkciót figyelembe vesszük, a méretek elfogadhatóak, legalábbis mivel az IQ—7100M mint egy könyv összecukható. Sajnos az új modell túl nehéz lett, a súlya miatt a zakózebben való hordásról lemondhatunk. A szervező ára tartozékok nélkül 499 márka, az előző modellel még 449 volt.

Színek a falon

A Sharp Electronics bemutatta a jövő színes televízióját. 14 colos képítő, LCD technika és 1,2 millió képpont ga-

rantálja a kontrasztokban és kiváló minőségű színekben gazdag képet. A folyadékkristályos display csupán 27 mm vastag. A nagyképernyős televízióktól az „akaszd a falra” TV-kig már csak egy kis lépést kell tenni.

Egészségesebb monitorok

A Mitsubishi Europe a képernyők előtt végzett munka egészségügyi szempontból történő vizsgálatai alapján egy olyan monitort készített, amely kisérgázása az alacsony frekvenciatartományban csökkentett mértékű. Az EUM—1481A(T) az első Mitsubishi display, amely új vezetőrétegezéssel készül. A cég szerint az alacsony frekvenciájú sugárzás bőrérzékenységet, látáskárosodást, sőt maradandó, bőr alatti károsodást is előidézhethet. A képernyő felületén kialakuló elektrosztatikus mező por-szemcséket „szív” magához, és azokat az ellentétesen feltöltődő szemlélő arcába „lövi ki”.

Az újszerű, úgynevezett „Conductive Coating” redukálja az elektrosztatikus mezőket a több mint 10 000 V-ról 50 V alatti értékekre. Kívánatos mellékhatásként ezzel a monitorok elektromágneses sugárzása is csökken. Ezzel a „környezetbarát” technológiával hamarosan minden Mitsubishi monitornál találkozni fogunk 14—21 colos képátmérrő-családban.

Dohánymentes övezet az ATARInál?

„Hat nap plusz szabadság nemdohányzóknak”, ezzel a hírral lepte meg a számítógépgyártókat az Atari. Sok humorral kellett ezt fogadnia Horst Hövelsnek, a cég személyzeti vezetőjének, aki szenvedélyes dohányos. Jó néhány meggyőződéses nemdohányzó próbálta meg őt leszoktatni erről. Bemutattak neki akupunktúrás kezelést, hoztak „leszoktató” tárgyakat, aroma nélküli cigarettákat, sőt hat kazetta terjedelmű „pszichológiai leszoktató útmutatást” is.

Érdekes módon a nemdohányzó szabály közzététele után megnőtt az Atarihoz benyújtott jelentkezések száma! Négy héten belül 30 %-kal több szakember érdeklődött, mint eddig. De a dohányzó lobby sem tétlenkedett. Még az Atari-termékek bojkottjára is fölhívtak! A Család- és Egészségügyi Szövetségi Minisztériumban viszont pozitív vissz-

hangra találtak. De mindezeket figyelmen kívül hagyva tovább folyik a „több szabadságot a nemdohányzóknak” akció az Atarinál. Hogy emiatt dohányzó kollégákból nemdohányzók lennének, az kérdéses. A személyzeti főnök például úgy eregeti a füstöt, mint eddig még soha.

A Szilikonvölgy-sztori

Aki szeretne tájékozódni a számítógépesek Eldorádójáról, a Szilikonvölgyről, az ezt háromféleképp teheti meg. A Silicon Valley Stroy egy 500 oldalas könyvből, egy százperces filmből és egy lemezből áll, az utóbbit az „Everything You Want” című filmből vágják össze. Az anyagok szóban és képen mesélik el a legendás számítógépvölgy történetét. Az olvasó és a néző megtudhatja, miért pont oda települt a számítógépipar, hogyan találták föl az integrált áramköröket, és hogy lett ebből személyi számítógép. Megismerkedhetünk a számítógépes technika és az elektronika kezdeti éveivel, követhetjük a fejlődést az elektroncsövektől a tranzistorokon, az integritás IC-ken, a mikroprocesszorokon át a PC-kig. Elolvashatjuk, hogy készült az első PC-BASIC, az első szövegszerkesztő, az első táblázatkezelő. A video 129, a könyv 49, a dal 6 márkába kerül.

A legkisebb PC

Nem nagyobb, mint egy levéltárca az Atari új „Portfolio” PC-je. Az aprócska, a mellényzsebbe betehető komputer teljesen IBM-kompatibilis, 128 kbyte RAM-mal rendelkezik. Az LCD kijelzője 8 sorban 40—40 karaktert képes megjelentetni, a grafikus felbontás 240x64 pixeles. Mivel helyhiány miatt nem lehetett lemezegységet beépíteni, a legfontosabb programokat —, mint a címnyilvántartást, az időtervezőt, a szövegszerkesztőt és — táblázatkezelőt — mindjárt beépítették. Akinek a 128 kbyte RAM nem elég, csek-kártya formátumú memóriabővítőket csatlakoztathat. Erre a célra a gép oldalán egy vágást alakítottak ki. A Portfolio 800 márkába kerül, azt a szaküzletekben kereshetjük.

Mivel a számítógép teljes rendszerbuszát kivezették, egy sor fejlesztés várható. Az Atari a szaküzleteken keresztül kb. 100 márkáért egy adaptert fog árulni, mellyel a Portfoliót egy PC-hez lehet illeszteni. Itt azután programokat, adatokat cserélhetünk. A Roda cég a legutóbbi düsseldorf-i Atari-vásáron már kínált olyan kiegészítőt a Portfolió-

hoz, amely egy 3 1/2 colos lemezegységet, egy elemcsomagot, monitor interfészt, és egy külső billentyűzetet tartalmaz. A komputert egyszerűen be kell tolni a kiegészítő egy erre szánt részébe. A bővítést PROfilonak hívják, az ára kb. 900 márká lesz.

PC-k: Minden napra újdonságok

Egyetlenegy számítógépes piac sem tud annyi újdonságot fölmutatni, mint a PC-ké. Mi van a körön belül és kívül?

A legutóbbi idők kulcsszava ebben a körben a 486-os szám. E mögött a jelölés mögött egy új számítógépes processzorcsalád rejtőzik, amelynek teljes jele 80486. Ezek hivatottak a 80386-os topmodellét föl váltani. Az új chipek az elődökhöz képest főleg a két-négyszer nagyobb órajel-frekvenciával (munkasebesség) tűnnek ki és a sokkal kompaktabb fölépítéssel. A több mint egymillió tranzistorral 1,7 négyzetcentiméteren létrehozta a processzoron belül egy úgynevezett coprocesszort, azaz segítőt, amely a fő processzort a számítások elvégzésétől mentesíti, illetve találunk más elemeket és funkciókat is. Ha ezt a C64 6510-esével hasonlítjuk össze, a 80486 több mint százszor gyorsabb. Némely esetben ez a faktor még nagyobb.

Egy másik terület, ahol a dolgok egyre kisebbek és gyorsabbak lesznek, a

háttértárolóké. Például a Zenith nemrég bemutatta az első 2 colos lemezt, amelyre 720 kbyte információ fér el. Az ehhez tartozó mini lemezegységet a legújabb laptop (hordozható mini PC-k, aktatáska méretben) generációba fogják beépíteni. A Hitachi egy olyan optikai meghajtót készített, ahol cartridge-enként (ez a tárolás egysége) 644 Mbyte(!) információ tárolható. Ezeket a tárolókat mint a CD lemezeket, lézersugár segítségével lehet kiolvasni, sőt egy mágneses mező segítségével még a fölülírás is lehetséges. 48-as csoportba összefogott, úgynevezett Library Unit 30 Gbyte (gigabyte) tárolókapacitást mutathat föl.

