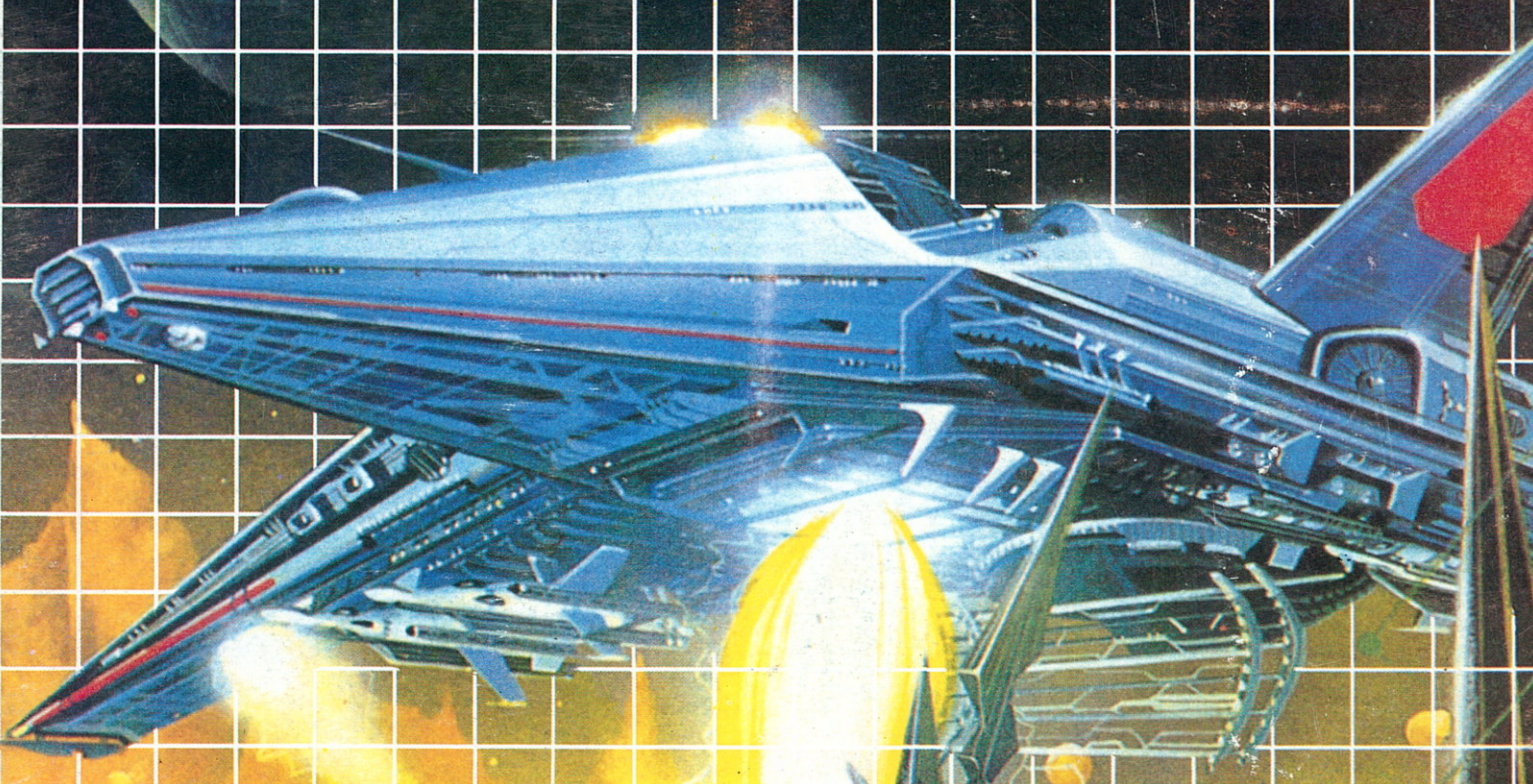


Az Országos Commodore Egyesület lapja

újság

1992/11



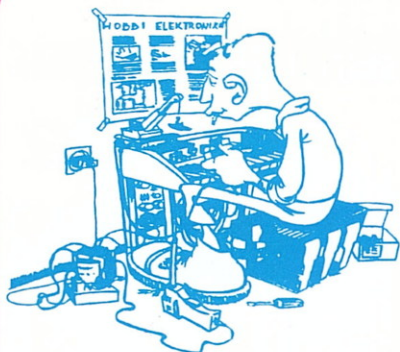
*Távodat-földolgozás*

Fastload kártya

Játék

**TOLOGATÓ**





# HOBBI ELEKTRONIKA

Urbán István mérnök áramköreinek szaküzlete

Budapest VII., Dózsa György út 16. (Dózsa-Jobbágy sarok)

Nyitva: H-P 10-17-ig Tel./fax: 122-8892

(Zárás után üzenetrögzítő)

**Super  
ajánlat!**

Több, mint tíz éve frok a Rádiótechnikában, vezetem a „Zenél hobby elektronika” fejlécű rovatot. A közölt témák nagyrészt szaküzletben - működő minta alapján - vásárolhat panelokat, részegységeket, egységcsomagokat.

## KÍNÁLATUNKBÓL:

### SZÁMÍTÁSTECHNIKA:

C64 bővítők:		egys.	éleszt.
RT89/3	PLOFI Datassette cartridge	1200 Ft	1500 Ft
	PLOFI Datassette/promon	1300 Ft	1700 Ft
	PLOFI Datassette/help	1300 Ft	1700 Ft
RT89/10	PLOFI Fastload cartridge	1300 Ft	1700 Ft
	PLOFI Fastload speedtape	1400 Ft	1900 Ft
RT90/3	PLOFI Simon's cartridge	1300 Ft	1700 Ft
	FINAL III cartridge		3450 Ft
	ACTION Replay VII		3450 Ft
RT89/11	Fényceruza, szoftverrel	950 Ft	1250 Ft
RT90/8	Hangdigitalizáló + szoftver	1350 Ft	1650 Ft
HE90/8	Hangkapcsoló	300 Ft	
RT89/4	EPROM-égető	3400 Ft	4500 Ft
	Égető szoftver lemezen	1300 Ft	1700 Ft
	Égető szoftvercartridge	1300 Ft	1700 Ft
	User csatlakozó	300 Ft	
RT91/3	EPROM-bank (256 Kbájt)	2900 Ft	4000 Ft
RT91/10	IC tesztter	3000 Ft	4500 Ft
RT91/7	Datassette gyorsmásoló	600 Ft	
HE91/7	Datassette fejbéállító	350 Ft	
HE90/12	CPU stop + reset	400 Ft	
	PAGEFOX szövegszerkesztő		3900 Ft
	Mini EPROM-bank	2300 Ft	3000 Ft

IBM bővítők:		egys.	éleszt.
RT91/4,5	IBM IC tesztter + szoftver	5900 Ft	9800 Ft
RT91/11,12	IBM EPROM-égető + szoftver	4800 Ft	9900 Ft
RT91/6	48 csat. VO kártya		3500 Ft

HE92/2	User-Centronics csatlakozó	950 Ft
RT92/3	TTL IC katalógus lemezen	600 Ft
	CMOS IC katalógus lemezen	600 Ft
	Dióda katalógus lemezen	600 Ft
	Tranzisztor katalógus lemezen	600 Ft

### ZENE - HANGTECHNIKA:

RT87/10	Fuzz-box torzító	670 Ft
RT90/1	KORG DST-1 torzító	1200 Ft
	KORG DST-3 torzító	1400 Ft
HE90/1	KORG OVERDRIVE torzító	920 Ft
RT87/12	Vau-vau gitáreffekt	640 Ft
RT88/9	Sztereó tremoló	980 Ft
RT88/7	Shifter	1200 Ft
HE90/7	Kiszajú gitárerősítő	700 Ft
RT89/5	Sztereó előerősítő	1500 Ft
HE91/12	Hangfrekvenciás erősítő	490 Ft
RT89/8	2 x 14 W sztereó erősítő	990 Ft
RT89/12	2 x 40 W sztereó erősítő	1800 Ft
RT89/1	100 W-os erősítő	1600 Ft
RT90/3	Rítusgép 32 rítussal	2500 Ft
RT90/10	Dobszintetizátor	3500 Ft
RT88/3	Süvöltő gitáreffekt	780 Ft
RT87/11	Ringmodulátor	995 Ft
RT92/4	Oktávemelő	700 Ft
	Visszhangosító MN3005-tel	4800 Ft

### VIDEÓS TÉMÁK:

RT92/6	RGB generátor	1760 Ft
RT92/11	PAL kóder	1400 Ft

### FÉNYTECHNIKA:

RT88/10	Diszkófény II (triak nélkül)	1400 Ft
	Diszkófény II (triakkal)	3000 Ft
RT88/12	Fényorgona	2000 Ft
RT90/2	Programozható futófény	1200 Ft
RT91/1	8 x 8-as fénymátrix + 16 K EPROM	2000 Ft
	8 x 8-hoz LED panel 10 mm-es LED-ekkel	2000 Ft
RT89/6	Kivezérlésjelző	700 Ft
RT91/8	Sziporkázó színgyűrű	650 Ft
	Kétszínű LED	48 Ft
HE91/10	Karácsonyi fényjáték	700 Ft
HE92/8	Knight Rider futófény	950 Ft

### EGYÉB HOBBI TÉMÁK:

RT88/10	Dallamgenerátor Z80-nal	1500 Ft
HE91/1	Dallamgenerátor UM3481-84	800 Ft
HE91/5	Dallamgenerátor UM66Txx	480 Ft
HE91/4	Beszédfelemelő VCP200-zal	750 Ft
	VCP200	1950 Ft
HE91/2	Kutyaugatás szintetizátor	800 Ft
RT91/8	Eb-vezérlő	400 Ft
HE91/6	Szűnyögriasztó	700 Ft
HE91/9	Multiszíreña	480 Ft

**Az árak a 25% ÁFA-t tartalmazzák!**

Választékunk folyamatosan bővül! A Rádiótechnikában, a Hobby Elektronikában megjelent nyák-tervek alapján készült nyomtatott áramköri lapok az üzletben megvásárolhatók.

### MŰSZERVÁSÁR

Bontott anyagok,  
használt műszerek  
bongészédje



Állandóan változó készlet!

Vidéki olvasóknak segít a szerző levelező, egységcsomagküldő szolgáltatása: a megrendelt csomagot postán utánvétellel elküldöm. Telefonon/faxon és levélben is rendelhet.

A HOBBI ELEKTRONIKÁ-hoz nem kell hosszú levél. Rendelését néhány sorban, egyértelműen közölje.

**Levélcím: 1656 Budapest Pf. 50.**

Az üzletben megvásárolhatók az RT évkönyvei, a Rádiótechnika és a Hobby Elektronika korábbi számai.



SZERENCSEJÁTEK RT



## MIT, HOGYAN, HOL, MIKOR?

**EGYESÜLETI ÜGYEK:** Egyesületünknek tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az egyesület irodájában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén szám-lát küldünk.

Pötyögőszolgálatunk valamint a szervizkedvezmény és az apróhirdetés lehetősége tagjaink rendelkezésére áll.

A **DEÁKPÁHOLY** tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy évre 777 forint.

A **PLUSZPÁHOLY** tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 1888 Ft.

A **SZUPERPÁHOLY** tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15x3 db vásárlási utalványt is. Az éves tagsági díj 20 900 Ft.

**ÜGYFÉLFOGADÁS:** Minden kedden és csütörtökön 12–16 óra között várjuk tagjainkat és az érdeklődőket.

**PÖTYÖGŐSZOLGÁLAT:** Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az egyesület irodájában vagy postai utánvétellel. Postacím: 1388 Budapest 62., Postafiók: 86.

**APRÓHIRDETÉS:** Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 100 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 1-76-22-57-es telefonszámon vagy levélben!

Vidéki pluszpáholy-tagjaink háromhavi tikett összegyűjtésekor igénybe vehetik a NOVOTRADE 2C Áruház csomagküldő szolgálatát.

### VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:

Baja, AXIS Kft.,  
Győri Bartók Béla Művelődési Ház,  
Jászberényi Városi Könyvtár,  
Kecskemét, SZIGMA—BIT,  
Pécsi Apáczai Csere János Gimnázium,  
Zalaegerszegi Ságvári Endre Gimnázium.

## Tagdíjbefizetés

*Kedves Tagtársaink!*

*Az év vége közeledtével felhívjuk figyelmüket az 1993. évi tagdíjuk befizetésére. Bár nem szeretnénk, de lehet, hogy kénytelenek leszünk emelni a tagdíjakat.*

*A tavalyihoz hasonlóan azonban az idén is lehetőséget adunk arra, hogy aki még 1992-ben befizeti a jövő évi tagdíját, az mindenképpen a régi áron teheti.*

*(Deákpáholy 777 Ft, pluszpáholy 1888 Ft, szuperpáholy 20 900 Ft.)*

*Régi tagjaink a csekket az újsággal együtt kapják meg, de elküldhetik a tagdíjat rózsaszínű postai utalványon is OTP számlaszámunkra. (MNB 217-98 292, OTP 565-3610-8)*

Országos Commodore Egyesület

## Újra indul

a Commodore Egyesület klubdelelőttje a Havana Közösségi Házban

(1181 Budapest, Kondor Béla sétány 8.).

Megközelíthető: a metró Kőbánya-Kispest-i végállomásától a piros 135-ös busszal.

Várunk benneteket minden hónap 3. vasárnap délelőtt 9 órától.

Legközelebb:

**november 15-én és  
december 20-án.**

## FELHÍVÁS

Tisztelt Tagtársunk, Olvasóink!

Kérjük, hogy ha rendelkeznek olyan működő számítógéppel, vagy jó állapotban lévő bármilyen számítógéptartozékkal amire nincs szükségük, akkor hozzák el egyesületi irodánkba. (Minden héten kedden és csütörtökön 11-15 óra között.)

Mi összegyűjtjük és eljuttatjuk határainkon túli (erdélyi, kárpátaljai és szlovákiai) magyar iskolákhoz.

Köszönettel:

Országos Commodore Egyesület

## Tisztelt Szerkesztőség

A lap nagyon tetszik, már eddig is sok hasznos dolgot találtam benne (91. júl. óta ismerem). A 92. ápr. számban „Adataink élettere” c. cikk keltette fel érdeklődésemet, a „Gépi kód...” c. sorozat mellett.

Van azonban egy sajnálatos értelem-zavaró hiba a cikkben. A 17. oldal utolsó 5 sora nem igaz!

Ha ugyanis „egy már létező szekvenciális állományt” írásra akarok megnyitni, pl.:

10 open 1,8,2, "teszt,s,w"

akkor a gép 63, FILE EXIST,00,00 hibajelzéssel visszautasítja az értelmetlen parancsot.

Természetesen van lehetőség a felülírásra:

10 open 1,8,2, "@: teszt,s,w"

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:

1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57

Felelős kiadó: Horváth Judit, az egyesület elnöke

Főszerkesztő: Rados Péter, az OCE főtítkára

Felelős szerkesztő: dr. Horváth András

Művészeti szerkesztő: Bausz Sándor

Levél cím: Commodore Újság, 1388 Budapest, 62. Pf.: 86.

Index: ISSN 0237-756 X

Terjeszti a Magyar Posta

Megvásárolható a hírlapárusoknál

92.0305 MSZH Nyomda és Kiadó Kft., Budapest

Felelős vezető: Nagy László



Ebben az esetben, de csak ebben az esetben megtörénik mindaz, ami a cikk idevonatkozó részében írva vagyon. Egyébként a DATA BECKER — NOVOTRADE kiadásában 1986-ban megjelent English—Szczepanowski páros által írt A VC-1541-es lemezegység programozása c. kiváló munkában ez IS megtalálható (31. oldal).

Üdvözlettel:

Salamon Zsigmond

*Amennyiben úgy értelmezte az említett problémát, hogy egy OPEN 1,8,2,"TESZT,S,W": file-t megnyit, majd lezárja (CLOSE 1) és újra meg akarja nyitni felülírásra a következő módon: OPEN 1,8,2,"TESZT,S,W" valóban az Ön által írt hibauzenet jelenik meg. Valóban, ha egy file-t felül akarunk írni azt előtte törölni kell.*

Kovács Mihály

## Tisztelt Szerkesztőség!

Ötletemmel azért fordulok önökhöz, mert nem szeretném, hogy sok printer tulajdonos rossz hibamegállapítás után, az írófej cseréjéhez folyamodna (ami nem kevés összeg).

A jelenség majdnem ugyanaz, mintha az írófej, vagy a hozzá csatlakozó vezeték hibásodtak volna meg (csak pontokat és vonalakat ír). Ez akkor szokott előfordulni, ha a printert sokáig nem használjuk. A festék rászárad a tűkre és nehezen vagy egyáltalán nem mozognak. A Kis Program begépelése és elindítása után az összes tű üzemmódotba kerül, és pár sor írása után már mind dolgozni fog. A sorok számát a 60 számú sorban lehet növelni, vagy csökkenteni.

Tisztelettel:

Stierbach Mihály

```

10 Rem Printer teszt. C-64/MPS 803
20 RV#=CHR$(10)
30 A#=""
40 :
50 OPEN 4,4
60 FOR I=1 TO 10
70 PRINT#4,RV#,A#,RV#,A#,RV#,A#,RV#,A#,RV#,A#,RV#,A#
80 NEXT I
90 :
100 FOR J=1 TO 1 : PRINT#4 : NEXT J
110 CLOSE 4
    
```

## Számkitalálós

### Tisztelt Szerkesztőség!

Két éve vagyok rendszeres olvasója lapjuknak, innen merítettem „bátorságomat” arra, hogy beküldjem Önöknek eme kis programomat. Én magam kb. négy évvel ezelőtt kerültem közelebbi kapcsolatba a számítógéppel, munkahelyemen. Azóta már saját 64-esemet bővölöm. Munkámmal kapcsolatos programokat is írok már, egyelőre még BASIC-ben, de időm javarészt a felhasználói része köti le. Az alább közölni óhajtott programomnak az alapja egy ismert játék (MOO), amelynek a lényege a következő.

Négy számot rejt el a számítógép (pontosabban egy négyjegyű számot), ezt a számot kell kitalálni minél kevesebb kísérletre. A gép a válaszait aszerint adja, hogy hány szám jó (amelyik a helyén van) és emellett még hány szám jó, ami nincs a helyén. A hely jó jelzése \*, míg a szám jó jelzést üres kör (○) jelzi, mindkét jel előtt egy szám, amely megmondja az egyes találatok számát. A program minden esetben különböző számokat dug el, üres hely sem lehet. A tipp bevitele sem fogad el „üres” helyet (SPACE), ellenben egyforma számot igen. A program számolja a játékok számát és minden játék végén kiszámolja a találati átlagot.

Az általam közölt lista csak egy programtörzs. Természetesen ezt még bőven lehet cicomázni, bővíteni, ez már egyéni ízlés dolga. Lehet, hogy algoritmust is lehet elvileg másat kiagyalni — ez egy a sok közül. Azonban kérem tisztelt számítógéppel fertőzött társaimat: aki megoldja azt, hogy egyforma számokat is el lehessen dugni, ne tartsa magában! Roppant kíváncsi vagyok rá!

64-es üdvözlettel:

Barna Péter

Barna Péter problémájának megoldása céljából először fejtjük meg programlistáját.

A 4 elemű tömböket a program elején hagyjuk későbbre. J és TM szerepe a 620—640 sorokból derül ki: a játékot és a találatások számát tárolják.

