

Az Országos Commodore Egyesület lapja

újság

1992/6

# GEOS

*Tippek-  
trükkök*



# Játékfilozófiák

*Egy nagyszerű FAX a kicsik között!*



**Navo  
TRADING**

**TELEFONSZALON**

1130 Budapest,  
Fürst Sándor u. 14/b.  
Tel.: 132-8739, 131-5573

**Ára:  
44 500 Ft  
+ÁFA**

**Műszaki paraméterek:**

- Postai engedély
- Egyszerű kezelhetőség
- Egyszerű kivitel
- Automatikus és kézi vezérlés
- Másolási funkció
- Nagy átviteli sebesség (20 mp/oldal)
- Automatikus adássebesség (9600/7200/4800/2400 bps)
- Azonosító adatok közlése (fejléc, dátum, cím, telefonszám)
- Beépített óra
- Kontrasztállíthatóság
- Normál és finom felbontás
- G 3 CCITT kompatibilis
- 216 mm x 30 m faxpapír
- Programozási funkciók (1—10 csengetésszám, adás-vételi riport)
- Automatikus tesztprogram
- Automata FAX — üzenetrögzítő átkapcsoló

**EFAX-101**



**Avatex**

**Viszonteladónak kedvezmény!**

Telefonok, üzenetrögzítők, faxok, kis és nagy központok, kábelek, alkatrészecskék stb. — Tv-k, videók, kamerák, autórádiók!



## MIT, HOGYAN, HOL, MIKOR?

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

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

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

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

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

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

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

**APRÓHIRDETÉS:** Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 100 forint. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

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

Kedvezményes ár! Tagoknak olcsóbb!

Az újságban eddig megjelent programok gépenként összegyűjtve megrendelhetők. VC 20, C16, PLUS/4, C128, C64. További felvilágosítást is adunk a 1-76-22-57-es telefonszámon vagy levélben!

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

### VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:

Baja, AXIS Kft.,  
Győri Bartók Béla Művelődési Ház,  
Jászberényi Városi Könyvtár,  
Kecskemét, SZIGMA—BIT,  
Pécsi Apáczai Csere János Gimnázium,  
Zalaegerszegi Ságvári Andre 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  
92.0157 MSZH Nyomda és Kiadó Kft., Budapest  
Felelős vezető: Nagy László

## Kedves Tagtársaink, Olvasóink!

### Egyesületi és szerkesztőségi CÍMÜNK MEGVÁLTOZOTT!

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

## Pötyögőszolgálat

A kiválasztott programok megrendelhetők levélben, utánvétellel.

Postacím:

1388 Budapest 62., Postafiók: 86.

Felvilágosítás telefonon:

1-76-22-57.

## Programok utántöltése a C64-nél

Ha egy BASIC programból (legtöbbször egy gépi kódú programot) utántöltéssel akarunk beolvastatni, akkor az nem is olyan nehéz. Igaz, néhány dolgot szem előtt kell tartani.

A legfontosabb az, hogy a C64 a programból történő betöltés után egy „GOTO első programsor” utasítást hajt végre. Ha például egy BASIC program első sorában egy gépi kódú rutint szeretnénk betölteni a 10 LOAD "XYZ" ,8,1-gyel, akkor a C64 szépen ki fogja magát „akasztani” azzal, hogy egy végtelen ciklusba lép. Ennek oka az előbb elmondottakban keresendő, hisz a gép a betöltés után ismét a 10. sorra lép, betölti a programot, majd ismét a 10. sorra lép...

Valamilyen módon tehát jelezniünk kell, hogy a kívánt program már betöltődött. A legegyszerűbben ezt az alábbi sorral érhetjük el:

```
10 IF A=0 THEN A=1 : LOAD "XYZ",8,1
```

Az első programindításnál, tehát a RUN-nál az összes változó a 0 értéket kapja. Ezért az IF feltétel teljesül, az A értékét 1-re állítjuk be, és a betöltés megtörténik. Ezután a C64 a már ismert GOTO-val az első sorra lép, de ilyenkor nem változnak meg a változók értékei. Most tehát az A értéke 1, azaz az IF feltétel nem teljesül. Így átugorjuk a LOAD-ot és a programunk máris rendben működik.

Ha több programot szeretnénk utántölteni, ez sem nehéz:

```
10 IF A=0 THEN A=1 : LOAD "P1",8,1
20 IF A=1 THEN A=2 : LOAD "P2",8,1
30 IF A=2 THEN A=3 : LOAD "P3",8,1
... és így tovább
```

Mindez azonban csupán a gépi kódú programokra érvényes! Ha BASIC programokat akarunk utántölteni, akkor az valamivel komplikáltabban megy csak. Mindaddig, amíg a betöltendő program rövidebb mint az első, bőven elég egy egyszerű LOAD parancsot használni. Az IF lekérdezésre nincs is szükség, mivel a betöltött programot azonnal el is indítja a gépünk. Ha azonban a betöltendő program hosszabb mint az, amelyik a tárban van, akkor bizony nem fog sikerülni a dolog.

A most bemutatott megoldás a legelőnyösebb az összes között, azt úgy a gépi kódban mint a BASIC-ban írt programokhoz használhatjuk. A megoldás lényege: a programozott parancsmód.

Ha a számunkra a billentyűzetpuffer fogalma ismert, akkor ezt a bekezdést nyugodtan átugorhatjuk. Ha nem akkor adjuk be a következő sor (sorszám nélkül), nyomjuk meg a RETURN, majd rögtön ezután nyomkorásszuk meg kissé a billentyűzetet mindaddig, amíg a READY meg nem jelenik:

```
FOR I=1 TO 2000 : NEXT
```

Nos a READY után azt vehetjük észre, hogy a gépünk megjegyezte a lenyomott billentyűket (maximum 10 darabot)! Ezeket a jeleket a gép az úgynevezett billentyűzetpufferbe rakja le. A jó ebben a dologban az, hogy néhány okos POKE utasítás segítségével egy BASIC programban ezeket a billentyű lenyomásokat szimulálni tudjuk. Ez tehát az említett programozott parancsmód.

Fölírjuk a LOAD parancsot — amely a második programot hivatott betölteni — a képernyőre, majd eljártsszuk a C64-nek, hogy megnyomtuk a RETURN gombot. Erre a LOAD parancs

végrehajtódik (parancs módban nincs szükség arra, hogy bármi-re is ügyeljünk azon kívül, hogy a tárban lévő BASIC program törlődik, ha BASIC programot hívunk be.) Tehát:

```
10 PRINT "{tr}[2 le]LOAD" CHR$(34)"XYZ" CHR$(34) ",8
20 PRINT "{4 le}RUN{home}";
```

A 10 sorban álló CHR\$(34) az idézőjelet helyettesíti, hiszen azt másképpen nem tudjuk sztringként a képernyőre vinni (PRINT " " " -ként??). A kapcsos zárójelet pedig ne keressük a billentyűn. Mi itt a nehezen kiolvasható kódképek helyett a szükséges billentyű feliratát/jelentését írtuk le. A {home} a „kurzor home”, a {le} a „kurzor le” billentyűket jelenti.

Ha ezt a programot RUN-nal indítjuk, akkor a 3. sorban megjelenik a LOAD parancs, majd valamivel alatta a RUN. Ha most kétszer egymás után megnyomhatnánk a RETURN billentyűt, és a lemezen lenne egy „XYZ” nevű programunk, máris megoldódott volna a problémánk. Mivel mi ezt programból akarjuk szimulálni, ezért a RETURN ASCII kódját be kell írunk a billentyűzet pufferbe. Ezt az alábbi módon lehet megtenni:

```
30 POKE 631,13 : POKE 632,13
```

A billentyűzet puffer ugyanis a 631—640 memóriacímen található, a RETURN ASCII kódja pedig a CHR\$(13). Nos ezután csak azt kell a C64 tudomására hozni, hogy a pufferben van valami:

```
40 POKE 198,2
```

Ezután utasítanunk kell a gépet, hogy nézzen is bele a pufferbe, tehát be kell fejezni a programot:

```
50 END
```

A billentyűzet puffer ezen kívül egy egész sor más lehetőséget is rejt magában (hagy említsük meg étvágycsinálónak az ön-modifikáló BASIC programokat, amelyek fölülírják magukat!). Aki ezzel egyszer megbarátkozott, biztos hogy nélkülözhetetlennek fogja érezni. Egy dolgot még figyelembe kell venni, és pedig azt, hogy a pufferbe csak ASCII értéket lehet pókolni, ezeket vagy a PRINT ASC " billentyű ", vagy a C64 kézikönyvének segítségével állapíthatjuk meg.

## Teszt készítése és értékelése PLus-4-en

Az iskolában elég gyakori feladat a tanárok számára tesztek összeállítása. A két program ezt a munkát segíti. A „TESZTKÉSZÍTŐ” programot betöltés után lehet futtatni. Be kell írni a feladatlap címét és a kérdések számát. Ezután következhet a kérdések és pontszámok begépelése. (Az a., b., c., ... lehetséges válaszok még a kérdéshez tartoznak!) A válaszhoz csak a helyes betűjelet vagy számot kell beírni. A program az egész feladatlapot lemezre menti.

A „TESZTERTEKELO” programmal lehet aktivizálni a feladatlapot. Lehetőség van a lap kinyomtatására, de a feleltetés a számítógép mellett is megtörténhet. Ilyenkor a képernyőn megjelenő kérdésekre a diák a billentyűzeten adja meg a válaszokat. A program végén a képernyőn megjelenik a teljesítmény a pontszám és a kérdések százalékában.

Szűcs Gergely  
Zalaegerszeg  
Ságvári Gimnázium

*A szerk. megj.: A programok alighanem beépített ékezetes karakterkészletű gépen készültek. Nekünk nem ilyen van, elnézést a listáért.*



```
1 REM SZUCS GERGELY
2 REM TESZTERTEKELO
3 REM PROGRAM
4 REM PLUS/4
10 PRINT "J":PRINTCHR$(14)
15 PRINTTAB(134)"* FILE NEVE";
16 INPUT TN$
17 PRINTTAB(125)"* FELELI NEVE";
18 INPUT NEV$
19 PRINT "J":PRINTCHR$(14)
20 PRINTTAB(130)"* PROGRAM TITLIDIK!";
50 OPEN 1,8,2,TN$+ ".S,R"
60 INPUT#1,TN$
70 INPUT#1,N
80 DIM V$(N),P$(N),VSZ$(N),K$(N)
85 REM GOTO 600
90 FOR I=1 TO N
100 A$="":K$(I)="";
105 DO UNTIL A$="*"
107 GET#1,A$:IF A$<>"*" THEN K$(I)=K$(I)+A$
110 LOOP
120 GET#1,GN$
130 GET#1,V$(I),GN$
150 INPUT#1,P(I)
160 NEXT I
170 CLOSE 1:PRINTCHR$(14):
175 GOTO 600
180 HIB=0:HELY=0:
181 OP=0:MAXP=0
185 FOR I=1 TO N
190 PRINT "*****";I;" KARDAS"
210 PRINT "*****";K$(I):PRINT "***** REM A HELYES VILASZ BETJEL TITLIDIK"
230 GETKEY VSZ$(I):PRINTVSZ$(I)
235 PRINTTAB(130)"* IL VILASZOLT JEL/HELY?"
250 GET W$:IF W$<>"I"ANDW$<>"\ "ANDW$<>"/"ANDW$<>"N" THEN 250
260 IF W$="/" OR W$="N" THEN 190
270 MAXP=MAXP+P(I):
280 IF VSZ$(I)=V$(I) THEN OP=OP+P(I):HELY=HELY+1:ELSE HIB=HIB+1
290 NEXT I
295 GOTO 400
300 PRINT "J"
305 PRINTTAB(135)"*RT KEL *****"
310 PRINT " IIBR:";HIB:PRINT
320 PRINT " IELYES:";HELY:PRINT
340 PRINT " SSZPONTSZAM:";OP:PRINT
350 PRINT " IELJESITMENY(KARDASEK %-BAN):";INT(100*HELY/N);"%":PRINT
360 PRINT " IELJESITMENY (PONTSZAM %-BAN):";INT(100*OP/MAXP);"%":PRINT
365 PRINT "*****SM TELJESITMENY/HELY?"
366 GETKEY GN$:IF GN$="I" OR GN$="\ " THEN 180
370 GOTO 600
400 PRINT "J":PRINTTAB(17)"*ENH:"
410 PRINT:PRINT:PRINT:PRINT " 1. *RT KEL ST KPERNYI RE"
420 PRINT:PRINT:PRINT " 2. *RT KEL S NYOMTAT RA"
425 PRINT:PRINT:PRINT " 3. XISSZA A FI MENUBE"
430 PRINT "***** * SORSZAMMAL VILASSZON!"
440 GETKEY O$:O=VAL(O$):IF O=3 THEN 600
450 IF (O-1)*(O-2)>0 THEN 440
460 IF O=1 THEN 300
470 OPEN 4,4,7
475 PRINT#4," ";NEV$
480 PRINT#4," *RTEKELES "
490 PRINT#4," IIBR:";HIB:PRINT
500 PRINT#4," IELYES:";HELY:PRINT
510 PRINT#4," SSZPONTSZAM:";OP:PRINT
520 PRINT#4," IELJESITMENY(KERDESEK %-BAN):";INT(100*HELY/N);"%":PRINT
523 PRINT#4," IELJESITMENY (PONTSZAM %-BAN):";INT(100*OP/MAXP);"%":PRINT
525 CLOSE 4
600 PRINT "J":PRINTTAB(17)"*ENH:"
610 PRINT:PRINT:PRINT:PRINT " 1. *IRAT S KPERNYI RE"
620 PRINT:PRINT:PRINT " 2. *IRAT S NYOMTAT RA"
625 PRINT:PRINT:PRINT " 3. XJSE "
630 PRINT "***** * SORSZAMMAL VILASSZON!"
640 GETKEY O$:O=VAL(O$):IF O=3 THEN END
650 IF (O-1)*(O-2)>0 THEN 640
655 PRINT "***** *OLGOZOM!!!"
660 IF O=1 THEN 180
665 OPEN 4,4,7:REM CMD 4
670 FOR I=1 TO N
680 PRINT#4," ";I;" . KERDES"
690 PRINT#4,K$(I)
700 NEXT I
710 CLOSE 1:CLOSE 4:PRINTCHR$(14):
720 GOTO 600
730 CLOSE 1:CLOSE 4:PRINTCHR$(14):
740 GOTO 600
```

READY.



```

10 REM *****
20 REM * TESZTKESZITO *
30 REM * PROGRAM *
40 REM * PLUS/4 GEPE *
50 REM *****
60 PRINT"J":PRINTCHR$(14):PRINTTAB(90)"TESZTKESZITO PROGRAM":PRINT:PRINT
70 INPUT"TESZT NEVE";TN$:PRINT
80 DO
90 INPUT" FELADATOK SZAMA";N:PRINT
100 LOOP UNTIL (N>0) AND (INT(N)=N)
110 DO
120 DIM K$(N),V$(N),P(N)
130 FOR I=1 TO N
140 PRINT"KARD S SZIVEG ET! " V $ T *-GAL JELEZZE!
150 A$="" :K$(I)=""
160 DO UNTIL A$="*"
170 GETKEY A$:K$(I)=K$(I)+A$
180 IF A$<>"*" THENPRINTA$;
190 LOOP
200 PRINT" HELYES V LASZ BET JEL ET!"
210 GETKEY V$(I):PRINT" HELYES V LASZ:";V$(I)
220 PRINT" PONT SZAM";:INPUT P(I)
230 PRINT"K$(I);" :PRINT"V$(I):PRINT" P(I);" PONT"
240 PRINTTAB(131)" AZ AORTOK SZIVEG ET?"
250 GETKEY W$:IF W$<>"\."AND W$<>"I" AND W$<>" /"AND W$<>"N"THEN 250
260 IF W$="/" OR W$="N" THEN 140
270 NEXT I
280 OPEN 1,8,1,TN$+ ".S,N"
290 PRINT#1,TN$,"
300 PRINT#1,N,"
310 FOR I=1 TO N
320 PRINT#1,K$(I)
330 PRINT#1,V$(I)
340 PRINT#1,P(I),"
350 NEXT I
360 CLOSE 1

READY.
    
```

# MAKR VILÁG



Szellemileg és lelkileg

**igényes utasok irodája**

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



Országos Commodore Egyesület tagjainak jelentős utazási kedvezmények!

# Játékfilozófiák a C64-esen

Egyetlen egy más számítógép sem rendelkezik azzal a hihetetlen játékválasztékkal, mint a C64-es. Ez éppen elég ok arra, hogy a játékok fejlődését összefoglaljuk és néhány tippet adjunk a játékvásárláshoz.

A játékszoftverek terén a C64-es egy olyan jelenség, amelynek az egész számítógépes körben nincsen párja. Mindenféle ízlésnek megfelelő játékot találhatunk. Emellett a legutóbbi időkben mindenekelőtt az olcsóbb termékek esetében figyelhető meg jól kivehetően a minőségi javulás. Ezért tehát az árat már régóta nem lehet a játék minőségének meghatározásához kritériumként alkalmazni. De vajon lehet-e előre látni, hogy egy játék értékes-e, és az elindult a klasszikussá válás útján? Ez mindenekelőtt azoknak a C64 tulajdonosoknak jelent problémát, akik csupán mellékesen foglalkoznak játékokkal, és csak abban az esetben vesznek ilyen programot, ha biztosan tudják, hogy az nem lesz bukta. Éppen ezért szeretnénk a továbbiakban olyan tanácsokat adni, amelyek fogódzót jelentenek a vásárláskor. Remélhetőleg ezzel a csöbe húzást el lehet kerülni. Itt most az akció- és a kalandjátékokra korlátozzuk a részletes elemzést. Másféle játékok esetében, mint például a stratégia-, szimulációs- vagy a szerepjátékos fajtáknál hasonló gondolatok érvényesek.