A perifériákat illetően is számos dologon csodálkozhatunk. A BMC cég például kábel nélküli egeret készített, amely a jeleket nem dróton keresztül, hanem infravörös sugarakkal továbbítja. A Cordless Mouse hasonló elven működik mint a videók és a TV-k távvezérlői. Így még precízebb vezérlés lehetséges, nincs probléma a túlfeszített vagy megcsavarodott zsinórral.

Hasonlóan szép formájú, mint a „farok nélküli” egér, a Philips új PC monitorja. A PM 11211 egy rendkívül lapos fekete-fehér LCD monitor, amelynek képernyőmélysége csupán 53 milliméter, súlya pedig mindössze 3,3 kg. Az LCD technika, mint azt a karórákon és zsebszámológépeken láthatjuk, számos előnyt nyújt. Nincs például statikus föl-töltődés, mágneses zavar, röntgensugárzás, sem beégés, mint az a képcsövek öregedésénél megfigyelhető.

A szoftverek világában is történt egy s más. Így a Microsoft és a Stardivision



cég is bemutatta az 5.0-s Word, illetve Starwriter szövegszerkesztőjét. Mindkét új verziószámú rendszer a DTP (kiadványszerkesztés) felé hajló tendenciát mutat. Így a nyomtatás előtt egy teljes DIN A4-es oldalt tudunk megtekinteni a preview üzemmódban.

Új utakon halad a Microsoft a legújabb Pascal Compilerjével is. Az 5.0-ás verzió, hasonlóképpen a Borland 5.5-ös Pascaljához, olyan fordítót kínál, amely az úgynevezett „objektorientált programozást” támogatja. Ennél a teljesen új elvnel az eddig alkalmazott program és adatok (változó tartalmak) szigorú elválasztása teljesen elmarad. Ezt a két elemet ugyanis úgynevezett „objektokká” fogjuk össze, amelyek úgynevezett rugalmasabb programalakítást tesznek lehetővé.

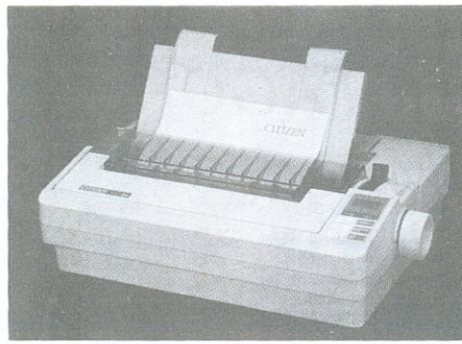
Mi is egy PC?

A PC (Personal Computer, azaz személyi számítógép) jelölés alatt foglaljuk össze mindazokat a profi feladatok megoldására készített számítógépeket, amelyeket az MS-DOS operációs rendszerrel szállítanak. Többféle kategóriába soroljuk be ezeket a PC-eket, ahol azonban az egyetlen definitív különbség az alkalmazott központi egység (mikroprocesszor) és az ezzel összefüggő munkasebesség.

Az egyes PC-kategóriák		
Jelölés	Processzor	Órajel
XT	8086	4,77MHz
	8088	4,77MHz
	8088-1	10 MHz
AT	286	12, 16, 20 MHz
	386	16, 25, 33MHz
	486	33-tól MHz

Új Citizen printer

Hosszú szünet után a Citizen új 24 tűs printert mutatott be, melynek képességei igencsak figyelemre méltóak. Anthony Odhams (Senior Product Marketing Manager Europe) kijelentése szerint az új nyomtató a Swift 24 névre hallgat, s ez lesz az elkövetkezendő két év legnagyobb számban eladott printere! A Swift 24 teljesítménye 192 cps sebesség, négy beépített betűkép, 8 kbyte puffer, papírparkoltató funkció, LCD kijelző, automatikus papírbehúzás, színes nyomtatás, kombinált húzó- és tolótraktor, és számos más is. Például mód van saját egyéni funkcióink programozásá-



ra. A Swift 24 kompatibilis az Epson LQ 850-essel és az X24 IBM proprinterrel, s természetesen a NEC P6 Plus-szal is. Ezen kívül használhatunk jelkészlet modulokat is. Az ár az adóval együtt 1098 márka körül lesz. A színes bővítés ára 104 márka, egy jelkészlet modul 128, a festékszalag 16 (a színes 55) márkába fog kerülni.

Börzeprogram

49 márkáért most kapható egy tözdeprogram a C 64-eshez. Amint azt a gyártó közölte, a program a kurzusok kézi bevitelén alapul. Év végéig viszont boltba kerül egy olyan változat is, amely képes lesz az árfolyamokat a Btx-ről beolvasni és lemezre tárolni. A kezdés megkönnyítésére szinte minden, a (legkülönbözőbb) börzéken követett részvény árfolyamának eddigi alakulását lemezen szállítani tudják.

A program képes arra, hogy az árfolyamok alakulását szép grafikák formájában a képernyőre vetítse.

A Quizmaker 64 rövid tesztje

A kvízzjáték mindig alkalmas társasági elfoglaltság. Épp ez adta az ötletet a Quizmaker-programhoz, amelyet ráadásul emellett még tanulásra is föl lehet használni. A programcsomag két részből áll, a Makerből és a Quizból. Az elsővel a felhasználó elkészítheti a kérdőívet. Itt a helyes válasz megadása mellett két hamisat is be kell vinni. A beadottakat a Maker egy szekvenciális állományban tárolja, ahol az állomány neve szabadon választható. A már meglévő Quiz-állományokat bármikor meg lehet változtatni, ki lehet bővíteni. A Maker-programrész tisztán fölépített menüket tartalmaz, amelyek között a kurzor és a funkciók billentyűk segítségével lehet kapcsolgatni. A képernyős maszk véletlen szétrombolása így gyakorlatilag ki van zárva.

A Quiz-programrész azután behívja a kérdéseket és játékra hívja ki a résztvevőket. Maximum négy játékos teheti próbára tudását. Az induláskor mind a négy játékos beírja a nevét, de ha nem, úgy egy számot kap. Ezután kérdez a program a megadott állomány szerint, s a helyes válasz mellé a két hibásat is kiírja. Ha a kurzorgombokkal kiválasztott válasz helyes, a játékos öt pontot kap a „számlájára”. Rossz válasz esetén nem jár semmi. Ezután a sorban következő játékos kerül terítékre. A játék addig tart, amíg a kérdésforrás el nem apad. De az F1 megnyomásával előbb is be lehet fejezni.

Egy lemezen több Quiz-állomány is lehet, de a program mindig csak az elsőre hivatkozik. Ha más témát akarunk érinteni, azt külön lemezre kell fölvenni, vagy a lemezen meg kell cserélni a sorrendet.

Mivel mód van a kérdések és válaszok saját megszerkesztésére, a Quizmaker-program könnyen átalakulhat oktatóprogrammá. A Maker segítségével elkészítjük a megtanulandó anyaghoz illő kérdőívet. A Quizzel azután addig lehet „játszani”, amíg a kérdéses anyagot meg nem tanultuk. Így még a szóbilflázás is okozhat örömet. Külön ajándék ezenkívül az ár is, a Quizmaker 64 csupán 29 márkába kerül.

Az Atari és a maradék világ

Megtudjuk, mire képes az ST, s hogy mit lehet és érdemes vele csinálni.

Aki lemond a színes megjelenítésről, és megelégszik a monokróm képernyővel, igencsak meglepődik az Atari ST használatakor. Az SM 124-es típusjelű monitor ugyanis 70 Hz-es képismétlési frekvenciával rendelkezik, s a 640x400 képpontos grafikákat jóval élesebben képezi le, mint az 1000—2000 márkás árosztályba eső számítógépek. Ez az oka annak, hogy az Atari ST olyan könnyen specializálta magát a CAD alkalmazásokra és a szövegszerkesztésre. Ezekben az alkalmazásokban ez a számítógép a sokkal drágább PC rendszereket is zsebre vágja.

