

commodore

IX.évfolyam
1994/2

újság

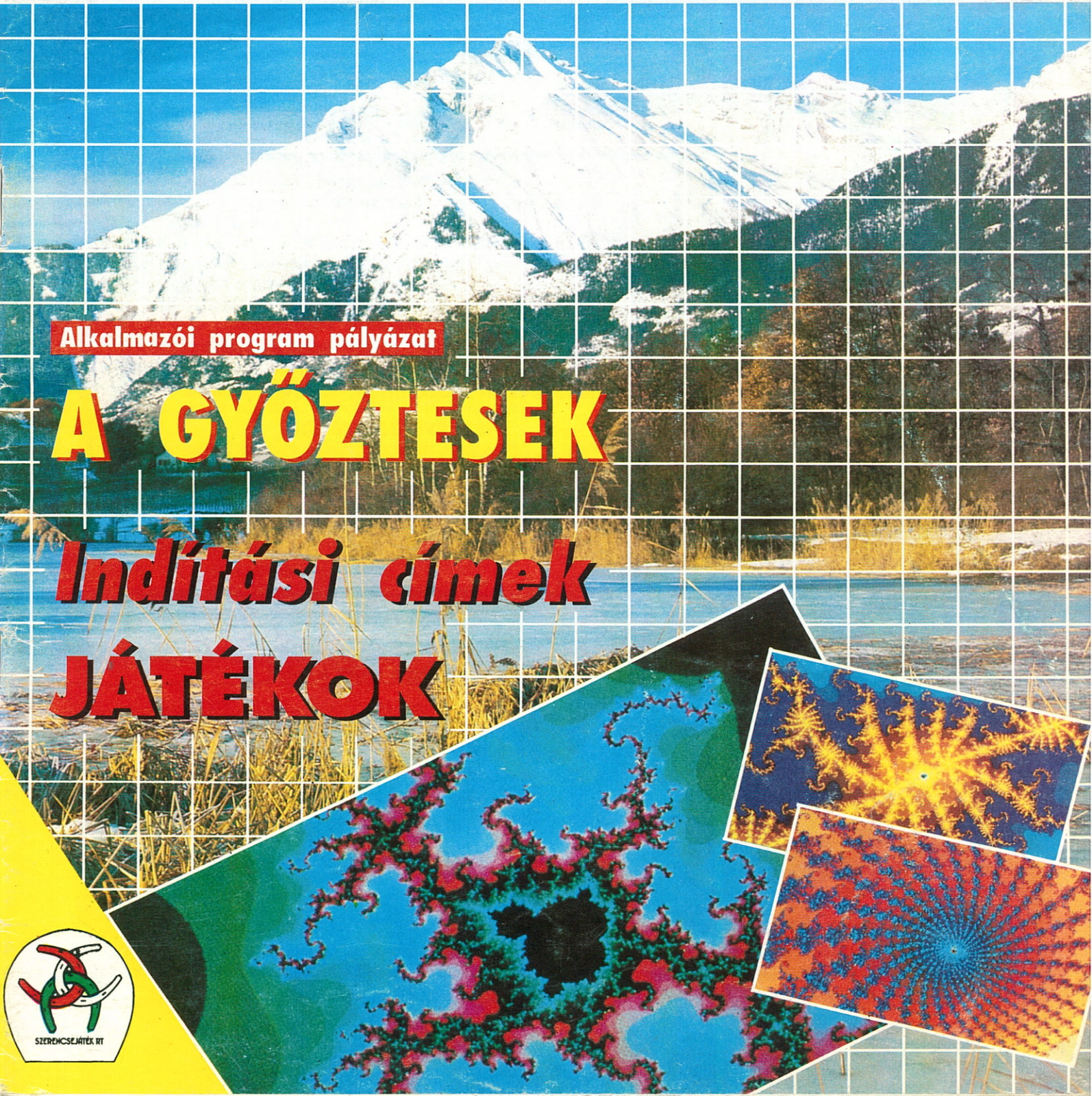
Az Országos
Commodore
Egyesület lapja

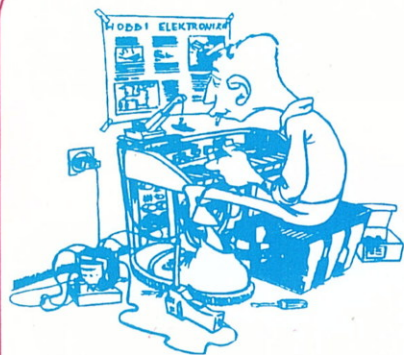
Alkalmazói program pályázat

A GYŐZTESEK

Indítási címek

JÁTÉKOK





HOBBI ELEKTRONIKA

**Super
ajánlat!**

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)

KÍNÁLATUNKBÓL

SZÁMÍTÁSTECHNIKA:

C64 BŐVÍTŐK		
	egys.	élesztve
PLOFI Datassette cartridge	1.200 Ft	1.500 Ft
PLOFI Help + cartridge	1.200 Ft	1.500 Ft
PLOFI FASTLOAD cartridge	1.200 Ft	1.500 Ft
PLOFI FASTLOAD + SPEEDTAPE	1.500 Ft	1.700 Ft
PLOFI SIMON'S cartridge	1.500 Ft	1.700 Ft
PLOFI JÁTÉK cartridge	1.500 Ft	1.700 Ft
ACTION REPLAY MK7 cartridge	—	3.450 Ft
ATOMIC POWER cartridge	—	3.450 Ft
FINAL III. cartridge	—	3.450 Ft
PAGEFOX DTP cartridge	3.200 Ft	3.900 Ft
MINI EPROMBANK cartridge	2.300 Ft	2.900 Ft
EPROMBANK 256 Kb-át	2.900 Ft	4.000 Ft
C64 DOKTOR teszt cartridge	2.300 Ft	3.000 Ft
EPROMÉGETŐ (2716-27512-ig)	3.400 Ft	4.500 Ft
IC TESZTER (kb. 180 TTL IC)	2.700 Ft	3.900 Ft
256K RAM BŐVÍTŐ RAM floppi	—	11.800 Ft
FÉNYCERUZA + szoftver	950 Ft	1.450 Ft
HANGDIGITALIZÁLÓ + szoftver	1.350 Ft	1.700 Ft
HANGKAPCSOLÓ + szoftver	700 Ft	990 Ft
DATASSETTE fejbeállító	390 Ft	—
CPU STOP + RESET	400 Ft	—
FREKVENCIAMÉRŐ 30 MHz	1.300 Ft	1.800 Ft
I/O BŐVÍTŐ 8255-tel	1.400 Ft	1.800 Ft
JOYSTICK automata	1.500 Ft	1.900 Ft
USER CENTRONICS kábel	750 Ft	—

IBM BŐVÍTŐK		
	egys.	élesztve
IBM IC TESZTER + szoftver	5.900 Ft	9.800 Ft
IBM EPROMÉGETŐ + szoftver	4.800 Ft	9.900 Ft
IBM EPROMÉGETŐ tápegység	1.200 Ft	↑ benne
IBM CODEC HANGKÁRTYA	3.800 Ft	4.800 Ft

IBM KOVOX	—	700 Ft	TANGÓ PLD jogtisza szoftver. <i>Kéjen árajánlatot!</i>
IBM AKTÍV KOVOX	1.600 Ft	—	TANGÓ NYÁKTEVEZŐ jogtisza szoftver
IBM I/O KÁRTYA 48 vonal	3.500 Ft	—	TTL, CMOS, TRANZISZTOR, DIÓDA, MEMÓRIA
IBM GAL PROGRAMOZÓ	6.500 Ft	8.500 Ft	katalógus floppylemezen, egységesen: 600 Ft/db

C-64 BÖRZE

Használt C-64-et, flopit adok-veszek.

Hibás gépet beszámítok, megvásárolok.

Megegyezés szerint

új IBM PC részegységre cserélek.

Áraink a 25% ÁFA-t tartalmazzák!

Szaküzletemben vásárolhat a Commodore Újság HOBBI ELEKTRONIKA rovatában megjelent hardverleírásokhoz (működő minta alapján) panelt, egységcsomagot, vagy készre szerelt áramkört.

MŰSZERVÁSÁR

Bontott anyagok,
használt műszerek
bőngésződéje



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.



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 1020 Ft.

A **PLUSZPÁHOLY** tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 2350 Ft.

A **SZUPERPÁHOLY** tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15×3 db vásárlási utalványt is. Az éves tagsági díj 24 000 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 300 Ft. 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 az 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.

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

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

Index: ISSN 0237-756 X

Terjeszti a Magyar Posta

Megvásárolható a hírlapárusoknál

93.0335 MSZH Nyomda és Kiadó Kft., Budapest

Felelős vezető: Nagy László

Tagdíjfizetés

Kedves Tagtársaink!

Egyesületünk közzéteszi, hogy a realitásokat figyelembe véve, mértéktartóan bár, de kénytelenek vagyunk tagdíjunkt február elsejétől megváltoztatni.

Deákpáholy 1 020 Ft

Pluszpáholy 2 350 Ft

Szuperpáholy 24 000 Ft

Országos Commodore Egyesület

C-64 C+4 AMIGA PC tulajdonosok!

Várunk benneteket az Országos Commodore Egyesület klubdélelőttjén a Havanna Közösségi Házban minden hónap 3. vasárnapján, délelőtt 9 órától. Gépet és hosszabítót hozzatok!

Cím: 1181 Budapest, Kondor Béla sétány 8.

Megközelíthető: a metró kőbánya-kispesti végállomásától a piros 136-os busszal.

Legközelebb:

február 20-án.

Tisztelt Szerkesztőség!

Az alábbi kis program a képernyő tartalmát lefelé tolja (SCROLL) el a megadott sorszámmal. Érdekessége az, hogy a képernyő felfelé gördül, de ami fent kilép, az alul újra belép a képernyőre és a kívánt helyen áll meg. A hibátlan működés feltétele az, hogy a képernyő nem tartalmazhat olyan sorokat, amelyek 39-nél több karakterből állnak, azaz amelyeknél soremelést történt.

A program:

100 X = 5:REM SRCOLLOZANDÓ SOROK (1—23)

110 FOR I = 1024 TO 1063 : POKE I + 960,PEEK(I)

120 NEXT I

130 FOR V = 1TO24-X:SYS59626:SYS59848:

140 NEXT V

150 FORE = 0TOX-1:POKE 781,E:SYS59903

160 NEXT E

170 SYS 58726

A 110—120. sor a képernyő első sorát az utolsó sorba írja, enélkül a „HOME” funkciót tölti be, a nem kívánatos soremelést elkerülésére, adott esetben elhagyható.

Ha a 150—170. sorokat elhagyjuk és helyettük a következőt írjuk be:

150 GOTO 130

egy végtelen felfelé gördülést kapunk.

Egy megjegyzés: Véleményem szerint ez a program-morzsa BASIC nyelven készült. Ezt azért írom, mert csodálkozva olvas-

tam, hogy a játékpályázaton az egyik programnál erősen kifogásolták, hogy az egy SYS utasítást tartalmazott. Megnéztem a pályázati felhívást és úgy láttam, hogy ezt többféleképpen lehet értelmezni, de ezek közül csak egynek van értelme. Ugyanis, ha szigorúan vesszük, hogy a program nem hívhat meg „gépi kódú programrészeket”, akkor az összes pályamunkát el kellett volna utasítani. Ugyanis tudomásom szerint minden BASIC utasítás egy sor „gép kódú programrészt” hív meg, sőt már egy gomb lenyomása is ezt teszi. A SYS csak annyiban különbözik a többi-től, hogy annak egy címet is adhatunk. Ha csak SYS-szel nem szabad meghívni, akkor azt kell írni, hogy: „SYS-t nem tartalmazhat a program”. Legközelebb aztán lehet olyan pályázatot kiírni, amelyik pl. a FOR—NEXT ciklust tiltja meg.

Egyszóval, szerintem az az elfogadható értelmezés, hogy a program nem tartalmazhat, nem állíthat elő és nem hívhat meg olyan gépi kódú programrészeket, amelyeket a gyári C64 (vagy C+4) nem tartalmaz.

Tisztelettel:

Pályi Sándor

Tisztelt Szerkesztőség!

Küldtem a címükre egy levelet, amelyben képernyő lefelé görgetéséről írtam, hamar rájöttem, hogy elhamarkodtam. Nem tudom, hogy ennek van-e köze ahhoz, hogy egy kis cikk jelent meg „Képernyőgörgetés” címen. Ebben leírnak egy három utasításból álló módot, amivel lefelé lehet görgetni a képernyőt. Amikor megláttam, igen csodálkoztam, mert ugyanezt én is próbáltam már, de használhatatlannak találtam. Igaz ugyan, hogy egy sorral lejjebb vitte a képernyő tartalmát, de ha utána a kurzort a 0. sorba akartam felvinni, az 1. sorig még fel lehetett tornáztatni, de onnan visszaugrott a 4. sorba, és onnan csak lefelé volt hajlandó menni. Ha a következő programot írtam be:

```
10 ?"1. SOR"
20 ?"2. SOR"
30 ?"3. SOR"
```

40 ?"4. SOR"

50 POKE 214,255:SYS 59749:SYS 58640

60 ?CHR\$(19) "0. SOR"

akkor RUN parancsra a 0. sor üressé vált, utána kiírta az 1., 2., 3. sort, de a 4. sor kiírása helyett azt átírva a 0. sor jelent meg ahelyett, hogy a 0. sor kiírása a 0. sorba ment volna.

Amikor az újságban láttam a dolgot, csak arra gondolhattam, hogy hibás a gépem. Kipróbáltam egy másik gépen is (az C-64 II. típusú volt), hajszálpontosan ugyanúgy viselkedett. Mivel annak a valószínűsége, hogy két különböző gyártási sorozatból származó, kifogástalanul működő számítógépnek ugyanaz, a többi gépre nem jellemző hibája lenne, igen kicsi; nagy a valószínűsége annak, hogy az Önök által közölt eljárás hibás, használhatatlan.

Azt már csak a magam megnyugtatóására írom meg, hogy a beküldött verzió azért volt elhamarkodott, mert az első sor másolása az utolsóba nem túl elegáns. Jobb is van nála. Jelenlegi verzióm:

```
100 SYS 59777
```

```
110 FORI=OTO24-X: SYS59626: SYS59848
```

```
120 NEXTI
```

```
130 FORI=OTOX-1: POKE781,I: SYS599903
```

```
140 NEXTI
```

Az X-et, tehát azt az értéket, hogy hány sorral akarjuk legörgetni a képernyőt, természetesen meg kell adni.

A 100. sor ugyanazt csinálja, mint az Önök három utasítása, csak rövidebb, ugyanazzal a hibával. A hiba megszüntetésére eddig csak a SYS 59626 utasítást találtam, tehát meg lehet csinálni, hogy eggyel több sorral vittem le a képernyőt, aztán egyet visszaemelek rajta, de ekkor az utolsó sor elvész.

A 110—120. sor gördít, a 130—140. sor törli a felesleget. Így azért működik, mert indításkor eggyel lejjebb viszem a sort, tehát a 0. sor 1. sorra változik, így a felfelé irányuló legördítésnél nem vész el. Ez a programocska vagy szubrutin még mindig tökéletesen működött, de ha előtte az Önök által közölt utasítást használták, ajánlatos legalább egy resetet csinálni.

Tisztelettel:

Pályi Sándor

Tisztelt COMMODORE ÚJSÁG!

A PLUS/4 MAGYARUL NYOMTAT programok fejlesztése során találkoztam néhány problémával, melyeknek megoldása talán másoknak is tanulságos lehet, hasonló munkáknál.

Nem tartozik a programokhoz, de talán érdemes tőlük függetlenül közölni az alábbi — mások számára is — hasznosítható „melléktermékeket”.

1; A program használja a \$F400-tól kezdődő RAM területet, ezért a BASIC végcímet (\$37—38) át kell írja \$F000-ra, saját védelmére. Mivel eredetileg BASIC területre töltődik, futtatás után ezt fel kell szabadítani. Ez NEW rutinnal lehetséges, de itt vigyázni kell, mert előtte a Z bitet 1-re kell állítani pl. LDA#\$00-val.

Nekem ennek ellenére általában BREAK-kel tért vissza. Ezért inkább a JMP 8000-et („hidegstart”) választottam, ami tökéletes, csak az előzetesen beállított végcímet (\$F0) visszaírta \$FD-re. Kiderült, hogy a végcímet nem a \$37—38-ba, hanem a \$0533—0534-be kell beírni. A hidegindítás rutin ugyanis innen veszi ki a \$37—38-ba beírandó értéket.

2; A ROM mögötti RAM lapozásához, a megszakítást le-

kell tiltani, de minden visszatérés után, (STA \$FF3E) engedélyezni kell. Ez ismert tény. Ezt azonban akkor is meg kell tenni, ha az átírt — RAM-ban futó — rutin ROM-ból volt meghíva és RTS-sel tér vissza. Ilyenkor az eljárás: RTS helyett, JMP alsó RAM-ba (pl. \$065E) majd ott; STA \$FF3E:CLI:RTS, és így biztosított a zavartalan visszatérés.

Ez nekem konkrét hibát okozott. Az átírt billentyűdekódoló rutin részlet, tartalmazott egy RTS-t arra, hogyha Control/S billentyű kombináció (kiírás megállítása) jön. Ekkor „bolondult” meg a gép. Utólag kiderült: NEKI volt igaz. A fenti megoldással minden rendbejött.

3; A fent leírt hiba „guberálása” közben számtalanszor megnéztem a PLUS/4 belső felépítése című könyv \$DBA6-DBAE részletét. Itt az szerepel szövegben: Ha a beolvasott karakter nem <CR> akkor ugrás. Az ehhez tartozó CPX #\$0D valóban a CR (dec13) kódja. Mint később a két könyv (PLUS/4 ROM lista) összevetésével kiderült, ez megtévesztett engem, de (sajnos) a könyv íróját is. Ebben az állapotban ugyanis még nem a billentyű CHR\$(x) értéke szerepel, hanem a mátrix kódja, ami ebben az esetben véletlenül (CTRL/S) azonos a RETURN-nel. Ezért ha valaki forgatja ezt a könyvet, kérem sürgősen javítsa ki a \$DBAB sor szövegében a <CR>-t CTRL+S-re.

4; Talán feltűnt valakinek a program nézegetése közben hogy rengeteg veremből ki utasítás (PLA stb.) fordul elő. Természetesen ezt megpróbáltam egy szubrutinba ágyazni, de mivel az is verembe írja a visszatérési címet, teljesen megkeverte a kivett adatokat. Ezért inkább feladtam, és a többszöri beírást választottam.

Hát ennyi az, amiről úgy gondolom hogy mások is okul-

hatnak. Nem vagyok profi programozó, tehát lehet hogy alapvető hibákat is elkövettem, de hosszú kínlódás után mégiscsak működik az, aminek működni kell. És ezt tartom fontosnak, meg — számomra — sikerélménynek. Ha ezenkívül még hasznos is, az csak előny.

Tisztelettel
Novák György

Tisztelt Szerkesztőség

Mellékelek egy levelet (az appendixekkel együtt) író és olvasó társamnak Szabó Balázsnak. A C-Újság 1992/7—8. számában kér segítséget MPS 1230-as nyomtatójával kapcsolatban. Talán a neki szánt soraim másokat is érdekelhetnek. Különösen a 3. lista, amelynek assembly programja a C64, és a hozzá csatlakoztatott MPS 1230-as nyomtató ékezetesítését oldja meg.

Ha a 6. listához fűzzük a datas programot (csak adatokat tartalmaz), máris rendelkezésünkre áll egy komplett BASIC betöltő program a fenti feladat végrehajtására. Betöltés után, a SYS 49516 parancsra a ROM-ba másolja a karaktergenerátor kisbetűs/nagybetűs részét, majd „helyükre rakja” az ékezetes karaktereket inverzükkel együtt.

A SYS 49796 hatására pedig az EPSON FX 80 emulációnál teszi ugyanezt. Mindkét programrész a mutatók szükséges átállítását is elvégzi.

A program gépikódú és futtatható változata, a datas programmal együtt megtalálható a lemezen.

Kedves Szabó Balázs!

Sajnos a PLUS/4-et nem ismerem, hogy mégis vállalkozom problémájának megoldására, azt csupán az indokolja, hogy könyítok valamit a gépi kódú programozáshoz, valamint nekem is MPS 1230-as nyomtatóm van. (E levelet is „vele” írom.) És végül de nem utolsósorban szívügyem az „ékezetesítés”.

Programja — amennyiben a hivatkozási címek korrektek — logikus, bár némileg túlkomplikált. Márpedig még a gépi kódú programozásnál is „illik” odafigyelni a futási időre. (Nem mint ha ez esetben gyakorlati jelentősége lenne ennek, de hát jobb az ilyesmit idejekorán megszokni.)

A címeket azért említettem, mert a programban szereplő \$0324/\$0325-ös vektor, valamint a \$99-es cím a C64-nél az input műveletekkel kapcsolatos. (Persze ez lehet a két gép közötti eltérés is.)

De hibát okozhat az is, hogy programjában az X és Y regisztereket anélkül veszi igénybe, hogy előtte értéküket elmentené.

Az 1. lista a CHROUT rutint, a byte IEC buszra történő küldéséig mutatja. Látható, hogy nem „bolygatja” a regisztereket! (Amelyek később felhasználandó lényeges értékeket tartalmazhatnak.)

Mielőtt azonban továbbmennénk, érdemes felidézni néhány lényeges dolgot.

A CHROUT rutin, a \$9A címről kiolvasott aktuális egység számok alapján, különböző helyekre „irányítja” a kiírandó byte-ot.