## Játékok öregeknek, fiataloknak

Már jóideje léteznek olyan emberek, az úgynevezett játékdizajnerek, akik jónevű szoftverházak alkalmazásában állnak, és semmi más feladatuk nincs, mint új játékokat keresni, és azokat programkonceptióba önteni. A konkrét programozással ezután csupán kismértékben foglalkoznak; az adott számítógépre a megfelelő áttételt már teammunkában végzik el — rendszerint mások. Új játéktrendeket megjósolni, a gyorsan változó számítógépes iparban szinte hihetetlenül nehéz. Ugyanakkor azt a tényt is meg kell állapítanunk, hogy a piacon mindig találni valami egészen újat. Állíthatjuk, hogy főleg azok a játékok lehetnek sikeresek, amelyek bármilyen korú számítógéptulaj-

donos által játszhatók. Ennek egyik oka az, hogy a megvett játékot a vevő később is mindig szívesen elő fogja venni, másrészt pedig az egyéb korosztályhoz tartozó játékosok vagy nézők ilyenkor gyakorta fellelkesülnek, de legalábbis érdeklődést tanúsítanak az adott játék iránt. Ez utóbbi különösen az ifjúságot érinti, akik rendszeresen igyekeznek a szüleiket meggyőzni arról, mire is képes (Wintergames, Summergames, a különböző golf, tenisz és asztalitenisz játékok, az Exploding Fist, Uchi Mata stb.) részben ezzel magyarázható. Maga a játékölet az életből ellesett téma, amelyet mindenki azonnal megért.

## Játékok a játéktermekből

A játékgyártásnak a legutóbbi időben tapasztalt egyik legjelentősebb trendje volt a játéktermek slágereinek adaptálása. Ez azt jelenti, hogy a közönség előtt már sikert aratott játékokat egyszerűen átírják a megfelelő géptípusokra. Ez a szoftvercégeknek első látásra legalábbis könnyű utat jelent jó programok megírásához. Az automatáknál rendkívül egyszerűen meg lehet állapítani a bedobott pénzérmék számából, mennyire vált be az adott játék. Ezért ki lehet indulni abból, hogy a játék a szoftverpiacon is jól eladható lesz. Mivel az ilyen játékok jogdíja az automaták gyártóinak legtöbbször igen sokba kerül, a szoftvercég mindjárt választás elé kerül: Mennyit investáljon még az adott projektbe, hogy kompromisszumot lehessen kötni a kedvező ár és a játék vonzóereje között. Hiszen egy más gépre történő átírás sem megy egy csapásra, ráadásul ez nem éppen programozói rutinfeladat. Azon kívül a vásárlók sem szívesen adnak ki sok pénzt, mégha mégoly sikeres játék átírásáról van is szó. Ezért gyakorta találkozhatók villámtermelési hatással, amikor a programból teljesen eltűnik a játékteremben átélt lebilincselő hatás, és ez alapján annak a néven kívül semmi köze sincs az eredeti termékhez.

A programozói munka minősége legnagyobb részben a programozó tehetségétől és tudásától, na és attól az időtől függ, amely a rendelkezésére áll az adott pro-

jekt befejezéséhez. Csakis ez lehet az oka annak, hogy a jó ötlet és az eredeti játéktaktika ellenére sok játék egyáltalán nem okoz örömet. Tehát minden mégoly sokat ígérő névnek is előbb bizonyítania kell a tehetségét az adott géptípuson. A győzelmi babérok előre történő kiosztásával tehát csínján kell bánni!

## A játékos frusztrációja

Jobb Amigára történő átírást, mint amelyet a játéktermekben megismert Marble Madness program esetében tapasztaltunk, valószínűleg nem fogunk jelenleg találni. Készült persze a C64-hez is egy jólsikerült változat. A C64-nek tehát semmi oka arra, hogy szégyenkezve elbújjon a nagyobb gépek mögött. Éppen elég, kizárólag a C64-re írt játéklágeret ismerünk, és léteznek jó adaptációk a „Coin-up” klasszikusok esetében is. Gondoljunk csak az azóta klasszikussá vált slágerre, a „Ghost and Goblins”-re. Ennek átírata jól sikerült, bár a C64-es változat könnyebb mint a játéktermekben játszott eredeti. És itt jelentkezik az újabb probléma. A játéktermekben ugyanis azon igyekeznek, hogy a játékos ne maradjon sokáig „életben”, így gyakrabban essen kísértésbe, és a további játékhoz újabb érmét dobjon a perselybe. Végül is itt a haszon az elsőrangú szempont. Az egyes szintek gyakorta teljesíthetetlenül nehezek, és a játéklehetőségek, értsd emberké vagy hasonlók meglehetősen korlátozva vannak. De miért lenne pont erre szükség a piacon ma fellelhető akciójátékok legnagyobb részének esetében? Meglehetősen idegesítő az, ha az adott játék vevője, aki az asztalra tette a vételárat, odahaza semmi esélyt nem kap ahhoz, hogy valaha is eljusson a magasabb játékszintekre legalább a megtekintés erejéig, a játszásról már nem is szólva. Mással nem is magyarázható az örökélet POKE-ok és egyéb segítségk iránt mutatkozó oly nagy érdeklődés. Mialatt a játéktermek közönségénél abból lehet kiindulni, hogy ők olyan megszállott játékosok, akik hajlandók több pénzt kiadni az ilyesfajta örömkért. Ugyanez azonban már egyáltalán nem érvényes a C64 tulajdonosok legnagyobb részére. Szinte szívfájdalom-

mal lehet csak olyan játékklasszikusokra visszagondolni, mint amilyen az Epyx „Jumpman”-je. Hiszen melyik mai játéknál találkozhatunk a kezdéskor elegendő számú emberkével, választható nehézségi fokkal és egy olyan Highscore-listával, amely lemezre menthető? Ez utóbbi esetben a program minden egyes eredmény mögé azt is följegyzi, melyik nehézségi fokozatban értük el a pontszámot. Még a kezdők számára is örömet okoz ha látják az eredményeket, mialatt a beavatottak is megtalálják a számításukat. Hasonló kategóriába esik a Space Taxi program is.

Az értékelésnél egy újabb szempont az, mennyire lehet a játékfigurát a képernyőn az adott beviteli eszközzel vezérelni. Az akciójátékok esetében ez általában a joystick, bár vannak kivételek is. De erre később még vissza fogunk térni. Olyan játékoknál, amelyek újdonságként jelennek meg a piacon, gyakran leragad az ember például a még sosem látott grafikánál vagy az eredeti játékstratégiánál. Holott főleg az akciójátékok esetében ajánlható melegen a saját joystick bevitele az adott üzletbe, hogy azzal lehessen megállapítani a vásárolni kívánt játék vezérelhetőségét. Hiszen mit használ a számunkra a legszebb grafika, a legfűlbemászóbb zene, ha cserébe a joystick lekérdezéséért felelős rutin olyan lassú, hogy rendes játékokra nincs is mód. Bár ez a példa sarkított eset, de újra meg újra meg kell állapítanunk, hogy egyik-másik esetben a program koncepcióját nem oldották meg megfelelően a játék vezérlése szempontjából. Ha tehát az adott játékot egy ilyen igazi teszteléssel, a saját joystickkal problémamentesen tudjuk vezérelni, tehát mondjuk a 45 fokban végzett mozgásokat is végre tudta hajtani a program, akkor biztosak lehetünk abban, hogy az nem okoz otthon sem csalódást. Gyakorta a játékos a program nehézségeért a programötletet teszi felelőssé, holott egy megfelelő joystick lekérdezés, esetleg egy joystickcsere minden problémát megoldana, és jócskán megnövelné a játékkedvet is.

Nos vajon át lehet-e vinni ezeket a kritériumokat másféle játékokra? Hogy pontosabban fogalmazzunk, mire kell a kalandjátékoknál figyelni, ha ott is objektív kritériumokat szeretnénk bevezetni?

## Kalandjáték logika

Ha azok közé tartoznánk, akik szeretik magukat más személyek szerepébe képzelni, és eközben csavaros feladványo-

kat megoldani, vagy bátor hőstetteket véghezvinni, akkor nincs is másra szükség, mint egy jó adventure-re, de olyanra, amely az igényeinknek meg is felel. Egy kalandjátékban ugyanis mindezeket a dolgokat „per számítógép billentyűzet” lehet megtenni, ugyanakkor az a legjobb, ha teljesen bele tudunk merülni az alterego szerepébe. Minél jobb a kalandjátékban ábrázolt mesterséges világ, annál könnyebben lehet az abban fölvetett problémákat megoldani. De ezen kívül több más tényező is befolyásolja az adott kalandjáték minőségét.

Ahhoz, hogy egy kalandjáték magjának játékelméletét körülhatároljuk, olyan programokat mutatunk be, amelyek nem tartoznak ebbe a kategóriába. Így sokkal könnyebben lehet körüljárni, miről is akarunk a következőkben szólni.

## Mi is az az adventure?

Azok a játékok, amelyeket kizárólag a joystickkal kell irányítani, és a játékosról „valós” idejű cselekvést követelnek, minden bizonnyal más kategóriába tartoznak. De hova osszuk be akkor a legutóbbi időkben a piacon egyre növekvő számban megjelenő, úgynevezett action-adventure játékokat, vagy mit kezdünk azokkal a szöveges kalandjátékokkal, amelyek olyan játékbetétekkel rendelkeznek, amelyek „valós” időben játszódnak? A megvilágítás kedvéért néhány példát sorolnánk föl, amelyek az egyszerű kategorizálást extrém nehézé teszik. Olyan „klasszikus” programokat nézünk, amelyekkel valószínűleg mindenki találkozott már, így valóban tudni fogja, mit is akarunk mondani. A „Wizardry” például egy az említett akció-kalandjátékok közül, amelyben a főszereplőt a joystickkal kell a különböző jelenetek között mozgatni, hogy az ott esetleg föllelhető tárgyakat értelemszerűen alkalmazhassuk máshol. A Wizardry az egyik legszebb és egyúttal legnehezebb action-adventure. Nem arról van szó, mint a tiszta akciójátékok esetében, hogy a kisebb nehézségi fokozatú szintektől a nagyobbak felé haladunk. Itt például néha olyan helyszínekre is vissza kell térnünk, ahol egyszer már jártunk, mert csak így lehet a feladott problémát megoldani. Más fantázia-szerepjátékoknál, mint például az „Ultima” sorozatnál, a játékfigurát nem is a joystickkal kell mozgatni, hanem a billentyűzetten keresztül beadott irányadással. Például az Ultima IV esetében, amely az egyik legjobb

ban „fölszerelt” szerepjáték, relatív sok szót, néha egész mondatokat is be kell adni ahhoz, hogy a többi játékfigurával kapcsolatba lehessen kerülni. Azután még néhány kalandjáték, amelyek főleg szöveges beviteletet várnak el, mégis kiesnek a sorból. Aki például a „Critical Mass” játéknál nem reagál elég gyorsan, hogy kiugorjon a liftből, hamar elvérzik. Vagy a „Dragonworld” esetében, ahol a további játékhoz a szükséges pénzt egy kis ügyességi játékkal kell megkeresni. Három ilyen kis játék is van ebben a programban. Ezen kívül aki egy igazi 3D labirintusban nem igazodik el, mint amellyel az „Asylum” rendelkezik, az akár jobb ha bele sem kezd a játékba. Ezek a határesetek gyakorta jelentősen növelik a játék értékét, mivel új elemekkel rendelkeznek. A jövőben bizonyára találkozunk még hasonló jellegű játékokkal. De a kiindulási alaphoz vissztérve, a következőkben akkor fogunk adventure-ről beszélni, ha az adott játék a következő kritériumoknak tesz eleget:

- a játékek vezérlése legnagyobb részben a billentyűzetten keresztül történik;
- a szereplő egy képzelt játékfelület különböző helyiségeiben „szabadon” mozoghat, természetesen mindig az adott játéksituáció függvényében, tehát például a lezárt ajtókat ki kell nyitni stb..., mielőtt a következő helyiségbe jutnánk;
- a játékosnak van ideje a csavaros feladványokat megoldani;
- a játékot egy ültőhelyben nem lehet megoldani;
- a játék állását rögzíteni lehet valamilyen adathordozón.

Miután végre sikerült definiálnunk a kalandjáték fogalmát, máris kiesnek a most említett játékok, az Asylum, a Dragonworld és a Critical Mass. Tehát az is tény, hogy ha sok forgalmazó a terméket az adventure jelzővel reklámozza, amit nem tekinthetünk félrevezetésnek, végeredményben semmit sem mond arról, mit is rejt végül maga a program.

## Hogy mondom meg a számítógépnek?

Nos akkor hogyan tudja a potenciális adventure vásárló eldönteni, milyen objektív kritériumok alapján hozza meg döntését? Mik azok a dolgok, amelyek — eltekintve a játék ötletétől — befolyással vannak a kalandjáték minőségére? Nos egy kalandjáték úgynevezett parserje, te-



hát az a programrész, amelyik a játékos szöveges beviteleit kiértékeli, az adott játék egyik legjobban körüljárható része. Ezt a billentyűzet beviteli rutin összefüggésben két szempontból kell megvizsgálni:

Úgy van-e a program fölépítve, hogy az a játékos begépelte válaszait elég gyorsan fölfogja és kiértékeli? Hiszen egy mégoly izgalmas kalandjátéknál is bosszantó, ha a játékosnak állandóan ellenőriznie kell, hogy a program minden egyes leütött betűt elfogadott-e, azaz megjelenítette-e a képernyőn. A gyorsan gépelőket egy lassú parser jobban bosszantja, mint a kezdőket. Egyébként a tapasztalat azt mutatja, hogy éppen a kalandjátékok esetében növelik meg a beviteli sebességet. Ez egy olyan probléma, amely kizárólag a szerző programozói tevékenységére vezethető vissza. Ha ő egy jó, gépi kódban megírt programrutinnal dolgoztat, akkor azzal a mégoly villámkezdő játékosokat is kielégítheti. Viszont egy lassú modantkiértékelés úgy a kezdőknek mint a programozóknak sok fejfájást okozhat. Egy kalandjáték ugyanis a legtöbbször úgy van kialakítva, hogy a játékost órákig lekösse a számítógép előtt, és mindezt napokon és heteken keresztül. Egy túl lassan dolgozó parser tehát felér egy komoly tortúrával! Egy gyakorta azzal függ össze, hogy néhány kalandjáték akkora szókincsel rendelkezik, hogy bizony időnként a meghajtóban lévő lemezt kell segítségül hívnia. Ez pedig a 1541-es floppy esetében meglehetősen sok időt vesz igénybe. Itt tehát csak egy floppygyorsító lehet a megoldás, legyen az a programban mellékelve, vagy a felhasználó által hardveresen beépítve. Éppen ezért a vásárlás előtt az üzletben törekedni kell arra, hogy a játék ilyen vonatkozásairól is informálódjunk, mondjuk egy konkrét teszttel.

Intenzíven kell a kalandjátékkal akkor is foglalkozni, ha a következő szempont tekintetében is világosan akarunk látni. Ez pedig egy próbajáték alkalmával meglehetősen nehezen dönthető el. A második kérdés ugyanis úgy hangzik: Milyen intelligens a játék parserje és hány szót képes megérteni? Egy jól dokumentált program leírásában talán megtalálható, hányféle szót is ismer a kiszemelt játék, és talán azt is, mennyire pontosan érti azokat. Sajnos azonban a legtöbb esetben egyéb segítségre vagyunk utalva, mondjuk egy diszk- vagy egy gépi kódú monitorra, hogy betekinthessünk a programba, hogy megállapítsuk milyen szavakat is

ért meg az ténylegesen. Itt tehát legfőljebb a hasonló típusú kalandjátékokkal szerzett tapasztalatunk, vagy egy számítógépes lap részletes tesztje segíthet. Ilyenkor ugyanis abból lehet kiindulni, hogy amennyire komplex a parser, annál több örömet nyújt a program. A megnövelt szókincsel a játék sokkal közelebb kerül a realitáshoz.

Külön kell szólni a magyar nyelvű kalandjátékokról. Könnyű ugyanis angolul (még németül is) parsert írni, ahol nem kell az ékezetekkel bajlódni, és a prefixekkel az összes esetet ki lehet fejteni. Lehet hogy sokakat nem zavar, másoknak talán a játéktól is elveszi a kedvét, hogy (a programozó legjobb szándéka ellenére is) gyógyos gyerekként kell a kezeléssel bajlódni, úgymint inni víz, kutat szoba, átúsz folyó. Valószínűleg kell még várni az olyan magyar parserre, mint amilyen az angol The Guild of Thieves játékban van: Ez olyan komplex mondatokat is megért, mint a „Go into the kitchen and find the bag...”

## Szöveg vagy grafika

Az, hogy nekünk megfelel-e a csak szöveges kalandjáték, esetleg a grafikus támogatást elhagyhatatlannak tartjuk, csak a saját ízlésünktől függ. A program értékelését tekintve ezért csupán néhány megjegyzést tennénk. Egy kizárólag szöveges kalandjáték esetében fontos, hogy jól kiismerjük magunkat a játék parserjének „anyanyelvében”. Bár vannak már jó szöveges kalandjátékok német nyelven is, gondoljunk csak a „Lapis Philosophorum” játékra, amelyet kivételesen lefordítottak angolból, mégis az ilyesfajta szóalkotás legnagyobb része csak angolul élvezhető. Viszont a tisztán szöveges játékok, amelyek legismertebb szerzője az amerikai Infocom cég, rendelkeznek a legnagyobb szókincsel és a legtöbb jól felszerelt helyiséggel, játékszituációval. Ezzel szemben ugyanis a grafika viszonylag nagy tárolóterületet igényel, legyen az a gépben vagy a diszken. Cserébe viszont sokat ad a játékosnak azzal, hogy őt vizuálisan támogatja. Amennyiben pedig az még jól is van megcsinálva, még emeli is a játékkedvet és a játék minőségét is. Jópár olyan kalandjáték létezik, amelyek több lemezoldalra vannak elosztva, így sok képet és szöveget adnak a felhasználónak. Ilyen játék a rendkívül bonyolult és izgalmas „Fahrenheit 451” program. A játék komplexitását és változatosságát te-

kintve a Tellarium játékokat nehezen lehet túlszárnyalni, mialatt a „Borrowed Time” a kifinomult képernyőfelosztás és a menütechnika vonatkozásában adott újat. A „The Pawn” az a program, amelyik a C64-re írt kalandjátékok esetében új mércét állított fel, mind a parser komplexitását, mind a megjelenített grafikát illetően. Újabb mérföldű a már említett Guild of Thieves, és persze napjaink slágere a Zak McCracken. Amit a jövőben az adventure szektorban újdonságként várni lehet, az animációs grafika illetve jelenetek 3D ábrázolása. A programozók újabb ötleteit tehát semmilyen határ nem korlátozza.