De a számítási sebességet tekintve is fönn hordhatja az orrát ez a rendszer. Vagy egy 68000-es, 32 bites adatbusszal rendelkező, 8 MHz-cel ketyegtetett mikroprocesszor. Ez a gép tehát gyorsabb, mint a 7,44 MHz-es Amiga, és az olcsóbb PC-AT gépek. Összehasonlításképpen a C64 8 bites adatbusszal és közel 1 MHz-es órajel-frekvenciával rendelkezik. Sajnos azonban az ST nagy sebessége szinte alig véteti magát észre.

EGYESÜLETI TAGOK FÓRUMA

AMIGA

Használt alapkiépítésű AMIGA 500-at keresek megvételre. Péter Gábor, 6729 Szeged, Szigony u. 13. Telefon: (62)—29—491

AMIGÁRA több mint 100 rajz, animációs, zene-, video-, adat- és szövegkezelő programot ajánlok. Kérésre listát küldök és bemutatót tartok. Belicki János, 6000 Kecskemét, Muskotály u. 17.

AMIGA programok olcsón eladók. 1 MB-os bővítőt vennék. Lemezek eladók! Vassics Tamás, 8855 Belezna, Kossuth u. 40.

VEGYES

C 64 programokat eladok lemezen. Válaszborítékért listát küldök. Marosvári György 1112 Budapest, Hattár út 51.

VC 1541—II alig használt lemezegységet cserélnék 1551-re vagy 1551—II-re. Keresem PLUS/4-re a JCL programfejlesztő rendszert vagy FASS assemblert. Szabó Balázs, 8491 Karakószöröcsög, Kossuth u. 50.

Keresem kazettára az ACTION SERVICE programot. Cserébe adom a DÉMONOK BIRODALMA és a GHOSTBUSTERS I. játékot. Furmann Dávid, 3529 Miskolc, Oszip István út 1. I/3.

PLUS/4 és C 16

C 64

Eladó bővített C16 + magnó + 300 darab program + szakkönyvek. Árajánlatot kérek. Ugyanitt programcsere illetve eladás. Tornósy Gábor, 2944 Bana, Petőfi u. 9.

Programcsere C64-en! Kb. 100 programom van. Keresem az ELITE program leírását. Cséry Csanád, 7625 Pécs, Németh László út 38. (72)—41—510.

Használt, olcsó, üzemképes, VC1541-gyel kompatibilis floppy diszket vennék C64-hez. Tóth Attila, 1161 Budapest, Pálya u. 80.

C64-es programok kazettán eladók, már 4 forinttól. Közel 900 program 88—89-es programokkal csere is. Cserénél listát kérünk és küldünk. TSG. 2800 Tatabánya 3, Pf.: 3023.

PLUS/4 programcsere lemezen és kazettán! Lemezes programot keresek, listát küldök. Szabó Károly, 6413 Kunfehértó, Ady E. u. 10.

SZUPER II. C64-re! Szuperprogramokkal teli lemezek (DS, DD) eladók, 85 forint/darab. Csak válaszborítékban küldök tájékoztatót. Oláh Lajos, 3014 Hort, Kossuth u. 147.

C16 és PLUS/4 programcsere kazettán. Keresem: A—07, PED, Games Creator. Adok: Tirna Nog, Elite, Aliens, Mercenary I—II. Listát kérek. Biáth Péter 3300 Eger, Donát út 5/A.

Eladók C64 és AMIGA lemezek 100/180 forint/darab. Listát küldök, esetleg cserélek. Németi Ferenc, 1182 Budapest, Nagyegyed u. 8/A.

PLUS/4 programok cseréje. Szöllősi László, 4033 Debrecen, Hodászi Lukács u. 7/A.

Keresek MIDI programokat, interfész kapcsolási rajzot C64-re. Ezenkívül mindenféle szakirodalom érdekel. SÜRGŐS! Laurency Tamás, 2220 Vecsés, Szabadkai út 65.

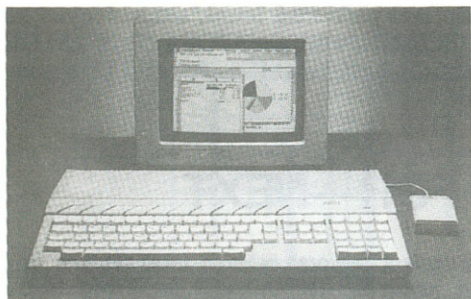
PLUS/4 programokat cserélnék kazettán. Keresem a GAMES CREATOR nevű programot. Zatykó Zsolt, 2151 Fót, Hegyalja út 58/A.

8—16 Kbyte turbokártyák TAPE/FLOPPY programokkal olcsón eladók. Borbás Dénes, 3623 Borsodszentgyörgy, Szentgyörgy út 10.

C64-re, PLUS/4-re játékos felhasználói programokat adok, cserélek. Válaszborítékért listát küldök. Szántai László, 2801 Tatabánya, Pf. 396

Ez az Amigán is megfigyelhető probléma abban gyökerezik, hogy a legtöbb szoftvergyártó a programjait magasabb nyelveken, pl. Pascalban vagy C-ben írja. Ezek pedig még ilyen frekvenciák mellett is lassúbbak, mint a gépi kód (mondjuk akár a C 64-esen!), ráadásul igen sok tárolóterületet emésztenek föl. Ez a gond egyre inkább a PC családot is eléri. Sajnos azt lehet megfigyelni, hogy a gépgyártók egyre gyorsabb számítógépeket készítenek, egyre több RAM-mal, a szoftvergyártók viszont kényelemből egyre lassúbb és a tárolókat szinte zabáló programokat írnak. Időközben ez odáig ment, hogy a gépek operációs rendszerét is magasabb nyelven írják, amelyek a kezdők számára sokkal jobban megnehezítik a beszállást, mint a gépi kód.

Azok az idők, amikor a C 64-esnél az operációs rutinokba közvetlenül belép-



hettünk, elmúltak. Mind az Atarinál, mind az Amiga és a PC-k esetében föl kell tölteni a processzorregisztereket meghatározott értékekkel, amelyre egy adott megszakítást váltunk ki, mire föl hívjuk a kívánt operációs rendszer rutint.

A nevezett számítógépek közül mégis inkább az Atari való kezdőknek. Talán ez ugyanis az egyetlen a modern számítógépek közül, amelyet megfelelő szakirodalom birtokában gyorsan és könnyen megérthetünk, programozhatunk. Ennek oka pedig az egyszerű hardverstruktúra. Mialatt az Amiga multitaskingra (több program „egyszerre” futtatása) is képes, az Atari az egyszerűbb szinteken marad. Így nem kell plusz korlátozásokkal foglalkozni. Hasonló példát találunk a PC-knél is, ahol a különböző hardveres kiegészítők, kártyák miatt az ebből fakadó, a gyártók által rögzített korlátozásokat kell megemésztenuk. A gyártókból azonban, mint ez köztudott, néha harapófogóval sem tudjuk kihúzni az információt. Ha szerencsénk van, a

grafikus kártyához kapunk 2—3 fénymásolt angol leírást...

Az Atari ST tehát tud mit nyújtani a rendszerzsongelben a gépvásárlóknak, méghozzá olyan dolgokat, amelyek másolhol nincsenek meg. Egyébként a vásárló ma az Atariból háromfélét is találhat a boltokban.

A legkisebb az ST családban az 520 STFM. 512 kbyte RAM-ot és az M jel szerint beépített TV-modulátort találunk benne. Így ezt a modellt a televízió antennabemenetére kötve is használhatjuk. Nem utolsó szempont ez a drága monitorral szemben. Persze sokat ettől nem várhatunk, ez a megoldás semmiképp sem konkurenciaképes a monitorral.

A nagyobb testvér, az ST 1040 két kivitelben készül. Mindkettőben találunk egy beépített 3 1/2 colos floppyt, 1 Mbyte munkatárolót. Az STFM-ben találunk modulátort, az STF-ben pedig nem, ez a kettő közti különbség.