A számkiválasztás (190—320 sorok) az RND függvény segítségével történik. A négyelemű B tömb elemei 0 és 9 közötti egészek lehetnek, de a számjegyek ismétlődését szerzőnk kiküszöbölte, hiszen éppen ezzel nem birkózott meg. M\$ karakterlánc formájában tartalmazza az elrejtett számot. A tippet szintén 4 hosszúságú karakterláncként K\$-ba olvassa be a program, majd ellenőrzi, valóban számjegyeket írtunk-e be. A 410-es sor



```

0      100 REM*****
0      110 REM  SZAMKITALALOS
0      120 REM      MOD V2.1
0      130 REM  BARNA-SOFT 1991.JAN
0      140 REM*****
0      150 DIMM$(4),K$(4),K(4),H(4),B(4)
0      160 J=0:TM=0
0      170 J=J+1
0      180 PRINT"J"
0      190 REM***SZAMKIVALASZTAS***
0      200 M$=""
0      210 FORI=1TO4
0      220 :B(I)=INT(RND(0)*10):IFB(I)>9THEN220
0      230 :C=0
0      240 :IFI=1THEN290
0      250 :FORG=1TOI-1
0      260 : IFB(G)=B(I)THENC=1
0      270 : NEXTG
0      280 :IFC=1THEN220
0      290 :B#=RIGHT$(STR$(B(I)),1)
0      300 :M#=M#+B#
0      310 NEXTI
0      320 PRINT"#####VALASZTOTTAM SZAMOKAT";TAB(30)"JATEK:";J;"#
0      330 REM***TIPPELES***
0      340 T=0
0      350 T=T+1
0      360 PRINT"##";T;TAB(5);"/ TIPP"
0      370 POKE19,1:INPUT"#####";K#:POKE19,0:PRINTCHR$(13)
0      380 IFLNK#<>4THENPRINT"J":GOTO370
0      390 FORK=1TO4
0      400 :IFMID$(K#,K,1)<"0"ORMID$(K#,K,1)>"9"THENPRINT"J":GOTO370
0      410 :IFMID$(K#,K,1)=CHR$(32)THENPRINT"J":GOTO370
0      420 NEXTK
0      430 REM***SZAMOK OSSZEHASONLITASA***
0      440 KS=0:KH=0
0      450 FORD=1TO4:K(D)=0:H(D)=0:NEXTD
0      460 FORK=1TO4
0      470 :K$(K)=MID$(K#,K,1)
0      480 FORD=1TO4
0      490 :M$(D)=MID$(M#,D,1)
0      500 :IFK(D)=1THEN540
0      510 :IFK$(K)<>M$(D)THEN570
0      520 :K(D)=1
0      530 :KS=KS+K(D)
0      540 :IFK=DANDK$(K)=M$(D)ANDH(D)=0THENH(D)=1:GOTO560
0      550 :GOTO570
0      560 :KH=KH+H(D)
0      570 NEXTD,K
0      580 V=KS-KH:F=KH
0      590 REM***EREDMENYEK***
0      600 PRINT"J";TAB(20);F;"##";CHR$(42);"###";V;"##";CHR$(119)
0      610 IFF<>4THEN350
0      620 PRINT"#####K.";T;"KISERLET ALATT TALALTAD KI !!!"
0      630 TM=TM+T
0      640 SZ=TM/J:SA=(INT(SZ*100+.5))/100
0      650 PRINT"#####MEGEGYSZER ? (I/N)";TAB(25)"ATLAG:";SA
0      660 WAIT198,1:GETV#:IFV#<>"I"ANDV#<>"N"THEN660
0      670 IFV#="I"THEN170
0      680 PRINT"#####KOSZI A JATEKOT !"

0      READY.
    
```



felesleges, az előző sor minden karakterkódot visszadob, a „0” és „9” közöttiek kivételével.

A program lényegi része az összehasonlítás a 430—580 sorokban található. 440-ben lenullázza a „jó hely” (KH) és „jó szám” (KS) találatok számát. Mint látni fogjuk, a négyelemű K() és H() tömbök d-ik eleme akkor egyes, ha a dugott szám d-ik jegyét K\$ valamelyik jegye eltalálta —K(D)—, illetve a helyén találta el —H(D). E két tömb nullázása a 450-es sor. Ezután két egymásba ágyazott ciklus következik. A külső, K ciklusváltozóval a tipp jegyein fut végig és K\$( ) megfelelő elemeibe teszi azokat. A d ciklusváltozóval a belső ciklus a dugott szám jegyeit vizsgálja. A számjegyek M\$(D)-be kerülnek. Kicsit zavaró a K\$, M\$ karakterláncok és az ugyanilyen nevű K\$( ), M\$( ) karaktertömbök együttes használata. Az 500-as sorban a program elágazik aszerint, hogy a dugott szám d-ik jegyére adtunk-e már pontot (K(D) = 1).

Ha nem, és a tipp aktuális jegye nem egyezik meg az elrejtett szám d-ik jegyével, nincs találat, folytatjuk az összehasonlítást. Ha a két számjegyet megegyezik, a dugott szám d-ik jegyéért pontot adunk (K(D) = 1) és a „jó szám” találatok számát (KS) növeljük. Ezután még meg kell vizsgálni, nincs-e egyben „jó hely” találat is. Ha van, akkor a d-ik jegyre „jó hely” találatot is adunk (H(D) = 1) és növeljük a „jó hely” találatok számát. Tehát KS-ben valójában a kétféle találat összes száma gyűlik, míg a KH-ban a „jó hely” találatoké.

Szerzőnk észrevette, hogy ha a jegyek összehasonlításakor K(D) már 1, ez még nem jelenti azt, hogy továbbmehetünk, mondván erre a jegyre már adtunk találatot. Ugyanis előfordulhat, hogy a tipp egy korábbi jegye megegyezett az eldugott szám d-ik jegyével — így lett K(D) 1 — a tipp aktuális jegye viszont helyben is és értékben is megegyezik M\$(D)-vel, ezért szükségünk van az 540-es sor feltételes utasítására és ha a feltétel teljesül, meg kell adni a „jó hely” találatot is. Ebben a sorban a három AND kapcsolatba hozott feltétele közül a második és harmadik felesleges. Ha a K\$(K) és M\$(D) nem azonos, nem jutunk 540-re 510 miatt. H(D) pedig sehol másutt nem kap értéket a kettős ciklusban így értéke biztos 0.

Az összehasonlítás befejeztével a „jó hely” találatokat KH-ból F-be, KS és KH különbségét — az igazi „jó szám” találatok számát — V-be írjuk át és kijelöljük.

Miért nem jó ez az algoritmus? Próbáljuk pontosan megfogalmazni a találatjelzés szabályát. Adott két négyjegyű szám: a gép által választott (eldugott, kitalálendő) és a tipp. Ha a két szám azonos helyiértéken ugyanaz a számjegy áll, „jó hely” találat jár érte. Ha tehát a két szám 1324 és 2341, egy „jó hely” találat jár, mert mindkét számban a második helyen hármas áll. A „jó szám” találatok szerzőnk szerint azt mutatják, „hány szám jó, ami nincs a helyén”. Hogy is van ez? Ha a választott szám 1234, a tipp 5111, akkor van három egyesünk, ami jó, csak nincs a he-

lyén. Akkor ezért három találat jár? Nem bizony. Egy rejtett számjegyet eltalálásért csak egy találatot adunk. Ez persze fordítva is igaz. Ha a választott szám 5111 és a tipp 1234, a tipp egyesével a rejtett számnak csak egy egyesét találhatjuk el. A nem jó helyen álló számjegyekre tehát az a szabály, hogy ha az egyik számban A darab, a másik számban B darab van egy számjegyből, akkor ezéért min(A,B) találat jár, ahol min(A,B) a két szám közül a kisebbiket jelöli.

Példánk első esetére — rejtett = 1234, tipp = 5111 — a program helyesen működik, hiszen a külső ciklusban a tipp első egyeséhez érve a belső ciklus K(1)-et egyre állítja, így a külső ciklus további körében a belső ciklus D = 1-re már elkerüli az 510—530 sorokat. A fordított esetben — rejtett = 5111, tipp = 1234 — viszont baj van! A külső ciklus első körében a belső ciklus K(2)-t, K(3)-t és K(4)-et egyre állítja és KS-t háromszor növeli. Megvan a hiba! K(D) 1-re állítása azt megakadályozza, hogy a rejtett szám egy jegyét a tipp több jegyével „eltaláljuk”, de az ellen semmi sem véd, hogy a tipp egy jegyével a rejtett szám több jegyéért is találatot kapjunk.

Javítsuk ki a programot! A 230—280 sorokat elhagyhatjuk, hiszen most már nem gond a jegyek ismétlődése az elrejtett számban. A javasolt algoritmus külön kezeli a „jó hely” és „jó szám” találatokat. A K() tömb elemei most azt jelölik, hogy a tipp valamelyik jegyére adunk találatot, a H() elemei pedig azt, hogy a választott szám valamelyik jegyére. A „jó hely” találatot H()-ban és K()-ban kettős, a „jó számot” egyes jelöli.

Az új lista 450-es sorában lenullázzuk a K() és H() tömböket, valamint feltöltjük a K\$( ) és M\$( ) karaktertömböket. (Az eredeti program ez utóbbit feleslegesen négyszer csinálta végig.)

A 460—475 ciklus a „jó hely” találatokat keresi és állítja K() és H() megfelelő elemeit 2-re. A „jó szám” találatokat most is kettős ciklussal keressük. Ha K() aktuális eleme nem 0, ez azt jelenti, hogy a tippnek erre a jegyére már „jó hely” találatot adtunk, így a K ciklussal továbbléphetünk (477-es sor). A belső ciklusban hasonló a helyzet. Ha a rejtett szám D-ik jegyére már adtunk találatot, áttérhetünk a következő jegyre (500-as sor). Természetesen akkor is továbbléphetünk, ha a két éppen összehasonlított jegy nem egyezik (510). Ha van találat, H() és K() megfelelő elemét 1-re állítjuk, és a belső ciklust befejezhetjük, hiszen a tipp aktuális jegyének és a dugott szám további jegyeinek esetleges egyezésére már nem adhatunk találatot.

A kettős ciklus után a „jó hely” találatokat a K()-ban lévő kettősök, a „jó szám” találatokat az egyesek összeszámlálásával kapjuk meg.

Akinek megtetszett a téma, megírhatja a program másik felét, vagyis azt, mikor a gép találja ki az általunk elrejtett számokat.

Horváth András

○	100 REM*****	○
	110 REM SZAMKITALALOS	
○	120 REM	○
	130 REM JAVITOTT VALTOZAT	
○	140 REM*****	○
	150 DIMM\$(4),K\$(4),K(4),H(4),B(4)	
○	160 J=0:TM=0	○
	170 J=J+1	
○	180 PRINT"J	○
	190 REM***SZAMKIVALASZTAS***	
○	200 M\$=""	○
	210 FORI=1TO4	○



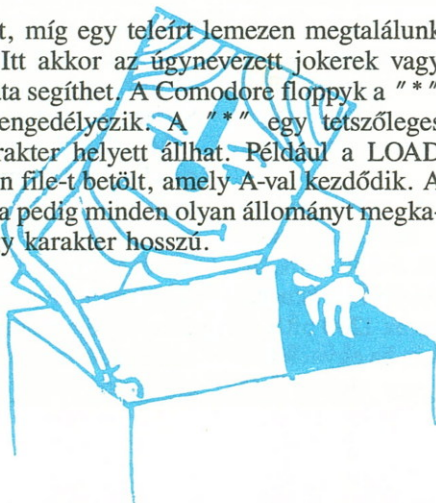
```

○ 220 :B(I)=INT(RND(0)*10):IFB(I)>9THEN220
230 :B#=RIGHT$(STR$(B(I)),1)
○ 300 :M#=M#+B#
310 NEXTI
○ 320 PRINT"#####VALASZTOTTAM SZAMOKAT";TAB(30)"JATEK:";J;"#
330 REM***TIPPELES***
340 T=0
○ 350 T=T+1
360 PRINT"#####";T;TAB(5);"/ TIPP"
○ 370 POKE19,1:INPUT"#####";K#:POKE19,0:PRINTCHR$(13)
380 IFLEN(K#)>4THENPRINT"TT":GOTO370
390 FORK=1TO4
○ 400 :IFMID$(K#,K,1)<"0"ORMID$(K#,K,1)>"9"THENPRINT"TT":GOTO370
420 NEXTK
○ 430 REM***SZAMOK OSSZEHASONLITASA***
440 KS=0:KH=0
○ 450 FORD=1TO4:K(D)=0:H(D)=0:M$(D)=MID$(M#,D,1):K$(D)=MID$(K#,D,1):NEXTD
460 FORK=1TO4
○ 470 :IF K$(K)=M$(K) THEN K(K)=2:H(K)=2
475 NEXT K
○ 476 FOR K=1 TO 4
477 :IF K(K)<>0 THEN 575
480 FORD=1TO4
○ 500 :IFH(D)<>0THEN570
510 :IFK$(K)<>M$(D)THEN570
○ 520 :H(D)=1:K(K)=1:D=4
570 NEXT D
575 NEXT K
○ 580 F=0:V=0:FOR K=1 TO 4
581 IF K(K)=2 THEN F=F+1
582 IF K(K)=1 THEN V=V+1
583 NEXT K
○ 590 REM***EREDMENYEK***
600 PRINT"TT";TAB(20);F;"#";CHR$(42);"###";V;"#";CHR$(119)
○ 610 IFF<>4THEN350
620 PRINT"#####OK.";T;"KISERLET ALATT TALALTAD KI !!"
630 TM=TM+T
○ 640 SZ=TM/J:SA=(INT(SZ*100+.5))/100
650 PRINT"#####MEGEGYSZER ? (I/N)";TAB(25)"ATLAG:";SA
○ 660 WAIT198,1:GETV#:IFV#<>"I"ANDV#<>"N"THEN660
670 IFV#="I"THEN170
○ 680 PRINT"#####KOSZI A JATEKOT !"

READY.
    
```

## Aki keres az talál

Sokszor sokáig tart, míg egy teleírt lemezen megtalálunk egy kívánt állományt. Itt akkor az úgynevezett jokerek vagy "Wildcardok" használata segíthet. A Comodore floppyk a "\*" és a "?" használataát engedélyezik. A "\*" egy tetszőleges sztring, a "?" egy karakter helyett állhat. Például a LOAD "\$:A\*",8 minden olyan file-t betölt, amely A-val kezdődik. A LOAD "\$:?" ,8 hatására pedig minden olyan állományt megkapunk, amelyek neve egy karakter hosszú.

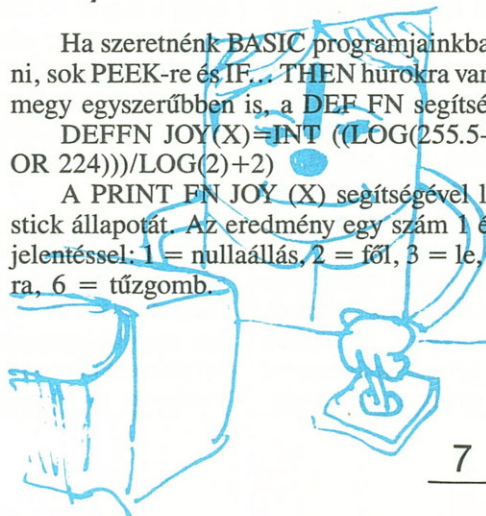


## Joystick lekérdezés

Ha szeretnénk BASIC programjainkban joysticket használni, sok PEEK-re és IF... THEN hurokra van szükség. De a dolog megy egyszerűbben is, a DEF FN segítségével.

```
DEFN JOY(X)=INT ((LOG(255.5-(PEEK(56322-X)
OR 224)))/LOG(2)+2)
```

A PRINT FN JOY (X) segítségével lekérdezhetjük a joystick állapotát. Az eredmény egy szám 1 és 6 között, az alábbi jelentéssel: 1 = nullaállítás, 2 = föl, 3 = le, 4 = balra, 5 = jobbra, 6 = tűzgomb.





# Mi az a távadatfeldolgozás?

Lehet hogy már hallottuk ezeket a fogalmakat: TAF, Baud érték, modem, akusztikus csatoló, RS232C illesztés és hasonlók. De hogy mi is az a TAF, talán ma sem értjük. Nos a cikk megmutatja mi ez, mire is jó és mit lehet vele kezdeni.

Az talán mindjárt nyilvánvaló, hogy a felsorolásban mindazokat a fogalmakat igyekeztünk megadni, amelyek a távadatfeldolgozással, röviden TAF-fel, tehát a számítógépes távközléssel állnak valamilyen kapcsolatban. Maga a TAF fogalom nem egy új területet takar, a számítógépes távközléssel mindennap találkozhatunk, legyen az műholdas időjárási felvételek továbbítása, vagy tengerentúli telefonforgalom. Mindezt a mi 64-esünk is tudja, természetesen a megfelelő közegben, méretekben és minőségben.

Nézzünk mindjárt egy gyakorlati példát. Mondjuk Szegeden lakunk és éppen egy budapesti barátunkkal beszélgetünk telefonon. Ő éppen a legújabb hiperszuper játékprogramját ecseteli. Milyen jó is volna nekünk is kipróbálni! Erre megoldás az, hogy a barát valamilyen adathordozóra másolja a programot, becsomagolja, elviszi a postára, földadja. Mi pedig lessük a postást és imádkozunk.

Mennyivel egyszerűbben lehetne megoldani a dolgot, ha az anyagot mindjárt a beszélgetés alatt át lehetne adni. Erre pedig van is lehetőség, mivel a számítógépünk képes arra, hogy nagy távolságra a programokat, adatokat, szövegeket juttasson el, még hozzá éppen a telefonhálózaton keresztül. Ez az, amit mi a TAF rövidítéssel illetünk.

Persze most mindjárt adódik is a kérdés, hogyan lehet a számítógépet a telefonvonalra csatlakoztatni. Hisz a számítógép se beszélni nem tud, se a beszédet nem érti meg. A telefonra való csatlakoztatást sem tudjuk helyben megoldani. Sőt talán nem is lehet ezt megtenni. Mit tegyünk hát?

A rejtvény megoldását akusztikus csatolónak hívják. Amint az a fogalomban is benne van, valamiféle akusztikus, tehát hangátviteli dolog van benne, a megcsatlakoztatás. Már meg is van a magyarázat. Egy akusztikus csatoló segítségével a számítógépet rá lehet csatlakoztatni a telefonvonalra. A csatoló egyik vége tehát a számítógépünkbe „illik”, a másik vége pedig a telefonkagylóra. Ez utóbbit kell ráhelyezni arra a két gumipárnaszerű kerek csatlakozásra, amely azután a gép-

ből érkező jeleket hangokká alakítja. Szükséges azonban még egy program is, amelyet terminálprogramnak is szoktak nevezni. Ez a felelős azért, hogy az adatok megfelelő kódolással kerüljenek kiküldésre, illetve adatok fogadása esetén ez a program dekódolja a jeleket.

A dolog azonban még így sem olyan egyszerű. A küldő és a fogadó félnek ugyanis néhány dologban még meg is kell egyezniük. Ilyen például az átvitel sebessége. Mi emberek sem vagyunk képesek például megérteni azokat, akik rettenetesen hadarnak. Arról pedig nem is szólva, hogy a számítógépes átvitel nem beszédrel történik, hanem egymást igen gyorsan követő impulzusok segítségével. A fogadó fél egyetlen egy füttyjelet — ezekkel történik az átvitel — sem mulaszthat el. Nos éppen ezért kell megegyezni a „beszéd” sebességében. Ennek a jellemzője az úgynevezett Baud-érték. Itt már bizonyos átviteli szabványok is léteznek, amelyeket az amatőr és a profi TAF-felhasználók általánosan használnak.

Ahhoz, hogy megértsük a Baud-érték jelentését tudnunk kell, hogy az átvitel a telefonon keresztül sorosan történik. Ez azt jelenti, hogy minden adatbyte-ot felbontunk annak különálló legkisebb egységeire a bitekre, majd ezeket libasorban továbbítjuk. Ez a távközlést egyszerűvé teszi, hiszen a bitek esetében csupán két állapotot kell közölnünk: éspedig van-e bit vagy nincs. Azaz a fogadó fél ha magas hangot hall, bekapcsolja az átkódolt információ adatbyte-jában a megfelelő bitet, ha alacsony, akkor annak értékének nullát vesz.

Egy ilyen átviteli sebesség például a 300 bps. A bps a bit per secundum helyett áll, 1 bps jelent 1 Baudot, ez utóbbi a régebbi elnevezés. A 300 bps 30 byte/mp sebességet jelent. Mielőtt még azt hinnénk, hogy a szerző rosszul számolt, gondoljuk át azt is, hogy az adatbyte mellett jó ha átvisszünk egy start- és egy stopbitet is! Így egy „TAF-átviteli byte” 10 igazi bitből áll össze. Ha mi akusztikus csatolóval dolgozunk, akkor a 300 bit/s a szabvány átviteli sebesség.

Egy újabb átviteli sebességérték szabvány az 1200 bit/s. Bár ez az átvitel négyszer gyorsabb mint a 300 bit/s, de még ma is kevés az olyan C64-es akusztikus csatoló, amely képes az 1200 bit/s-os átvitelre. Akkor már jobb, ha egy telefonmodemet (már megint egy idegen szó!)

használunk. A modem abban különbözik az akusztikus csatolótól, hogy az gyakorlatilag egy különálló készülék, amelyet közvetlenül a telefonhálózatra kell kapcsolni. Ezzel nemcsak az átvitel sebessége nő meg, hanem az átviteli minőség is megjavul, hisz megspóroljuk a telefonkagylón keresztül történő bizonytalanabb átvitelt. Mivel ilyen készülék csatlakoztatása csak a posta külön engedélyével lehet, ezért általánosabb az akusztikus csatoló használata. Más, engedély nélküli modem csatlakoztatása nem megengedett. Persze amikor egy modem előnyeiről beszélünk, ez nem jelenti azt, hogy egy (olcsó) akusztikus csatolóval nem lehet hiba nélkül dolgozni. Csupán bizonyos komfortbeli kompromisszumokat kell kötni.

