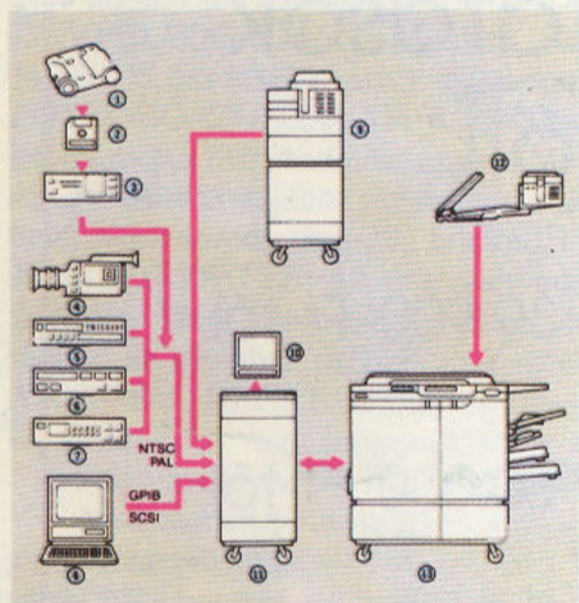


## GÉPKER Kft. — Canon szervíz

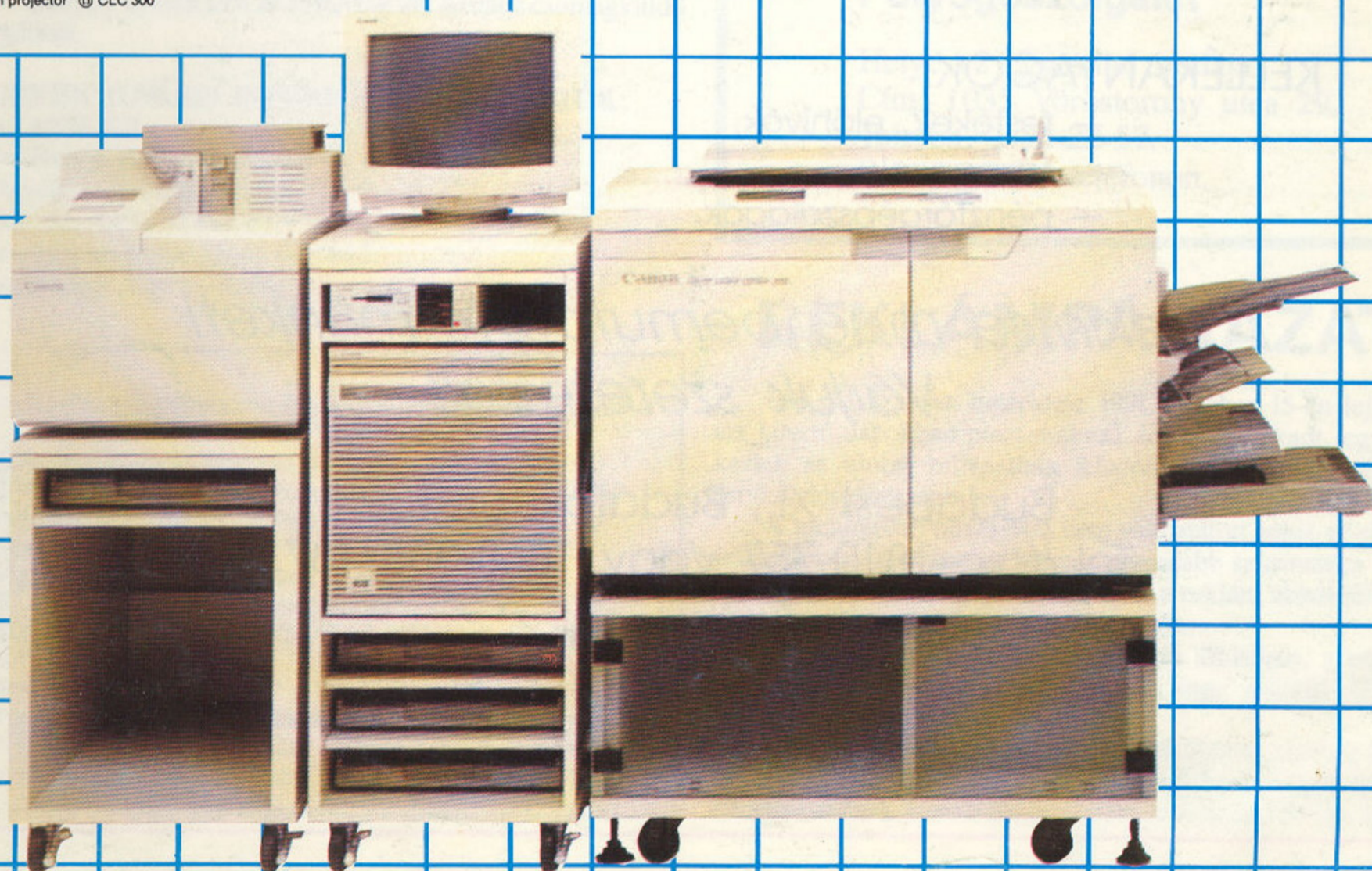
1139 Budapest, Frangepán u. 7.

### CLC 300 digitális lézersugaras másológép

Az eredeti és a másolatok nagysága A/3-ig terjedhet.  
Percenként 5 teljesen színes másolatot készít. Egyszínűből percenként 10-et.  
A/4-es fekete-fehérből 20-at percenként.  
Zoom tartomány 50-400%.  
Felbontás 400 dpi (pont/hüvelyk) = 16 pont/mm<sup>2</sup>  
256 színfokozat színenként, összesen kb. 17 millió színárnyalat.  
Automatikus eredeti felmérés.



- ① still video camera ② video floppy ③ still video player
- ④ video camera ⑤ VCR ⑥ video disk player ⑦ TV tuner
- ⑧ computer ⑨ film scanner ⑩ monitor ⑪ IPU
- ⑫ film projector ⑬ CLC 300





**SZÁMALK-CED**  
Kereskedelmi Kft.

## Magas szintű irodatechnikai szolgáltatások a SZÁMALK-CED-től!

UTAX — Fénymásolók  
és teljes tartozékok (12-féle géptípus) 39 900 Ft-tól

NIKKAM pénztárgépek  
(APEH-engedélyes — visszaigényelhető) 50 000 Ft-tól

SAFAX telefaxok (francia) 89 000 Ft-tól

### KELLÉKANYAGOK:

- festékek, előhívók,
- faxpapírok,
- pénztárgépszalagok.

*Tekintse meg bemutatótermünket!  
Várjuk szeretettel!*

Budapest XI., Budafoki út 109.  
Tel./fax: 1810-757 vagy 1868-333/137.

**CÉL A CED!**

## MIT, HOGYAN, HOL, MIKOR?

**EGYESÜLETI ÜGYEK:** Egyesületünknek tagja lehet mindenki, aki a tagsági díjat befizeti. A tagdíjat személyesen az egyesület irodájában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát 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 egy évre 777 forint.

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

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

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

**PÖTYÖGŐSZOLGÁLAT:** Az újságban megjelenő programokat másolja a megrendelők részére. Megrendelhető személyesen az egyesület irodájában vagy postai utánvétellel. Cím: 1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57.

**APRÓHIRDETÉS:** Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 80 forint. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

A **C-ÚJSÁG RÉGEBBI SZÁMAI** megvásárolhatók az egyesület irodájában, vagy megrendelhetők utánvétellel.

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk a 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  
91.1238 FTV-MSZH Nyomda és Kiadó Kft., Budapest  
Felelős vezető: Nagy László

## Kedves Tagtársaink, Olvasóink!

### Egyesületi és szerkesztőségi CÍMÜNK MEGVÁLTOZIK.

1992. február 1-jétől új címünk:

**1025 Vöröstorony utca 29.**

**Telefon: 1-76-22-57.**

Megközelíthető a Batthyány térről a 11-es autóbusszal. A Kapy utcai megállónál kell leszállni. Elképzelhető, hogy a költözés miatt kicsit akadozni fog a kapcsolattartás Önökkel. Ezért megértésüket és türelmüket kérjük. Lapunk megjelenésében fennakadás nem lesz.

**Postacímünk változatlan:**

**1388 Budapest 62., Postafiók: 86.**

OCE

## Tagdíjbefizetés

Köszönjük, hogy eleget téve év végi felhívásunknak, befizették az 1992. évi tagdíjat. Kérjük, hogy aki valamilyen okból esetleg még nem fizette ki választott tagdíját, mielőbb tegye meg.

OCE

## Pötyögőszolgálat

Helye az egyesület irodája.

Cím: 1035 Vöröstorony utca 29.

Telefon: 1-76-22-57.

Felvilágosítás telefonon.

## REKLÁMPÁLYÁZAT

Bár pályázatunk határideje 1991. október 15-én lejárt, az azt követő napokban postásunknak sok dolga akadt, egyre érkeztek az utolsó pillanatban feladott pályaművek. Végül 61 produkciót értékeltünk.

A mennyiségi növekedés nem változtatott sokat az előzetes értékelésben leírtakon — bizony, magasabb színvonalra számítottunk. Akadt néhány pályamű, ahol a reklám abból áll, hogy a reklámszöveg megjelenik a képernyőn.

Sok programban láthattunk egy-két állóképet, a kép alatt megjelenő mozgó szöveggel, zenével vagy anélkül. (Például Mikó Imre: Alfa Áruház.)

Valamivel jobb hatásúak azok a reklámok, melyek állóképekkel vagy anélkül működnek, de a reklámszövegek szép karakterkészlettel, jó mozgással, színes villogással hívják fel a fi

gyelmet. Ilyen Szele József filmsláger reklámja. Ebben a kategóriában a gépi kódú programok mellett akadnak BASIC programok is. Erdei Péter Top Gun Magazin reklámja a Graphics Basic lehetőségeit kihasználva mozgatja ízléses karaktereit. Ebből a típusból is érkeztek zenélő és csendes művek.

Hogy ebben a műfajban is lehet tartalmasat készíteni, mutatja a 14 éves Varga Renáta Tungram Disk reklámja, mely egyszerű eszközökkel mozgófilmszerűen meséli el történetét.

A gépi kódú programok nagy része felrajzol egy háttérképet, elindít egy népszerű zenét, és a háttér előtt mozgat egyes alakzatokat. Ilyen Balda János Fogyi, vagy Ignác István Kontrax reklámja. Az utóbbinak erénye, hogy egyszeri futás után leáll, de reset-re újra indul.

A legjobb művek a előzőek erényeit egyesítik. Kár, hogy igazi frappáns reklámtémával nem találkoztunk. Talán mond valamit, hogy a legmeghökkenőbb Lukucza György ötlete volt, aki a Temetkezési Intézetnek csinált egy kis propagandát...

Kérjük jövőd pályázataink résztvevőit és mindazokat, akik szerkesztőségünkbe programot küldenek be, jelöljék meg, milyen gépen dolgoztak, mert sok felesleges munkát okoz a C64-es és PLUS/4-es programok szétválogatása.

Az alábbiakban közöljük mindazok nevét, aki egyéves előfizetést kaptak jutalmul. Szakértő zsűrink úgy döntött, hogy a fődíjat nem adjuk ki.

OCE

Poscher Norbert	Mór
Szombathelyi Attila	Győr
Kun Szilárd	Budapest
Gyuricza Ferenc	Érd II.
Budavári Tamás	Kartal
Kesik ifj. Lajos	Budapest
Molnár Zsolt	Kecskemét
Antal István	Budapest
Loós József	Debrecen
Bertalan Róbert	Budapest
Bojtos Péter	Jászberény
Szemánszki István	Kazincbarcika
Kovács Tibor	Ajak
Farkas Péter	Kecskemét
Badenszky Csaba	Tatabánya
Rybka Roland	Budapest
Halmos János	Budapest

Murányi József	Mezőkövesd
Szücs Gábor	Miskolc
Schaeffer Krisztián	Kecskéd
Pölöskei Csaba	Budapest
Szalai Tamás	Budapest
Nádor Csaba	Budapest
Pálinkó Kristóf	Budapest
Tápai Béla	Zalaegerszeg
Szakálli Gábor R.	Budapest
Valkó Dániel	Budapest
Sasvári Gábor	Pécs
Szöllösi Lóránd	Hajdúszoboszló
Kun Gábor	Harsány
Ludánszky Attila	Hajdúböszörmény
Pankaczy Csaba	Eger
Mányoki ifj. László	Kaposvár
Könözy László	Miskolc
Burghard Tamás	Székesfehérvár
Zsohár János	Kaposvár
Szakács Tibor	Budapest
Mátis Miklós	Öcsény
Dóra Csilla	Zalaegerszeg
Szlovák Miklós	Budapest
Román Gellért	Budapest
Gere Miklós	Putnok
Zsila Zsolt	Budapest
Csáki Béla	Szekszárd
Marhefka István	Miskolc
Ignác József	Békéscsaba
Varga Renáta	Gyöngyöspata
Mikó Imre	Réde
Simkó Zsolt	Budapest
Balogh Ignác	Szentes
Lucz Géza	Kaposvár
Tóth Balázs	Nagykanizsa
Lengyel Béla	Kecskemét
Balda János	Makó
Ignác István	Tornyosnémeti
Szatmári Elemér	Kétpó
Dénes Zoltán	Budapest
Szele József	Nagykanizsa
Markó József	Tibolddaróc
Erdei Péter	Tibolddaróc

## Tisztelt Elnökség!

A mellékelt lemezen két játékprogramot találnak, amelyekkel eredetileg kívántam benevezni az OCE pályázatra. Mivel azonban foglalkozásom (és talán tudásom) alapján a profik közé tartozom, úgy éreztem, sportszerűtlenség lenne a győzelemre törni amatőrök között. Tudom, ez a kiírásban nem szerepelt, de ez lehetett csupán feledékenység is.

Most, hogy a határidő lejárt, automatikusan kivontam magam a versengésből. Van azonban valami, ami bizonyára nem sérti a pályázat feltételeit. Szeretném, ha programjaimat szemügyre vennék, és megbírálnák olyan értelemben, hogy felkerülhetnek-e a Pötyögő Szolgálat révén forgalmazandó többi játék közé. Tekinthezik ezt akár közönséges becsvágynak is.

Köszönettel:

Hódi Gyula

## PATIENCE QUARTET

337

Napjainkban úgy tűnik, kezd visszaszorulni a „lövöldözős” és „máskálós”, sokszor bizony kissé egyhangú játékok legendás uralma, és egyre gyakrabban találkozhatunk újszerű, a játékos szellemi ügyességét próbára tevő játékokkal. Ebbe a családba tartozik egy már nagyanyaink által is művelt játék, a pasziánsz.

Sokan vannak, akik még mindig összetévesztik a kártyajóslást (kártyavetést) a pasziánszal. Míg az előbbi egy babonás szokás, amely a megkevert kártyacsomag lapjainak sorrendjéből próbál jövőbeli eseményekre következtetni, addig ez utóbbi egy szigorúan meghatározott szabályok szerint játszott, a sorstól teljesen független, egyszemélyes kártyajáték.

A lényege minden pasziánsznak ugyanaz: egy megkevert csomag kártyát kell egy meghatározott alakzatból egy másikba átcsoportosítanunk. Eközben csak korlátozott számú lapot moz-

gathatunk egyszerre, vagy a mozgásra szolgáló tartalékterület mérete van szűkre szabva, vagy más hasonló szabály köti meg a kezünket; mindenesetre a megoldáshoz szükség van egy kis taktikára, jó megfigyelőképességre, jó emlékezetre, és egy csipet szerencsére.

A Patience Quartet négy különböző nehézségi fokú pasziánszjátékot kínál egyszerre. Ezek a billentyűzetről kényelmesen kezelhető módon, a gép ellenőrzésével és segítségével játszhatók. A program különlegessége, hogy amíg a kártyával játszott pasziánsz vagy sikerül, vagy nem, addig a program egy pontszámmal árnyaltabban értékeli a teljesítményt. A ponttáblázat természetesen lemezre vagy kazettára menthető, és így részesülhetünk a rekorddöntés élményéből, vagy éppen hitelesen összehasonlítható barátunk (vagy ellenségünk) ponttáblájával is.

A programban részletes leírás található a játékok szabályairól és kezelésükről.

## MONOPOLY

338

Aki játszott már akár a Gazdálkodj okosan!, akár a Capitaly nevű táblás társasjátékkal, az már megismerkedhetett egy olyan játékkal, amelyben a szerencse és a jó taktikázás egyaránt fontos, ügyelnünk kell ellenfeleink manővereire, és kipróbálhatjuk ügyességünket a pénz szerencsés befektetésének művészetében. Mindkét játék őse a Monopoly, amelyet főleg az Egyesült Államokban régóta kedvelnek.

Olyan programot készíteni, amely méltó ellenfelünk lehet egy ilyen vetélkedésben, még sajnos nem tudunk. Írható azonban olyan program még egy számítógépre is, amely leveszi a társaság válláról az aprólékos könyvelés fáradságát, minden lényeges információt bármikor hozzáférhetővé tesz, szigorúan ügyel a szabályok betartására, és kényelmesebbé, gördülékenyebbé teszi a játékot; már csak társakat kell hozzá keresnünk.

Ez a Monopoly program négy személy számára teszi lehetővé a játékot. Készítője alaposan kiegészítette a játék eredeti szabályait, és számos esetben érdekesebb, esetleg a valós élethez kissé közelebb álló megoldást kínál. Néhány példa:

Az árverésen csak azok vehetnek részt, akiknek elegendő készpénzük van.

Az árverés és az eladás során a minimális értéket a gép határozza meg, a manipulációk visszaszorítására.

Aki börtönben van, az nem vehet részt árverésen, és a bevételeit a Bank kapja.

A börtönből való szabaduláshoz kifizetendő pénz összegét a játékos anyagi helyzete és visszaeső volta is befolyásolja.

A börtön nem menedékhely, a bentmaradásért is fizetni kell.

A Free Parking Jackpot (főnyeremény) értéke a Bank bevételei szerint növekszik, akár valóban „főnyeremény” méretűre is.

A Bank minden pénzáttalásból 5%(!) forgalmi adót levon, és ez a FP Jackporthoz adódik.

A saját ingatlanra való lépés újabb dobásra jogosít fel.

Az elzálogosított ingatlan visszaváltása csak 10 körig lehetséges, és csak egyre drágábban.

A jövedelem- és luxusadó értékét a gép mindig a pillanatnyi helyzet szerint számítja ki, a nem adóköteles javakat nem veszi figyelembe.

A Szerencsekártyákhoz tartozó paramétereket (fizetendő vagy kapott összeg stb.) a gép véletlenszerűen, egy meghatározott tartományból választja.

A játék befejeztével a gép részletes értékelést készít a játékosok vagyoni helyzetéről, és ez alapján állapítja meg a végső sorrendet.

Ezek a kiragadott ötletek is érzékeltethetik, hogy a játék kellemes, izgalmas és érdekes szórakozást nyújthat egy kis társaságnak. Bár a program a játékosokhoz angol nyelven szól, tartozik hozzá egy magyar nyelvű leírás is, amelynek segítségével szótár nélkül sem tévedhetünk el. Bár egy kis szótárral akár még hasznos kifejezéseket is elsajátíthatunk játék közben.

## Tisztelt szerkesztőség!

Az alábbi program az ismert „akasztófa játék” számítógépes változata. A számítógép kitalál egy szót, amit a játékosnak betűnként kell megfejtene. (Ezeket a szavakat természetesen

cserélni lehet.) Ha olyan betűre kérdez rá a játékos, amely nem szerepel az adott szóban, a számítógép biztosan „felakasztja”. Az F1 gomb leütésével egy betű segítséget kérhetünk.

Tisztelettel:  
Magyar Gábor

```

100 Poke 788,52:Poke 792,193:Poke 808,234
110 rem *****
120 rem *
130 rem * akasztofa *
140 rem *
150 rem * magyar gabor *
160 rem *
170 rem *****
180 dim a$(111):dim b$(15):Print chr$(142)
190 p=1024:sz=55296:v=53248
200 el=0:t=0:t1=0:s=0
210 Poke v+21,0:Poke v+16,0
220 Poke53280,14:Poke53281,14
230 Print"*****"
240 Print" a. jatekban kitalalando koznevek!"

