

**commodore**

IX.évfolyam  
**1994/5**

**újság**

Az Országos  
Commodore  
Egyesület lapja



**ÚJ ÚJ ÚJ ÚJ ÚJ**

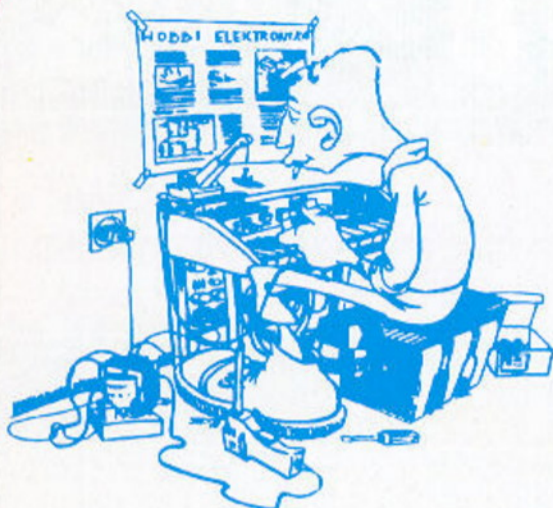
**PÁLYÁZAT**

**Epromégető – Quasi tömbváltozók**

**Az Amiga család története**



SZERENCSEJÁTÉK RT



# HOBBI ELEKTRONIKA

**Super  
ajánlat!**

**Urbán István mérnök áramköreinek szaküzlete**

Budapest VII., Dózsa György út 16. (Dózsa-Jobbágy sarok)  
Nyitva: H-P 10-17-ig Tel./fax: 122-8892  
(Zárás után üzenetrögzítő)

## KÍNÁLATUNKBÓL

### SZÁMÍTÁSTECHNIKA:

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

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

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

## C-64 BÖRZE

Használt C-64-et, floppy adok-veszek.  
Hibás gépet beszámítok, megvásárolok.  
Megegyezés szerint  
új IBM PC részegységre cserélek.

### Áraink a 25% ÁFA-t tartalmazzák!

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

### MŰSZERVÁSÁR

Bontott anyagok,  
használt műszerek  
bőngésződjé



Állandóan változó készlet!

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

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

**Levélcím: 1656 Budapest Pf. 50.**

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



## MIT, HOGYAN, HOL, MIKOR?

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

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

A **DEÁKPÁHOLY** tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy évre 1020 Ft.

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

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

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

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

**APRÓHIRDETÉS:** Az egyesületi tagoknak ingyen áll rendelkezésére. Nem tagoknak a hirdetés ára 300 Ft. A hirdetés módja: az újságban megjelenő nyomtatvány kitöltésével.

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

Kedvezményes ár! Tagoknak olcsóbb!

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

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

### VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:

Baja, AXIS Kft.

Győri Bartók Béla Művelődési Ház,

Jászberényi Városi Könyvtár,

Kecskemét, SZIGMA—BIT,

Pécsi Apáczai Csere János Gimnázium,

Zalaegerszegi Ságvári Endre Gimnázium.

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:

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

Felelős kiadó: Horváth Judit, az egyesület elnöke

Főszerkesztő: Rados Péter, az OCE főtítkára

Felelős szerkesztő: dr. Horváth András

Művészeti szerkesztő: Bausz Sándor

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

Index: ISSN 0237-756 X

Terjeszti a Nemzeti Hírlapkereskedelmi Rt. és

a regionális részvénytársaságok

Megvásárolható a hírlapárusoknál

94.0083 MSZH Nyomda és Kiadó Kft., Budapest

Felelős vezető: Nagy László

## Tisztelt Egyesületi Tagok, Kedves Olvasóink!

Bizonyára sokaknak feltűnt, hogy az utóbbi időben lapunk nemcsak Commodore 64 és PLUS4 témákkal foglalkozik. A Váltó sorozat, a PC-s alkalmazói programok ismertetése, a Commodore és PC közti adatátviteli lehetőségekről szóló írások, a PC-s hardver kiegészítők jelzik, hogy nyitunk a személyi számítógépek irányába.

Ez a nyitás azonban nem jelenti a jó öreg Commodore elhagyását. Célunk továbbra is a számítástechnikával most ismerkedők, a gépet már nemcsak játszoeszköznek tekintők segítése. Azonban nem hagyhatjuk figyelmen kívül azt a folyamatot, melynek eredményeként ma már egyre több családban van PC és egyre többen a programozás alapjait is ezen kezdik elsajátítani.

Továbbra is várjuk olvasóink beküldött írásait, programjait. Szeretnénk bátorítani azokat, akik a programozást PC-n kezdik. Bizonyára sok olyan témáról tudnak írni, ahol a mondanivaló lényege nem szigorúan egy géptípushoz kötődik, hanem algoritmikus, programozás-technikai fogás, a mellékelt program vagy programkészlet pedig ennek illusztrációja, amit — legyen az bár GWBASIC-ben írva — az is nyomon tud követni, aki eddig csak a C=-gépek BASIC nyelvét ismerte.

Kérjük munkatársainkat, cikkeik beküldésekor képzeljék magukat egy pillanatra a szerkesztők helyébe és lehetőleg nyomdakész anyagot küldjenek. Nem tudunk mit kezdeni az olvashatatlan, kézzel írott szövegekkel. Arra meg különösen nincs módunk, hogy nyomtatott, vagy pláne kézzel írott programlistákat kipróbálás végett beírjunk. Ezért kérjük, a programokat adathordozón — lehetőleg lemezen — küldjék.