Azaz ha be akarunk lépni a TAF világába, szükséges egy terminálprogram, meg egy akusztikus csatoló, amely az adatainkat úgy dolgozza föl, hogy azokat át lehessen vinni a telefonvonalon keresztül.

## Mi az előnye a dolognak?

Ugyanakkor van még valami, amit figyelembe kell venni. Ez pedig a készülékeknek a számítógéphez való csatlakoztatása. A gyártók itt is kiegyeztek egy normában, amelyet RS232C-nek hívnak. Egy ilyen szabvány csatlakozóhoz (amely legtöbbször 25 pólusú) csatlakoztathatunk egy akusztikus csatolót. A C64 és a C128 gépeknél, amelyek bár föl vannak készítve ilyen illesztésekre, mégsem találunk megfelelő csatlakozást. Éppen ezért szükséges még egy RS232C interfész beszerzése is. Ha ez is a rendelkezésünkre áll, már semmi sem gátolhat meg bennünket abban, hogy ki is használjuk a lehetőségeket.

Nos ha mindezeket beszereztük, akkor máris előttünk áll a nagy lehetőség. Hogy mik is ezek? Például kommunikálhatunk más számítógép tulajdonosokkal. Szövegadatainkat pl. átjuttathatjuk egy IBM PC-re! Ha a profi felhasználásra gondolunk, akkor az adatbankokkal való kapcsolatra kell gondolni. Igaz, ez idehaza egy C64-es felhasználónak kevésbé lehetséges. A legkomolyabb előny a távoli gépekkel fölvehető kapcsolat. A Commodore Újságot kezdetektől olvasók emlékezhetnek még arra a „távyszerkesztési” kísérletre, amely be is vált, ahol a tudósi-



tók különböző városokból közvetlenül telefonon adták le tudósításaikat a szerkesztőségi számítógépbe.

A cikk írójának is volt módja belekóstolni a távadatfeldolgozás témába, amikor egy régebbi BNV-n rátalált egy akusztikus csatolót (nálunk gyakorta modemet említenek ebben az esetben is) forgalmazó cégre. Sikeres tesztet bonyolítottunk le egy terminálprogrammal és a Textomat Plus szövegszerkesztővel (amely beépített terminálprogram funkciókkal

rendelkezik) két C64-essel, azután a cég programjával egy C64 és egy IBM PC között. Az újságunk és ez a teszt is azt mutatta, hogy még a magyar telefonvonalak mellett is jó eredménnyel kísérletezhünk. A szerző ismeretségi körében van olyan személy, aki ezen keresztül külföldi, főleg német adatbankokkal, mailboxokkal (ezek elektronikus postafiókok) tudott kapcsolatot teremteni, már amennyiben azok díjtalan szolgáltatásokat is kínáltak. Ezeket ugyanis bárki igénybe ve-

heti. Komolyabb szolgáltatásokért, programokért, hírekért stb. azonban már fizetni kell, az ilyen hozzáférést csakis bejelentkezett tagoknak teszik lehetővé.

Mindent összevetve a TAF felhasználó előtt óriási információs lehetőség és információs tartalom nyílik meg. Gyakorta még az is szerepet játszik, hogy a telefonszámla mellett szinte más költség nem is jelentkezik, tehát a TAF még szórakoztatóként is megfelel.

Szolnoki Béla

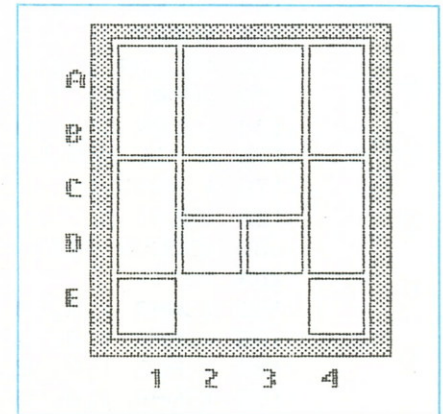
## Tologató 2

Ez a játék nagyon hasonló a Commodore újság 1991/7–8 számában megjelent tologatóhoz. Most a feladat: az ábrán látható táblán a nagy négyzetet a lapocskák tologatásával az alsó sor közepére vinni. Ez nehezebb és hosszabb, mint az első változatban, de nem lehetetlen.

Ha valakinek megvan az első tologató, nem kell az egész programlistát begépelnie, csak néhány sort kicserélnie. Ezek: 126–128: a tábla rajza; 194–216: a feladat szövege és 362: a végállás lekérdezése.

A pötyögőszolgáltatól megrendelhető a játék (Simon's Basic alatt futó) C64-es változata is.

Az első tologató megoldása 62 lépésben:



Katona János

```

01. C2-C4  02. C1-C3  03. A1-B1  04. A3-A1  05. B3-A3  06. C3-B3
07. B3-B4  08. B1-B2  09. D1-B1  10. D2-D1  11. D3-D2  12. E3-E2
13. C4-E4  14. B4-D4  15. B2-B3  16. B1-B2  17. D1-B1  18. D2-D1
19. D4-D3  20. E4-D4  21. E2-E3  22. D1-E1  23. D3-D1  24. D4-D2
25. B3-C3  26. A3-B3  27. A1-A3  28. B1-A1  29. D1-C1  30. D2-D1
31. B2-C2  32. A1-A2  33. C1-A1  34. D1-B1  35. C2-C1  36. A2-C2
37. B1-B2  38. B2-A2  39. B3-B1  40. A3-B3  41. A2-A4  42. A1-A3
43. B1-A1  44. B3-B2  45. A4-B4  46. A3-A4  47. A1-A2  48. C1-A1
49. C2-C1  50. C3-C2  51. B4-D4  52. A4-C4  53. A2-A3  54. B2-B3
55. A1-A2  56. C1-A1  57. C2-C1  58. D4-D3  59. D3-C3  60. E3-D3
61. E1-E3  62. C1-D1.
    
```

```

100 :REM *****
102 :REM * C= UJSAG SORSZAM: *
104 :REM * C+4 *
106 :REM * KATONA JANOS: TOLOGATO 2 *
108 :REM *****
110 :COLOR 4,1 : COLOR 0,1
112 :N=5:M=4
114 :DIM M(N+1,M+1):DIM R(N+1,M+1)
116 :FOR I=0 TO N+1
118 : FOR J=0 TO M+1
120 : READ M(I,J)
122 : NEXT J
124 :NEXT I
126 :DATA -9,-9,-9,-9,-9,-9,-9,3,4,-5,3,-9,-9,-3,-6,-7,-3,-9
128 :DATA -9,3,2,-2,3,-9,-9,-3,1,1,-3,-9,-9,1,0,0,1,-9,-9,-9,-9,-9,-9,-9
130 :F$="BIZTOS ? ( I/N )"
132 :G$="HELYTELEN! [RETURN]"
134 :A$="":B$="":C$="":D$="
    
```





```
136 :GRAPHIC 1,1
138 :COLOR 1,2,0
140 :P=8 : S=3*P
142 :FOR I=0 TO 3*N+1
144 : CHAR 1,0,I," ",1
146 : CHAR 1,1+3*M,I," ",1
148 :NEXT I
150 :FOR I=0 TO 3*M+1
152 : CHAR 1,I,0," ",1
154 : CHAR 1,I,1+3*N," ",1
156 :NEXT I
158 :FOR I=1 TO N
160 : CHAR 1,0,2+(I-1)*3,CHR$(I+64),1
162 :NEXT I
164 :FOR I=1 TO M
166 : CHAR 1,2+(I-1)*3,1+3*N,CHR$(I+48),1
168 :NEXT I
170 :COLOR 1,2,4
172 :FOR K=1 TO N
174 : FOR L=1 TO M
176 : IF M(K,L)>0 THEN I=K:J=L:GOSUB 380
178 : NEXT L
180 :NEXT K
182 :GOSUB 424
184 :Q=14
186 :CHAR 1,Q, 0," "
188 :CHAR 1,Q, 1,"FELADATOD A LAPOCSKAK "
190 :CHAR 1,Q, 2," "
192 :CHAR 1,Q, 3,"TOLOGATÁSAVAL A NAGY "
194 :CHAR 1,Q, 4," "
196 :CHAR 1,Q, 5,"NEGYZETET AZ ALSÓ SOR "
198 :CHAR 1,Q, 6," "
200 :CHAR 1,Q, 7,"KOZEPERE VINNI.A HUZASHOZ "
202 :CHAR 1,Q, 8," "
204 :CHAR 1,Q, 9,"A MOZGATANDÓ LAPOCSKA BAL "
206 :CHAR 1,Q,10," "
208 :CHAR 1,Q,11,"FELSŐ SARKÁNAK JELENLEGI "
210 :CIRCLE 1,148, 85,3,3,0,60,90,1 : BOX 1,121,52,122,53
212 :CHAR 1,Q,12," "
214 :CHAR 1,Q,13,"ES LEENDO KOORDINATAJÁT "
216 :CIRCLE 1,180,101,3,3,0,60,90,1 : BOX 1,125,52,126,53
218 :CHAR 1, Q, 15,"KELL MEGADNI."
220 :CHAR 1, 2, 17," "
222 :CHAR 1, 2, 18,"HUZASOD: -> "
224 :CHAR 1, 23, 21," "
226 :CHAR 1, 23, 22,"LEPESSZAM: "
228 :LS=0
230 :CHAR 1, 35, 22, STR$(LS)
232 :DO
234 : DO
236 : R=0
238 : GOSUB 410
240 : DO
242 : Z$=A$+B$+" "+C$+D$
244 : CHAR 0,10, 18,Z$
246 : CHAR 1,12, 18,"->"
248 : CHAR 0, 2, 20,F$
250 : CHAR 0, 2, 22,G$
252 : A$="":S1=0
254 : DO
256 : GET A$
258 : IF A$(">") THEN S1=ASC(A$)-64
260 : LOOP UNTIL 0<S1 AND S1<N+1
```





```
○ 262 : CHAR,10, 18,CHR$(S1) ○
○ 264 : B$="":O1=0 ○
○ 266 : DO ○
○ 268 : GET B$ ○
○ 270 : IF B$(">") THEN O1=ASC(B$)-48 ○
○ 272 : LOOP UNTIL O1<0 AND O1<M+1 ○
○ 274 : CHAR,11, 18,CHR$(O1+48) ○
○ 276 : C$="":S2=0 ○
○ 278 : DO ○
○ 280 : GET C$ ○
○ 282 : IF C$(">") THEN S2=ASC(C$)-64 ○
○ 284 : LOOP UNTIL S2<0 AND S2<N+1 ○
○ 286 : CHAR, 14, 18,CHR$(S2) ○
○ 288 : D$="":O2=0 ○
○ 290 : DO ○
○ 292 : GET D$ ○
○ 294 : IF D$(">") THEN O2=ASC(D$)-48 ○
○ 296 : LOOP UNTIL O2<0 AND O2<M+1 ○
○ 298 : CHAR, 15, 18,CHR$(O2+48) ○
○ 300 : CHAR, 2, 20,F$ ○
○ 302 : E$="" ○
○ 304 : DO ○
○ 306 : GET E$ ○
○ 308 : LOOP UNTIL E$="I" OR E$="N" ○
○ 310 : LOOP UNTIL E$="I" ○
○ 312 : V=S2-S1: F=O2-O1 : W=V+F ○
○ 314 : IF R(S1,O1)<=0 THEN R=1 ○
○ 316 : IF V<>0 AND F<>0 THEN R=1 ○
○ 318 : IF ABS(V)>2 OR ABS(F)>2 THEN R=1 ○
○ 320 : IF W=0 THEN R=1 ○
○ 322 : IF R=1 THEN GOTO 336 ○
○ 324 : IF R(S1,O1)=1 THEN A=S1:B=O1:GOSUB 438 ○
○ 326 : IF R(S1,O1)=2 THEN C=S1:D=O1:GOSUB 396 ○
○ 328 : IF R(S1,O1)=3 THEN A=S1+SGN(W+ABS(W)):B=O1:GOSUB 438 ○
○ 330 : IF R(S1,O1)=3 THEN A=A-SGN(W):GOSUB 438 ○
○ 332 : IF R(S1,O1)=4 THEN C=S1+SGN(W+ABS(W)):D=O1:GOSUB 396 ○
○ 334 : IF R(S1,O1)=4 THEN C=C-SGN(W):GOSUB 396 ○
○ 336 : IF R=1 THEN CHAR, 2, 22,G$ ○
○ 338 : H$="":H=0 ○
○ 340 : DO ○
○ 342 : GET H$ ○
○ 344 : IF H$(">") THEN H=ASC(H$) ○
○ 346 : LOOP UNTIL H=13 OR R=0 ○
○ 348 : LOOP UNTIL R=0 ○
○ 350 : I=S1:J=O1:GOSUB 388 ○
○ 352 : GOSUB 424 ○
○ 354 : I=S2:J=O2:GOSUB 380 ○
○ 356 : LS=LS+1 ○
○ 358 : CHAR 0, 35, 22,STR$(LS-1) ○
○ 360 : CHAR 1, 35, 22,STR$(LS) ○
○ 362 : LOOP UNTIL M(4,2)=4 ○
○ 364 : FOR I=1 TO 1000: NEXT I ○
○ 366 : GRAPHIC 0 :PRINT"┘" ○
○ 368 :PRINT:PRINT:PRINT" " ○
○ 370 :PRINT " GRATULALOK! " ○
○ 372 :PRINT:PRINT:IF LS>99 THEN PRINT" "; ○
○ 374 :PRINT" " ○
○ 376 :PRINT" MEGOLDOTTAD"LS"LEPESBEN":PRINT:PRINT ○
○ 378 :END ○
○ 380 :REM RAJZ ○
○ 382 : XX=P+(J-1)*S : YY=P+(I-1)*S ○
○ 384 : BOX 1,XX,YY,XX+S*(2-M(I,J)+INT(M(I,J)/2)*2)-2,YY+S*(1+INT(M(I,J)/3))-2 ○
○ 386 :RETURN ○
```







```

○ 388 :REM TORLES
390 : XX=P+(J-1)*S : YY=P+(I-1)*S
○ 392 : BOX 0,XX,YY,XX+S*(2-R(I,J))+INT(R(I,J)/2)*2)-2,YY+S*(1+INT(R(I,J)/3))-2
394 :RETURN
○ 396 :REM KETTES
398 : A=C:B=D+SGN(W+ABS(W)):GOSUB 438
○ 400 : B=B-SGN(W):GOSUB 438
402 :RETURN
○ 404 :REM LEPES
406 : IF M(X+V,Y+F)=0 THEN M(X+V,Y+F)=M(X,Y):M(X,Y)=0: ELSE :R=1
408 :RETURN
○ 410 :REM ATOLVAS
412 : FOR I=0 TO N+1
○ 414 :   FOR J=0 TO M+1
416 :     M(I,J)=R(I,J)
○ 418 :   NEXT J
420 : NEXT I
422 :RETURN
○ 424 :REM VISSZA
426 : FOR I=0 TO N+1
○ 428 :   FOR J=0 TO M+1
430 :     R(I,J)=M(I,J)
○ 432 :   NEXT J
434 : NEXT I
436 :RETURN
○ 438 :REM EGYES
440 : MU=0
○ 442 : IF ABS(W)=1 THEN X=A:Y=B:MU=0:GOSUB 404
444 : IF ABS(W)=2 THEN V=V/2:F=F/2:X=A:Y=B:MU=1:GOSUB 404
○ 446 : IF MU=1 THEN X=A+V:Y=B+F:GOSUB 404
448 : IF MU=1 THEN V=2*V:F=2*F
○ 450 :RETURN

○ READY.

```

## Directory C64

Ha ez nem válik egyik leggyakrabban használt segédprogrammóddá, akkor nem nagyon használd a C64-esedet! A progra-

mocská futtatása után bármely lemez tartalomjegyzékét megnézhetjük SYS 850 "\$:\*" hívásával.

Szelektív tartalomjegyzék a kérdőjel szokásos használatával kérhető. A SHIFT gomb megállítja a tartalomjegyzék futását a képernyőn, a RUN-STOP-pal pedig kiléphetünk belőle.

```

○ 0 REM ML DIRECTORY 64
10 K=850:FOR J=0 TO 134:READ A:POKE K+J,A:T=T+A:NEXT
○ 20 IF T<17974 THEN PRINT"HIABA DATA-SOROKBAN...":END
30 PRINT"SYS"K","CHR$(34)"#;"CHR$(34)" DIRECTORY-HOZ"
○ 50 DATA 32,121,0,201,44,208,120,32,115,0,201,34,208,113,32,115,0,201,36
60 DATA 208,106,169,234,133,130,133,131,166,122,164,123,134,7,162,254,134
○ 70 DATA 2,162,0,32,115,0,232,201,0,208,248,138,166,7,32,189,255,169,8,170
80 DATA 160,0,32,186,255,32,192,255,162,8,32,198,255,160,6,208,2,160,4
○ 90 DATA 170,32,207,255,136,208,249,164,144,208,32,32,205,189,169,32,32,22
100 DATA 231,32,207,255,208,248,230,2,169,13,32,22,231,174,141,2,202,240
○ 110 DATA 250,165,197,201,63,208,211,32,204,255,169,8,32,195,255,169,240
120 DATA 133,130,169,239,133,131,96
○ READY.

```



# Alap

## Segédprogram C-128-ra

### A program szolgáltatásai:

- ékezetes kis- és nagybetűk normál és inverz módban úgy képernyőn mind nyomtatón
  - direkt módban kezdeményezett listázásnál a nyomtatón is „rendes” listát ad
  - képernyő 2—24 sorának másolása nyomtatóra
  - REM után sem tokenizálja a magyar karaktereket
  - BASIC utasítások bevitele egyetlen billentyűvel
- korlátozások: az új karakterek nem használhatók file névnek; a nyomtatóra ígért „rendes” lista csak 99,9%-ban teljesül; adatokat idézőjelben kell elhelyezni.

A program hardver igénye: C-128, 80 karakteres monitor, MPS 803-as nyomtató

Indítás: SYS 5685, utána NEW.

### Tanácsok az „Alap” program használatához:

A program RUN/RESTOR vagy RESET után sys 5685-el aktivizálható.

A magyar ékezetes kis betűk:

C=+betű illetve  
 ö=C=+\*; ő=C=+@;  
 ü=C=+!; ü=!

Nagy betűk mint fent +SHIFT. (Ez megegyezik DATAMAT 128 billentyű kezelésével.)

A betűk ASC értéke \$a0—b9 tartományba esik. Az ismertett betűkön kívül még egy ál idézőjel van (\$b2 értékkel). Tehát a b3—b9 tartomány nincs kihasználva, úgy a képernyő mind a nyomtató data táblákban meg van a helyük, szabadon kreálhatunk Ft. ... stb. karaktereket.

Nyomtatón, inverz új karaktereinket csak program módban produkálja, egyébként nem inverz karaktereket nyomtat.

Ha a prg. lista nyomtatással nem kezdenénk semmit, idézőjeles üzemmódban tűvezérléssel előállított karaktereink helyett minden tű datára egy valamilyen karaktert nyomtatna. Ennek kikerülésére az új chROUT rutin DIREKT módban elrendelt lista nyomtatás esetén az idézőjel helyett az ál idézőjelet küldi ki, melyről

azonban a listázó rutin nem értesül. Ez az ami csak 99,9%-osan működik, ugyanis saját, hosszabb BASIC programjaimban van olyan konstelláció mikor is a listázó rutin mégiscsak kikerül az idézőjel módból, s tokenizálni kezdi a magyar karaktereket.

Ez a hiba képernyőre történő listázásnál nem fordul elő.

A prg. új karaktereinket, új kódtábláinkból állítja elő. A SHIFT+C= lenyomása az 1. számú míg az ALT billentyű lenyomása 2. számú ALT tábla mutatóit tölti a megfelelő vektorba. Az új kódtáblák adatait szabadon manipulálhatjuk.

Az 1. sz. ALT táblában vannak a nagy ékezetes karakterek. A többi helyen egy mutató van, amely azt tükrözi, hogy az illető billentyű lenyomásával a 15-ös szelet \$ 4417 címétől kezdődően elhelyezett BASIC szavak közül hányadikat akarjuk megjeleníteni. A felsorolás megegyezik a „Commodore 16, PLUS/4, C 64, C 128 Információs kártya” (Dr. Ury László 1986) által ismertett sorrenddel kivéve a POT — POINTE közötti tartományt, mert az a gépben az összes többi után van. (A könyv növekvő token szerinti listát ismertet.)