1-es egység számnál a magnóra,

2-esnél az RS232-es illesztőre

3-asnál a képernyőre,

4-es és 15-ös egység szám között az IEC buszra. Ide csatlakoznak a nyomtatók

	REM 1. LISTA	
○	:	
	REM C64 CHROUT	
	:	
○	F1CA 48 PHA	
	F1CB A5 9A LDA 9A	
	F1CD C9 03 CMP #03	
○	F1CF D0 04 BNE F1D5	
	F1D1 68 PLA	
	F1D2 4C 16 E7 JMP E716	
○	F1D5 90 04 BCC F1DB	
	F1D7 68 PLA	
	F1D8 4C DD ED JMP EDDD	
○	F1DB 4A LSR A	

4—5-ös, valamint a lemezegységek 8—15-ös (gyakorlatilag 8—11-es) egység számmal.

Azt, hogy a csatlakoztatott berendezések közül melyik legyen aktív, az OPEN utasítással jelöljük ki.

A fentiekből következik, hogy a CHAROUT rutin előtt kell megvizsgálni azt, hogy a byte a 4-es számú egységre irányul-e?

A 2. lista „SAJAT RUTIN”-jét megnézve, látható hogy első két sora megegyezik az eredeti rutin első soraival.

3. sora ellenőrizni, hogy az aktuális egység szám 4-e? Ha nem, visszaadja a további feldolgozást az eredeti rutin 3. sorára.

4-es egység szám esetén pedig, az Y regiszter értékének elmentése után, a konvertáló táblázatból, Y-nal indexelve kiolvasott ASCII érték az akumulátorba kerül.

Visszatöltve Y értékét, az IEC buszra kerül a kiírandó byte.

Az ASCII táblázat alkalmazása egyúttal megoldja a BASIC/PC konverziót is.

A nyomtató korrekt működésének további feltétele az is, hogy a SET UP eljárás segítségével beállítsa az alábbi üzemmódokat: LF=LF+CR; CR=CR+LF

A nyomtató csak ezután hajt végre soremelést is a CHR\$(13) hatására.

	0 ; 2. LISTA	
○	10 ; IBM GP./SPANYOL	
	20 ;	
	30 MUTLO=#0326	
○	40 MUTHI=#0327	
	50 DEVNR=#9A	
	60 KEZDC=#C000	
○	70 IECBU=#EDDD	
	80 CHROU=#F1CD	
	90 ;	
○	100 *=#KEZDC	
	110 ;	
	120 TABL .BYTE #00,#01,#02,#03,#04,#05,#06,#07	
○	130 .BYTE #08,#09,#0A,#0B,#0C,#0D,#0E,#0F	
	140 .BYTE #10,#11,#12,#13,#14,#15,#16,#17	
	150 .BYTE #18,#19,#1A,#1B,#1C,#1D,#1E,#1F	
○	160 .BYTE #20,#21,#22,#23,#24,#25,#26,#27	
	170 .BYTE #28,#29,#2A,#2B,#2C,#2D,#2E,#2F	
	180 .BYTE #30,#31,#32,#33,#34,#35,#36,#37	
○	190 .BYTE #38,#39,#3A,#3B,#3C,#3D,#3E,#3F	
	200 .BYTE #40,#41,#42,#43,#44,#45,#46,#47	

```

210 .BYTE $68,$69,$6A,$6B,$6C,$6D,$6E,$6F
220 .BYTE $70,$71,$72,$73,$74,$75,$76,$77
230 .BYTE $78,$79,$7A,$7B,$7C,$7D,$7E,$7F
240 .BYTE $80,$81,$82,$83,$84,$85,$86,$87
250 .BYTE $88,$89,$8A,$8B,$8C,$8D,$8E,$8F
260 .BYTE $90,$91,$92,$93,$94,$95,$96,$97
270 .BYTE $98,$99,$9A,$9B,$9C,$9D,$9E,$9F
280 .BYTE $A0,$A1,$A2,$A3,$A4,$A5,$A6,$A7
290 .BYTE $A8,$A9,$AA,$AB,$AC,$AD,$AE,$AF
300 .BYTE $B0,$B1,$B2,$B3,$B4,$B5,$B6,$B7
310 .BYTE $B8,$B9,$BA,$BB,$BC,$BD,$BE,$BF
320 .BYTE $C0,$C1,$C2,$C3,$C4,$C5,$C6,$C7
330 .BYTE $C8,$C9,$CA,$CB,$CC,$CD,$CE,$CF
340 .BYTE $D0,$D1,$D2,$D3,$D4,$D5,$D6,$D7
350 .BYTE $D8,$D9,$DA,$DB,$DC,$DD,$DE,$DF
360 .BYTE $E0,$E1,$E2,$E3,$E4,$E5,$E6,$E7
370 .BYTE $E8,$E9,$EA,$EB,$EC,$ED,$EE,$EF
380 .BYTE $F0,$F1,$F2,$F3,$F4,$F5,$F6,$F7
390 .BYTE $F8,$F9,$FA,$FB,$FC,$FD,$FE,$FF
400 ;
410 ; SAJAT RUTIN
420 ;
430 PHA
440 LDA DEVNR
450 CMP #$04
460 BNE RET
470 STY PUFF
480 PLA
490 TAY
500 LDA TABL,Y
510 LDY PUFF
520 JMP IECBU
530 RET JMP CHROU
540 PUFF .BYTE 00
550 ;
560 ; INICIALIZALAS A VALOS CIM SZERINT
570 ;
580 LDA #$00
590 LDY #$C1
600 STA MUTLO
610 STY MUTHI
620 RTS
630 .END
    
```

READY.

Végül néhány megjegyzés, vagy ha úgy tetszik jótanács: Az ékezetes billentyűk kiosztását visszafejtve, az enyhén szólva kissé eklektikus. Ugyanis nem tesz eleget a következő „hármasszabálynak”:

Az ékezetes billentyűkiosztás legyen;

1. logikus, ergo
2. könnyen megjegyezhető, és végül
3. ne kelljen „akkordot fogni” egy-egy ékezetes karakter „megszólaltatásához”.

Jómagam az alábbi billentyűkiosztást dolgoztam ki:

```

A E T Y U I O P @
C = á é ű ü ú í ó ö ő
CTRL Á É Ú Ü Ű Í Ú Ő
    
```

(A CTRL kódtábla átírása sem ördögösség.)

A fentiek szerint, ha sikerül a címeket és a rutinokat PLUS/4-re „váltani”, problémája megoldható. Ám sokkal „elegánsabb” és teljesebb megoldás a „betölthető” karakterek alkalmazása.

Ezzel a módszerrel ugyanis a hosszú, kettős ékezetek is (ő;ö;ű;ü) előállíthatók.

Az EPSON FX 80 emulációt kell választani, és az ékezetes karaktereket az ASCII 161–185 közötti tartományba kell betölteni.

A gépi kódú megoldást a 3. lista tartalmazza. Sajnos csak a C64-re írott, de bőven kommentált változatot tudom prezentálni. A zárójelben lévő címek a KERNAL ugrótábla címei. Remélem ez is megkönnyíti az átírást.

Jó és sikeres munkát kívánva üdvözlö:

Szász Sándor

```

0 ; 3. LISTA
10 ;
20 ; EK/64/EPSON
30 ;
40 *=$C000
50 ;
60 ; KONVERZIOS KODTABLA
70 ;
80 TABL .BYTE $00,$01,$02,$03,$04,$05,$06,$07
90 .BYTE $08,$09,$0A,$0B,$0C,$0D,$0E,$0F
100 .BYTE $10,$11,$12,$13,$14,$15,$16,$17
110 .BYTE $18,$19,$1A,$1B,$1C,$1D,$1E,$1F
120 .BYTE $20,$21,$22,$23,$24,$25,$26,$27
130 .BYTE $28,$29,$2A,$2B,$2C,$2D,$2E,$2F
140 .BYTE $30,$31,$32,$33,$34,$35,$36,$37
150 .BYTE $38,$39,$3A,$3B,$3C,$3D,$3E,$3F
160 .BYTE $40,$41,$42,$43,$44,$45,$46,$47
170 .BYTE $48,$49,$4A,$4B,$4C,$4D,$4E,$4F
180 .BYTE $50,$51,$52,$53,$54,$55,$56,$57
190 .BYTE $58,$59,$5A,$5B,$5C,$5D,$5E,$5F
200 .BYTE $60,$61,$62,$63,$64,$65,$66,$67
210 .BYTE $68,$69,$6A,$6B,$6C,$6D,$6E,$6F
220 .BYTE $70,$71,$72,$73,$74,$75,$76,$77
230 .BYTE $78,$79,$7A,$7B,$7C,$7D,$7E,$7F
240 .BYTE $80,$81,$82,$83,$84,$85,$86,$87
250 .BYTE $88,$89,$8A,$8B,$8C,$8D,$8E,$8F
260 .BYTE $90,$91,$92,$93,$94,$95,$96,$97
270 .BYTE $98,$99,$9A,$9B,$9C,$9D,$9E,$9F
280 .BYTE $A0,$A1,$A2,$A3,$A4,$A5,$A6,$A7
290 .BYTE $A8,$A9,$AA,$AB,$AC,$AD,$AE,$AF
300 .BYTE $B0,$B1,$B2,$B3,$B4,$B5,$B6,$B7
310 .BYTE $B8,$B9,$BA,$BB,$BC,$BD,$BE,$BF
320 .BYTE $C0,$C1,$C2,$C3,$C4,$C5,$C6,$C7
330 .BYTE $C8,$C9,$CA,$CB,$CC,$CD,$CE,$CF
340 .BYTE $D0,$D1,$D2,$D3,$D4,$D5,$D6,$D7
350 .BYTE $D8,$D9,$DA,$DB,$DC,$DD,$DE,$DF
360 .BYTE $E0,$E1,$E2,$E3,$E4,$E5,$E6,$E7
370 .BYTE $E8,$E9,$EA,$EB,$EC,$ED,$EE,$EF
380 .BYTE $F0,$F1,$F2,$F3,$F4,$F5,$F6,$F7
390 .BYTE $F8,$F9,$FA,$FB,$FC,$FD,$FE,$FF
400 ;
410 ; SAJAT RUTIN
420 ;
430 PHA
440 LDA $9A
450 CMP #$04
460 BNE RET
470 STY PUFF
480 PLA
490 TAY
500 LDA TABL,Y
510 LDY PUFF
520 JMP $EDDD ; IEC BUSZ
530 RET JMP $F1CD ; CHROUT RUTIN 3. SORA
540 PUFF .BYTE 00
550 ;
560 ; CTRL KODTABLA
570 ;
580 KOD .BYTE $FF,$FF,$FF,$FF,$FF,$FF,$FF,$FF
590 .BYTE $1C,$17,$AD,$9F,$1A,$13,$AC,$FF
600 .BYTE $9C,$12,$04,$1E,$03,$06,$A5,$18
610 .BYTE $1F,$B4,$07,$9E,$02,$08,$B5,$16
620 .BYTE $12,$A1,$0A,$92,$0D,$0B,$B6,$0E
630 .BYTE $FF,$AA,$0C,$FF,$FF,$1B,$A7,$FF
640 .BYTE $1C,$FF,$1D,$FF,$FF,$1F,$1E,$FF
650 .BYTE $90,$06,$FF,$05,$FF,$FF,$11,$FF
660 .BYTE $FF ; FONTOS BYTE !!
670 ;
680 ; CTRL RUTIN
690 ;
700 LDA $02BD ; "CTRL LENYOMVA ?"
710 CMP #$04
    
```



```
○ 720 BEQ RUT
730 JMP $EB4B ; NINCS
740 RUT LDY #CB ; IGEN
○ 750 LDA KOD,Y
760 JMP $EAE4
770 ;
○ 780 ; KAR.GEN. MASOLAS
790 ;
800 SEI ; SYS 49516
○ 810 LDA #33
820 STA #01
830 LDX #00
○ 840 LDY #08
850 FOR LDA $D800,X ; KARAKTER GENERATOR
860 CEL STA $EB00,X ; RAM
○ 870 INX
880 BNE FOR
890 INC FOR+2
○ 900 INC CEL+2
910 DEY
920 BNE FOR
○ 930 LDA #37
940 STA #01
950 CLI
○ 960 ;
970 ; EKEZETES KARAKTEREK TOLTESE
980 ;
○ 990 LDX #00
1000 MOD LDA EKE,X
1010 STA $EB0B,X
○ 1020 LDA #FF
1030 SEC
○ 1040 SBC EKE,X
1050 STA $EF0B,X ; INVERZ KARAKTEREK
1060 INX
○ 1070 CPX #CB
1080 BNE MOD
1090 LDA #C4 ; MUTATOK ATALLITASA
1100 LDX #1B
○ 1110 STA $0288
1120 STA $DD00 ; CIA REGISZTEREK
1130 STX $D018
○ 1140 LDA #5A
1150 LDX #C1
○ 1160 STA $028F
1170 STX $0290
1180 JMP $E544 ; KEPERNYO TORLES
1190 ;
○ 1200 ; EKEZETES KARAKTER ADATOK
1210 ;
○ 1220 EKE .BYTE $79,$33,$32,$30,$30,$30,$78,$00
1230 .BYTE $0C,$18,$00,$38,$18,$18,$3C,$00
1240 .BYTE $33,$66,$00,$66,$66,$66,$3E,$00
○ 1250 .BYTE $33,$66,$00,$3C,$66,$66,$3C,$00
1260 .BYTE $63,$C6,$00,$C3,$C3,$C3,$7E,$00
1270 .BYTE $CC,$CC,$33,$33,$CC,$CC,$33,$33
○ 1280 .BYTE $33,$66,$00,$7E,$C3,$C3,$7E,$00
1290 .BYTE $00,$00,$00,$00,$00,$18,$18,$30
1300 .BYTE $00,$00,$18,$00,$00,$18,$00,$00
○ 1310 .BYTE $66,$66,$00,$7E,$C3,$C3,$7E,$00
1320 .BYTE $18,$18,$18,$1F,$1F,$15,$18,$18
1330 .BYTE $06,$7E,$6C,$60,$78,$60,$7E,$00
○ 1340 .BYTE $1B,$3F,$66,$7E,$66,$66,$66,$00
1350 .BYTE $00,$00,$00,$F8,$F8,$18,$18,$18
1360 .BYTE $66,$66,$00,$3C,$66,$66,$3C,$00
○ 1370 .BYTE $0C,$18,$3C,$06,$3E,$66,$3E,$00
1380 .BYTE $0C,$18,$3C,$66,$7E,$60,$3C,$00
1390 .BYTE $00,$00,$00,$FF,$FF,$18,$18,$18
○ 1400 .BYTE $18,$18,$18,$F8,$F8,$18,$18,$18
1410 .BYTE $66,$66,$00,$C3,$C3,$C3,$7E,$00
1420 .BYTE $18,$DB,$DB,$C3,$C3,$C3,$7E,$00
○ 1430 .BYTE $18,$7E,$DB,$C3,$C3,$C3,$7E,$00
1440 .BYTE $66,$66,$00,$66,$66,$66,$3E,$00
1450 .BYTE $0C,$18,$00,$66,$66,$66,$3E,$00
○ 1460 .BYTE $0C,$18,$00,$3C,$66,$66,$3C,$00
1470 ;
○ 1480 ; PRINTER ELOKESZITES: SYS 49796
1490 ;
1500 LDA #01
1510 LDX #04
○ 1520 LDY #00
1530 JSR $FE00 ; ($FFBA)
1540 LDA #00
```

```
○ 1550 JSR $FE18 ; ($FF90)
1560 JSR $F34A ; ($FFC0)
1570 LDX #01
1580 JSR $F250 ; ($FFC9)
○ 1590 ;
1600 ; EKEZETES KARAKTEREK TOLTESE PRINTERBE
1610 ;
○ 1620 LDA #C1
1630 LDX #C2
○ 1640 STA #F7
1650 STX #F8
1660 F1 LDY #00
○ 1670 LDA ($F7),Y
1680 CMP #FF
1690 BEQ F2
○ 1700 JSR $EDDD ; IEC BUSZ
1710 INC #F7
1720 BNE F1
○ 1730 INC #F8
1740 JMP F1
1750 F2 LDA #00
○ 1760 LDX #C1
1770 STA $0326
1780 STX $0327
○ 1790 RTS
1800 ;
1810 ; EPSON KODOK
○ 1820 ;
1830 .BYTE $18,$1B,$3A,$00,$00
1840 .BYTE $00,$1B,$26,$00,$A1,$B9
○ 1850 .BYTE $80,$00,$00,$82,$00,$FE
1860 .BYTE $00,$82,$20,$40,$80,$00
1870 .BYTE $80,$00,$22,$00,$3E,$40
○ 1880 .BYTE $82,$00,$00,$00,$00,$00
1890 .BYTE $80,$3C,$02,$40,$82,$00
1900 .BYTE $42,$80,$3C,$02,$00,$00
○ 1910 .BYTE $80,$1C,$22,$00,$62,$80
1920 .BYTE $22,$40,$A2,$1C,$00,$00
1930 .BYTE $80,$7C,$02,$00,$C2,$00
○ 1940 .BYTE $C2,$00,$02,$7C,$00,$00
1950 .BYTE $80,$00,$1C,$22,$00,$22
1960 .BYTE $14,$08,$14,$22,$00,$00
○ 1970 .BYTE $80,$3C,$42,$00,$E2,$00
1980 .BYTE $E2,$00,$42,$3C,$00,$00
1990 .BYTE $00,$00,$00,$0C,$01,$0E
○ 2000 .BYTE $00,$00,$00,$00,$00,$00
2010 .BYTE $80,$00,$00,$36,$00,$36
2020 .BYTE $00,$00,$00,$00,$00,$00
○ 2030 .BYTE $80,$3C,$42,$80,$42,$00
2040 .BYTE $42,$80,$42,$3C,$00,$00
2050 .BYTE $80,$00,$00,$E0,$00,$00
○ 2060 .BYTE $00,$E0,$00,$00,$00,$00
2070 .BYTE $80,$3E,$00,$2A,$00,$2A
2080 .BYTE $40,$AA,$00,$22,$00,$00
○ 2090 .BYTE $80,$0E,$10,$28,$40,$88
2100 .BYTE $40,$28,$50,$8E,$00,$00
2110 .BYTE $00,$00,$00,$00,$6C,$01
○ 2120 .BYTE $6E,$00,$00,$00,$00,$00
2130 .BYTE $80,$1C,$22,$80,$22,$00
2140 .BYTE $22,$80,$22,$1C,$00,$00
○ 2150 .BYTE $80,$04,$0A,$20,$0A,$60
2160 .BYTE $8A,$20,$1C,$02,$00,$00
2170 .BYTE $80,$1C,$22,$08,$22,$48
○ 2180 .BYTE $A2,$08,$22,$18,$00,$00
2190 .BYTE $80,$00,$00,$00,$00,$00
2200 .BYTE $00,$00,$00,$00,$00,$00
○ 2210 .BYTE $80,$00,$00,$00,$00,$00
2220 .BYTE $00,$00,$00,$00,$00,$00
2230 .BYTE $80,$7C,$02,$00,$82,$00
○ 2240 .BYTE $82,$00,$02,$7C,$00,$00
2250 .BYTE $80,$FC,$02,$00,$02,$C0
2260 .BYTE $02,$00,$02,$FC,$00,$00
○ 2270 .BYTE $80,$3C,$42,$00,$42,$A0
2280 .BYTE $42,$00,$42,$3C,$00,$00
2290 .BYTE $80,$3C,$02,$80,$02,$00
○ 2300 .BYTE $02,$80,$3C,$02,$00,$00
2310 .BYTE $80,$3C,$02,$00,$42,$80
2320 .BYTE $02,$00,$3C,$02,$00,$00
○ 2330 .BYTE $80,$1C,$22,$00,$22,$40
2340 .BYTE $A2,$00,$22,$1C,$00,$00
2350 .BYTE $1B,$25,$01,$00,$11,$FF
○ 2360 .END ; #50174
READY.
```



Tisztelt Szerkesztőség!

Egy újabb játékprogramot szeretnék felajánlani a COM-MODORE Újságban való közlés céljára.

A játék neve: CIMBO-64

A játék a TV-ből is ismert játék C-64-es változata.

A játék rövid ismertetése:

A tábla 45 mezőből áll. Amelyik mezőn van korong, az sárga, amelyiken nincsen, az szürke színű. Egy korongot úgy vehetünk le a tábláról, hogy egy mellette lévő koronggal átlépünk felette, egy olyan helyre, ahol nincs korong. Átlépni vízszintesen, függőlegesen, vagy átlósan lehet.

A játékidő 4 perc. A cél, hogy minél kevesebb korong maradjon a táblán.