```

339



```

C 250 Print"      nem tartalmaznak ekezeses betuket"
260 for i=0 to 1000:next:if a#<>" then 290
O 270 for i=1 to 111:read a$(i):next i
280 for i=832 to 894:read q:poke i,q:next i
O 290 i=int(110*rnd(1))+1:a#=a$(i):l=len(a#)
O 300 Print"##### szo";l;"betubol all"
310 for i=1 to l
O 320 poke sz+123+i,0:poke p+123+i,42:next i
330 Print"#####"
O 340 Print"U-----I"
350 Print"Belhasznalt betuk:B"
360 for i=1 to 10
C 370 Print"B          B":next i
380 Print"J-----K"
O 390 Print"/f1:segitseg/";
400 gosub 860:gosub980:gosub1560
O 410 for i=1 to l
420 b$(i)=mid$(a#,i,1):next i
430 get c#:Print chr$(142);:if c#=""then430
C 440 if c#=chr$(133) then gosub 790
450 b=(asc(c#))-64
O 460 if b<1 or b>26 then 430
470 for i=1 to l
O 480 if b$(i)=c# then pokep+123+i,b:t=1
490 next i
O 500 if t=1 then 530
510 t1=t1+1:if t1=7 then 670
C 520 on t1 gosub 1080,1140,1220,1310,1400,1480
530 el=el+2
O 540 if el/16=int(el/16) then el=el+66
550 pokep+560+el,b
C 560 for i=1 to l
570 if peek(p+123+i)>42 then next i:goto 590
580 t=0:goto 430
C 590 Print"#####":Printtab(29)"      ":Printtab(29)"#      "
600 gosub1080:gosub1140:gosub1220:gosub1310:gosub1400:gosub1480
C 610 poke v+29,0:poke v+23,0
620 poke v+16,1:poke v,1:poke v+1,215
C 630 Print"#####":Printtab(25)"#####voztel!!!#"
640 Print"#####";100-int(s/1*100);"%-os teljesitmeny"
C 650 get c#:if c#=""then 650
660 goto 200
O 670 gosub 1030
680 Print"#####":Printtab(24)"#####vesztettel!!!#"
O 690 Print"#####a szo":Print"#####";a#
700 Print"#####Molt"
O 710 poke v+29,0:pokev+23,0
720 x=int(81*rnd(1))+230
730 y=int(141*rnd(1))+60
O 740 if x>255 then poke v+16,1:x=x-255:pokev,x:pokev+1,y:goto 760
750 pokev+16,0:pokev,x:pokev+1,y
O 760 for i=0 to 200:next i
770 get c#:if c#<>" then 200
780 goto 720
C 790 if s=1 then 450
800 for i=1 to l
O 810 if peek(p+123+i)=42 then pokep+123+i,asc(b$(i))-64:s=s+1:goto 830
820 next i
830 for i=1 to l
C 840 if peek(p+123+i)>42 then next i:goto 590
850 return
O 860 Print"#####":Printtab(27)"#####      "
870 Printtab(27)"#####      "
O 880 Printtab(27)"#####      "
890 Printtab(27)"#####      "
O 900 Printtab(27)"#####      "

```



- 1580 poke v+39,0:poke v+21,1:return
- 1590 data. sas,dinnye,manjalica,jog,ablak,koncsolya,kompreszor,kompromisszum
- 1600 data. baromfi,kazetta,csatorna,lemez,program,tengerpart,csécsebecse
- 1610 data. paplan,kemence,foci,szatyor,polc,matrica,tulajdonos,szobortalapzat
- 1620 data. haza,kocsi,elem,pipacs,pingvin,tanmenet,parancsnok,parasztaszonny
- 1630 data. kacsa,gomb,bomba,posta,gerelye,uborka,fal,kilincs,fa,uc,vers
- 1640 data. antenna,pelenka,kapocs,farok,zebra,antilop,zsinor,kecske,kutya,kupak
- 1650 data. lufi,perec,zsemle,csoki,hang,petrezselyem,dohoz,lyuk,birka,szem
- 1660 data. tank,arany,bronz,patak,hungarocel,kalap,sin,mozdony,borz,medve,tigris
- 1670 data. teke,kocka,eper,krumpli,cseresznye,meggy,alma,notring,toll,alkony
- 1680 data. cylinder,centrum,cigaretta,citrom,narancs,barack,turista,serif
- 1690 data. kard,puska,pisztoly,csavar,vas,csata,haj,comb,ruha,csend,angyal
- 1700 data. vitorla,hely,keret,december,november,szeptember,centrifuga,fog,hegy
- 1710 data. 0,4,0,0,6,0,0,7,0,0,31,128,0,127,224,1,255,0,3,254,0,7,255,0
- 1720 data. 7,255,128,15,194,192,15,194,96,31,254,48,31,254,24,31,252,28
- 1730 data. 63,252,62,63,252,63,63,252,109,127,254,108,127,254,204,231,192,204
- 1740 data. 7,193,140
- ready.

## Tisztelt Commodore Újság!

Olvastam a Commodore Újság 1991. évi 2. számát. A 4. oldalon találtam egy nem működő programot, s a szerzője azt kérte, hogy segítsünk neki, ha tudjuk, hol a hiba. Én azt hiszem megtaláltam a hibát, ennek elmagyarázásához viszont szükséges az interrupt (megszakítás) részletes ismerete.

A számítógép mikroprocesszora úgy van megépítve, hogy minden 100-ad másodpercben megszakítja az addig futó gépi kódú programot, és hozzáfog egy másik program végrehajtásához, ezt a programot nevezik interruptnak. Ez a rutin végez olyan műveleteket, mint pl. az óra növelése, kazettásmagnó motorjának figyelése, osztott képernyőnél a megfelelő grafikus/karakteres kép váltása stb. Ha ezzel a programmal végzett, akkor ott folytatja, ahol az előbb abbahagyja. Most nézzük, hogy ezt hogyan is csinálja!

Amikor itt az ideje a megszakításnak, akkor a processzor betölti az utasítás számlálójába a \$FFFE-\$FFFF címen elhelyezkedő 2 byte-ot, ami a PLUS 4-en a \$FCB3 címet adja, tehát erre ugrik a program. Itt az történik, hogy elmenti a verembe a regisztereket, hogy visszatéréskor ugyanúgy folytatódhasson a program, mint meghívás előtt. Utána ugrik a \$CE00-ra, lényegében itt kezdődik a megszakítási rutin. Ennek végeztével a program visszatér a \$FCBE címre, itt visszatöltődnek a regiszterek, s egy RTI utasítás vet véget az interruptnak. Itt megjegyzem, hogy a hibák elkerülésének okáért az interruptot le lehet tiltani egy SEI utasítással. Ilyenkor addig nem generálódik megszakítás, míg egy CLI utasítással ezt nem engedélyeztük. A hiba okának felderítéséhez egy másik dologgal, a ROM/RAM lapozással is meg kell ismerkednünk:

A ROM/RAM lapozásra azért van szükség, mert e két memória fedi egymást a \$8000 és \$FFFF között, s hogy el tudja dönteni a processzor, hogy melyikből kérünk adatot. Mivel az LDA utasítás alapvetően \$8000 felett a ROM-ot olvassa, ezért lapozás nélkül az „alatta lévő” RAM-ot nem tudnánk használni. Ezért: ha programban kiadunk egy STA \$FF3F utasítást, akkor \$8000 felett MINDEN hivatkozás a RAM-ra fog vonatkozni. Visszakapcsolni a ROM-ra az alábbi utasítással lehet: STA \$FF3E. Itt jön a keveredés! Mi van akkor, ha a RAM van felül

lapozva (STA \$FF3F) és megszakítás generálódik? Ekkor a RAM \$FFFE-\$FFFF címét használja, s mivel ott véletlenszerű cím van, ezért több dolog előfordulhat: a program lemerevedhet, egy csomó BREAK-et ír ki, vagy egy „értelmes” programot hajt végre véletlenül, és ezzel is okozhat kárt.

A megoldás egyszerű: mielőtt a RAM-ot lapoznánk, felülre, tiltsuk le az interruptot, majd a ROM újbóli engedélyezése után engedélyezzük az interruptot is. A gyakorlatban ez így néz ki:

```
SEI          ;Interrupt letiltása
STA $FF3F   ;RAM felülre lapozása
. . .       ;Itt lehet beolvasni
. . .       ;a byte-ot a RAM-ból
. . .
STA $FF3E   ;ROM felülre lapozása
CLI         ;Megszakítás engedélyezése
```

Ezt a leírást nem csupán Láng Attilának szánom (talán azért sikerült ilyen hosszúra), hanem a Commodore újság többi lelkes olvasójának is, akiket érdekel a gépi nyelv és számítógépük működési elve. Ezért ha lehet, akkor e levelet a Commodore újságban közöljék le. A következő levelemben, ha önöket érdekli, akkor a PLUS/4-nél a resettel fogok foglalkozni.

Az újságban megindulhatna egy másik rejtvénytársorozat, mégpedig olyan, hogy egy hibátlanul látszó gépi kódú listában kell megtalálni a hibát. Én — ha önöket érdekli — küldök egy-két anyagot a saját dolgaimból.

Batman

## Tisztelt Szerkesztőség!

Mellékelten küldöm a közkedvelt DISK DEMON referencia kártyáját. A referencia kártya a 64er magazin nyomán készült.

Tisztelettel:

Honti József



## LOAD

(Csak a lemez meghajtó lehet bekapcsolva!)

## DISK DEMON

## BEJELENTKEZÉS

SPACE	Program start - turbóval
CRTL SPACE	Program start - turbóval és a fej inicializálásával

## PARANCSONK

R	Blokk olvasás
SHIFT R	Blokk olvasás, ha előzőleg 23-as vagy 27-es RED ERROR volt
+	Sektor = Sektor + 1
-	Sektor = Sektor - 1
SHIFT +	Track = Track + 1
SHIFT -	Track = Track - 1
C= +	Következő teljes sáv
C= -	Előző teljes sáv
N	Logikailag láncolt szektor olvasása. (Nem működik, ha a fájl utolsó blokkján vagyunk!)
L	Előzőleg olvasott blokk visszahívása
C= N	Láncolt blokk áttekintése
W	Aktuális blokk lemezre írása
SHIFT W	Aktuális blokk -fejléc módosítás utáni- lemezre írása
T	Félsávós olvasási mód BE/KI (Lemzre írásnál a szomszédos sáv törlődik!)
E	Szerkesztő menu hívása

## KÉPERNYŐ

<b>1. Sor</b>	
Track	Aktuális sáv adatok
Sektor	Aktuális szektor adatok
Byte	Aktuális bájt érték a szerkesztőben
<b>2. Sor</b>	Lemz meghajtó hibáüzenetei
<b>3. Sor</b>	
COMMAND	Aktuális parancs kijelzése
SPEED	Aktuális lemezsebesség (A=automatikus)
<b>4+5. Sor</b>	Blokkfejléc adatai
SGN	Blokkfej azonosító értéke
CHK	Ellenőrzőösszeg
SEC	Szektorszám
TRC	Sávszám
ID2 ID1	Valódi blokk ID értékek
BYT	Ezeknek a bájtoknak az értékét a DOS figyelmen kívül hagyja
<b>6. Sor</b>	Adatblokk információk
B-SGN	Adatblokk azonosítója (normál:\$08)
B-CHK	Adatblokk ellenőrzőösszeg
H-SGN	Blokkfej azonosító értéke, amelynek a lemezen kellene lennie! (Olvasáskor a DD. nem figyel.)
<b>Szerkesztőmező</b>	Ez a DISK-DEMON munkaterülete.

## EGYEB FUNKCIÓK

B	Bitszinkronizáció módosítása (Sebesség kijelzése (A=Automatikus))
SHIFT B	Teljes lemezsebesség értékei
S	Lemez tesztelés (szkennelés) + = hibátlan blokk Zahl = hiba száma █ = gyilkos sáv Táblázat átnézése gombnyomással RUN/STOP: kilépés a funkcióból
SHIFT S	Utolsó hibatáblázat vissza
F	Formázás adott sávon, adott fejléccel
U	Sáv törlése (ERROR 21)
K	Gyilkos sáv írása
C	Beteg sáv gyógyítása
Q	Megadott bájt-sorozat keresése (Max: 15 karakter, utolsó kar: \$00) Keresés RUN/STOP-ra indul
D	Lemez tartalomjegyzék a képernyőre
0..9	Szerkesztő felcserélése a pufferral
CRTL C=	Átkapcsolás HEXA/DECIMALIS között
C= RUN/STOP	Kilépés BASIC-be (RESTART: RUN)

## EDITOR PARANCSONK

SHIFT INS/DEL	Befűzés
INS/DEL	Törlés a kurzorpozíció alatt
SHIFT CLR/HOME	Adatblokk feltöltése \$00-val
CLR/HOME	Kurzor a legelső sorra ugrik
SHIFT RETURN	Kurzor az adatmezőbe lép
F1	Blokk a \$00-dik bájtól mutatva
F3	Második blokkfél mutatása
F5	Görgetés felfelé
F7	Görgetés lefelé
H	Fejléc adatainak módosítása (Uigyázz)
I	Adatblokk azonosítók módosítása
RUN/STOP	Szerkesztő elhagyása

TRACK: 18	SECTOR: 000	BYTEN: \$00							
COMMAND: █	DISK - DEMON	SPEED: A							
SGN	CHK	SEC	TRC	ID2	ID1	BYT	BYT	01234567	
\$08	\$16	\$00	\$12	\$35	\$31	\$00	\$35	HVQR5101	
B-SHN: 007		B-CHK: 163		H-SGN: 008					
: 00 :	12	01	41	00	04	10	02	08	2AACCDPH
: 08 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 10 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 18 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 20 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 28 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 30 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 38 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 40 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 48 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 50 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 58 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 60 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 68 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 70 :	00	00	00	00	00	00	00	00	CCCCCCCC
: 78 :	00	00	00	00	00	00	00	00	CCCCCCCC

## Tisztelt Tagtársaim!

Bizonyára többüknek birtokában van a LABEL BASIC nevű program, de használatát nem ismerik. Talán segíteni tudok az alábbi ismertetővel.

A Commodore PLUS/4 (és bővítő C16) nagy memóriaterületet bocsát a BASIC-programozás rendelkezésére. Így aztán nemcsak terjedelmes, de bonyolult programok készítésére is lehetőség nyílik. Bár van néhány olyan utasítás a PLUS/4-nek, mely megkönnyíti a programozó munkáját (DO, LOOP, WHILE, UNTIL, EXIT), az ugró- és szubrutinhívó utasítások mégiscsak nehezen megjegyezhető számokra hivatkoznak, mely számok aztán sorok betoldásakor, újraszámolásakor meg is változnak.

A LABEL BASIC javítani próbál ezen a helyzeten. Mint nevéből is kitűnik, ez egy címkéket használó BASIC-bővítés.

Bármely sor elejére címke írható, pl.:

10 @ NYOMTAT: PRINT A;"DB."

Erre a sorra ugrani bármely ugróutasítással lehet, mint az eredeti BASIC-verzióban, de lehet a LABEL BASIC szokásos szintaxisa szerint is:

50 GOTO 10 £ NYOMTAT

vagy:

60 IF A > B THEN 10 £ NYOMTAT

Miért kell címkét használni, ha úgyis oda kell írni a sorszámot az utasítás és a font (£) jel közé?

De nem kell odaírni! Az ugró- vagy szubrutinhívó utasítások után a font jellel megjelölt pusztá címkét is odaírhatjuk.

Így: 50 GOTO £ NYOMTAT

Ha ezután egy R parancsot adunk, az összes címkehivatkozással ellátott ugróutasítás után beíródik a címkének megfelelő sorszám (tehát ez lesz belőle: 50 GOTO 10 £ NYOMTAT).

Ha az ugróutasítás után helytelen sorszám volt, azt is kijavítja.

A W utasítás kiírja az összes címkét.

A T utasítás bekapcsolja a nyomkövetést. Nem sorszámok, hanem az aktuális címkék íródnak ki.

A O (off) utasítás kikapcsolja a nyomkövetést. A T és a O utasítás független a TRON-tól és a TROFF-tól.

A L parancs listáz. Kétféle módon lehet:

L @ CIMKE ... az @ CIMKE-től kezdve listáz

L £ CIMKE ... kiírja az összes olyan sort, amelyben hivatkoznak @ CIMKE-re

Parancs módban a tesztelést könnyíti meg, hogy GOTO, GOSUB, RUN után sorszám nélkül állhat £ CIMKE.

És a poén! Miután megszerkesztettük, kipróbáltuk a bonyolult programot, kiadhatjuk a C parancsot. Ez a parancs olyan programmá változtatja a LABEL BASIC alatt szerkesztett programot, amely aztán LABEL BASIC nélkül is futtatható.

Tagtársi üdvözlettel:

Koszper Vilmos

## Világújdonság



14 nyelv egy szótárban — 182 különféle  
szótár egy zsebben  
Bármelyik nyelvről bármelyikre.  
100 000 szavas memória!

MAGYAR	NÉMET
CSEH	ANGOL
SZLOVÁK	FRANCIA
SZERB	OLASZ
LENGYEL	SPANYOL
OROSZ	PORTUGÁL
SVÉD	DÁN

Nyelvenként — 6000 szó — 50 témakörben 550 mondat. Teljes ábécék!  
8 számjegyes számológép!

Automatikus kikapcsolás! — 4 darab elemmel működik, mely tartozék!

Ára: ~~15 900 Ft~~ helyett **14 900 Ft (ÁFA-val)** Visszonteladóknak nagykereskedelmi ár!

Vásárlási utalvány és megrendelőlap újágunk 34. oldalán található

# Lexikon

## I. rész

*Sokan hajlamosak elfeledkezni arról, hogy vannak számítógépes kezdők is, akiknek bizony a szakkifejezésekkel gondjaik vannak. Akik nem tudják, egy grafika mitől lehet „híres”, vagy mit keres az amerikai hírszerzés (CIA), egy lapos építésű hajófajta (monitor), főleg pedig egy közlekedési eszköz (busz) a számítógépben. Elsősorban nekik indítjuk most a fogalomlexikon sorozatunkat. Ha nem is akadémiai, de remélhetőleg érthető megfogalmazásban igyekszünk közreadni olyan ismereteket, amelyek segíthetnek eligazodni ebben az ismeretlen, új világban.*

### RAM (Random Access Memory)

Azaz: közvetlen elérésű tároló. Egy RAM-ba adatokat be lehet írni, és azokat onnan ki is lehet olvasni. Ennek a tárolótípusnak a legnagyobb előnye az, hogy minden egyes bitet fölül lehet írni, vagyis meg lehet változtatni, szemben a ROM-mal, ahol az adatokat nem lehet megváltoztatni. A hátrány viszont az, hogy a bevitt információk a számítógép kikapcsolásakor elvesznek. A RAM-ok rengeteg apró kis kondenzátorból épülnek föl, amelyeknél a töltési állapot hordozza a bináris információt („0” vagy „1”). A tároló építőelemeknek szükségük van egy úgynevezett „Refresh-jelre”, amely a kondenzátorok töltését meghatározott időközönként „felfrissíti”. Erre azért van szükség, mivel a kondenzátorok egy idő után kisülnek. Ekkor viszont a tárolt információ menthetetlenül elveszne.