Mint jeleztem két ALT tábla van, tehát 2. számú ALT tábla teljesen üres, illetve jelenleg \$ 05-el van feltöltve. Az ide beírt sorszámoknak megfelelő BASIC alapszavak az ALT+billentyű lenyomása nyomán fognak megjelenni. Az 1. sz. ALT táblát saját igényeim szerint Paramétereztem, de természetesen ezen bárki könnyen változtathat, minek után a SHIFT+C=+billentyű használat az általa kívánt szavakat fogja produkálni.

### „ALAP” assembler prg.

feladat	cím	használt regiszterek
\$fb-fc által meghatározott mutató értékét növeli \$be értékével	\$1300—130d	fb, fc, be
saját képernyő karakterek értékeit a video RAM-ba másolja	\$130e—1343	fb, fc, be, f8, f9, c5
a tényleges beírás	\$17ae—17c9	
	\$17ca—17d5	
data tábla	\$1400—14cf	
hrdcpy	\$1376—13e1	9e, 9f, f8, f9
folytatása ha nyomtatóra	\$1850—1878	c5=00 pufferből képernyőre
ha pufferbe	\$1880—189f	c5=ff nyomtatás
byte out, képernyő vége?	\$1346—1364	c5=80 képernyőről pufferbe
segéd tábla asc-re váltáshoz	\$17ef—17f2	
prg start	\$1635—1656	fb, fc, be, c5,
folytatás	\$18d0—18e8	
új vektorok és tábla címek	\$1800—1812	
billentyűzet letapogatáshoz	\$13f2—13ff	
ezek datái	\$1659—1711	
chROUT	\$17e3—17ee	fb, fc, fd, 9b, 9c,
ha nyomtatóra	\$1aa0—1ab7	
prg módban inverz be vagy ki?	\$1aa0—1ab7	
tű data változtatás inverznél	\$1712—17ad	
tű data tábla	\$1813—183f	
szerkesztő billentyűk manipulálása	\$18bf—18c5	
hrdcpy nyomtató zárása	\$1900—191b	74
REM utáni tokenizálás letiltása	\$191e—198f	fc, fb, c5,
basic szó egy billentyűvel (funkció-billentyűk manipulálása)	\$1990—1a2f	
mutató LO tábla	\$15db—1634	
1. ALT tábla	\$1a30—1a88	
2. ALT tábla		
Klaviatúra táblák		
alap	\$14d0—1528	
SHIFT	\$1529—1581	
C=	\$1582—15da	
ALT-ok lásd fent		
CONTROL eredeti maradt		





Az elvárásainkat megvalósító rutin lényegében úgy működik, hogy a mutatók alapja meghatározott szó karaktereit az F8 billentyű tárolójába helyezi, s megnyomott billentyűként F8-at jelez. Ennek megfelelően a továbbiakban az F8 billentyűt nem definiálhatjuk. Az eredeti jelentés (monitor) így is egy mozdulattal produkálhat SHIFT+C+=+m. Csupán a return a plusz erőfeszítés, amit a HELP után is meg kell tennünk.

Ezek után nézzük mit eredményez a többi billentyű SHIFT+C= egyidejű használatát.

Révész György

numerikus blokk:

billentyű	szó
1	dclear
2	scratch
3	header
4	do
5	loop
6	until
7	run
8	end
9	char
0	dverify
+	pudéf
-	while
ENTER	exit
	rename

billentyű	szó	billentyű	szó
ESC	asc	r	rem
TAB	cont	t	bload
HELP	help	y	bsave
LINE FEED	sys	p	begin
fel	on	@	bend
le	goto	*	clr
balra	gosub	STOP	stop
jobbra	return	d	dim
F1	for	f	fast
F3	to	g	if
F5	step	h	then
F7	next	j	else
-	fre	k	key
1	chr\$	l	list
2	left\$	:	read
3	right\$	;	data
4	mid\$	=	restore
5	instr	z	dec
6	str\$	x	hex\$
7	val	c	catalos
8	int	v	wait
9	len	b	bank
0	abs	n	new
+	open	m	monitor
-	close	,	peek
£	record	.	poke
HOME	input #	/	print #
DEL	get	CRSR fel	or
q	cmd	CRSR balra	and
w	window	SPACE	using

```

○ monitor
  pc sr ac xr yr sp
  b000 00 00 00 00 f8
○
○ >01300 48 a5 fb 18 65 be 85 fb 90 02 e6 fc 68 60 a9 ff:
○
○ >01310 85 fb a9 35 85 fc a9 00 85 f8 a9 14 85 f9 a2 00:
○
○ >01320 a9 01 85 be a0 00 20 00 13 20 ae 17 c8 c0 08 d0:
○
○ >01330 f5 e8 e0 18 f0 d7 a9 08 85 be 20 00 13 a9 08 65:
○
○ >01340 f8 85 f8 4c 20 13 20 59 16 a5 9e 18 69 01 85 9e:
○
○ >01350 90 02 e6 9f a5 9f c9 07 d0 0a a5 9e c9 d0 d0 04:
○
○ >01360 a9 01 85 f9 60 ea ea ea ea ea ea ea ea ea ea ea:
○
○ >01370 ea ea ea ea ea a3 20 dd 02 a0 07 a9 04 aa 20 ba:
○
○ >01380 ff a9 00 20 bd ff a2 0f 20 68 ff 20 c0 ff a2 04:
○
○ >01390 20 c9 ff a9 a0 85 9e a9 00 85 9f a5 9e a2 13 20:
○
○ >013a0 ca 17 a5 9f ca 20 ca 17 a2 1f 8e 00 d6 2c 00 d6:
○
○ >013b0 10 fb ad 01 d6 c9 80 30 02 a2 20 a0 03 c9 60 10:
○
○ >013c0 0b 88 c9 40 10 06 88 c9 20 30 01 88 18 79 ef 17:

```









>015d0	39 33 08 30 2e 91 11 9d 1d ff ff 22 0d 30 03 02:	
>015e0	25 2a 31 4a 96 a1 4b 51 a1 a3 01 54 10 07 45 88:	
>015f0	9f 8d 52 46 8c 0c 36 7e 28 ad 13 44 a5 55 37 7a:	
>01600	79 a7 23 20 94 1c 21 18 08 95 43 8e 1d 04 05 01:	
>01610	0d b1 19 48 39 04 49 7b 02 1e 11 6a 01 6c 1b 72:	
>01620	6b 0b 91 47 5d 7d 1f 6d 7c 60 71 08 90 75 12 0a:	
>01630	0e 0f ff ff 00 20 dd 02 20 d6 17 a9 00 85 c5 20:	
>01640	0e 13 a9 ff 85 c5 a9 ff 85 fb a9 3d 85 fc 20 16:	
>01650	13 a9 0e 20 79 ef 4c d0 18 48 a5 9a c9 04 f0 03:	
>01660	4c 0c 17 a5 7f c9 00 f0 10 4c 90 1a 8d 9f 1a 48:	
>01670	ea ea a5 c5 c9 ff d0 51 ea 68 48 c9 22 d0 08 a9:	
>01680	b2 20 ca 16 4c ac 16 c9 0d f0 3e c9 0d f0 3a d0:	
>01690	1d ea ea 18 69 40 8d c8 16 a9 12 20 e3 17 ad c8:	
>016a0	16 20 e3 17 a9 92 20 e3 17 ea ea ea 68 60 c9 81:	
>016b0	30 07 c9 9f 30 dd 4c c9 16 c9 1f 30 d6 4c c9 16:	
>016c0	ea ea ea ea ea ea ea ea 53 68 c9 a0 30 3f c9 b7:	
>016d0	10 3b 85 fd 98 48 8a 48 a2 12 86 fb a2 17 86 fc:	
>016e0	a2 06 86 be a0 00 a2 a0 e4 fd f0 07 20 00 13 e8:	
>016f0	4c e8 16 a9 08 20 e3 17 b1 fb 20 a0 1a c8 c0 06:	
>01700	d0 f6 a9 0f 20 e3 17 68 aa 68 a8 60 68 4c 79 ef:	
>01710	ea ea a0 d4 d6 bd c0 80 f8 94 96 95 f8 80 b8 d4:	
>01720	d6 d5 88 80 fc d4 d6 d5 c4 80 80 c8 fa c1 80 80:	
>01730	80 c4 fc c6 81 80 b0 c8 ca c9 b0 80 b8 c4 c6 c5:	
>01740	b8 80 b8 c5 c4 c5 b8 80 bd c2 c2 c2 bd 80 b0 ca:	
>01750	c9 c8 b2 81 b8 c6 c5 c6 b9 80 b8 c0 c2 b9 c0 80:	
>01760	bc c0 c2 c1 bc 80 ba c0 c0 ba c0 80 bc c1 c0 c1:	
>01770	bc 80 ba c1 c0 ba c1 80 ba c1 c0 c2 b9 80 80 8e:	
>01780	80 8e 80 80 02 02 02 02 02 80 03 03 03 03 03 80:	
>01790	04 04 04 04 04 80 05 05 05 05 05 80 06 06 06 06:	
>017a0	06 80 07 07 07 07 07 80 08 08 08 08 08 80 8a 48:	
>017b0	a5 fb a2 13 20 ca 17 a5 fc a2 12 20 ca 17 b1 f8:	
>017c0	45 c5 a2 1f 20 ca 17 68 aa 60 8e 00 d6 2c 00 d6:	





```

>017d0 10 fb 8d 01 d6 60 78 a9 59 8d 26 03 a9 16 8d 27:
>017e0 03 58 60 84 9b 86 9c 20 79 ef a4 9b a6 9c 60 00:
>017f0 40 20 40 ea ea ea ea ea ea ea ea ea ea ea ea:
>01800 78 a2 00 bd f2 13 9d 3a 03 e8 e0 0e d0 f5 58 60:
>01810 ea ea ea 48 a5 d3 c9 03 f0 17 c9 08 d0 0a a9 30:
>01820 8d 46 03 a9 1a 8d 47 03 a9 00 8d 07 10 68 4c e1:
>01830 c5 a9 db 8d 46 03 a9 15 8d 47 03 a9 08 85 d3 4c:
>01840 28 18 a5 f8 18 69 01 85 f8 90 02 e6 f9 60 ea ea:
>01850 68 c9 26 d0 02 a9 0d 20 46 13 e6 f8 a5 f8 c9 50:
>01860 d0 09 a9 0d 20 59 16 a9 00 85 f8 a5 f9 c9 01 f0:
>01870 4e a5 d3 c9 08 f0 48 4c 9b 13 ea ea ea ea ea d6:
>01880 a9 f8 8d b9 02 a0 00 a2 3f 68 20 af 02 20 42 18:
>01890 20 49 13 a5 f9 c9 01 d0 01 60 4c 9b 13 ea ea a3:
>018a0 a9 f8 8d aa 02 a0 00 a2 3f 20 a2 02 20 79 ef 20:
>018b0 42 18 a5 f9 c9 01 d0 01 60 20 49 13 4c a0 18 a9:
>018c0 0d 20 59 16 a9 04 20 c3 ff 20 cc ff 60 da ea d3:
>018d0 a9 80 8d c5 0a 85 f7 a9 00 85 f8 85 c5 8d 06 03:
>018e0 a9 19 8d 07 03 a9 13 8d f2 17 20 00 18 a9 40 8d:
>018f0 f2 17 60 ea ea ea ea ea ea ea ea ea ea ea ea:
>01900 08 48 a5 74 c9 8f f0 0b 68 c9 8f d0 02 85 74 28:
>01910 4c 51 51 a9 fe 85 11 a9 00 85 74 4c 08 19 48 a5:
>01920 d3 c9 08 d0 65 68 c9 a1 90 04 4c 87 19 ea 48 a9:
>01930 44 85 fc 68 c9 48 30 08 e6 fc c9 7d 30 02 e6 fc:
>01940 aa bd 8f 19 85 fb a2 00 86 c5 bd 00 10 18 65 c5:
>01950 85 c5 e8 e0 07 d0 f3 a0 00 a2 0f a9 fb 8d aa 02:
>01960 20 a2 02 20 a2 02 c9 80 10 0b a6 c5 9d 0a 10 c8:
>01970 e6 c5 4c 59 19 38 e9 80 a6 c5 9d 0a 10 c8 8c 07:
>01980 10 a9 00 85 c5 a9 8c 4c ad c6 68 4c ad c6 ea ea:
>01990 17 1a 1d 21 25 2b 30 33 37 3a 3e 41 43 4a 4f 55:
>019a0 58 5c 5e 62 66 6a 70 73 77 7d 82 86 8a 8d 90 93:
>019b0 97 9c 9f a2 a6 a8 aa ae b2 b5 b9 ba bb bc bd be:
>019c0 c1 c3 c4 c5 c6 c9 cc cf d2 d5 d8 db de e1 e4 e7:

```









# C-64 bővítések

## Gyorstöltő modul C-64 lemezegységhez (Fastload kártya)

Ez olyan bővítő, amely a C-64 és a hozzá csatlakozó lemezegység működését gyorsítja, kezelését nagyon kényelmessé teszi, mert a sok billentyű működtetése helyett (LOAD...; LIST) elegendő egy-egy gomb megnyomása a lemezegység működtetéséhez.

### A Fastload kártya működése

A Fastload használatával minden lemezművelet ideje kb. a negyedére csökken, és ezt úgy csinálja, hogy egyetlen bitet sem foglal el a C-64 memóriájából.

Ennek köszönhető, hogy szinte minden programmal hajlandó együttműködni, és bármilyen hosszú programot betölt. Ezeket az előnyös tulajdonságokat egy ötletes hardver megoldásának köszöheti. Az ötlet lényege az, hogy a cartridge csak addig van bekapcsolva, amíg segít betölteni a programot. Ha ezt elvégezte, saját magát elfelejti. Ezt az 1. ábrán látható kapcsolási rajz szerint úgy oldották meg, hogy a ROM jelenlét jelzését a gép felé C<sub>1</sub> feszültség szintje biztosítja. Ez nyugalmi helyzetben +5 V-on van.

C<sub>1</sub> kisütését a program betöltése alatt a ROML periodikusan végzi, hogy ne legyen ideje feltöltődni. Befejezve a program betörését, a ROML pulzálása megszűnik, C<sub>1</sub> feltöltődik +5 V-ra, a cartridge önmagát kikapcsolja. A Fastload programja autostartos, ezért a gép bekapcsolása után mindig bejelentkezik, és a kurzor alatt megjelenik a Fastload felirat. Az autostartos betöltésnél I/O<sub>1</sub> pulzálja a C<sub>1</sub> feszültségét.

### A Fastload kezelése

A cartridge programja három fő részre tagolható. Ezek: a lemez gyors beolvasását segítő rész, egyéb lemezművelet menüvel támogató rész és egy monitor rész. Ezek használata nem nehéz, a beolvasást segítő rész kifejezetten egyszerű,

de a cartridge adta lehetőséget csak akkor tudjuk kihasználni, ha rendelkezésünkre áll egy használati utasítás is, ezért röviden most ezt ismertetem.

A lemez beolvasásához rendelkezésre álló parancsok:

\$ vagy >\$: a tartalomjegyzék betöltése. A memóriában levő program érintetlen marad. A listázás CTRL-lel lassítható, RUN/STOP-pal megállítható.

C = RUN/STOP: az első file betöltése, elindítása RUN-nal — mint a LOAD " : \* " ,8,1 (gépi kódú programok esetén ne használjuk)

/NÉV: BASIC program betöltése — mint a LOAD " Név " ,8

%NÉV: Gépi kódú program betöltése — mint a LOAD " Név " ,8,1

A név idézőjel nélkül használható, és megengedett a tartalomjegyzékből való hívás. A név elé írjuk a / vagy % karaktereket, és a felesleges számokat töröljük. Ezután RETURN és kész.

### Menüvel jelentkező parancsok:

£: a főmenü hívása  
A lemez tartalomjegyzék  
B visszatérés BASIC-be  
C a másolómenü hívása  
D a Fastload kikapcsolása. Visszatérés a RESET gomb működtetésével.

E a lemez editálása, a megjelenő kép: EDIT DISKETTE TRACK nn SECTOR nn.

A track és szektor számát hexában vagy „#” jellel a szám előtt decimálisan adhatjuk meg. Műveletkiválasztás: Read (olvas), Write (ír), Quit (kilép)

C: Copy menü hívása

A tartalomjegyzék

B visszatérés a főmenübe

C teljes lemez másolása formatált lemezre

E egyes file-ok másolása. Használható a „?”; “\*” jel

F formatálás

F: File menü

A tartalomjegyzék

B visszatérés a főmenübe

C file Copy (lásd copy menü)

D file törlés

E file írásvédelem (a név mögött egy „<” jel jelenik meg)

F file írásvédelem oldása

G file átnevezése

### Monitor:

Hívása: !; kilépés: %

A Fastload monitorok sok utasítása eltér az általánosan használt monitorékétól, de a tapasztalat szerint ezek az eltérések csak előnyére válnak.

A parancskarakterek a számok (címek, argumentumok) előtt, között és mögött is lehetnek. A számok között vessző, space és egyéb írásjel is állhat. A hexa alakban való megoldáson kívül is lehet a „#” jellel decimális számot írni és a „/” jel után ASCII kóddal hivatkozni.

Használhatunk műveleti jeleket:

& logikai ÉS (AND)

! logikai VAGY (OR)

? logikai KIZÁRÓ VAGY (EXOR)

+,- matematikai műveletek

### Parancsok:

cím: G: a címen kezdődő szubrutin végrehajtása.

cím 1 cím 2 H byte 1, byte 2... byte n: a cím 1 és 2 közötti területen keres egy byte 1,2... byte n karakter sorozatot vagy kombinációt. Összesen tízet.

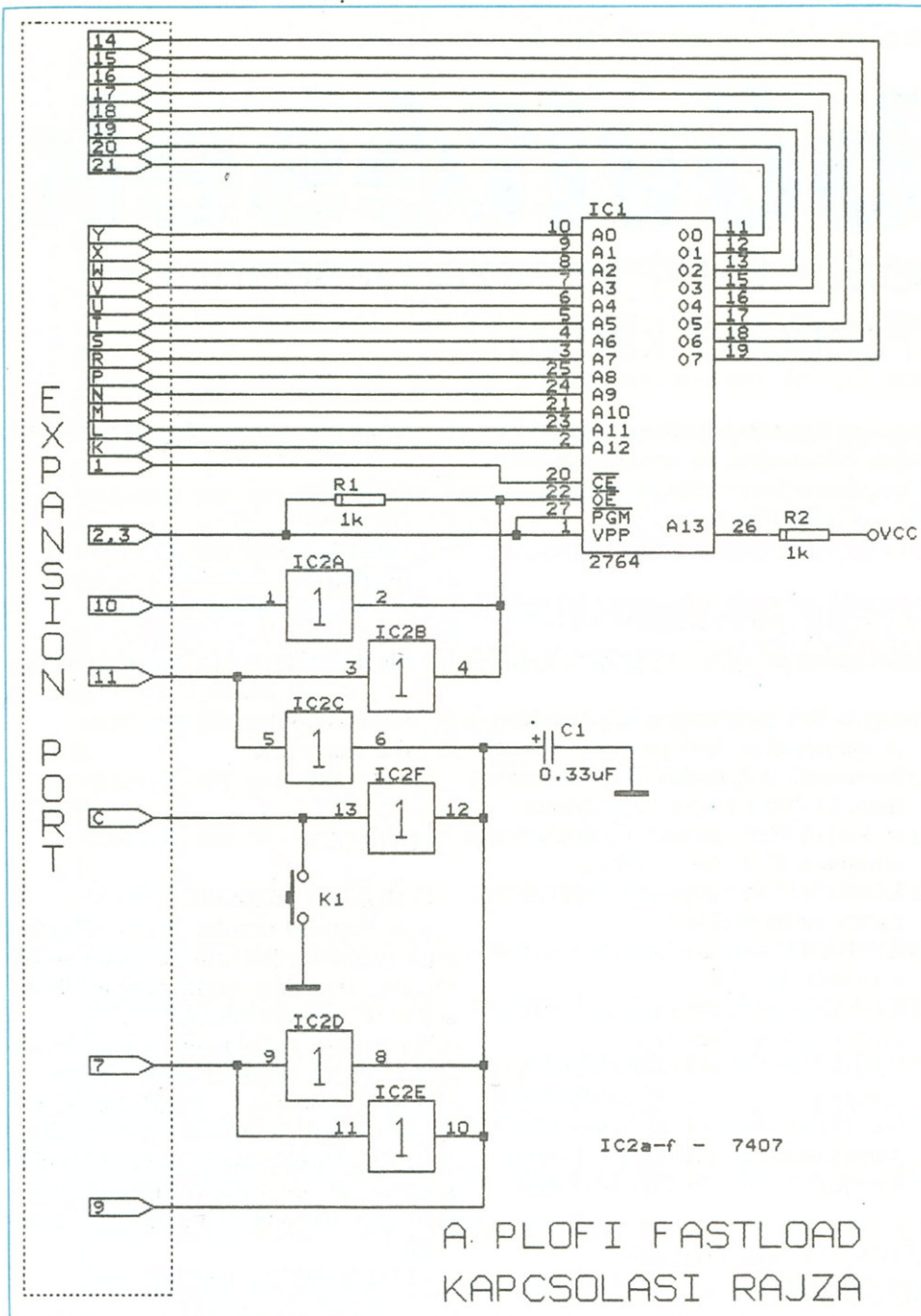
Pl. 0000 2222 H A9A9 7090

byte kombináció

A 0000 és 2222 közötti területen A9 és 70...90 értékek kombinációját keresi.

cím J: a címen kezdődő programot végrehajtja.