Az Atari ST család számára írt alkalmazói szoftverek önmagukért beszélnek. Semmiképpen nem lehet a számtalan játékot és nyomtatóprogramot elfelejteni. Óriási szoftverkinálatot nyit meg a felhasználók előtt a Mac emulátor, amely Aladin vagy Spectre 128 névre hallgathat. Ezzel a tulajdonosok az Apple Macintosh programjait használhatják.

Az Atari ST 1040 3 1/2 inches meghajtóval és 1 Mbyte RAM-mal. Az operációs rendszer a ROM-ban található.

Felhasználói és játéprogramokat cserélek lemezen és kazettán. Listát kérek és küldök. Kocsis Sándor L. 1039 Budapest, Hollós K. L. u. 6.

Keresem a SIMONS BASIC-et CARTRIDGE formában. Árajánlatot erre a címre kérek: Farkas Máttyás, 2755 Kocsér, Szabadság u. 71.

Keresem a BATTLE CHESS két lemezoldalas sakkprogramot. Gyenis Attila, 7396 Magyarszék, Kosuth u. 74.

Eladó PLUS/4 számítógép, magnó, 2 darab joystick, VC 1541-II floppy, 660 darab program. KÜLÖN-KÜLÖN is! Érdeklődni: Budapest XX. Erdélyi u. 23.

Keresem a VIRGIN GAMES FA CUP 87 című játékát. Cserébe leírást, játékokat adok, esetleg listát küldök. Serfőző Tamás, 1014 Budapest, Uri u. 72.

Keresem a FINAL III. CARTRIDGE magyar nyelvű leírását. Pusztai István, 7940 Szentlőrinci út 23/A.

Eprom-égető kártya és vezérlő software olcsón eladó. Válaszboríték ellenében tájékoztatót küldök. Bánvölgyi Sándor, 7632 Pécs, Éva u. 3.

Fényceruza 1250 Ft-ért postai utánvétellel. Joyball C64-hez 1250 Ft. COMPUTEAM, 7400 Kaposvár.

ELADÓ! ZX 81 számítógép + 64 kbyte bővítő + szakirodalom. Osztrics László, 2132 Göd-felső, Rákóczi u. 93.

C 64 programok eladók (10 Ft/db). Listát küldök. Körössy Attila, 4400 Nyíregyháza, Irinyi János út 2/B.

ELADÓ! C128-as garanciával + 1571 floppy drive + 10 PARROT lemez + 2 joy. Ára 40 000 Ft. Török László, 4150 Püspökladány, Arany J. u. 45.

Eladó C 64 + 1541 II drive + 30 darab lemez + 2 darab joy + 64K CARTRIDGE + 15 kötet szakirodalom. Irányár: 40—45 000 Forint. Kapuvári Sándor, 2400 Dunaújváros, Szabadság útja 28. IG/3.

Tippek trükkök



AZ AUTOSTART

A mai profi programok szinte mindegyike rendelkezik egy úgynevezett autostarttal, ami azt jelenti, hogy a program a betöltés után „magától” elindul. Ez nemcsak kényelmi szempontból jó, de egyfajta védelmet is jelent a kíváncsi szemek elől. Hiszen a programot a betöltés után nem lehet csak úgy kilistázni, megtekinteni.

Az ilyen autostart létrehozásához számtalan bonyolult módszer van, amelyeket ezért csak komolyabb assembler-programozói ismeretek birtokában lehet megvalósítani. Pedig van egy egyszerű BASIC-megoldás, ahol csak néhány POKE szükséges. Hívjuk be tehát a preparálandó programot, és adjuk be parancsmódban az alábbi szekvenciát:

```
PRINT "(clr)POKE45,"; PEEK(45);": POKE46,";
PEEK (46);":RUN"
```

Ha most megnyomjuk a <RETURN>-t, úgy a képernyőt töröljük, és oda egy parancsot írunk. Két sorral lejjebb a READY, jelzés áll, alatta a villogó kurzor.

Menjünk a <kurzor le> billentyűvel még két sorral lejjebb, és írjuk be az alábbiakat:

```
POKE 631,19: POKE 632,13: POKE 198,2: POKE 43,198:
POKE 44,0: SAVE "név",8
```

A <RETURN> megnyomását követően az autostart program a lemezre kerül. A kijelzett SYNTAX ERROR-ral ne törődjünk.

Adjuk be most még a POKE 43,1: POKE 44,8 utasításokat, és már készen is vagyunk. A programunkat bármikor betölthetjük a LOAD "név",8,1 utasítással. Nagyon fontos itt a ,1-es másodlagos cím, hogy a tárolóban minden a megfelelő helyre kerüljön. A most bemutatott eljárás az alábbi módon működik. A POKE 43,198: POKE 44,0 hatására a BASIC tárolót a C 64-es 198-as címére állítjuk be. A következő SAVE hatására tehát mindent lemezre mentünk, ami a 198-as cím és a programok vége között áll. Itt fejt ki a módszerünk a hatását. A POKE 631,19: POKE 632,13: POKE 198,2 azt a látszatot kelti a számítógépben, hogy a felhasználó megnyomta a <HOME> és a <RETURN> gombokat. A SAVE során ezt is lemezre mentettük.

Amikor behívjuk abszolút módon, azaz a megfelelő helyre a programot, az elérakott információk is a gépbe kerülnek. A számítógép „beveszi” a csetet, és végrehajtja az utasítást. Mivel a képernyő tartalmát is kimentettük és most visszaolvastuk, a kurzor a POKE 45... POKE 46 sorban áll, és még egy <RETURN>-t hajt végre. Ekkor a BASIC területmutatókat helyreállítjuk, és a programot automatikusan elindítjuk. Ha szeretnénk a programvédelmet perfektté tenni, akkor legyen az első sor a 10 POKE 808,225, ami a <STOP> (és a <STOP/RESTORE> kombinációt) hatástalanítja. Az első program-sor törölje a képernyőt.

MÁSFÉLE FOR—NEXT

A FOR—NEXT hurok a leggyakrabban használatos hurok-szervezési módszer. Most két alkalmazást mutatunk be.

1. Hurkot szoktunk használni függvénytáblázatok készítéséhez: 10 DIM Y(100): FOR X=1 TO 100:Y(X)=3 * X+4: NEXT X

Ha szeretnénk a hurkot megszakítani, például az Y(X) függvény értéke nagyobb lesz, mint 200, vagy ha már mind a száz függvényérték megvan, egyszerűen módosíthatjuk a programot:

```
10 DIM Y(100): FOR X=1 TO 100:Y(X)=3*X+4
35 X=X-(Y[X]-200)*(100-X): NEXT X
```

2. Ezenkívül megnövelhetjük a hurokban a lépéstávolságot:

```
10 SW=1: FOR X=1 TO 100 STEP SW: PRINT X,SW
40 SW=SW*2:X=X+SW: NEXT X
```

Az SW lépéstávolságot minden hurokátfutásnál megketőzzük. Egyébként a fenti 35-ös sorban mutatott trükk egy logikai művelet. Ennek a finomságairól a 89/6 számban külön cikket is olvashatunk.