### ROM (Read Only Memory)

Azaz: csak olvasható tároló, fixtároló. A ROM építőelemek olyan rögzítetten beprogramozott adatokat tartalmaznak, amelyeket nem lehet megváltoztatni. Előny: Az adatok a gép kikapcsolása után is megmaradnak. ROM építőelemeket használnak arra, hogy operációs rendsze-

rek azonnal a bekapcsolás után a rendelkezésre álljanak. Mivel a ROM-okat mindig a gyártó programozza, és azokat mindenkor az adott rendszerre illesztik, ezért az azokban található programok esetében gyakran használják a „firmware” kifejezést. A ROM építőelemek meglehetősen drágán készülnek, mivel azokhoz speciális „maszkokat” (mikrofilmek a chippek megvilágításához) kell kifejleszteni. A C64-ben használt ROM-ok lábcskáinak kiosztása kompatibilis a 25xx sorozatú EPROM-okéval, ezért azokat minden probléma nélkül ki lehet cserélni saját programozású EPROM-okra (például egy 2564-es típusúra).

### EPROM (Erasable Programmable Read Only Memory)

Azaz: törölhető, programozható, csak olvasható tároló. Az EPROM-ok hasonlóan a ROM-okhoz rögzített értékek tárolására szolgálnak. Ez azt jelenti tehát, hogy a kikapcsolás után sem vesznek el az abban található információk. A hátrány, hogy ezt az információt nem tudjuk megváltoztatni. Ezeket az EPROM-okat gyakorta arra használjuk, hogy saját operációs rendszert vagy jelkészletet programozzunk, majd ezeket az építőelemeket az eredetiekkel kicseréljük. Az EPROM-ok igen alkalmasak kis sorozatú számítógé-

pekhez, mivel a programozásuk viszonylag olcsó, és a költségek alacsonyak. Az EPROM-okat ultraibolya fényvel törölni lehet, majd ezt követően egy EPROM programozóval (EPROM égető) újra lehet programozni.

### Operációs rendszer (KERNEL ROM)

Az operációs rendszer egy olyan, gépi kódban megírt program, amely azzal a feladattal rendelkezik, hogy a felhasználót tehermentesítse. Az operációs rendszer átveszi a felhasználótól az összes feladatot, amelyet egy számítógép jobban tud megoldani, és az a felhasználót csupán feltartaná. Így például az operációs rendszer a (C64/C128 esetében ez egy ROM-ban áll) feladatai közé tartozik az adatok fogadása illetve kivitele, amelyeket a perifériák (nyomtató, floppy) küldenek, illetve amit a gép azok felé irányít. Ugyanígy az operációs rendszerre van szükség a kurzor villogtatásához, vagy a billentyűzet lekérdezéséhez. Ez veszi át azt a feladatot is, ami akkor jelentkezik, ha mondjuk egy BASIC sort kell elhelyezni a memóriában. Ehhez olyan dolgokat kell csinálni, amelyek a felhasználót csupán hátráltatnák a munkájában.

### HIRES grafika (High Resolution Grafik)

Azaz: nagyfelbontású grafika. A számítógép rendes állapotban szöveges módban dolgozik. Ebben a módban pontosan 1000 jel ábrázolása lehetséges. Ez a szövegekhez megfelelő, grafikák esetében azonban ennyi kevés. Ezért hozták létre a HiRes módot. Ebben az üzemmódban a rendelkezésre álló 64 000 képpont mindegyikét egyenként is el lehet érni. Így nagyfelbontású grafikákat is meg lehet jeleníttetni a képernyőn. Ez azonban igen helyigényes. Egy normál szöveges tároló 1 kbyte kapacitással kell hogy rendelkezzen, nagyfelbontású grafikánál a szükséges tárolókapacitás már 8 kbyte-ra növekszik. A nagyobb számítógépek a szövegeket, mivel elegendő tárolóhellyel rendelkeznek, közvetlenül a HiRes képernyőre írják, külön szöveges képernyőre nincsen szükség.

### BASIC interpreter

A BASIC interpreter teszi a számítógépet képessé arra, hogy az a beadott parancsokat és utasításokat, melyeket a fel-

## Soros busz

A soros busz intézi el az adatforgalmat a perifériák felé. A soros buszra floppyt, nyomtatót vagy plottert lehet csatlakoztatni. A soros átvitel esetében egy byte-ot nem lehet egyszerre küldeni, csak bitenként. Éppen ezért a kommunikációnak ez a módja nem mondható gyorsnak, amely különösen a nagy periféria forgalom esetében hátrány. Az átvitel előnye viszont, hogy nincsen szükség sok vezetékre. Szükséges egy, amelyik az információt bitenként viszi át (Serial Data In/Out), egy másik, amelyik az órajelet hordozza (Serial CLK), végül a harmadik arra felügyel, hogy a periféria készenlétben áll-e az adatok fogadására és küldésére. A kört a föld és a RESET zárja le.

## Párhuzamos adatátvitel

A párhuzamos adattovábbítás a leggyorsabb átviteli lehetőség. A soros átvittel szemben itt több vezetékre van szükség a kapcsolat létrehozásához. Ezek pedig a következők: 8 vezeték egy egész byte egyszerre történő továbbításához. Ehhez jön még a föld-, és két speciális vezérlő jeleket hordozó vezeték is. Ez utóbbiak értelme a következő: Az egyik a számítógép kérdezi le a meghívott perifériát, vajon az készen áll-e az adatok küldésére vagy fogadására. A második vezetéken válaszol a készülék a számítógépnek. Ha ez válasz negatív, akkor a gép mindaddig vár, amíg a periféria a készséget nem jelzi.

Folytatjuk



GLOBIOS



## 3M DISZKETTEK

		A teljes mennyiségre	
		100 db felett	1000 db felett
5.25 DS DD	52 Ft	48 Ft	44 Ft
DS HD	82 Ft	78 Ft	72 Ft
3,5 DS DD	78 Ft	74 Ft	68 Ft
DS HD	148 Ft	140 Ft	128 Ft

## MONITORÁLLVÁNY

3—14 kg teherbírású	14 200 Ft
8—24 kg teherbírású	14 800 Ft
Kiegészíthető telefonasztallal, támasztólábbal és egyéb hasznos elemmel.	

További ajánlatunk: zajcsökkentő doboz nyomtatókhoz, PC-munkahely, nyomtatóasztal.

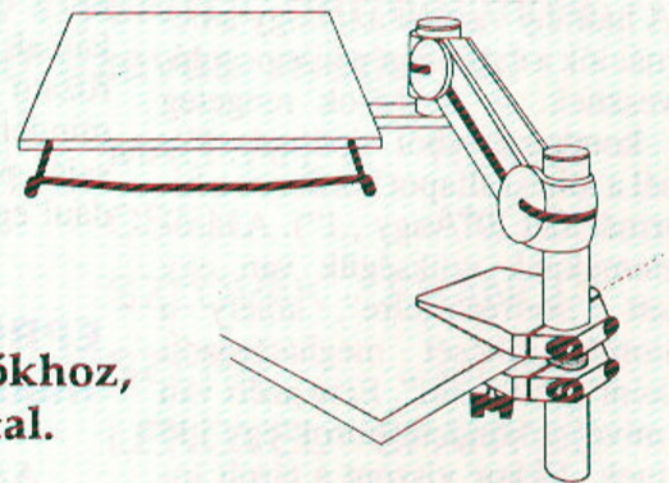
Árainkhoz ÁFA-t számítunk!

Üzletünk a III. kapunál 8-tól 16 óráig áll vásárlóink rendelkezésére.

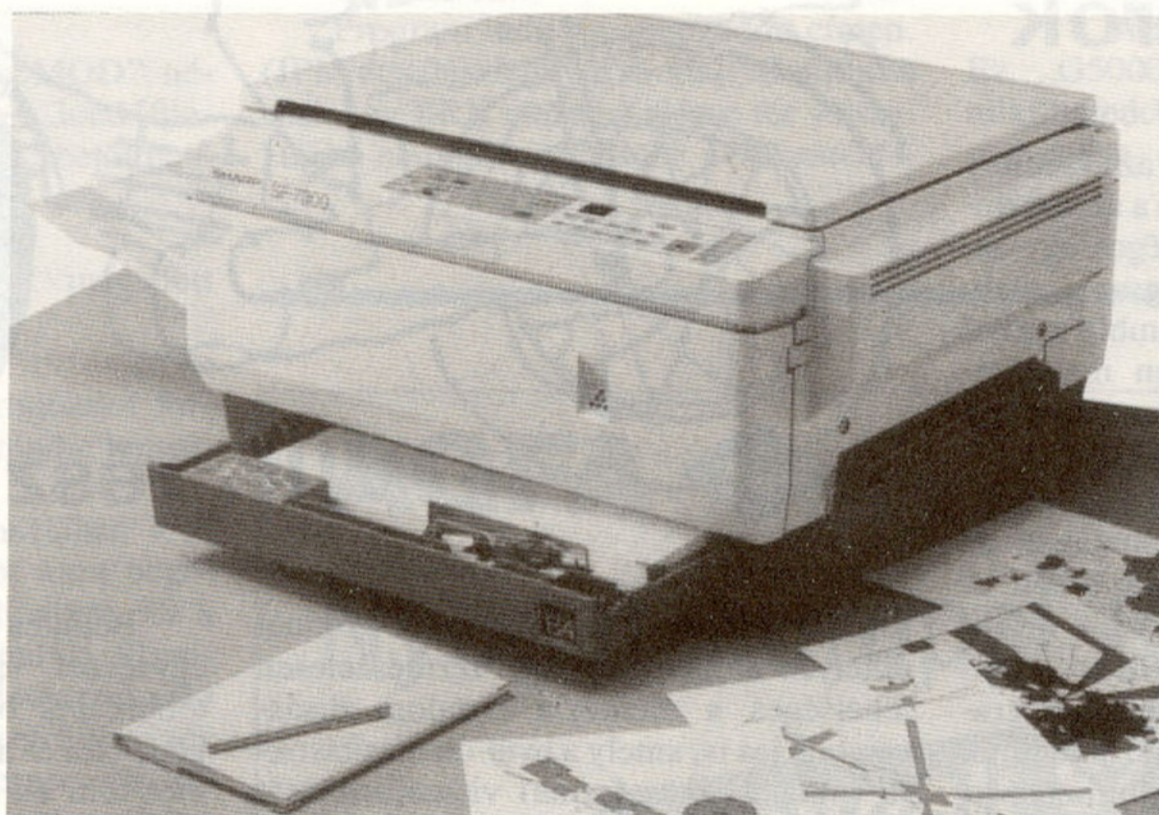
Közvetítő ügynökök és viszonteladók jelentkezését is várjuk, magas jutalékkal!

**MOM GLOBIOS Kft.**

1124 Budapest, Csörsz u. 35.  
Levél cím: 1399 Budapest, Pf.: 701/413.  
Tel.: 156-4122/587 vagy 155-4730  
Fax: 155-9736, Telex: 22-4151



# A MEGBÍZHATÓ ÜZLETTÁRS



	Régi ár	Új ár		Régi ár	Új ár
Z-30	<del>66 000 + ÁFA</del>	59 900 + ÁFA	SF-7800	<del>214 900 + ÁFA</del>	184 900 + ÁFA
Z-50	<del>79 900 + ÁFA</del>	69 900 + ÁFA	SF-7850	<del>249 900 + ÁFA</del>	214 900 + ÁFA
SF-6100	<del>119 900 + ÁFA</del>	99 000 + ÁFA	SF-8300	<del>359 900 + ÁFA</del>	290 000 + ÁFA
SF-7300	<del>124 900 + ÁFA</del>	109 900 + ÁFA	ALSÓ SZEKRÉNY SE 7800		15 000 + ÁFA
SF-7320	<del>159 900 + ÁFA</del>	119 900 + ÁFA	ALSÓ SZEKRÉNY SE 7850		15 000 + ÁFA
SF-7370	<del>199 900 + ÁFA</del>	144 900 + ÁFA			

**Áraink egy év garanciát és a kellékanyagok árát is tartalmazzák!**

SHARP termékek a **KOPI-KER**-től  
KERESKEDELMI KFT.

1054 Budapest, Kálmán Imre u. 27. Telefon: 132-4392, 111-2083, 132-2544

## Gépi kódú programozás Commodore gépeken

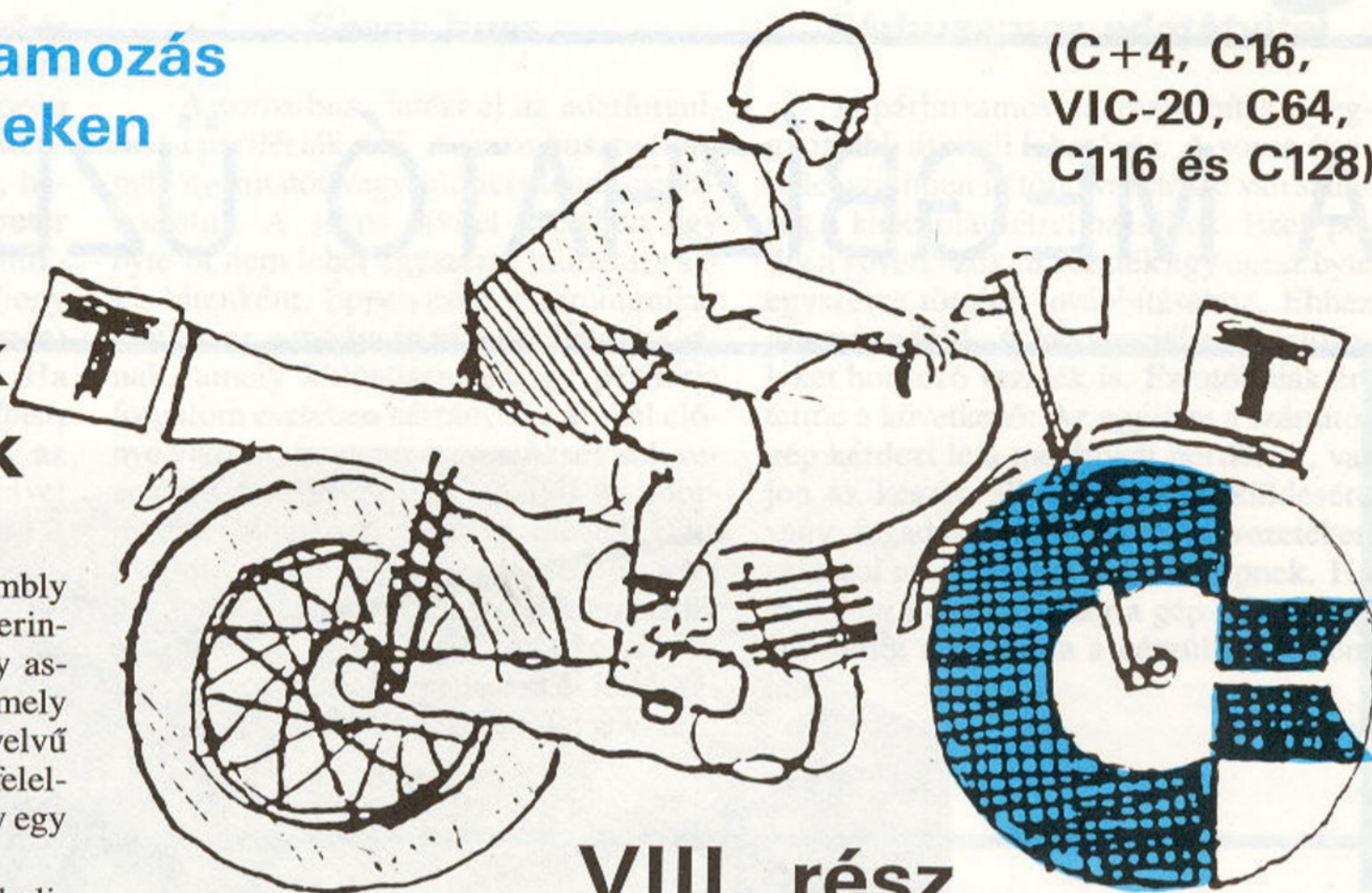
(C+4, C16,  
VIC-20, C64,  
C116 és C128)

### Assemblerek és monitorok

Említettük már, hogy mi az assembly nyelvvel foglalkozunk, s nem a szószertintti gépi kóddal. Ehhez szükséges egy assembler (összeszerelő) program, mely ténylegesen elhelyezi az assembly nyelvű utasításainknak egyértelműen megfeleltethető byte-okat a memóriában (vagy egy file-ban, ha lemezre fordítunk).

Fontos különbséget tenni a szimbolikus assembler (ezt értjük általában az „assembler” alatt) és a monitor között. A szimbolikus assembler program orientált, valamely szerkesztő (például a BASIC rendszer) segítségével megírt programot fordít le gépi kódra. Használhatunk tízes, kettes és tizenhatos számrendszert (elérjük írt „%” és „\$” jellel). Lehetőségünk van a tényleges memóriacímek helyett ún. szimbólumok (több karakteres változónevek) használatára. Ezekre hivatkozhatunk ugráskor és jelölhetnek adatot is. Ez jelentős könnyítés, hiszen amikor javítanunk kell a programban (tipikus eset: be kell szűnni néhány utasítást) elég a szerkesztővel átírni (beírni) módosított (vagy új) sorokat és ismét lefordítani a programot. Ezt a memóriában ténylegesen elhelyezkedő programmal nem tehetjük meg. Ott át kell másolni nem használt memória területre a programot, itt beleszúrunk, a további részeket emögé másoljuk vissza, majd az egészprogramban átírjuk a megváltozott részre való címhivatkozásokat (eltolódnak az ugráscímek, az adatok helye...). Ez iszonyúan kényelmetlen már egy rövid programnál is. Ezekért cserébe fizetünk a fordítási idővel amit ki kell várni.

A monitor a memóriában való vájkálásra alkalmas. Az előbb említett másolás-beírás-visszamasolás-módosítás műveleteket ezzel lehet elvégezni. Bár láttuk, hogy ez kényelmetlen, mégsem nélkülözhető. Nagyon rövid programok (a mi példáink) beírására, módosítására alkalmas gyors segédeszköz. A beírás itt egy sor assembly formájú beírását jelenti, ahol nem használhatunk szimbólumokat, csak konkrét értékeket. Ezt a sort egy, a monitorba beépített, egysoros assembler



### VIII. rész

fordítja le byte-okra, amelyeket azonnal elhelyez a memóriában. Természetesen lehetséges a memória tartalom disassemblálása is, amely a byte-ok utasításokra (mnemonikus forma) visszafordítását jelenti. Ez sokkal egyszerűbb, mint ahogy tűnik: beütünk egy sort, azt kilistázzuk és ugyanúgy néz ki, mint ahogyan beütöttük, beleírhatunk, újra listázhatjuk stb. Általános elterjedt egyéb szolgáltatások: memória hexa formájú listázása javítási lehetőséggel, memóriaterületek másolása, összehasonlítása, feltöltése egy értékkel, kimentése, betöltése.

Szerencsénkre létezik egy assembler rendszer, mely a VC-20 kivételével valamennyi C = gépre elkészült (és könnyen megszerezhető). Az a JCL-ASSEMBLER, amivel ezentúl foglalkozunk.

A monitorral nincs ilyen nagy szerencsénk. A C+4-ben és a C-128-ban lévő monitorok azonos módon használhatóak és a legtöbb C-64-es monitor (pl. amit JCL-ASSEMBLER C-64-es változata tartalmaz) is ezekre hasonlít, de eltérések könnyen lehetnek. VC-20-ra assemblert nem ismerünk, monitort is csak egyet, a TG-MON-t, de az nem fut mindenféle bővítővel (pl. a JK-s bővítő nem jó neki).

A JCL-ASSAMBLER-ről részletes leírást a „100+4 Játékok és felhasználói programok C16-PLUS/4” című könyvben találhatunk. Minden gépre azonos a kezelés lényege, egy-két parancsban térnek csak el, ami a gépek eltérő BASIC-rendszeréből adódik. A JCL-ASSEMBLER vázlatos kezelése:

Betöltés után RUN-nal indítsunk. Az egyrészes változat azonnal elindul, a többrészes beazonosítja, milyen típusú gépen fut és az annak megfelelő főprogramot tölti és indítja el. Bejelentkezéskor kiírja az új utasításokat, majd visszatér a BASIC szerkesztőbe. Ekkor feltétlenül üssük be az EDIT parancsot.