## Összefoglalás

A számítógépes iparágban jelentkező örült gyors fejlődés — hogyan is lehetne ez másként — megtalálta az utat a játékszektorba is. A jövőt tekintve 8 bites kisgépeken is az RS232 illesztésen keresztül összekötött két számítógépre írt játékok fognak megjeleníteni, amelyek lehetővé teszik a szimultán játékot. Néhány játékprogram esetében, 16 bites gépeken máris találhatunk ilyen megoldást. Így két játékos játszhat ugyanazzal a játékkal, holott mondjuk egymástól többszáz kilométer távolságra vannak. Dolgoznak a házi számítógépek forradalmian új képteknikáján is, ahol szintén nagy javulások várhatók. Ezt a feladatot a video és a számítógép összekapcsolása fogja biztosítani, ahogy azt a fejlesztők jelenlegi elképzelései alapján állítani lehet.

Attól függetlenül, hogy milyen fantasztikus újdonságokkal fogunk majd találkozni, ha most vagyunk abban a szituációban, hogy szükség van egy játék értékelésére, akkor remélhetőleg ez a cikk nyújt valamilyen segítséget. Mert látatlanba ítélni a hangzatos címek, játékkermi ismeretségi fok, vagy a levegőben szállongó „hírek” alapján nem igazán ajánlatos. Sokkal inkább hasznos lehet a vásárlás előtt elolvasni a szaklapok játéktesztjeit, ismertetéseit. Ezzel előbb el lehet kerülni a vásárlás utáni bosszankodást. Ugyanakkor minden esetben ragaszkodjunk egy, a helyszínen elvégzett tesztjátékkal könnyen megszerezhető kiindulási tapasztalatszerzéshez, amivel szintén eldönthetjük, hogy a játék tényleg megfelel-e. Összességében reméljük, hogy a cikkünk elolvasása után talán kedvet kaptunk arra, hogy időt szakítsunk egy játékprogramra. Nos ehhez sok sikert kívánunk.



# GEOS

## Az installáció titka

A GEOS installálás semmi más, mint egy aktív másolásvédelem. Az aktív ebben az értelemben azt jelenti, hogy a felhasználónak magának is együtt kell működnie. Szerencsére azonban ezt a műveletet csak egyszer kell elvégezni, s az így inicializált lemezzel utóbb problémamentesen dolgozhatunk. Azoknak a felhasználóknak, akik update keretében újabb GEOS verzió(kat) szereztek be, az új lemezeket is illeszteniük kell. Kissé rémisztő persze az az utalás, hogy a hibásan installált lemezeket nem cserélik ki.

De mi is történik az installálás során? Lényegében annyi, hogy egy véletlenszámot generálunk, amelyet azután beírunk a legfontosabb alkalmazásokba és a lemezzre. A GEOS azután a betöltés során a bootlemezen található véletlenszámot átveszi, és összeveti azt az alkalmazás által adott számmal. Ha ezek egyeznek, a töltés megtörténik, ellenkező esetben nem. Az „újbootolás szükséges, használjuk azt a lemezt amit az inicializálásnál” üzenetet kapjuk. A véletlenszámot sorozatszámának szokás nevezni, de ez természetesen nem egyezik meg az eredeti lemez címkéjében álló gyár(tás)i számmal.

Aki a GEOS 1.3-ról a 2.0-ra száll át, annak az alábbiakat mondhatjuk. Amikor a rendszer megkérdezi, hogy „az adott alkalmazás egy, korábban már installált alkalmazás-e” feltétlenül igennel (német változatban JA, angolnál YES) válaszoljunk! Csak ebben az esetben ugorjuk át az új véletlenszám generálás rutint és vesszük át a régit. Mivel az installálást nem lehet visszacsinálni, rendkívül figyelmesen kell eljárunk. Mindez azt is jelenti, hogy az installált programokat

nem lehet másnak eladni! Az új tulajdonos ugyanis az eredeti rendszerlemez (amivel az installálás történt) nélkül az alkalmazást nem tudja használni.

## Kicsiből nagyot és vissza

Gyakran fordul elő, hogy egy többoldalas szöveg betűméretét utólag módosítjuk. Például kicsinyítjük, hogy a terjedelm csökkenjen. Erre a célra a „Seite waehlen” (oldalválasztás) menüpont a legalkalmasabb a Geowrite-nál. Itt két lehetőségünk van. Vagy az első laptól előre felé dolgozzuk föl az oldalakat az utolsóig, vagy fordítva. Első pillantásra a két módszer ugyanazt takarja. A második módszer azonban sokkal jobb! Mert ha a szöveg elején csökkentjük a méretet, a lap alján hely szabadul föl, mire oda szöveg kerül. Ez nemcsak időtrabló művelet, de megvan az a veszélye is, hogy a GEOS az „áthúzott” szövegrészt a kicsinyítésből kihagyja. A nagyításnál (pl. 12-ről 24 pontra) ugyanez az eset.

## A GEOS CP óra

Végre kapható egy akkumulátorosan védett, és a GEOS-hoz illesztett valósidejű órákártya, amely nem foglalja le a user portot. Így az is, aki párhuzamos kábelről működteti a nyomtatóját, használhatja a panelt. Az óra az időt, a dátumot, és a napnevet tartalmazza, és egyszerűen a kazettaportra kell csatlakoztatni. A szökökvek fölismerése automatikus. Az évszámot is tárolja az építőelem, amelyet a fordulónál természetesen korrigálunk.

A NYÁK lapon van egy üres foglalás is. Ez a későbbi bővítések számára van

főntartva. Például egy 256 byte-os tárolóelem számára. Egy megfelelő program segítségével kapcsolóóra építése is lehetővé válik. De az óra, a gyártó szerint más készülékek illesztésére is használható. A panelhez mellékelt programot rá kell másolni a bootlemezzre. A betöltés így automatikus. A GEOS Desktop indításakor az aktuális dátum és idő „magától” a helyére kerül. A modul az 1.3-as változattól fölfelé használható. A GEOS CP óra a programmal együtt 69 márkába kerül.

## Desktop egér és joystick nélkül

Tulajdonképpen a GEOS egér és joystick nélkül elképzelhetetlen. Ez így helyes, de ettől függetlenül a GEOS 2.0 Desktopban a legtöbb funkció a billentyűzetről is elérhető. Ez pedig gyorsabb mint az egérrel való aktiválás. Például az egyes állományokat a (C= X) (x=számjegy) segítségével szelektálhatjuk, a (C= Z)-vel indíthatjuk és a (C= P)-vel nyomtatjuk.

## A pont a lényeg

Némely írásképnél, pl. LW-Roma, LW-Barrows, LW-California előfordul, hogy a mondat végén álló pont „elvész”, mivel túl közel áll valamely más karakterhez. Egy apró trükkel azonban ezt a hibát elkerülhetjük. Jelöljük meg a pontot az egérrel. Ez meglehetősen precíz vezérlést kíván. Eredményként egy keskeny inverz mezőt kapunk. Ezt követően a stflust a (C= I)-vel dőltre (kurzív) változtatjuk. A pont máris lényegesen jobban kiemelődik.

# Lexikon

## V. rész

### Állomány

Az állomány összetartozó információk gyűjteménye, amelyek együtt vannak letárolva; adathalmazként is szokták nevezni. Az állomány angol neve „file”. A számítógép számára az állomány semmi mást nem jelent, mint egy igazgatási egységet. A számítógép az adathalmazt egy meghatározott formában rögzíti a lemezen vagy a kazettán. Például a CP/M állományok egy lemezen olyan rekordokból állnak, amelyek 128 byte hosszúak. A 1541-es Commodore floppy ugyanezt 256 byte-os blokkokra osztja el.

A felhasználó számára fontos megkülönböztetési kritérium az állományokkal kapcsolatban az információkhoz való hozzáférés módja. Főleg kétféle hozzáférés között teszünk különbséget, éspedig a soros (szekvenciális) és a relatív hozzáférés, illetve így a szekvenciális és a relatív állományok között.

### Szekvenciális állomány

Ez az adathalmazok egy olyan szervezési formáját jelzi, ahol az egyes rekordok az információt hordozó felületre egymás után kerültek felírásra. Ezt az eljárást minden olyan esetben alkalmazzák, amikor az állomány olyan információkat tartalmaz, amelyeket mindig „előlről” „hátrafelé” kell olvasni (például programok esetében). A soros állományú adatok esetében hátrány, hogy egy bizonyos információra nem lehet tetszőlegesen hivatkozni, hanem mindig „libasorban” kell keresni mindaddig, amíg azt el nem érjük. Természetesen a hátrébb található adatok megtalálásához több idő kell, mint az elől állókéhoz. Ezen kívül hátrány az is, hogy új információkat csak a legutolsó rekord mögé lehet fölírni. Ha az adatokat szalag-

ra akarjuk kimenteni, akkor csak soros állomány felírására van mód.

### Lemezszervezés

Ahhoz, hogy adatokat írassunk a lemezre, vagy onnan beolvassunk azokat, a DOS-nak föl kell ismernie az író/olvasó fej fizikai helyzetét. A pozícionálás meghatározásához szükséges jelölést a DOS el is helyezi a lemezen, ezt a folyamatot nevezzük „formatálásnak”.

A 1541-es Commodore floppy a lemezt a formatálás alatt 35 koncentrikus sávra (angolul track) osztja föl. Minden egyes sáv további részegységekre, az úgynevezett szektorokra (angolul sectors) oszlik. A 1541-es formátumban az egyes track-ek (attól függően hogy milyen messze vannak a középponttól) 17–21 szektorral rendelkeznek, amelyek közül minden egyes szektor 254 adatbyte-ot képes tárolni. A Commodore esetében a szektorokat blokkoknak (blocks) is szokták nevezni.

### Lemezmonitor

A diszkmonitor egy olyan program, amelyik lehetővé teszi a felhasználónak, hogy közvetlenül a lemezen manipulálhasson.

Az elv a következő: A monitor programnak megadjuk a kívánt blokk címét, a legtöbbször hexadecimális formában. Például annak a blokknak az esetében, ahol a BAM áll (track 18, szektor 0) ez \$12 és \$100. A lemezmonitor azután beolvassa a kívánt blokkot közvetlenül a lemezről a RAM-ba. Ezután azt ki lehet listázni, meg lehet nézni, meg lehet változtatni, és természetesen (a megváltoztatott formában) vissza lehet írni a lemezre. Például a megfelelő byte-ok megváltoztatásával a BAM-

ban vagy a tartalomjegyzékben egész lemezeket és állományokat lehet törlés vagy fölülírás ellen védeni. Ily módon programok megváltoztatására is mód van.

### Közvetlen hozzáférés

A hozzáférés (angolul access) az a mód, amellyel egy adathordozó egyes tárolóhelyeit „felkeressük”. A Commodore meghajtók esetében a közvetlen hozzáférés alatt azt kell érteni, hogy a lemez egy blokkját közvetlenül el lehet érni anélkül, hogy be kellene olvasatni azt az állományt, amelyben az adott blokk áll. A Commodore meghajtók DOS-ja egy sor olyan parancsot ismer, amellyel a közvetlen hozzáférés viszonylag könnyen megvalósítható. A programozásnál azonban rendkívül óvatosan kell eljárni, mivel a DOS síkján a parancsok nincsenek ellenőrizve: Ha például arra utasítjuk a DOS-t, hogy olvasson be valamit a 95. sávról, akkor az meg is kísérel az író/olvasó fejet oda vinni, holott ez a sáv egyáltalán nem létezik.

### Relatív állomány

A relatív adattárolásnál abból indulunk ki, hogy minden egyes rekord azonos hosszúságú. Minden egyes bevitelhez, amely méretét a felhasználó korábban definiálta, hozzárendeljük a rekordot, amelyekre azután a későbbi felhasználáskor közvetlenül hivatkozhatunk. Gyakorta egy plusz soros állományú adathalmazban, az úgynevezett „indexhalmazban” tároljuk, hol található (fizikailag) egy meghatározott rekord a lemezen. Ily módon rendkívül gyorsan hozzá lehet férni a kívánt adatokhoz, mivel az index halmaz állandóan benn állhat a számítógép RAM-jában anélkül, hogy sok helyet foglalna el. Ezen kívül ez alapján igen gyorsan lehet szortírozni, mivel nincsen szükség a tulajdonképpeni adatok, csak az index halmaz bizonyos kritériumok szerinti rendezésére.

### DOS (Disk Operating System)

Ahhoz, hogy egy számítógép egyáltalán képes legyen egy nagy tömegű adatok kezelésére szolgáló egységet kezelni, szüksége van egy speciális operációs rendszerre. A DOS egy számítógép „lemez operációs rendszere”. Sok számítógép esetében a háttértárolókat bővítőpanelnélkül külön kell megvenni, ahol a DOS-t rendszerint mellékelik is.

A Commodore meghajtók különleges kategóriába tartoznak, mivel ezek már a gyártáskor „intelligensé” lesznek téve, ami annyit tesz, hogy rendelkeznek egy saját, fixen beépített operációs rendszerrel, amelyre BASIC-ből hivatkozni lehet. Ez a módszer azért előnyös, mert így nem foglalunk le tárolóterületet. Ezen kívül a meghajtó képes egy időben a géppel, de attól függetlenül lemezmanipulációkat végrehajtani (például egy lemez VALIDATE-ja).

### BAM (Block Availability Map)

Azaz: blokk foglaltsági térkép. A BAM egy olyan tartalomjegyzék a lemezen, ahol föl van tüntetve az adott lemez foglalt és szabad blokkjainak száma. Minden egyes blokk rendelkezik egy jelzőbittel. Ha az logikailag 1, akkor a megfelelő blokk szabad. Ha viszont a lemezre valamit írunk, az nyilván elfoglal egy

vagy több blokkot is. Hogy a DOS tudja, mely blokkok foglaltak, ezért azok jelzőbitjei a nulla értéket kapják.

A BAM a Commodore 1541-es és 1570-es meghajtók esetében a 18-as track 0. szektorában, a 1571-es esetében a 18-as és az 53-as track 0. szektorában áll. A BAM csupán a felét foglalja el a blokknak, a többi rész a lemez nevének, ID-jének stb. rögzítésére szolgál. A 18-as sáv maradék részét a tartalomjegyzék (directory) foglalja el.

## NOVOTRADE—2C Kft. ÁRLISTA

### Hardverek

C64 alapgép	14 600 Ft
VC 1541 drive	16 600 Ft
Datasette	3 000 Ft
Amiga 500 alapgép	49 990 Ft
Amiga 2000 alapgép	125 600 Ft
Amiga mouse	4 000 Ft
C 1084S sztereómonitor	32 000 Ft
C 1802 monitor	25 000 Ft
C64 mouse	3 500 Ft
Amiga RF modulátor	3 900 Ft
Amiga tárbővítő (512 Kb)	8 000 Ft
Amiga AT kártya	47 675 Ft
Amiga digitalizáló	29 700 Ft

### C64 játékok neve

	Kazetta	Lemez
Chamonix Challenge	499 Ft	549 Ft
Eddie Edwards Super Ski	499 Ft	549 Ft
Eszkimó	345 Ft	—
Hostages	549 Ft	599 Ft
Impossible Mission II	581 Ft	668 Ft
Diamond/I Want More	—	549 Ft
Nautilus	399 Ft	—
Ninja Testvérek	399 Ft	—
Operation Neptune	—	599 Ft
Prohibition	499 Ft	—
Rettenhetetlen	390 Ft	—
Rolling Twins	399 Ft	549 Ft
Sim City	—	599 Ft
Smaragdvár	345 Ft	—
Space Knight	340 Ft	—
Space Racer	—	549 Ft
Tin Tin On The Moon	—	599 Ft
Warlock Quest	499 Ft	549 Ft
Waterpolo	450 Ft	—
Xonox	399 Ft	499 Ft

### Hardverkiegészítők

2 RCA kábel	410 Ft
3 RCA kábel	575 Ft
Hálózati kábel	480 Ft
Adatkábel (soros bus)	360 Ft
Antennakábel	340 Ft
Antenna váltókapcsoló	490 Ft
C64 Euro-kábel	685 Ft
Amiga Euro-kábel	1250 Ft
Ékezetes Eprom SP-180	2545 Ft
Ékezetes Eprom MPS 1230	3500 Ft
Mikrokapcsolós joystick	1000 Ft
Műszerész porszívó	490 Ft
C64 tápegység	3500 Ft
1541 tápegység	2700 Ft
14" monofilter	990 Ft
14" colourfilter	1100 Ft
Lemeztartó 3 db-os 5,25"	99 Ft
Lemeztartó 50 db-os 5,25"	700 Ft
Lemeztartó 100 db-os 5,25"	820 Ft
Lemeztartó 10 db-os 3,5"	160 Ft
Lemeztartó 40 db-os 3,5"	700 Ft
Lemeztartó F80 db-os 3,5"	850 Ft
Lemeztartó 140 db-os 3,5"	1300 Ft
Mouse pad	245 Ft
Festékszalag FX-1050	675 Ft
Festékszalag MPS-1230	700 Ft
Display monitorszemüveg	750 Ft

**Viszonteladók! Felhasználók!**

*Dyras festékszalagok  
nagy választékban és olcsón  
kaphatók!*

# Még többet ésszel!

## II. rész

Van három olyan formai megoldás, amellyel programunkat olvashatóbbá tehetjük. A legfontosabb a kellő számú, körültekintően elhelyezett REM utasítás, más néven komment(ár). Ha egy programot ügyesen és strukturáltan írtunk meg, akkor biztosan fel lehet azt osztani legfeljebb 20–25 soros, logikailag önálló szakaszokra, modulokra. Ha minden ilyen szakasz elé egy REM-et szúrunk, röviden utalva a modul feladatára, akkor a program meglepően áttekinthetővé válhat. Érthetetlen, hogy az újságban is gyakran látható szóözőnök gazdái hogy találnak meg azonnal egy kiválasztott rutint, mikor a javítást végzik.