JÁTÉK

ALIENS (Activision)

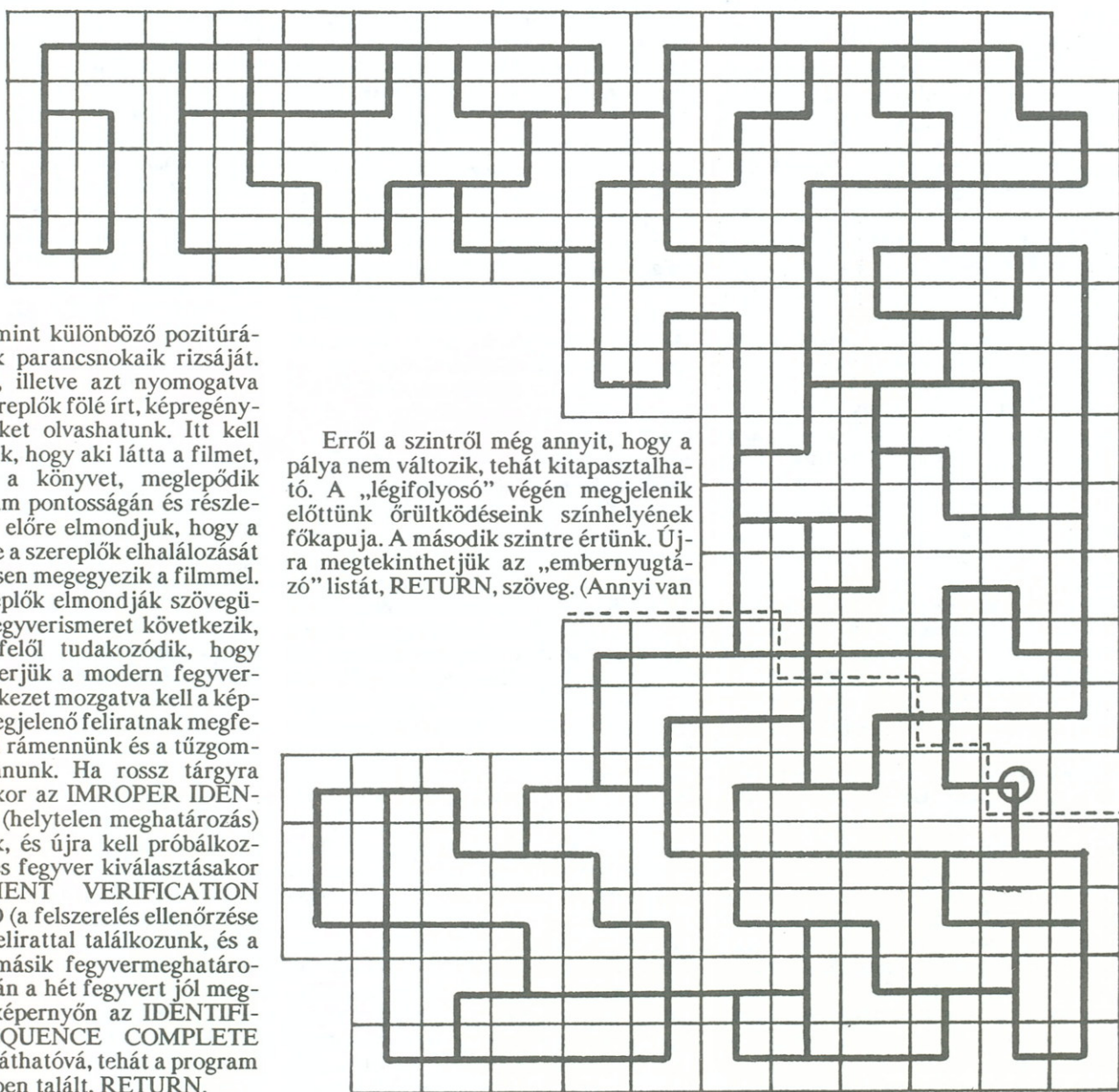
Mikor barátaimat megkérdeztem, hogy hány Aliens-feldolgozást ismernek, mindenki rávágta, hogy Electric Dreams, és kész. Lehet, hogy az Activision feldolgozása nem terjedt el Magyarhonban, lehet persze az is, hogy nem jó játék. De ez nem így van. Reméljük, hogy ezen leírást olvasóknak megváltozik a véleménye erről a játékról. Ha betöltjük a programot, némi villogás kíséretében a Light Circle betörőfiúknak

gyönyörködhetünk, majd egy kevés idő után andalító mormogás közepette megjelenik a forgalmazó neve, majd a szerzők, a rájuk jellemző nagyképűséggel. Írta, rendezte, zeneszerző stb. Ha nem vagyunk kíváncsiak a távolból előtűnő programnévre akkor RETURN (ezek után a RETURN mindig a billentyű megnyomására vonatkozik). Ekkor a program közli nyolc játékosunk, szereplőnk mibenlétét (Ok.: rendben van; Missing: megsemmisült). A szereplők nevei alatt a gép kér egy öt jegyből álló kódot (erre még később visszatérünk), vagy a RETURN lenyomását szorgalmazza. Ha ezt megtettük, akkor egy szöveg válik láthatóvá, elmesélve a nálunk A bolygó neve: Halál néven ismeretes történet előzményeit. Ezek után egy hevenyészett rajzot láthatunk, rajta csapatunk

Újra lista következik, rajta embereink hogyléte. Kapunk egy kódot (a kódkról majd később szólunk). RETURN. Szöveg következik (ezalatt a gép tölt), majd RETURN. Egy úrsíkló orrában találjuk magunkat, előttünk egy vonalakból felépülő célkeresztyszerűséggel, jobb oldalt pedig egy tűréshatárjelzővel. Főül eközben szövegek váltakoznak, biztosítva minket a hajó állapota felől, közben hallhatjuk felbúgni a hajtóműveket. Ezután visszaszámlálás következik és nem is enyhe rázkódás és vibrálás kíséretében elindul a hajónk. Vidám zenei aláfestéssel körök érkeznek egymás után. Ezekben kell nekünk bravúros mozdulatokkal repülnünk. Ha kicsúszunk, némi okítás után újra játszhatjuk a szintet. Nem kell megijedni, ha a szint teljesítése nem sikerül századikra sem, tovább kell próbálkoznunk (meglesz a gyümölcse).

mint a rosseb.) Úgyhogy, RETURN. Egy labirintusrendszerbe kerültünk (nyugi, ehhez van egy csinos kis térkép)! Célunk a csapat közös terepjárójának megtalálása. Négy emberünket kell vezérelnünk, váltakozva. A képernyő alján nyolc kisebb látható, alattunk az F1, F3, F5, F7 felirattal. Ezek a betűk az F funkcióbillentyűket jelentik. Minden egyes betűjelhez két, egymás fölött lévő képernyő tartozik. Az alsó a helyzetünket jelzi: ha a monitor „pontoz”, akkor minden rendben van, nem vagyunk veszélyben, viszont ha pirosan villog, veszélyben vagyunk, a szörnyikék egy csoportja megtámadt minket. Ilyenkor a funkcióbillentyűkkel veszélyeztetett emberünkre kapcsolunk és máris a labirintus egy másik pontja válik láthatóvá, középen szereplőnkkel. Gáz van, harcba kell szállnunk az amúgy sem túl bizalomgerjesztő szörnyekkel. Ez sorozatlövőnkkel meg-

II. pálya



embereivel, amint különböző pozitúrában hallgatják parancsnokaik rizsáját. RETURN-nel, illetve azt nyomogatva más-más a szereplők fölé írt, képregényszerű szövegeket olvashatunk. Itt kell megjegyeznünk, hogy aki látta a filmet, vagy olvasta a könyvet, meglepődik majd a program pontosságán és részletességén. Már előre elmondjuk, hogy a játék — kivéve a szereplők elhalálózását — szinte teljesen megegyezik a filmmel. Szóval a szereplők elmondják szövegüket. Ezután fegyverismeret következik, a program afelől tudakozódik, hogy mennyire ismerjük a modern fegyverkészletet. Egy kezet mozgatva kell a képernyő alján megjelenő feliratnak megfelelő fegyverrel rámenünk és a tűzgombot megnyomnunk. Ha rossz tárgyra mutatunk, akkor az IMPROPER IDENTIFICATION (helytelen meghatározás) választ kapjuk, és újra kell próbálkoznunk. A helyes fegyver kiválasztásakor az EQUIPMENT VERIFICATION CONFIRMED (a felszerelés ellenőrzése jóváhagyva) felirattal találkozunk, és a gép kér egy másik fegyvermeghatározást stb. Miután a hét fegyvert jól megmutattuk, a képernyőn az IDENTIFICATION SEQUENCE COMPLETE mondat válik láthatóvá, tehát a program mindent rendben talált, RETURN.