Ilyenkor a szokásos módon lehet sorzámmal ellátott sorokat bevinni, amelybe most assembly utasításokat írhatunk. Minden címkét, mnemonikot és operandust legalább egy szóközzel kell egymástól elválasztani. Ha a sorban szerepel „;”, akkor az utána álló részt megjegyzésnek veszi a fordító (átlépi). Egy sorban (emiat és az áttekinthetőség miatt) csak egy utasítás állhat.

A programot speciális sorokkal kell kezdeni, amelyek megadják, hova (milyen memória címre) fordítsa a programunkat (ez a „\*=” parancs). Ha kész a programunk, a kimentést PUT „filenév” paranccsal tehetjük meg. A betöltésre a GET „filenév” parancs szolgál.

Az ASM parancs beírásával indíthatjuk el a fordítást. Ezután a gép megkérdezi a fordítás paramétereit. Minden paraméter elhagyható, de valamit persze meg kell adnunk közülük, ha több, akkor vesszővel elválasztva. A paraméterek (a kettőspont nem részük):

filenév:

A fordítandó (lemezről fordításkor) és a fordított (lemezre fordításkor) program neve (lemezről, -re esetén majdnem azonos!)

s:

Képernyőre készít fordítási listát.

p:

A nyomtatóra készít fordítási listát. Ha van nyomtatónk, a fontos programokról és hiba kereséskor kérjünk listát, nem fogjuk megbánni, szépen szerkesztett, jól áttekinthető.

o:

A fordítás lemezre történik, „file-név,OBJ” néven készül el a fordítás, a formátum speciális szabvány szerinti, ne használjuk.

m:

Lemezre fordít „file-név.MOD” néven. Ez lemezről „load” file-név.MOD”,8,1” utasítással betölthető tényleges gépi kódú program lesz.

x:

Memóriában fordít a megadott helyre.

c:

A memóriában lévő programot fordítja. Ha nem adjuk meg, mindig lemezről akar fordítani.

Pl. „x,c” esetén a memóriában lévő forrásszöveget (a programot amit begépeztünk) lefordítja a kijelölt helyre. Ez a folyamat igen gyors, 500 sornál kb. 10 másodperc. A lemezről (vagy lemezre) való fordítás alig lassab a lemezes töltési (és mentési) műveleteknél. Persze ez 100 blokkos forrás programnál már néhány perc.

Pl.: 1000 \*=\$5000

1010 ldx #15

1020 ide txa

1030 clc

1040 adc #\$40

1050 jsr \$ffd2

1060 dex

1070 bne ide

1080 rts

Fordítás után (ASM ... „x,c”) sys5\$4096-ra ki fogja írni a „onmlkjhg-fedcba” betűket.

## Monitorok kezelése:

Indítás után (ami a beépítettnél MONITOR, egyébként betöltés és RUN vagy SYSkezdőcím) bejelentkezik és kiírja a  $\mu$ P regisztereit.

Egykarakteres parancsokat értelmez, mely után szóközzel (néhol vesszővel) elválasztva kell a paramétereket írni. Minden szám hexadecimális, de itt nem kell kiírni elé a „\$” jelet, kivétel az egysoros assembláláskor az utasításban szereplő adatok előtt. Ott is csak hexadecimális számot fogad el, de a „\$” jel kötelező. Gyakori parancsok (a „:” nem részük):

R:

Kiírja a  $\mu$ P regisztereit. Átírhatjuk őket, tesztelésnél hasznos.

M kezdőcím végcím:

Hexadecimális formában listázza ki a megadott memóriatartományt.

D kezdőcím végcím:

Disassemblálva (visszafejtve) írja ki a megadott memóriatartományt.

A kezdőcím mnemonik operandus (ha van):

Utasítás lefordítása gépi kódra. Egysoros assembler.

G kezdőcím:

Program indítása a megadott címen. (JPM utasítást hajt végre oda). Monitorba csak egy BRK utasítást elérve tér vissza. Ha mi nem gondoskodunk a visszatérés-

ről, csak a RESET gomb segít. Ezt hívják „vakrepülésnek”.

X:

Visszatérés BASIC-be.

Pl.: A 5000 ldx #\$0f

A 5002 txa

A 5003 clc

A 5004 adc #\$40

A 5006 jsr \$ffd2

A 5009 dex

A 500a bne \$5002

A 500c brk

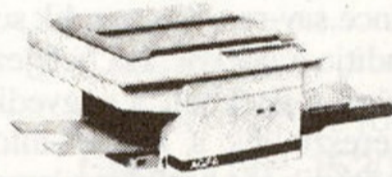
Ez „G5000” hatására kiírja az „onmlkjhg-fedcba” szöveget, majd visszatér monitorba.

Nagy Tamás

# AGFA

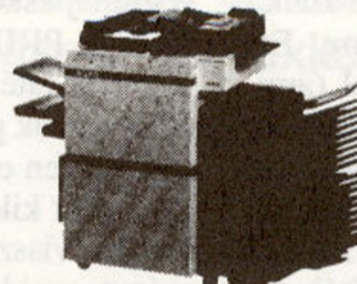
## Nyugat-európai Precizitás

### Azonnali szállítás



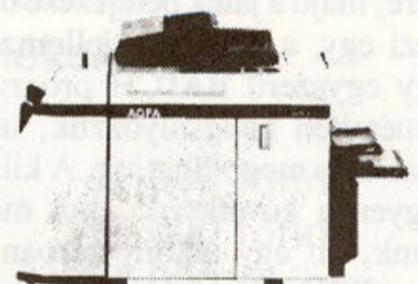
**Agfa X18**

Asztali fénymásoló  
50—200% Zoom-mal,  
gombnyomásra színes  
is másol.



**Agfa X38**

Út a csúcra.  
Színes másolás, Zoom.  
Percenként 30 másolat,  
kiépíthető 20 fiókos  
sorterral.



**Agfa X58**

Intelligens, nagy teljesítményű fénymásoló,  
automatikus lapadagolóval.

A Nyugat-európai technológiával készült AGFA FÉNYMÁSOLÓK garantálják a tökéletes minőséget; a COPY-gomb megnyomásával automatikusan végrehajtja a kiválasztott műveleteket.

A jól felszerelt budapesti Agfa-raktárból azonnal szállítjuk a kellékanyagokat és az eredeti Agfa alkatrészeket.

**És mindezt FORINTÉRT**

Ezek után mi akadály, hogy az információs kupont elküldje nekünk.

Autorisiert von Agfa-Gevaert Ges.m.b.H., Wien  
Az Agfa Wien hivatalos Képviseletje

**ASI** 

AGFA-ASI, 1113 Budapest, Bartók Béla út 120.

Kérem, rövid időn belül informáljanak az Agfa fénymásolókról.

Vállalat/Név: \_\_\_\_\_

Cím: \_\_\_\_\_

Telefon: \_\_\_\_\_

Áttérés *COMMODORE*-ről *IBM* gépre

# VÁLTÓ

Mottó:  
Minden gépnél van jobb

## A kötegelt állományok (**BATCH**)

A rendszeren futtatható programok a következő kiterjesztésűek lehetnek: COM, EXE és BAT. Az első kettő valóban programot jelent, míg a BAT egy úgynevezett kötegelt állomány, tulajdonképpen parancsok sorozata. Vegyük az átnevezésnél használt példát ismét. Ott arról volt szó, hogy mielőtt valamelyik játékos játszani kezdi a PRINCE játékot, a kimentett állást tartalmazó állományt átnevezi eredeti nevére, majd a játék befejezése után visszanevezi egy, a játékosra jellemző névre. Ezt egy egyszerű BATCH programmal (ami, ismételt hangsúlyozzuk, utasítások sorozata) is megoldhatjuk. A kiindulópont legyen a következő: ahol most tartózkodunk, ott egy alkönyvtárban — melynek neve PRINCE — található a teljes játék, továbbá a játékosok nevével jelzett mentésállományok. Ekkor a következő batch-programmal oldhatjuk meg a feladatot:

```
CD PRINCE
REN %1 PRINCE.SAV
PRINCE.EXE
REN PRINCE.SAV %1
CD ..
```

Ezt a programot például a COPY parancsot használva vihetjük be a gépbe, a következő módon:

```
COPY CON prindit.bat
```

Bevisszük a sorokat, enterrel lezárva, majd <ctrl> +z után elkészült a BATCH állomány, amit TYPE PRINDIT.BAT paranccsal ki is listázhatunk a képernyőre. A COPY paranccsnál azt használtuk ki, hogy a honnan hova paraméterek között a perifériák szabvány nevei is lehetnek, így jelen esetben „magyarra fordítva” azt mondtuk, MÁSOLJ a BILLENTYŰZETRŐL a PRINDIT.BAT állományba.

Ha kész a prindit állomány, azt a következőképpen használhatjuk:

### PRINDIT.BAT JATEKOS

Nézzük akkor sorról sorra, mit csinál a programunk. Az első sor egyértelmű, belépünk a PRINCE alkatalógusba. A második sorban kihasználjuk, hogy paraméterezni tudjuk a programokat, így a %1 helyébe a JATEKOS szöveg kerül, azaz az a szó, amit a PRINDIT.BAT után beírtunk. Tehát ha például B játékos játszik, akkor beüti: PRINDIT.BAT B, és a második sor végrehajtáskor a következő lesz:

```
REN B PRINCE.SAV
```

vagyis a játékos nevével jelzett állományt áthelyezzük prince.sav-ra. A harmadik sor egyszerűen elindítja a játékot. Ha befejeztük, azaz kilépünk a játékból, a negyedik sorban visszakereszteljük a mentésállományt, majd az ötödik sor hatására visszalépünk egy katalógust, vagyis oda kerülünk, ahonnan elindultunk.

Látható, hogy a program végrehajtása közben a játékprogram mintegy szubrutin-ként működött, amit jól ki tudtunk használni. Vigyázat! A BATCH programokból hívott BATCH programok esetén ez nem igaz! Pontosabban így nem igaz, mert ott is meg lehet oldani, csak egy kicsit körülményesebben, ezt a cikk vége felé ismergetjük.

A BATCH programok egy minimális szintű programozási nyelvet is rendelkezésre bocsájtanak, ezzel már programozhatjuk is gépünket. Ennek a „nyelvnek” az utasításai a következők:

```
ECHO
FOR
GOTO
IF
SHIFT
PAUSE
REM
```

Az utasítások működése a következő:

**ECHO** — Ezzel az utasítással a BATCH programunk kijelzéseit tilthatjuk le, illetve engedélyezhetjük. Alapesetben BATCH program végrehajtásakor minden sor és a végrehajtás eredménye is megjelenik a képernyőn, csakúgy, mintha a billentyűzetről vittük volna be. Az ECHO OFF hatására ezt a kijelzést tiltjuk le. Ha a BATCH programban egyéb programot hívunk meg, például egy játékot indítunk, akkor természetesen annak a képét azért látni fogjuk. Az ECHO utasítással szöveget is vihetünk a képernyőre, az alábbi formában:

```
ECHO Hello baby
```

Paraméter nélküli meghívása esetén az aktuális állapotot jelzi ki (on/off). Az ECHO OFF parancs előtt a kukac karaktert használva már az ECHO OFF sem jelenik meg.

**FOR** — Ciklusszervező utasítás, csakúgy, mint a BASIC-ben. Lehetővé teszi, hogy DOS utasításokat többször hajtsunk végre. Nemcsak BATCH programból használható. Pontos formája:

```
FOR %%változó IN (változólista) DO utasítás
```

A két százalékjellel és névvel jelölt változó a változólista elemeinek értékét veszi fel az utasítás végrehajtásakor. Az utasításban szerepelhet a %%változó. Egy egyszerű példával bemutatva:

```
FOR %%A IN (első.txt,masodik.txt,harmadik.txt) DO COPY C:eredeti.txt C:%%A
```

Ha ezt egy BATCH programként beírjuk és futtatjuk, a C lemezegységen lévő eredeti.txt állományt lemásolja három példányban, különböző, a változólistában megadott neveken. Természetesen ez is csak egy kiragadott példa, a FOR nagyon sokféle lehetőségeiből. Ha nem BATCH programban használjuk az utasítást, hanem egyszerű parancsként, akkor a változót nem kettő, hanem csak egy százalékjellel kell jelölni.

**GOTO** — Ugró utasítás, ami a két következő utasítással kombinálva jól alkalmazható ciklusok és elágazások szervezéséhez. Használatát ott példaként is láthatjuk majd, most csak annyit kell tudni róla, hogy a BATCH programban elhelyezett címkére ugrik. A címke: NÉV alakú, az utasítás ekkor GOTO CIMKE.

**IF** — Feltételes utasítás. Ha az általa vizsgált feltétel teljesül, végrehajtja a következő parancsot, egyébként nem. A feltételt ellenkezőjére is változtathatjuk a NOT tagadással. Sűrűn találkozhatunk vele különböző programcsomagok (több álló-



mányból álló programok) installációs részében. Ezeket általában a hajlékony lemezről szeretnénk feltenni a winchesterre, és ezt a másolást egy BATCH programmal vezérelni.

```
IF NOT EXIST %2:program.exe
GOTO MASOL
```

A programban a %2 a hívás során az aktuális paraméterek közül a második, itt például annak a lemeznek a betűjele, ahova másolunk. Ezt mondjuk egy README.TXT állományban közölhetjük a felhasználóval, például úgy, hogy az állomány tartalma: „Az installációs program az INST.BAT állomány. Használata: INST.BAT HONNAN HOVA.” Ekkor a fenti programrészlet megvizsgálja, a céllemezen létezik-e a program.exe állomány, és ha nem, elugrik a MASOL címkére. Ez akkor lehet hasznos, ha több lemeznyi anyagot kell felmásolni, mert így, ha ezt a lemezt már felvittük, leáll a másolás. Megjegyzés: Az EXIST — létezésvizsgálat — utasításon kívül az ERRORLEVEL-t is használhatjuk elágazások és ciklusok szervezésekor. Ez egy változó, melynek értéke 0, ha nem volt hiba az előző parancs végrehajtásakor. Használata hasonlít a BASIC ONERROR utasításához.

**SHIFT** — A paraméterek váltására szolgál. Mint már volt róla szó, a programokat paraméterezve is meghívhatjuk. Ha több paraméterünk van, akkor azokra a paraméter sorszámával hivatkozhatunk. Például több lemezegységünk is van (mondjuk két hajlékony lemez, és egy winchester, A, B és C betűjelű) és szeretnénk mindegyikről tartalomjegyzéket kérni a DIR paranccsal. Ezt megtehetjük a DIR háromszor egymás utáni kiadásával:

```
DIR A:
DIR B:
DIR C:
```

Ugyanezt bevihetjük BATCH programba, ami ugyanígy nézne ki de ekkor nem állna meg a listázás a lemezek után. Ráadásul még szeretnénk, hogy a listázás oldalanként is megálljon. Ezt elérhetjük a következő BATCH programmal:

```
DIR A:/P
PAUSE
DIR B:/P
PAUSE
DIR C:/P
PAUSE
```

Ha tetszőleges sorrendet szeretnénk, akkor a következő változtatással ezt is megtehetjük:

```
DIR %1/P
PAUSE
DIR %2/P
```

```
PAUSE
DIR %3/P
PAUSE
```

Ha viszont nem tudjuk előre, hány lemezegység katalógusát akarjuk megnézni, a következő legyen a programunk:

```
:UJRA
DIR %1/P
PAUSE
SHIFT
GOTO UJRA
```

Ez a program legyen például diri.bat nevű. Ekkor használata a következő: DIRI meghajtó, meghajtó, ...

A program első sora egy címke az ug-

ráshoz. A második sorban kiírja az első meghajtó directory-ját, oldalanként. A harmadik sor billentyűlenyomásra várakozik a teljes lemezkatalógus kiírása után, és a negyedik sor lépteti (SHIFT=váltás, eltolás) a következő paramétert a %1 helyére.

**PAUSE** — A feldolgozást megállítja, vár egy billentyű leütésére. Csakúgy mint az ECHO parancsnál, itt is adhatunk meg szöveget, amit kiír a képernyőre. Használható listázás, másolás stb. tördelésére, vagy figyelmeztető szöveg kiírására, például BATCH programból kiadott törlések



**GÉPKER  
Kft.**

**Gépkereskedelmi és Ügyviteltechnikai Kft.**

A Gépkér Kft. Canon szerviz az alábbi kedvező árakon értékesíti Canon fénymásoló- és faxkészülékeit:

Canon FC2	49.900 forint
Canon NP 1010	119.000 forint
Canon NP 1520	204.000 forint
Canon NP 2020	269.000 forint
Canon NP 3825	359.000 forint
Canon NP 6650	899.000 forint
Canon NP 8530	1.690.000 forint
Canon CLC 300	1.690.000 forint
Canon fax 80	54.000 forint
Canon fax 120	79.000 forint
Canon fax 270S	109.000 forint
ASI NT 1104 pénztárgép*	40.000 forint

\*mely teljes összegben visszaigényelhető az APEH-től

Az árak áfa nélkül értendőek és tartalmazzák az 1 év garanciát is. Vásárlásai esetén további 5% kedvezményt biztosítunk.

Cím: Bp., XIII., Frangepán u. 7.

Tel.: 120-9420 \* 129-9377 \* Fax: 120-9420

előtt. (Ugyanis ha nincs a BREAK=OFF megadva, a PAUSE várakzásakor a programot <ctrl>+C-vel leállíthatjuk.)

**REM** — Magyarázó szöveg írható a programba, ha nem volt előtte ECHO OFF, megjelenik a képernyőn.  
REM Ez egy magyarázat

## A BATCH programok láncolása

Ha a BATCH programból hívunk egy programot, annak lefutása után a vezérlés visszakerül a hívás utáni utasításra. Ezt használtuk ki a REM utasításhoz adott példában is. Ha azonban a BATCH programból újabb BATCH programot akarunk hívni, a COMMAND.COM egy különleges lehetőségét kell használnunk. Ez a /C kapcsoló, ami arra utasítja a parancsprocesszort, hogy hajtsa végre a megadott programot, majd folytassa a feldolgozást a hívás helyétől. A használata a következő:

COMMAND [keresési út] /C [keresési út]masodik.bat [esetleges paraméterek]  
Ha ez a sor szerepel a BATCH progra-

munkban, akkor a keresési út2 végén található BATCH programot végrehajtja az esetleges paraméterekkel, majd az első BATCH programunk fut tovább. Az első keresési út a COMMAND.COM elérését határozza meg.

## Egy különleges állomány, az AUTOEXEC.BAT

Ez is egy egyszerű kötegelt állomány, különlegessé az teszi, hogy minden rendszerindításkor automatikusan elindul. Ezáltal igen alkalmas arra, hogy az alapbeállításokat ebbe az állományba vegyük bele. A rendszer akkor is elindul, ha nincs ilyen állomány, de használatával kényelmesebbé tehetjük munkánkat.

## Mi kerüljön bele az AUTOEXEC állományba?

Még az első részek egyikében elkövettem egy hibát, ezt szeretném most itt korrigálni. Ott azt állítottam, hogy az XT kategóriájú gépek a DATE és TIME parancsokkal atomatikusan bekérik az időt

és dátumot, míg az AT gépek mivel van beépített órájuk ezt nem teszik. Nos, bármilyen IBM kompatibilis géppel dolgozunk, ha nincs AUTOEXEC, a gép induláskor kérni fogja a dátumot és az időt. Tehát ha készítünk AUTOEXEC.BAT programot, és nincs a gépünkben „örökóra”, akkor ennek a két parancsnak mindenképpen helye van az AUTOEXEC-ben. Ezen kívül a leggyakrabban használt programok elérési útját is illik beállítani a PATH paranccsal, valamint a PROMPT beállítását is itt szokás megtenni. Ha nincs rá ok, hogy ne tegyük meg, akkor a BREAK-on parancsot is vegyük be az AUTOEXEC-be. Ezt a parancsot a rendszer konfigurációs állományába (erről a következő részben) is betehetjük, ekkor ide felesleges berakni. Ezen kívül általában az utolsó parancs annak a programnak az elindítása, amellyel leggyakrabban dolgozunk. Ez többnyire valamelyik DOS SHELL, vagyis keretrendszer vagy héj, esetleg egy programozási nyelv (mint ahogy a Commodore bekapcsolás után BASIC-ben van). Ezek a programok a gép kezelését megkönnyítik és felgyorsítják, a későbbiekben bővebben is megismerkedünk velük.