A. PLOFI FASTLOAD KAPCSOLASI RAJZA

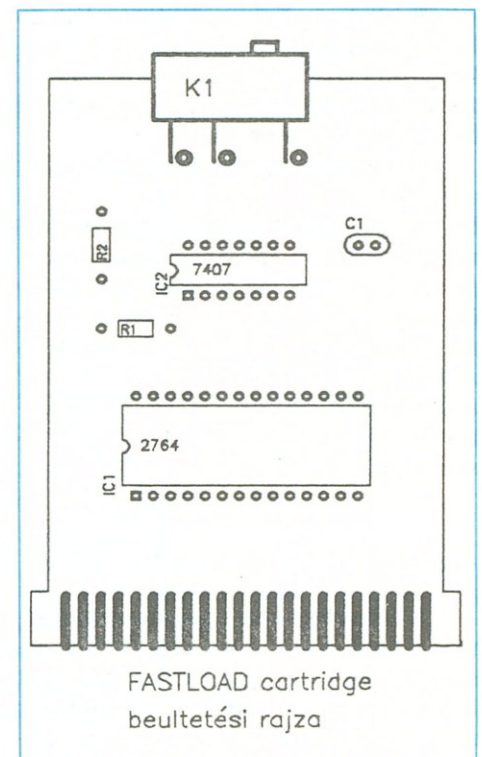
arg 1 arg 2 cím 1 cím 2 I: a cím 1 és cím 2 között disszaszemblija a programot, és arg 1 arg 2 között ki írja.  
 cím 1 cím 2 L: cím 1 cím 2 között disszaszemblija a programot, és kiírja a képernyőre.  
 cím 1 cím 2 cím 3 M: a cím 2 és cím 3 közötti területet a cím 3-tól kezdődően átmásolja. Ha a cím 1 alacsonyabb, mint cím 2, akkor a tartományon belüli másolás is lehetséges.  
 N: NOP nincs művelet.  
 Q 0000 1111 2222 3333 4444 Q  
 Program relokálás. A 3333 és 4444 címek közé eső program 1111 és 2222 közé eső program 1111 és 2222 közé eső

abszolút címeit relokálja. 0000-t kivon belőlük.  
 cím arg R „NÉV”: a NÉV file-t betölti az arg eszköztől a címtől kezdve.  
 cím S: a címtől kezdve folyamatosan végrehajtja a programot, és kiírja a regiszterek tartalmát.  
 cím T: a címtől kezdve folyamatosan végrehajtja a programot, és kijelzi a regiszterek tartalmát.  
 cím 1 cím 2 3 V: a cím 2 cím 3 közötti területet összehasonlítja a cím 1-től kezdődő területtel. Az eltérést azonnal jelzi.  
 cím 1 cím 2 arg W „NÉV”: a cím 1 cím 2 terület a NÉV file-ba menti az argumentumban megadott eszközre.

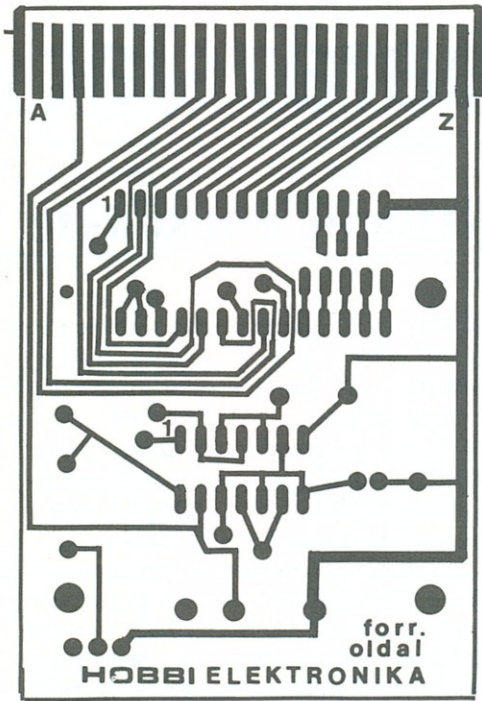
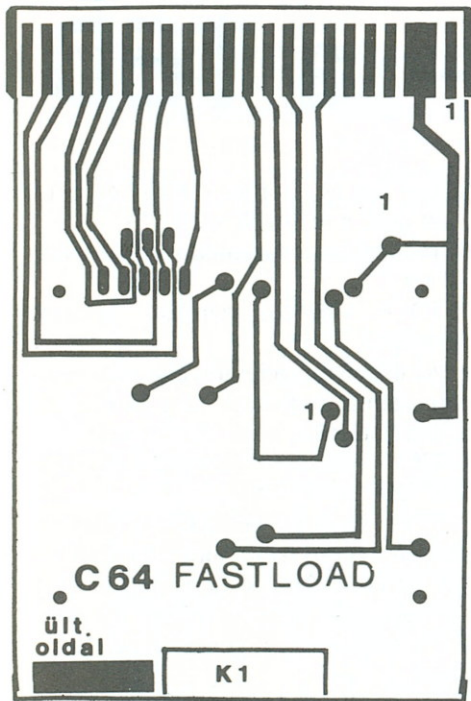
arg1arg2cím1 cím 1 és cím 2 között cím 2 X } disszaszemblija és kiírja az  
 Y : arg1 és arg2 közötti indirekt címeket. (X csak az X-nél, Y csak Y-nál címzettek, @ mindkettőt.)  
 @ )  
 cím 1 cím 2 cím 3 cím 4 Z: a cím 3 és cím 4 közötti terület disszaszemblija, és kiírja az összes abszolút vagy nulláslap referenciát, amely a cím 1 és cím 2 közé esik, beleértve az elágazások referenciáit is.  
 cím 1 cím 2 \*: cím 1 cím 2 közötti területet a képernyőn listázza hexadecimális alakban és ASCII kódban.  
 AA XX PP SP: kijelzi a regisztereket. Módosítás is lehetséges, de pl. az XX módosításához az AA-t is meg kell adni.

## A Fastload cartridge szerelése, élesztése

A nyomtatott áramkört (2. ábra) a háttároló vonalak mentén reszeljük méretre, közben többször illesszük a bővítő port csatlakozójához, hogy pontos illesztést kapjunk a két oldal ütköztetésével. A nyák furatgalvanizált, 3 helyzetfurattal illeszkedik a dobozhoz, hogy mindig az alkatrész oldal nézzen a feliratozott fedélről le kell vágni az egyik helyezőcsapot.  
 Ez az egységcsomag dobozánál megtörtént, csak ellenőrizni kell.







IC foglalatot ne használjunk, mert a doboz belső mérete ezt nem teszi lehetővé. A jól elkészített kártya biztosan működőképes. Ha véletlenül mégis ki kellene emelni az IC-t, az amatőr eszközökkel is elvégezhető, de ha ez gondot okoz, ebben is segít a szerző szolgáltatása.

A gondosan elkészített cartridge élesztésével nincs gond, statikus mérés nem kell végezni rajta. Rá kell dugni a *ki-kapcsolt* gépre. Ha a kurzor alatt megjelenik a FASTLOAD felirat, a használati utasítás szerint ellenőrizzük a működést. (A cikkünkhöz kapcsolódó hirdetés borítónk második oldalán, vásárlási utalvány a 34. oldalon.)

## C-64 bővítések

```

0 REM AUTO VERIFY 64
1 :
10 FOR J=832 TO 935:READ Q:POKE J,Q:NEXT
20 SYS 832:PRINT:PRINT"AUTO VERIFY BEKAPCSOLVA":NEW
30 DATA 120,173,50,3,141,107,3,173,51,3,141,108,3,169,99,141
40 DATA 50,3,169,3,141,51,3,88,96,6,69,82,82,79,82,255
50 DATA 79,75,255,234,169,111,72,169,3,72,32,111,223,104,104,234
60 DATA 169,1,162,255,160,255,32,213,255,32,183,255,201,64,240,8
70 DATA 169,0,141,89,3,76,141,3,169,6,141,89,3,234,24,169
80 DATA 13,32,210,255,174,89,3,189,90,3,201,255,240,7,232,32
90 DATA 210,255,76,151,3,234,96,0
    
```

READY.

### Auto verify C-64

Akinek már veszett el programja lemezhiba vagy más ok miatt bizonyára szívesen fogadja ezt a kis gépi kódú rutint.

Az Auto verify automatikusan ellenőríz (VERIFY parancs) minden olyan BASIC programot, melyet lemezre mentünk.

Gépeljük be, mentjük el lemezre, azután futtassuk (RUN). Ezután az Auto verify-t minden használat előtt SYS 832-vel aktiválhatjuk. A RUN-STOP/RESTORE megnyomásával kikapcsolható. Az Auto verify az összes gyorsítottóval és a cartridge-ekkel kompatibilis.

### RAM CLEAR

A RAM CLEAR egy olyan kis utility, melynek kezdő és haldó C64 tulajdonos egyaránt jó hasznát veheti. Futtatásával a lemezen létrehozunk egy RAM CLEAR 64 nevű programot. Ezt LOAD "RAM CLEAR 64",8,1-el betölthetjük és SYS 828-cal indíthatjuk. Hatására egy másodperc alatt mindent törölünk a memóriából a 2048 és 53247 címek között. Ha gépünkbe BASIC programot és különböző segédprogramokat töltöttünk, a RAM CLEAR segítségével a gép kikapcsolása nélkül törölhetjük azokat.

```

0 REM RAM CLEAR
1 :
10 OPEN 8,8,8,"0:RAM CLEAR 64.P.W"
20 FOR T=1 TO 54:READ B:PRINT#8,CHR#(B):NEXT
30 CLOSE 8:END
40 DATA 60,3,165,1,72,41,254,133,1,169,0,133,251,133,55,168,162,8,134
50 DATA 252,134,44,145,251,200,208,251,230,252,166,252,224,208,208,243
60 DATA 104,133,1,169,1,133,43,169,160,133,56,169,0,32,66,166,76,116
70 DATA 164
    
```

READY.



# Képernyővarázslat — a TED

## Második rész — a regiszterek áttekintése

Remélem, még mindig akadnak olyan olvasók, akiket nem riasztott el az első rész. Sajnálatos tapasztalat ugyanis, hogy sok programozó a hardware-hez nem ért, és ezért azt nem szereti (én sem értek hozzá igazán, de szeretem).

Térjünk hát rá arra, amit egy programozó elvár egy ilyen cikksorozattól, vagyis a dolog szoftver oldalára, de előre figyelmeztetek mindenkit, hogy még a hardverről is lesz szó...

A TED számos regiszterrel rendelkezik, ezért úgy gondolom, hasznos lesz átfutni rajtuk először.

A regiszterek báziscíme \$FF00 (dec. 65280)

\$FF00 — első időzítő alacsony bájt

\$FF01 — első időzítő magas bájt

\$FF02 — második időzítő alacsony bájt

\$FF03 — második időzítő magas bájt

\$FF04 — harmadik időzítő alacsony bájt

\$FF05 — harmadik időzítő magas bájt

Kvarcvezérelt időzítőkről van szó, pontos időzítés érhető el velük viszonylag széles tartományban (elméletileg 1,13 mikroszekundumtól 74,14 milliszekundumig).

\$FF06 — videóvezérlő regiszter

\$FF07 — videóvezérlő regiszter

Talán ez a két leggyakrabban használt regiszter.

\$FF08 — billentyűzet állapot (latch)

A billentyűzet és a joystick-ok figyelmében játszik szerepet.

\$FF09 — megszakítás érzékelő regiszter

\$FF0A — megszakítás engedélyező regiszter

A különböző megszakításokat engedélyzhetjük és érzékelhetjük velük.

\$FF0B — regisztermegszakítás sora alsó nyolc bit, felső bit \$FF0A-ban

A képernyő bizonyos sorában megszakítást válthatunk ki, ennek helyét határozza meg.

\$FF0C — hardware kurzorpozíció felső két bit

\$FF0D — hardware kurzorpozíció alsó bitek

A PLUS/4-ben a kurzort — a C-64-tyel ellentétben — a gép hardware-ből állítja elő — a C-64-ben ez a megszakítórutin dolgoz.

\$FF0E — 1. hanggenerátor frekvencia alsó bitek

\$FF0F — 2. hanggenerátor frekvencia alsó bitek

\$FF10 — 2. hanggenerátor frekvencia felső két bit

\$FF11 — hangvezérlő regiszter

\$FF12 — bittérkép helye, karaktergenerátor ROM/RAM kapcsoló, 1. hanggenerátor frekvencia felső két bit

Két hanggenerátorunk van, lényegesen egyszerűbbek (és kisebb teljesítményűek), mint a C-64 SID chipje, tíz biten ábrázolt frekvenciával, 8 hangerővel (hmm, erről még lesz egy-két szó), mely közös. Négyszögjelet szolgáltatnak, de a kettes fehérzajt is tud, és hála az égnek analóg üzemmód is létezik (erről még sok szó lesz).

\$FF13 — karaktergenerátor kezdőcíme, SINGLE CLOCK beállítása (ismerős, nemde?), ROM/RAM lapozás kiolvasható belőle (később...).

Mint láthattuk, a karaktergenerátor és bitmap helyének meghatározása bonyolultnak látszik. Szerencsére azonban a hasznos bitek úgy helyezkednek el az esetek nagy részében, hogy a felső bájtot egyszerűen beírva minden bit a helyére kerül.

\$FF14 — szín- és képernyőmemória (karakteres üzemmód) vagy szín- és fényerőmemória (bittérképes üzemmód) helye

Szintén bárhova könnyen áthelyezhető.

\$FF15 — háttérszínregiszter

\$FF16 — első segédszínregiszter

\$FF17 — második segédszínregiszter

\$FF18 — harmadik segédszínregiszter

\$FF19 — keretszínregiszter

Mindegyiket 121 szín (15 szín 8—8 árnyalata, plusz a fekete, melynek minden árnyalata egyformán fekete) közül választhatjuk teljesen szabadon. Az azonos fényerejű (árnyalatú) színek fekete-fehér televízió, monitoron, valamint JUNOSZTY-on egyforma szürkék, erre vigyázni kell, ha azt szeretnénk, hogy programunkat mindenki „élvezhesse”.

\$FF1A — karakterpozíció megjelenítés felső két bit

\$FF1B — karakterpozíció megjelenítés alsó bitek

A két regiszter által mutatott érték karakter soronként 40-nel nő (tudniillik 40

karakter van egy sorban). Hogy ez mire jó, azt csak a tervezők tudják(?).

\$FF1C — videósor számláló regiszter felső bit

\$FF1D — videósor számláló regiszter alsó bitek

Az éppen megjelenítés alatt álló rastersor számát tudhatjuk meg belőle. Jó grafikai hatásokat lehet elérnie vele.

\$FF1E — vízszintes videópozíció számláló felső nyolc bit

A TED hasonló belső kilenc bites regiszterének felső nyolc bitje. Az alsó bit olyan gyorsan változik, hogy programból nem figyelhető meg. Grafikus hatásoknál, és véletlenszám-generálásnál használható.

\$FF1F — villogás (kurzor, illetve villogó szövegek) számláló, karakter soron belüli rastersorszámoló

Szerepe a magyarázathoz kiderül. No comment.

Végezetül a TED még két úgynevezett „fantomregiszterrel” is rendelkezik:

\$FF3E — ROM kiválasztás

\$FF3F — RAM kiválasztás

Az \$FF3E címre bármit és bárhogy írva a ROM lapozódik felülre. Ilyenkor \$8000—\$FCFF-ig, és \$FF40—\$FFFF-ig minden olvasás ROM-ra vonatkozik (írni mindig csak a RAM-on lehet — no, vajon miért?). \$FF3F-re írva a RAM „kikapcsolódik felülre”, ilyenkor azt olvassuk \$8000—\$FCFF és \$FF40—\$FFFF között. (\$FD00—\$FF3F között ugyanis úgynevezett I/O (Input/Output: bevitel/kivitel) terület van, a gép ezen keresztül érintkezik a külvilággal). Ezen fantomregiszterekbe bármit írhatunk (leggyakrabban STA utasítást használnak), de bármely, írással járó művelet (például ROL, INC stb. megfelelő). Olvasni lehet őket, de ez nem vált ki lapozást, az eredmény pedig „zöldség”, semmi köze ahhoz, amit beírtunk.

A karaktergenerátor hossza 2k, de ebből 1k is elég a megjelenítéshez (128 karakter + inverzek), ez az alapállapot. Ezért a karaktergenerátor helyét kilobájtos felbontással tudjuk megadni (tehát nem rakhatjuk \$23F5-re, csak \$2400-ra, vagy bármely más egész kbájt kezdetére) ebben az üzemmódban. Ha azonban a megjelenítést 256 karakteres módban kapcsoljuk (nincsenek inverzek), a TED a címet csak 2k-s pontossággal érzékeli, vagyis a karaktergenerátort \$2000-re vagy \$2400-ra kapcsolva ugyanazt a képet kapjuk.



A szín- és képernyő- (szín- és fényerő-) memória hossza  $2 \cdot 1024$  bájt (2kbájt). Mivel a képernyő méretei: 25 sor, 40 karakter/sor, összesen 1000 karakter fér el rajta, a többi rejtve marad, ide egy rövid rutint is írhatunk (24 bájt nem sok, de gyakran elég). A képer-

nyő helyét 2k felbontással adhatjuk meg.

A grafikus kép (bittérkép) hossza  $320 \cdot 200 : 8 = 8000$  bájt, ami igen jelentős. Helyét 8k-s lépésekben választhatjuk ki. Egy teljes grafikus kép terjedelme 10 kbájt, hiszen a 8k bittérképhez csatlakozik 2k szín- és fénymemória. Alapállás-

ban a BASIC \$2000—\$3F3F-re rakja a bittérképet (újabb 192 rejtett bájt!), a szín- és fénymemória pedig a \$1800—\$1FFF7 területet foglalja el.

A következő számban elkezdzünk programozni: a megszakításokról lesz szó.

## HATÁRTALAN INPUT

Az INPUT parancs hiányosságai miatt rendszeresen igyekeznek azt valahogy feljavítani. Az egyszerű kérdőjel-eltüntető POKE utasítástól a komplex INPUT helyettesítő programokig mindenféle megoldással találkozhatunk. Egy egyszerű megoldást a mellékelt *INPUT unlimited* program kínál (lásd a listát).

A bevitel során a C128 minden jele használható, amelyeket egy sztringváltozóba rakunk el. Még egy INPUT# parancs szimulációjára is sor kerül. Így lemezről vagy kazettáról is átvehetünk adatokat. A 184-es cella dönti el, hogy az adatokat a billentyűről vagy egy állományból

kell-e venni. Ott áll a logikai állományszám. Ha a billentyűzetet akarjuk használni, a **POKE 184,0**-t kell beadni. Elméletileg a C128 ilyenkor az adatokat a nullás csatornából olvassa. Mivel ez nem létezik, a gép a billentyűzethez nyúl.

Az INPUT unlimited alkalmazását a programban az 1—390. sorok mutatják. A tényleges rutin a 400—480. sorokban áll. Ebben a példában tetszőleges szöveget írhatunk lemezre. Üres sorban megnyomott (RETURN) zárja a bevitelt és az állományt is. A második részben megnyitjuk az állományt, és az adatokat szintén a mi programunkkal olvassuk be.