Erről a szintről még annyit, hogy a pálya nem változik, tehát kitapasztalható. A „légifolyosó” végén megjelenik előttünk örülködéseink színhelyének főkapuja. A második szintre értünk. Újra megtekinthetjük az „embernyugtató” listát, RETURN, szöveg. (Annyi van

oldható. A harchoz kapcsolódik a jobb felső sarokban található mozgásdetektor, ez a műszer jelzi az ellenség támadásának irányát, annak helyét és távolságát, kis pamacs formájában. Ennek a szerkezetnek szinte semmi hasznát nem vehetjük, mert mire bemérjük a szörnyeket, addigra már az orrunkban vannak. Lelkes, kitartó közdelmünk közben vigyázzunk, mert ha a szörnyek közel érnek hozzánk, elkapnak és társas étkezés keretében, öntelt vigyorral elfogyasztanak minket. Ha az összes támadóval végeztünk, akkor képernyőnk újra nyugalmi állapotot vesz fel és végre szabadon mozoghatunk. Az alsó monitoroknál még van három funkciójuk. Ha nem világog egy esetleges támadáskor, hanem már egyfolytában pirosan ég, akkor ez azt jelenti, hogy harcosunkat nyakon csípték és csak akkor tudjuk megmenteni, ha egy másik emberünk belép a veszélyeztetett szobába, ekkor a szörnyikék meguktól eltakarodnak. Zölden világít monitorunk, ha megtaláljuk a terepjárót (de messze van még az!).

Visszatérve a műszerfalhoz, a nyolc kis képernyő közül a felső négy az emberek szívritmusát, EKG-ját akarja jelezni. Aki netán nem ismerné az EKG-vonalakat, annak elmondjuk, hogy ha az EKG-vonal szaggatottabb, akkor ipsékn nyugtalan. Ha már egyenes vonalakat látunk, akkor már löhetjük (mármost az emberünket). Térjünk át most a bal sarokban található képre. Ez az éppen aktuális harcosunkat mutatja a számára kijelölt helyen, a labirintus egy pontján. Hapekunkat a joystickkel tudjuk irányítani. A labirintus megfelelően bonyolult, de nagymértékben megkönnyíti tájékozódásunkat az egyes helyeken a falra ragadó nyálkás váladék, mely trutumó néven fut és jelöli a térkép is. Ugyancsak jelzi a térkép a terepjárót, melynek megtalálása a cél. Néhány jó tanács (poénból). Mivel a négy embert a program nem mindig ugyanoda rakja ki, azért érdemes először betájolni magunkat a térképen, aztán érdemes egymást is megkeresni, ha ugyanis két vagy több ember együtt halad, akkor nem támadnak rájuk. Ha megleltük azt a (...) szép terepjárót, akkor várjunk és a gép kijelzi a visszatértek számát, utána lista, feltüntetve az esetleges veszteséget is. Kódszámot kapunk, RETURN. Szöveg (hm), RETURN. És voila, itt a harmadik szint. Ez is szép hely, biz'a. Ehhez a szinthez talán még több ügyesség kell, mint az előzőhöz (fura mi?). Egy téglalapokkal telerakott térben ábrázolt folyosón vagyunk. Mi a jobb oldalon állunk, rézsutosan (térben) mozoghatunk. Kezünkben drága, jól megbecsült barátunk, egy lángszorító. Ez a szerkezet elég hatékony és sokrétű. Lehet például öngyújtó (ez csak vicc volt). Az a feladatunk, hogy kicsit távolabb várakozó társainkat fe-

dezzük menekülésük közben. A többiek úgy gondolták, hogy jobb, ha menekülnek, ennek érdekében egy embert odaállítottak egy bazi nagy fémajtó elé, hogy ugyan vágja már körbe. Nos, amíg ő szenved, addig nekünk is kell. Bal oldalról felénk szaladnak az előbbi szinten már megismert drágaságok és nekünk a joystick gombjának lelkes nyomkodásával a pokolra kell küldenünk őket. Ez az egész kín addig tart, amíg a képernyő alján villogó lángpötty körbe nem ér az ajtón. Megnyugtatóul közöljük, hogy a pontocska döglassú. A szörnyek pedig egyre gyorsabban futnak felénk. Mi történik, ha valamelyiket nem tudjuk lelőni? Semmi különös, mindössze annyi, hogy a hátunk mögé került példány elvisz egy békésen várakozó embert a mieink közül. Hogy hány emberünk van, azaz hány szörnyet engedhetünk be, az az előző szinten való teljesítményünk függ. Néhány tanács. (Tudom, hogy mindenkinek elege van belőle, de úgy érezzük, hogy ha nem mondhadjuk el, akkor szétrobbanunk, stilszerűen.) Ellenségeink félnek már a lángszorító alaphangjától is, így még ha nem is tudjuk lelőni egyiket-másikat, mert nem értünk oda időben, menjünk csak oda és a szörnyek jó esetben „visszatáncolnak” fegyverünk csöve elé. Célszerű az egyszer megválasztott irányban továbbhaladni, mert így lehet, hogy kevesebb szörny jut mögénk. Ha teljesítettük ezt a pályát is (ebben erősen kételkedünk), akkor ismét előjön a lista, kódszám, RETURN. Összekötőszöveg, RETURN. Egy csúzdarendszer falai közt találjuk magunkat. Feladatunk, az egymást keresztező és egymás fölött és alatt elhaladó „csöveken” át a szörnyeket jelképező világító rombuszok elől menekülni az OPERATIONS ROOM (vezérlőterem)ből, a DROP SHIP (siklóhajó)be. Mi a csőhálózatot fölülről látjuk, nyolc (ha embert veszítettünk, ez a szám a veszteség arányában csökken) harcosunk egyszerre halad. Kis világító rombuszok jelzik a szörnyek helyét és haladásuk irányát. Ha ezek a pontok elkapnak minket, pár hanghatás kíséretében egyik embertársunkat le kell robbantanunk a csoporttól, mert ha ezt nem tesszük meg, a szörnyek többünket is elfogyaszthatnak egy ember helyett. A csatornahálózat elég bonyolult (hát persze, nem is tudom, mi-ben reménykedtünk), térképet mégsem érdemes készíteni a kifelé vezető utakról, mert ezekből több is van, és azért sem, mert útvonalunk a szörnyek támadásától, illetve annak irányától függ. Párt ötlet... Menekülés közben érdemes figyelni az előttünk lévő utat, hogyha elénk akarna jönni egy-egy drágaság, akkor idejében elkanyarodjunk. És még egy. Érdemes minél kacskaringósabban haladni, mert az ellenség gyorsabb, mint mi, ezért egyenes szakaszban jobban ér-