Lengyel István

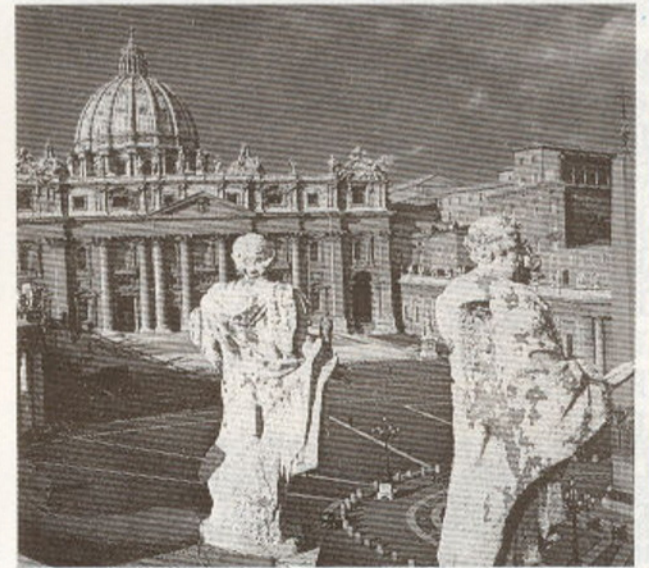
# MAKR VILÁG



Szellemileg és lelkileg

**igényes utasok irodája**

Budapest, Üllői út 11–13. Telefon: 118-36-36



**Országos Commodore Egyesület tagjainak jelentős utazási kedvezmények!**

# Bevezető az Amiga programozásához

Ha rossz dobozt húzunk ki a konzervpiramisból, akkor az egész nagy robajjal összeomlik. A programozók hasonló problémával találják magukat szembe, ha a szükséges információkat rossz helyről veszik ki az operációs rendszerből.

Ha úgy használunk egy funkciót, hogy helyzetét nem határozzuk meg a „piramisban”, akkor a program nem fog működni, viszont ha értjük a kérdéses program felépítését, akkor a programozás hatékonyabb és problémáktól mentesebb lesz.

Relatívan egyszerű gépeknél, mint például a C64, mindennek megvan a fix helye a memóriában, és így a programozó hamar megtanulja, hogy hova kell ugrania, mondjuk egy karakter kiíratásakor.

Nem így az Amigánál. Egy szöveg megjelenítésekor a felhasználónak legtöbbször egy írható felületet kell kreálnia, majd meghatározni a felirat színét, betűtípusát, hosszúságát, stílusát és csak ezután jöhet a printelés.

Az Amiga bonyolultságát hatékonysága okozza. A C64-es egy fix méretű ugrótáblát használ ahhoz, hogy a Kernel rutinokat beléptesse a ROM-ba, ezért nincs mód arra, hogy több ilyen rutint egyszerre futtassunk.

Mivel a programok legtöbb részét fix memóriahelyre kell behívni, párhuzamosan nem tud több programot működtetni.

Még olyan kis hardware változtatások is, mint például a memória 128 Kbyte-ra való növelése, vagy a szövegmező 40-ről 80 karakteresre állítása meg tudja akadályozni a program futtatását.

Ennek kiküszöbölése érdekében a Commodore szakemberei létrehoztak egy új operációs rendszert, ahol nem feltétlenül számít a memória mérete, a programok címezése és helyzete a tárban. Végeredményül a jól megírt programok felül tudnak kerülni a hardware problémákon (memóriabővítés, gyorsabb processzor, új raktározási-fortélyok, különböző input megoldások) software változtatások nélkül.

Az egyik módja, hogy az Amiga elkerüli a fixált méretek alkalmazását, a data struktúrák linkelt listaként való használata.

Minden egyes tagja a linked listának (Node) magába foglalja a következő Node címét, így hát tudjuk, hogy egy Node hol helyezkedik el a memóriában, lényegében az összes őt követő Node címét is tudjuk.

Az Amiga linked listákat használ a rendszer könyvtárak (library) raktározására is, amelyek a ROM-ba való szabvány belépési pontokat tartalmazzák.

Mivel ezeknek a listáknak nincsen meghatározott hosszúságuk, ezért könnyen bővíthetőek. Ahhoz, hogy kiegészítsük ezen listákat, az operációs rendszer számára meg kell határozni egy fix pontot, amelyhez viszonyítva a rendszer és mi is eligazodhatunk a memóriában. Ilyen fix pont a location 4, ismertebb néven AbsExecBase, amely magába foglalja az Exec könyvtár báziscímét. Ez a könyvtár mindig nyitva található és az AbsExecBase alapján meg tudjuk határozni az összes Exec rutin (pl. OpenLibrary, OpenDevice) címét.

Ha már tudjuk a library báziscímét, akkor könnyen bevitethetünk különböző funkciókat az Entry (bemenet) rutinnal.

## Rendszer könyvtárak (library-k)

Az operációs rendszer összes utasítása csoportokba van foglalva. Ezen csoportok közül a legtöbbet használt (pl. dos.library, graphics.library) és a rendszer számára legfontosabb rutinok egy 256 Kbyte-os, úgynevezett Kickstart ROM-ban kaptak helyet. Azok, amelyek ide nem fértek be (pl. diskfont.library, icon.library) a Workbench lemez Libs directory-jában helyezkednek el, és szükség esetén a gép behívja őket a RAM-ba.

Ahhoz, hogy használni tudjunk egy funkciót, meg kell nyitni az őt tartalmazó könyvtárat (kivéve az Exec, mert ezt a rendszer automatikusan kinyitja).

Ekkor megkapjuk a kérdéses library báziscímét, és így egy ugrási táblázat segítségével már könnyen meghatározhatjuk a funkció belépési címét. Mikor végleg befejeztük egy könyvtár használatát, ne felejtsük el „szép illedelmesen” becsukni azt a CloseLibrary funkcióval.

Egy könyvtár megnyitásának, bezárásának assembly és C nyelvű programját láthatjuk az első listán.

A példa szerint először betöltjük az Exec báziscímét az A6-os regiszterbe, a megnyitandó könyvtár nevét (intuition.library) pedig A1-be. Ezután megadjuk a verziószámot (0, ennek a rutinok párhuzamos futásának megteremtésében van nagy feladata), meghívjuk az aktuális könyvtárat megnyitó ROM-rutint. Az új library báziscímét elmentjük (IntuitionBase), majd ellenőrizzük, hogy valóban sikerült megnyitni a könyvtárat.

A bezárás pont fordítva történik. Először A1-be betöltjük a library báziscímét és ezután meghívjuk a bezáró rutint.

## Device-ok

A device-ok tulajdonképpen a library-k egy variációi, amelyek segítenek a különböző rendszerekkel (printer, modem, clipboard) folytatott kommunikációban.

Hasonlóan a könyvtárakhoz, néhányuk szintén a Kickstart ROM-ban található, míg a maradék a Workbench lemez Devs directory-jában leledzik.

A device-ok megnyitása és bezárása hasonló a könyvtárkéhez csak kezelésükben mutatkozik lényeges különbség.

Ahhoz, hogy használhassunk egy device-t, először fel kell állítsunk egy adatblokkot (IORequest), ami egy mutatót tartalmaz az üzenet (Message) portra vonatkozóan. Az IORequest

blokk elkészítése után ki tudjuk nyitni az aktuális device-t, az OpenDevice Exec funkcióval.

A második listán egy device megnyitására láthatunk példát C nyelven.

Miután felállítottuk az IORequest-et és kinyitottuk a device-t, elkezdhetjük a kommunikációt, a DoIO, vagy a SendIO Exec rutinokkal.

Meg kell jegyezni, hogy a Lattice standard Amiga funkciók könyvtára (amiga.lib) rendelkezik több support (segély) funkcióval (CreatePort, CreateExtIO), amelyek a számunkra szükséges adatsruktúrák felállításában nyújtanak segítséget.

## Resources

Az Exec tartalmaz még egy linked listás információhalmozatot, amelyek különleges kommunikációs és egyéb funkciókat biztosítanak. Ezek összefoglaló neve resources (kiegészítések).

Ahhoz, hogy használhassuk az ezek által biztosított rutinokat, meg kell őket nyitni az Exec OpenResource nevű funkciójával.

## Include file-ok

A legtöbb ROM Kernel funkció meghívása előtt sok adatot kell beállítani. Például egy ablak kinyitása előtt többek között meg kell adni az aktuális screen-t, az ablak legkisebb és legnagyobb méretét, felíratát és még rengeteg mindent.

Az Amiga rendszere ezeket az információkat adatstruktúráként tárolja, amelyek kompozíciója az általunk használt fordító include file-jaiban található. Ilyen include file-okat az összes C

és a legtöbb assembly (Devpac, MetaComCo) fordítót használ. Ha nincsen ilyen, (Seka) akkor manuálisan kell megszerkeszteni őket.

Ezek a file-ok data blokkok receptjeit tartalmazzák, melyek megadják a felhasználó és a fordító számára a szükséges adatok milyenségét.

Ha megértettük rendszerüket, ilyen blokkok készítésére mi is képesek vagyunk, ezzel megkönnyítve a programozást. Hasonló, nem standard, bár az include készletben lévő adatblokk a NewWindow is, amely tulajdonképpen a Window struktúrára épül, a legtöbb helyzetben helyettesítve azt.

Az include-ok tanulmányozása azért is hasznos, mert segítségével megérthetjük, hogy hogyan működnek, hogyan kapcsolódnak egymásba az adatblokkot használó rendszerrutinok.

A harmadik listán az IntuiText struktúrát láthatjuk.

Az hogy tudjuk milyen információk kellenek az IntuiText-hez segít megérteni az olyan rutinokat, mint például a PrintText, amely az IntuiText struktúrát inputjaként kezeli.

Más adatstruktúrákban is, csak úgy, mint a linked listákban és végig az egész Amiga rendszerben hasonló kapcsolatokkal gyakran találkozhatunk.

Az Amiga programozása látszólag nagyon bonyolult, de azonnal megérthetjük, ha rájövünk milyen filozófia rejlik a gép belsejében.

Az Amiga bonyolult multitasking rendszere konzervpiramisszerű képződmény. A kezdőknek először a tetején érdemes matatni, minél tapasztaltabbak vagyunk, annál mélyebbre merülhetünk, de vigyázva, mert egy rossz mozdulat és az egész piramis összeomlik.

TPE

## Library-k

Az 1.3-as rendszer ROM-jából és a Workbench lemezről.

NÉV	FUNKCIÓ
diskfont	Rutinokat tartalmaz a disk bázisú szöveg-fontok kezelésére.
dos	A lemezekkel való különböző manipulációkra szolgál.
exec	Task-k, megszakítások, I/O rutinok irányítója, az operációs rendszer gerince.
expansion	A bővítő, alakító funkció rendszerbe való integrálódását segíti.
graphics	Grafikai, animációs funkciókat tartalmaz.
icon	A Workbench ikonok megjelenítését segíti.
intuition	Screenek, ablakok, menük... kezelésére és irányítására szolgál. A legtöbbet használt library.
layers	A grafikai funkciókra épül. A képekkel és ablakokkal való különleges feladatokra szolgál.
mathffp	Lebegőpontos számolási rutinokat tartalmaz.
mathieedoubbas	Duplapontosságú IEEE matematikai rutinok.
mathieedoubtarns	Duplapontosságú IEEE, transzcendális matematikai rutinokat tartalmaz.
mathtarns	Lebegőpontos transzcendális rutinok.
translator	Angol szövegstringek fonetikussá való alakítására szolgál.

**Device-ok**

Az 1.3-as rendszer ROM-jából és a Worbench lemezről.

NÉV	FUNKCIÓ
audio	Az Amiga négy hangcsatornájáért felelős.
console	A billentyűzet leolvasásában használatos, a beolvasott adatokat továbbítja az ablakba és mozgatja a kurzort.
gameport	A két botkormány portot ellenőrzi és hírt ad az esetleges történekekről.
input	A gameport és a keyboard device üzeneteit egy csatornára tereli.
keyboard	A billentyűzeten bekövetkező változásokról ad hírt.
narrator	Fonetikus szöveget beszéd formájában kiküldi a hangcsatornákra.
parallel	A párhuzamos porttal végez I/O kommunikációt.
printer	A serial és parallel device-val kommunikál a preferences adatai szerint.
serial	A soros porttal végez I/O kommunikációt.
timer	Task signálok bevezetésére szolgál, a CIA 8520-as chipen keresztül.
trackdisk	Alacsonyszintű drive kezelő rutinokat tartalmaz.

**Resource**

Az 1.3-as rendszer ROM-jából és a Worbench lemezről.

NÉV	FUNKCIÓ
cia	Átmeneti lehetőséget biztosít a két CIA 8520-as chip között.
disk	Különleges kontrollt tesz lehetővé a meghajtók fölött.
misc	Különleges hozzáférést biztosít a soros és párhuzamos portok fölött.
potgo	Az egér, botkormány, fényceruza... között kiosztja a POTGO regisztereket.

**Könyvtár megnyitása assembly-ben**

```
ExecBase:      equ      4           ; Mutatók beállítása
OpenLibrary:  equ      -$228        ;
CloseLibrary: equ      -$19e        ;
```

```
    move.l     ExecBase,a6          ; ExecBase betöltése
    lea       IntuitionName,a1     ; Library név betöltése
    clr.l     d0                    ;
    jsr       OpenLibrary(a6)      ; Library nyitás
    move.l    d0, IntuitionBase     ; Báziscím elmentése
    beq       error                ; Vizsgálat

    move.l    IntuitionBase,a1     ; Báziscím betöltése
    jsr       CloseLibrary(a6)     ; Library zárás
    rts                                     ; Visszatérés
```

```
error: illegal                ; Kizárásos kilépés
```

```
IntuitionName: dc.b  „intuition.library”, 0
IntuitionBase: dc.l 0
```

## Library nyitás C-ben

```

/* A szükséges include behívása */
#include <intuition/intuition.h>

/* Mutató az IntuitionBase struktúrára */
struct IntuitionBase *IntuitionBase;
main<>
{
    /* Library nyitás */
    IntuitionBase=OpenLibrary („intuition.library“,0);

    /* Kizárásos kilépés */
    if (IntuitionBase==0) exit <>;

    /* Library zárás */
    CloseLibrary (IntuitionBase);
}

```

## Az IntuiText struktúra

```

struct IntuiText
{
    UBYTE   FrontPen, BackPen;
    UBYTE   DrawMode;
    SHORT   LeftEdge;
    SHORT   TopEdge;

    struct   TextAttr *ITextFont;

    UBYTE   *IText;

    struct   IntuiText *NextText;
};

```

# NOVOTRADE

## Commodore-termékek fogyasztói árlistája

Commodore 64	14 600 Ft
1541 floppy	16 600 Ft
Magnó	3 000 Ft
1802 monitor	25 000 Ft
Nyomtató	22 500 Ft
Használt zöld mono monitor	4 500 Ft

**Áraink az ÁFA-t tartalmazzák!**

## NOVOTRADE 2C Áruház

Budapest XIII., Balzac utca 35.

# Egér kontra botkormány

A joystick és a játék két, egymástól elválaszthatatlan fogalom. De a joystickok többre is képesek. Most szeretnénk néhány alternatívát és komoly felhasználási területet is bemutatni.

Mindenekelőtt a játékbárátok azok, akik nem nélkülözhetik a joystickot. Vajon van egyáltalán a világon olyan játékos, aki el tudná képzeni az életét enélkül az eszköz nélkül? Bizonytalán nincsen. De vajon nem lehet a botkormányt egyéb hasznos dologra is fölhasználni? Most be is mutatunk egy ilyen felhasználást.

## A működési elv

A számítógép oldaláról nézve a joystick öt különböző kapcsolási állapotot tud ábrázolni (a négy „égtáj” és a tűzgomb). Az alapállapotban ezek a kapcsolók úgy viselkednek, mintha ott sem lennének. A bemeneti vezeték high potenciálon van, amelyet egy ellenálláson keresztül kap a tápfeszültségből. A joystick mozgásával ezt a vezetékét a földdel zárjuk rövidre, mire a C64-es a CIA1 programozásának megfelelően reagál. Egy meghatározott tárolócímet egy definiált értékre állítunk be, amit a programozó a saját rutinjával lekérdezhet. Ezzel a joystick elvét gyakorlatilag máris elmondtuk. Az idők során különböző technikák fejlődtek ki, a fönti eljárás komfortos megvalósítására (kapcsolástechnika). A legjobban a mikrokapcsolók és a fémyelvezés érintkezők váltak be. Az ilyen „belsősegekkel” ellátott joystickok azok, amelyek tartósak és hosszú életűek.

Vannak persze más joystickok is, azaz olyanok, amelyek más technikát alkalmaznak. Ezeket a változatokat is bemutatjuk.

Egy ideje kaphatók a piacon olyan joystickok, amelyekben stabil fémgolyós kapcsoló található. A fogantyú meghosszabbításánál található egy masszív fémgolyó, amelyet a kar mozgásánál a beépített fémfal egyikének nyomunk. A korábban említett rövidzárlatot ezzel hozzuk létre.

Teljesen más technikát használnak a higanykapcsolós rendszerek. Ezek belsejében négy, nem egészen vízszintesen elhelyezett csövecské van. Ezekben két kontaktus található, amelyeket a csőben lévő higanycsöpecské köthet össze.

Érdekes változat az infravörös fényvel történő jelátviteli technika is. Ennél az eljárásnál nem kell kábelt alkalmazni. Sajnos azonban ez a technika nem igazán tudott elterjedni, mivel igen érzékeny a zavaró fényekre. Ugyanakkor a kezelés sem igazán biztonságos (nagyobb távolságoknál problémát jelenthet a célkészülék beállítása).

A legtöbb joystick manapság már rendelkezik a folyamatos tüzelést biztosító kapcsolóval. Ez a gyors tüzelést igénylő játékoknál jelent nagy előnyt. A tűzgomb gyors lenyomását és elengedését itt az elektronika szimulálja helyettünk. Némely botkormánynál még arra is lehetőségünk van, hogy szabályozzuk a tűzgyorsaságot, vagy esetleg teljesen letiltjuk azt.

## Alternatívaként: a trackballok

Az egerektől eltekintve a trackball jelent a joystick mellett egy elfogadható alternatívát. A trackball esetében egy olyan go-

lyóról van szó, amelyet beépítettek egy stabil házba. Ha ezt a golyót mozgatjuk, akkor azzal különböző jeleket hozhatunk létre, amelyeket azután a gépünk a joystickhoz hasonlóan dolgozhat föl. Elméletileg a trackball egy fejreállított egér. A fölépítésük szerint kétféle trackball létezik. Az egyik egy joystickot szimulál, a másik egy egeret utánoz. Programozástechnikailag nézve az előbbi azt jelenti, hogy csak a már ismert kilenc irányt tudjuk megjeleníteni (előre, hátra, jobbra, balra, a négy átló és a tűzgomb). A trackballok másik fajtája valamelyest változatosabb működésre képes. Azok az X/Y tengely elvén működnek. Ehhez pedig az összes lehetséges golyópozícióhoz tartozik egy meghatározott jel. A lekérdezés ebben az esetben nem az említett 9 lehetőségre irányul, hanem a megfelelő koordinátákra. Ezek az értékek úgy tekinthetők, mintha a golyóra egy X/Y koordináta rendszert fektetnénk. Mindezekkel a lekérdezési változatoságnál több lehetőségünk van a normál joystickhoz képest. Mialatt például egy grafikus programnál a joystickkal csupán 45 fokos szögben rajzolhatunk vonalat, úgy a második fajta trackballal tetszőleges szögeket használhatunk. Igaz, meg kell említenünk egy fontos véleményt is: ezeknek az igazán pontosan dolgozó trackballoknak az alkalmazási területe jóval túlmutat a C64-es lehetőségein. Ezeket például radarállomásokban használják, ahol a 9 lehetséges lekérdezés kevés lenne. A C64-re írt ilyen felhasználói program igencsak szokatlan megoldás lenne. Ettől függetlenül vannak ilyen trackballok a C64-hez.