### Az INPUT unlimited használható a C128 INPUT parancsa helyett

```

1 REM
100 T$=""
110 PRINT "AZ ALLOMANY NEVE:";:GOSUB 400
120 PRINT:DN$=T$:IF L=0 OR L>16 THEN END
130 :
200 OPEN B,B,B,"@:"+DN$+"",S,W
210 PRINT "KEREM A SZOVEGET BEADNI:";:PRINT
220 POKE 184,0:GOSUB 400:PRINT
230 IF T$<>" " THEN PRINT#B,T$:GOTO 220
240 CLOSE B
250 :
300 PRINT "AZ UJRABEOLVASAS:";:PRINT
310 OPEN B,B,B,DN$
320 T$="":DIM A$(100)
330 FOR I=1 TO 100
340 GOSUB 400:A$(I)=T$:PRINT T$
350 IF ST=64 THEN N=I:I=100
360 NEXT
370 CLOSE B
380 :
390 END
400 REM ** AZ INPUT UNLIMITED RUTIN **
410 REM
420 BANK 15:L=PEEK(186)
430 IF L=1 OR L>4 THEN SYS 65478,,PEEK(184)
440 SYS 20371:SYS 65484:PRINT " ";
450 T$="":BANK1:L=POINTER(T$)
460 POKE L,181:POKE L+1,0:POKE L+2,2
470 L=INSTR(T$,CHR$(0))-1
480 T$=LEFT$(T$,1):BANK15:RETURN

```

## C128-as tippek, trükkök

Térjünk azonban rá a program működésére:

A 420-as sorban beolvassuk az aktuális készülékszámot, amelyet az L változóba írunk. Ha ez a szám a floppynak vagy a magnónak felel meg, a C128 a CHKIN rutint használja a bevitel előkészítésére (430-as sor). A bevitel a 440-es sorban történik. A BASIN rutin olvassa be a karaktereket a definiált készülékről, amíg a 13-as kód (return) elő nem kerül. Maximum 160 karakter olvasható be így. Ha a beviteli készülék a billentyűzet, a karakterek a képernyőn is megjelennek. Az információ a BASIC beviteli pufferbe kerül a \$200-tól (768-as cella). Végjelzésként a nullakód szolgál.

A POINTER paranccsal a változók hosszát 161-re (160 plusz a nullakód), a változó tartalmára irányuló mutatót pedig a BASIC beviteli pufferre állítjuk. Az INSTR paranccsal határozzuk meg a szöveg hosszát. A LEFT\$ alkalmazásával pedig a változó a megfelelő hosszúsági értéket kapja.

## RESETVÉDELEM ELHÁRÍTÁS

Vannak olyan programok, amelyek reset védelemként a \$8004-es (32772) címtől modul szimulálnak:

```
>08004 C3 C2 CD 38 30 00 00
00:CBM80
```

A C64-es ilyen esetben resetnél arra a rutinra ugrik, amely címét a \$8000/8001-es tárolók határozzák meg. Rendszerint így indítják el a játékokat vagy az alkalmazói programokat. Ha egy ilyen programot ki akarunk lőni a tárolóból, csak a kikapcsolás segít. Vagy mégsem? A BANK 0:POKE 32772,0 a C128-as módban tönkreteszi a moduljelzést, így azt a C64-es nem ismeri föl. A GO 64-gyel tehát a C64-es módba léphetünk.



# Még többet ésszel!

## VI. rész

Őszintén remélem, hogy kevesen voltak, akik hiába próbáltak működő megoldást készíteni a legutóbbi feladványhoz. Az előző részben bemutatott fogás ismeretében ugyanis nem lehet a feladatot igazán nehéznek nevezni. Két megoldást ígértem, ezek az 1. példában láthatók. Kivételesen kihagytam a szokásos szóközöket azért, hogy a parancs beleférjen a két képernyő-sorba.

Rögtön látszik, hogy ugyanazt a relációs műveletet használtam fel két eltérő módon. Az első változatban vagy X-et, vagy I-t teszek az X változóba. Ez egyszerű, és érthető ötlet. A másik változat egy kicsit érdekesebb: megfelelő I esetén az X-et annyival növelem meg, hogy értéke I-vel legyen egyenlő. Első pillantásra talán úgy tűnik, kihagytam, hogy ilyenkor I nagyobb X-nél. Ez nem igaz, a módszer csökkenő ciklusra is jól működik.

Nehogy azt higgye bárki is, hogy ez a reláció egy sosem látott új dolog! Minden nap rendszeresen használjuk, csak egy szokványosabb környezetben, az IF utasításban. Az IF leírása általában valami ilyesmi: IF (feltétel) THEN (utasítások) vagy GOTO (sorszám). Ez a titokzatos (feltétel), mint az a további magyarázatokban következni szokott, kifejezések összehasonlításából áll. Egy ilyen leírás azonban szakszerűtlen és hiányos.

„Szubrutinként” vizsgáljuk meg, hogy a BASIC milyen ún. adattípusokat különböztet meg! Egyik a numerikus, más néven aritmetikai típus, amely lehet konstans (állandó), függvény vagy változó, ez utóbbiban skalár vagy tömb, illetve egész vagy valós. Másik csoport a karakteres (string) konstans, függvény, és skalár vagy tömbváltozó. A karakteres típus jele a dollár (\$), az egészé a százalék (%), a tömbé pedig a zárójelben levő argumentum. Kifejezés a fenti típusok akármelyike, vagy ezek és operátorok (műveleti jelek) logikus láncolata lehet. Ha tehát leírunk egy programba valamit, ami nem utasítás vagy operátor, akkor az interpreter azt kifejezésként fogja értelmezni, ha lehet. Nem vagyok igazán szabatos, de remélem, követhető. A 2. példán néhány kifejezést mutatok. És most „Return”.

Mivel a BASIC logikus nyelv, ezért az IF után csak szabályos adattípus engedhető meg; „feltétel” pedig nincs a legális típusok között. Ezért azok fogalmaznak pontosan, akik az IF-t így írják le: IF (numerikus kifejezés) THEN stb.

Bármily hihetetlen, mégis ez a helyes definíció. De hát már a múlt részben is kiderült, hogy a reláció egy számot eredményező, azaz numerikus művelet, nem igaz? Ilyen megfontolás alapján helyes lehet a 3. példa is, hiszen az ST egy numerikus változó, egyben kifejezés is. Tudom, hogy ez némelyeknek szokatlan; olvastam már olyat is, hogy egy programban az ilyen sor talán szándékosan elrejtett hiba lehet (92/7. 10. o.)

Szerencsére könnyen megjegyezhető, hogy a BASIC hogyan értelmezi az ilyen feltételt: ha nulla, akkor hamis, egyébként igaz. Tehát a 3. példában, amely, mondjuk, egy file olvasásának része, a CLOSE akkor hajtodik végre, ha az ST értéke nem egyenlő nullával. Mivel ez file végét vagy valamilyen átviteli hibát jelez, ezért a példa, persze némileg kibővítvé, éppen jó lehet nekünk. De ha ez a megfogalmazás ugyanazt jelenti, mint az 'ST<0', akkor miért nem írjuk az utóbbit? Hát ezért, mert amaz rövidebb, és az interpreter is hamarabb megérti.

Ilyen megvilágításban akkor mit jelenthet a 4. példán látható feltétel? Mivel jól tudjuk, hogy a példa helyes és megfelelő

körülmények között értelmes is, ezért természetesen ennek a feltételnek is számszerű értéket kell jelenteni. A relációról ezt már elmondtam, de már akkor utaltam bizonyos „többi” logikai műveletekre. Ezek az AND, az OR és a NOT (az 'és', 'vagy' és 'nem'), melyek kettes számrendszerbeli (bináris) számok bitjeivel végezhetőek. A továbbhaladás érdekében hadd feltételezzem, hogy aki nincs tisztában a számrendszerekkel, az máshonnan begyűjti a szükséges tudnivalókat. Emlékeztetőül: #156 = %10011100 = \$9C, használva a Commodore környezetben szokásos jelölést a tízes, kettes és tizenhatos számrendszerekre.  $156 = 1 \cdot 2^7 + 0 \cdot 2^6 + \dots + 0 \cdot 2^0$ .

Legjobb lesz, ha most rögtön kitérek a BASIC számábrázolási rendszerére, mert nagy szükségünk lesz ennek ismertetésére. A számítógép minden adatot byte-okra osztott tárolóban, kettes számrendszerbeli alakban őriz. Az egész számok tárolására mindig két byte-ot használ, ezek baloldali, legnagyobb helyiértékű bitje a 15., a jobboldali 0. számú. A 0 és 32767 közötti számok ábrázolása egyszerű, az egyenes bináris alakkal megegyező, vezető nullákkal kipótolva. A negatív, -32768 és -1 közötti egészeket a gép más rendben, az ún. kettes komplementum alakban ábrázolja. Ennek előállítását a következő: vegyük a szám abszolút értékének bináris alakját, két byte hosszúságig minden bitjét váltjuk ellenkezőjére, majd adjunk hozzá egyet. (A megadott tartományon kívüli számok bináris képe már nem fér el két byte-on.) Így lesz érthető az előző részben említett furcsaság, hogy a %1111111111111111 jelentése -1. A kíváncsiak ki is próbálhatják tudományukat az 5. példa ellenőrzésével.

Kérlek, kedves Olvasó, légy kicsit türelemmel, remélhetőleg meg tudlak győzni, hogy ez a bűvészkedés egyáltalán nem felesleges. Ha ugyanis már binárisra tudjuk váltani az egész számokat, akkor megbeszélhetjük a logikai műveletek szabályait is.

Az AND művelet elve a következő: X and Y eredménye pontosan akkor 1, ha X és Y értéke is 1, egyébként 0. Figyelem, ezek csak bitek! A BASIC két egész(re kerekített) szám között úgy végzi ezt el, hogy a számok bitjei között külön-külön hajtja végre, és eredményként az így kapott bitsorozat által jelentett számot kapjuk. Szerintem érdemes ezt azonnal ki is próbálni (6. pl.).

Az OR szabálya: X or Y eredménye pontosan akkor 0, ha X és Y is 0, egyébként 1. A művelet elvégzésének módja megegyezik az előbb leírtakkal (7. pl.). Mellesleg olyan logikai műveletet, amely két bitből egy harmadikat képez, összesen tizenhat létezik, ennek bizonyításától az idegesebbek kedvéért eltekintek. Mivel a többi 14 műveletnek a BASIC-ben általában nincs kódszava, ezért most nem is foglalkozunk velük.

Hátra van még a NOT művelet. Ez kissé kilóg a sorból, mert nem két szám között, hanem egyetlen számon lehet elvégezni; ez a negatív előjel a logikai műveletek csoportjában. A negatív számok ábrázolása kapcsán már találkoztunk vele, mivel a bitek ellenkező értékre váltása más néven maga a NOT (7. pl.).

Igen lényeges tudnivaló a műveletek prioritása, azaz rangsorolásuk a végrehajtás sorrendjének szempontjából. A BASIC a NOT-ot végzi el először, aztán az AND-et, végül az OR-t; azonos prioritás esetén balról jobbra haladva (8. pl.). Ezen a sorrenden zárójelek megfelelő közbeiktatásával tudunk módosítani.

Most már ismerjük a logikai műveleteket, de elég hülye ötletnek látszik, hogy ha egy feltételt akarunk összeállítani, akkor bitekkel kelljen vacakolni. Annál is inkább, mert a feltételekben azért a leggyakrabban relációk szerepelnek, amelyek csupán



kétféle eredményt adhatnak. Ilyenkor valóban felesleges a számoltatás, de az új ismeretek birtokában más szemmel nézhetjük az ilyen megszokott dolgokat is. És ez soha nem lehet olyan hasznos, mint egy érthetetlen, titokzatos hiba (szakzsargonban ufó) vizsgálatakor. Nézzük csak a 9. példát! A korábban elmondottak (3. pl.) hatására az első változatot lerövidítettem, és döbbenetesen észleltem, hogy rosszul működik. Az ötletemben az a hiba, hogy a gép először elvégzi a feltétel kiértékelését, azaz egy AND műveletet két változó értéke között, és csak a végeredményről dönti el, hogy igaz-e (nem nulla) vagy sem. A=1 és B=2 esetben a két változat másként fut, most már remélhetőleg teljesen nyilvánvaló okból.

A logikai műveletek szükségesek az IF-hez, de igazi értéküket akkor tapasztalhatjuk, amikor biteket akarunk megvizsgálni vagy beállítani. Például egy grafikus bittérképen a C című byte 3. bitjét bekapcsolni, a 6. és 1. számút pedig kioltani akarjuk (10. pl.). Vagy egy PRINT-et akarunk akkor, ha a 2. portra dugott joysticken lenyomták a tűzgombot (11. pl.). Ilyen esetekben létrehozunk egy olyan bitsorozatot (más néven maszkot), amellyel ki-

zárólag a kiválasztott biteket módosíthatjuk. Természetesen ilyenkor is csak a tízes számrendszerben fejezhetjük ki magunkat, ehhez kellett a számábrázolás témáját megtárgyalnunk. Azt mindenki képzeletére bízom, hányféle felhasználását tudjuk ennek a módszernek felsorolni a grafikát, zenét, perifériákat kezelő, adattömörítő vagy egyéb programokban.

Az első feladvány is erre a témakörre készült. Létezik a 14 megemléttelen művelet között egy, amellyel egy szám megadott bitjei ellenkezőjükkre kapcsolhatók, előző értékük ismerete nélkül is. Erre a műveletre a legtöbb BASIC-ben nem találunk kódszót, de ez is (mint mindegyik) megvalósítható a már megismert műveletekkel. Állítsuk át tehát az A % értékének 2., 4. és 5. bitjét egyetlen értékadó utasításban!

Második feladat: a 12. példa ciklusában hogyan lehet a 2. sort befejezni úgy, hogy a feltétel teljesülésekor a futás a 4. soron folytatódjon? Többféle megoldás lehetséges, ezek összehasonlításával fogunk legközelebb foglalkozni.

Hódi Gyula

(1.)  
 X=0:N(X)=-1:FORI=1TO500:V=N(I)>N(X):X=-I\*V-X\*NOT(V):NEXTI:PRINTX,N(X)  
 X=0:N(X)=-1:FORI=1TO500:X=X-(I-X)\*(N(I)>N(X)):NEXTI:PRINTX,N(X)

(2.)  
 -1.4        A+1        B/A        A(2↑B%+1)\*RND(C)        INT(FNA%(H(0)))  
 0        A<0        NOT(C%(D2,A+B))>=(A AND 128)<<(B OR VAL(Q1\$))  
 "ABC"        MID\$(STR\$(INT(VAL(RIGHT\$(X\$(1,X\*K),(H<B)+2))),8,A-B)

(3.)  
 100 IF ST THEN CLOSE 2

(4.)  
 100 IF I>100 OR (ST AND 192) THEN CLOSE 2: GOTO 130

(5.)  
 1179=%00000010010011011        -12879=%1↓00110110110001

(6.)  
 173 =    %00000000010101101        8291 =    %00100000001100011  
 69 =    %00000000001000101        -1147 =    %11111011110000101  
 --- AND -----        ----- AND -----  
 5 =    %00000000000000101        8193 =    %00100000000000001

(7.)  
 173 =    %00000000010101101  
 69 =    %00000000001000101        -1147 =    %11111011110000101  
 --- OR -----        ----- NOT -----  
 237 =    %00000000011101101        1146 =    %00000010001111010

(8.)  
 A AND NOT B OR NOT C AND D = ((A AND (NOT B)) OR ((NOT C) AND D))

(9.)  
 45 IF A<>0 AND B<>0 THEN PRINT "EGYIK SEM NULLA."  
 45 IF A AND B THEN PRINT "EGYIK SEM NULLA.":REM ROSSZ!!!

(10.)  
 POKE C,PEEK(C) AND 189 OR 8 :REM (189=%10111101 8=%00001000)

(11.)  
 IF NOT(PEEK(56320)) AND 16 THEN PRINT "TUZI!"

(12.)  
 1 FOR I=1 TO 1000:PRINT CHR\$(145);TI\$  
 2 GET V\$:IF V\$>" " THEN ...  
 3 NEXT I  
 4 ::



# Lexikon

## XI. rész

### Nyomtató

A nyomtatók arra szolgálnak, hogy segítségével adatokat (grafikát, szöveget, táblázatokat) papírra vagy fóliára vessünk. Mivel a nyomtatókkal szemben támasztott követelmények eltérőek, többféle technika fejlődött ki. Két fő osztályt különböztethetünk meg, mégpedig a mechanikus (mátrix-, margarétafejes-, láncos nyomtatók) és a nem mechanikus (thermo-, tintasugaras-, lézernyomtatók) készülékeket.

A házi számítógépes világban főleg a mátrixnyomtatókat részesítik előnyben, mivel ezeket a rugalmasság, az ár és a szolgáltatások tekintetében szinte nem lehet fölülmülni. Így a jelkészlet megváltoztatása, grafika nyomtatása problémamentesen megy. Egy margarétafejes készülékkel például csak szövegeket lehet kinyomtatni. Ha más jelkészlettel akarunk dolgozni, ki kell cserélni a betűkereket.

### Plotter

A plotter egy olyan rajzgép, amely a számítógép által generált jeleket papírra írja. Van mechanikus és elektronikus plotter. Utóbbi egy katódsugárral közvetlenül mikrofilmre rajzol, ez az alkalmazás azonban messze van a házi számítógépes felhasználástól.

A mechanikus plottereknél egy vagy több (a színes ábrázoláshoz) tollat mozgatunk a papír fölött, amely vagy tekerésre van csévéelve, vagy sík alapon rögzített. Végtelenített papírral a dobplotterek rajzolnak. A toll itt csak az Y irányba (a függőleges tengely mentén) mozog, az X irányú mozgást egy alkalmas mechanizmus végzi el a papírral. A laposágas plottereknél a papír rögzítve van, a tollak pedig X/Y irányba is mozognak.

### Interfész

Ha a periférius készülékek (pl. a nyomtató) és a számítógép csatlakozásai nem egyeznek meg, akkor a két készüléket egy plusz elemmel kell illeszteni. Általában megvan a lehetőség arra, hogy a szükséges átkódolásokat az interfész elvégezze.

Az interfész a legkülönbözőbb számítógépes és periféria jelek „kiegyenlítésére” képes, így a két készülék kommunikálni tud egymással. Nyomtatóinterfészszel például egy másik nyomtatót is szimulálni lehet. Ez különösen a Commodore gépek esetében jelentős, mivel sok program csak a Commodore nyomtatókkal (például az MPS801-es) hajlandó együttműködni. Speciális interfészek például rendelkeznek a Commodore jelkészlettel, így azokat olyan nyomtatók is megjelenítik, amelyek különben erre nem lennének képesek.

### Traktor

Attól függően, hogy a nyomtató milyen papírt használ, különféle továbbítási lehetőségek állnak a rendelkezésre. A végtelenített papírok esetében alkalmazott leggyakoribb eljárások a traktor, a dörzstovábbítás (frikció) és a tüskézett tengely.

A felhasználók között főleg a traktor terjedt el. Ennél az eljárásnál a leporelló-papírnak lyukacsos széllel kell rendelkeznie, amelyet azután két, vezetőtüskékkel ellátott forgó gyűrűre, vagy szalagszerű alkatrészre kell helyezni. Ezek egy sín bal és jobb oldalán találhatók, és azokat a kívánt szélességnek megfelelően be lehet állítani. A leporellópapír legbiztonságosabb továbbítási módja ez, mivel a továbbító tüskékkel erős papírfogást lehet elérni. A frikciós eljárásokkal szemben a pa-

pír a nyomtatás során soha nem ferdülhet el.

### Centronics illesztés

Ezt a Centronics cég által kifejlesztett illesztést igen sok neves nyomtatógyártó átvette, ezért azt nem hivatalos normának lehet tekinteni. Egyik jellemzője a relatív csekély hardveres ráfordítás. Két Centronics illesztéssel rendelkező készülék összekapcsolásához elegendő egy 36 eres kábel és két megfelelő kiosztású csatlakozó.

A C64 és a C128 nem rendelkezik beépített Centronics illesztéssel. Viszont műszakilag nem jelent problémát a user porton egy hasonló illesztést a megfelelő szoftverrel létrehozni. Ennél a megoldásnál egy összekötő (vagy más néven fordító) kábelen kívül (user port-Centronics) semmilyen más hardverre nincs szükség. Igen sok profi szövegföldolgozó és grafikus program eleve rendelkezik beépített szoftverrel erre a célra.

### Nyomtatópuffer, spooler

Hogy a nyomtató hosszabb szövegek, komolyabb grafikák kivitelekor ne legyen sokáig blokkolva, sok készülékbe építettek puffertárolót (angolul buffer), azaz köztes tárolót. Ez a puffer az információkat (vagy azok egy részét) fölfogja, ezáltal a számítógép ismét használható lesz további feladatok ellátására. Ha a puffer túl kicsi, vagy nincs is, akkor lehetőség van spooler felszerelésére. Egy hardveres spoolert a számítógép és a nyomtató közé kell csatlakoztatni, az ugyanúgy átveszi az adatokat, mintha az a nyomtatóba (vagy annak pufferébe) került volna. Szoftveres spoolernek azt a programot nevezzük, amelyik a számítógépben lefoglal egy szabad tárolóterületet, az adatokat ott köztes tárolással tárolja és azokat függetlenül a programtól a nyomtatóra küldi.