Tisztelettel:
Haraszi Árpád

```

1 REM *****
2 REM * C= UJSAG 1993 *
3 REM *
4 REM * CIMBO *
5 REM *
6 REM * HARASZTI ARPAD *
7 REM *****
10 :
45 FOR I=1 TO 9: N$(I)="HARASZTI SOFT": M(I)=45: NEXT
50 POKE 53280,0: POKE 53281,0
55 DIM A$(45), B(45), C(99)
58 GOSUB 175: GOSUB 185
60 PRINT "***** - START"
65 PRINT "***** - SZABALYOK"
70 PRINT "***** - TOP LIST"
75 PRINT "***** - KILEPES"
80 GET A$: IF A$="" THEN 88
81 IF A$=" " THEN 800
82 IF A$=" " THEN 900
83 IF A$=" " THEN END
100 X1$=" " : X2$="*****" : X3$=X1$+" " : X4$=X3$+"*****" : PC=44
102 X5$=" " : X6$="*****"
105 PRINT "*****"
108 PRINT "*****"
110 PRINT "*****"
115 PRINT X1$ "*****"
120 PRINT X1$ "*****"
125 PRINT X1$ "*****"
130 PRINT X1$ "*****"
135 PRINT X4$ : PRINT X4$ : PRINT X3$ : X5$
140 PRINT "*****"
145 FOR I=1 TO 6 : PRINT "*****" : NEXT I
150 PRINT "*****"
155 PRINT X3$ "*****"
160 FOR I=1 TO 5 : PRINT X4$ : NEXT
165 PRINT X1$ "*****"
170 PRINT "***** HARASZTI" : PRINT " SOFTWARE" : PRINT " 1993"
172 GOSUB 193 : GOTO 195
175 X$="*****"
180 FOR I=1 TO 45 : A$(I)=X$+RIGHT$(STR$(I),2)
183 NEXT
184 RETURN
185 FOR I=1 TO 45 : READ A, B : A(I)=A : B(I)=B : NEXT
186 DATA 12,4,15,4,18,4,12,6,15,6,18,6,12,8,15,8,18,8,3,10,6,18,9,10,12,18
187 DATA 15,18,18,18,21,18,24,18,27,10,3,12,6,12,9,12,12,15,12,18,12,21,12
188 DATA 24,12,27,12,3,14,6,14,9,14,12,14,15,14,18,14,21,14,24,14,27,14
189 DATA 12,16,15,16,18,16,12,18,15,18,18,18,12,20,15,20,18,20
192 RETURN
193 FOR I=1 TO 45 : C(I)=1 : NEXT : RETURN
195 FOR I=1 TO 45 : POKE 211, A(I) : POKE 214, B(I) : SYS 58732 : PRINT "*****" : NEXT
200 PRINT "*****"
210 POKE 211, 24 : POKE 214, 19 : SYS 58732 : PRINT "*****"
215 GET A$: IF A$="" THEN 215
218 A=INT(44*RND(1))+1
220 PRINT "*****"
221 PRINT TAB(24) "*****"
222 PRINT "*****" : POKE 211, A(A) : POKE 214, B(A) : SYS 58732
223 TI$="000000"
230 GET A$ : GOTO 960
231 IF C2=2 THEN POKE 211, 35 : POKE 214, 28 : SYS 58732 : PRINT "*****"
232 IF A$=CHR$(13) AND SZ>0 AND SZ<46 AND C(SZ)=1 AND C2=0 THEN C2=2 : S1=SZ : PX=0 : SZ=0
233 IF A$=CHR$(13) AND SZ>0 AND SZ<46 AND C2=2 THEN S2=SZ : PX=0 : SZ=0 : C2=0 : GOTO 580
234 IF A$=CHR$(13) THEN 237
235 IF A$<>" " THEN IF PX<0 THEN PX=1 : SZ=VAL(A$) : GOTO 237
236 IF A$<>" " THEN SZ=10*SZ+VAL(A$) : PX=0
237 IF SZ>0 AND SZ<46 THEN POKE 211, 34 : POKE 214, 13+C2 : SYS 58732 : PRINT "*****"
240 IF C2=0 THEN POKE 211, 35 : POKE 214, 19 : SYS 58732 : PRINT "*****"
400 PZ=36-(VAL(MID$(TI$,4,1))*9)+(VAL(MID$(TI$,5,2))/6) : IF PZ<0 THEN 5800
401 POKE 211, PZ : POKE 214, 24 : SYS 58732 : PRINT "*****"
402 IF PC=0 THEN 5800
405 GOTO 230
500 Z$=TI$ : POKE 211, 34 : POKE 214, 21 : SYS 58732 : PRINT "*****"
505 IF C(S2)=1 THEN GOTO 230
510 IF C(S1)=0 THEN GOTO 230

```




```

515 RESTORE:FORI=1TO90:READA:NEXT
518 FOR I=1 TO 93:READA,B,C
520 IF S1=A AND S2=B AND C(C)=1 THEN C(C)=0:C(S1)=0:C(S2)=1:S3=0:GOTO 600
521 IF S2=A AND S1=B AND C(C)=1 THEN C(C)=0:C(S1)=0:C(S2)=1:S3=C:GOTO 600
525 NEXT:GOTO 230
600 POKE211,A(S1):POKE214,B(S1):SYS58732:PRINT"R#(S1)
602 POKE211,S5:POKE214,Z1:SYS58732:PRINT"
605 POKE211,A(S2):POKE214,B(S2):SYS58732:PRINT"R#(S2)
608 POKE211,A(S3):POKE214,B(S3):SYS58732:PRINT"R#(S3):PC=PC-1
610 TI#=Z#:GOTO 230
700 DATA 1,3,2,1,9,5,1,7,4,2,8,5,3,9,6,3,7,5,4,13,7,4,15,8,4,6,5,5,14,8
705 DATA 5,12,7,5,16,9,6,13,8,6,15,9,7,9,8,7,20,12,7,22,13,7,24,14,8,21,13
710 DATA8,23,14,8,25,15,9,22,14,9,24,15,9,26,16,10,12,11,10,28,19,10,30,20
715 DATA 11,13,12,11,29,20,11,31,21,12,28,20,12,30,21,12,32,22,13,29,21
720 DATA13,31,22,13,33,23,14,30,22,14,32,23,14,34,24,15,31,23,15,33,24,15,35,25
725 DATA16,32,24,16,34,25,16,36,26,17,33,25,17,35,26,18,34,26,18,36,27
730 DATA12,14,13,13,15,14,14,16,15,15,17,16,16,18,17,19,21,20,22,21,21,23,22
735 DATA22,24,23,23,25,24,24,26,25,25,27,26,28,30,29,29,31,30,30,32,31,31,33,32
740 DATA32,34,33,33,35,34,34,36,35,37,39,38,40,42,41,43,45,44,22,37,31
745 DATA33,38,32,24,39,33,31,40,37,32,41,38,33,42,39,37,43,40,38,44,41,39,45,42
750 DATA21,38,31,22,39,32,24,37,32,25,38,33,31,42,38,33,40,38,37,45,41,39,43,41
800 DATA 34,41,39,30,41,37,39,26,34,20,37,30,21,38,31
800 PRINT"
810 PRINT"KORONG A TABLA 45 MEZOBOL ALL. AMELYIK MEZON"
815 PRINT"AVAN KORONG, AZ SARGA, AMELYIKEN NINC,S"
820 PRINT"MAZ SZURKE, EGY KORONGOT UGY VEHETUNK LE"
825 PRINT"AR TABLAROL, HUGY EGY KORONGGAL ATLEPUNK"
830 PRINT"FOLOTTE EGY OLYAN HELYTE, AMOL NINC,S"
835 PRINT"KORONG, ATLEPNI VIZSZINTESEN, FUGGU-"
840 PRINT"MEGESEN, VAGY ATLOSAN LEHET."
845 PRINT"MINEL DEL : MINEL KEVESEBB KORONG MARADJON"
850 PRINT"AR TABLAN. JATEKIDU : % PERC."
860 PRINT"
PRESS ANY KEY !";
865 GETA#:IFA#=""THEN865
870 GOTO 60
900 PRINT"
902 PRINT"
905 PRINT"
910 FORI=1TO 9:PRINT"
920 PRINT"
925 PRINT"
:FORI=1TO9:PRINT"
:NEXT
930 PRINT"
:FORI=1TO9:PRINT"
:NEXT
935 PRINT"
- LOAD TOP
- SAVE TOP
- MENU"
940 GETA#:IFA#=""THEN940
942 IF A#=""THEN 1000
944 IF A#=""THEN 1100
950 GOTO 60
960 IF A#=""THEN 60
962 IF A#=""S"THEN PX=0:SZ=0:GOTO 230
965 IFA#=""K"THEN 5000
970 GOTO 231
1000 OPEN15,8,15:OPEN2,8,2,"0:CIMBO-TOP,S,R":GOSUB 1200
1010 IF VC=2THENVC=0:GOTO1150
1020 FORI=1TO9:INPUT#2,N#(I)
1030 INPUT#2,M(I)
1040 NEXTI:GOTO 1150
1100 OPEN15,8,15:OPEN2,8,2,"00:CIMBO-TOP,S,W":GOSUB1200
1110 IFVC=2THENVC=0:GOTO1150
1120 FORI=1TO9:PRINT#2,N#(I)
1130 PRINT#2,M(I)
1140 NEXTI
1150 CLOSE2:CLOSE2,8,2:CLOSE15:GOTO 900
1200 PRINT"
:A=0:B=0:C=0:B#=""":INPUT#15,A,B#,C,D:IF A>0 AND AC=19 THEN RETURN
1210 PRINT"
DISK-HIBA !"
1220 GETA#:IFA#=""THEN 1220
1230 VC=2:CLOSE2:CLOSE2,8,2:CLOSE15:RETURN
5000 PRINT"
GAME OVER !"
5010 PRINT"
:IF P<0 THEN PRINT"
LEJART AZ IDU !"
5020 PRINT"
"PC" OB KORONG MARADT."
5025 FORI=1 TO 10:GETA#:NEXTI
5030 IF PC>M(9) THEN 5100
5040 PRINT"
FELKERULT A TOP LISTARA !"
5050 INPUT"
MI A NEVE ";N#
5055 IF LEN(N#)>14 THEN PRINT"TTTT":GOTO5040
5060 M(9)=PC:DF=1:N#(9)=N#
5065 FORI=0TO45:FORA=1TO9
5070 IFM(A)=1THEN0(DF)=M(A):0#(DF)=N#(A):DF=DF+1:IF DF=10 THEN 5080
5075 NEXTA:NEXTI
5080 FORI=1TO9:N#(I)=0#(I):M(I)=0(I):NEXTI
5100 PRINT"
PRESS ANY KEY !"
5110 GETA#:IFA#=""THEN 5110
5120 GOTO 60

```

READY.





Alkalmazói
program
pályázat

Az első

TRIANGRAM 3D

diagramszerkesztő leírása és kezelése

Oktatásban, vállalkozásban és még az élet számos más területén hasznosan alkalmazhatjuk a diagramokat, amelyek segítségével adat sorokat, folyamatokat tehetünk szemléletessé.

A TRIANGRAM program ezt teszi lehetővé és mellékesen még némi adattárolásra is felhasználható.

Úszerű a programban az, hogy egy képernyőn, egy koordináta rendszerben három diagramot szerkeszt

meg három színben és háromdimenziós kirajzolásban.

Egy-egy diagram max. 24 oszlopból állhat. Magasságuk értéke 1—99999 között tehát max. ötjegyű számokkal adható meg. Megadható egy 20 karakteres szöveg, amely lehet a cím, illetve egy rövid információ, továbbá mindegyik oszlopsorhoz egy 7 karakteres megjegyzés is.

Az adatok és a szövegek lemezen tárolhatók és szükség szerint ismételten behívhatók és kirajzolhatók.

A program felhasználó-barát. Használatát a képernyőn megjelenő MENÜ, tájékoztatások, hangjelzések segítik.

Ellenőrzi a lemez, ill. a meghajtó állapotát, az eredményt vagy a hibát kiírja a képernyőre és vár a kijávitásra.

A program használatához szükséges eszközök:

1. COMMODORE C-64 tip. számítógép
2. Lemez meghajtó (floppi drive)
3. Monitor, ill. TV-készülék (lehetőleg színes)

A programot tartalmazó lemezen található meg négy DEMO adatállomány, amely betöltve négyféle felhasználási módot mutat be. Természetesen ezeken kívül még számos lehetőség is adódik.

```

1 rem *****
2 rem *
3 rem *      t r i a g r a m      3d      *
4 rem *
5 rem * c-64      harna m. andras *
6 rem *****
7 poke 53281,0: poke 53280,0: poke 808,227: poke 657,128: goto 80
8 dim a(no,ns): return
9 poke 211,0: poke 214,s: sys 58732: return
10 z=0: sz$="": rem      get rutin
20 if z=nc+1 then 70
23 print chr$(18)chr$(62)chr$(146);
25 get g$: if g$="" or g$=chr$(34) then 25
27 if g$="[Dwn]" or g$="[Up]" or g$="[Rgt]" or g$="[Lft]" then gosub 560: goto 25
29 if z>0 and g$=chr$(20) then 60
35 if asc(g$)>ak and asc(g$)<av then 50
38 if g$<>chr$(13) then 25
41 goto 70
50 z=z+1: sz$=sz$+g$
55 print chr$(157)g$,: goto 20
60 z=z-1: sz$=left$(sz$,z): goto 55
70 print chr$(157)" ": return
80 tg$="tova[Up][Lft].[Dwn]bb: egy gomb ![Up]"
85 si=54272
90 kr$="[Clr][Yel][10: ]k r a m p u s z o f t[8: ] [2:Dwn][Wht]"
99 rem      - - - f o p r o g r a m - - -
100 gosub 1900: rem      cimkep
110 gosub 1700: rem      uj/tarolt ?
120 if ta=1 then 300
130 gosub 1400: rem      adat beadas
140 gosub 1200: rem      elokeszites
150 gosub 1000: rem      szovegek
160 gosub 800: rem      kirajzolas
200 print kr$
205 print tab(9)"[3:Dwn]U[2:s*]I[4: ]."
210 print tab(9)"Bf1B vege. uj triagram"
215 print tab(9)"J[2:s*]K"
220 print tab(9)"U[2:s*]I"
225 print tab(9)"Bf3B adat lista"
230 print tab(9)"J[2:s*]K"
235 print tab(9)"U[2:s*]I[9: ]."
240 print tab(9)"Bf5B kimentes lemezre"

```

TRIAGRAM 3D diagramszerkesztő program felépítése

A programot készítette: Harna M. András, 1993. augusztus.

Konfiguráció: 1. C-64 tip. számítógép
2. 1541 tip. floppy drive
3. Monitor vagy TV-készülék

Programnyelv: BASIC

Hossza: 8100 byte, lemezen 32 blokk 290 prg. sor.

Adatok tárolása: Lemezen SEQ-FILE-ban

Kezelés: Billentyűzetről

A program egy FŐPROGRAM-ból, az általa hívott szubrutinokból és egy MENÜ-ből hívható programrészekből áll.
A prg. felépítése sorszám szerint:

1—6 Lista cím.
7 Színbeállítás. RUN/stop és C=/SHIFT gombok letiltása. Innen a 80. sorra lép.
8 DIM szubrutin
9 Pozíció szubrutin
10—60 GET szubrutin
80—90 Állandók. Továbblépés kiírás. SID alapért. Képernyő fejléc.

FŐPROGRAM

100 Gosub 1900 Hívja a címképet. Innen hívható az egy képernyőnyi INFO.
110 Gosub 1700 Választás, hogy új vagy tárolt adatokkal dolgozzon.
120 Ha tárolt, akkor a 300. sorra lép.
130 Gosub 1400 Új adatok bekérése. Hívja a GET-rutint. Hibás adatok kizárása.
140 Gosub 1200 Előkészíti a koordináta rendszert és a 3D-s oszlopraiz karaktereit.
150 Gosub 1000 A képernyőre írandó szövegek. Kiírások függőlegesen.
160 Gosub 800 A beadott vagy behívott adatok alapján szerkeszti és kirajzolja a TRIAGRAM-ot.
Egy gomb leütésére a MENÜ-re lép.

MENÜ

200—260 A MENÜ képernyője.
265 GET. Választás a funkció-gombokkal.

A MENÜ SZERKEZETE:

F1 BEFEJEZÉS 280	F3 ADATLISTA 700	F5 KIMENTÉS 400	F7 leütésére: KIRAJZOLÁS 140-es sorra lép.
280 Befejezi a programot. Oldja a letiltásokat. Előkészíti az újra futtatást.			
300 ADATBETÖLTÉS	Innen meghívható a DIRECTORY (600. sor) Bekéri a FILE nevet. Hívja DISK STATUS vizsgálatát (500. sor). Kiírja az erdményt. Hiba esetén kiírja azt, majd várakozik a javításra. Betölti az adatokat és visszalép a MENÜ-re.		
400 ADATKIMENTÉS.	Mint az előzőnél.		
500 DISK STATUS vizsgálat.			
550 Hangjelzés	1. GONG 2. GRRRRR. Hibás beütés esetén.		
600 DIR kiíró rutin.			
700 ADATLISTA kiírás.	Függőlegesen kiírja az összes adatokat és hozzájuk tartozó szövegeket. A helyérték helyes kiírást a LOG függv. alkalmazásával pozicionálja.		

Néhány fontosabb változó:

Pozicionáló rutin
NO = oszlopok száma
NS = sorok száma
BN\$ = Címszöveg
MJ\$ (1) MJ\$ (2) MJ\$ (3) az oszlopokhoz tartozó megjegyzések
MA = a legnagyobb oszlopérték
A/NO, NS/ tömbben tárolja az adatokat

O = oszlop S = sor
a TRIAGRAM-ban

A program természetesen továbbfejleszhető. A 300 soros limit miatt néhány rutin nem került be a programba.
Néhány tipp a fejlesztésre:

- AUTOSTART beépítése (az adatvédelem is megoldható).
- Adattárolás szalagon is.
- Az oszlopsorok színeinek megválaszthatósága.
- Az adatok utólagos módosíthatósága.



```
○ 245 print tab(9)"J[2:s*]K"  
250 print tab(9)"U[2:s*]I"spc(18)".  
○ 255 print tab(9)"Bf7B triagram rajzolas"  
○ 260 print tab(9)"J[2:s*]K": gosub 550  
265 get g$: if g$="" then 265  
○ 270 if asc(g$)<133 or asc(g$)>136 then 265  
275 on asc(g$)-132 goto 280,700,400,140  
○ 280 print "[Clr][6:Dwn][4: ]ve[Up][Lft].[Dwn]ge a programnak ! [2:Dwn]":  
poke 808,237: poke 650,0  
284 print tab(15)"[10:=P]"  
○ 285 print " kiva[Up][Lft].[Dwn]n me[Up][Lft].[Dwn]g egy [RvOn]  
triagram [RvOf]-ot ? i / n[3:Dwn]"  
286 get g$: if g$="i" then run  
○ 287 if g$<>"n" then 286  
288 print "[3:Dwn][5: ]v i s z o n t l a[Up][Lft].[Dwn] t a[Up][Lft].  
[Dwn] s r a ! [3:Dwn]": end  
○ 290 rem -----  
300 print kr$  
○ 304 print " [RvOn][11: ][=@] . [9: ][=@] "  
305 print " [RvOn] adatok betoltese lemezrol:"  
306 print " [27:=Y]"  
○ 310 open 15,8,15,"i"  
315 open 2,8,2,fi$+",s,r": gosub 500: if h=1 then close 2: close 15: goto 110  
○ 320 input#2,bn$: input#2,ma  
325 input#2,no: input#2,ns  
330 for j=1 to ns: input#2,mj$(j): next j  
○ 335 gosub 8  
340 if ns=1 then 360  
○ 345 if ns=2 then 355  
350 for i=1 to no: input#2,a(i,3): next  
355 for i=1 to no: input#2,a(i,2): next  
○ 360 for i=1 to no: input#2,a(i,1): next  
365 close 2: close 15  
○ 369 print "[2:Dwn] [RvOn][4: ][=@] . [4: ]"  
370 print " [RvOn] betoltes kesz !"  
371 print " [16:=U][3:Dwn]"  
○ 375 print " "tg$: gosub 550: poke 198,0: wait 198,1: goto 200  
395 :  
○ 400 gosub 600: print kr$  
404 print " [RvOn][14: ].[11: ]"  
405 print " [RvOn] adatok kimentese lemezre:"  
○ 409 print "[2:Dwn] .[7: ]."  
410 print " kerem a tarolando file nevet ! [Dwn][7: ][s\]"  
○ 412 input " (max. 16 karakter) ";fi$  
414 if len(fi$)>16 or len(fi$)<1 then 400  
418 open 15,8,15,"i"  
○ 420 open 2,8,2,"@: "+fi$+",s,w": gosub 500: if h=1 then close 2: close 15: goto 200  
430 print#2,bn$: print#2,ma  
432 print#2,no: print#2,ns  
○ 435 for j=1 to ns: print#2,mj$(j): next j  
442 if ns=1 then 457  
○ 444 if ns=2 then 453  
450 for i=1 to no: print#2,a(i,3): next i  
453 for i=1 to no: print#2,a(i,2): next i  
○ 457 for i=1 to no: print#2,a(i,1): next i  
460 close 2  
○ 464 print "[Dwn] [RvOn][7: ]. [4: ]"  
465 print " [RvOn] kimentes kesz !"  
466 print " [16:=U][Dwn]": gosub 550  
○ 470 print " "tg$: poke 198,0: wait 198,1  
480 goto 200  
○ 500 input#15,en,em$,et,es  
510 if en=0 then print "[2:Dwn] [RvOn] disk ok! [RvOf][2:Dwn]": h=0: gosub 550: return  
515 if en=62 then print "[2:Dwn] [RvOn] nincs ilyen file ! [RvOf][2:Dwn]": goto 520  
○ 516 if en=65 then print "[2:Dwn][RvOn] nincs mar hely a lemezen ! [RvOf]": goto 520  
517 print "[2:Dwn][RvOn] hiba a meghajtoban ! [RvOf][2:Dwn]"  
○ 520 gosub 560: print tg$: poke 198,0: wait 198,1  
525 close 2: close 15: h=1: return  
530 :  
○ 550 for i=1 to 24: poke si+i,0: next  
552 poke si+1,35: poke si+5,12: poke si+15,6: poke si+24,15: poke si+4,21  
554 for t=1 to 1500: next : poke si+4,20
```

A PROGRAM KEZELÉSE:

Betöltés után RUN-nal indítható.

A program címképpel jelentkezik be. Innen kérhető egy rövid információ (I vagy N gomb).

Nemleges válasz vagy az információ után a program megkérdezi, hogy új vagy a lemezen tárolt adatokkal kíván-e dolgozni (U vagy T).