## Menüvezérlés joystickkel

Igen kevés programozó veszi a fáradságot, hogy átgondolja a joystick komolyabb felhasználását is. Hisz semmi sem szól például az ellen, hogy a játékokban, grafikus programokban operációs rendszerekben megszokott menüvezérlést akár saját programjainkban is alkalmazzuk. Nagyobb számítógépek esetében, mint amilyen például a Commodore Amiga is, a felhasználót az egér segíti eligazodni a komplex feladatok között. Hogy ez a C64-nél is lehetséges, azt a GEOS, ez a grafikus operációs rendszerbővítés is bizonyítja. Tehát hasonló komfortot a C64-gyel is el lehet érni akkor is, ha mondjuk a számunkra nem éri meg egy egér vétele. Lehetséges jól működő, a joystickot lekérdező programot BASIC-ben is írni. Például:

```
10 PO=56320:KEP=1024:SZIN=54272
20 ME=PEEK (KEP):POKE,KEP,160:POKE KEP+SZIN,
PEEK (646)
30 JS=PEEK (PO):MOZGAS=0
40 IF (JS AND 1)=0 THEN MOZGAS=MOZGAS-40
50 IF (JS AND 2)=0 THEN MOZGAS=MOZGAS+40
60 IF (JS AND 4)=0 THEN MOZGAS=MOZGAS-1
70 IF (JS AND 8)=0 THEN MOZGAS=MOZGAS+1
80 IF (JS AND 16)=0 THEN POKE KEP,ME:END
90 IF KEP+MOZGAS<1024 OR KEP+MOZGAS>2023
THEN MOZGAS=0
100 POKE KEP,ME:KEP=KEP+MOZGAS:GOTO 20
```

Ezzel a programmal a beszínezett kurzort vezéreljük a képernyőn keresztül. A joystickot a 2-es portba kell csatlakoztatni. De nézzük meg egy kicsit közelebbről a fönti programot.

Az a cím, ahol a joystickból érkező jelet lekérdezzük, az 1-es porthoz az 56231, a programunk által használt 2-es porthoz 56320. Ezt az értéket rakjuk le a PO változóba. A képernyő tárolójának kezdőcíme az 1024, ezt rejti a KEP változó. A tartomány végcíme a 2023. Az aktuális képernyői pont elhelyezéséhez az aktuális szint is be kell állítani a megfelelő tárolóhelyre (20. sor, a képpont plusz 54272). A program RUN-nal történő elindítása után a kurzor a bal felső sarokban van. (A videó tároló kezdete.) Ha esetleg valami ott állna, akkor azt nem szabad fölülírni, ezért azt megjegyezzük az ME-ben (20. sor). Ezután a mi vezérelendő jelünket, melynek ASCII értéke 160, erre a helyre visszük. Az aktuális szint az operációs rendszer a 646-os címen tárolja. A 30. sorban kezdődik a tulajdonképpeni joystick lekérdezés. Ehhez előbb ki kell olvasni az információs regisztert. Az értéket a JS-be visszük. A MOZGAS nevű változó testesíti meg a mozgások összességét. Ezt persze a lekérdezés előtt előbb nullázni kell. Az átvitt értékből a számunkra csupán az első bit érdekes, amelyeket az AND logikai operátorral kiszűrünk. A joystick által aktivált bit értéke 0, a nyugalmi állapot jelzése az 1. Ahhoz, hogy a képernyőn valamilyen mozgást ábrázolhassunk fölfelé, le kell vonni a jel pozíciójából 40-et. Egy sorral följebb a JS első bitje viszi a kurzort. Ezt a lépést a 40. sorban kérdezzük le. A lefelé irányuló mozgást a második bit jelzi (50. sor), jobbra a harmadik, balra a negyedik bit a mérvadó. Ezek egy levonását illetve hozzáadását vonják maguk után az adott pozíciót tekintve (60. és 70. sor). A programban az átlós mozgásokat is lekérdezzük, ez akkor is így van, ha ez az első pillanatban nem látszik. A balra lefelé történő mozgás eredményeként például az IF lekérdezés után +39-et kapunk. A 80. sorban az utolsó lekérdezés a tűzgombra vonatkozik. Ha azt megnyomjuk, akkor visszaállítjuk a bal felső sarokban található jelet, és a programot befejezzük. A 90. sorban még azt vizsgáljuk meg, hogy a mozgás benn van-e

a megengedett keretben. Ezután a program a fölülírt jelet visszaviszi az eredeti helyére, majd visszaugrik a 20. sorra. Ezzel a BASIC-ból történő joystick lekérdezést szemléletesen demonstrálhatjuk.

## Mire figyeljünk a vételnél?

De melyik modellt vegyük meg, ha már egyszer szándékunkban áll pénzt kiadni? Milyen kritériumokat jó megfogalmazni ehhez?

A kiválasztási kritériumok egyik legfontosabb pontja a marcolhatóság. A rúdnak jól kell a kézben feküdnie, a joysticknek pedig jól vezethetőnek kell lenni. Másik ilyen szempont az az út, melyet a karnak meg kell tennie addig, amíg a kívánt kontaktus lezárul. Igencsak ajánlható a stabil ház, ha az eszköz erős megterhelésnek lesz kitéve. Ezt hamar megtanuljuk majd becsülni! Ha a botkormányt állandó helyen akarjuk használni, akkor az öntapadó lábacsokról sem szabad lemondanunk. További szerepet játszik természetesen az ár. A komfortnak persze itt is ára van. De az is igaz, hogy megéri egyszer mélyen belenyúlni a zsebbe, semmint négyhetente egyszer újabb olcsójánost beszerezni. A számunkra optimális joystick vétele előtt teszteljük le a kérdéses eszközt egy általunk választott programmal. Ezt a kérést egyetlen kereskedő sem tagadhatja meg! Tulajdonképpen csakis ekkor mondható el igazán, hogy meghozhatjuk a számunkra minden bizonnyal fontos döntést, ha nem is az adott márkát illetően, de legalább a konstrukciót tekintve. Az NSZK-ban kapható készülékek nagy száma miatt bizonyos, hogy megtaláljuk a nekünk valót.

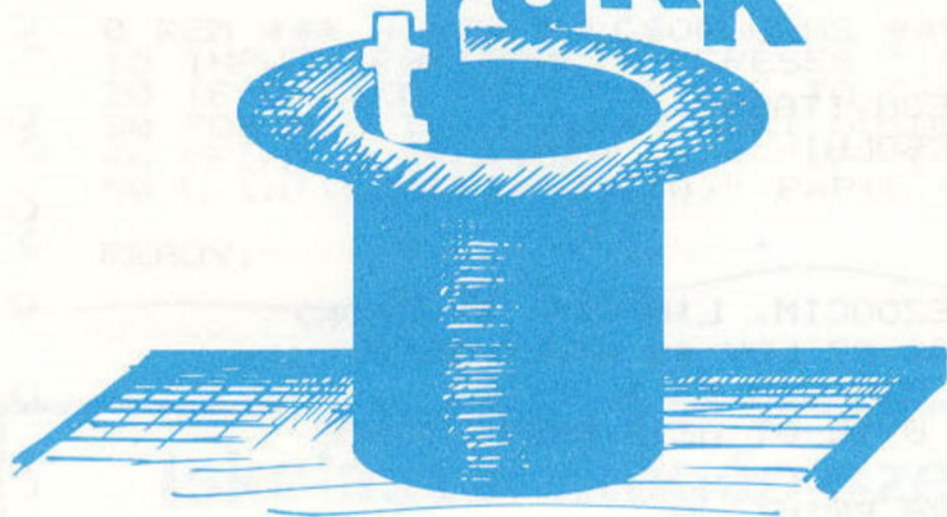
Szolnoki Béla

## Az Országos Commodore Egyesület szolgáltatásai:

C64-be átkapcsolható új operációs rendszer (Speed) + reset beépítése:	2000 Ft	Epromégető (2716-tól 27256-ig)	4300 Ft
1541 kompatibilis lemezegységbe Speeddos beépítése (átkapcsolhatóan) 40 TRACK (+ 85 blokk/lemezoldal), valamint párhuzamos 15 pólusú Canon csatlakozó beépítése:	2000 Ft	8—16 Kbyte-os epromkártya (cartridge, eprom nélkül)	600 Ft
C64 USER-port 1541-es lemezegység összekötő párhuzamos kábel:	1300 Ft	C64-hez tároló oszcilloszkóp (párhuzamos kábel nélkül)	7500 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése:	700 Ft	A háttértárhoz epromok programozása (kész programok, vagy saját, hozott programok beégetésével) egységesen:	500 Ft
PAGEFOX magyar ékezetes szövegszerkesztővel rendelkező cartridge:		C64 bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
(Epson típusú nyomtató min. 640 képpontos szükséges a nyomtatáshoz)	7500 Ft	C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
FASTLOAD (lemezes gyorstöltő, másoló, monitor)	1400 Ft		
TTL IC-TESTER cartridge + program	4300 Ft	A fenti bővítések megrendelhetők levélben, vagy az OCE irodájában személyesen, minden páratlan héten, csütörtökön 17—18 óra között.	
288/256 Kbyte-os eprombank (vezérlő eprommal)	4700 Ft	Árainkat az alkatrészarak változásai befolyásolhatják.	



# Tippek trükkök



## DIRECTORY RUTIN

Számos esetben van szükség arra, hogy programunkba egy, a lemezegységben lévő adathordozó tartalomjegyzékét kiíró rutin kerüljön. Mi most egy erősen optimalizált rövid programot

adunk, amelyet könnyen fölhasználhatunk. A program a 64 byte hosszú SS-Directory.

Hogy a programot használhassuk, a megadott (\$C000) területre kell tölteni. Ez a cím lesz a kezdőcím is, amelyet a KC változóban rögzítünk. Ha ezek után meg akarjuk nézni a tartalomjegyzéket, az alábbi parancsot kell kiadni: SYS KC "\$:keresosztring",KSZ

A keresosztring megfelel a directorynál megszokott sztringnek, ahol a jokerek az eredetinek megfelelően használhatók. A "\$:\* =p" például az összes programfile-t adja meg. A KSZ a készülékszámot jelenti. A directory kijelzését a <SHIFT>, a <C=> vagy a <CTRL> megállítja, a <STOP> megszakítja. Ha az olvasás alatt hiba lép föl, a művelet félbeszakítjuk.

Ha a rutint a SYS KC+3 fölhívással indítjuk, akkor a legutolsó beállított paramétereket használjuk. Ezt a beugrási címet kell akkor is használni, ha a rutint gépi kódú programokból akarjuk fölhívni. Ezelőtt a SETFLS (\$FFBA) és a SETNAM (\$FFBD) rutinokat fölhasználva nekünk kell beállítani a készülékcímet és az állománynevet (legtöbbször \$). A rutin 100 százaléig relokálható, azaz gond nélkül a tároló bármely szabad részébe vihető. Az eredeti változat a \$C000-nál kezdődik. Ha egy másik címre (CI) akarjuk tolni, már az alábbi BASIC sorok is megteszik:

```
FOR I=0 TO 63:POKE CI+I,PEEK(49152+I):NEXT:REM SS-Directory
```

A rutin a kernel ROM standard beugrási címeit használja, így megváltoztatott operációs rendszerek (pl. floppygyorsítók) esetén sem kell problémákkal számolnunk. A híváskor a BASIC ROM-nak aktívnak kell lennie. A rutin a nulláslapon csak azokat a tárolókat használja, amelyeket a kernel rutinok.

```

○ 1000 REM *****
○ 1010 REM * *
○ 1020 REM * S S D I R E C T O R Y *
○ 1030 REM * *
○ 1040 REM *****
○ 1050 :
○ 1060 :
○ 1070 REM *** FORRASSZOVEG PROFI-ASS
○ 1080 SYS 8*4096 ; REM VAGY SYS 9*4096
○ 1090 .OPT 00
○ 1100 *=$C000
○ 2000 ;
○ 2010 ;*** OPERACIOS RENDSZER RUTINOK
○ 2020 ;
○ 2030 TALK = $FFB4 ; KESZULEK A SOROS BUSZON KULDESRE
○ 2040 TKSA = $FF96 ; MASODLAGOS CIMET A TALK-RA
○ 2050 BUSIN = $FFA5 ; KARAKTERBEOLVASAS A BUS-ROL
○ 2060 CHROUT = $FFD2 ; KARAKTER IRAS (A KEPERNYORE)
○ 2070 OPEN1 = $F3D5 ; FILE NYITAS A SOROS BUSZON
○ 2080 CLOSE1 = $F642 ; FILE ZARAS A SOROS BUSZON
○ 2090 LINPRT = $BDCD ; AZ X/A-BAN ALLO SZAMOT DECIMALISAN KIADNI
○ 2100 PRTSPC = $AB3F ; SZOKOZ KIADAS
○ 2110 CRDO = $AAD7 ; KOCSIVISSZA VEGREHAJTAS
○ 2220 ILLERR = $B248 ; ILLEGAL QUANTITY HIBAJELZES KIADASA
○ 2230 ;
○ 2240 ;*** A SZUKSEGES CIMEK ***
○ 2250 ;
○ 2260 SEKADR = 185 ; AKTUALIS MASODLAGOS CIM
○

```

```

2270 GERADR = 186 ; AKTUALIS KESZULEKSZAM
2280 ST      = 144 ; A SOROS PORTOK STATUSZA
2290 STOP   = 145 ; A STOP GOMB CIME
2300 SHIFT  = 653 ; A SHIFT/C=/CTRL GOMBOK CIME
3000 ;
3010 ;*** A DIREKTORI RUTIN ***
3020 ;*** RELLOKALHATO ***
3060 JSR 57812 ; FILENEV BEOLVASASA
3110 LDY #0 ; A MASODLAGOS CIMET
3120 STY SEKADR ; OLVASASRA KAPCSOLNI
3130 JSR OPEN1 ; A (DIRECTORY) FILE MEGNYITASA
3140 LDA GERADR ; A FLOPPYT ADASRA KAPCSOLNI
3150 JSR TALK
3160 LDA #96
3170 JSR TKSA
3180 LDY #5 ; 6 BYTE (KEZDESNEI, KEZDOCIM, LINKCIM, BLOKKOK)
3190 .BYT #2C ; 'ALCAZOTT' BIT PARANCS AZ LDY #3 ATUGRASARA
3200 L1 LDY #3 ; 4 BYTE (KESOBB, CSAK LINKCIM ES BLOKKOK)
3210 L5 TAX ; AZ UTOLSO BEOLVASOTT BYTE-OT AZ X REGISZTERBE
3220 JSR BUSIN ; UJ BYTE BEOLVASASA
3230 DEY ; MAR 6/4 BYTE BEOLVASVA PRINT
3240 BPL L5 ; HA NEM, UJRA L5-RE
3250 LDA ST ; A STATUSZ MEG OKE PRINT
3260 BNE L4 ; HA NEM, L4-RE
3270 JSR LINPRT ; BLOKKOK KIADASA (AZ X/A-BAN VANNAK!)
3280 JSR PRTSPC ; URES LEPES KIADASA
3290 L2 JSR BUSIN ; BYTE BEOLVASASA
3300 BEQ L3 ; EGYENLO 0 (=A BEJEGYZES VEGEPRINT), HA IGEN L3-RA
3310 LDX ST ; STATUSZ MEG OKE PRINT
3320 BNE L3 ; HA NEM, L3-RA
3330 JSR CHROUT ; KARAKTER KIADASA
3340 BNE L2 ; ES MINDJART L2-RE, MIVEL A KARAKTER MINDEN ESETBEN <> 0
3350 L3 JSR CRDO ; KOCSIVISSZA VEGREHAJTAS
3360 BCC L1 ; ISMET L1-RE MIVEL CRDO UTAN A CARRY TOROLVE
3370 L4 JMP CLOSE1 ; FILE ZARAS ES VEGE

```

## KÉPERNYŐTÖRLÉS MÁSHOGY

A C64 képernyőtörlésének nem kell feltétlenül unalmasnak lennie. Tegyük a dolgot érdekesebbé. Miután beírtuk és elmentettük a listát, futtassuk le. Ha a 40-es sort kiemeljük a programból, a képernyőtörlés önálló szubrutinként is használható SYS49152-vel.

```

0 REM *** KEPERNYO TORLES - MASHOGY ! ***
10 PRINT CHR$(147):FOR T=1 TO 1000:PRINT ".":NEXT
20 FOR I=49152 TO 49274:READ Z:POKE I,Z:CK=CK+Z:NEXT
30 IF CK<>13883 THEN PRINT "HIBA AZ ADATSORBAN...!":END
40 SYS 49152
50 DATA 162,0,160,41,169,64,133,251,169,32,157,0,4,153,40,4,157,80,4
60 DATA 153,120,4,157,160,4,153,200,4,157,240,4,153,24,5,157,64,5,153
70 DATA 104,5,157,144,5,153,184,5,157,224,5,153,8,6,157,48,6,153,88,6
80 DATA 157,128,6,153,168,6,157,208,6,153,248,6,157,32,7,153,72,7,157
90 DATA 112,7,153,152,7,157,192,7,232,136,32,179,238,32,179,238,32,179
100 DATA 238,32,179,238,32,179,238,32,179,238,32,179,238,32,179,238,32
110 DATA 179,238,32,179,238,192,0,208,143,96,153

```

READY.

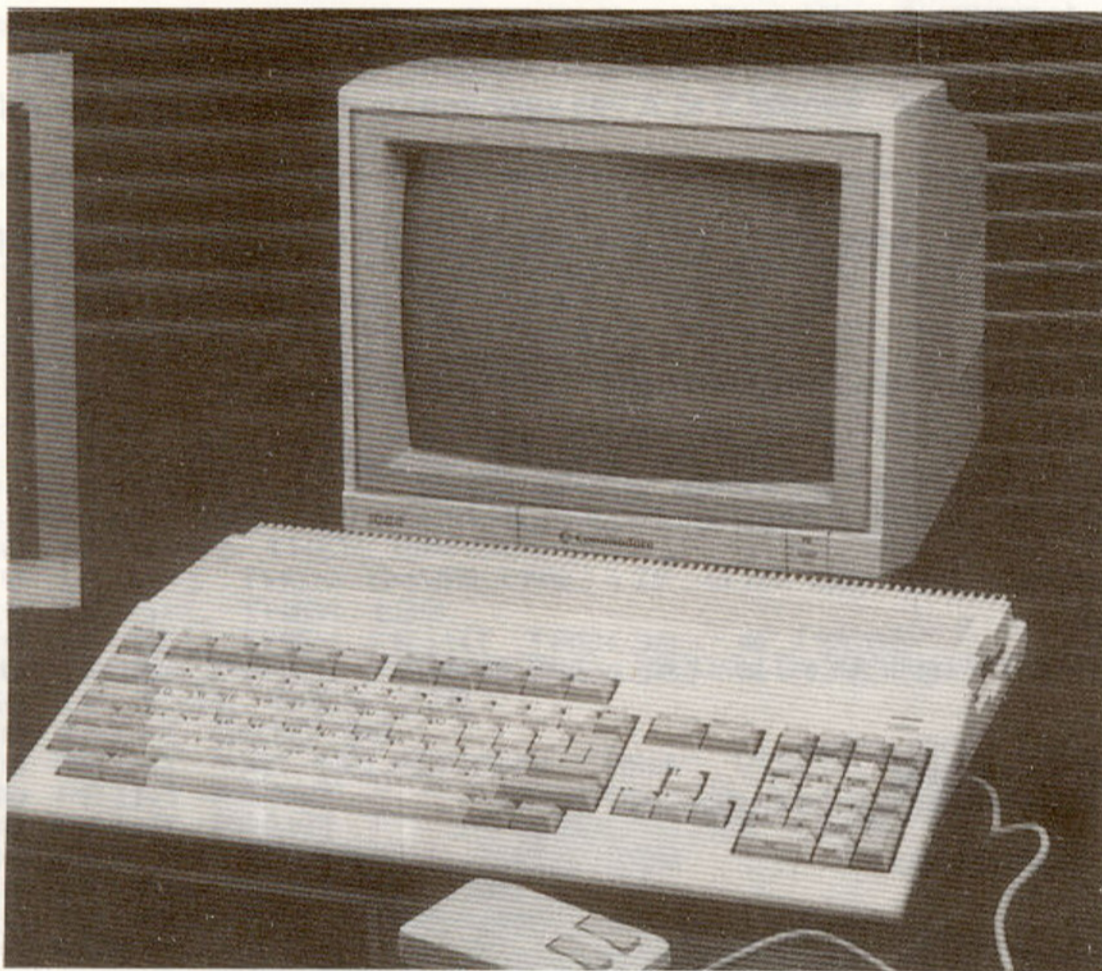
# NINCS TÚLCSORDULÁS

Aki már próbált matematikai képleteket kiszámítani C64-esén, bizonyára találkozott a gép korlátaival. Például a legnagyobb szám, melynek faktoriálisát kiszámíthatjuk a 33, míg a legtöbb zsebszámológépen ez az érték 69. Ez a program a problémát tízes alapú logaritmus alkalmazásával kerüli meg. A legnagyobb kiszámítható faktoriális 5109! Persze minél nagyobb a szám, annál tovább tart a számolás.

```

0 REM *** NINCS TULCSORDULAS ***
10 INPUT "FAKTORALIS KERESÉS ";N:F=1:P=0:C=1
20 IF C+11<N THEN FOR X=C TO C+9:F=F*X:NEXT:C=C+10:GOSUB 50:GOTO 20
30 FOR X=C TO N:F=F*X:NEXT:GOSUB 50
40 PRINT N;"FAKTORALIS = ";F;"E";P:GOTO 10
50 L=INT(LOG(F)/LOG(10)):P=P+L:F=F/10^L:PRINT L:RETURN
READY.
  