### Szövegföldolgozás

A szövegföldolgozás (angolul Word-Processing) az írásbeli munkák számítógéppel való automatizálását jelenti. Ehhez a gépen kívül megfelelő programra, a szövegföldolgozó programra is szükség van. A szövegföldolgozáshoz tartozik a szöveg bevitele, megváltozása, szerkesztése (editálás) de azok más programokkal való összekapcsolása is. (Utóbbira példa lehet körlevelek készítése azonos szöveggel de más-más személyi adatokkal.)



A számítógép — a szövegföldolgozó programtól függően — olyan lehetőségeket ad, amelyet a hagyományos írógépes földolgozásnál el sem tudunk képzelni. Megemlíthető itt a hibák korrigálása még a képernyőn, gyakran használatos frázisok, fejléc és lábjegyzet beszúrása a szövegbe gombnyomásra. A szöveges blokkok (szavak, mondatok, bekezdések) könnyű kezelhetősége (törlés, másolás, áthelyezés, kivágás), de a teljesen automatikus keresés és csere sem jelent problémát.

## Hardcopy

Hardcopynak nevezzük egy szöveges vagy grafikus tárolóban álló információnak papírra vagy fóliára nyomtatott másolatát.

Igen sok számítógép (s ezek között vannak házi számítógépek is) rendelkezik beépített hardcopy funkcióval, amellyel az aktuális képernyő tartalmát gombnyo-

másra nyomtatóra lehet vinni. A C64-es is rendelkezik ilyen bővített operációs rendszerekkel, de ezen kívül számos hardcopy modul is létezik.

Ha azonban az operációs rendszert nem akarjuk megváltoztatni, sem pedig modulokat dugdosni a bővítő portba, akkor szoftveres hardcopy rutinra van szükségünk. Egy ilyen programot be kell tölteni a számítógép egy szabad területére, s az gombnyomásra „kiolvassa” a megfelelő tárolót és az ott talált információkat ki-nyomatja.

## BOOT64/2

A program Commodore 128-on fut. A forráslistában a name symbolum után be kell írni a betöltendő program nevét. Ezt a programot a gép C-128-as módban tölti be és átmegy 64-es módba, ahol elindítja. A leghosszabb betölthető program 176 blokk.

A lefordított programra AUTO-BOOT-ot kell tennünk, hogy az rendszeindításkor bejöjjön. Ezt könnyedén megtehetjük: a TEST-DEMO lemezen rajta levő AUTO BOOT program segítségével. A programtípusra azt kell válaszolnunk, hogy „B”.

Gyárfás Richard

Line#	Addr	Code	Source
00001	0000		;sysdec("1800")
00002	0000		* = \$1800
00003	1800		;
00004	1800		port = 1
00005	1800		mhz = \$d030
00006	1800		run64 = 2
00007	1800		config = \$d505
00008	1800		vic = \$d000
00009	1800		prbyte = \$ff02
00010	1800		wrtxt = \$able
00011	1800		buf = \$33c
00012	1800		setlfs = \$ffba
00013	1800		setnam = \$ffbd
00014	1800		load = \$ffd5
00015	1800		open = \$ffc0
00016	1800		close = \$ffc3
00017	1800		setbnk = \$ff68
00018	1800		add = \$22
00019	1800		add1 = \$24
00020	1800		;
00021	1800		;
00022	1800		;
00023	1800	20 e7 18	jsr loadpr
00024	1803	a9 e3	lda #\$e3
00025	1805	85 01	sta port
00026	1807	a9 00	lda #0
00027	1809	85 00	sta port-1
00028	180b	a2 00	ldx #0
00029	180d	bd 14 18	loop1 lda rspace,x
00030	1810	95 02	sta run64,x
00031	1812	e8	inx
00032	1813	c9 ff	cmp #\$ff
00033	1815	d0 f6	bne loop1
00034	1817	a9 00	lda #0
00035	1819	8d 30 d0	sta mhz
00036	181c	4c 02 00	jmp run64
00037	181f		;
00038	181f	a9 7f	rspace lda #\$7f
00039	1821	8d 05 d5	sta config
00040	1824	4c 27 18	jmp gob4
00041	1827		;
00042	1827	a2 ff	gob4 ldx #\$ff
00043	1829	78	sei
00044	182a	9a	txs
00045	182b	d8	clb
00046	182c	8e 10 d0	stx vic+16
00047	182f	20 a3 fd	jsr \$fda3 ; Megszakítás inicializalo
00048	1832	20 50 fd	jsr \$fd50 ; Munkaterulet inicializalasa
00049	1835	20 15 fd	jsr \$fd15 ; Hardware IO vektorok beallitasa
00050	1838	20 5b ff	jsr \$ff5b ; Video-reset
00051	183b	58	cli
00052	183c	20 53 e4	jsr \$e453 ; Basic vektorok beallitasa
00053	183f	20 bf e3	jsr \$e3bf ; Basic RAM inicializalasa
00054	1842	20 22 e4	jsr \$e422 ; Start C64
00055	1845		;
00056	1845	a9 00	lda #0
00057	1847	8d 20 d0	sta vic+32
00058	184a	8d 21 d0	sta vic+33
00059	184d	a9 93	lda #147
00060	184f	20 d2 ff	jsr prbyte
00061	1852	a9 0c	lda #12
00062	1854	8d 86 02	sta 646
00063	1857		;
00064	1857	a0 19	ldy #stttext
00065	1859	a9 13	lda #kstttext
00066	185b	20 1e ab	jsr wrtxt
00067	185e	a2 00	ldx #0
00068	1860	bd 86 18	loop2 lda loadru,x
00069	1863	fd 3c 03	sta buf,x
00070	1866	e8	inx
00071	1867	e0 a0	cpx #\$a0
00072	1869	d0 f5	bne loop2
00073	186b	a9 0f	lda #15
00074	186d	a2 06	ldx #6
00075	186f	a0 0f	ldy #15
00076	1871	20 ba ff	jsr setlfs
00077	1874	a9 05	lda #5
00078	1876	a2 5d	ldx #optxt
00079	1878	a0 19	ldy #optxt
00080	187a	20 bd ff	jsr setnam
00081	187d	20 c0 ff	jsr open
00082	1880	20 c3 ff	jsr close
00083	1883	4c 3c 03	jmp buf
00084	1886		;
00085	1886		;
00086	1886		loadru
00087	1886	a9 00	lda #0
00088	1888	85 22	sta add
00089	188a	a9 20	lda #20
00090	188c	85 23	sta add+1
00091	188e	a9 01	lda #1
00092	1890	85 24	sta add1
00093	1892	a9 08	lda #8



```

00094 1894 85 25      sta add1+1
00095 1896          ;
00096 1896 a0 00     ldy #0
00097 1898 78      sei
00098 1899 a9 35     lda #53
00099 189b 85 01     sta port
00100 189d          ;
00101 189d b1 22     loopc lda (add),y
00102 189f 91 24     sta (add1),y
00103 18a1 a5 22     lda add
00104 18a3 a5 23     lda add+1
00105 18a5 ee 20 d0  inc #d020
00106 18a8 18      cbc
00107 18a9 a5 22     lda add
00108 18ab 69 01     adc #1
00109 18ad 85 22     sta add
00110 18af a5 23     lda add+1
00111 18b1 69 00     adc #0
00112 18b3 85 23     sta add+1
00113 18b5 18      cbc
00114 18b6 a5 24     lda add1
00115 18b8 69 01     adc #1
00116 18ba 85 24     sta add1
00117 18bc a5 25     lda add+1
00118 18be 69 00     adc #0
00119 18c0 85 25     sta add+1
00120 18c2 a5 22     co2  lda add
00121 18c4 c9 ff     cmp #fff
00122 18c6 d0 d5     bne loopc
00123 18c8 a5 23     lda add+1
00124 18ca c9 cf     cmp #cf
00125 18cc d0 cf     bne loopc
00126 18ce a5 24     lda add1
00127 18d0 85 2d     sta $2d
00128 18d2 a5 25     lda add+1
00129 18d4 85 2e     sta $2e
00130 18d6 a9 37     lda #55
00131 18d8 85 01     sta port
00132 18da 58      cii
00133 18db 20 33 a5   jsr 42291
00134 18de 20 5e a6   jsr $a60e      ; Basic-kezdet program mutato
00135 18e1 20 60 a6   jsr $a660      ; CLR
00136 18e4 4c ae a7   jmp $a7ae      ; Interpreter ciklus
00137 18e7          ;
00138 18e7 a9 00     loadpr lda #0
00139 18e9 20 68 ff   jsr setbnk

00140 18ec a9 02     lda #2
00141 18ee a2 08     ldx #8
00142 18f0 a0 00     ldy #0
00143 18f2 20 ba ff   jsr setlfs
00144 18f5 a2 00     ldx #0
00145 18f7 bd 62 19   loop3 lda name,x
00146 18fa f0 05     beq quit1
00147 18fc e8        inx
00148 18fd e0 10     cpx #16
00149 18ff d0 fe     bne loop3
00150 1901 8a        quit1 txa
00151 1902 a2 62     ldx #<name
00152 1904 a0 19     ldy #>name
00153 1906 20 bd ff   jsr setnam
00154 1909 a9 00     lda #0
00155 190b a2 00     ldx #0
00156 190d a0 20     ldy #20
00157 190f 20 d5 ff   jsr load
00158 1912 60        rts
00159 1913          ;
00160 1913          ;
00161 1913          sttext.byte 9,14,19
00162 1916          .byte 'The Dark Vision Software An AGLOSGFT',13
00163 193b          .byte 'Present',13
00164 194f          .byte 'Booting ...',13,13,0
00165 195d          ;
00166 195d          optxt .byte 'u0:m0'
00167 1962          ;
00168 1962          name .byte ' ',0 ; ide kerul a betoltendo program neve

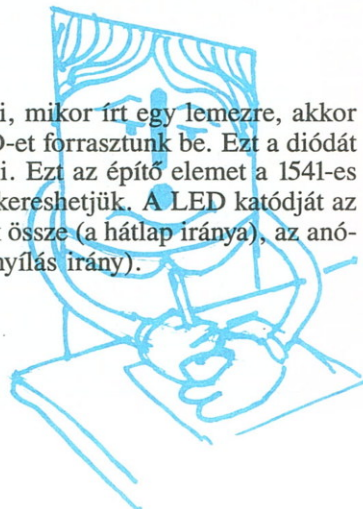
end of assembly, error count = 00000

add 0022  add1 0024  buf 033c  close ffc3
co2 18c2  config d505  gob4 1627  load  ffd5
loadpr 18e7  loadru 1886  loop1 160d  loop2 1860
loop3 18f7  loopc 189d  mhz  d030  name 1962
open  ffc0  optxt 195d  port 0001  prbyte ffd2
quit1 1901  rspace 181f  run64 0002  setbnk ffd8
setlfs ffb8  setnam ffbf  sttext 1913  vic  d000
wrtxt  able

```

## Írásellenőrző LED a floppyra

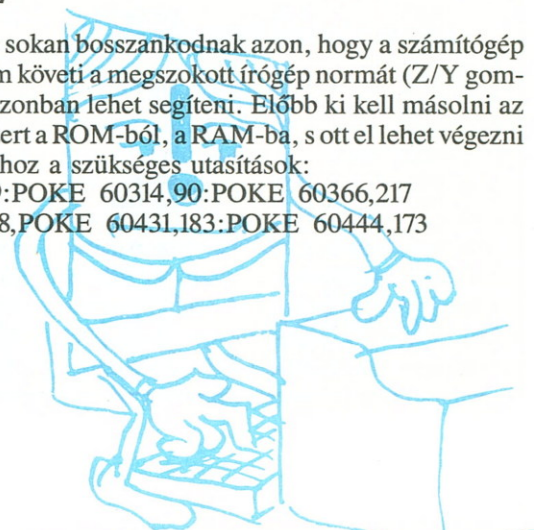
Ha szeretnénk azt tudni, ki, mikor írt egy lemezre, akkor a legjobb, ha egy írásjelző LED-et forrasztunk be. Ezt a diódát az R51-es ellenállásra kell kötni. Ezt az építő elemet a 1541-es floppy NYÁK bal oldalán elől kereshetjük. A LED katódját az ellenállás hátsó részével kössük össze (a hátlap iránya), az anódot pedig az elejével (a lemeznyílás irány).



## Billentyűk átcserélése

Lehet, hogy sokan bosszankodnak azon, hogy a számítógép billentyűzete nem követi a megszokott írógép normát (Z/Y gombot). A dolgon azonban lehet segíteni. Előbb ki kell másolni az operációs rendszert a ROM-ból, a RAM-ba, s ott el lehet végezni a cserét. Utóbbihoz a szükséges utasítások:

POKE 60301,89:POKE 60314,90:POKE 60366,217  
 POKE 60379,218,POKE 60431,183:POKE 60444,173





# 1.2. GOTO 2.0

## AMIGA

Az alábbiakban bemutatott, egyszerűen elkészíthető és aránylag olcsó (kb. 2000 Ft) hardware kiegészítés lehetővé teszi, hogy két különböző verziójú Kickstart ROM között kapcsolhassunk aszerint, hogy melyikre van éppen szükségünk.

Ezzel az eszközzel képesek leszünk, hogy a régi, már kissé elavult verziójú (1.2, 1.3.) ROM-unk mellett egy sokkal korszerűbb 2.0-as ROM-ból is bútolhassunk. Így nyerünk egy sereg új, 2.0-as Kickstart alakzatot, és ami még ennél is fontosabb: megőrizzük eddigi kompatibilitásunkat!

Persze e sok jónak ára van, mégpedig az, hogy nem leszünk teljes egészében 2.0 kompatibilisak, mivel csak a ROM lett kicserélve. Akit ez utóbbi probléma nem foglalkoztat és veszi a bátorságot egy kis barkácsoláshoz, úgy gondolom annak igazán sok élményben lesz része.

Akit érdekelnek a hardware vonatkozások is, azoknak röviden, így aktivizálódik az Amigák ROM-ja: Ezek a ROM-ok két láb segítségével aktivizálhatók. Az Amiga, ha be akar lépni a ROM-ba, akkor ezt az Output-Enable lábön keresztül teszi.

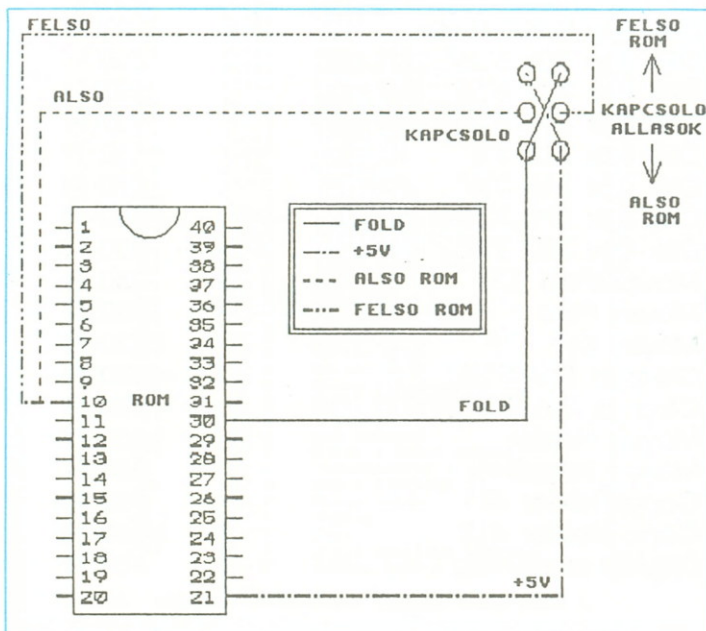
Egy általános Amigában a Chip-Select (10-es) láb 0 voltot van (tehát föld), ha ekkor aktivizálódik az Output-Enable láb, akkor a ROM is aktivizálódik, de ha a Chip-Select +5 voltot kap, akkor nem tud aktivizálódni. A mi kapcsolónk alapján nem csinál mást, mint hogy ezt a két állást (0/+5 volt) cserélgeti a 10-es lábön.

Gondolom a profik ez alapján már el is készítették a maguk kiegészítését és át is lapozták a cikket, de inkább induljunk ki abból, hogy nem mindenki ennyire profi és írjuk le a munka menetét:

*Hozzávalók egy személyre:*

1 db 40 pin-es DIP tok

1 db 2x3 pólusú kapcsoló (lehetőleg olyan, ami középen nem kapcsol ki)



4 db 30 cm-es szigetelt vezeték (mondjuk egy 4-es szalagkábel)  
1 db 2.0-as Kickstart ROM

Ez utóbbit nem szükséges azonnal megvásárolni, mert a kiegészítés működik nélküle is (persze csak régi verzióban), és ha pénzhez jutunk, vagy megjön a legújabb 2.0-as ROM szállítmány a sarki KERAVILL-ba, akkor bármikor behelyezhetjük és már működik is.

Aki nem eléggé gyakorlott forrasztási témában és ragaszkodik a saját kezű elkészítéshez, annak biztonsági okokból ajánlatos mg egy régi típusú ROM-ot is beszereznie (természetesen így már nem is olyan olcsó).

*És most lássuk a medvét:*

- Elsőként áramtalanítsuk a gépet, majd nyissuk fel a hasát és emeljük le az árnyékoló lemezt. Ezután óvatosan húzzuk ki a gyári ROM-ot (vigyázva a lábak épségére) és jól jegyezzük meg, hogy hogyan volt a foglalásban.
- Most keressük meg a 10-es lábat. Ezt úgy tehetjük, hogy kezünkbe vesszük az IC-t úgy, hogy a mélyedés a szélén tőlünk távolabbra essen, és balról számoljunk lefele 10-et.

- Most, hogy megvan, óvatosan fogjuk meg (nehogy letörjön) és hajlítsuk ki, úgy hogy vízszintesen balra mutasson.
- Ez utóbbi műveletet végezzük el a DIP token is.
- Ezután tegyük a foglalatot a kiemelt ROM hasára úgy, hogy a hajlított lábak egymás fölé kerüljenek. Nézzük meg, hogy az összes láb érintkezik-e (kivéve a meghajlítottat).

Most nyomjuk óvatosan a tokot a ROM-hoz és forrasztuk össze a lábakat (kivéve a kihajlítottakat). Nincs szükség óriási forrasztókra, épp csak akkorákra, hogy biztosak legyünk benne, hogy nem válik szét. Ügyeljünk rá, hogy ne melegítsünk sokáig, egy-egy lábat, mert ezt az IC-k nem igazán kedvelik. Ha ezzel megvagyunk vizsgáljuk meg, hogy a forrasztók nem érintkeznek egymással.

A könnyebb tájékozódás kedvéért jelöljük meg valahogyan a tokot (pl. bevágás, rovatka) azon az oldalon, ahol a ROM-on lévő kis fül van (így biztosan jól rakjuk majd be az új ROM-ot).

- Most nézzük a vezetékeket: vágjuk le a szalagkábel 15–20 cm-es hosszúságúra, bontsuk részekre és blankoljuk meg mindkét oldalát.

Ezen kívül vágjunk még le két színből 3–3 cm-es darabot és szintén blankoljuk meg.

- A kapcsolónál elsőként a sarkokat kössük keresztre a két rövidebb (3 cm-es) vezetékkel, ezután forrasztuk a többi kábelt a megfelelő helyre az alábbiak szerint:

A rövid vezetékkel azonos színű kábeleket kössük a kapcsoló egyik oldalán lévő két lábhoz úgy, hogy az azonos színű drótok érintkezzenek.

Ezután a két középső lábhoz forrasztuk oda a maradék két kábelt. Ezzel meg is van a kapcsoló.

- Tehát most van egy kapcsolónk egy kazal kábellel és egy tok-szerelvényünk külön-külön. Ami még hátra van csak az, hogy ezeket valahogyan összeillesszük:

A kapcsoló középső lábairól lejövő egyik vezetékét kössük a tok, a másikat a ROM kihajlított részéhez. A másik



- két vezetékét pedig a 21-es és 30-as lábakhoz (lásd az ábrát).
- Ezután döntsük el, hogy hová akarjuk helyezni a kapcsolót (A500-nál javasolt hely: az F1-es billentyű fölött, az AMIGA felirattal egy magasságban), ha ez megvan készítsünk ide egy megfelelő méretű lyukat.
  - Ha kéznél van egy 2.0-as ROM, akkor óvatosan nyomjuk bele a foglalatba (a ROM füle a tokon lévő jelzésünk felé mutasson!).
  - Finoman fogjuk meg az egész szerkentyűt és helyezzük el az alaplapon lévő foglaltba (ügyelve, hogy helyes irányba álljon és a lábak nem hajoltak-e el valamerre).
  - Szereljük fel a kapcsolót a tervezett helyre (figyeljünk a csusz kábelvégekre is!).
  - Még utoljára nézzük meg, hogy a ROM-ok helyes irányban állnak-e (az alaplap, az IC-k és a tok jelzése egymás fölött van), a lábak a helyükön vannak és nem érnek-e össze.
  - Most gyorsan csukjuk le a gép fedelét. Ha ezzel megvagyunk, akkor **ELKÉSZÜLTÜNK!!!**
- Figyelmeztetés:*
- A kapcsolót csak a gép áramtalanítása után használjuk (hiszen ki tudja, hogy mit fog szólni ahhoz a gép, ha a ROM-kód hirtelen megváltozik a program közepén?)!
  - Ha olyan kapcsolónk van, amelyiknek van egy középső „üres” állása vagy nincs még bent a 2.0-ás ROM, akkor bekapcsolás előtt győződjünk meg arról, hogy nem e ez van kiválasztva!