ÚJ ADATOK BEADÁSA:

1. SZÖVEG bekérés. Beírható egy max. 20 karakterből álló szöveg. A beírás után RETURN-al kell véglegesíteni. A szöveg lehetséges hosszát egy ■ jel mutatja. Ennél hosszabb szöveget nem fogad el, ill. újra kéri. Téves beütést az INST/DEL gomb leütésével visszalépve lehet javítani.

2. OSZLOPSZÁM bekérés. A prg. bekér egy számot 1—24 között. Ez határozza meg, hogy egy sorban (diagramban) hány oszlop fog megjelenni. A kiíráson kívüli más számot vagy jelet nem fogad el. Ilyenkor újra kérdez. A kurzor a téves beadáson áll. Ezt felülírva lehet a már helyes adatot beadni.

3. SOROK SZÁMA. Bekér egy számot, amely 1—2—3 lehet. Ennek megfelelő számú sor (diagram) fog kirajzolódni. Egynél több sor esetén az

1-es sor lesz a hátsó, míg a nagyobb számú sor lesz a hozzánk közelebb eső.

Ezek után sorra bekéri az oszlopok magassági adatait. Beadható egy max. ötjegyű szám. Javítás az INST/DEL gombbal. Az ötödik számjegy után továbblép. Kevesebb számjegy esetén a RETURN gombbal lehet továbblépni.

Ha egy sor összes adatai (a 2. pont szerinti számban) megvannak, bekér egy max. 7 karakterből álló szöveget, amely a sorhoz tartozó megjegyzés lesz. Beírás mint az 1. pontban.

Az adatbeadás befejezése után a program megszerkeszti és kirajzolja a TRIAGRAM-ot. A kirajzolás után egy gomb leütésére a MENÜ jelentkezik.

TÁROLT ADATOK BEHÍVÁSA:

Ezt választva először lehetőség van a lemez tartalomjegyzékének a megtekintésére.

Nemleges válasz, ill. a tartalomjegyzék kiírása után bekéri azt az ADAT-FILE nevet, amelyet betölteni kívánunk. A név beírására a meghajtott kezelés szabályai érvényesek. Max. 16 karakter és használható a * is.

Ezután a program megvizsgálja a disk állapotát, az eredményt kiírja. Hiba esetén várakozik a kijavításra.

Az adatok betöltése után a MENÜ jelentkezik.

MENÜ:

A menüből a funkció-gombok valamelyikének a leütésével lehet választani.

F1 A program befejezése, ill. új TRIAGRAM indítása. A memóriából az adatok törlődnek és a RETURN leütésére előlről indul a prg.

F3 Képernyőre írja a memóriában levő adatokat és a hozzátartozó szövegeket. A kiírás függőlegesen történik. Balszélen a triagram címe, majd a három oszlopsor adatai a hozzátartozó megjegyzéssel. Innen a MENÜ-re lehet visszatérni.

F5 Az aktuális adatokat lemezre menti ki. Beírás mint a betöltésnél.

VIGYÁZAT! Ha olyan nevet adunk meg, amilyen már van a lemezen, akkor az abban levő adatok elvesznek, mert a most betöltődő adathalmaz felülírja azt.

Kimentés után a menüre lép vissza. Innen lehet befejezni a munkát, ill. új TRIAGRAM-ot kezdeni.

F7 Kirajzolja a TRIAGRAM-ot. Egy gomb leütésére a MENÜ-re tér vissza.

Jó munkát és hasznos időtöltést kíván a program készítője:

Harma M. András

```

○ 556 return
560 for i=1 to 20: poke 54296,15
562 for t=1 to 20: next : poke 54296,0
○ 564 for j=1 to 3: next : next
566 return
○ 600 print kr$
604 print "[3:Dwn][RvOn] .[21: ] . . [RvOf][=K]"
605 print "[RvOn] keri a lemez tart.jegyzeket ?[RvOf][=K] i / n"
○ 606 print "[30:=U][=V]"
610 get g$: if g$="" then 610
○ 612 if g$="n" then return
614 if g$<>"i" then 610
620 print kr$
○ 625 open 1,8,0,"$": poke 781,1: sys 65478: get a$,a$,a$,a$: x$=chr$(0): print " 0 ";
630 for i=1 to 7: get a$,b$,c$,d$: print a$b$c$d$;: next : print
○ 635 get a$,a$,a$,b$: if st then sys 65484: close 1: goto 645
640 print asc(a$+x$)+256*asc(b$+x$);: goto 630
645 print "[2:Dwn] "tg$: poke 198,0: wait 198,1
○ 650 return
700 print "[Clr]": ad$=" adatlista "
710 for i=1 to 11: o=0: s=i+3: gosub 9: print
○ "[=L][RvOn]"mid$(ad$,i,1)"[RvOf][=J]": next
715 j=1: if len(bn$)<10 then j=1
○ 720 for x=1 to len(bn$)
725 o=5: s=j*x: gosub 9: print mid$(bn$,x,1): next
730 print "[Wht]": h=11
○ 735 for x=1 to ns
740 for y=1 to no
○ 745 o=h-(int(log(a(y,x))/log(10))): s=y-1: gosub 9: print a(y,x)" B"
750 next y: h=h+10: next x
780 for x=1 to 7: o=14: s=x+2: gosub 9: print "[Pnk]"mid$(mj$(1),x,1): next
○ 785 for x=1 to 7: o=24: s=x+2: gosub 9: print "[Yel]"mid$(mj$(2),x,1): next
790 for x=1 to 7: o=34: s=x+2: gosub 9: print "[LBlu]"mid$(mj$(3),x,1): next
    
```



```
○ 792 o=0: s=22: gosub 9: print "[Wht]menure:": print " f7": gosub 550 ○
794 get g$: if g$="" then 794
796 if g$="[f7]" then 200
○ 798 goto 794 ○
800 for x=1 to ns
○ 810 if x>0 then print "[Red]": if x>1 then print "[Yel]": if x>2 then print "[Blu]" ○
820 for y=no to 1 step -1
825 xp=10+(y-1)*bk+x-ns: yp=20+x
○ 830 k=int(a(y,x)/ma*20+.5) ○
840 if k>=1 then o=xp: s=yp: gosub 9: print f$
○ 842 if k=1 then o=xp: s=yp: gosub 9: print e$ ○
845 o=xp: s=yp: gosub 9: print "[RvOn][Wht][=*][RvOf]"
850 print "[Hom]"
○ 855 if x>0 then print "[Red]": if x>1 then print "[Yel]": if x>2 then print "[Blu]" ○
860 if k>=3 then for m=1 to k-2: o=xp: s=yp-m: gosub 9: print m$: next
○ 870 if k>=1 then o=xp: s=yp-k: gosub 9: print t$ ○
880 if k>=3 then o=xp: s=yp-m: gosub 9: print e$
890 if k<1 then o=xp: s=yp: gosub 9: print o$
○ 900 if y<no and k=z then o=ox: s=oy: gosub 9: print "[RvOn] [RvOf]" ○
910 z=k: ox=xp: oy=yp-k
915 next y: next x
○ 920 o=0: s=24: gosub 9: print "[Wht]"tg$: gosub 550: poke 198,0: wait 198,1 ○
930 return
990 :
○ 1000 j=1: if len(bn$)<10 then j=2 ○
1010 for x=1 to len(bn$): o=0: s=j*x: gosub 9: print mid$(bn$,x,1): next
○ 1020 for x=1 to 7: o=38: s=x: gosub 9: print "[Pnk]"mid$(mj$(1),x,1): next ○
1035 for x=1 to 7: o=38: s=x+8: gosub 9: print "[Yel]"mid$(mj$(2),x,1): next
○ 1045 for x=1 to 7: o=38: s=x+16: gosub 9: print "[LBlu]"mid$(mj$(3),x,1): next ○
1050 print "[Wht][Up]": b=4
1100 if ma-(int(ma/b)*b)<>0 then ma=ma+1: goto 1100
○ 1105 g=100 ○
1110 if max>100 then g=10: if max>1000 then g=1
○ 1115 for x=0 to 20 step 5 ○
1120 k=int((20-x)*ma*g/20+.5)/g
1125 l=len(str$(k))
○ 1130 o=9-ns-l: s=x: gosub 9: print k ○
1135 next : return
1140 :
○ 1200 print kr$: br=int(25/no)-1: bk=br+1 ○
1220 t$="[=*][RvOn]"
○ 1225 if bk>=2 then for x=1 to br: t$=t$+" ": next ○
1230 t$=t$+"[=*][RvOf]": m$="[H][RvOn]"
1240 if bk>=2 then for x=1 to br: m$=m$+" ": next
○ 1250 m$=m$+"[M][RvOf]": f$="[RvOn][=*]" ○
1255 if bk>=2 then for x=1 to br: f$=f$+" ": next
1260 f$=f$+"[M][RvOf][Up]"
○ 1265 for x=1 to br+2: f$=f$+"[Lft]": next ○
1270 o$="[=*][RvOn]"
○ 1275 if bk>=2 then for x=1 to br: o$=o$+" ": next ○
1280 o$=o$+"[=*][RvOf]": e$="[G][RvOn]"
1285 if bk>=2 then for x=1 to br: e$=e$+"[=Y]": next
○ 1290 e$=e$+"P[RvOf]": f$=f$+e$: k$="L" ○
1300 for x=1 to bk*no+ns: k$=k$+" ": next
○ 1305 k$=k$+"[s@]" ○
1315 for x=0 to 20: o=10-ns: s=x: gosub 9: print k$: next
1325 k$="[=@]L"
○ 1330 for x=1 to no*bk+ns: k$=k$+"[=@]": next ○
1340 k$=k$+"[s@]"
○ 1350 for x=0 to 20 step 5: o=9-ns: s=x: gosub 9: print k$: next ○
1360 return
1380 :
○ 1400 print kr$ ○
1409 print " .spc(19)."
○ 1410 print " kerem az triagram cimet[4: ][Dwn][s\]": o=9: s=6: gosub 9 ○
1420 nc=20: ak=31: av=93: gosub 10
1425 bn$=sz$
○ 1430 o=2: s=9: gosub 9: print "ha[Up][Lft].[Dwn]ny oszlop (adat) lesz (1-24)" ○
1435 nc=2: ak=47: av=57: o=34: s=9: gosub 9: gosub 10
1440 no=val(sz$): if no<1 or no>24 then gosub 560: print "[5:Lft] ": goto 1430
○ 1445 o=2: s=11: gosub 9: print "ha[Up][Lft].[Dwn]ny sor (diagram) legyen (1-3) ?" ○
1450 nc=3: ak=48: av=52: o=34: s=11: gosub 9: gosub 10
1455 ns=val(sz$): if ns<1 or ns>3 then gosub 560: print "[5:Lft] ": goto 1445
```

```

○ 1456 .
1457 ma=20: gosub 8
○ 1460 print "[Dwn] ke[Up][Lft].[Dwn]rem az adatokat ! (1-[5:9])"
○ 1465 for x=1 to ns
1470 print : print "[4: ]"x;"[Lft]. diagram [Dwn]"
○ 1475 for y=1 to no
1480 print " "y;"[Lft]. adat: ";
1485 nc=5: ak=47: av=58: gosub 10
○ 1510 a(y,x)=val(sz$)
1520 if a(y,x)=0 or a(y,x)>99999 then gosub 560: print tab(25)"[Up][Pnk]":
a(y,x)"[Wht][Dwn]": goto 1480
○ 1540 if a(y,x)>ma then ma=a(y,x)
1550 next y: gosub 550
○ 1570 print "[Dwn] ke[Up][Lft].[Dwn]rem az "x".sorhoz a megjegyze[Up][Lft].[Dwn]st."
1575 print "[Dwn] max. 7 kar. ";
1580 nc=7: ak=31: av=93: gosub 10
○ 1585 mj$(x)=sz$
1590 next x: return
○ 1700 print kr$
1710 print " U[s*]I."spc(14)"U[s*]I"spc(11)."
1712 print " BtBarolt adatok BuBj adat beiras"
○ 1714 print " J[s*]K"spc(15)"J[s*]K[2:Dwn]"
1716 ta=0: gosub 550
○ 1720 get g$: if g$="" then 1720
1722 if g$="u" then return
1724 if g$<>"t" then 1720
○ 1722 if g$="u" then return
1724 if g$<>"t" then 1720
1726 gosub 600
○ 1730 print kr$
1740 print "[2:Dwn] .[9: ][=@][5: ][=@]"
○ 1745 print " kerem a betoltendo file nevet ![Dwn][7: ][s\]"
1750 input " (max. 16 karakter) ";fi$
1755 if len(fi$)>16 or len(fi$)<1 then 400
○ 1760 ta=1: return
1900 print kr$
○ 1902 print "[6: ][=A][=R][=S] [=A][s*]I [=R] U[s*]I U[s*]I
[=A][s*]I U[s*]I [=A]IU[=S]"
1904 print "[7: ]B [=Q][=R]K B [=Q][s*][=W] B [=S] [=Q][=R]K [=Q][s*][=W] BJKB"
○ 1906 print "[7: ][=E] [=E]J[=X] [=E] [=E] [=E] J[s*]K [=E]J[=X] [=E] [=E] [=E]
[=E][2:Dwn]"
○ 1910 print "[8: ]3d diagram szerkeszto[Up][Lft][=@][Dwn] prg.[3:Dwn]"
1914 print " ."spc(33)."
1915 print " keszitette:[11: ]harna m. andras"
○ 1916 print tab(25)"szeged 1993[4:Dwn]"
1919 print tab(12)".[6: ]."
1920 print "[6: ]informaciot ker ? i / n"
○ 1925 gosub 550
1930 get g$: if g$="" then 1930
○ 1932 if g$="n" then return
1934 if g$<>"i" then 1930
1940 print "[Clr]"
○ 1945 print "[Up][Yel][9: ]k r a m p u s z o f t[8: ][Wht]"
1949 print "[8: ]."
○ 1950 print "a prg. harom oszlop-diagramot mutat be"
1952 print "[11: ]."
1955 print "egy koordinata-rendszerben."
○ 1957 print "[14: ]."
1960 print "az oszlopok szama max. 24 lehet."
○ 1962 print "[10: ]."spc(22)."
1965 print "a kirajzolas beirt vagy lemezen tarolt"
1967 print "[12: ]. [=@] ."
○ 1970 print "adatok alapjan tortenhet."
1972 print "[9: ].[6: ]. .[16: ][=@] "
○ 1975 print "a beadható max. érték: [5:9] (5. jegyü "
1979 print " ."
1980 print "szam.)"
○ 1984 print "[RvOn][s\][5: ][=@][4: ][=@][5: ][=@][20: ][=*]"
1985 print "[RvOn] lehetoleg ugyeljünk arra, hogy a 2-es "
1987 print "[RvOn][5: ]. .[30: ]"
○ 1988 print "[RvOn] sor értékei kisebbek legyenek mint az "
1990 print "[RvOn][5: ]. .[13: ]. .[14: ]"
1992 print "[RvOn] 1-ese es a 3-as sor értékei kisebbek "
    
```

```

1993 print "[RvOn][26: ].[12: ]"
1994 print "[=*][RvOn] legyenek mint a 2-es sore ![9: ][RvOf][s\]"
1995 print "[Dwn][5: ]tova[Up][Lft].[Dwn]bb: egy gomb[2:Up]"
1996 poke 198,0: wait 198,1: return
1997 rem ----- checksum -----
1998 poke 65,peek(122): poke 66,peek(123)
1999 v=peek(43)+peek(44)*256: read c$: p=0
2000 n=peek(v+2)+peek(v+3)*256: if n=1997 then end
2001 p=p+1: if p>len(c$) then read c$: p=1
2002 v=v+4: s=0: print n""
2003 b=peek(v): v=v+1: if b then s=s+b: goto 2003
2004 if (s and 31)<>asc(mid$(c$,p))-64 then print "error in"n
2005 goto 2000
2006 data "ecqc[ebjvxqhltyqdpdy\zvrujya]ukehzahw@vwnuwj@xvx[aeqls]bzenjt"
2007 data "w@\keot@_`r qoplz w]v[sx@tdhd_vs]\[r\mov[t_[^ [t_zoewn@q]nz_kkpo"
2008 data "ez]xchxn@_bhbxa^`vltk]mmahckh_mpphscxxkeolnzc_vamuojb_vrrjzu"
2009 data "_nh@fr^wzdggrux^uyxtnzzvxxwdibgoaza@ax^osh]o\c@mgbzb^bxcvcctu"
2010 data "_x@zgvlzralh_gfo[pwkjkkkopk]sk]kocgksgt]g_h"

ready.

```

"Megj.: A lista begépelése előtt olvassuk el a 93/12.számban megjelentt BASIC "LISTÁZÓ c. cikket. Az ott leírtakhoz annyit kell hozzátenni, hogy a használt nyomtatón megjelenő

^ (kitévő) helyett felnyíl
 \ (backslash) helyett fontjel
 _ (hángosköz) helyett balnyíl

irandó.

Alkalmazói
 program
 pályázat

A harmadik

A filenév rendezőprogram megírására az készített, hogy a Commodore gépek Basic-jéből hiányoltam azt az igen előnyös tulajdonságot, mellyel az IBM gépek rendelkeznek, miszerint a mágneslemez tartalomjegyzékének listázása ABC sorrendben történhet. Ennek az előnyét nem kell részleteznem, de csak párat megemlítve pl. gyorsabb kikereshetőség a listából, rendezettebb, jobb áttekinthetőség stb.

Mint ismeretes a Commodore gépek filenév listázása a programok felvételének sorrendjében történik.

A filenév rendezőprogram birtokában lehetőségünk nyílik mágneslemezeink fileneveinek bármikori ABC-sorrendbe rakására, mely a mágnes-

lemezen effektíven megy végbe. Így a filenév listázás bármely változatát alkalmazva ugyanazt a rendezett listát kapjuk. (Dir: Directory: LOAD"\$",8). Mivel én is többfajta Commodore géppel rendelkezem, törekedtem arra, hogy a program bármely típusú Commodore gépen futtatható legyen, ne tartalmazzon olyan Basic-parancsot, melyet a Basic V200 ne értene meg.

FELHASZNÁLÓI ÚTMUTATÓ A FILENÉV RENDEZŐ- PROGRAMHOZ

A filenév rendezőprogram betöltése, majd listázása után máris látható a program fejlécében az utalás arra, hogy több típusú gépen is használható. A programlistát REM-sorok tördelik, melyek a program által végzendő egyes részfeladatokat jelölik a jobb megérthetőség kedvéért.

Az elvégzendő részfeladatok:
 — címkép megjelenítés, előkészületek.
 — Írásvédelem ellenőrzés.
 — A tartalomjegyzék aktív blokkjainak memorizálása.
 — Filenevek memorizálása, filebejegyzések memorizálása.
 — Filebejegyzések rendezése a filenevek ABC-sorrendbe való rendezése alapján.
 — Rendezett filebejegyzések visszairása a mágneslemez blokkjaira.
 — Felszólítás a tartalomjegyzék kiiratására, ellenőrzésére.