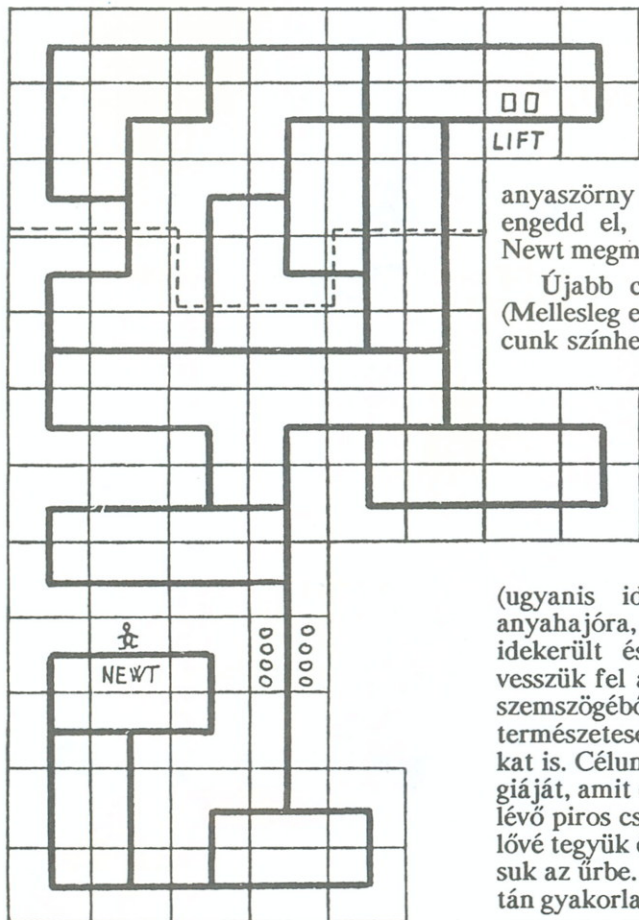
vényesül. Ha teljesítettük ezen küldetésünket is, elébe nézhetünk az immáron ötödik szintnek. Előtte azonban újra tanulmányozhatjuk a személyi listánkat, a kapott kódot, RETURN. Párbeszédet olvashatunk, RETURN, és itt az ötödik pálya. Akik nem látták volna a filmet, vagy nem ismernék a történetet, azoknak elmondjuk, hogy egy szerencsétlen véletlen miatt Newt, a szörnyekkel teli bázison talált egyedüli túlélő kislány, kikerül Ripley, a főhős nő oltalmából és az ellenség kezébe kerül. Ripley megpróbálja még a bázis felrobbantása előtt megmenteni Newt-ot. Ez a célja az ötödik szintnek (lásd: térkép). A hősnőt mi vezetjük egy hasonló labirintusban, mint a második szinten. Mégis nehezebb feladatra vállalkozunk, mint ott, mert itt lövedékünk is fogy és korlátozott idő is áll rendelkezésünkre. Mindenekelőtt kell egy pár szót szólni a tájékoztató „műszerfal”-ról. Ez alatt értendő a képernyő négy ablaka, ami részben alul, részben pedig a jobb felső sarokban van. Az alsó három ablak balról jobbra a következő funkciókat végzi. FLARES: világító lövedék, nyolc darab áll rendelkezésünkre. (Szép a színük!) Ezeket a petárdákat az égvilágon semmire sem tudjuk használni, illetve talán a tájékozódáshoz, de mivel a térkép teljesen pontos (ez természetes!), nincsen semmi haszna a világítógránátoknak, amiket egyébként a spece-vel tehetünk le. A szörnyekben nem tesznek kárt, csak a kanyarokban letéve használhatjuk a visszafelé úton.

AMMO: lövedék, lőszer. Ez a tárunkban lévő lövedékek számát jelzi. Induláskor 99 darab lőszerünk van (biztató, ugye?), ez a szörnyek lövésekor rohamosan fogy. (Na tessék!) Ráadásul takarékoskodni kell, mert a visszafelé vezető úton szükségünk lesz a sok lövedékre.

TIME: idő. Ez a rendelkezésünkre álló időt jelzi percben és másodpercben. Itt kell megemlítenünk, hogy a képernyő alján hosszú ablak található. Itt jelenik meg percenként a figyelmeztetés: XY MIN. TO REACH SAFE DISTANCE. (xy percünk van, hogy eltakarodjunk a fészkes fenébe). Az XY helyére az éppen aktuális percállás kerül.

A jobb sarokban található az utolsó műszer, ami a kislány távolságát jelzi méterben és irányát égtájak szerint. Tehát a műszer így néz ki: XY méter/XY égtáj. Ezzel tisztáztuk a „műszerfal”-at, most a gyakorlati dolgok. Mi (azaz Ripley) egy liftből szállunk ki (jelölve a térképen), hogy megkeressük a kislányt (jelölve). Miután megtaláltuk Newt-ot, vissza kell mennünk a liftbe. A kislány szerencsére engedelmesen követ minket. Amikor viszont már visszafelé bandukolunk, meglepetés ér minket (sajnos nem karácsonyi és még csak nem is mi-

V. pálya



anyaszörny felé, valami ilyesfélét, hogy enged el, te szemét. (Ézt bizonyára Newt megmentése érdekében teszi.)

Újabb cselekvésnek nézünk elébe. (Mellesleg ez a hatodik és utolsó.) Harcunk színhelye a Sulaco űrhajó dokkja

(ugyanis időben visszajutottunk az anyahajóra, de az anyaszörny trükkösen idekerült és most egy rakodógéppel vesszük fel a harcot ellene). Mi Ripley szemszögéből bambásszuk a dolgokat és természetesen mi irányítjuk a gépkarokat is. Célunk az, hogy a királynő energiáját, amit egyébként a képernyő alján lévő piros csík jelez, a semmivel egyenlővé tegyük és aztán barátunkat kihajítsuk az űrbe. Ennek megvalósítása pusztán gyakorlati feladat. A két rakodókart egy, azaz egy darab, bármilyen típusú joystick segítségével tudjuk mozgatni. Egyszerre csak egy kart tudunk használni, éppen azt, amelyik a képernyőn van. Szörnyünk energiáját a minimum alá kell csökkentenünk. Ez legjobban és leggyorsabban úgy érhető el, hogy a királynőt „beszorítjuk” a képernyő egyik oldalára (hogy, ez melyik, az mindenkinek a térlátásától függ), és ott hosszan tartó pofozkodásba kezdünk, vagyis barátunkat ütemesen a falhoz kenjük. Persze ő próbál elmenekülni a szélről, de ez kitarító és precíz munkával megakadályozható. Ha az ügyek folyását jól intézzük, akkor az alsó piros csík fogyni kezd és amikor elfogy, akkor... még messze nincs vége a dolognak. Egyébként érdemes a szörnyet még akkor is verni, amikor már nem látjuk a piros energiacsíkot, mert ezzel is csökken az energiája és később kezd csak el újra növekedni (mert ezt is tudja ám!). Szóval a cél az, hogy a szörnyikét, azt az édeset (mondhatni cukorfalatot) a tűzgomb megnyomásával fogjuk meg a derekánál (itt a szerzők közt is vita van, tudniillik még Alan Dean Foster, a könyv szerzője sem tudja definiálni ezeket a rusnya szörnyeket, így mi sem tudjuk, hogy amit megfogtunk rajta, az a dereka-e, a feje, vagy a nyaka, mármint a szörnynek. (És olyan jól fogjuk, ne is engedjük el). Ha netán nem sikerülne megfognunk és közben ellen-

kulás). Egy igen nagy termetű anyaszörny vár minket a tojásteremben (a térképen leledzik) azzal a nem titkolt szándékkal, hogy ha nekimegyünk (ő sosem támad magától ránk), elfogyasszon. Az ember hajlamos a naivitásra, ezért elkezdünk rálőni. Ezt a dolgot addig lehet csinálni, amíg el nem fogy az összes töltényünk, de az anyaszörnynek semmi baja sem lesz. Ki érti ezt? Mi, a leírás szerzői. Abból kell kiindulnunk, hogy a szörnykirálynő védelmezi az általa tojt rengeteg tojást. Ennek egyenes következménye, hogyha elkezdjük lőni a tojásokat, akkor a királynő elhúzódik a tojások felé, védeni azokat. Ha ezt megteszi, nekünk már csak az a dolgunk, hogy miközben folyamatosan tüzelünk a tojásokra, eközben elosonjunk a főszörny mellett. Ha kiértünk a teremből, álljunk meg és pihenjük ki magunkat. (Már ha erre lehetőség van.) Szerencsére az anyaszörny nem követ minket, ezért ne cselekedjünk fejvesztve. Ha szerencsésen visszaértünk a liftbe, várjunk és a liftajtó szépen becsukódik. (Na, ezzel is megvónánk.) Személyi lista kóddal és RETURN-nel. Töltelékek, rizsák... RETURN. Dokkajtó emelkedik fel és tűrhető grafikájú képet tár szemünk elé. Ripley egy rakodógép karjait lóbálva egy szép, kimívelt mondatot küld az