Ha ráadásul a szubrutint eltérő módon jelöljük meg, akkor elkerülhető az, hogy GOTO utasítással lépünk rá, aminél nagyobb marhaságot nem nagyon tudok elképzelni. Ajánlom az 1. példán látható megoldást, nekem beválik. Egy készülő programnak rendszerint a vázát írom meg először. Az új szubrutinokat is előbb elhelyezem (ld. 2000), ettől máris szabályosan meghívhatók, azaz a program futtatható.

Megjegyzem, szerencsétlen dolognak tartom, amikor a hivatkozások a REM-ekre mutatnak. A futó változathoz takarékosági okokból néha ki szokás a kommenteket törölni, akkor pedig a program nyomban tele lesz hibával.

A második javaslat: a program időnkénti fellazítása. Íráskor kényelmetlen dolog ügyelni a szép, szóközökkel tagolt alakra. A tárgiénny és a gyorsaság szempontjából jó a takarékos, tömör forráskód. Olvasáskor viszont szemet bántó a tömörség, fárasztó a változók és az utasítások szétválasztása. A Pötyögő-listán 267. számmal megtalálható program jól használható mindkét esetben.

A harmadik javaslat: a kész program egyenletes átszámozása. Nem nélkülözhetetlen, de valahogy jó benyomást kelt, és ez azért nem teljesen lényegtelen.

A múltkor belekezdtem a strukturált programozás módszerének egy nagyon vázlatos ismertetésébe. Ismétlésül a lényeg: a feladatot egyre kisebb, önálló részekre bontjuk, eljutva végül egy olyan állapothoz, ahol minden részt pontosan megoldhatunk egy elemi programmal. Ezek összeillesztve megoldják az eredeti feladatot. (Kérem azokat, akik ismerői a témának, bocsássák meg a szabados átfogalmazást.) A módszer minden magas szintű nyelven követhető, ezért szinte kötelező. Az olvashatóság az elemi felosztásból következik, amelyet egyéb módszerekkel fokozhatunk, lásd fent.

Megneveztem a program három alapelemét, de nem említettem a ciklust, amelyet mindannyian használunk. Megjegyzem, a módszer klasszikus elmélete szerint a program szekvenciából, elágazásból és ciklusból áll, a BASIC-hez azonban ez nem áll elég közel. Amíg ugyanis az elágazás valahogy megoldható az IF utasítással, ciklus-utasítás nincs. A FOR ne tévesszen meg senkit, az csak a ciklus egy fajtájára jó.

Ciklus az a programszakasz, amelyet egy feltétel teljesülésétől függően meg kell ismételni. Sok nyelvben értelmes lehet az, ami a 2. példában látható. Az első sor magyarul szó szerint ezt jelenti: csináld, amíg  $F=0$ , a következőt... A LOOP a visszalépés utasítása. A ciklusmagban az I értékét kiírjuk, megnövel-

jük, és beállítjuk az F új értékét. Az ' $F=0$ ' a ciklus feltétele, azaz amíg ez teljesül, a ciklusmag végrehajtódik.

Ez a mini program nem BASIC nyelven készült. Ilyet azonban mi is tudunk, nem igaz? Csak át kell fogalmazni kissé a ciklus lényegét. Tehát: ha az ' $F=0$ ' feltétel igaz, akkor hajtsd végre a ciklusmagot, majd kezd elölről. Ez egy közönséges feltételes elágazás, aminek a BASIC megfelelője az IF (ld. 3. példa).

Nagyon szép, csak hogy jó lenne, ha a ciklusmag tetszőleges hosszú lehetne, nem csak annyi, amennyi egy IF-be elfér. Ez a feladat megegyezik a következővel: ha ' $F=0$ ' igaz, akkor ciklusmag és visszalépés, egyébként ugorj a visszalépés utáni pontra (4. példa). Még mindig nem jó, a C64 nem ismeri az ELSE (azaz 'egyébként') utasítást, amiért sokan joggal is szidják. De nézzük csak: ha ' $F=0$ ' nem igaz, akkor átlépés, egyébként ciklusmag (5. példa). Ez már pontosan megoldható a legprimitívebb BASIC-ban is, és tökéletesen megfelel a leírt feladatnak. Részletkérdés, hogy a ' $\text{NOT } F=0$ ' helyett az ' $F \neq 0$ ' az ésszerű.

Hát igen, megoldottuk ugyan a problémát, de ehhez ugranunk kellett a programban. Azok, akik olyan nyelven dolgoznak, ahol van DO, vagy hasonló ciklus-utasítás, gyakorlatilag nem szorulnak az ugrásra, és ezért állandóan nyafognak, ha egy GOTO utasítást látnak. Néha van ám valami igazuk. Az a baj ezzel a fránya GOTO-val, hogy teljesen fejtellenül átléphetek vele minden szerkezeti határt, és akkor aztán lőttek a strukturálásnak is meg az áttekinthetőségnek is. Ha viszont csak ilyen elemi szerkezetek részeként használjuk, előre megtervezett módon, akkor ezzel valójában a nyelvet egészítettük ki jellegzetes elemekkel, a módszer kívánalmi szerint. Az ilyen GOTO igazából a strukturálás része.

Azért van valami lehetőség az ugrádozás minimalizálására. A 3. példánál elmondottakat úgy hajthatjuk végre tetszőleges méretű ciklusmaggal, hogy azt egy szubrutinba helyezzük át (6. példa). Mivel a szubrutin az eljárás (lásd alapelemek) egy nehezebb megfelelője (BASIC-ben az egyetlen), ezért a megoldás csökkenti a program elemeinek a számát, ami jó. Nem jó viszont, hogy ez több helyet igényel, és a hívás külön időbe kerül. A műveleti időt, főleg egy interpretált nyelvnel, nem lehet figyelmen kívül hagyni. Lehet választani.

Van a 3. példában még egy nagyon fontos adalék, az 1. sor. Ha egy ciklust megírunk, amely például egy változó értékétől függően működik, akkor ennek a változónak megfelelő kezdőértékről kell gondoskodnunk. Ha ez elmarad (tipikus hiba), akkor a program működése látszólag határozatlan lehet. Persze nem kötelező minden ciklus előtt kezdőértéket adni, de kötelező gondolni rá.

Hadd mondjak még valamit a 2. példáról. Furcsa látvány lehet a BASIC-hez szokott szemnek a szokatlan tördelés. Kialakult hagyomány a strukturált elemekre épülő nyelvekben, hogy a ciklusban és elágazásban a logikailag alárendelt sorokat — mint a ciklusmag — kissé beljebb kezdik. Ennek pusztán a program még olvashatóbbá tétele az egyedüli célja, és nem is kötelező. Vannak, akik ezt a többszintű tördelést összetévesztik magával a strukturálással, és vannak, akik ezt a BASIC-ben is megpróbálják meghonosítani. Mivel az elterjedtebb BASIC rendszerek szövegszerkesztői ezt a tördelést általában nem támogatják,



ugyanakkor a programok enélkül is elég világossá tehetők, ezért én feleslegesnek tartom, és a példákban sem alkalmazom.

Az eddigi példák ciklusai az ún. előtesztelő fajtából valók voltak. Ez annyit jelent, hogy a feltétel vizsgálatát a ciklusmag előtt végezzük el. Lehet, hogy a ciklus egyszer sem fut le. A 7. példa egy hátulatesztelő ciklust mutat be, amely biztosan lefut legalább egyszer. Mindig a feladat szabja meg, hol melyiket alkalmazzuk. A 8. példával mindenki kipróbálhatja, hogy a saját rendszerében a FOR melyikhez tartozik. Egyébként néha szükség van belületesztelő ciklusra is, ahol a feltétel-vizsgálat (és kilépés) a ciklusmagban van.

A 9. példán az előző FOR utasítás hátulatesztelő behelyettesítése látható, hiszen ezt is meg lehet oldani elemi programokkal.

Sajnos már nincs hely a másik kérdés (kilépés a ciklusból) megtárgyalására, majd legközelebb. Azért felteszek két újabb kérdést is:

— a 10. példa egy megjelent programból van szabadon kiemelve. Célja a csillag kiírása a DATA után, az S értékétől függetlenül azonos helyen. Több helyen hibás. Hol?

— hogy lehetne ezt egyszerűbben megoldani?

Ha valaki egy érdekes problémára akadt, vagy valami kérdése van az elmondottakról, netán javasol egy témát, a szerkesztőség címére megírhatja.

Hódi Gyula

# Még többet ésszel!

```
(1.)
100 REM *** PROGRAM NEVE *
110 REM (C) SZERZO ES DATUM
...
239 REM ----- MENU
240 PRINT "*****": ...
...
1999 REM ===== EDITOR
2000 ;;
2090 RETURN
...
2999 REM ===== GET
3000 WAIT 197,64: ...
```

```
(2.)
DO WHILE F=0
PRINT I
I:=I+1
GET F
LOOP
```

```
(3.)
1 F=0
2 IF F=0 THEN PRINT I: I=I+1: GET F: GOTO 2
```

```
(4.)
2 IF F=0 THEN 3 ELSE 5
3 PRINT I: I=I+1: GET F
4 GOTO 2
5 ...
```

```
(9.)
1 I=1
2 PRINT I
3 I=I+1: IF I<=0 THEN 2
```

```
(5.)
2 IF NOT F=0 THEN 5
3 PRINT I: I=I+1: GET F
4 GOTO 2
5 ...
```

```
(6.)
2 IF F=0 THEN GOSUB 100: GOTO 2
3 ...
100 PRINT I: I=I+1: GET F
101 RETURN
```

```
(7.)
2 PRINT I: I=I+1: GET F
3 IF F=0 THEN 2
```

```
(8.)
FOR I=1 TO 0: PRINT I: NEXT I
```

```
(10.)
0 S=INT(RND(1)*13000)+20
30 IF S<100 THEN Z#="||"
31 IF S>100 AND S<=1000 THEN Z#="|||"
32 IF S>=10000 THEN Z#="||||"
34 PRINT "*****";S;"|||";Z#;"*"
99 WAIT 197,63: GOTO 0
```

Áttérés COMMODORE-ról IBM gépre

# VÁLTÓ

Mottó:

Minden gépnél van jobb

## PCTOOLS ÉS NORTON COMMANDER

### PcTools

A DOS SHELL-ek közül ez fejlődött a legnagyobbra, legalábbis méretben. Legutolsó verziója mintegy 6 Megabyte, így használhatóságának a winchester tárolókapacitása szab határt. A PcTools ezt megelőző verzióját (verzióit) azonban sokan használják, segédprogramként vagy állandó DOS SHELL-ként. Ez a verzió a PC Shell névre hallgat, és egy univerzális lemezmatató program. Kezelése legördülő menükben történik. A menüsört a képernyő legfelső sorában látjuk, a menübe az (ALT) vagy a F10 billentyű lenyomásával juthatunk. A második sorban a lehetséges lemez meghajtók vannak felsorolva, míg a képernyő legalsó sorában egy rövid magyarázatot találhatunk az aktuális menüponthoz. A képernyő fentmaradó felülete két részre van osztva, a bal oldalon a faszervezetet látjuk — itt lehet a különböző alkönyvtárak között közlekedni —, a jobb oldalon pedig az aktuális alkönyvtárban szereplő bejegyzések találhatók. A két ablak között a TAB billentyűvel közlekedhetünk, az aktuális ablakot a keret megvastagítása (kétszerezése) jelzi. A főmenü menüpontjai szinte megegyeznek a PathMinder menüpontjaival:

FILE — egyes bejegyzésekkel, vagy azok egy csoportjával végezhető műveletek. A lehetséges műveletek választéka azonban sokkal bővebb, mint a PM esetében, például közvetlenül kereshetünk ASCII karakter sorozatot vagy byte-sorozatot a kiválasztott file-ban.

DISK — teljes (halékony- vagy merev-) lemezre vonatkozó parancsok, a másolástól az alkatalógusok ide-oda helyezgetéséig minden, ami csak szóba jöhet.

OPTION — az opciók állítása, a saját

menü megszerkesztése—módosítása, a beállítás kimentése.

APPLICATIONS — Az alkalmazói menü. A PcTools saját külső alkalmazásai automatikusan bekerülnek.

SPECIAL — különleges lehetőségek. Teljes rendszerinformáció, törölt bejegyzések visszaállítása, lemeztérkép, file-térkép, katalógusrendezés és memóriatérkép.

HELP — Egy bővebb segítség, minden menüponthoz, tevékenységhez. Az F1 billentyűvel is aktivizálható.

A menüpontokat rámutatásos módszerrel és kiemelt betűvel is elérhetjük. Az aktuális meghajtót (CTRL) + (meghajtó betűjele) lenyomásával változtathatjuk meg. A PcTools már lehetőséget ad a DOS közvetlen használatára is, azaz ki lehet lépni a programból úgy, hogy a megfelelő DOS műveletek végrehajtása után az EXIT beütésével visszakerüljünk a PcTools-ba. (Ideiglenes kilépés, (shift) + (F9))

A PcTools 5.01-es (PcShell 5.01) verziójának beépített szövegszerkesztője a Word Processor. Ezzel módosíthatjuk a szövegfile típusú állományainkat, vagy ezzel készíthetünk újakat (TXT, DOC vagy BAT állományok).

### Sokak szerint a király: NORTON COMMANDER

A Norton Commander (továbbiakban NC) talán a legkönnyebben kezelhető DOS SHELL, melynek nagy erőssége a másolás. Maga a program csak egy része PETER NORTON és társai segédprogramjainak, de a mindennapi használat során felmerülő igényeket szinte 100 száza-

lékosan kielégíti. A programcsomag további részei:

NORTON INTEGRATOR — sok kis hasznos segédprogram gyűjteménye.

NORTON EDITOR — Nortonék szövegszerkesztője.

NORTON UTILITIES — Lemezkarbantartó, javító.

NAV — NORTON ANTI VIR — A legújabb Norton-féle programcsomag, vírusmegelőzés, keresés, írtás.

Mi most az NC 3.0-ás verzió alapján mutatjuk be a program működését. Az NC indítása után egy vagy két ablakot nyit a képernyőn, melyek alatt a kurzor ugyanúgy látszik, mintha csak a DOS-ban lennénk. Ebben a sorban az eddig megismertek szerint tetszőleges DOS parancsot kiadhatunk, mintha nem is lennénk benne az NC-ben. Az ablakok nagysága (18 vagy 10 sor, illetve 20 vagy 12 sor) szintén állítható. A két sornyi eltérés onnan adódik, hogy az ablak alatt létrehozhatunk egy egysoros, úgynevezett MINI STATUS képet, ahol arról a bejegyzésről láthatunk bővebb információt, amelyiken az inverz (vagy színes monitoron más színnel) jelölt programkurzor áll. Az ablak három vagy négy hasábra osztott, ha teljes információt kérünk, akkor négy hasábrban a következőket látjuk: bejegyzés neve, hossza, keletkezés dátuma, ideje. A MINI STATUS mindig ilyen formátumú. A háromhasábrs ablak esetén csak a bejegyzés neve látható, de ilyenkor egy akár 59 bejegyzésből álló alkatalógust is láthatunk egyszerre. Az ablakban közlekedni a kurzorvezérlőkkel lehet, az ablakok között a TAB lenyomásával válthatunk, ha mindkét ablak ki van téve. Ablakot kitenni levenni a következő billentyű-kombinációkkal lehet:

(ALT) + (F1) illetve (ALT) + (F2) — A bal illetve jobb oldali ablak kirakása, ha nem volt kint, illetve az adott ablakban az aktuális meghajtó megváltoztatása.

(CTRL) + (0) — Ablakok ki- illetve bekapcsolása, mindig az előző állapot szerint. Tehát ha egy ablak van bekapcsolva, azt kikapcsolja, ha nincs bekapcsolt ablak, azt (vagy azokat) kapcsolja be, amelyek utóljára be volt kapcsolva.

(CTRL) + (P) — A nem aktuális ablak ki- illetve bekapcsolása.

Az ablakon belül, az alkatalógusok váltása is igen egyszerű, ha egy katalógusneven állva (enter)-t nyomunk, belép az adott katalógusba, és minden alkatalógusban kijelzi a .. állományt, melyen (enter)-t nyomva visszaléphetünk az egy szinttel feljebbi katalógusba.

A képernyő alsó sorában (ha nincs kikapcsolva) egytől tízig számozott téglalapokat látunk, bennük felirattal. Ez egy-

szertű segítség, a funkcióbillentyűk felsorolása. Ha nincs átállítva, akkor a következőket láthatjuk:

1 — HELP Egy nagyon jó HELP állomány. Az állomány minket érdeklő részét a kurzorral kiválaszthatjuk, (enter) leütésére már olvashatjuk is. Gyakorlatilag egy teljes kézikönyv szövege a HELP állomány. A régebbi verzióknál itt a billentyűzetkezelésre vonatkozó HELP-et találunk.

2 — MENÜ Általunk definiált felhasználói menü jelenik meg a képernyő közepén, választani rámutatásos módszerrel, vagy billentyűnyomással (első betű, illetve funkcióbillentyű) lehet.

3 — VIEW Betekintés a kiválasztott bejegyzésbe. Katalógusnév esetén nem működik. A listát több szövegszerkesztő formátuma szerint vagy HEXA formában is kérhetjük, és kereshetünk is benne.

4 — EDIT Állomány javítása, szer-

kesztése. Az NC-ben megtalálható saját, vagy külön definiált szövegszerkesztő használatával.

5 — COPY Kijelölt állományok másolása. A kijelölés az INS gombbal, vagy a 'hosszú-' vagy 'szürkeplusz' billentyűvel történhet. Ez utóbbi esetben az összes (\*.\*) vagy valami alapján (például \*.TXT) meghatározott bejegyzéseket jelöli ki. A téves kijelölés az INS ismételt megnyomásával illetve a 'szürkemínusz' billentyűvel szüntethető meg. Természetesen az egy bejegyzésre vonatkozó műveletek mindig arra a bejegyzésre vonatkoznak, amelyik inverzben látható.