Futtatva a programot először a címkép jelentkezik be, mely tartalmazza a program nevét, egy rövid mondatba foglalva a program lényegét, valamint utasít a rendezendő disk behelyezésére a meghajtóba. A címkép az „I” (igen) megnyomására vár ami indítja a rendezést.


```
10 rem *****
15 rem *
20 rem *      disk-filenev rendezo      *
30 rem *
40 rem *      keszitette: hegedus gyorgy  *
50 rem *      1993.07.22                  *
60 rem *      c16/116/64/+4/128          *
70 rem *****
75 :
80 print "[Clr][10:Rgt][3:Dwn][20:*]"
81 print "[10:Rgt]*[18: ]*"
82 print "[10:Rgt]* filenev rendezo *"
83 print "[10:Rgt]*[18: ]*"
84 print "{10:Rgt}[20:*]"
85 print : print : print : print "[3:Rgt]ez a program lehetove teszi, hogy"
86 print "[3:Rgt]egy mageslemez fileneveit abc sor-"
87 print "[3:Rgt]rendbe kapjuk meg a lemez tartalom-"
88 print "[3:Rgt]jegyzeknek listazasakor."
89 print "[3:Rgt][5:Dwn]kerem helyezze a rendezendo file-"
90 print "[3:Rgt]neveket tartalmazo mageslemezt a"
91 print "[3:Rgt]meghajtoba![4: ]indithatok? <i/n>"
92 get l$: if l$<>"i" then 92
95 print "[Clr]"
100 clr
110 dim t(18),s(18),j$(144),k$(144)
200 rem *****
210 rem *      az irasvedelem ellenorzese *
220 rem *****
230 open 15,8,15,"i": open 2,8,2,"#"
240 print#15,"b-r";2;0;18;0
250 print#15,"b-p";2;0
260 print#15,"u2";2;0;18;0
270 input#15,x1$
280 if val(x1$)<>26 then 340
290 print "[Clr][10:Dwn][Rgt][RvOn]ez a disk irasvedo füllel van ellatva![RvOf]"
: close 2: close 15
300 for v=0 to 5000: next v: run
310 rem *****
320 rem *a tartalomjegyzek blokkjainak*
325 rem *memorizalasa t(j)-be, s(j)-be*
330 rem *****
340 t(0)=18: s(0)=1: j=0: e=0: n=0: t=18: s=1
350 rem *****itt kezdodik a ciklus****
360 j=j+1
370 print#15,"b-r";2;0;t;s
380 print#15,"b-p";2;0
390 get #2,x$: if x$="" then x$=chr$(0)
400 t(j)=asc(x$): t=asc(x$)
410 get #2,x$: if x$="" then x$=chr$(0)
420 s(j)=asc(x$): s=asc(x$)
430 :
440 :
450 rem *****
460 rem *      a filenevek memo.-sa j$(i) *
470 rem *****
480 print "[Hom][10:Rgt][12:Dwn]filenevek betoltese"
490 print#15,"u1";2;0;t(j-1);s(j-1)
500 for x=0 to 7
510 print#15,"b-p";2;x*32
520 for y=1 to 32
530 get #2,x$: if x$="" then x$=chr$(0)
540 gg$=gg$+x$
550 next y
560 if asc(mid$(gg$,6,1))<>0 then e=e+1: k$(e)=gg$
570 gg$=""
580 next x
590 :
600 for x=0 to 7
610 print#15,"b-p";2;x*32+5
620 for y=0 to 15
630 get #2,x$: if x$="" then x$=chr$(0)
635 if x$=chr$(160) then 650
640 ff$=ff$+x$
650 next y
```

```

660 if asc(ff$)<>0 then n=n+1: j$(n)=ff$
670 ff$=""
680 next x
690 if t<>0 then goto 350
700 close 2: close 15: print "[Clr]"
710 rem *****
720 rem *   j$(i) rendezese +k$(i)   *
730 rem *****
735 print "[10:Rgt][12:Dwn]kis turelmet, rendezek..."
740 for i=1 to n-1
750 if j$(i)<=j$(i+1) then 800
760 s$=j$(i+1): sz$=k$(i+1)
770 for p=i to 1 step -1
780 if j$(p)>s$ then j$(p+1)=j$(p): k$(p+1)=k$(p): next p
790 j$(p+1)=s$: k$(p+1)=sz$
800 next i: print "[Clr]"
810 rem *****
820 rem * a rendezett k$(e)-k vissza- *
830 rem * irasa a lemez blokkjaira *
840 rem *****
850 open 15,8,15,"i": open 2,8,2,"#"
860 for a=0 to j-1
870 for b=0 to 7
880 print#15,"u1";2;0;t(a);s(a)
890 print#15,"b-p";2;b*32
900 print#2,k$(b+z+1);: print b+z+1". "mid$(k$(b+z+1),6,16)
910 print#15,"u2";2;0;t(a);s(a)
915 next b: b=0
920 z=z+8
930 for c=0 to 7
940 print#15,"u1";2;0;t(a);s(a)
950 print#15,"b-p";2;c*32
960 o$=chr$(0)+chr$(0)
970 print#2,o$;
980 print#15,"u2";2;0;t(a);s(a)
990 next c: c=0
1000 print#15,"u1";2;0;t(a);s(a)
1010 print#15,"b-p";2;0
1015 w$=chr$(t(a+1))+chr$(s(a+1))
1020 print#2,w$;
1030 print#15,"u2";2;0;t(a);s(a)
1040 next a
1050 close 2
1060 print#15,"i0": close 15
1070 print "[3:Rgt][12:Dwn]kerem ellenorizze a directory-t..."
1080 end
1081 rem ----- checksum -----
1082 poke 65,peek(122): poke 66,peek(123)
1083 v=peek(43)+peek(44)*256: read c$: p=0
1084 n=peek(v+2)+peek(v+3)*256: if n=1081 then end
1085 p=p+1: if p>len(c$) then read c$: p=1
1086 v=v+4: s=0: print n "[Up]"
1087 b=peek(v): v=v+1: if b then s=s+b: goto 1087
1088 if (s and 31)<>asc(mid$(c$,p))-64 then print "error in"n
1089 goto 1084
1090 data "oc[co@eozmsssgscrviruip\vewetve\cxhxebyekeadergrezzebe^wv^fr\"
1091 data "[chzzv]frjx[_fz[\eder[zmfgr[uecketn@mhinbhami_dndmehlncrgh@"

ready.
    
```

Megj.: A lista begépelése előtt olvassuk el a 93/12.számban megjelentt BASIC LISTÁZÓ c. cikket. Az ott leirtakhoz annyit kell hozzátenni, hogy a használt nyomtatón megjelenő

^ (kitevő) helyett felnyíl
 \ (backslash) helyett fontjel
 _ (hangosköz) helyett balnyíl

irandó.

A program futása közben informálja a felhasználót az éppen végzendő munkafázisról, így nyomon követhető érzékelhető a program üzemelése. A rendezés végeztével megkezdődik a rendezett file-nevek visszarakása a diskre, miközben sorszámozva megjelennek a képernyőn az egyes file-nevek, nyolcas csoportokban.

A program háromszor hajt végre inicializálást:

1. Az írásvédelem ellenőrzése előtt a program elején.

2. A rendezett file-nevek visszaírása előtt.

3. A rendezett file-nevek visszaírása végén.

Majd végezetül utasít a tartalomjegyzék helyességének ellenőrzésére.

MEGJEGYZÉS:

A file-név rendező a tartalomjegyzéken szereplő összes file-t rendezi: a törölt file-okat is. Ezért tapasztalható olyan jelenség, hogy a visszaírás során a képernyőre, törölt file-neveket is kiír a sorbarendezett helyére.

Ne lepődjünk meg azon, hogy ha a tartalomjegyzék ellenőrzésekor ezeket a törölt file-neveket nem találjuk a listában. Azon se lepődjünk meg, hogy ha egy ilyen törölt file-okat tartalmazó mágneslemez file-jait rendezve a legközelebb ki-mentett programunk neve nem a tartalomjegyzék lista végén fog megjelenni. Használjuk újra a file-név rendezőprogramot!

Sok sikert kívánok mindenkinek a program felfedezéséhez.

Hegedüs György

File-név rendezőprogram működési leírása

A címkép megjelenítése végén a 92 sor ciklusa várja a program indítását, mely kiszűri az esetleges téves billentyű lenyomását. Ha akarunk itt még kiléphetünk STOP gombbal a programból, de később már nem ajánlatos!

110 sor: az indexelt változók dimenzionálását végzi.

200—300 sor: az írásvédelem ellenőrzése

a mágneslemez 18-as track 00-ás szektorának (a disk fejlécét és a BAM-t tart.) olvasása majd visszaírásának próbálása, mely ha sikertelen volt, akkor a 270 sor INPUT utasítása az X1\$-be beírja a lemezhiba-üzenetet, s a 26-os számú hiba esetén kiírja, hogy a disk írásvédő leragasztással rendelkezik, majd kis idő múlva újraindítja az egész programot RUN-nal. Sikeres visszaírás esetén elkezdődik maga a rendező program.

340 sor: Változók alapértékeinek beállítása.

350—690 sor: Olvasó-memorizáló ciklus, mely hossza a beolvasott T értékétől függ.

Változók: T// : jelenlegi track száma
 S// : jelenlegi szektor száma
 J : ciklus számláló (hány blokkon található a tart.-jegyzék)
 T : következő track száma
 S : következő szektor száma
 FF\$: jelenlegi file-név
 GC\$: jelenlegi file-név-bejegyzés
 J\$// : file-nevek
 E : index számláló (file-név-bejegyzések részére)
 N : index számláló (file-nevek részére)
 K\$// : file-név-bejegyzések

Megjegyzés:

Egy file-név-bejegyzés a disk-en tartalmazza: — a file típusát
 — a file helyét a disk-en
 — a file nevét
 — a file blokkokban mért hosszát.

Minden blokkra a tartalomjegyzékből 8 file adata kerülhet. Minden ilyen blokk első két byte-ja a következő tartalomjegyzék-blokk helyét jelöli.

350—450 sor: Olvasó rutin, mely a következő olvasandó blokk helyét határozza meg. Memorizálja az adott track és szektorszámot T// és S// -be.

480 sor: A felhasználó informálása az adatbetöltés részfeladatról.

490—590 sor: Olvasó rutin, mely a file-név-bejegyzéseket memorizálja KS// -be.

600—680 sor: Olvasó rutin, mely a file-neveket memorizálja JS// -be.

735 sor: A felhasználó informálása a rendezés részfeladatról.

740—800 sor: Rendező rutin.

A file-neveket rendezi és vele együtt a file-név-bejegyzéseket. Gyorsrendező vagy más néven index-táblás rendező, mely lényege, hogy az éppen aktuális adatot behelyezi a névsor szerinti helyére majd veszi a következőt.

Magát a tárolt adatot nem bántja, csak a hivatkozási indexszámot módosítja.

850—1050 sor: A file-nevek alapján rendezett file-név-bejegyzések visszaírása a disk-re

A memorizált T// - és S// -k szerinti helyekre visszaírja a rutin, nyolcasával a rendezett file-név-bejegyzéseket, miközben a képernyőről informálódhatunk az éppen felvétel alatt álló file-névről, ami egymás alá íródik sorszámozva.

1060 sor: Végző inicializálás.

1070 sor: Felszólítás a tartalomjegyzék ellenőrzésére.

képernyő alján megjelennek az alkatrész adatai. Ezzel vége is van a programnak, de még megkérdezi, hogy kíván-e újra dekódolni. Ha az I-t írjuk be, akkor előlről kezdődik a program. Ha az N-t írjuk be, akkor először minden BASIC mutató visszaállítódik alaphelyzetbe, majd a program megsemmisíti önmagát.

Fontosabb változók:

A	= 2-es JOY állapota.
C, D	= A mutatónyíl pozíciója.
E	= A kiírandó sor száma.
S	= Hányadik bevétel?
JX/S/	= Bevitelek értékei.
O	= Hány ohmos az alkatrész?
KO	= Hány Kohmos az alkatrész?
PO	= Pontosság.
KI \$	= Üzemfeszültség.

Megjegyzés:

A programban eredetileg nagyon gyorsan lehetett vezérelni a nyilat. Ezért szinte lehetetlen volt kiválasztani az adott színt. Ennek kiküszöbölésére a programban elhelyeztem egy FOR-NEXT időzítő ciklust.

Az 1-es töltőprogram algoritmusai:

```
LOAD '1-ES TOLTOPRG.' ,8,1
```

```
↓
AUTOSTART 0
```

```
↓
FELIRAT 10
```

```
↓
2-es töltőprg. behívása 30
```

A 2-es töltőprogram algoritmusai:

```
START 0
```

```
↓
Ékezetes betűk behívása 10
```

```
↓
FŐPROGRAM behívása 30
```

A főprogram algoritmusai:

```
START 0
```

```
↓
SPRITE behívása 50
```

```
↓
KÉPERNYŐMASZK 100
```

```
↓
ADATBEVITEL 200
```

```
↓
Adatok kiértékelése 330
```

```
↓
Adatok kiírása 510
```

```
↓
ÚJRA kérdése 530
```

```
IGEN NEM
```

```
↓
```

```
RUN NEW
```

EXTRÁK: — ékezetes betűk

— RUN/STOP + RESTORE

védelem

— ékezetes betűk

— RUN/STOP + RESTORE

védelem

Varga Zoltán

```

○ 252 print "[Hom]": for p=1 to e: print "[Dwn]": next : print tab(11+jy)" ^ " : ○
253 for i=1 to 100: next : goto 210
○ 260 jy=jy+1 ○
261 if jy>d then jy=d ○
262 print "[Hom]": for p=1 to e: print "[Dwn]": next : print tab(11+jy)" ^ " : ○
263 for i=1 to 100: next : goto 210 ○
270 jx(s)=jy: return ○
290 sz(0)=val(sz$(6)) ○
300 print "[Hom][6:Dwn] 1. csIk: [RvOn][Blk] [Mar] [Red] [Org] [Yel] [Grn] ○
[Blu] [Pur] [Gry1] [Wht] [RvOf][Red]" ○
301 print "[Hom][7:Dwn]"tab(11)" ^ [Hom]" ○
302 c=0: d=9: e=6: gosub 209: print "[Hom]" ○
303 for p=1 to 150: next ○
305 print "[Hom][9:Dwn] 2. csIk: [RvOn][Blk] [Mar] [Red] [Org] [Yel] [Grn] ○
[Blu] [Pur] [Gry1] [Wht] [RvOf][Red]" ○
306 print "[Hom][10:Dwn]"tab(11)" ^ [Hom]" ○
307 c=0: d=9: e=9: gosub 209: print "[Hom]" ○
308 for p=1 to 150: next ○
310 print "[Hom][12:Dwn] 3. csIk: [RvOn][Mar] [Red] [Org] [Yel] [Grn] [Blu] ○
[Pur] [Gry1] [Yel]a[Gry1]e[Wht] [RvOf][Red]" ○
312 print "[Hom][13:Dwn]"tab(11)" ^ [Hom]" ○
314 c=0: d=10: e=12: gosub 209: print "[Hom]" ○
315 for p=1 to 150: next ○
320 print "[Hom][15:Dwn] 4. csIk: [RvOn][Mar] [Red] [Yel]a[Gry1] [Wht] ○
[RvOf][Red]" ○
322 print "[Hom][16:Dwn]"tab(11)" ^ [Hom]" ○
324 c=0: d=4: e=15: gosub 209: print "[Hom]" ○
325 for p=1 to 150: next ○
330 print "[Hom][18:Dwn] 5. csIk: [RvOn][Mar] [Red] [Org] [Yel] [Grn] ○
[Blu] [Pur] [Gry1] [Yel]a[Gry1]e[Wht] [Red]nincs[RvOf]" ○
332 print "[Hom][19:Dwn]"tab(11)" ^ [Hom]" ○
334 c=0: d=11: e=18: gosub 209 ○
350 rem ○
355 rem *** adatok kiertekelese *** ○
357 dim sz(50): j=0 ○
360 data "0","1","2","3","4","5","6","7","8","9" ○
370 for i=0 to 9: read a$(i): next i ○
380 sz$(1)=a$(jx(1)): sz$(2)=a$(jx(2)) ○
385 sz$(6)=sz$(1)+sz$(2) ○
400 sz(0)=val(sz$(6)) ○
410 if jx(3)=8 then jx(3)--1 ○
415 if jx(3)=9 then jx(3)--2 ○
420 sz(2)=10^(jx(3)+1) ○
430 data 1,2,5,10,20 ○
440 for i=0 to 4: read a(i): next i ○
445 sz(4)=a(jx(4)) ○
    
```

```

450 data "100","200","300","400","500","600","700","800","1000","2000","900"," "
460 for i=0 to 11: read a$(i): next i
470 if jx(5)=11 then jo=1: jy=11
475 sz(3)=val(a$(jy))
480 o=sz(0)*sz(2)
483 ko=o/1000
488 po=sz(4)
490 if j0=0 then ki$=str$(sz(3)): ki$=ki$+"v"
495 if jo=1 then ki$="nincs adat"
500 rem
510 rem *** adatok kiirasa ***
520 print "[Hom][20:Dwn][Rgt]";o;"\=";ko:"k\"
525 print " pontossAg:";po;"% max.fesz.:";ki$
528 for i=1 to 1000: next
530 print "[Dwn]"tab(13)"Ujra ? (i/n)"
540 get a$: if a$="" then 540
545 if a$="i" then run
550 if a$="n" then 560
555 goto 540
560 poke 53269,0: poke 53280,14: poke 53281,6: print "[Clr]": poke 44,8:
    poke 2048,0
570 poke 53272,21: print "[LBlu]": new
571 rem ----- checksum -----
572 poke 65,peek(122): poke 66,peek(123)
573 v=peek(43)+peek(44)*256: read c$: p=0
574 n=peek(v+2)+peek(v+3)*256: if n=571 then end
575 p=p+1: if p>len(c$) then read c$: p=1
576 v=v+4: s=0: print n"[Up]"
577 b=peek(v): v=v+1: if b then s=s+b: goto 577
578 if (s and 31)<>>asc(mid$(c$,p))-64 then print "error in"n
579 goto 574
580 data "^aipykpfolmocyofmzd_o^alhfop\r[j]^ttcqtscqtcnnjajb]djapvjaclj"
581 data "dvso\odnz[nvxcne]ww[rkic^qoc^anjf@vbxv"
    
```

ready.

Megj.: A lista begépelése előtt olvassuk el a 93/12.számban megjelentt BASIC LISTÁZÓ c. cikket. Az ott leírtakhoz annyit kell hozzátenni, hogy a használt nyomtatón megjelenő

^ (kitevő) helyett felnyíl
 \ (backslash) helyett fontjel
 _ (hangosköz) helyett balnyíl

irandó.

ASCII-kódból képernyőkód

A C64-nél a képernyőkódok eltérnek a nemzetközileg használatos ASCII-kódotól (American Standard Code for Information Interchange). Emiatt különösen a szövegfeldolgozásban fontos egy olyan függvény (funkció), amelyik az ASCII-kódokat képernyőkódokká alakítja. Ez a függvény a következő:

```
DEF FN AB(A) = A + 33 * (A=255) + 64 * (A > 63) +
+ 32 * (A < 96) - 32 * (A < 160) + 64 * (A < 191)
```

Ezek után a PRINT FN AB(ASC("X")) függvényhívás az X karakter képernyőkódját írja a képernyőre.

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 azokat határainkon túli (erdélyi, kárpátaljai és szlovákiai) magyar iskolákhoz. Köszönettel:

Országos Commodore Egyesület

Indítási címek gyűjteménye

Plus/4 gépen futó játékokhoz

Sokszor okoz bosszúságot, hogy bár sikerült megszereznünk egy-egy jó játékot, de hiába próbálkozunk, nem tudjuk elindítani őket. Az alábbi címek ehhez nyújtanak segítséget.

Megjegyzés a listához. Egy-egy játékprogramnak több, többé vagy kevésbé tömörített változata is közkézen forog. Így lehetséges, hogy ugyanahhoz a programhoz több indítási cím is tarthat.

Ha az elindítási cím valakinél nem működne, az nem azért történt, mert rossz adatot közöltünk, hanem azért, mert nem ugyanazon cracker által feltört változattal rendelkeznek.

Összeállította:

Loós József

Sor-szám	Program (file) neve	Indítási címek
1.	A.C.E.	SYS 18625
2.	AIRWOLF	SYS 7633
3.	ALIENS	SYS 4125
4.	APOLLO MISSION	G 3B2E
5.	ARENA 3000	SYS 6708
6.	AURIGA	SYS 5222
7.	AUTOBAHN	SYS 16029
8.	BABY BERKS 1.	SYS 6144
9.	BANDITS	SYS 8112
10.	BASEBALL	SYS 15872
11.	BATTLE	SYS 4111
12.	BEACH HEAD	SYS 29184
13.	BERKS III	G 1088,G 108A,SYS 4232
14.	BIATLON	G 4C4A
15.	BIG-MAC	G 3131,SYS 7000, G 1B58,SYS 12593
16.	BLAGGER	G 241A,SYS 9242
17.	BLAZE	G 2600,SYS 9728,SYS 9742
18.	BMX RACER	SYS 16288,SYS 8330
19.	BONGO C. S.	SYS 4109
20.	BRIDGEHEAD	SYS 8192, G 2000
21.	CATACOMB	SYS 4352
22.	COMMANDO	SYS 9843, G 2673,SYS 12849
23.	CORMAN	SYS 4112
24.	CRUNCHER	SYS 8448
25.	CSODAS SIMON	SYS 14547
26.	CUOMO	SYS 6500
27.	CUTHBERT IN TOMBS	G 100E
28.	CYBORG	SYS 11103
29.	DARK TOWER-1	G 1E40,SYS 7744
30.	DEATH RACE	G 392A
31.	DEFENCE (XZAP)	SYS 4352,SYS 4346
32.	DELAY-2	G 1010
33.	DIAGON	SYS 12888
34.	DIZZY DICE	SYS 4136
35.	DKS	SYS 4128
36.	DORK'S DILEMMA	SYS 6501
37.	ENIGMA	G 4800
38.	EXORCIST	SYS 4576
39.	FALCON PETROL	G 3600,G 2000,SYS 8192
40.	FINDERS KEEPERS	SYS 17925
41.	FINGERS MALONE	G 16A0,SYS 23176,SYS 5792
42.	FIRE ANT	G 1B03,SYS 6915,G 1B01
43.	FLIGHT PATH 737	SYS 16124
44.	FLING A.C.E.	SYS 18625
45.	FLOWER	SYS 4130
46.	FOOTBALL	G 3716
47.	FORMA-1	G 3A8C,SYS 14988
48.	FURY	SYS 4120
49.	GALAXIANS	SYS 12960

Sor-szám	Program (file) neve	Indítási címek
50.	GHOST TOWN	G 3EF9,SYS 7042
51.	GHOSTS N GOBLINS	SYS 4343
52.	GOLDRUSH	G 1004,SYS 4100
53.	GRAPHIC MASTER	SYS 4128
54.	GREEN BERET	G 1B00
55.	GULLWING FALCON	SYS 8192
56.	GUNLAW	SYS 4120
57.	GWNN	SYS 14336
58.	HARBOUR ATTACK	SYS 9216
59.	HUNGARORING	SYS 8184,G 1FF8
60.	ICICLE WORKS	SYS 24576
61.	IN THE COOCER	SYS 4352
62.	INTERNATIONAL KARATE	SYS 17920
63.	INVADERS	SYS 4110
64.	JET SET WILLY	SYS 10752
65.	JOEY	G 1020
66.	JUMBO JET	SYS 25600
67.	KANE	SYS 15868
68.	KARATE KING	SYS 18177
69.	KICKSTART	SYS 8192
70.	KNOCK OUT	SYS 8448
71.	KUNG-FU KID	G 2000,SYS 8112,SYS 8192
72.	L'UMO	SYS 6500,G 1964
73.	LAS VEGAS	SYS 8192,SYS 1552
74.	LASER 900	SYS 4096
75.	LASER ZONA	SYS 12928
76.	LAWN TENIS	SYS 9950,SYS 9952
77.	LEAPER	SYS 16385,G 4001
78.	LEGONNAIRE	SYS 4112
79.	LOCO CO-CO	G 1300,SYS 5158,SYS 5342,SYS 5389
80.	LOCO CO-CO	SYS 4128
81.	LOCOMOTION	G 1100,SYS 4352
82.	MAIL TRAIL	SYS 4098
83.	MANIC MINER	SYS 7840
84.	MATRIX	SYS 4122
85.	MAYHEM	G 118C
86.	MEGAZAP	G 1000
87.	MERCENARY	SYS 23552
88.	MILITARY	SYS 4608
89.	MONKEY MAGIC	G 2955
90.	MR. PUNIVERSE	G 1AF4
91.	MYRIAD 16	G 1A00,SYS 10755
92.	NETRUN 2000	SYS 4100
93.	NINJA MASTER	SYS 15104
94.	NUMBER BUILDER	SYS 16384
95.	NUMBER CHASER	SYS 16384
96.	OLYMPIC	SYS 13103
97.	OLYMPIC SKIER	SYS 16192
98.	ONE MAN'S DROID	SYS 5273
99.	OUT ON A LIMB	SYS 5888,G 1004,SYS 5883
100.	OUT ON A LIMB 2-4.	SYS 4100
101.	PACMAN	SYS 4112
102.	PACMANIA	SYS 4113,G 1011,SYS 4112
103.	PANCHO	SYS 4112
104.	PETALS OF DOOM	G 3CE9,SYS 15593
105.	PETCH	SYS 20736
106.	PIRATES	SYS 9142
107.	POTTIT	SYS 5728
108.	PROSPECTOR PETE	G 1800
109.	PULSAR	SYS 10078,SYS 10072
110.	PUNCHY	SYS 4112
111.	RAFFLES	G 3B70,SYS 15216
112.	RAIDER	G 1000,SYS 6928
113.	RAILING	SYS 8448
114.	RAZOS UT	SYS 20480