```

## Iskolaszámítógép-szerviz és Kereskedelmi Bt.



### PROGRAMOK ÁRUSÍTÁSA ÉS MENEDZSELÉSE

**AMIGA** és **Commodore** számítógépek javítása és eladása

Átalánydíjas javítás kedvező áron

C16 bővítés 64 kbyte-ra

PC ÁRUSÍTÁS

Csúcstechnológiájú PANASONIC telefonok és telefonrendszerek



1088 Budapest, Rákóczi út 26.  
Telefon: 1-182-972,  
1-381-139  
Telefax: 1-182-972

6000 Kecskemét,  
Március 15. u. 14.  
Telefon: 06/76/47-626

# C64 KARAKTER EDITOR

Ez a program lehetővé teszi a C64-es mind az 512 karakterének tetszőleges módosítását. Elindítás után egy gépi kódú rutin áthelyezi a jelkészletet a RAM-ba a \$ 3000—\$ 4000 területre. Ezután bejelentkezik a program menüje. <F1>-el a program segítségével készített (és lemezre mentett) saját elképzelésű karakterkészletet lehet betölteni, amennyiben tovább szeretnénk módosítani. <F7>-tel az elkészült karakterkészlet lemezre menthető .£ névvel. (Ha ugyanazon a lemezen több karakterkészletet akarunk tárolni, a program ismételt használata előtt a már meglévőt át kell nevezni, mert a program a .£ nevű file-t felülírja!)

<F3>-mal a \$ 3000—\$ 4000 területen lévő karakterkészlet módosítható. A billentyű lenyomása után kiíródik a képernyőre a 256 karakter. A <SHIFT> + <C=> billentyűkkel ekkor átkapcsolhatunk a másik 256 karakterre, ily módon elérhető a teljes

készlet. A kurzormozgató billentyűk és a <RETURN> által kiválasztott karakter kinagyítva megjelenik, és a ., \*, illetve a kurzormozgatók segítségével átszerkeszthető. Átszerkesztés után a <RETURN> lenyomásával visszatérünk a 256 karakter közé a következő módosítandó kiválasztásához. Ez kb. 2 másodpercet vesz igénybe, mivel a karakter módosítása és a karakterkészlet újbóli kiírása BASIC-ben történik. A kívánt karakterek módosítása után a <←> billentyűvel visszatérhetünk a menübe.

Ha a program segítségével elkészített karakterkészletet saját programban szeretnénk használni, akkor az új karakterkészlet ,8,1-el történő betöltése után a poke 53272,29 vagy poke 53272,31 paranccsal kapcsolhatunk át a saját tervezésű jelekre. Visszakapcsolás a poke 53272,21 vagy poke 53272,23 attól függően, hogy a NAGYBETŰ/GRAFIKUS JEL, illetve a kisbetű/NAGYBETŰ készletet akarjuk-e használni. A „karaktertervezéshez” sok siker kíván:

Maurer Péter

```

0 REM *****
1 REM * C= UJSAG          SORSZAM: 340 *
2 REM * C64 KARAKTERTERVEZO *
3 REM * PROGRAM : MAURER PETER *
4 REM *****
9 IFL=1THEN34
10 POKE646,0:POKE53281,11:POKE53280,11
11 DATA 169,0,133,251,133,253,169,208,133,252
12 DATA 169,48,133,254,120,169,51,133,1,160
13 DATA 0,177,251,145,253,200,208,249,230,252
14 DATA 230,254,165,252,201,224,208,239,169,55
15 DATA 133,1,88,169,28,141,24,208,96
20 FORZ=49152TO49200
21 READ A:POKEZ,A:S=S+A:NEXT
22 IF SC>7789 THEN PRINT"HIRA A CHAR.COPY ADATAIBAN":END
23 SYS 49152
30 PRINT"J":POKE211,24:POKE214,24:SYS58732:PRINT"##(C) PETIKA 1990"
31 PRINT"#####IF1 ##KARAKTERKESZLET BETOLTES ALTETEL"
32 PRINT"#####IF3 ##KARAKTERKESZLET MODOSITAS ATALTEL"
33 PRINT"#####IF7 ##KARAKTERKESZLET KIMENTESTEL"
34 B=PEEK(203)-2:IFB<0THEN34
35 ONBGO TO900,400,99:GOTO34
99 PRINT"J"
100 FORI=0TO255:POKE1024+I,I:NEXT
105 POKE211,0:POKE214,23:SYS58732:PRINT"##[+]:SZERK. VETTEL"
110 GOSUB500
111 POKE198,0:WAIT198,1:GETA#
112 IFA#=" "THENIFK<55551THENX=X+1:GOSUB500
113 IFA#=" "THENIFX>0THENX=X-1:GOSUB500
114 IFA#=" "THENIFY<6THENY=Y+1:IFY=6ANDX>15THENY=5
115 IFA#=" "THENGOSUB500
116 IFA#=" "THENIFY>0THENY=Y-1:GOSUB500
117 IFA#=CHR$(13)THEN130
118 IFA#="+ "THEN30
119 GOTO111
130 K=PEEK(209)+256*PEEK(210)+PEEK(211)-1024:POKE1520,K:POKE657,128
131 PRINT"#####"
200 AD=12288+(K-256*(PEEK(53272)=31))*8
201 FORI=0TO7:M(I)=PEEK(AD+I):NEXT
202 FORI=0TO7:FORN=7TO0STEP-1
203 IF(M(I)AND2^n)THENPRINT"##*";:GOTO205
204 PRINT"##.";:
205 NEXTN:PRINT:NEXTI
300 Q=0:W=0:GOSUB600
301 POKE198,0:WAIT198,1:GETA#
302 IFA#=" "THENIFQ<7THENG=Q+1:GOSUB600
303 IFA#=" "THENIFQ>0THENG=Q-1:GOSUB600
304 IFA#=" "THENIFW<15THENW=W+1:GOSUB600
305 IFA#=" "THENIFW>0THENW=W-1:GOSUB600
306 IFA#=CHR$(13)THEN311
307 IFA#="*"THENJ=42:GOSUB700
308 IFA#="."THENJ=46:GOSUB700
309 IFA#="+ "THENPOKE1520,32:POKE55296+K,0:POKEP,7:GOSUB800:GOTO301
310 GOTO301
311 POKE1520,32:POKE55296+K,0:POKEP,7:GOSUB800:GOTO100
400 INPUT"FILE NEV###.£#####":N#
410 L=1:LOAD"+N#,8,1
500 POKE211,X:POKE214,Y:SYS58732
510 K=PEEK(209)+256*PEEK(210)+PEEK(211)+54272
    
```

```

520 POKEK,7:POKEK-1,0:POKEK+1,0:POKEK-40,0:POKEK+40,0:RETURN
600 POKE211,0:POKE214,W:SYS58732
610 P=PEEK(209)+256*PEEK(210)+PEEK(211)+54272
620 POKEP,0:POKEP-1,7:POKEP+1,7:POKEP-40,7:POKEP+40,7:RETURN
700 POKEPEEK(209)+256*PEEK(210)+PEEK(211),J:RETURN
800 FORL=0TO7
810 FORI=0TO7:U(I)=(PEEK(1344+L*40+I)=42)*2↑(7-I):NEXT
820 Z(L)=0:FORI=0TO7:Z(L)=Z(L)-U(I):NEXT:NEXT
830 FORI=0TO7:POKEAD+I,Z(I):NEXT:POKE657,0:RETURN
900 FORM=0TO4:POKE828+M,PEEK(43+M):NEXT
910 POKE43,0:POKE44,48:POKE45,0:POKE46,64
920 SAVE"@: .£",8
930 FORM=0TO4:POKE43+M,PEEK(828+M):NEXT
940 END

READY.
    
```

## BETŰ – SZÁM KITALÁLÓ

Ebben a programban egy betűt vagy egy számot kell kitalálnunk egy adott tartományban. A program futtatásakor tízszer tippelhetünk egy 1 és 100 közötti számra vagy egy A és Z közötti

betűre. A program minden esetben megmondja, hogy az eldugott betű/szám kisebb vagy nagyobb-e mint a tipp. A 20-as sor átírásával változtathatjuk a megengedett tipppek számát (X) és a legnagyobb eldugható számot (N). Így jó kísérleti eszközt kapunk annak vizsgálatára, mi az összefüggés N és a szükséges tipppek száma között.

```

0 REM *** BETU VAGY SZAM KITALALO ***
10 INPUT "BETU VAGY SZAM LEGYEN ? (B/S)";Q$
20 X=10:N=100:T=1:IF Q$="" THEN 10
30 IF Q$="B" THEN N$="BETU":M$="A-Z":N=26:W=65
40 IF Q$<>"B" THEN N$="SZAM":M$="1-"+STR$(N):W=1
50 PRINT "GONDOLTAM EGY "N$"<-RA,-RE>"
60 PRINT " "M$" KOZOTT." :PRINT " "X"SZER PROBALKOZHATSZ"
70 L=INT(RND(0)*N+W):PRINT:PRINT
80 AN$=STR$(L):IF W=65 THEN AN$=" "+CHR$(L)
90 PRINT "MIRE GONDOLTAL";A$:IF A$="" THEN 90
100 IF W=65 THEN G=ASC(A$):IF G<65 OR G>90 THEN 90
110 IF W=1 THEN G=VAL(A$):IF G>N OR G<1 THEN 90
120 IF G=L THEN 180
130 IF G>L THEN PRINT TAB(30)"KISEBB"
140 IF G<L THEN PRINT TAB(30)"NAGYOBB"
150 IF T>=X THEN 170
160 T=T+1:GOTO 90
170 PRINT "A FELADVANY "AN$" VOLT":GOTO 190
180 PRINT:PRINT " NAGYON JO !!!":PRINT:PRINT "KITALALTAD "T"LEPESBOL"
190 PRINT:PRINT " JATSZOL UJRA (I/N)?"
200 GET P$:IF P$="" THEN 200
210 IF P$="N" THEN END
220 IF P$<>"I" THEN 200
230 RUN

READY.
    
```

## BASIC VÁLTOZÓ CÍME

A 71/72-es tárolócellák tartalmazzák az aktuális változó címét. Hogy egy tetszőleges változó kezdőcímét meghatározhassuk, az alábbiakat kell tenni:

```

2000 A$=A$
2010 POKE 180,PEEK(71):POKE 181,PEEK (72)
2020 AD=PEEK(180)+256*PEEK(181)
    
```

A 2000-es sorban előbb aktualizáljuk a változót, mire az operációs rendszer a nevezett címekre írja a változónak a tárolóban elfoglalt helyét. Ezután ezt az értéket kimásoljuk a 180/181-es címekre, majd az értéket az AD változóba rakjuk. A cím kiolvasását azért kell egy köztes tárolón keresztül kezelni, mert az AD fölhívásakor már nem az A\$, hanem az AD cím állna a 71/72-es tárolócellákban!

## BIZTONSÁGI RÁKÉRDEZÉS

A C128-ason bizonyos diszk parancsok kiadásakor a végrehajtás előtt megjelenik az „ARE YOU SURE?” (biztos vagy benne?) kérdés. Az alábbi kis rutin ugyanezt teszi a C64-en vagy a 128-as 64 üzemmódjában a NEW parancs kiadásakor. Gépeljük be a listát, mentsük el lemezre, majd futtassuk. Ezután valahányszor kiadjuk a NEW-t, megjelenik az „Are you sure?” prompt. Az Y kivételével bármelyik billentyű lenyomása után a memória tartalma sértetlen marad. Komoly segítség hebehurgya Basic programozóknak!

```

0 REM *** BIZTONSÁGI RÁKÉRDEZÉS ***
10 FOR J=53093 TO 53245:READ A:POKE J,A:NEXT:SYS53093
20 PRINT CHR$(147)"ARE YOU SURE ?" PROMPT AKTIV."
30 DATA 173,4,3,141,254,207,173,5,3,141,255,207,169,124,141,4,3,169,207
40 DATA 141,5,3,96,173,0,2,201,78,240,3,108,254,207,173,1,2,201,69,208
50 DATA 246,173,2,2,201,87,208,239,169,13,32,210,255,169,65,32,210,255
60 DATA 169,82,32,210,255,169,69,32,210,255,169,32,32,210,255,169,89,32
70 DATA 210,255,169,79,32,210,255,169,85,32,210,255,169,32,32,210,255
80 DATA 169,83,32,210,255,169,85,32,210,255,169,82,32,210,255,169,69,32
90 DATA 210,255,169,63,32,210,255,169,0,133,198,32,228,255,201,0,240
100 DATA 249,24,201,89,208,3,108,254,207,24,169,32,162,0,157,0,2,232,224
110 DATA 81,208,248,24,108,254,207
READY.
```

## TÖMBÖK EGYSZERŰ TÖRLÉSE

Ha sokat programozunk BASIC-ben, könnyen megtörténik, hogy a nem használt tömbök száma megnő. Sajnos azonban nincs mód ezek CLR-rel vagy valami más paranccsal való törlé-

sére. Épp ezt valósítja meg azonban a mellékelt „Array-Clr” rutin. A kis program néhány BASIC sorból áll, amelyek egy adott tömböt törölnek, de a többi változót érintetlenül hagyják.

A rutint az NA\$="xx\$":GOSUB 10000 utasítással kell fölhívni. Az itt közreadott programban néhány, a hatást szemléltető sort is találunk. A tulajdonképpeni tömbtörölő rész a 10000-res sornál kezdődik. A NA\$ változóba annak a tömbnek a nevét kell írni, amelytől meg akarunk szabadulni.

```

1 REM *****
2 REM ***** ARRAY-CLR *****
3 REM *****
4 REM
5 REM KIVALASZTOTT TOMBOKET(!) TOROL
6 REM
7 REM - BEMUTATOPROGRAM -
8 REM
9 REM
10 PRINT"#####"TAB(9)"ARRAY-CLR"
20 PRINTTAB(8)"#####"
30 PRINT:PRINT
40 REM *****
50 DIM A$(10000):NA$="A$"
60 REM ↑↑ ↑↑ ITT IRJUK BE AZ ADOTT VALTOZO NEVET
70 REM *****
75 F=FRE(0):IF F<0 THEN F=F+65536
80 PRINT"SZABAD TAROLO ERTEKE AZ ARRAYCLR MEGHIVASA ELOTT: ";F
90 TI$="000000":GOSUB 10000:T=INT(TI/.6+.5)/100
95 F=FRE(0):IF F<0 THEN F=F+65536
100 PRINT"... ES "T" MASODPERC ELTELTEVEL MAR"F" BYTE SZABAD!"
110 END
140 :
9998 REM 'ARRAY-CLR' JON
9999 :
10000 B1=ASC(NA$+CHR$(0))
10010 B2=ASC(MID$(NA$,2,1)+CHR$(0))
10020 B3=ASC(RIGHT$(NA$,1)+CHR$(0))
```

Folytatás a 32. oldalon



# Plusz egy jelmondat

Helyezze el az alábbi szavakat, betűcsoportokat — hét kivételével — az ábrában! A hét megmaradt szóból állíthatja össze a jelmondatot.

## VÍZSZINTES:

Kétbetűsök: IP, BM, KI, OL, AA, ÓA, TT, SS, EB, DE, NE.

Hárombetűsök: MOS, SÁL, ÁGI, MÁK, ETA, ARA, LÁT, KOM, TEJ, ASA, KOS, ALI, CSÓ, LLL, LOE, MEG.

Négybetűsök: ZOLA, FÓKA, ILLÓ, TEVE, KIAD, TRIÓ.

Ötbetűsök: NEDZS, CSIGA, MORES, AZ IRÓ, KORÁN, AKADÓ, HÁROM, CSILI.

Hatbetűsök: BAGOLY, UNOKÁM (fordítva).

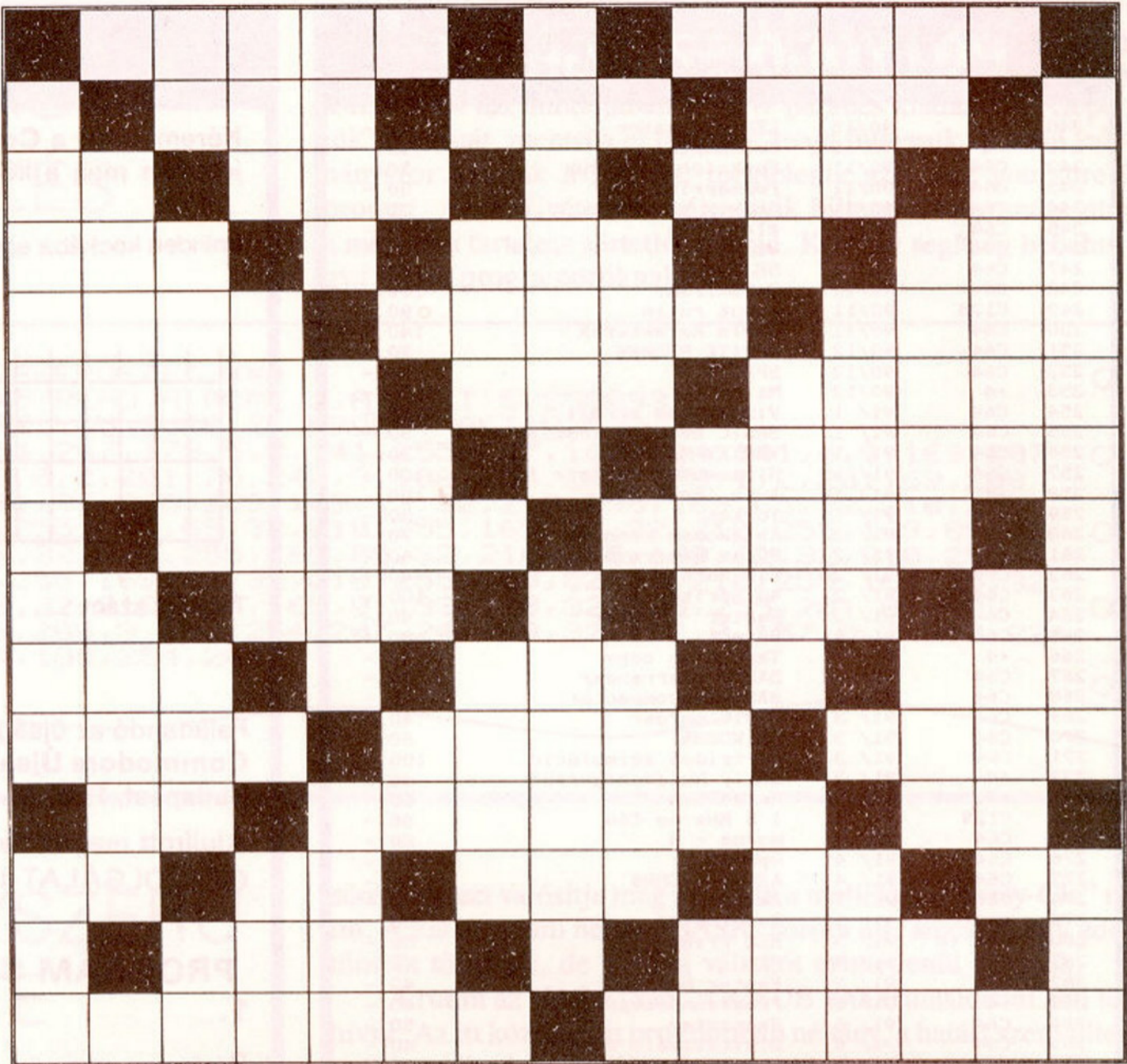
Hétbetűsök: CSARNOK, HALLJÁK, SKORPIÓ, NÓTASZÓ.