6 — RENMOV Átnevezés, áthelyezés. A kijelölt állományt átnevezhetjük, vagy átmásolhatjuk máshová úgy, hogy az eredeti helyről törli.

7 — MKDIR Alkatalógus létrehozása az aktuális katalógusban.

8 — DELETE Az aktuális bejegyzés

vagy kijelölt bejegyzések törlése.

9 — PULLDN Atlépés a legördülő menübe.

10 — QUIT Kilépés az NC-ből.

A legördülő menünek öt pontja van, LEFT, FILES, COMMANDS, OPTIONS és RIGHT. A LEFT és RIGHT menüpontban a jobb és bal oldali ablakok jellemzőit állíthatjuk be, a FILES menüpontban az F1—F8, F10 funkciókat, a kijelölést és megszüntetést, az attribútumállítást, illetve ha van modem, a file-ok forgalmazását találjuk. A COMMANDS menüpontban néhány saját belső parancsot használhatunk, mint például filekeresés, míg az OPTIONS-ban az alapbeállításokat állíthatjuk.

Az említett segédprogramok hivatalosan is beszerezhetőek Magyarországon, néhány ezer (néhány tízezer) forintos áron...

Lengyel István

## Abszolút töltés és mentés

Kezdjük először egy adott tárolóterület kimentésével. A legjobb példa ehhez a képernyő, mert ott mindig látni lehet a történéseket. A képernyő (alapállapotban) az 1024-es címen kezdődik és a 2024-ig tart (40 karakterx25 sor)

A betöltéshez vagy a mentéshez előbb át kell adni a paramétereket. Ez az alábbi módon történik: **SYS 57812 "név",8**

Meglehető lehet, hogy a SYS után nincs vessző, de ha belegondolunk, a LOAD/SAVE utasítás után sem kell ilyen írásjelet kitenni. Az egységyszám előtt pedig ott is, itt is kell vessző.

Ezután azt a területet kell rögzíteni, amelyet tárolni akarunk (a töltés később jön). Ehhez egy kicsit számolni kell:

POKE 193, KC—256\*INT(KC/256):POKE 194,KC/256

POKE 174,VC—256\*INT(VC/256):POKE 194,VC/256

-Mivel egy cím (a maximális érték a 65535) nem fér be egy tárolócellába (max. érték 256), két részre (byte) kell bontani. A két érték lesz a magasabb értékű és az alacsonyabb értékű byte. A magasabb értékű byte a cím/256 eredményének egész részét kapja. Az alacsonyabb érték kiszámítása kicsit komplikáltabb. Szükség van a magasabb egész értékre, amit megszorozunk 256-tal, majd ezt levonjuk az eredeti számból. Az így kapott byte-ok mindegyike kisebb lesz mint 256. Ha belegondolunk, nem tettünk mást, mint a címet „kétjegyű”, 256-os alapú szám-má alakítottuk át: cím=HI\*256+LO

A főnti BASIC sorokkal a szükséges cellákba írjuk az értékeket. A mi esetünkben a kezdőcím KC=1024, a végcím VC=2024 lesz. Most már csak a tárolás megkezdésére kell utasítást adnunk:

SYS 62957

A kívánt terület (itt a képernyő) lemezre kerül.

Az így kimentett információ eredeti helyre való töltése a jól ismert LOAD "név",8,1-gyel lehetséges. De manipulációkat is vé-

gezhetünk. Töltsük be például a képernyőt egy sorral „följebb”. Itt is előbb a nevet kell átadni: **SYS 57812 "név",8**

A töltési cím TC=1024—40=984 lesz. A főnti képlettel számított high és low értékeket most a 781/782-es cellákba kell írni:

POKE 781, TC—256\*INT(TC/256):POKE 782,TC/256

Ha töltésről van szó, a 780-as címre a 0 értéket kell vinni: **POKE 780,0** (VERIFY-nál ott egy 1-esnek kell állni). Ha minden paramétert megadtunk, **SYS 57717**-re a képernyő tartalma pontosan egy sorral följebb töltődik vissza. Természetesen ezzel a módszerrel más tárolóterületek is kimenthetők, betölthetők (pl. sprite-ok).

## BASIC programozók tippjei

Használjuk ki a beépített operációs rendszer rutinokat! Így BASIC programjaink hatékonyabbak lesznek. Amit BASIC-ben hosszú sorokkal kell megoldani, néha egyetlen SYS utasítás elvégzi. Például:

SYS 58778  
SYS 59626  
SYS 64738  
SYS 64767  
SYS 58592

Inicializáljuk a videochipet.  
Egy sorral följebb görgetjük a képernyőt.  
BASIC hidegindítás.  
BASIC hidegindítás, a színek maradnak.  
Billentyűre várunk (kazetta FOUND rutin)  
A C=, CTRL, —, 1, 2, szóköz, Q billentyűk megnyomásakor továbblépünk. STOP=break.

POKE 214,X:POKE211,Y  
SYS 58732  
POKE 781,X:SYS59903  
POKE 649,0  
POKE 808,230

Kurzor az X. sorban az Y. oszlopba.  
Az X. képernyői sor törlése.  
Billentyűzet lekapcsolás (pufferméret=0).  
A RUN/STOP gomb kiiktatása.



## Gépi kódú programozás Commodore gépeken

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

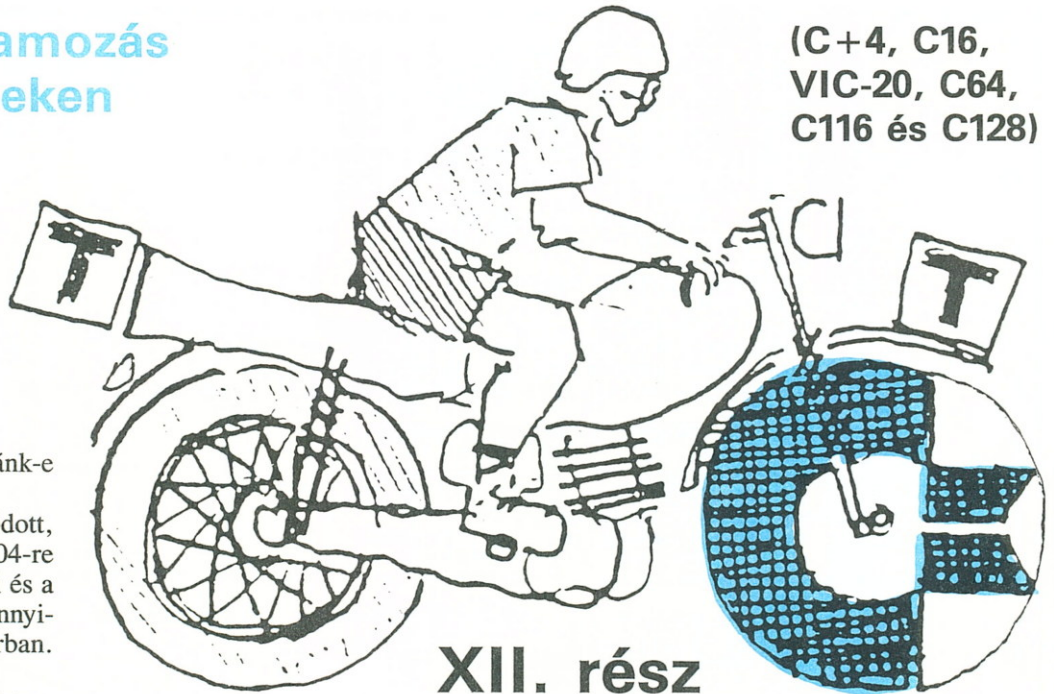
### Csináljunk órát

Az ötlet elcsépett, de meg tudnánk-csinálni? Próbáljuk meg!

A mellékelt program C+4-re íródott, a betöltőben a 0C-ket kell csak 04-re (C64) vagy 1E-re (VC-20) cserélni és a sorvégén álló ellenőrzőösszegeből annyi-szor 8-at levonni, ahány 0C állt a sorban. C128-on csak 64-es módban fut.

Jó lenne, ha programunk úgy futna, hogy közben lehessen BASIC programot írni, esetleg futtatni is. Ehhez legalkalmasabb eszköz a megszakítás. C-gépeken a videojelet előállító IC másodpercenként kb. 100-szor küld megszakítást a  $\mu$ P-nek. Ekkor, a \$FFFE-FFFF címek tartalmát ugrócímként értelmezve, elugrik a  $\mu$ P a megszakítást kezelő szubrutinra (persze előbb a stack-be menti el a visszatérési címet és a státusz regisztert). Ez a rutin ROM-ban van. Mégis lehetőség van az eltérítésére, mert tartalmaz egy JMP (\$0314) utasítást. Itt egy a ROM-ba visszamutató cím van. Írjuk ezt át a saját rutinunkra és annak lefutása után ugorjunk a régi címre. Tehát lesz egy rutin ami átírja a \$0314 vektort, majd visszatér BASIC-be. Innentől a rendszer futtatgatja néha a rutinunkat.

Ez így megy is. Mit tegyen a programunk. Számláljon kb. 110-et (egy másodpercig) utána növelje meg a másodpercet egygel, ha az 10, akkor a tízmásodpercet



### XII. rész

növelje, ha az 6, akkor a perc egyest növelje stb... ahogyan egy digitális óra is működik. Célszerűen a képernyőmemóriában lévő számokat közvetlenül növelje, így nem kell külön számolni, meg konvertálni... Ennek hátránya, hogyha beleírunk az órába karaktereket, az megvadul és összevevicsa fog számolni. Ezt talán el lehet nézni ennek az egyszerű jószágnak.

Problémát okozhat, hogy és mikor írhatjuk át a megszakításvektort, a \$0314-0315-ös cím tartalmát. Világos, hogy kellemetlen lenne, ha éppen akkor érkezne megszakítás, amikor még csak az egyik byte-ot írtuk át (jól eltévedne a processzor). Tiltsuk le a megszakításokat (SEI) addig, amíg a vektort változtatjuk, utána minden mehet tovább (CLI). Nem tanácsos BASIC programból POKE-kal megszakítás vektort átírni (pedig itthon megjelenő újságban is láttam TÖBB-

SZÖR ilyen szarvashibát), ugyanis BASIC programok futása alatt szükségszerűen engedélyezett a megszakítás (belső óra és egyebek miatt), továbbá nem olyan gyors a BASIC, hogy állíthatnánk: 'Úgysem jön a két POKE között megszakítás!', mert jön, de legalábbis jöhet, akkor pedig mindig rosszkor.

Megjegyeznénk, hogy a  $\mu$ P bekapcsoláskor és RESET-kor a \$FFFC-FFFO címről vesz az indítócímet. Ha pedig NMI érkezik (kivéve C+4, mert ott ez nem létezik), ami egy megszakítás fajta, amit nem lehet semmiféle módon letiltani, nem érdekli az I flag állapota, akkor a \$FFFA-FFFF címekekről veszi a rutin címet. Ez a három fő vektor amit az  $\mu$ P maga tudja (hardver szinten), ebből az ügyes rendszer teszi lehetővé a további vektorozást pl. \$0314 és több más hasonló.

Nagy Tamás

○	10000	DATA	4C,06,50,00,00,71,78,AD,0248	○
○	10001	DATA	14,03,5D,03,60,AD,15,03,01CC	
○	10002	DATA	8D,04,60,A9,1F,8D,14,03,025D	○
○	10003	DATA	A9,60,5D,15,03,50,60,A9,030F	
○	10004	DATA	3A,8D,02,0C,8D,05,0C,CE,0241	
○	10005	DATA	05,60,00,6A,A9,71,8D,05,0340	
○	10006	DATA	60,EE,07,0C,AD,07,0C,C9,02EA	○
○	10007	DATA	39,90,5B,A9,30,8D,07,0C,029D	
○	10008	DATA	EE,06,0C,AD,06,0C,C9,36,02BE	
○	10009	DATA	90,4C,A9,30,8D,06,0C,EE,0342	○
○	10010	DATA	04,0C,AD,04,0C,C9,39,90,025F	
○	10011	DATA	3D,A9,30,8D,04,0C,EE,03,02A4	
○	10012	DATA	0C,AD,03,0C,C9,36,90,2E,0285	○
○	10013	DATA	A9,30,8D,03,0C,EE,01,0C,0270	
○	10014	DATA	AD,01,0C,C9,34,90,1F,C9,032F	
○	10015	DATA	39,80,07,AD,00,0C,C9,32,02A4	○
○	10016	DATA	90,14,A9,30,8D,01,0C,EE,0305	



```

0 10017 DATA 00,0C,AD,00,0C,C9,33,90,0251
0 10018 DATA 05,A9,30,8D,00,0C,6C,03,01E5
0 10019 DATA 60,00,00,00,00,00,00,00,0000
0 20000 AS$="0123456789ABCDEF"
0 20001 DIMH(ASC("F"))
0 20002 FORI=0TO15:H(ASC(MID$(AS$,I+1)))=I
0 20003 NEXT
0 20004 BA=6*4096
0 20005 PRINT"KONTROLL INDUL !"
0 20006 FORI=0TO19:C=0
0 20007 FORB=0TO7
0 20008 READA$:GOSUB20029
0 20009 C=C+A
0 20010 NEXTB:PRINT". ";
0 20011 READB$:A$=LEFT$(B$,2):GOSUB20029
0 20012 B=A:A$=RIGHT$(B$,2):GOSUB20029
0 20013 IFB*256+A<>C THENPRINT"ADATHIBA A"10000+I"--EDIK SORBAN !!!":STOP
0 20014 NEXTI
0 20015 PRINT"KONTROLL KESZ !"
0 20017 RESTORE
0 20018 FORI=0TO19
0 20019 FORB=0TO7
0 20020 READA$:GOSUB20029
0 20021 POKEBA+I*8+B,A
0 20022 NEXTB:PRINT". ";
0 20023 READA$
0 20024 NEXTI
0 20025 PRINT"OK."
0 20026 PRINT"INDITASA : "
0 20027 PRINT"SYS"BA"
0 20028 END
0 20029 H=ASC(A$):L=ASC(RIGHT$(A$,1))
0 20030 A=H*16+L:RETURN

```

```

0 00001 0000 ;put"@demo3.src
0 00002 0000 '
0 00003 0000 * = $6000
0 00004 6000 4c 06 60 jmp start
0 00005 6003 vektor = $0314 ;vektor helye
0 00006 6003 base = $0c00
0 00007 6003 00 00 rvekt ,wor 0 ;ide tároljuk
0 00008 6003 71 cnt ,byt 113 ;időzítés kb. 1s
0 00009 6006 ;
0 00010 6006 78 start sei ;most KUSS
0 00011 6007 ad 14 03 lda vektor ;elmentjük
0 00012 600a 8d 03 60 sta rvekt ;az előzőt
0 00013 600d ad 15 03 lda vektor+1 ;hogy ott tudjuk
0 00014 6010 8d 04 60 sta rvekt+1 ;folytatni
0 00015 6013 a9 1f lda #ujira ;beírjuk
0 00016 6015 8d 14 03 sta vektor ;a saját rutin
0 00017 6018 a9 60 lda #ujira ;címét
0 00018 601a 8d 15 03 sta vektor+1
0 00019 601d 58 cli ;így már mehet
0 00020 601e 60 rts ;írja a megszák.
0 00021 601f ;
0 00022 601f a9 3a ujira lda #' ;jó, ha van
0 00023 6021 8d 02 0c sta base+2
0 00024 6024 8d 05 0c sta base+5
0 00025 6027 ce 05 60 dec cnt
0 00026 602a d0 6a bne iraveg ;eltelt egy mp.?
0 00027 602c a9 71 lda #113
0 00028 602e 8d 05 60 sta cnt
0 00029 6031 ee 07 0c inc base+7
0 00030 6034 ad 07 0c lda base+7
0 00031 6037 c9 39 cmp #'9'+1
0 00032 6039 90 5b bcc iraveg ;masodp. egyes
0 00033 603b a9 30 lda #'0'
0 00034 603d 8d 07 0c sta base+7
0 00035 6040 ee 06 0c inc base+6
0 00036 6043 ad 06 0c lda base+6
0 00037 6046 c9 36 cmp #'6'
0 00038 6048 90 4c bcc iraveg ;masodp. tizes
0 00039 604a a9 30 lda #'0'
0 00040 604c 8d 06 0c sta base+6
0 00041 604f ee 04 0c inc base+4
0 00042 6052 ad 04 0c lda base+4
0 00043 6055 c9 39 cmp #'9'+1
0 00044 6057 90 3d bcc iraveg ;perc egyes
0 00045 6059 a9 30 lda #'0'

```



```

00046 605b 8d 04 0c      sta. base+4
00047 605e ee 03 0c      inc. base+3
00048 6061 ad 03 0c      lda. base+3
00049 6064 c9 36      cmp #'6'
00050 6066 90 2e      bcc irqve9      ;perc tizes
00051 6068 a9 30      lda. #'0'
00052 606a 8d 03 0c      sta. base+3
00053 606d ee 01 0c      inc. base+1
00054 6070 ad 01 0c      lda. base+1
00055 6073 c9 34      cmp #'4'
00056 6075 90 1f      bcc irqve9
00057 6077 c9 39      cmp #'9'+1      ;ora. egyes
00058 6079 b0 07      bcs kell
00059 607b ad 00 0c      lda. base+0
00060 607e c9 32      cmp #'2'
00061 6080 90 14      bcc irqve9
00062 6082 a9 30      lda. #'0'      kell
00063 6084 8d 01 0c      sta. base+1
00064 6087 ee 00 0c      inc. base+0
00065 608a ad 00 0c      lda. base+0
00066 608d c9 33      cmp #'3'
00067 608f 90 05      bcc irqve9      ;ora. tizes
00068 6091 a9 30      lda. #'0'
00069 6093 8d 00 0c      sta. base+0
00070 6096 6c 03 60      irqve9 jmp (vekt)      ;az elozore

end of assembly', error count = 00000

base      0c00      cnt      6085      irqve9  6096      kell      6082
vekt      6003      start   6086      ujira   601f      vektor   0314

```

# AMIGA

## DOS programok

Sokan azon a véleményen vannak, hogy a CLI rendszere egyáltalán nem alkalmas programok írására, akik így gondolkodnak, azoknak annyiból igazuk van, hogy az AmigaDOS-ban valóban nem lehet magas vagy viszonylag magas szintű programokat készíteni, de egyes esetekben, segítségével egészen egyszerűen oldhatunk meg, a kezdő felhasználók számára bonyolultnak tűnő problémákat.