(Folytatás a 26. oldalon)

C-64 bővítések

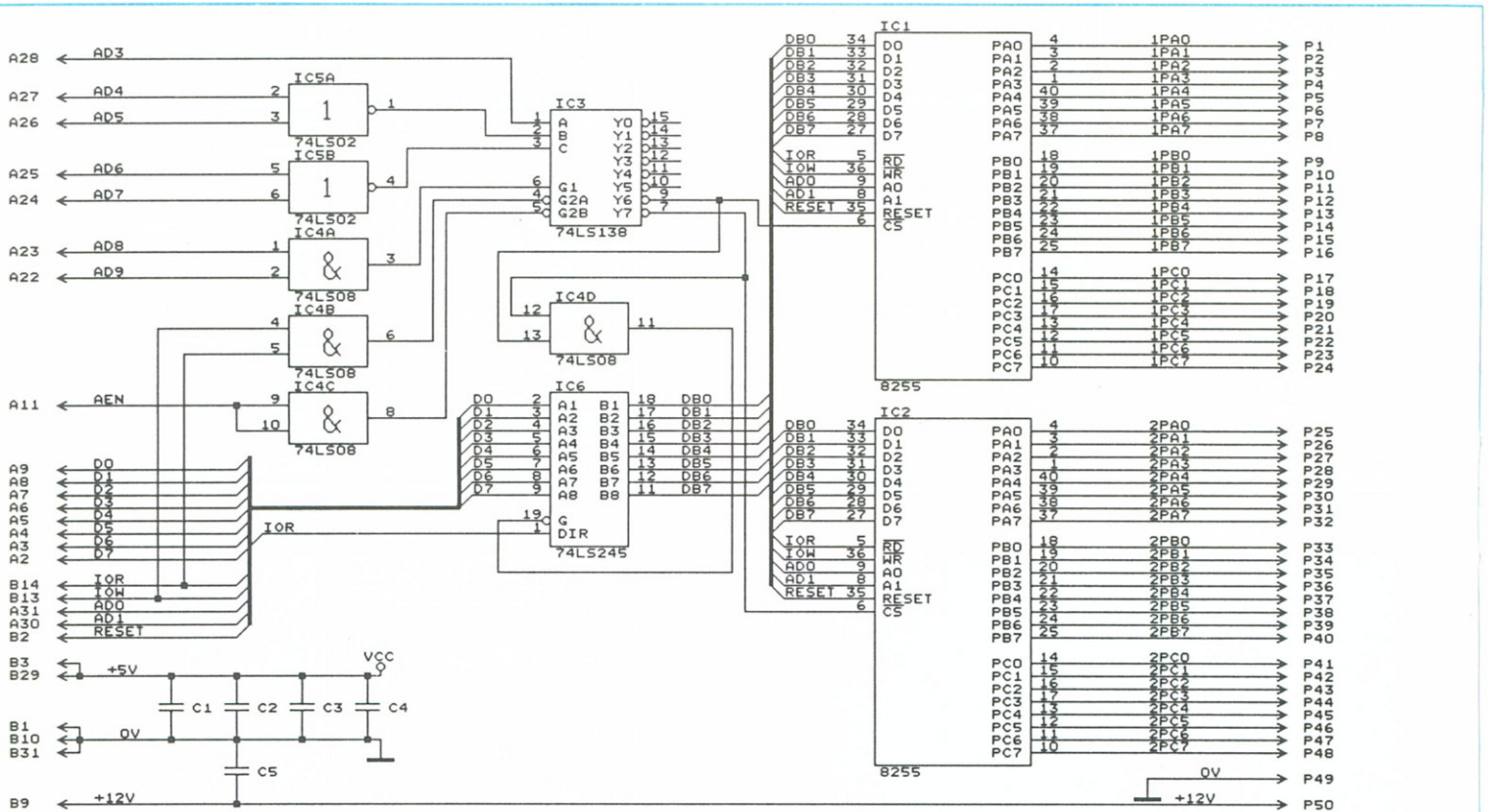
48 CSATORNÁS I/O KÁRTYA IBM PC-HEZ

Ha a számítógépet nemcsak irodai feladatok elvégzésére használja, hanem valamilyen műszaki feladat megoldására is alkalmazza, ahol felmerül a számítógép és a külvilág kapcsolatának megteremtése, biztosan szüksége lesz egy olyan kártyára, ami ezt elvégzi. Ezt a kártyát érdemes olyan formában megtervezni, hogy univerzálisan, sokféle feladat elvégzésére alkalmas legyen. Rendelkezzen sok ki- és bemeneti ponttal, amelyek a

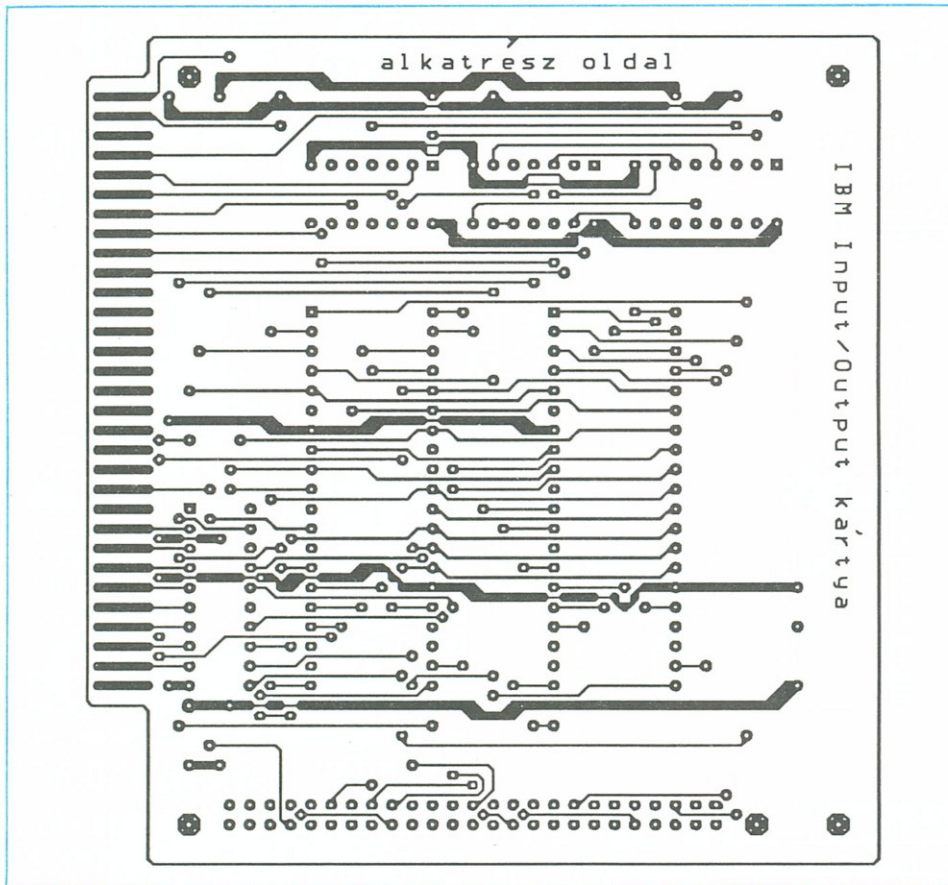
probléma megoldásának megfelelően szoftverből működtethető. Ez a leírás egy ilyen bővítő elkészítését ismerteti olyan szinten, hogy az áramkör ez alapján az üz-

letemben kapható egységcsomagból, kevés hardver ismerettel is megépíthető legyen. A többi már szoftverrel elintézhető. Az I/O kártya kapcsolási rajza az 1. ábrán látható. A felépítése egyszerű, csupán két PPI-ből és a hozzátartozó címkóder komplexumból áll. A címkóder az AD0, AD1...AD9, IOR, IOW vonalak állapotának dekódolásával a következő címeket generálja:

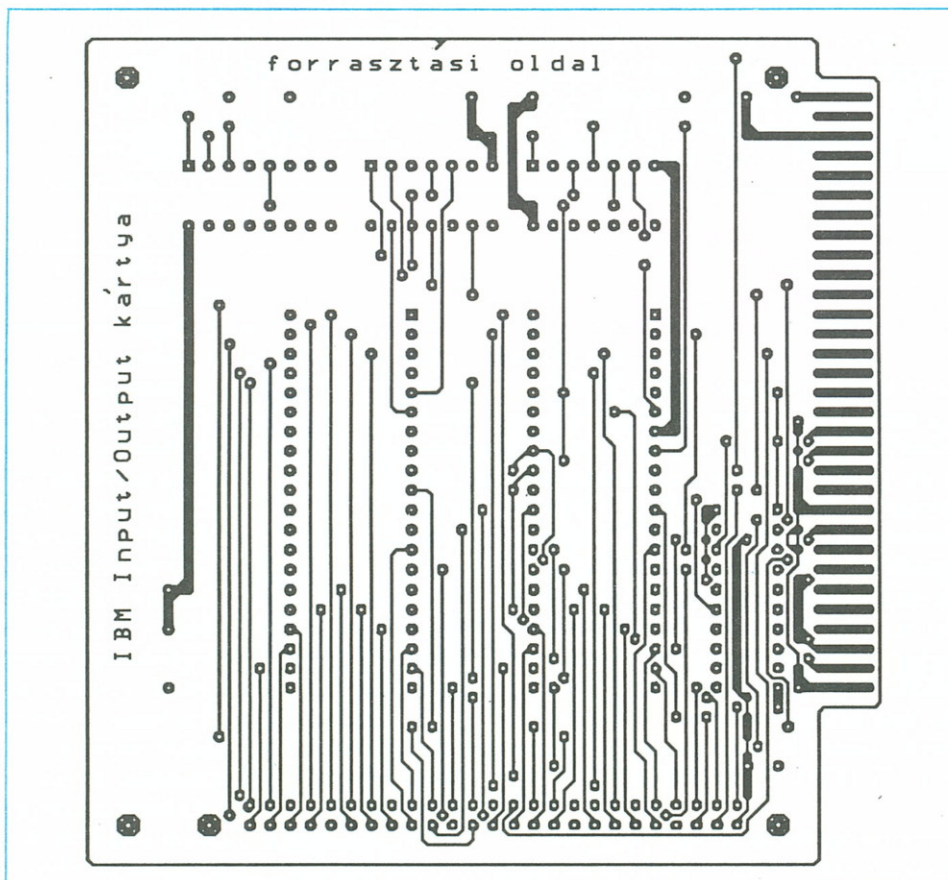
IC1 A port:	300H	304H	IC2 A port:	308H	30CH
IC1 B port:	301H	305H	IC2 B port:	309H	30DH
IC1 C port:	302H	306H	IC2 C port:	30AH	30EH
Parancs:	303H	307H	Parancs:	30BH	30FH



Title		IBM PC-hez I/O kártya
Size	Document Number	REV
A	901118BJ-203	1.0
Date:	January 4, 1991	Sheet 1 of 1



2. ábra. Alkatrész oldal



2. ábra. Forrasztási oldal

A címdekódoló redundáns, mert az AD2 vonal állapotát nem figyeli, ezért mind a két cím érvényes és teszőlegesen használható.

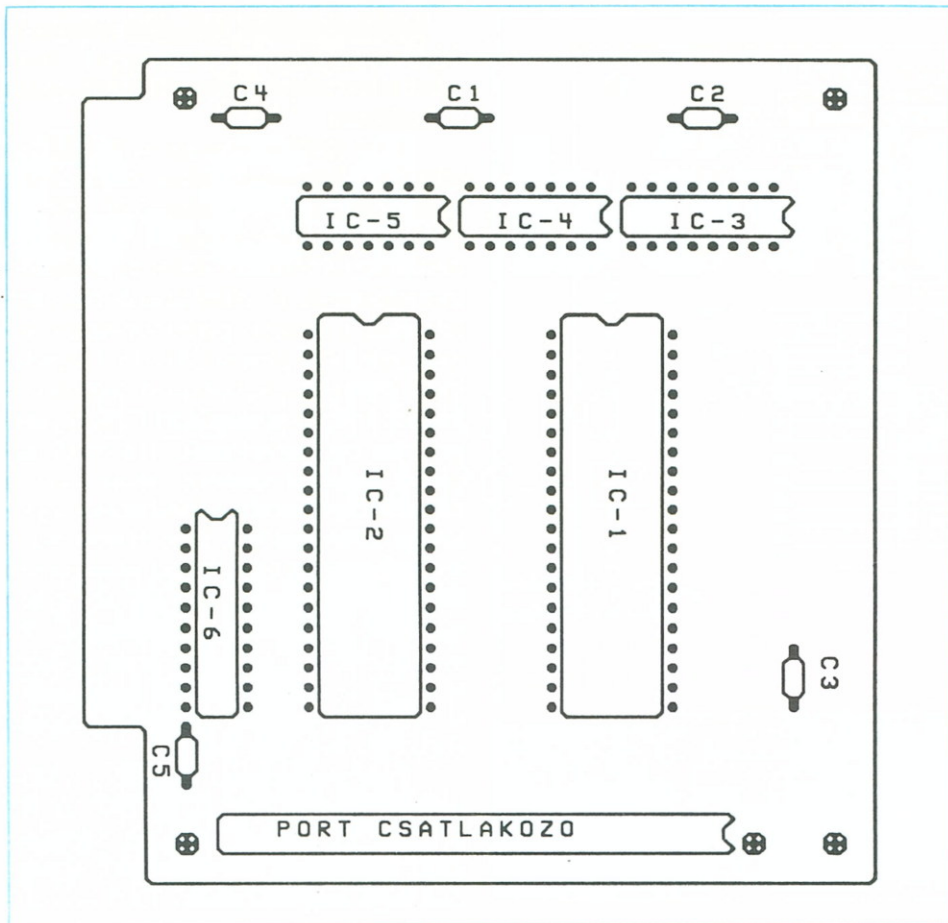
A két 8255 adatvonala az IC6 8 bites kétirányú buszmeghajtóval kapcsolódik a számítógép adatbuszára (D0...D7), a 2×31 pontos sloton keresztül. Ez lehetővé teszi, hogy XT-hez és AT-hez egyaránt felhasználhassuk. A külső áramkörök csatlakozását a panellel párhuzamosan elhelyezkedő 2×25 pólusú CANON-G csatlakozó biztosítja. A kivezetett csatlakozópontok a 8255-re megadott TTL egységgel (fan out) terhelhetők. Ha ennél nagyobb terhelést szeretne rákapcsolni, a teljesítményillesztést a feladatnak megfelelően biztosítani kell. Erre itt nem tudok konkrét receptet adni, hiszen nemigen van két egyforma feladat, ami egy az egyben ugyanazt az illesztést alkalmazná. Ha nagyfeszültségű rendszert kíván működtetni, a hálózat leválasztását optocsatolóval vagy más hasonló szigetelést biztosító eszközzel meg kell oldani.

Ilyen leválasztásra ajánlom a triakos kártyát, aminek leírása az 1993-as RT évkönyv 33. oldalán található. Mintadarabja üzletben megtekinthető, egységcsomagja megvásárolható. Ez 9 egymástól független nullátmeneti kapcsolót tartalmaz, ami egyenként 600 W-tal terhelhető.

Eredetileg számítógéppel vezérelt futófény és fényreklám számára lett kifejlesztve, de használható minden olyan feladat vezérlésére, ahol elegendő a nullátmeneti bekapcsolás, nem követelmény a folyási szög vezérlés: például hőmérséklet szabályozás, a motor, szelep ki-be kapcsolás stb.

Az áramkör szerelése, élesztése

Az áramkör a 2. ábrán látható kétoldalon huzalozott, furatgalvanizált NYÁK-lemezre építhető fel. Mivel ennek elkészítése házi eszközökkel körülményes, célszerű a HOBBI ELEKTRONIKA üzletben megvásárolni. Tel.: 1228-892. A NYÁK alapos ellenőrzése után a csatlakozósort a slotnak megfelelően igazítsa méretre, majd ültesse be az alkatrészeket a 3. ábra alapján. A 8255 alá használjon foglalatot, hogy szükség esetén könnyen cserélhető legyen. Ha mindennel kész, még egy alapos ellenőrzés után dugaszolja be egy szabad bővítőcsatlakozóba és kapcsolja be a gépet. Ha rendben bejelentkezik a rendszer, megpróbálkozhat a kártya felprogramozásával. A ki- és bemeneteket



3. ábra.

a megadott portcímeken tudja írni, olvasni. A címzett bájt minden bitje megfelelő egy-egy portvonalnak. Input üzemmódban beolvasva a bájtot megkapja a portok aktuális állapotát.

Output üzemmódban pedig a bájt megfelelő módosításával változtathatja az egyes kimeneti vonalak állapotát. A 8255 felépítéséről, programozásáról, regisztereiről nem adok részletes információt, ez-

zel számtalan katalógusban találkozhat, de az RT is több alkalommal bemutatta már: pl. a 1992/9-es számban, hogy egyet említsek.

A kártya programozásával itt nem foglalkozom, az a megadott cím és parancsbájtok ismeretében a felhasználóra vár.

I/O kártya árajánlata

Az I/O kártya mintadarabja megtekinthető, egységcsomagban vagy készre szerelve megvásárolható a szerző üzletében. Ára egységcsomagban 3500,— Ft.

Külön vásárolható kétoldalas furatfémzett panel, ára 1200, Ft. Folyamatosan kapható egységcsomag, panel és részegység a C-újságban korábban megjelent cikkeimhez is. Ezek egy része működés közben megtekinthető, kipróbálható. Árajánlat a 2. oldalon.

A vidéki olvasóknak segít a szerző levelező-egységcsomagküldő szolgáltatása. A megrendelt csomagot postán utánvétellel elküldöm. Telefonon és levélben is rendelhet.

A HOBBI ELEKTRONIKÁHOZ nem kell hosszú levél. Rendelését röviden egyértelműen közölje.

Levél cím:

HOBBI ELEKTRONIKA

1656 BUDAPEST PF.50.

Üzlet cím: BUDAPEST

VII. Dózsa György út. 16.

Jobbágy u. sarok

Nyitva: H—P 10—17-ig,

zárás után üzenetrögzítő.

Tel./Fax: 122-8892

(Folytatás a 23. oldalról)

Sorszám	Program (file) neve	Indítási címek
115.	REACH FOR THE SKY	G 2000,SYS 8192
116.	RED MOON	SYS 4128
117.	ROBIN TO THE RESCUE	SYS 9984
118.	ROCKMANN	SYS 4144,SYS 4102
119.	ROCKMANN 2.	SYS 4112
120.	SEA STRIKE	G 1300
121.	SETTING	SYS 6400
122.	SHAOLIN	G 2000,SYS 8192
123.	SKIER	SYS 16192
124.	SKRAMBLE	SYS 14848,G 1E40
125.	SKYHAWK	G 1061,SYS 4178,SYS 4176,G 1060
126.	SNOOKER	SYS 16384
127.	SOLAR	SYS 8448
128.	SPACE FRECKS	G 1010,SYS 4224,G 1080,SYS 4112
129.	SPACE SWEEP	G 2000,SYS 8192
130.	SPECTIPEDE	G 127A
131.	SPEED KING	G 3DB0
132.	SQUIRM	SYS 9220
133.	STAR COMMANDER	SYS 4120
134.	STELAR WARS	SYS 4110
135.	STREET OLYMPICS	SYS 13100
136.	STROM A.B.	G 5100
137.	SUICIDE RUN	SYS 8448
138.	SUPER CRAN	SYS 4950
139.	TANYERTORONY	SYS 11776
140.	TENISZ	SYS 9952
141.	TERRA COGNITA	G 2000,SYS 8192

Sorszám	Program (file) neve	Indítási címek
142.	THAI BOXING	SYS 4864
143.	TIMESLIP	SYS 4158
144.	TOM +4	SYS 22528
145.	TOM THUMB	SYS 22528
146.	TORPEDO ALLEY	G 24F2,SYS 4120
147.	TOWER OF EVIL	SYS 5120
148.	TRAIL MAIL	G 1000
149.	TRAILBLASER	SYS 9408
150.	TURBO RACER	SYS 9506
151.	TUTTI FRUTTI	G 2000,SYS 8192
152.	TYCOON TEX	SYS 6144
153.	US DRAG RACING	SYS 10252
154.	VEGAS JACKPOT	SYS 8192
155.	VOX	SYS 16199
156.	WATER GRANDPRIX	SYS 7450,G 100E
157.	WILD CUP	SYS 14100,G 3714
158.	WILD WESTERN	SYS 4112
159.	WORLD CUP (FOOTBALL)	SYS 14102
160.	XARGON WARS	SYS 7296
161.	XZAP	SYS 4346
162.	YIE AR KUNG FU 1.	SYS 10795,SYS 12900
163.	YIE AR KUNG FU 2.	SYS 10795
164.	ZAP'EM	G 3360,SYS 13152
165.	ZAXON	SYS 6972
166.	ZODIAK	SYS 7748
167.	ZONE CONTROL	G 1002

C64 & AMIGA ROVAT

BUMBLE BEE & FRIEND TWO CREW



DEMÓK

Helló mindenkinek!