## FÜGGŐLEGES:

Kétbetűsök: EM, IS, KO, CS, DI, IO, TÓ, GI, OS.

Hárombetűsök: TAS, ILA, DOM, SIR, YDA, LOT, TEJ, MÁK, IGA, BIA, LEO, ETO.

Négybetűsök: ZSÁK, CSAP, CSAL, ÖLNE, SÁTA, CSAK, KERT.



Ötbetűsök: PRÓZA, ILLAT, ALIGA, A RÓKA, OLASZ, KÁROS, FOKOS, KOROM, BOKÁN, OLVAD.

Hatbetűs: ZSÍROS.

Hétbetűsök: SZAVUNK, NÁZÁRET.

Tízbetűsök: KI AZ A BAKAI, HALLGASSÁK, EMMANUELLE.

A rejtvény szórakoztató jellegű, tehát nem kell beküldeni.

Mokos István

Folytatás a 30. oldalról

```

10030 IF B3>37 THEN B3=0
10040 IF B1<65 THEN RETURN
10045 IF B2=B3 THEN B2=0
10050 IF B3=37 THEN B1=B1+128:B2=B2+128:REM **** INTEGER ****
10060 IF B3=36 THEN B2=B2+128:REM **** STRING ****
10065 REM **** BASIC TAROLO ATVIZSGALASA ****
10070 I=0:XY=0:FOR I=PEEK(47)+256*PEEK(48) TO PEEK(49)+256*PEEK(50)
10080 IF B1=PEEK(I) AND B2=PEEK(I+1) GOTO 10110 :REM MEGVAN A TOMB !
10090 XY=I:I=I+PEEK(I+2)+256*PEEK(I+3)-1:IF XY<I THENNEXT
10100 RETURN :REM NINCS MEG
10105 :
10110 B1=I:B2=PEEK(B1+2)+256*PEEK(B1+3)
10120 FORI=B1+B2 TO PEEK(49)+256*PEEK(50):POKEI-B2,PEEK(I):NEXT
10130 B1=I-B2-1:POKE50,B1/256:POKE49,B1-PEEK(50)*256
10140 RETURN
    
```

READY.



Mindenféle Commodore IC-k, elektronikus alkatrészek beszerzése, C64 és perifériáinak javítása.  
Tel.: 1-731-783 (üzenet).

Ha bármi gondod van a C64-gyel, segítek. CODI=Commodore DIsc & Commodore Információs Szolgálat.  
Varsányi Gábor, Szombat-hely, Nagy L. u. 11.

C64 program és információ-csere lemezen és kazettán.  
Székely Krisztián,  
tel.: 96/20-364.

Itt az OTHIS SOFTWARE!  
Az országban egyedülálló, ingyenes, képes katalógus! Szuper olcsó árak! C64-re és Amigára egyaránt! Írjál, de válaszboríték nem fontos!  
OTHIS SOFTWARE, 2600 Vác, Eperfa u. 6.

Figyelem! Az OTHIS SOFTWARE megjelenteti Amigára a Sim City és a Lords of Doom magyar nyelvű változatát! A programokat díszcsomagolásban, kiegészítőkkal hozzuk forgalomba. Kérésre bővebb tájékoztatót küldünk.  
OTHIS SOFTWARE, 2600 Vác, Eperfa u. 6.

C64 programokat adok, cserélek, 1 program 8 forint, sex 10 ft, vannak 90—91-es.  
Balogh Zsolt, 4031 Debrecen, István u. 51. 1/b.

A SCAN 84-nél többet ismerő vírusölő 500 Ft. Küldj vírust, ingyen megkapod. Kérj tájékoztatót!  
Répás Sándor, 1519 Pf.: 376.

+4 és 64 programok cserébe lemezen és kazettán.  
Solymosi Béla, 1204 Budapest, Dessewffy 41.

C16—Plus/4—C116-ra lemezen és kazettán olvasható programújságot készítetek. Ingyenes. Küldj adathordozót. '91 végén nyolc szám kész.  
Láng Attila D. 1107 Budapest, Gyűrű u. 28. II/6.

Action Replay. KickStart 2.0, Teletext decoder, 512 k bővítő. Syncro Express/Cyclone adapter és még sok más Amiga kiegészítő, valamint egy Amiga 500-as eladó.  
Érdeklődj az 129-5955-ös telefonszámon, Kiss Tamásnál.

C64 + 1541 floppy + final III + joy + 40 db lemez + szakirodalom. Irányár 33 ezer forint.  
Érdeklődni levélben Láris Gergely 8200 Veszprém, Jutasi u. 59/9. vagy telefonon 80/20-773.

Keresem a Newspaper és az időrégész című programokat lemezen. Csere esetén listát küldök.  
Berecz Balázs 3925 Prügy, Móricz Zs. 41/a.

Elcserélném ZX spectrumot tartozékokkal (irodalommal, multiface interface, joy stb.) enterprise 128 vagy Plus/4 vagy C64-re.  
Szunyogh Zsolt, 2040 Budapest, Baross u. 94.

Segítséget kérek C64-es „MUSIC SHOP” program használatához, térítés ellenében is.  
Varga László, 1148 Budapest, Nagy Lajos kir. utja 76/b. II. 2.

C64-es programok nagy választékban eladók, csak lemezen. Egységár: 100 Ft/lemez. Válaszborítékra listát küldök.  
Nagy István, 1202 Budapest, Mártírok útja 151.

C+4 programok 5 Ft/db-ért eladók. Válaszborítékért listát küldök.  
Bujdosó Csaba, 2340 Kiskunlacháza, Tavasz 7.

Eladó IBM XT különlegesen jó kiépítésben. Különleges grafikus kártya, nagy winchester, óra, 2 floppy, sok program stb. AT szuper VGA-val. Kérek ajánlatot.  
Láng Attila D. 1107 Budapest, Gyűrű u. 28. II/6.

Eladó egy Commodore 64-es számítógép OC-118-as drive-val (C-1541 kompatibilis), joystick-kel, sok programmal és hozzávaló szakirodalommal. Ár megegyezés szerint.  
Vigh György, tel.: napközben 156-2520; este 271-7224.

C64-re már 3 Ft-tól eladók kiváló játék és felhasználói programok lemeze és kazettára egyaránt.  
K. K., 6750 Szeged-Algyő, Bartók B. 6.

Commodore +4-re játékokat keresek vételre. Esetleg csere is szóba jöhet.  
Kovács Zoltán, 2040 Budapest, Naphegy u. 24.

C16—Plus/4—C116-ra programot cserélek és adok. Cse-realappal nem rendelkezőknek is ingyen, minden mennyiségben.  
Láng Attila D. 1107 Budapest, Gyűrű u. 28. II/6.

IBM PC-programokat cserélek és adok. Listát kérek-küldök. 360 K-s, 1,2 Mb-os, 720 K-s lemezen. Vírusmentesen!  
Láng Attila D. 1107 Budapest, Gyűrű u. 28. II/6.

Eladó C64+1541, 2 floppy + 1802 Commodore monitor + 140 lemez + könyvek. Irányár 49 000 Ft.  
Buláth András, 1-426-374.

Eladó Enterprise 128 számítógép + magnó + kazetták (10 db) + könyvek. Irányár 9900 Ft.  
Buláth András, tel.: 1-426-374.

C64 egy program 10 Ft, egy teljes kazettaoldal 150 Ft, szexprogram 15 Ft.  
Gönczi Sándor 3910 Tokaj, Ady Endre u. 10.

Geos felhasználók jelentkezését várom, lemez-tapasztalt-cserére, illetve azokét, akik bővítéseket készítenek amatőr szinten.  
Nagy Péter, 1056 Budapest, Molnár u. 20.

Keresem AZF-16 Compat-pilot, F-195 Tealth fighter, f-29 Retaliator, F/A18 Interceptor, M1 tank platoon operation stealth, A-10 tank killer, és a Fighter bonker című C64 programokat. Cserélek és veszek olcsón. Leveleket várok!  
Tagsági szám: 1370.

C64 egy program 10 forint. Egy teljes kazettaoldal 150 Ft. Szexprogram 15 Ft.  
Gönczi Sándor, 3910 Tokaj, Ady Endre út 10.

Eladó C64-II+1541-II + Speeddos + egér + könyvek + 160 programlemez = 25 000 Ft, epromégető 3000 Ft A/D 2500 Ft.  
Tel.: (46) 79-759

C+4-re szimulátorokat veszek, listát kérek. Merce-naryz, Elite stb.  
Siető Péter, 3900 Szerencs, Ondi út 1/B.

C64-re Giga-Cad és Amiga-Paint használati utasítás lemezen programonként 100 Ft, nyomtatva 350 Ft + postaköltség.  
Apró Lajos, 4220 Hajdúböszörmény, Táncsics M. krt. 72.

C64 programok lemezen olcsón eladók. Válaszborítékért listát küldök. Leveleket „programfutár” jellegre a szerkesztőségbe.

Totókulcskészítő programok (8 kbyte-os) C64-re. Válaszborítékért tájékoztató.  
Nagy Miklós, 4244 Újfehértó, Pf.: 41.

C64/128 programlemezek a legolcsóbban 65 Ft/db áron eladók. 300 lemeznyi program a választék.  
Kopácsi Lajosné, 1031 Budapest, Vízimolnár u. 2. X/95.

Kapható a D and T kártyacsatlád tagjai magnó és floppy-file kezelésére C64-hez. Közületeknek is utánvétellel.  
Jakab Péter, 1046 Budapest, Török I. út 25. Tel.: 169-1466/2768.

**AGFA**

fénymásoló rendszer

## 10 000 forintos vásárlási utalvány

Beváltható fénymásoló vásárlása esetén

 az **ASI** Kft.-nél  
 Budapest XI., Bartók Béla út 120.  
 Telefon: 185-1507, fax: 185-1760

**Érvényes: február 29-ig.**

Több utalvány a vásárlás (nettó ár) 5%-ig használható fel!

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

1053 Budapest, Magyar u. 12—14	Telefon: 117-3551
1083 Budapest, Szigony u. 9.	Telefon: 134-3153
1191 Budapest, Gábor Á. sétány 3.	Telefon: 127-4763
3525 Miskolc, Fazekas u. 1—3.	Telefon: 46-17-011
4034 Debrecen, Holló L. u. 14.	Telefon: 52-32-863
5600 Békéscsaba, Bartók B. u. 37.	Telefon: 66-27-195
6724 Szeged, Csongrádi sugárút 76.	Telefon: 62-13-377
7624 Pécs, Jurisics M. u. 17.	Telefon: 72-11-812
8000 Székesfehérvár, Széchenyi u. 15/a.	Telefon: 22-12-711
9700 Szombathely, Szalonok u. 31.	Telefon: 94-13-419

Felvevőhelyek:

9024 Győr, Babits M. 75.	
6000 Kecskemét, Széchenyi tér 1—3.	Telefon: 76—23—720

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:

VC—20 memóriabővítés 3—27 kByte-os:	kiépítéstől függő
C—16, C—116 memóriájának bővítése 64 kByte-ra:	3500 Ft
C—16 belső 16 kByte-os EPROM bővítés:	1450 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	2800 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	2000 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	3200 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4	5000 Ft
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

### Kedvezmény nélkül:

Letilthatalan RESET C—16, C—116, +4 minden programot megállít	3000 Ft
+4 és 1541 kompatibilis lemezegység párhuzamosítása	1450 Ft

Az EPROM-ba hozott programokat vagy a már kész menük valamelyikét építjük be (bekapcsoláskor és RESET-kor menüvel jelentkeznek, kikapcsoláskor nem törölődik). A SOFT—ROM tetszőleges EPROM menü futtatására alkalmas (RESET-kor menüvel jelentkeznek, kikapcsoláskor törölődik).

A párhuzamosított lemezegységhez jár egy lemezoldalnyi speciális program, melyek az új lehetőséget kihasználják (20-szoros gyorsított, 15 másodperces lemezoldalmásoló stb.).

A fenti bővítések megrendelhetőek az OCE. irodájában a Pötyögőszolgálat napjain 16—18 óra között. Részletes felvilágosítást az 1-363-951-es telefonszámon tudunk nyújtani (főleg este). Árainkat az alkatrészárak változásai befolyásolhatják.

## FEBRUÁRI 60 Ft-os vásárlási utalvány

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

**Érvényes: február 29-ig.**

TKD—BUDAPEST

## FEBRUÁRI 400 Ft-os vásárlási utalvány

 Beváltható: TKD—BUDAPEST  
 1117 Budapest, Schönhercz Z. u. 21.  
 Egy gép vásárlásához egy utalvány használható fel!

**Érvényes: február 29-ig.**


### MEGRENDELŐLAP

 Megrendelek . . . . . db 14 nyelvű elektronikus szótárt.  
 Kérem a készüléket az alábbi címre postai utánvétellel elküldeni:

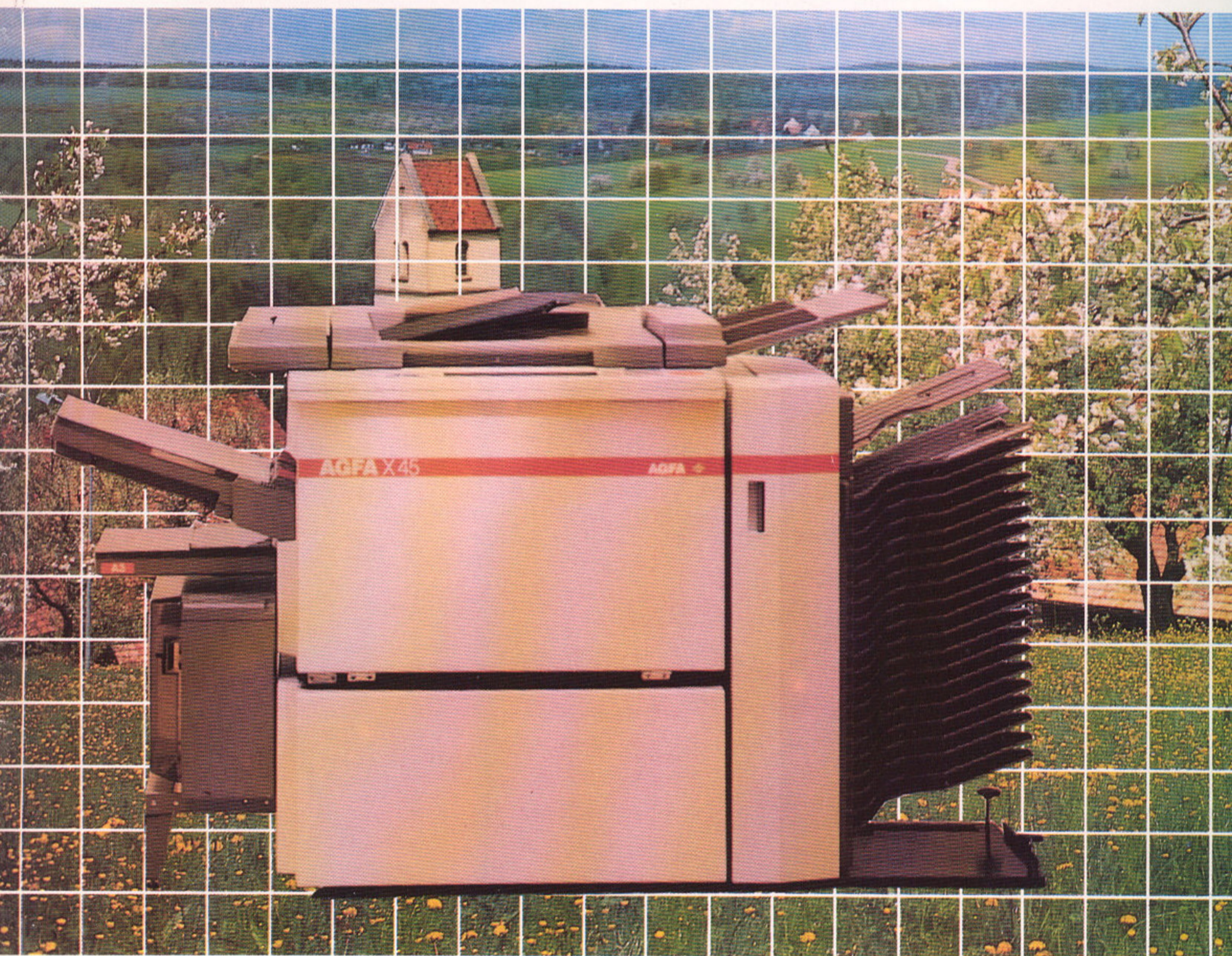
 NÉV: . . . . .  
 . . . . . irányítószám . . . . . (város)  
 . . . . . (járás, kerület) . . . . . utca,  
 . . . . . házsám . . . . . emelet, ajtó.

aláírás

 A megrendelőlapot TKD—BUDAPEST címére kérjük elküldeni:  
 1117 Budapest, Schönhercz Zoltán u. 21. Tel.: 181-0352

# Nyugat-európai Precizitás

## Azonnali szállítás



# AGFA



Ára: 58 Ft

**NOVOTRONE**  
SZERVIZ Kft.

**Központ:**

1053 Budapest, Heszlmann I. u. 9.  
Tel.: 117-4144 Tx: 22-7621 Fax: 117-9692

**Budapesti szervizeink:**

1053 Budapest, Magyar u. 12—14. Tel.: 117-3551  
1191 Budapest, Gábor A. u. 3. Tel.: 127-4763  
1083 Budapest, Szigony u. 9. Tel.: 134-3153

**Professzionális számítógépek  
PC XT/AT**

**Számítógépek és perifériák kölcsönzése:**

1092 Budapest, Bakáts tér 4.  
Tel.: 117-0061

**Otthoni és iskolai számítógépek**

**Lemezegységek, nyomtatók, hálózatok,  
monochrom és színes monitorok**

**COMMODORE, ATARI, VIDEOTON TVC  
márkaszerviz**

## Országos számítógép-szervizhálózat

**Gépek értékesítése  
szervizeinkben**



**NOVOTRON**

**PC shop:**

1053 Budapest, Magyar u. 1.  
Tel.: 118-9481

**SZÁMÍTÓGÉPEK MINTA UTÁNI ÉRTÉKESÍTÉSE**  
szállítással és üzembehelyezéssel

Átalánydíjas  
szerződés

Biztonság,  
gyorsaság

Örök -  
garancia