Az első program egy menü elkészítésére mutat példát, amely a számítógépekhez nem értőknek nagy hasznára lehet.

Segítségével egyszerűen, egyes gombok lenyomásával kaphatunk információkat a rendszerről, helyettesítve a parancsok begépelését (pl.: info, list, dir, CD, avail).

A második program azoknak a már gyakorlott programozóknak nyújt hasznos segítséget, akik sokat alkalmazzák az

```

CLI menu
=====

      .bra {
      .ket }

      echo ""
      echo ""
      echo "CLI Menu:"
      echo "l = LIST. d - DIR. i - INFO"
      echo "a - AVAIL. c - CD. q - QUIT"
      echo ""

      skip <* >NIL: ?

      ; INFO

      lab i
      info
      skip done

```



AmigaDOS 1.3 által nyújtott RAD: funkciót.

Bizonyára találkoztak már azzal a problémával, miszerint a RAD: beindítását nem végezhetik el büntetlenül a Startup file-ból, mert így reset hatására újrafomatálódik a memória. Ezt a problémát hidalja át ez a rövid program.

A programokat mindenki átírhatja, módosíthatja a saját elvárásainak megfelelően, kialakítva a számára legkedvezőbb feltételeket.

Bakos Gábor

```

○      ; DIR                                ○
      lab d
      dir
      skip done
○
○      ; AVAIL                              ○
      lab a
      avail
      skip done
○
○      ; LIST                               ○
      lab l
      list
      skip done
○
○      ; QUIT                               ○
      lab q
      quit
○
○      ; CD                                 ○
      lab c
      echo "Uj directory nev ?"
      cd <* >NIL: ?
      cd
      skip done
○
○      lab done
      wait 1
      execute menu
    
```

Startup RAD: formatalo

=====

```

failat 15
mount rad:
    
```

```

if not exists rad:c
  format drive rad: name RamDrive noicons
  makedir rad:c
endif
    
```



## KERET ANIMÁTOR

Kirakatokban látható futófényhez hasonló hatást érhetünk el ezzel a kis programmal. A 60–70 sorok szövegét cseréljük le a megjeleníteni kívántra. A színeket 80-as sor három POKE utasításával állíthatjuk be. Próbáljuk ki a különböző szín-értékeket 0 és 15 között.



```

0 REM *** SZIN-ROLLER ***
10 FOR T=49152 TO 49374:READ D:POKE T,D:CK=CK+D:NEXT
20 IF CK<>27434 THEN PRINT "HIBA AZ ADATSORBAN !":END
30 SYS 49152,1,2,3,6
40 PRINT CHR$(147)"BEKAPCSOLAS: SYS 49152,C1,C2,C3,S1"
50 DATA 32,155,183,142,60,3,32,155,183,142,61,3,32,155,183,142,62,3,32
60 DATA 155,183,134,251,120,162,96,142,20,3,162,192,142,21,3,88,96,32,181
70 DATA 192,174,66,3,224,1,208,3,76,49,234,173,60,3,160,0,153,0,216,153
80 DATA 120,216,153,240,216,153,104,217,153,224,217,153,88,218,153,208
90 DATA 218,153,72,219,153,192,219,200,192,40,208,224,173,61,3,160,0,153
100 DATA 40,216,153,160,216,153,24,217,153,144,217,153,8,218,153,128,218
110 DATA 153,248,218,153,112,219,200,192,40,208,227,173,62,3,160,0,153
120 DATA 80,216,153,208,216,153,64,217,153,184,217,153,48,218,153,168
130 DATA 218,153,32,219,153,152,219,200,192,40,208,227,173,62,3,141,63,3
140 DATA 173,61,3,141,62,3,173,60,3,141,61,3,173,63,3,141,60,3,76,49,234
150 DATA 238,65,3,173,65,3,197,251,208,9,169,0,141,65,3,141,66,3,96,169
160 DATA 1,141,66,3,96,120,162,219,142,20,3,162,192,142,21,3,88,96,76,49
170 DATA 234,0
READY.

```

## AMÖBA

Bizonyára vannak olyan számítógép tulajdonosok, akik nemcsak azért vettek maguknak egy gépet, hogy játszanak rajta, hanem, hogy saját programokat is írjanak. Ezzel a programmal azoknak szeretnék keveskedni, akik a két réteg közé tartoznak, mivel ez a program mind programozási, mind pedig játékos szelleműeknek való. A listában szereplő szokatlan karakterek:

$\uparrow$  = SHIFT + B  
 $\leftarrow$  = SHIFT + C  
 $|$  = C = + W  
 $\frac{1}{2}$  = C = + Q

Mahunka Zsolt

Tippek  
Trükkök

```

100 PRINTCHR$(147)CHR$(5)"";
110 A$="1 $\uparrow$ 2 $\leftarrow$ 3":PRINTSPC(20-LEN(A$)/2);A$
120 B$="1 $\uparrow$ 1 $\leftarrow$ 1":PRINTSPC(20-LEN(B$)/2);B$
130 C$="4 $\uparrow$ 5 $\leftarrow$ 6":PRINTSPC(20-LEN(C$)/2);C$
140 D$="1 $\uparrow$ 1 $\leftarrow$ 1":PRINTSPC(20-LEN(D$)/2);D$
150 E$="7 $\uparrow$ 8 $\leftarrow$ 9":PRINTSPC(20-LEN(E$)/2);E$
153 SA=0
155 POKE211,1:POKE214,10:SYS58640
160 INPUT"X/O:";XO$
170 IFXO$="X"THENQ=24
180 IFXO$="O"THENQ=15
185 POKE211,1:POKE214,12:SYS58640
190 INPUT"1,2,3,4,5,6,7,8,9:";S
200 IFS=1THENPOKE1081,Q
210 IFS=2THENPOKE1083,Q
220 IFS=3THENPOKE1085,Q
230 IFS=4THENPOKE1161,Q
240 IFS=5THENPOKE1163,Q
250 IFS=6THENPOKE1165,Q
260 IFS=7THENPOKE1241,Q
270 IFS=8THENPOKE1243,Q
280 IFS=9THENPOKE1245,Q
290 A=PEEK(1081):B=PEEK(1083):C=PEEK(1085)
292 D=PEEK(1161):E=PEEK(1163):F=PEEK(1165)
294 G=PEEK(1241):H=PEEK(1243):I=PEEK(1245)
300 REM-VIZSZINTES-
310 IFA=24ANDB=24ANDC=24THEN1000
320 IFA=15ANDB=15ANDC=15THEN1500

```





```

○          330 IFD=24ANDE=24ANDF=24THEN1000
○          340 IFD=15ANDE=15ANDF=15THEN1500
○          350 IFG=24ANDH=24ANDI=24THEN1000
○          360 IFG=15ANDH=15ANDI=15THEN1500
○          365 REM-FUGGOLEGES-
○          370 IFA=24ANDD=24ANDG=24THEN1000
○          380 IFA=15ANDD=15ANDG=15THEN1500
○          390 IFB=24ANDE=24ANDH=24THEN1000
○          400 IFB=15ANDE=15ANDH=15THEN1500
○          410 IFC=24ANDF=24ANDI=24THEN1000
○          420 IFC=15ANDF=15ANDI=15THEN1500
○          425 REM-ATLOS-
○          430 IFA=24ANDE=24ANDI=24THEN1000
○          440 IFA=15ANDE=15ANDI=15THEN1500
○          450 IFC=24ANDE=24ANDG=24THEN1000
○          460 IFC=15ANDE=15ANDG=15THEN1500
○          461 SA=SA+1
○          462 IFSA=9THENGOSUB2000
○          470 GOTO155
○          1000 PRINT"AZ -X- NYERTE EZT A JATSZMAT !!!":END
○          1500 PRINT"AZ -O- NYERTE EZT A JATSZMAT !!!":END
○          2000 PRINT"SAJNOS EZ DONTETLEN !!!":END

```

READY.

## SZÍN ROLLER

A szín roller abban különbözik a szokásos színúszató gépi kódú programoktól, hogy itt három szín gördül végig a képernyő tetszőleges szövegén. A program beírása és futtatása után a kívánt hatást a SYS 49152,C1,C2,C3,SP paranccsal érhetjük el. Itt a C1 változó az első, C2 a második, C3 pedig a harmadik szín kódja (0–15 közötti érték), SP pedig a görgetés sebessége (1 a leggyorsabb, 255 a lelassabb). A változók különböző értékeivel a színhatások széles variációja állítható elő. A görgetést SYS 49358-cal kapcsolhatjuk ki.

Tippek  
Trükkök

```

○          0 REM *** SZIN-ROLLER ***
○          10 FOR T=49152 TO 49374:READ D:POKE T,D:CK=CK+D:NEXT
○          20 IF CK<27434 THEN PRINT "HIBA AZ ADATSORBAN !":END
○          30 SYS 49152,1,2,3,6
○          40 PRINT CHR$(147)"BEKAPCSOLAS: SYS 49152,C1,C2,C3,SP"
○          50 DATA 32,155,183,142,60,3,32,155,183,142,61,3,32,155,183,142,62,3,32
○          60 DATA 155,183,134,251,120,162,36,142,20,3,162,192,142,21,3,88,96,32,181
○          70 DATA 192,174,66,3,224,1,208,3,76,49,234,173,60,3,160,0,153,0,216,153
○          80 DATA 120,216,153,240,216,153,104,217,153,224,217,153,88,218,153,208
○          90 DATA 218,153,72,219,153,192,219,200,192,40,208,224,173,61,3,160,0,153
○          100 DATA 40,216,153,160,216,153,24,217,153,144,217,153,8,218,153,128,218
○          110 DATA 153,248,218,153,112,219,200,192,40,208,227,173,62,3,160,0,153
○          120 DATA 80,216,153,208,216,153,64,217,153,184,217,153,48,218,153,168
○          130 DATA 218,153,32,219,153,152,219,200,192,40,208,227,173,62,3,141,63,3
○          140 DATA 173,61,3,141,62,3,173,60,3,141,61,3,173,63,3,141,60,3,76,49,234
○          150 DATA 238,65,3,173,65,3,197,251,208,9,169,0,141,65,3,141,66,3,96,169
○          160 DATA 1,141,66,3,96,120,162,219,142,20,3,162,192,142,21,3,88,96,76,49
○          170 DATA 234,0

```

READY.

## HINTA

Hintáztassuk 64-esünk egész képernyőjét. Önálló programként, vagy saját programunkba épített szubrutinként is alkalmazható.

```

0 REM *****
1 REM *   HINTA-C64   *
2 REM *****
3 REM
4 REM
10 POKE 53280,0:POKE53281,0:PRINTCHR$(147)
20 PRINTTAB(5)" "
30 PRINTTAB(6)" |   HINTA 64/128   |":PRINTTAB(5)" "
40 PRINT" "
50 PRINTTAB(11)" "
60 FOR LR=0 TO 7:POKE53270,(PEEK(53270)AND248)+LR:NEXT LR
70 FOR RL=7 TO 0 STEP -1:POKE53270,(PEEK(53270)AND248)+RL:NEXT RL:GOTO 60

READY.
    
```

## Mik azok a digi-hangzások?

A C64-nél kétféle „zenélés” között teszünk különbséget. A legegyszerűbb és leggyakrabban alkalmazott módszer, amikor „kész” zenéket digitalizálnak. Erre a célra szolgál a négybit-es zenedigitalizáló, pl. a Rosenplänter cég D.A.I.S.Y modulja. Ezek a készülékek „szétbontják” a zenéket digitális számértékekké, és az adatokat a tárolóba írják. A lejátszásnál ezeket az adatokat el kell juttatni a SID chiphez, annak is a hangerő regiszterébe. Minden egyes változtatásnál rövid „kattanást” hallhatunk. Ha ez a kattogás elég gyors, a fülünk nem impulzusokat, hanem hangot hall. Ezzel a módszerrel tudja a C64 a legjobb zenéket generálni. Bizony, nem sokkal maradunk el a HiFi minőségtől!

Sajnos a dologban az a gond, hogy ilyen zenéket (pl. egy gépi kódú monitorral) nem lehet programozni, hacsak nem folyik valakiben Sziszifusz vére... Így a digitalizálóról nem mondhatunk le. Ha ilyen nincs, és nincs az ismerőseinknek sem, általában a „jé milyen jó a hangzás ebben a demóban” felkiáltással szokás „kivenni” onnan a mások munkáját. Ez nem baj egészen addig, amíg nem próbáljuk meg „eladni” saját munkaként egy demóban. Vagyis a becsület azt kívánja, hogy ilyesmit csak magunknak vagy csak tesztelezésre csináljunk.

A második, azóta kevésbé praktizált módszer, hogy a SID erre a célra szolgáló regisztereit a megfelelő értékekkel töltjük. Amíg a digitalizáló ára magas volt, ez volt a legtöbb ember számára az egyetlen elérhető módszer.

## Egy új digi-zenerutin

Nemrég közreadtunk egy, digi-hangzások lejátszására használható rutint. Az azonban új szerzőnk számára túl rugalmatlan és hosszú volt. Ebből az okból egy új 4-bit-nmi rutint írt, amely plusz szolgáltatásként sebességállítási lehetőséggel is rendelke-



**GÉPKER**  
Kft.

Gépkereskedelmi és Ügyviteltechnikai Kft.

A Gépker 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

zik. Olyan nulláslap címeket, amelyeket a demók és intrók eleve használnak, az új rutin nem érint. Maga a program a \$C000–\$C07B területen áll, az indítás a SYS 49152-vel történhet. Ha meg akarjuk állítani a lejátszást, elég a SYS 49155 begépelése. A digik kezdőcíme (high byte) a \$C006-os, a végcím a \$C007-es pozícióban található. Ha a sebességet másképp akarjuk beállítani, a \$C008-as cellát kell fölülírni.

A 255 a leglassúbb sebesség. Ekkor 3800 nibble (1900 byte) játszódik le másodpercenként.

Hetven az (elméleti) leggyorsabb tempó. Ez 14300 nibble-nek (7150 byte) felel meg másodpercenként.

Ennél kisebb értékek nem hatnak a sebességre, mivel a rutinak magának is szüksége van időre a digik lejátszásához.

Mielőtt még behívnánk az előkészített digi adatokat és elindítanánk a lejátszórutint, be kell állítani a kezdő- és végcímeket. A rutin a lejátszás után ismét előre ugrik, egészen addig amíg a SYS 49155-tel csendet nem teremtünk.

# Tippek Trükkök

```

1 REM *****
2 REM *
3 REM * DIGIRUTIN - C64 *
4 REM *
5 REM *****
6 :
7 :
8 PRINT CHR$(147)"DATA BEOLVASAS ES ELLENORZES ...":J=49152:VE=49275:P=J
9 FOR B=0 TO 7:READ A#
10 L=ASC(MID$(A#,2,1))
11 H=ASC(MID$(A#,1,1))
12 L=L-48:IF L>9 THEN L=L-7
13 H=H-48:IF H>9 THEN H=H-7
14 PRINT"#####P";P=P+1
15 IF H>15 OR L>15 THEN 17
16 A=H*16+L:POKE J+B,A:T=T+A:NEXT B:READ A:IF A=T THEN 18
17 PRINT:PRINT"DATA HIBA ... SOR:"PEEK(64)*256+PEEK(63):END
18 T=0:J=J+8:IF J<VE THEN 9
19 PRINT"#####KESZ":END
20 DATA 4C,09,C0,4C,3A,C0,20,50,0715
21 DATA 80,20,3A,C0,A9,43,8D,18,0811
22 DATA 03,A9,C0,8D,19,03,8C,45,0742
23 DATA C0,AD,06,C0,8D,46,C0,8D,1107
24 DATA 5F,C0,AD,07,C0,8D,5B,C0,1083
25 DATA AD,08,C0,8D,04,DD,8C,05,0884
26 DATA DD,A9,81,8D,0D,DD,8D,0E,1049
27 DATA DD,60,A0,00,8C,0E,DD,AD,1025
28 DATA 8D,DD,60,48,AD,00,20,8D,0748
29 DATA 6F,C0,29,0F,8D,18,D4,EE,0974
30 DATA 45,C0,D0,0F,EE,46,C0,AD,1157
31 DATA 46,C0,C9,50,90,05,A9,20,0893
32 DATA 8D,46,C0,A9,6D,8D,18,03,0849
33 DATA AD,0D,DD,68,40,48,A9,00,0816
34 DATA 4A,4A,4A,4A,8D,18,D4,A9,0842
35 DATA 43,D0,EA,A9,00,00,00,00,0678

```

READY.



## HAT SOR FAGYASZTÁSA

Ez a program a C64 képernyőjének felső hat sorát fagyasztja. Ide írhatunk olyan szövegeket és adatokat, melyekre szükségünk van, nem akarjuk, hogy véletlenül törlődjenek. A fagyasztani kívánt sorok számát (legfeljebb hat) szorozzuk meg 40-nel, és ezt az értéket POKE-oljuk a 49182-re. Például hat sor fagyasztásához futtassuk le a programot, adjuk be a POKE 49182,240-et, majd a SYS 49152-t. A fagyasztott sorok „kiolvastása” SYS 49220-szal történhet.

Tippek  
Trükkök

```

0 REM *****
1 REM * HAT SOR FAGYASZTASA - C64-EN *
2 REM *****
3 REM
4 REM
10 FOR T=49152 TO 49235:READ D:CK=CK+D:POKE T,D:NEXT
20 IF CK <> 9252 THEN PRINT "HIBA A DATASOROKBAN ...":END
30 DATA 76,31,192,120,173,20,3,141,28,192,173,21,3,141,29,192,169,48,141
40 DATA 20,3,169,192,141,21,3,88,96,234,234,40,160,0,204,30,192,240,221
50 DATA 185,0,4,153,83,192,200,76,33,192,160,0,204,30,192,240,10,185,83
60 DATA 192,153,0,4,200,76,50,192,108,28,192,120,173,28,192,141,20,3,173
70 DATA 29,192,141,21,3,88,96,234
    
```

READY.