Ez a cikk egy új rovat indulásának az első állomása. Ebben a rovatban a C64-es demók világába szeretném elkalauzolni az olvasókat. Erre a rovatra azért is nagy szükség van, mert sokan nem is tudják, hogy mit is jelent ez a szó a számítógépek világában, hogy „demó”. A „demó” a „demonstration” angol szó barátságosabb, rövidített változata. Jelentése: bemutató. De felmerülhet a kérdés az embelekben, hogy kik készítik ezeket a bizonyos demókat? Körülbelül az 1980-as évek eleje felé számítógépes csapatok alakultak. Ezeknek a csapatoknak a tagjai általában olyan emberek voltak, akik az átlagosnál többet foglalkoztak a gépükkel, és olyan szintre fejlesztették számítógépes tudásukat, hogy képesek voltak produkálni saját készítésű dolgokat. Ezek a csapatok (világszerte) felvették egymással a kapcsolatot, és megalakították a C64-es scene-t. Ez a scene mindmáig él. A scene-re csak olyan csapatok tudnak bekerülni, akik tudnak valami keveset produkálni a 64-esen, és ezáltal a scene-n lévő csapatok elismerik a létezésüket és befogadják a soraikba. Ez a scene a csapatok között kölcsönösen egy Publikumot ad, de emellett remek lehetőséget ad az egymás közötti versengésre. Persze a scene-n belül nem minden csapat csinálja ugyanazt.

A scene megoszlik alapvetően legal és illegál csapatokra (de van olyan is aki egyszerre mind a kettő!). Az illegál csapatok általában crack-okkal (játékok töréseivel) foglalkoznak. Ilyenek például: IKARI+TALENT, DOMINATORS, LEGEND, RED SECTOR stb.... A legal csa-

patok játékokat, diszk-újságokat, zene illetve grafika kollektciókat és demókat készítenek. Ilyen például COSMOS, ORIGO, BLACK MAIL, 20CC, CREST és még sorolhatnám. De mit is tartalmaznak a demók? Az igazi demók három fő dolgot tartalmaznak:

1. grafikát,
2. zenét,
3. kódot.

Grafika alatt saját készítésű rajzokat kell érteni. Pl.: a csapat nevét megjelenítő logo, karakterkészletek, valamilyen igazi képet ábrázoló Picture.

Zene alatt saját készítésű, de nem mindig saját komponálású zenét kell érteni. Azok a csapatok, akiknek nincs zenészüik, azok általában más demóból vagy esetleg játékból lopott zenétet használnak demóikban.

Kód alatt természetesen a gépi kódú rutint kell érteni, amely valamilyen vizuális effektet rak ki a képernyőre. Persze demó és demó között van különbség, mert nem mindegy, hogy melyik csapat csinálja. Azok a kóderek, akik még nem tudnak olyan jól kódolni azok nem tudják kihasználni a gép képességeit 100%-osan. A jobb kóderek mindig olyan rutinokat képesek előállítani, amely rutinok a maguk műfajában valamivel jobbak mint az őket megelőzőek, vagy pedig teljesen újak, és ezért az újdonság erejével hatnak. Ilyen kóderek pl.: a CROSSBOW a CREST nevű csapatból.

Természetesen a számítógépes scene tagjai nem eredeti nevüket használják a scene-n, hanem kitalálnak maguknak egy álnevet.

A CROSSBOW is egy ilyen álnév, és a jelentése nyílpuska. Ez is mutatja, hogy teljesen mindegy, hogy mit választ magának az ember, mert ez a fantáziájára van bízva. De miért csinálnak demókat? Vannak akik csak hobbyból, vannak akik azért, hogy a top-listákon minél magasabbra kerüljenek, és vannak akik azért, hogy egy-egy demóversenyen induljanak a fődíj reményében. Ezeket a demóversenyeket úgynevezett számítógépes találkozókön, angolosan Partykon rendezik meg. Persze a Partykat is a scene valamelyik vállalkozó szellemű csapata szokta megszervezni. Külföldön már egy-két rangosabb Partyn elég magas díjakat lehet kapni egy demóversenyen (átszámítva kb. 20—30—40 ezer forintot).

A demók világában elég népszerűek az olyan Part-ok, amelyek world record-okat, vagyis világrekordokat tartalmaznak. Ez azt szokta jelenteni, hogy egy bizonyos effektből ugyanannyi raszter-idő alatt többet tud kirakni a képernyőre, mint az előző rekord. De, hogy ezt meg lehessen érteni, kitérek a raszteridő fogalmára:

Egy 6 színű scroll mozog fel-le a képernyőn. Mire a scroll-text véget ér a főlts is befejeződik, és a Part automatikusan elindul.

1. PART

Alul egy 1×1 karakteres fehér scroller halad. Felül két vízszintesen duplaszínű hullámzó szürke effektet rak ki 24 karakter magasan.

2. PART

X+Y irányban nagyított multicolour sprite grafikák mozgatása az alsó-felső és oldalsó kereteken felfelé+jobbra ill. balra. Nagyon nehéz időzítésű Part!

3. PART

Grafika Part. Felül egy AFLI logó, rajta sprite-bobok köröznek. Alul egy-két képernyő széles bitmap logó mozog fel-le és jobbra, illetve balra. A képernyő közepén egy vízszintesen nagyított spritescroll kommentálja az eseményeket.

4. PART

3 színű multicolour FILLED VECTOR-ok. Az object-ek transzformálódnak, mind a 3 dimenzió tengely körül forognak, és a mozgásuk realtime-ban kalkulálódnak. Mindezek mellett a vektor rutin nagyon gyors, kb. 3—4 frame körül lehet. A Part csúcspontja egy áttetsző (glenz) és transzformáló FILLED VECTOR, amely szintén gyorsan forog. A Partból a space megnyomásával akkor léphetünk csak ki, ha az összes object-et láttuk már.

5. PART

Egy stretch-elő scroll halad a képernyőn. Felül egy BOOZE DESIGN logó parkol. Ez nem is lenne nagy szám, hiszen

már rengeteg demóban találkozhattunk hasonló scrollokkal. A poén az, hogy a scroll az alsó keretre is kinyúlik! És emellett nincsen X irányban nagyítva! Szép raszterszíneket is láthatunk a scroll betűi alatt.

6. PART

A lemez megfordítása után egy szép shade-bob rutinban gyönyörködhetünk. A rutin 7 színű, 24 karakter széles és kb. 20 karakter magas.

7. PART

FLI-logó mellett a sideborder le van bontva. A kereten spriteokat láthatunk szép raszterszínekkel alatta. A kép alján egy nagy egyszínű scroll mozog fel-, illetve lefelé. Szép színes Part. És kissé nehézkes lehetett megkódolni, akárcsak a többi Partot...

8. PART

Hatalmas egész képernyős 3 színű transzformálódó FILLED VECTOROKAT láthatunk. A nagy méret miatt a rutin elég lassú, de ahhoz képes tűrhető gyorsaságú. Én úgy 10–12 frame-re tippelem.

9. PART

THE END Part. 2×2-es karakterekkel szöveg váltakozik. Semmi különös nincsen a képernyőn, de azért szép a szöveg váltó effekt.

Átfogó véleményem a demóról:

Mindvégig magas színvonalú kód tudásáról adott tanúbizonyságot az egész demó kódere: HCL. Ugyanez nem mondható el a grafikákról, de azért nem is voltak csúnyák (VODKA+HCL). Egy zene kivételével — amit HCL kódér léteére komponált — az összes zene lopott volt, ezért a zenét nem értékelem.

Kód: 90%.

Grafika: 65%.

A demót minden kedves demó gyűjtő figyelmébe ajánlom, mert nem szabad kihagyni a kollekciónkból.

A második demó egy 1992-es alkotás. A neve: MORE THAN NOPS/TABOO. Ez a csapat első igazi demója. Azóta már egy újabb alkotásuk is megjelent.

INTRO

Először is be kell írunk a nevünket, mert a demó regisztrál minden egyes embert aki megnézte.

Azután egy sideborderen mozgó csúnya logó toppan be és mozog jobbra+balra+fel+le. A logó közepén egy kis scroll csúszik be.

1. PART

Mivel a csapat lengyel tagokból áll, ezért Lengyelország térképét láthatjuk pörögni a képernyőn. A vektor-grafika előre le van animálva, szóval se nem a for-

gás, sem a fillezés nem realtime! Alul egy nagy sideborder-scrollt láthatunk.

2. PART

Egy kis lazítás képpen játszhatunk barátunkkal sideborder TRON játékot.

3. PART

Játék után játék következik. Egy kis kétszemélyes jégkorong-játék. A labdát az ellenfél kapujába kell mandínerezni.

4. PART

Egy nagyon szép sprite-zoomer effektet láthatunk az összes kereten rengeteg ötletes effektel és szépen kivitelezve. Nagyon látványos Part.

5. PART

Itt a vége, fuss el véle... szóval a THE END Partot láthatjuk. A képernyő tetején egy AFLI ,the end' logó álldogál. Lent egy upscroll fut felfelé. Megnézhetjük, hogy mennyi ideig néztük a demót, és az elején beadott nevünk felkerül az örök listára. A space megnyomásával egy meglepetés effektet láthatunk.

Kódolta: KM, MMS.

Grafika: alig volt az egész demóban belőle, de ami volt az nem volt túl szép.

Zene: lopott, szóval most sem értékelem.

Értékelés:

A gép egy másodperc alatt a képernyőt 50 alkalommal frissíti fel, vagyis 50-szer fut végig az a bizonyos elektron-sugár a képcsövön, ami az aktuális képet megjeleníti a számunkra. 1 darab ilyen képernyő-újrarajzolást egy megszakításnak, angolul FRAME-nek hívjuk. Egy ilyen megszakítás ugyan a másodpercnél csak a tört része alatt zajlik le, de ezalatt a gép rengeteg gépi kódú utasítást tud végrehajtani. Egy effekt csak akkor lesz az ember szeme előtt folyamatosan mozgó és szép, ha egy másodpercen belül mind az ötvenszer újból rajzolódik ki, de persze minden újrarajzolás között az effekt képe változik. Ha például egy effekt túl bonyolult ahhoz, hogy egy megszakítás alatt végrehajtható utasítások száma elegendő legyen a kirakásához, akkor már csak minden 2, 3, 4 stb. megszakításra tudja a gép kirakni a képernyőre, tehát a normális 50 frissítés helyett 25-ször, 12,5-szer stb. frissül a képernyő, és ezért az effekt akadózóan, nem szépen fog kinézni. Ezért általában egy effekt annál jobb, minél kevesebb frame alatt kerül ki a képernyőre. Ha egy frame alatt végrehajtható utasításoknak a fele is elég a bizonyos effekt kirakásához, akkor azt szokták mondani, hogy az effekt a fele raszteridőt használja csak ki.

Tehát a mennyiségi világrekordok attól függenek, hogy a kódér mennyire

tud a raszteridővel takarékoskodni egy frame-n belül, és ezáltal minél több effektet kirakni a képernyőre.

Egy jó példa erre a DYCP rekordok sorozata. A DYCP a Different Y Character Position rövidítése. Azt jelenti, hogy egy scroll fut a képernyőn jobbról-balra, melynek minden egyes karaktere saját Y mozgással rendelkezik, ezért egy szép szinuszos mozgást ad a scrollnak. A MEGADYCP-ek is ugyanúgy néznek ki, mint a sima DYCP-k, csak a szinuszos magasabb mint 6 karakter, és ezért bizonyos kódtechnikai szempontból nehezebb őket előállítani. Az első MEGADYCP-t valamikor a 80-as évek végén készítették el, és ekkortól kezdődtek a MEGADYCP rekordok megszületései. Először egy ABS/SPHINX nevű kódér készített 2 db MDYCP-t. Nemsokkal utána MADU-PLEC/NATO kiadott egy demót, aminek az egyik Partjában beállította az új rekordot, vagyis 3 db MDYCP-t. Majd VISION/CREST 5 darabot csinált, és még mindig nincsen vége. A CREST másik kódere CROSSBOW egy későbbi CREST demóba csinált 11 (!) darabot. Már ekkor alig lehetett a nagy kavalkádban a scrollokat elolvasni, de ez nem akadályozta meg az INTRUDERS nevű csapat kódérét, hogy készítsen 16 darabot. CROSSBOW/CREST és CRUZER/CAMELOT egy időben adta ki külön-külön demójában az új rekordot, ami 20 darab MDYCP-t tartalmazott. Ekkor már 1992-ben járunk. A tudomásom szerinti utolsó rekordot 1993 márciusa körül adta ki WAX/WOW nevű német kódér, aki 22 darab MDYCP-t állított elő. És még ki tudja, hogy a jövő mit tartogat a számunkra...

Ez a példa is mutatja, hogy néhány év alatt a kódolási technika milyen sokat tud fejlődni, és 2 MDYCP-ből 22-ig is el lehet jutni. De persze most se felejtjük el, hogy ezek a rekordok mind ugyanannyi raszteridőt ettek, és lám elég nagy különbségek jöttek létre. A kódéren múlik minden, hogy mennyire tud a raszteridőn spórolni.

Most következzen néhány demónak az értékelése. Az első paciens:

TOTALY STONED 2/BOZE DESIGN

Ez a csapat második demója, és az 1993-as „THE COMPUTER CROSSROAD” Partyn lett kiadva, ahol nem kevesebbet, mint a második helyet szerezte meg. A leírást Partról Partra fogom elkövetni.

INTRO

A képernyő tetején egy BD-logó váltogatja magát egy FLI-BOOZE DESIGN logóval. Alul egy 3 színű nagy scroll halad.



— Lapozzunk vissza az 1993/10-es számhoz és ott találunk öt rovatot, amit keresztrejtvényünkben rejtettünk el.

VÍZSZINTES: 1. Ritkán található az ember ilyen szórakoztató programmal. 13. Trója másik neve, mely alapítójától Ilosztól ered. 14. Ősi itáliai istenség. 16. Kettős betű. 18. Ritka férfinév. 20. Labor bal oldali mellékveze. 21. Somogyvári Rudolf monogramja. 22. Magyar Optikai Művek rövidítése. 24. Helység Szabolcs-Szatmár megyében. 26. Idegen pénznem. 27. Elán. 29. Van ilyen partner is. 30. Vulnusa. 31. Kálium és aktínium vegyjele. 33. Helység Veszprém megyében. 34. Dohányzik. 35. Ilyen fa is létezik. 37. Erős ital. 39. Hahnium vegyjele. 40. Néhány hónapja jelent meg a piacon egy új maszkalós ügyességi játékprogram. 41. Kétes(!). 43. Testrészt. 44. Kert. 45. Másik kettős betű. 47. Igavonó állat. 48. Helység Borsod-Abaúj-Zemplén megyében. 50. Ausztriai és németországi gépkocsik jelzései. 51. Azonos az 53. függőlegessel. 52. Komárom megyei város. 54. A föld burkának legkülső, főképpen szilíciumból és alumíniumból álló, kb. 100 km vastag rétege. 55. Fordítva: csecsemő. 57. Becézett férfinév. 59. Három különböző mássalhangzó az ABC-ből. 61. Ilyen pörkölt is van. 62. Gallium vegyjele. 63. Dohányzik. 65. Iráni uralkodó. 67. Római 550. 68. Hegység, itt rejtőzött Anzu, amikor elrabolta a sorstáblákat. 70. Patak. 72. Az alábbiakban ismertetésre kerülő program talán a legjobban sikerült játék a maga kategóriájában, a gazdasági szimulátorok között.

FÜGGŐLEGES: 2. Azonos az 51. vízszintessel. 3. Németországi város. 4. Végtelenül Európé anyja volt. 5. Lakoma. 6. Kubai és norvégiai gépkocsik jelzései. 7. Visszaír (!). 8. Idegen férfinév. 9. Védő-

KERESZTREJTVÉNY

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

szentje a szülő anyáknak. 10. Huzat. 11. Betű kimondva. 12. Törökországi kikötőváros. 15. Német város. 17. Gléda. 19. A műszaki tudománynak az az ága, amely elektroncsövekkel, ioncsövekkel,

tranzisztorokkal, félvezetőkkel foglalkozik, valamint ezek alkalmazását tárgyalja. 21. Vulnus. 23. Jeruzsálem elég messze van ide. 24. Olaszországi, svédországi, uruguayi gépkocsik jelzései. 25. Fiumétól délre sziget. 26. Két szó: irányzó, kötőszó. 28. Aszalt gyümölcs. 30. Csongrád megyéből való. 32. Záró. 34. Víz fajta. 35. Argentínai gépkocsijelzés. 36. Szülő. 37. Ferde. 38. Másik kettős betű. 39. Egyik szoftver neve. 42. Apollón papnője. 46. Egyik megyénk. 48. Ipari városunk. 49. Nóta. 51. Testrészt. 53. Titán vegyjele. 56. Csak félig Bükk. 58. ...ka, helység Tolna megyében. 60. Ilyen parancs van a honvédségnél is. 63. Azonos a 34. vízszintessel. 64. Becézett leánynév. 65. Vidéki vendégmarasztaló. 66. Létezik ilyen ábránd is. 68. Étélizesítő. 69. ... — napisten Udar-tutu fia volt. 70. Személyes névmás. 71. Róma közepe.

(A rejtvényben a rövid és a hosszú ékezetek között nem teszünk különbséget.) A rejtvényt nem kell beküldeni, mert szórakoztató jellegű!

Feladó:

.....

.....

ACOMP Kft.
Csomagküldő Szolgálat

B u d a p e s t

XIV. Álmos vezér park 20.

1141

Mokos István



Keresem megvételre használt (korábbi), de még jól működő ACTION REPLAY vagy hasonló menüvel rendelkező cartridge C-64-re, lehetőleg leírással. Cím: Kövesdi Tibor, 3700 Kazincbarcika, Szegfű u. 14.

C-64 számítógép újszerű állapotban eladó. (1541-II floppy, magnó, 2 joystick) Ára: 24 000 Ft. Érd.: Szabó: 12-87-251.

C-64-re 4800 játék, 1600 felhasználói programból válogathatsz. Válaszborítékért tájékoztató. Halász, 6001 Kecskemét, Pf.: 512.

C-64-re 3 Ft-tól eladók programok kazettára és lemezre. Széles választék, megbízhatóság. Válaszborítékért listát küldök. Cím: Kovács Krisztián, 6750 Szeged-Algyó, Bartók Béla u. 6.

Eladó egy működőképes Plus/4-es számítógép + magnó + programkazetták + Commodore Újságok. Ár ajánlatokat a következő címre kérek: Czifra Zoltán, 5231 Fegyvernek, Kiss J. u. 18/a.

NYÁK-tervező elektronikai nyilvántartó leírás programok C-64. Válaszborítékért ismertetőt küldök. Varsányi, Szombathely, Nagy László u. 11.

AMIGA 2000 eladó tv-modulátorral, 2 MB bővítővel, könyvekkel ára 50 000 Ft. Gódor József, 5126 Jászfényszaru, Pf.: 14.

Eladó GEOS C-64-re. Válaszborítékért tájékoztató alkalmazói és játék programokról. Várad, 2151 Fót, Pf.: 43.

C-64-hez olcsón eladó 1 db epromégető, 1 db eprombank és 2 db játékcartridge. Pázsit, 5650 Mezőberény, Várkonyi u. 1.

576-ban megjelent és egyéb C-64-es programok eladása lemezen, kazettán. Lemezes programok kazettás változatai. Olcsó árak. Deutsch Szabolcs, 8749 Zalakaros, Fő u. 19. Tel.: 93-340-481.

C-64-re programok 15 Ft-ért! Válaszborítékért lista! COMPI-CREW, 9730 Kőszeg, Deák F. u. 3/e. 4. ajtó.

C-64-es C+4 kazetták 200 Ft/db, lemezek 60 Ft/db áron! Listát küldök. Csernách Zoltán, 1171 Budapest, Dalnok u. 5.

Helló! Akarsz C-64-re olcsón programokat? Akkor írd! Cím: The best soft, 7400 Kaposvár, Hermann O. u. 60.

C-16 + 4-re 280 játék örökélet poke-jai, indítási címei. 7 oldal: 100 Ft. Soós Viktor, 7355 Nagymányok, Katona u. 11.

C-16 és +4-re programokat adok-veszek-cserélek (5 Ft/db). Soós Viktor Sovisoft, 7355 Nagymányok, Katona u. 11.

Szeretnék 1250 db lemezt C-16 + 4-re? 17 Ft-os bélyegért tájékoztató. Soós Viktor, 7355 Nagymányok, Katona u. 11.

C-16 + 4-re programleírások 10 Ft/gépelő oldal. 17 Ft-os bélyegért, tájékoztató. Soós Viktor, 7355 Nagymányok, Katona u. 11.

GEOS és POKE lexikon. NYÁK tervezőprogram C-64-re alkatrésznyilvántartó programok, C-64-bővítések. Varsányi, Szombathely, Nagy L. u. 11.