TPE

# NOVOTRADE—2C Kft. ÁRLISTA

## Hardverek

C-64 alapgép	14 900 Ft
VC 1541/II floppy drive	16 600 Ft
Datasette	3 000 Ft
Amiga 500 alapgép	49 990 Ft
Amiga 500 Plus alapgép	59 990 Ft
C 1085 color monitor	35 000 Ft
C 1802 color monitor	27 500 Ft
1351 Mouse (C-64-hez)	3 500 Ft
A 520 RF modulátor	3 900 Ft
Amiga tárbővítő	6 000 Ft
RocHard Hard disk illesztő	22 500 Ft
RocHard Hard disk illesztő + 40 Mbyte Winchester	50 000 Ft
RocGen Genlock RG300	10 750 Ft
RocGen Genlock RG300 Plus	18 125 Ft
Ethernet Card (A2000-hez)	35 000 Ft
MPS 1230 nyomtató	24 500 Ft

## C64 játékok

	kazetta	lemez
Chamonix Challenge	499 Ft	549 Ft
Eddie Edwards Super Ski	499 Ft	549 Ft
Hostages	549 Ft	599 Ft
Impossible Mission II	581 Ft	668 Ft
Rolling Twins/I want more...	—	549 Ft
Operation Neptune	—	599 Ft
Prohibition	499 Ft	—
Sim City	—	599 Ft
Space Racer	—	599 Ft
Tin Tin on the Moon	549 Ft	599 Ft
Warlock's Quest	499 Ft	549 Ft
Xonox	399 Ft	499 Ft
Import járékkazetták	549 Ft	—

## Hardverkiegészítők

2 RCA kábel	410 Ft
3 RCA kábel	575 Ft
Hálózati kábel	480 Ft
Adat kábel	360 Ft
Antenna kábel	340 Ft
Antenna váltókapcsoló	490 Ft
C64/128 Euro kábel	685 Ft
Amiga Euro kábel	1 250 Ft
Ékezetes Eprom SP180 VC	2 545 Ft
Ékezetes Eprom MPS 1230	3 500 Ft
Joy (Quick Shot II Plus)	850 Ft
Tápegység C64	3 000 Ft
Tápegység VC 1541/II	3 000 Ft
12" mono filter	700 Ft
12" color filter	750 Ft
14" mono filter	800 Ft
14" color filter	850 Ft
Disk Box 5'25" 3 db	99 Ft
Disk Box 307 5'25" 50 db	550 Ft
Disk Box 309 5'25" 100 db	750 Ft
Disk Box 310 5'25" 120 db	800 Ft
Disk Box 313 3'5" 10 db	150 Ft
Disk Box 318 3'5" 40 db	550 Ft
Disk Box 319 3'5" 80 db	750 Ft
Disk Box 320 3'5" 140 db	800 Ft
Mouse Pad 581	250 Ft
Mouse Pad + Holder	500 Ft
Mouse Set	1 000 Ft
Cleaner 562 5'25"	150 Ft
Cleaner 563 3'5"	150 Ft
Mouse Holder	250 Ft
Mouse takaró	250 Ft
Copy Holder 411	750 Ft
Copy Holder 412	1 250 Ft
Display szemüveg	750 Ft











Olcsón Commodore IC-k elektronikus alkatrészek beszerzése, C-64 és perifériáinak javítása. Tel.: 1-731-783 (üzenet).

C-64 konfiguráció eladó! Új alapgép (10 000), magnó (1500) + programkasszéták, MPS 801 printer (10 000) + új festékszalagok, szakirodalom, Action plusz cartridge (3000), joystick (2 db 800), lemeztartó (500) stb. Érdeklődni lehet a 176-2944 telefonszámon, vagy a FRIEND TWO címén (1399 Budapest, Pf.: 701/55).

C-64-re új játék- és felhasználói programok eladók. Amiga, IBM PC játékprogramok is megrendelhetők. Kérjük, tüntesse fel a gépe típusát, és a hozzá tartozó perifériákat! Bélyeggel ellátott válaszborítékért listát küldünk. (Vízhatlan csomagolás, kedvezmények!) FRIEND TWO CREW, 1399 Budapest, Pf.: 701/55.

C-64-es játékprogramok eladók kazettára. A játékprogram darabja 10 Ft. Bélyeggel ellátott válaszboríték ellenében listát küldök. Kiss László, 9024 Győr, Lajta út 22. VI/2.

Eladó egy kitűnő állapotban lévő Commodore 1541C floppy drive. A kikiáltási ár: 11 500 Ft. A floppy vásárlója ajándékba kap 5 db lemezt szuper programokkal, valamint egy lemezfejlesztítő készletet. Továbbá eladók C-64-es játéklemezek. Amíg a készlet tart. Kb. 110 db lemez van, majdnem mind '90 után megjelent játékokkal. Még egy lehetőség: egy bármilyen márkájú üres HD-s lemezért egy tele DD-st adok. És a végén egy extra ajánlat. A legelső, aki 15 000 Ft-ot ajánl, megkapja a floppyt, a 110 lemezt, a tisztítókészletet és még egy zárható mágneslemeztárt is. Cím: Gubacsi Zoltán, 4900 Fehérgyarmat, Vasvári P. u. 28.

C-64 gépi kódban tanulók! Eladók a következő könyvek: Lothar Englisch „Gépi kódú programozás a Commodore 64-esen” 241 Ft helyett 144 Ft-ért. Lothar Englisch „Gépi kódú programozás haladóknak C-64 & C-128” 319 Ft helyett 191 Ft-ért, Erdős Iván „Commodore 64 Assembly” 138 Ft helyett 82 Ft-ért. A három könyv együttesen csak 400 Ft !!! Továbbá eladó a Commodore Újság 1991-es teljes évfolyama, a könyvekhez hasonlóan 40%-os árkedvéménnyel 668 Ft helyett 400 Ft-ért. Minden könyv és újság kitűnő állapotban van! Cím: Gubacsi Zoltán, 4900 Fehérgyarmat, Vasvári P. u. 28.

Keresek C-16-os játékprogramokat és segédprogramokat. Postaköltséget vállalom. Bencsik Ferenc, 1202 Budapest, Szokolca u. 17.

Eladó C-128 + 1571-es garanciális floppy. Érdeklődni 18 óra után a 46/328-938-as számon lehet.

Eladó: Commodore VC-20 + magnó + 8 eredeti programkazetta + 1 lemez programokkal (kb. 100 db) + 3 játékcartridge. 1224 Budapest, VII. u. 4. Tel.: 2-272-260.

Eladó: 1531 hangosított magnó C-64 tápegység TDK Maxell 3M lemezek, az ármegegyezés szerint. 1224 Budapest, VII. u. 4. Tel.: 2-227-260.

Használhatatlan C-64 + 1541-es floppyt + MPS 802-803-as nyomtatót, monitort vásárolnék. Gerber, tel.: 72/35-594.

C-64, 128 programok Évától! Ár: 20 Ft/lemezoldal. Új No-Name lemezzel együtt 70 Ft/2 lemezoldal. Tájékoztatóhoz és listához szükséges 1 db lemez és 20 Ft-os bélyeg, vagy lemez nélkül 50 Ft. Kopácsi Lajosné, 1031 Budapest, Vízimolnár u. 2. X/95.

AMIGAProgramok Évától! Ár: 25 Ft/lemez. Új No-Name lemezzel együtt 80 Ft/lemez. A tájékoztatóhoz és listához szükséges 1 db lemez és 20 Ft-os bélyeg, vagy lemez nélkül 70 Ft. Kopácsi Lajosné, 1031 Budapest, Vízimolnár u. 2. X/95.

Itt a megújult OTHIS Software! Csak az ár maradt a régi! 50 Ft ellenében az országban egyedülálló képes katalógust küldünk! Ez nem csak egy egyszerű lista! Rendkívül kedvező klubtagsági feltételek, melyek tovább csökkentik az amúgy is kedvező árakat. SÓT! Karácsonyra CD-lemezjátst és egyéb értékes nyereményeket nyerhetsz! Ezenkívül Amiga zenéket tartalmazó kazetták és az országban egyedül Amiga demókat tartalmazó videokazetta! A katalógust a pénz megérkezése után azonnal postázzuk. Utánvétellel nem tudunk katalógust küldeni. Válaszboríték nem kell! C-64, lemez. Meglepetések, ajándékok, kedvezmények! OTHIS Software, 2601 Vác, Pf.: 342. OTHIS-Társ a játékban!

576-ban megjelent C-64-es programok eladása legolcsóbb áron, lemezen, kazettán. Lemezes programok kazettás változatai. Deutsch Szabolcs, 8749 Zalakaros, Fő u. 19.

C-64-es programok cseréje, eladása lemezen, kazettán. 1 lemezoldal 15 Ft, 1 kazettás program 10 Ft, utántöltős 80 Ft/db. Deutsch Szabolcs, 8749 Zalakaros, Fő u. 19.

Totókulcskészítő programok C-64-re (8 Kbyte-osak). Válaszborítékért tájékoztató. Nagy Miklós, Újfehértó, Pf.: 41.

Általános iskolai 4. oszt. német oktató- és szótárprogramok megrendelhetők csak lemezen C-64-re. Földesné, Szolnok, Győrffy J. u. 12. I/7.

Programkészítést C-64-re lemezen vállalom, levelezni tapasztalatcsere céljából. Földesné, Szolnok, Győrffy J. u. 2. I/7.

C+4-hez eladó új 1551-es japán floppy 9800 Ft. Hosszú Kálmán, 3300 Eger, Szabó S. u. 9.

C+4 - C-64 - 16 - Amiga programok lemezen, kazettán! Listát küldök! Mészáros György Tibor, 1221 Budapest, Anna u. 21. Tel.: 2-262-495.

HI C-64-esek! Szuperprogramok disk/magno Pl.: Street Rodpirates, olcsó árak! Kangaroo soft. Tóth Gábor, 5919 Pusztaföldvár, Rákóczi u. 43.

C-64-es sikerprogramok eladók (floppy, tape) Pl.: Street Rodpirates Kangaroo software. Tóth Gábor, 5919, Pusztaföldvár, Rákóczi u. 43.

C-64-re új és régi programok eladók. Válaszborítékért listát küldök. Kozma Gergely, Budapest, Szentkirályi u. 7. VIII/114.

Keresem a Behatolás a bázisra és a Lords of Chaos C program leírását. Cím: 9400 Sopron, Jegenye sor 5. Tel.: 99/28-826.

Helló C-64 tulajdonosok! Eladom vagy elcserélem lemezes és kazettás programaimat. Ezenkívül még keresek utántöltős programokat kazettára, itt csak csere érdekel. Válaszborítékért lista. Cím: 9400 Sopron, Jegenye sor 5. Tel.: 99/28-826.

Eladom PC AT-hoz a Windows 3.0 programot vagy elcserélném C-64-es hangdigitalizálóra, vagy törlőcartridge-re. Tel.: 99/28-826.

Programkészítést rendelésre vállalom C-64-re lemeze. Könyvelési és más programok rendelhetők. Borítékot kérek! Földesné, Szolnok, Győrffy J. u. 12. I/7.



## ACOMP Kft.

### NOVEMBERI

60 Ft-os

### vásárlási utalványa

Beváltható készpénzes vásárlás esetén a 1141 Budapest, Álmos vezér útja 17. szám alatti üzletben.

Érvényes: 1992. november 30-ig.

## MAKROVILÁG utazási iroda

### Beváltható utazás megrendelése esetén

az Üllői úti főirodában az alábbiak szerint:  
5 000 Ft-ig — 200 Ft kedvezmény  
10 000 Ft-ig — 400 Ft kedvezmény  
20 000 Ft-ig — 500 Ft kedvezmény  
20 000 Ft felett — 1000 Ft kedvezmény  
Csoportok jelentkezése esetén további kedvezményekről az irodában lehet tárgyalni

### NOVEMBERI

### 60 Ft-os vásárlási utalvány

Beváltható készpénzes vásárlás esetén a 2C Áruházban. Bp. XIII., Balzac u. 35.

Érvényes: 1992. november 30-ig.

## HOBBI ELEKTRONIKA

### NOVEMBERI

### vásárlási utalványa

Értéke:

**5000 Ft-ig 80Ft,  
5000 Ft felett 10%**

Beváltható a Hobbi Elektronika Kft.-nél. Budapest VII., Dózsa György u. 16. Telefon: 122-8892  
Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

A NOVOTRADE SZERVÍZ Kft. az alább felsorolt szervízeiben mindenféle szervízzolgáltatás munkadíjából 10% kedvezményt ad az egyesületi tagoknak.

1053 Budapest, Magyar u. 12—14	Telefon: 117-3551
1083 Budapest, Szigony u. 9.	Telefon: 134-3153
1191 Budapest, Gábor Á. sétány 3.	Telefon: 127-4763
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:  
9024 Győr, Babits M. 75. Telefon: 76-23-720  
6000 Kecskemét, Széchenyi tér 1—3.

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

**NOVOTRADE**  
SZERVÍZ Kft.

## Az Országos Commodore Egyesület szolgáltatásai

### Egyesületi tagoknak 20% kedvezmény:

VC—20 memóriabővítés 3—27 kByte-os:	kiépitéstől függő
C—16, C—116 memóriájának bővítése 64 kByte-ra:	3500 Ft
C—16 belső 16 kByte-os EPROM bővítés:	1450 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	2800 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	2000 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	3200 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4 FÉK C—16, C—116, +4 potméteres sebességváltoztatás 0%-tól 100%-ig fokozatmentesen	5000 Ft
TTL IC-teszter (Cartridge+lemezen a program)	2000 Ft
+4, C—16, C—116 UNI—ROM modul különféle kiépítésekben:	4300 Ft
— 8 kByte SOFT—ROM	3400 Ft
— 16 kByte SOFT—ROM	4000 Ft
— 8 kByte SOFT—ROM 16 kByte EPROM	4400 Ft
— 16 kByte SOFT—ROM 16 kByte EPROM	5000 Ft
— 16 kByte EPROM	2200 Ft

### Egyesület tagoknak 30% kedvezmény:

Speeddos (átkapcsolható) operációs rendszer beépítése (C64 átalakítás, lemezegység átalakítás + párhuzamos kábel)	5000 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése	900 Ft
PAGEFOX magyar ékezetes kiadvány-szerkesztő cartridge (a teljes A/4-es oldal kinyomtatásához 640 pont/soros nyomtató szükséges minimum, pl. Citizen 120D)	5500 Ft
FASTLOAD cartridge (lemezes gyorstöltő, másoló, monitor)	1500 Ft
TTL IC-teszter cartridge + program	4300 Ft
288/256 Kbyte-os eprombank (vezérlő eprommal)	5000 Ft
Epromégető (2716-tól 27512-ig)	5000 Ft
C64-hez tároló oszcilloszkóp	8000 Ft
C64-bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
256 K RAM-diszk (256 Kbyte RAM-mal)	14000 Ft
256 K RAM-diszk (64 Kbyte RAM-mal)	9000 Ft
2×64 Kbyte-os cartridge igény szerinti programokkal feltöltve	4300 Ft
64 Kbyte-os cartridge igény szerinti programokkal feltöltve	3000 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az eprommal együtt egységiesen	700 Ft
Árainkat az alkatrészárak változásai befolyásolhatják.	
A fenti bővítések megrendelhetők levélben az O.C.E. címén, valamint személyesen a havonta rendezendő klubdélelőttön, ahol rendszeres bemutatót is tartunk.	

Postázás esetén 100 Ft postaköltség kerül felszámításra.

A kedvezmény igénybevételéhez az O.C.E. tagsági igazolvány bemutatása szükséges.



# Cserélhető lemezes winchester!

A SyQuest Technology  
hivatalos magyarországi disztribútora:

**NOVOTRADE**  
SZERVIZ Kft.

Cím: 1053 Budapest, Henszlmann I. u. 9.  
Telefon: 117-4144 Telefax: 117-9692



<b>SQ 555 (meghajtó)</b>	<b>39 900 Ft</b>
<b>SQ 400 (44 MB/lemez)</b>	<b>8 400 Ft</b>
<b>SQ 5110 (meghajtó)</b>	<b>61 400 Ft</b>
<b>SQ 800 (88 MB/lemez)</b>	<b>11 900 Ft</b>

Áraink végfelhasználói árak és ÁFA-t nem tartalmaznak.

#### Meghatalmazott dealereink:

Microteam Kft., 1145 Budapest, Róna u. 127. Tel./fax: 184-1226	BX-Next Kft., 3434 Mályi, Bercsényi út 50. Tel.: 46/91-117
Professzionál Kft., 1033 Budapest, Kaszásdűlő u. 5. Tel.: 167-0024 fax: 167-0289	Onyx Szoftver Kft., 1118 Budapest, Mányoki u. 14/B. Tel.: 165-3325
Professzionál Kft., Miskolci Kirendeltség, 3525 Miskolc, Szabó L. u. 37. Tel.: 46/56-079	Novotrade PC Kft., 1136 Budapest, Sallai u. 25. Tel.: 149-0798 Fax: 131-0734
Professzionál Kft., Békéscsabai Kirendeltség, 5600 Békéscsaba, Andrássy u. 75. Tel.: 66/28-584	Almárium Kft., 1137 Budapest, Pozsonyi u. 21-23. Tel.: 111-2830 Fax: 112-3647
Korall Kft., 2800 Tatabánya, Március 15. út 3. Tel.: 34/11-714	Microchip Kft., 8000 Székesfehérvár, Élmunkás u. 47. Tel.: 22/25-514
3S Computer Kiszövetkezet, 6723 Szeged, Kemes u. 6. Tel.: 62/26-277, Fax: 62/26-347	Navigátor Kft., 4400 Nyíregyháza, Kórház út 26/B. Tel./fax: 42/41-972
Volánelektronika Vercomp Kft., 9024 Győr, Dr. Petz Lajos u. 7. Tel./fax: 96/12-520	Alfadat Kft., 2803 Tatabánya, Tóth-Bucsocki út 12. Tel.: 34/10-234, 10-405, Fax: 34/10-729
Novotrade Miskolc Kft., 3530 Miskolc, Vörösmarty u. 51. Tel./fax: 46/49-489	Digitech Kft., 7101 Szekszárd, Rákóczi u. 6. Tel.: 74/16-874
Elektrosoft Kft., 5000 Szolnok, József A. u. 6-8. Tel.: 56/42-880, Fax: 56/44-222	Interface Kft., 1116 Budapest, Hunyadi J. út 162. Tel.: 166-5322/58, 55, Fax: 226-3793
Számadó Kft., 6000 Kecskemét, Dózsa Gy. u. 29. Tel.: 76/21-455, Fax: 76/21-462	Netrend Rt., 1089 Budapest, Elnök u. 1. Tel.: 113-8217, Fax: 113-9537
	Omnis Kft., 2840 Oroszlány, Münnich F. u. 23. Tel.: 34/60-832
	2R Periféria Kft., 1071 Budapest, Peterdy u. 30. Tel.: 122-3034 Fax: 142-3308
	ProComp Kft., 8900 Zalaegerszeg, Bíró M. út 8. Pf.: 275. Tel.: 92/11-373

A magyar olimpiai csapat arany fokozatú támogatója







# ACOMP

## AKCIÓ

Amiga 500

~~44 900 Ft helyett~~

**40 900 Ft**

Amiga 500+

~~54 900 Ft helyett~~

**46 900 Ft**

Amiga 600

~~54 900 Ft helyett~~

**46 900 Ft**

Commodore 64

~~14 600 Ft helyett~~

**12 900 Ft**

**ACOMP Számítástechnikai Kft.**

1141 Budapest, Álmos vezér útja 17. Tel.: 183-1817, Fax: 251-2523

Áraink az ÁFA-t tartalmazzák!

Nyitvatartás 9-től 18 óráig, szombaton: 9-től 12 óráig.

Vidéki vásárlóknak utánvételes csomagküldő szolgálat!