## MEMÓRIA DISPLAY

Programozók figyelem! Unjátok már, hogy egy memória rekesz tartalmának ellenőrzéséhez le kell állítani a programot és PEEK(X)-szel megnézni az értéket? A memória display tetszőleges rekesz tartalmát binárisan kijelzi a képernyő bal felső sarkában.

A program futtatása után (a program egyébként a memóriában bárhol elhelyezhető) megadhatjuk a kijelezni kívánt rekesz címét (például a botkormány bemenetét) és attól kezdve folyamatosan figyelhetjük a rekesz tartalmát. Az érték fehéren jelenik meg, a szint POKE 828,X paranccsal módosíthatjuk, ahol X 0 és 15 közötti szám.

Tippek  
Trükkök

```

0 REM *** MEMORIA DISPLAY ***
10 FOR T=828 TO 897:READ A:POKE T,A:NEXT
20 INPUT "BELEPESI PONT : ";LC:HI=LC/256
30 LO=LC-256*HI:POKE 861,LO:POKE 862,HI
40 SYS 828
50 DATA 120,174,20,3,172,21,3,142,130,3,140,131,3,162,85,160,3,142,20,3
60 DATA 140,21,3,88,96,169,128,141,132,3,160,0,173,133,3,45,132,3,205
70 DATA 132,3,240,5,169,48,76,110,3,169,49,153,0,4,169,1,153,0,216,200
80 DATA 192,8,240,6,110,132,3,76,92,3,76
    
```

READY.

# SZÍNES KARAKTERVILLOGÁS

A program a C64 tizenhat színét pörgeti végig a képernyő összes byte-ján. Bekapcsolása SYS 49152-vel, kikapcsolása SYS 49217-tel.

A színek elég gyorsan futnak, a képernyő könnyen olvashatatlanná válik, ezért használjunk lehetőleg nagyméretű betűket és grafikát. A „villogó keret” trükkel összekapcsolva érdekes hatást érhetünk el.

Tippek  
Trükkök

```

0 REM *****
1 REM * SZÍNES KARAKTERVILLOGÁS *
2 REM *****
3 REM
4 REM
10 FOR T=49152 TO 49231:READ D:POKE T,D:NEXT T:
20 DATA 169,255,141,14,212,141,15,212,169,128,141,18,212,120,173,20,3
30 DATA 141,38,192,173,21,3,141,39,192,169,40,141,20,3,169,192,141,21,3
40 DATA 88,96,234,234,162,0,173,27,212,157,0,216,157,0,217,157,0,218
50 DATA 157,0,219,232,224,0,208,236,108,38,192,120,173,38,192,141,20,3
60 DATA 173,39,192,141,21,3,88,96
    
```

READY.

# CIRCLE

Ez a program tulajdonképpen egy CLI kiegészítés, mellyel ellipsziseket rajzolhatunk a képernyő tetszőleges pontjára, tetszőleges színben, látványosabbá téve programjainkat.

Tippek  
Trükkök

```

/*
    CIRCLE x y r1 r2 col

    AztecC:  cc +l <file>
            ln <file>.o -lc32 -lm -o

    Code by TPE
*/

#include <functions.h>
#include <exec/types.h>
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
struct RastPort *RP;

main(argc,argv)
int argc;
char *argv[];
{
    int x,y,r1,r2,col;
    
```

AMIGA

```

    if(argc != 6)
    {
        printf("USAGE: CIRCLE x y r1 r2 col\n");
        exit();
    }

    if(!(IntuitionBase=(struct IntuitionBase *)
        OpenLibrary("intuition.library",0))) exit(FALSE);

    if(!(GfxBase=(struct GfxBase *)
        OpenLibrary("graphics.library",0)))
    {
        CloseLibrary(IntuitionBase);
        exit(FALSE);
    }

    RP = &(IntuitionBase->ActiveScreen->RastPort);

    x = atoi(argv[1]);
    y = atoi(argv[2]);
    r1 = atoi(argv[3]);
    r2 = atoi(argv[4]);
    col= atoi(argv[5]);

    SetAPen(RP,col);

    DrawEllipse(RP,x,y,r1,r2);

    CloseLibrary(IntuitionBase);
    CloseLibrary(GfxBase);
    exit(TRUE);
}
    
```

## VILLOGÓ KERET

A program többszínű mintát varázsol a C64 képernyő keretére. A következő, „hinta” trükkel együtt alkalmazva különösen hatásos.

Tippek  
Trükkök

```

0 REM *****
1 REM * VILLOGO KERET *
2 REM *****
3 REM
4 REM
10 POKE 53298,0:POKE53281,0:C=81:X=1:GOTO 100
20 FORD=1024TD1063:POKE 0,C:NEXTD:FORD=1103TD2024STEP40:POKE0,C:NEXTD
30 FORD=2024TD1984STEP-1:POKE0,C:NEXTD:FORD=1984TD1024STEP-40:POKE0,C:NEXTD
40 FORD=55296TD55335:POKE0,X:NEXT:FORD=55375TD56295STEP40:POKE0,X:NEXTD
50 FORD=56295TD56256STEP-1:POKE0,X:NEXTD:FORD=56256TD55296STEP-40:POKE0,X:NEXTD
60 X=X+1:IF X>254 THEN X=1
70 GET A$:IF A#="" THEN RETURN
80 END
90 RETURN
100 PRINT CHR$(147)
110 PRINT TAB(10)"M KERET VILLOGTATAS":PRINT TAB(12)"MOC64/128 GEPEKRE"
120 PRINT:PRINT:PRINT TAB(3)"NYOMJ MEG EGY BILLENTYUT, HA VEGE !":GOSUB 20
130 GOSUB 40
140 GOTO 130

READY.
    
```



**SZÁMALK-CED**  
**Kereskedelmi Kft.**

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

UTAX-Fénymásolók és teljes tartozékok (12-féle géptípus)	39 900 Ft-tól + ÁFA
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!**

# BOX

AMIGA

Tippek  
Trükkök

```

\*
    BOX x y w h col

    AztecC:  cc +l <file>
            ln <file>.o -lc32 -lm -o

    Code by TPE

*/

#include <functions.h>
#include <exec/types.h>
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>

struct IntuitionBase  *IntuitionBase;
struct GfxBase        *GfxBase;
struct RastPort       *RP;

main(argc,argv)

int     argc;
char   *argv[];
{
    int     x,y,w,h,col;

    if(argc != 6)
    {
        printf("USAGE: BOX x y w h col\n");
        exit();
    }

    if(!(IntuitionBase=(struct IntuitionBase *)
        OpenLibrary("intuition.library",0))) exit(FALSE);

    if(!(GfxBase=(struct GfxBase *)
        OpenLibrary("graphics.library",0)))
    {
        CloseLibrary(IntuitionBase);
        exit(FALSE);
    }

    RP = &(IntuitionBase->ActiveScreen->RastPort);

    x = atoi(argv[1]);
    y = atoi(argv[2]);
    w = atoi(argv[3]);
    h = atoi(argv[4]);
    col= atoi(argv[5]);

    SetAPen(RP,col);

    Move(RP,x,y);

    Draw(RP,x+w,y);
    Draw(RP,x+w,y+h);
    Draw(RP,x,y+h);
    Draw(RP,x,y);
    CloseLibrary(IntuitionBase);
    CloseLibrary(GfxBase);
    exit(TRUE);
}
    
```

Ez a program nagyban hasonlít a már ismertetett CIRCLE nevűre. A különbség csupán annyi, hogy míg a CIRCLE köröket rajzolt a CLI képernyőjére, addig a BOX téglalapokat.

Az utasítás formátuma:

BOX x y w h col

ahol az

x — a téglalap bal felső sarkának x koordinátája

y — a sarok y koordinátája

w — a vízszintes oldal hossza

h — a függőleges oldal hossza

col — a rajzolás színének száma

## SZORZÓTÁBLA GYAKORLÓ

A szorzótábla gyakorlása nem kell, hogy mumus legyen a gyerekeknek. A program beírása, mentése és futtatása után véletlenszerűen jelennek meg a szorzási feladatok. A gyerek csak a helyes válasz megadása után mehet tovább a következő feladatra, illetve léphet ki a programból.

```

0 REM *****
1 REM * SZORZOTABLA GYAKORLAS *
2 REM *****
3 REM
4 REM
5 CLR:A#=CHR$(32):B#="*":C#=""
10 A=INT(RND(0)*99)+1:A=INT(A/10)
20 B=INT(RND(0)*99)+1:B=INT(B/10)
30 PRINTA#;A;B#;B;C#
40 INPUT "A KOVETKEZO A VALASZ";D
50 C=A*B
60 IF D=C THEN 80
70 PRINT "ROSSZ":GOTO 30
80 PRINT "HELYES ! AKARSZ MASIKAT ?"
90 GET A#:IF A#="" THEN 90
100 IF A#="I" GOTO 5
READY.
    
```

## MOZGÓ SZÖVEG

Ezzel a rutinnal címszövegeinket tehetjük mozgalmasabbá. A szöveget írjuk a T\$-ba (10-es sor), majd figyeljük a képernyőt. A szöveg egyre csökkenő amplitúdóval ugrál, majd megáll a képernyő közepén, aztán újra kezd.

Tippek  
Trükkök

```

0 REM *****
1 REM * MOZGO SZOVEG *
2 REM *****
3 REM
4 REM
10 POKE 53281,0:POKE 53280,0:PRINT CHR$(147):T#="*** IDE IRO A SZOVEGET ***"
20 DEF FNT(X)=(40-LEN(T#))/2
30 Y=23:X1=1:X2=23:POKE214,Y:PRINT:PRINT TAB(FNT(.))T#
40 FOR I=X2 TO X1 STEP -1:PRINT CHR$(147):POKE 214,I:PRINT:PRINT TAB(FNT(.))T#
50 FOR P=1 TO 2:NEXT:POKE646,INT(RND(1)*(16-2)+2):NEXT:X2=X2-1
60 IF X2=12 THEN 20
70 FOR I=X1 TO X2:PRINT CHR$(147):POKE 214,I:PRINT:PRINT TAB(FNT(.))T#
80 FOR P=1 TO 2:NEXT:POKE 646,INT(RND(1)*(16-2)+2):NEXT:X1=X1+1
90 GOTO 40
READY.
    
```

```

;
;      ADDMem.s
;
;      Seka assembler
;
movem.l  a0-a6/d0-d7,-(a7)
move.l   $0004,a6
move.l   start(pc),a0
move.l   end(pc),a1
suba.l   a0,a1
move.l   a1,d0
moveq    #$05,d1
moveq    #$00,d2
move.l   #$00000000,a1
jsr      -618(a6)
movem.l  (a7)+,a0-a6/d0-d7
rts

start:   dc.l   $200000,0
end:     dc.l   $3ffffff,0
    
```

## ADDMem

Ez az assembly program azoknak a felhasználóknak nyújt hasznos segítséget, akik nem „auto-boot-os” memoriabővítővel rendelkeznek.

Használata felhívja a gép figyelmét a külső kiegészítésre, így használni is tudjuk azokat.

Ugyan igen sokféle programmal lehet biztosítani ezt a hatást, de azok mind 2000 byte-on felüli hosszal rendelkeznek, tetemes helyet elfoglalva így a lemezen.

Ez a programocska a maga 96 byte-jával szépen meghúzza magát és könnyebben, gyorsabban biztosítja a kívánt hatást.

A bootolás kezdőcímet a START változóban kell beállítani (ezt nem érdemes megváltoztatni), a végcímet értelem szerűen az END-ben.

A program az alap beállítással egy 2 MByte-os memóriát tud kezelni.

AMIGA

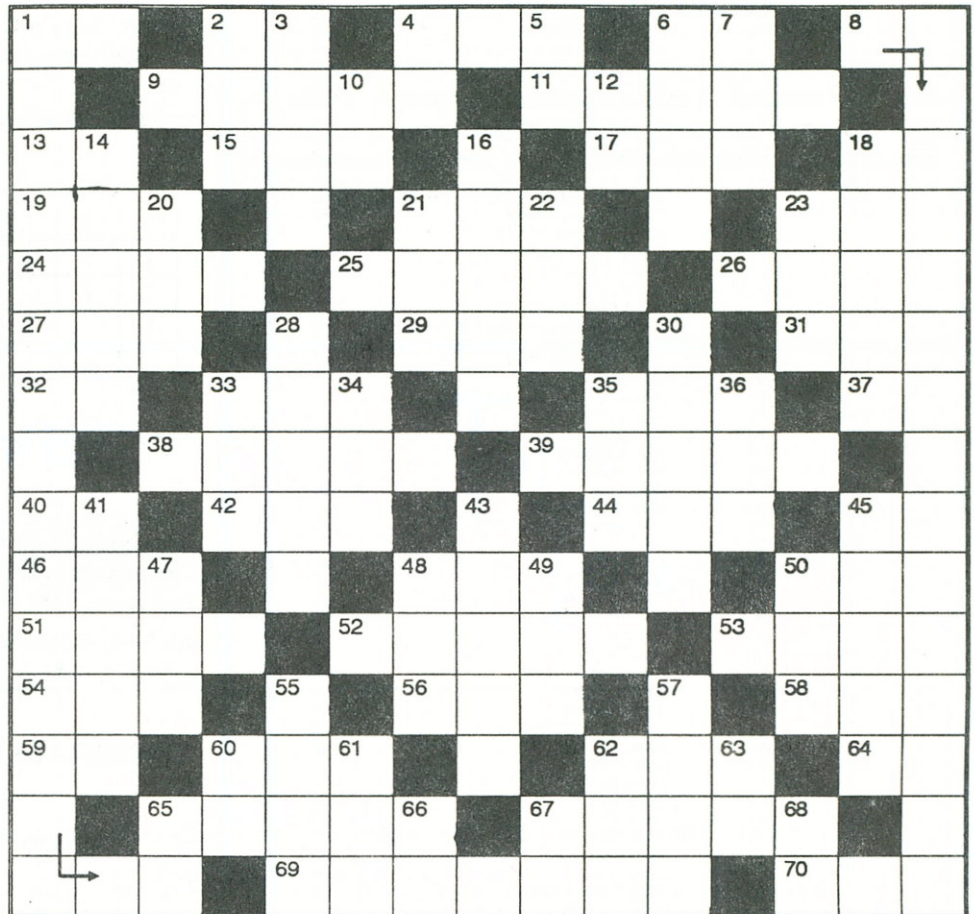


**FRANCE, ANATOLE (1844—1924)** A fehér kövön című munkájából rejtettünk el egy sort a következő sorrendben: — Függőleges 1., vízszintes 8. sorokban.

**VÍZSZINTES:** 1. Névelő. 2. Egyszerű számszám. 4. Krétei hegy. 6. Ozmium vegyjele. 9. Város Michigan államban (USA). 11. Tetőpont. 13. Éktelen számszám. 15. Ótóriai ital. 17. Szövőállat. 18. Folyadék. 19. Szeszes ital. 21. Sporteszköz? 23. Thaiföldi, spanyolországi, kambodzsai gépkocsik jelzései. 24. Végtelenül a halott gyermekeit sirató anya jelképe. 15. Eb név. 26. Fél-sziget Ukrajnában. 27. Hiányosan Varsó. 29. Több homokszivattyó a Szaharában. 31. Karon is viseljük. 32. Gát páratlan betűi. 33. Tartó. 35. Nyakát szegte pap. 37. Fogó közepe. 38. Görög szabály. 39. Megvehető. 40. Orosz folyó. 42. Orosz uralkodó volt. 44. Ilyen szakadás is van. 45. Francia részvénytársaság. 46. Ből-társaság. 48. Vaspálya. 50. Egészséges ital. 51. Ilyen szigetek is vannak. 52. Tejtermék. 53. Ritka férfinév. 54. Férfinév. 56. Kórházi osztály. 58. Magyar Optikai Művek röviden. 59. E nap. 60. Maró anyag. 62. Amerikai Egyesült Államok röviden. 64. Németh Béla névjele. 65. Tyúk termék. 67. Égtáj. 69. Angol filmsorozat címe. 70. Holló beszéd.

**FÜGGŐLEGES:** 2. Létezett. 3. Német kikötőváros. 4. Fitt közepe. 5. Azonos az 1. vízszintessel. 6. Helység Borsod-Abaúj-Zemplén megyében. 7. Terület. 10. Nátrium vegyjele. 12. Csak félig eped. 14. Könyvből másolt. 16. Remegés. 18. Másoló. 20. Kérdőszó. 21. Étélizestítő. 22. Azonos a 29. vízszintessel. 23. Keverve tor. 28. E név a zsidóban: galamb. 30. Német

## KERESZTREJTVÉNY



ragyogás. 33. Fejér megyei helység. 34. Ilyen társ is van. 35. Azonos a vízszintessel. 36. Azonos a 4. vízszintessel. 41. Testrész. 43. Német pala. 45. Vezetéknév is lehet. 47. Kínai hossz mérték? 48. Vulcanus. 49. Latin semmi. 50. Rokonom. 55.

Jelky András 1973. július 30-án itt született világhíró, utazó. 57. Szovjetunió egyik népe. 60. Étélizestítő. 61. Dunamenti város. 62. Azonos a vízszintessel. 63. Névelők. 65. Titán vegyjele. 66. Kétes(!). 67. Egészséges. 68. Egyforma betűk.

## A komfortos INPUT

Az INPUT parancs kérdőjele néha zavaró. Az egyik módszer ennek lenyeletésre az alábbi: `OPEN 1,0:INPUT#1,A$:CLOSE 1`. Itt az INPUT-ot a billentyűzet „file” megnyitásával helyettesítjük. De használhatjuk az INPUT előtt a `POKE 19,1` utasítást, amit azonban az INPUT után azonnal `POKE 19,0` hatástalanítani kell.