Színvonalas C-64-es programok eladása lemezen, kazettán. Lemezes programok kazettás változatai. Olcsó árak! Kérj listát! Deutsch Szabolcs, 8749 Zalakaros, Fő u. 19. tel.: 93-318-481.

C-64-re játékok 15 Ft-ért! Válaszborítékért lista! COMPI-CREW, 3730 Kőszeg, Deák F. u. 3/e.

Eladó C-64 + 1571 floppy + magnó + 1 joy + lemezek + irodalom. 20 000 Ft-ért. Telefonon: 1-639-673 (esti órákban).

C-64 programcsere lemezen Action replay-t és leírást keresek! Cím: Cseh Balázs, 8636 Balatonöszöd, Szabadság út 112.

PÖTYÖGŐ SZOLGÁLAT

1	C64	86/ 0	GOTO X Databecker	40.-
2	C64	86/ 0	GOTO X Data Welt	40.-
3	C64	86/ 0	Hangzó billentyűk	40.-
4	C64	86/ 0	Memória kiírás	40.-
5	C16	86/ 0	Rajzoló program	90.-
6	C64	86/ 1	Atsorszámzó program	40.-
7	C64	86/ 1	Help trace	40.-
8	C16	86/ 1	Hangmemória	110.-
9	C64	86/ 1	Teke home computer	60.-
10	C64	86/ 1	Csak kezdőknek	40.-
11	C64	86/ 1	Billentyűkódok	40.-
12	C64	86/ 1	Autonumber	60.-
13	C16	86/ 1	Basic bővítő	60.-
14	C64	86/ 1	BASIC billentyűzet	60.-
15	C64	86/ 3	Oszlopdiagram 1.	90.-
16	C64	86/ 2	Perifériavizsgálat	40.-
17	C64	86/ 2	Sprite mozgítás IRQ	60.-
18	C64	86/ 2	NOSCROLL	60.-
19	C64	86/ 3	Lemezmenü	40.-
20	C64	86/ 3	Ekezetes nagybetűk I.	40.-
21	C64	86/ 3	Ekezetes nagybetűk II.	40.-
22	C64	86/ 3	Notesz	110.-
23	C16	86/ 2	TURBOBASIC	130.-
24	VC20	86/ 3	Péter és Pál	90.-
25	C64	86/ 4	Mozart	90.-
26	C64	86/ 3	Segítség listázáshoz	60.-
27	C64	86/ 3	MERGE	60.-
28	C16	86/ 3	Billentyűzet kódoló	40.-
29	C16	86/ 3	Kódoló szabadon	40.-
30	C64	86/ 4	AUTO-INSERT	40.-
31	C64/+4	86/ 4	Kalandjáték/Gengszter	110.-
32	VC20	86/ 5	TG MONITOR	130.-
33	C16	86/ 4	Ablakok	60.-
34	C16	86/ 5	Karácsony	60.-
35	C64	86/ 6	Kockás zene	60.-
36	C64	86/ 5	Képűjság	60.-
37	C64	86/ 5	Sprite editor	90.-
38	C16	86/ 6	SAVE LOAD szignál	40.-
39	C64	86/ 6	Rubik kocka síkban	90.-
40	C64/S	86/ 6	Oszlopdiagramm 3.	60.-
41	VC20	86/ 6	Winnwtou I.	60.-
42	VC20	86/ 6	Winnwtou II.	60.-
43	C64	86/ 6	Stringrendező	60.-
44	C64	87/ 1	Lemezátnevező	40.-
45	C64	87/ 1	Ora	60.-
46	C64/S	87/ 1	Hiperkocka	60.-
47	C16	87/ 1	ASSEMBLER	130.-
48	C16	87/ 2	Takarító	60.-
49	C64	87/ 2	Quadro-vizió	40.-
50	C16	87/ 2	Telezsák	130.-
51	C64	87/ 3	Stingkereső	40.-
53	C16	87/ 3	Hangos billentyűk	40.-
54	VC20	87/ 3	Rajzoló	40.-
55	+4	87/ 4	Karaktertervező	110.-
56	C64	87/ 4	Dupla BASIC	40.-
57	VC20	87/ 4	Ferde torony	90.-
58	C64	87/ 5	GEOMASTER/lefordított/	130.-
59	C64	87/ 5	GEOMASTER/BASIC változat/	110.-
60	C64	87/ 5	Golyózó	110.-
61	+4	87/ 5	Magyar BASIC	110.-
62	+4	87/ 5	Kukkoló-BASIC monitor	60.-
63	+4	87/ 6	Sprite editor +	110.-
64	C64	87/ 6	Magnó directory	130.-
65	C16	87/ 6	Nagyfelbontású hardcopy	110.-
66	C16	87/ 6	Sztereo vizió	40.-
67	C16	87/ 6	Kukkoló-Szuper peek	90.-
68	C16	87/ 7	Mentőv	40.-
69	+4	87/ 7	Ablakozó	110.-
70	C64	87/ 8	Commo-do-re/BASIC vált.	130.-
71	C64	87/ 8	COMMO-do-re/lefordított/	130.-
72	+4	87/ 8	Kukkoló-bővített monitor	60.-
73	+4	87/ 8	Ekezetes betűk	60.-
74	+4	87/ 8	PLUTO-rajzoló program	130.-
75	VC20	87/ 8	Majomfogó	90.-
76	C64	87/ 8	Többszínű döntések	90.-
77	C64	87/ 8	Raszter	60.-
78	C16	87/10	Append	90.-
79	C64	87/10	Nagyító	90.-
80	C64	87/10	Makrók	90.-
81	C128	87/11	Stuktúra	60.-
82	C128	87/11	Emlékeztető	60.-
83	C64	87/11	Monitorka	90.-
84	C16	87/11	Címről-címre	60.-
85	C64	87/12	Gyors scroll	90.-
86	C64	87/12	1541	90.-
87	C64	87/11	1541 + Monitorka	130.-
88	C16	87/12	Lokális változók	90.-
89	C16	87/12	Tekergő	90.-
90	C64	88/ 1	Fekete-fehér	40.-
91	C16	88/ 1	Datásító	60.-
92	C64	88/ 1	Programkönyvtár	90.-
93	C16	88/ 1	Katasztrófa	60.-
94	VC20	88/ 2	TG BASIC	130.-
95	C64	88/ 2	Topy typer	90.-
96	C64	88/ 2	sprite editor super	130.-
97	C16	88/ 2	katasztrófa	60.-
98	C64	88/ 4	Modulgrafika	130.-
99	C64	88/ 4	Segítség!	60.-





PÖTYÖGŐ SZOLGÁLAT

100	C64	88/ 5	Gurítsd a golyót!	130.-
101	+4	88/ 5	Egyenesrajzoló	60.-
102	+4	88/ 6	Szótár	130.-
103	C16	88/ 7	Gitáriskola	130.-
104	C64	88/ 7	Gyors BASIC	60.-
105	C128	88/ 7	Super hardcopy	90.-
106	C128	88/ 7	Grafikus nyomtatás	90.-
107	C64	88/ 9	Fast validate	130.-
109	C64	88/ 9	data bővítő	60.-
110	C64	88/ 9	DEF FN/X/	60.-
111	+4	88/10	Magyarosító	110.-
112	C64	88/10	Diszkkézelő	90.-
113	C16	88/10	DISASSEMBLER	60.-
114	C64	88/11	Négy képernyő	40.-
115	C64	88/11	Ablakok	40.-
116	C64	88/11	ASCII képernyő	40.-
117	C64	88/11	Pillanatfelvétel	40.-
118	C64	88/11	Blinky	40.-
119	C64	88/11	Színes villogtató	40.-
120	C64	88/11	80 karakteres üzem	40.-
121	C64	88/11	Villogtató	40.-
122	C64	88/11	Joystick lekérdező	40.-
123	C64	88/11	Gyors sprite mozgatás	40.-
124	C64	88/11	DEF FN - másképpen	40.-
125	C64	88/12	Szuper input	90.-
126	C64	88/12	Ugráló színsávok	60.-
127	C64	89/ 1	PRINT AT	40.-
128	C16	89/ 1	Funkcióbillentyűk	40.-
129	C64	88/12	Minihelp	40.-
130	C64	89/ 2	Házi pók	90.-
131	C64	89/ 2	Csillagos ég	60.-
132	C64	89/ 1	Szalagról lemezre	90.-
133	C64	89/ 1	Kniffel	110.-
134	+4	89/ 1	Zombi	130.-
135	C64	89/ 1	Kétoldalas listázó	130.-
136	+4	89/ 1	Objektív osztályozás	110.-
137	+4	89/ 2	Tömörítő	40.-
138	+4	89/ 2	Kifejtő	40.-
139	+4	89/ 3	3-plusz-1 magyarul	90.-
140	+4	89/ 3	Napló-kezelő	60.-
141	+4	89/ 3	Analóg óra	40.-
142	C128	89/ 3	Javitott renumber	90.-
143	+4	91/ 6	Beszélő C+4	40.-
144	C64	89/ 4	Keret csikozás	40.-
145	C64	89/ 4	Mozgó felirat	40.-
146	C64	89/ 4	26. sor	90.-
147	C16	89/ 4	Optinput	60.-
148	C16	89/ 4	Turbo-block	60.-
149	C16	89/ 4	Finom scroll	60.-
150	+4	89/ 4	Animáció	130.-
151	C128	89/ 4	Pót funkciók	40.-
152	PLUS/4	89/ 5	Betűradar	90.-
153	C64	89/ 5	BHP vírus kereső	90.-
154	C64	89/ 5	Képmutató	130.-
155	C128	89/ 5	VDC mint RAM floppy	60.-
156	C128	89/ 5	Joy Sprite	60.-
157	C64	89/ 5	Javitott list	60.-
158	C64	89/ 5	Zenei segédlet	130.-
159	C16	89/ 6	Szinbeállító	60.-
160	C16	89/ 6	PLUS/1 listázó	90.-
161	+4	89/ 6	Fast validate	130.-
162	C64	89/ 6	Balnyilcopy	90.-
163	C64	89/ 6	3-D labirintus	130.-
164	C64	89/ 6	Mini vírus	90.-
165	C128	89/ 6	VDC-window	60.-
166	C64	89/ 6	Cimkeíró	60.-
167	C64	89/ 6	Piros-fehér-zöld képernyő	40.-
168	C64	89/ 7	GEN-GET-INPUT	90.-
169	+4	89/ 7	Szinbeállítás	60.-
170	+4	89/ 7	Nagyító /079 átírás/	110.-
171	+4	89/ 7	80 karakteres szövegszerk	180.-
172	C16	89/ 7	Ékezetes betűkészlet	130.-
173	C16	89/ 7	Kisebb ékezetes nagybetűk	90.-
174	C64	89/ 7	Gyors directory	90.-
175	C64	89/ 7	AUTOSTART	90.-
176	C64	89/ 7	MULTI 16	90.-
177	C64	89/ 7	MULTI 16 DEMO	90.-
178	C64	89/ 7	Egér 64	90.-
179	C16	89/ 7	Flash-Load	90.-
180	C16	89/ 7	BASIC-LISTA-SCROLL	90.-
181	+4	89/ 7	2x2 karakterek	40.-
182	+4	89/ 7	2x2-es karakterek	40.-
183	C16	89/ 7	Kereső	40.-
184	C16	89/ 7	Borítékcimező	110.-
185	C64	89/ 7	DEC-HEX-BIN	130.-
186	C64	89/ 7	Object save	60.-
187	C64	89/ 9	Sprite	90.-
188	C64	89/ 9	Bioritmus	60.-
189	C16	89/ 9	Turbo Trans	150.-
190	C64	89/10	Névjegy	60.-
191	C64	89/10	Kórokozó	90.-
192	C64	89/10	SCAN	90.-
193	+4	89/10	DISC CRACK	130.-
194	C64	89/11	Függvényábrázolás	180.-
195	C64-16	89/11	Uj Optinput	90.-
196	+4	89/11	Számkirakó	130.-
197	C64	89/11	Céllövölde	60.-
198	C64	89/12	Esperanto	110.-

199	+4	89/12	Lépegető	90.-
200	+4	90/ 1	Raszter bűvölet	40.-
201	+4	90/ 1	TOOL-7.0	110.-
202	C64	90/ 1	Ora a kereten	90.-
203	C64	90/ 1	Mutatópálca	60.-
204	C64	90/ 1	Karaktertervező	90.-
205	C64/S	90/ 2	Csillagászat	130.-
206	C64/S	90/ 2	Térképismeret	60.-
207	C64	90/ 2	Smooth scroll	60.-
208	C64	90/ 2	File tester	60.-
209	C64	90/ 2	Line Tracer	60.-
210	C64	90/ 2	Supervisor	60.-
211	+4	90/ 2	Véletlenszerű képcsere	90.-
212	+4	90/ 3	Karakter leíró	40.-
213	C16	90/ 4	Zsolnai átlagszámító	60.-
214	C64	90/ 4	Labdák	60.-
215	C64	90/ 4	Funkcióbillentyűk	90.-
216	C64	90/ 4	Futó felirat	90.-
217	C64	90/ 4	Jegyzetömb	40.-
218	C64	90/ 5	Videofeliratozó	60.-
219	C64	90/ 5	DUMP	60.-
220	+4	90/ 5	Bűvös négyzet	90.-
221	C64	90/ 5	Szuper BONGO	60.-
222	C64	90/ 6	Videokatalógus	110.-
223	C64	90/ 5	Datakészítő	90.-
224	C128	90/ 6	Cartridge szimulátor	130.-
225	C128	90/ 6	Double charset	60.-
226	+4	90/ 7	Hatos lotto	60.-
227	+4	90/ 7	BIT-MAP	60.-
228	C64	90/ 7	GEOS BOOT MAKER	60.-
229	C64	90/ 7	Screen editor	90.-
230	C64	90/ 7	Szekvenciális filelistázó	40.-
231	C64	90/ 7	Editorbővítő	130.-
232	C64	90/ 7	Kockapóker	130.-
233	C64	90/ 7	Lemez bűvész	90.-
234	C64	90/ 7	szuper autostart	60.-
235	C64	90/ 7	Sipoló billentyűk	40.-
236	+4	90/ 9	Directory print	90.-
237	+4	90/ 9	Orarend	130.-
238	+4	90/10	Nyelvi gyakorló	110.-
239	C64	90/10	Eprom szerkesztő	60.-
240	C64	90/10	LED indikátor	60.-
241	+4	90/10	öröknaptár	110.-
242	C64	90/11	Funkcióbillentyűk	60.-
243	C64	90/11	Formaprint	60.-
244	C64	90/11	Csikos képernyő	40.-
245	C64	90/11	Életjáték	90.-
246	C64	90/11	Videózó	130.-
247	C64	90/11	SEJT 64	180.-
248	C64	90/11	Betűkirakó	130.-
249	C128	90/11	Input rutin	110.-
250	C64	90/11	Dupla karakterek	180.-
251	C64	90/12	SPRITE PEEPER	90.-
252	C64	90/12	SPRITE	60.-
253	+4	90/12	Malom	180.-
254	C64	91/ 1	Vízszintes scroll	60.-
255	C64	91/ 1	BASIC betöltő készítő	60.-
256	C64	91/ 1	VERIFY MASTER	90.-
257	C64	91/ 1	Hibamentes replace	130.-
258	+4	91/ 1	Auto Input	130.-
259	C64	91/ 1	Törtek	60.-
260	C64	91/ 1	Ablakozás könnyen	90.-
261	+4	91/ 2	80-as képernyő	80.-
262	C64	91/ 2	Fizikateszt	50.-
263	C64	91/ 2	40 Sprite	130.-
264	C64	91/ 2	Raster - Master	80.-
265	C64	91/ 3	Rullett	100.-
266	+4	91/ 3	Tape-disc copy	110.-
267	C64	91/ 3	BASIC rearranger	50.-
268	C64	91/ 3	BASIC decompactor	50.-
269	C64	91/ 3	BASIC merger	50.-
270	C64	91/ 3	SEIKOSHA	80.-
271	C64	91/ 3	Cartridge szimuláció	130.-
272	+4	91/ 3	Sally kulcsszókereső	50.-
273	+4	91/ 3	Merge C+4	80.-
274	C128	91/ 4	1.5 MHz-es C64	100.-
275	C64	91/ 4	Morze - H	80.-
276	C64	91/ 4	Gprint	50.-
277	C64	91/ 4	ASCII - CHR\$	50.-
278	C64	91/ 5	Rendező	80.-
279	C64	91/ 5	Keretbeíró	60.-
280	C64 +4	91/ 5	NLQ Print	60.-
281	C64	91/ 5	FLD	50.-
282	C64	91/ 5	Raster Split	50.-
283	+4	91/ 6	Programnyilvántartó	80.-
284	C64	91/ 6	Regiszter	60.-
285	C64	91/ 6	Digitalizált zene	50.-
286	C64	91/ 6	Ékezetes MPS 802	80.-
287	C64	91/ 7	*Ékezetes iratkészítő	100.-
288	+4	91/ 7	Bűvös téglalap	100.-
289	C64	91/ 7	Dominó - H	100.-
290	C64	91/ 7	Raktáros	80.-
291	+4	91/ 7	Tologató	60.-
292	C64	91/ 7	Aren 2000	50.-
293	C64	91/ 7	Bad Lamps	80.-
294	C64	91/ 7	Operation Patriot	50.-
295	+4	91/ 7	Mozaik - B	90.-
296	+4	91/ 7	Logi Kód	60.-
297	C64	92/ 1	Databasevívó	50.-
298	+4	91/ 7	Bomber	50.-
299	+4	91/ 7	Dominó - K	50.-
300	+4	91/ 7	The Wall	40.-
301	C64	91/ 7	Mozaik - K	50.-

ACOMP Kft.

FEBRUÁRI

60 Ft-os

vásárlási utalványa

Beváltható 500 Ft feletti készpénzes vásárlás esetén a 1135 Budapest, Szent László u. 74/A. és 1191 Budapest, Katica u. 9. szám alatti üzletekben.

Érvényes: 1994. február 28-ig.

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.

1083 Budapest, Szigony u. 8.	Tel.: 1343-153
3525 Miskolc, Fazekas u. 1-3.	Tel.: 46/321-488
5600 Békéscsaba, Bartók B. u. 37.	Tel.: 66/327-195
6724 Szeged, Csongrádi sugárút 76.	Tel.: 62/493-185
9700 Szombathely, Szalonak u. 31.	Tel.: 94/314-519

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

NOVOTRADE
SZERVÍZ Kft.

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

Az Országos Commodore Egyesület szolgáltatásai

Egyesületi tagoknak 20% kedvezmény:

	kiépitéstől függő
VC—20 memóriabővítés 3—27 kByte-os:	3500 Ft
C—16, C—116 memóriájának bővítése 64 kByte-ra:	1450 Ft
C—16 belső 16 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2800 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	2000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	3200 Ft
C—16 és 1541 kompatibilis lemezzegység párhuzamosítása:	5000 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	2000 Ft
TTL IC-teszter (Cartridge+lemezen a program)	4300 Ft
+4, C—16, C—116 UNI—ROM modul különféle kiépítésekben:	
— 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, lemezzegység átalakítás + párhuzamos kábel)	5000 Ft
1541 kompatibilis lemezzegységbe 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
2x64 Kbyte-os cartridge igény szerinti programokkal feltöltve	4.300 Ft
64 Kbyte-os cartridge igény szerinti programokkal feltöltve	3000 Ft
Képujság (teletext) dekóder C-64-re	10000 Ft
1764-es RAM bővítő GEOS-hoz is használható	11250 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az eprommal együtt egységesen	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.

FEBRUÁRI 60 Ft-os vásárlási utalvány

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

Érvényes: 1994. február 28-ig.

HOBBI ELEKTRONIKA

FEBRUÁRI 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!



SyQuest
TECHNOLOGY

Hivatalos magyarországi disztribútora:

NOVOTRADE

SZERVIZ Kft.

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

Cserélhető lemezes winchester!

Szervizeinkben és szerződött viszonteladóinkon keresztül az alábbiakat forgalmazzuk:

SQ 555	44 MB-os 5 $\frac{1}{4}$	SCSI drive,	SQ 400 lemez	(44 MB)
SQ 5110	88 MB-os 5 $\frac{1}{4}$	SCSI drive,	SQ 800 lemez	(88 MB)
SQ 3105A	105 MB-os 3 $\frac{1}{2}$	AT-BUS drive,	SQ 310 lemez	(105 MB)
SQ 2542A	42 MB-os 2 $\frac{1}{2}$	AT-BUS drive,	SQ 240 lemez	(42 MB)

Számítógépszerviz !

Személyi számítógépek és perifériák javítását, karbantartását továbbra is tőlünk rendelje meg! Gyors, pontos és megbízható! Örök garanciát kap, ha általánydíjas javítási és karbantartási szerződésformánkkal védi nagyértékű gépei műszaki állapotát.

Várjuk jelentkezését az alábbi szervizeinkben:

1083 Budapest,	Szigony u. 8.	Tel.: 1343-153
3525 Miskolc,	Fazekas u. 1-3.	Tel.: 46/321-488
5600 Békéscsaba,	Bartók B. u. 37.	Tel.: 66/327-195
6724 Szeged,	Csongrádi sgt. 76.	Tel.: 62/493-185
9700 Szombathely,	Szalónak u. 31.	Tel.: 94/314-519

NOVOTRADE SZERVIZ KFT.



Ára: 88 Ft

ACOMP

SZÁMÍTÁSTECHNIKAI KFT.

1125 Budapest, Királyhágó utca 2.

Tel.: 156-6790

Fax: 251-2385

1135 Budapest, Szent László út. 74./A.

Tel.: 149-6165

Fax: 251-2385

1191 Budapest, Katica utca 9.

Tel.: 147-0625

Fax: 177-9419