ségünk energiája megnőtt, akkor újra kell pofoznunk, majd megfognunk. Tehát ha megfogtuk, vigyük őt a képernyő közepén alul elhelyezkedő légszilipajtóhoz. És az ajtó, lássunk csodát, magától kinyílik. Ezután vegyünk könnyes búcsút foglyunktól és eresszük ki őt az űrbe... (Ugye mennyire sajnáljuk.) Bevezetett. Innentől kezdve már csak gyönyörködnünk kell. Újra egy képet láthatunk, rajta Bishoppal, a kettészakadt androiddal, aki éppen a huzattól elröpült kislányt tartja egyik kezével és közben ilyenként és eresszük ki őt a légszilipből éppen előkászálódó, fáradt hősnőnk felé: — Nem is rossz egy embertől...! (Köszönjük szépen, Bishop.) Aztán a kép eltűnik, szöveg jön: Newt és Ripley beszélgetnek. Ezután végre, a játék folyamán most először, egy idegnyugtató, bájos képet láthatunk: Newt és Ripley egy hibernáló készülékben békésen szunyál. Alattuk pedig egy nagybetűs felirat hirdeti: THE END!. Phuuu, hát ezen is túl lennénk,

Most pedig a kódokról, amik a leírás folyamán többször megjelennek. No, szóval ezt az egész hülyeséget lerövidítjük, mégpedig a következő módszerrel. A játék elején van egy hely (utalás a leírás elejére), ahol a program kér egy ötjegyű kódot. Munkára fel és gépeljük be, mondjuk legutóbbi kalandunk alkalmával szervírozott kódot (négy szám és egy betű) és hoppá, máris onnan folytathatjuk, ahol legutóbb abbahagytuk. Így nem kell az eddig teljesített szinteket újra játszani. Mivel a szép leírás szerzői végigjátszották az egész játékot (hogy minek, azt a franc tudja), ezért tudják az összes szint teljes emberszámú kódját. (Nem semmi, mi?) Ezen tudásunkkal szeretnénk a köz javát szolgálni, tehát a kódok: (az első szintre nem kell ugyebár kód!)

- II. szintre: 2727H
- III. szintre: 1506E
- IV. szintre: 5761H
- V. szintre: 0640C
- VI. szintre: 0663F

Ezek a kódok tutifixek. Az Aliens tulajdonképpen jól játszható, jó grafikájú (kmm,mm), idegroncsoló, Activision játék (különös ugye?). A rengeteg Aliensfeldolgozás közül talán a legszerencsésebb, bár az idegmarcangoló szörnyjátékok között nem áll az első helyen. Mégis úgy gondoljuk, hogy aki leül ezzel a programmal játszani, az nem fogja hamar elunni magát. Az is biztos, hogy aki elolvassa ezt a leírást, jobban fogja ismerni a filmet, a programot, a könyvet, a történetet, mint a saját nagyanyját. (Bye!)

Keleti Arthur - Szczy Viktor

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 kedvezménnyel vásárolhatják meg a következő programokat:

A Novotrade 2C-Áruházban 20 %-os kedvezménnyel vásárolhatják meg a következő termékeket:

	RÉGI ÁR	ÚJ ÁR
BOTICELLI C+4-re	1 225,-	980,-
TVC Operációs rendszer	190,-	152,-
TVC Hardware	267,-	213,60
TCV ROM	290,-	232,-

Felhívjuk Kedves Vásárlóink figyelmét, hogy a 2C-Áruházban egyes szakkönyvek és játékprogramok 150,- Ft és 99,- Ft-os áron vásárolhatók.



Júniusi

60 forintos vásárlási utalvány

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

Érvényes: szeptember 30-ig



Júniusi

60 forintos vásárlási utalvány

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

Érvényes: július 31-ig

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

C 16 beépíthető 64 KByte memóriabővítő

1990,- Ft	10%
16-64-es átkapcsoló beépítés munkadíja	150,- Ft
ROMTURBO 16	490,- Ft
együttes megrendelés esetén	770,- Ft
árengedménnyel:	3400,- Ft
	3060,- Ft

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



A Fotoelektronik-Novotrade KFT az alább felsorolt szervizeiben mindenféle szervizszolgáltatás munkadíjából 10% kedvezményt ad az egyesületi tagoknak. Határidő: nincs.

A kedvezményt nyújtó szervizeink:

1053 Budapest, Magyar u. 12-14.	Telefon: 1-173-551
1083 Budapest, Szigony u. 9.	Telefon: 1-343-153
1191 Budapest, Gábor Á. sétány 3.	Telefon: 1-274-763
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
6724 Szeged, Csongrádi sugárút 76.	Telefon: 62-13-377
7624 Pécs, Jurisics M. u. 17.	Telefon: 72-11-812
8000 Székesfehérvár, Széchenyi u. 15/a.	Telefon: 22-12-711
9700 Szombathely, Szalonok u. 31.	Telefon: 94-13-419

Felvevőhelyek:

7400 Kaposvár, Május 1. u. 21.	Telefon: 82-17-503
9024 Győr, Bem J. tér 1.	Telefon: 96-12-802
6000 Kecskemét, Széchenyi tér 1-3.	Telefon: 76-23-720

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



Professzionál

ORSZÁGOS SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÓ HÁLÓZAT



MARKETING — KERESKEDELEM — TELEPÍTÉS — SZERVIZ

Bízva ránk fejlesztési elképzeléseinek megvalósítását!
Bízva ránk számítógépeinek szervizét!

PROFI MUNKÁT VÉGGÜNK!

PROFESSZIONÁL SZERVIZ KFT.
1033 Kaszásdűlő u. 5.
Vevőszolgálat telefon: 1886-101, 1686-230,
1884-356

PROFESSZIONÁL KERESKEDELMI KSZ
1033 Kaszásdűlő u. 5.
Telefon: 1870-348
Kereskedelmi iroda: 1884-358



KOGINFORM
 MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET
 Levélcím: 1325 Újpest 1. Pf.: 159
 Telefon: 189-6142, 169-2989, 160-0611 Telefax: 169-2989, 189-6142



A SZÁMÍTÁSTECHNIKÁVAL TÁMOGATOTT OKTATÁSÉRT!



A KOGINFORM várja azoknak a jelentkezését, akik bármilyen formában hozzá tudnak járulni a számítástechnikával támogatott oktatás sikeréhez. Legyen ez az ember pedagógus, számítástechnikai szakember, diák, vállalkozó, gyermeke jövőjéért aggódó szülő.

Jelentkezhetnek:

- iskolák, amelyek rendelkeznek iskolá-számítógépekkel (oktatóprogramokkal), és be szeretnék építeni az oktatóprogramokat a tantervbe;
- tanárok, programozók, szülők, ... akiknek van (még nem forgalmazott) oktatóprogramjuk;
- pedagógusok, akik szeretnék oktatóprogramokat vagy forgatókönyveket írni, ill. akik ezekhez ötleteket tudnak adni;
- pedagógusok, akiknek van már gyakorlati tapasztalatuk az oktatóprogramok és a számítástechnika iskolai alkalmazásában;
- az S-O-S dolgozat!, Charlie feleltetőrendszer, UNIFEL, Feladatgenerátor és egyéb, ehhez hasonló programokhoz írt feladatsorokat kidolgozók;
- a Szókinasztár szótárprogramhoz különböző idegen nyelvű szótárakat készítő.