Ha a kérdőjel helyett mást akarunk, az alábbi sort aktiváljuk minden INPUT parancs előtt:

```
POKE631,20:POKE632,20:POKE633,(ASC("a kívánt jel")):POKE634,32:POKE198,4
```

## RESET adatvesztés nélkül

A programozás során gyakran előfordul, hogy szinte mindent elállítottunk (jelkészletek, képernyők stb.). Ilyen esetekben adjuk be az alábbi egysoros. Ennek segítségével az összes lényeges mutatót helyreállítjuk. Azonban ha nagyfelbontású grafikát is használunk, akkor a dolog nem fog működni, mert ilyenkor a program más területen áll mint amit itt feltételezünk. `POKE 648,4:SYS 64789:SYS 58451:SYS 58784`

Ez a BASIC sor egy programon belül is jól használható.



Eladó C64 + VC 1541 floppy, 19 000 Ft. Érd.: 15-től 19 óráig. Salamon, 1039 Budapest, Sarkadi u. 3. IV. 13.

Eladó! C64/II + 1541/II + 1802, színes monitor tartozékokkal (45 000), külön is. Batiz Norbert, 5630 Békés, Szőlő u. 24.

C64-es program cserелеmezen kb. 1200 db. program. Minden levélre válaszolok. Nagy Csaba, 1072 Budapest, Akácfa u. 30.

DS/DD 5,25"-os lemezek 35 Ft-tól! C64 programok 20—30 Ft/lemezoldal áron kaphatók. Olcsó ár — jó minőség! Nagy István, 1202 Budapest, Mártírok útja 151.

C64 programok és resetek olcsón eladók. Válaszborítékra listát küldök. Leveleket szerkesztőségbe. Tagsági szám: 1618

Vennék Commodore 64 vagy +4 típusú számítógépet olcsón. Illés Ferenc, 7629 Pécs, Sztahanov u.6.

C + 4-es programcsere kazettán. Közel 1500 db program. Listát kérek. Sipos István, 7400 Kaposvár, Mező u. 15.

Final III. Cartridge eladó magyar nyelvű leírással. Ára: 3800 Ft. Godina Zoltán, 8800 Nagykanizsa, Miklósfu u. 83.

C64-re programokat adok el olcsón. Válaszborítékra listát küldök. Kozut Antal, 1138 Budapest, Gyöngyösi sétány 12.

C + 4-re színvonalas programok cseréje és eladása (10 Ft/db). Válaszborítékért listát küldök! Program: 04. 06-án 890 db! Hartung Gábor, 6045 Ladánybene, Gödörálás dűllő 55.

Eladó C64, 1541 Drive. Ár megegyezés szerint. Neumann Balázs, Budapest, Tel.: 2512-695.

Geos programok adásvétele. Programok, német és magyar nyelvű információs kártyák. Geos programok rendelésre is. Geos 2.0, GeoCalc GeoCHart, MegaPach 1—2, DeskPah—GeoDex, Geofile. Válaszboríték esetén tudok a kérdésekre válaszolni, programok utánvétellel. Nagy Péter, 1056 Budapest, Molnár u. 20.

Olcsón eladó C64 + Drive + 2 Joy + prog., C 128 + Drive + Printer + Monitor. Monitor (IBM, C64) külön is. Gidai Olivér, Tel.: 1764-836

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

Eladó Rox—data technika, 256 kb-os eprombank 4000—3000 Ft-ért. Kazetták egység 2000 Ft-ért. Super Base könyv 150 Ft-ért. Nagy Péter, 1056 Budapest, Molnár u. 20.

C64-re programleírásokat keresek, cserébe másolok, lemezen és kazettán. Czifra Zoltán, 5231 Fegyvernek, Kiss János u. 18/a.

Amigához! Eladók új 3,5"-os Drive-ok 9000 és egy 14"-os Philips színes monitor. 3M DS/DD lemezek és német nyelvű irodalom 950 Ft. Üzenet: Bakos Gábor, 1-862-310.

Keresem Geos 1.3 és Geos V 2.0 lemezen ármegjelöléssel. Apró Lajos, 4220 Hajdúböszörmény, Tánicsics krt. 72.

C64 Giga-cad leírás eladó. Lemezen 150 Ft, nyomtatva 400 Ft. Apró Lajos, 4220 Hajdúböszörmény, Tánicsics krt. 72.

C64 + floppy + 2 Joy eladó. Zolnai László, 3520 Miskolc, Vörösmarty u. 72. Tel.: (46)323-493, Fax: (46)364-327.

Magnósok, figyelem! Eladók utánvétellel a legjobb programok kazettán. Listát küldök. Árak: 1 db kazetta + 1 db kollekción + postaköltség = 340 Ft; 1 db kazetta + 2 db kollekción + postaköltség = 450 Ft; 2 db kazetta + 4 db kollekción + postaköltség = 900 Ft. Minden további kazetta 400 Ft. Maximum megrendelhetőség 10 kazetta. Cím: 3300 Eger, Bercsényi út 51/a. II/19. Tel.: 19-085.

C64 és Amiga sikerprogramok olcsón eladók lemezen. C64-re képes katalógust, Amigára ismertető listát küldök. Gyorsaság, megbízhatóság, egyedülállóan olcsó árak. Zymosis Soft, 2621 Verőce, Asztalos János u. 8/b.

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

Plus 4 + Magnó + 500 program + irodalom + 1 db joystick eladó. Ára: 10 000 Ft. Érdeklődni: 46/328-938 18 óra után lehet. Kalászi Pál, Miskolc, Vándor u.4.

C 64-es programok eladók 90 Ft/lemez. Válaszborítékért listát küldök. Farkas István, 8900 Zalaegerszeg, Kelemen u. 16.

C64-es gépet adok C+4 gépért cserébe. Ormos Zoltán, Budapest, 4. sz. Postán maradó 1364. vagy tel.: 133-5156.

C64-es kazettások! Rengeteg utántöltős játék eladó! 15 Ft-os bélyegért listát küldök. Szűcs Sándor, 6440 Jánoshalma, Pf.: 18.

C64-re adok olcsón programokat. Ára: 5 Ft. Csere is érdekel. Válaszborítékért listát küldök. Eladó C64 + 1541 II floppy + 40 lemez + színes monitor + 2 joy. 40 000 Ft. Érdeklődni lehet: Varga János, Nagytarcsa, Rákóczi u. 63. Csak levélben.

Ujdonság! C64 alsósoknak, — oktatóprogram-csomag, 80 program + leírás együtt csak 1500 Ft, kazettán 1900 Ft. C64 nyelvtan — az általános iskolások számára csak lemezen, leírással együtt 1500 Ft. Tom számolni tanít —1. osztályosoknak TVC-re, C + 4-re /10 program, 1200 Ft lemezen, illetve kazettán. Anyanyelvünk — alsósoknak TVC-re, C + 4-re /8 program együtt, lemezen illetve kazettán 1200 Ft. Megrendelhető: SULI-Soft 1327 Újpest 3, Pf.: 91.

C64, C128, AMIGA programlemezek eladók 70 illetve 80 Ft/db. Válaszborítékért listát és tájékoztatót küldök. Kopácsi Lajosné, 1031 Budapest, Vízimolnár u. 2. X/95.

Amigához 1084S Monitor, 512k, hang, képdigik, 800 lemez, tele programokkal, küldő drive 3,5; 5,25. Németi F. 1675 Budapest, Pf.: 116.

Plus 4-es programok cseréje, eladása kazettán. Nagy Attila, 1223 Bp., Tüzliliom u. 17.

C128-at keresek megvételre. Konkrét árajánlatot kérek az alábbi címre: Király Richárd, 3245 Recsk, Arany J. u. 25.

Yo! Amigások! Ha érdekelnek a legújabb programok külföldről, írj!!! Soldier, 1188 Budapest, Damjanich u. 22.

Keresem Seres—Fenyő—Gyalogh: A forth programozási nyelvkönyvet. Pályi Sándor, 8182 Peremartonygyártelep.

Keresem C64-re lemezen a Turry CanII Katakis Simcity, Dragonwars, Oil Imperium, Last Ninja II—III, Rick D. II. játékokat. Lipták Zoltán, 2131 Göd, Váci M. u. 28.

Számlázó, Raktárnyilvántartó, SZJA-számító, Útnyilvántartó, Pénztárkönyv, Naplófőkönyv programok C64-re és PC-re. Tel.: 1789-213.

**AGFA** *Agfa*

fénymásoló rendszer

## 10 000 forintos vásárlási utalvány

Beváltható fénymásoló vásárlása esetén

 az **ASI** *ATI* Kft.-nél

 Budapest XI., Bartók Béla út 120.  
Telefon: 185-1507, fax: 185-1760

Érvényes: 1992. június 30.

 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, Szilgony u. 9.

Telefon: 134-3153

1191 Budapest, Gábor Á. sétány 3.

Telefon: 127-4763

3525 Miskolc, Fazekas u. 1—3.

Telefon: 46-17-011

4034 Debrecen, Holló L. u. 14.

Telefon: 52-32-863

5600 Békéscsaba, Bartók B. u. 37.

Telefon: 66-27-195

6724 Szeged, Csongrádi sugárút 76.

Telefon: 62-13-377

7624 Pécs, Jurisics M. u. 17.

Telefon: 72-11-812

8000 Székesfehérvár, Széchenyi u. 15/a.

Telefon: 22-12-711

9700 Szombathely, Szalonok u. 31.

Telefon: 94-13-419

Felvevőhelyek:

9024 Győr, Babits M. 75.

Telefon: 76-23-720

6000 Kecskemét, Széchenyi tér 1—3.

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

**NOVOTRADE**  
SZERVÍZ Kft.

## 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:

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

### Kedvezmény nélkül:

C64-be átkapcsolható új operációs rendszer (Speed) + reset beépítése:	2000 Ft
1541 kompatibilis lemezegységbe Speeddos beépítése (átkapcsolhatóan)	
40 TRACK (+ 85 blokk/lemezoldal), valamint párhuzamos 15 pólusú Canon csatlakozó beépítése:	2000 Ft
C64 USER-port 1541-es lemezegység összekötő párhuzamos kábel:	1300 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése:	800 Ft
PAGEFOX magyar ékezetes szövegszerkesztővel rendelkező cartridge:	6000 Ft
(Epson típusú nyomtató min. 640 képpontos szükséges a nyomtatáshoz)	
FASTLOAD (lemezes gyorsító, másoló, monitor)	1400 Ft
TTL IC-TESTER cartridge + program	4300 Ft
288/256 Kbyte-os eprombank (vezérlő eprommal)	5340 Ft
Epromégető (2716-tól 27256-ig)	4300 Ft
8—16 Kbyte-os epromkártya (cartridge, eprom nélkül)	600 Ft
C64-hez tároló oszcilloszkóp (párhuzamos kábel nélkül)	7500 Ft
A háttértárakhoz epromok programozása (kész programok, vagy saját, hozott programok beégetésével) egységesen:	500 Ft
C64 bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
256K RAM-diszk	13000 Ft
64/256K RAM-diszk	9000 Ft
256K RAM-diszk (RAM-ok nélkül)	7500 Ft
64 Kbyte-os cartridge komplett programokkal, vagy igény szerint összeállítva	2500 Ft

 A fenti bővítések megrendelhetők levélben, vagy az OCE irodájában személyesen.  
Ha személyesen kívánja megrendelni, kérjük, előtte telefonáljon.

Árainkat az alkatrészárak változásai befolyásolhatják.

**JÚNIUSI**

## 60 Ft-os vásárlási utalvány

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

Érvényes: 1992. június 30.

**TKD-BUDAPEST**
**JÚNIUSI**

## 400 Ft-os vásárlási utalvány

 Beváltható: TKD-BUDAPEST  
1117 Budapest, Schönhercz Z. u. 21.

 Egy szótárgép vásárlásához  
egy utalvány használható fel!

Érvényes: 1992. június 30.

# Cserélhető lemezes winchester!

A SyQuest Technology  
hivatalos magyarországi disztribútora:

**NOVOTRADE**  
SZERVIZ Kft.

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



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

Áraink végfelhasználói árak és ÁFA-t nem tartalmaznak.

## Meghatalmazott dealereink:

Microteam Kft., 1145 Budapest, Róna u. 127.  
Tel./fax: 184-1226

Professzionál Kft., 1033 Budapest, Kaszásdűlő u. 5.  
Tel.: 167-0024 fax: 167-0289

Professzionál Kft., Miskolci Kirendeltség,  
3525 Miskolc, Szabó L. u. 37.  
Tel.: 46/56-079

Professzionál Kft., Békéscsabai Kirendeltség,  
5600 Békéscsaba, Andrásy u. 75.  
Tel.: 66/28-584

Korall Kft., 2800 Tatabánya, Március 15. út 3.  
Tel.: 34/11-714

3S Computer Kiszövetkezet, 6723 Szeged, Kemes u. 6.  
Tel.: 62/26-277, Fax: 62/26-347

Volánelektronika Vercomp Kft., 9024 Győr, Dr. Petz Lajos u. 7.  
Tel./fax: 96/12-520

Novotrade Miskolc Kft., 3530 Miskolc, Vörösmarty u. 51.  
Tel./fax: 46/49-489

Elektrosoft Kft., 5000 Szolnok, József A. u. 6-8.  
Tel.: 56/42-880, Fax: 56/44-222

Számadó Kft., 6000 Kecskemét, Dózsa Gy. u. 29.  
Tel.: 76/21-455, Fax: 76/21-462

BX-Next Kft., 3434 Mályi, Bercsényi út 50.  
Tel.: 46/91-117

Onyx Szoftver Kft., 1118 Budapest, Mányoki u. 14/B.  
Tel.: 165-3325

Novotrade PC Kft., 1136 Budapest, Sallai u. 25.  
Tel.: 149-0798 Fax: 131-0734

Almárium Kft., 1137 Budapest, Pozsonyi u. 21-23.  
Tel.: 111-2830 Fax: 112-3647

Microchip Kft., 8000 Székesfehérvár, Élmunkás u. 47.  
Tel.: 22/25-514

Navigátor Kft., 4400 Nyíregyháza, Kórház út 26/B.  
Tel./fax: 42/41-972

Alfadat Kft., 2803 Tatabánya, Tóth-Bucsoki út 12.  
Tel.: 34/10-234, 10-405, Fax: 34/10-729

Digitech Kft., 7101 Szekszárd, Rákóczi u. 6.  
Tel.: 74/16-874

Interface Kft., 1116 Budapest, Hunyadi J. út 162.  
Tel.: 166-5322/58, 55, Fax: 226-3793

Netrend Rt., 1089 Budapest, Elnök u. 1.  
Tel.: 113-8217, Fax: 113-9537

Omnis Kft., 2840 Oroszlány, Münnich F. u. 23.  
Tel.: 34/60-832

2R Periféria Kft., 1071 Budapest, Peterdy u. 30.  
Tel.: 122-3034 Fax: 142-3308

ProComp Kft., 8900 Zalaegerszeg, Bíró M. út 8. Pf.: 275.  
Tel.: 92/11-373

A magyar olimpiai csapat arany fokozatú támogatója



input: **MAX** output: **maximum**

A TUNGSRAM-MAX mágneslemez japán és amerikai alapanyagból, amerikai technológiával, high-tech berendezéseken készül. Minden egyes mágneslemez hibamentességét a teljes felület számítógépes mérőrendszerrel történő tesztelése garantálja.

**TUNGSRAM-MAX mágneslemezek**

Standard	Formázott	Színes lemezek műanyag dobozban
5,25" TM 2S2D 53 Ft	5,25" TMF 2S2D 61 Ft	5,25" TMP 2S2D 63 Ft
5,25" TM 2SHD 77 Ft	5,25" TMF 2SHD 87 Ft	5,25" TMP 2SHD 88 Ft
3,5" TM 2S2D 88 Ft	3,5" TMF 2S2D 102 Ft	3,5" TMP 2S2D 99 Ft
3,5" TM 2SHD 155 Ft	3,5" TMF 2SHD 173 Ft	3,5" TMP 2SHD 167 Ft

Árainkhoz ÁFA-t számítunk!

- CSEREGARANCIA: 20 ÉV VAGY 20 MILLIÓ FORDULAT!
- Mágneslemezek: no name és bulk csomagolásban is!
- Tárolódobozok, tisztítókészletek 3,5" és 5,25" méretben.
- Szoftvermásolás profi gépeken, írásvédő kivágás nélkül is!
- Vizonteladóknak 20% engedmény!
- Szoftverkészítőknek, nagyfelhasználóknak, diákoknak rendkívüli kedvezmények!
- Színes és formattált mágneslemezek, tárolódobozok, festékszalagok árusítása, szoftvermásolás és csomagolás, címkézés a szoftverkészítő igénye szerint.
- Kérje részletes árlistánkat!
- Tungsram Magnetic Media Budapest IV., Váci út 77. Tel.: 160-2233 Fax: 160-0925



**TUNGSRAM-MAX<sup>®</sup>**

**KÉPVISELETEINK:**

Agromark Kft. — Hódmezővásárhely,  
Andrássy út 50. Tel.: 06-62-41695  
Comtrade Kft. — Pécs,  
Majorossy u. 36. Tel.: 06-72-26063  
M és K Bt. — Kecskemét,  
Bajcsy-Zsilinszky u. 5. Tel.: 06-76-21878  
PGM Computer Kft. — Szeged,  
Csongrádi sugárút 22. Tel.: 06-62-14380  
Számker Kft. — Zalaegerszeg,  
Rákóczi út 4—8. Tel.: 06-92-14500/144  
Transzfer Kft. — Nyíregyháza,  
Hősök tere 7. Tel.: 06-42-10481

**HIVATALOS DEALEREINK:**

Novotrade 2C Kft. — Budapest XIII.,  
Balzac u. 35. Tel.: 140-2954  
Radiant Kft. — Budapest XIV.,  
Francia út 11. Tel.: 252-1999/266  
Westing Iroda — 1149 Budapest,  
Róna köz 12. Tel.: 163-7916

**TUNGSRAM MAGNETIC MEDIA RT.**  
H-1340 Budapest, Váci út 77.  
Tel.: 160-2233, Fax: 160-0925

**TUNGSRAM  
MAGNETIC MEDIA**



A magyar olimpiai csapat arany fokozatú támogatója