Célunk olyan írások közzététele, melyből az olvasó információt merít, az olvasottakat saját munkájában felhasználhatja. Ennek pedig magyarázat nélküli, hosszú programlisták esetében kicsi az esélye. Inkább kevesebb, rövidebb programot, programrészletet, de az legyen úgy dokumentálva, hogy a működés az olvasó számára is világos legyen.

Tagjaink és régebbi olvasóink előtt közismert, hogy Egyesületünk rendszeresen pályázatok kiírásával ösztönzi egy-egy témakörbe vágó programok beküldését. A pályázatokkal járó tevékenységek — a téma kiválasztása, a feltételek megvitatása, a beérkező pályázatok adminisztrációja, a szakzsűri összeállítása, a pályázatok értékelése, a díjazás, esetleg szponzorok keresése, a díjkiosztás lebonyolítása — kis egyesületi gárdánk számára külön feladatot jelentenek. Ezt azonban szívesen vállaljuk, ha úgy érezzük, jó cél érdekében tevékenykedünk.

Jelenleg is ilyen pályázat kiírására készülünk. Úgy gondoltuk, ezúttal egy konkrét, algoritmikus feladatot tűzünk ki. Ezzel elsősorban azt szeretnénk illusztrálni, hogy — a fentiekkel összhangban — egy nem túl nehéz algoritmikus probléma megoldása szinte gépfüggetlen, a nyerő program PC-n ugyanúgy elkészíthető, mint C-64-en. Ezen kívül a pályázatok rangsorolása is teljesen objektívvé tehető.

Országos Commodore Egyesület

(Pályázati felhívásunk a 4. oldalon.)

# PÁLYÁZAT

Az Országos Commodore Egyesület pályázatot hirdet. Jellege: Algoritmus pályázat. A pályázaton bárki részt vehet. A pályázat célja egy konkrét játék algoritmusának megvalósítása.

Sokan kedvelik — lapunkban is többször szerepelt — a leginkább talán MASTER MIND néven közismert játékot. Az 1992/11. számban részletesen írtunk róla. Lényege röviden: Két játékos játssza. Az egyik elrejt egy négybetűs kódot, ahol a betűk A-tól F-ig terjedhetnek. Például CFCA. A másik játékos megpróbálja ezt kitalálni úgy, hogy ő is ilyen kódokkal kísérletezik. Az első játékos jelzi a találatok számát, mégpedig a saját helyén eltalált betűért egy világos, a rossz helyen eltalált betűért egy sötét pontot ad. Példánkban ha a tipp BFDC, akkor egy világos pontot adunk a saját helyén eltalált F-ért és egy sötétet a rossz helyen eltalált C-ért. (Figyelem, egy C-vel csak az egyik tipp-beli C-t lehet eltalálni, és viszont, ha a tipp FBFD, erre csak egy sötét pont jár, mert az elrejtett kódban csak egy F van.) A játék a kód kitalálásáig (négy világos pont), vagy egy előre adott lépésszám-korlát eléréséig tart.

A pályázat tárgya annak a BASIC programnak a megírása, mely nem hosszabb 300 sornál és a lehető leggyorsabban kitalálja az elrejtett kódot. A beérkező pályaműveket úgy fogjuk elbírálni, hogy generálunk véletlenszerűen néhány (5–10) kódot, majd ezeket feladjuk a programnak. A program tippjeire válaszul beírjuk a világos és sötét találatok számát. A program pontszámát a tippek kitalálására fordított lépések számának összege adja. A legkisebb pontszámú program a nyertes.

A programot C-64, C/PLUS4 BASIC, vagy PC esetén GWBASIC nyelven, hajlékony lemezen kérjük beküldeni az egyesület címére.

**Beküldési határidő: október 10.**

A pályázat díjai számítástechnikai eszközök és kiegészítők.

**C-64 C+4 AMIGA PC  
tulajdonosok!**

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

Cím: 1181 Budapest, Kondor Béla sétány 8.  
Megközelíthető: a metró kőbánya–kispesti végállomásától a piros 136-os busszal.  
Legközelebb:

**május 15-én.**

## FELHÍVÁS

Tisztelt Tagtársunk, Olvasóink!

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

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

Köszönettel:

Országos Commodore Egyesület

## Tippek — Trükkök

### SZINEZŐDŐ KARAKTEREK

SYS49152 parancs beírása után a gép kurzorvillanásonként színezi a karaktereket.

```

○ 100 REM *****
○ 110 REM * SZINEZODO KARKATEREK C-64*
○ 120 REM * , , AKT = SYS49152 *
○ 130 REM * KESZITETTE : BRUCE MAESTRO *
○ 140 REM * (OLASZ ENDRE) *
○ 150 REM *****
○ 160 :
○ 170 FOR I= 49152 TO 49165
○ 180 : READ A:POKEI,A:S=S+A
○ 190 NEXT
○ 200 IF S<> 4710 THEN PRINT"HIBA"
○ 210 DATA 120,162,013,160,192,142,020
○ 220 DATA 003,140,021,003,088,096,238
○ 230 DATA 134,002,162,250,160,250,232
○ 240 DATA 224,000,208,251,200,192,000
○ 250 DATA 208,246,076,049,234,234
○

```

**Alkalmazói  
program  
pályázat**

# Kalkulátor-szimulátor

Ez úton megküldöm Önöknek az Önök által meghirdetett „Alkalmazói program pályázat”-ra programomat, melynek címe: Kalkulátor-szimulátor.

A program mint neve is mutatja, egy kalkulátort szimulál a C-64-esen. A program a lemez A oldalán található és 67 blokkot foglal el.

Indítás után kb. 40 másodpercet várni kell. Ez alatt az idő alatt a program áttölti a karakterkészletet, a szükséges karaktereket átírja, felrajzolja a képernyőt, beállítja a megfelelő változókat.

A képernyő 3 fő részre tagolódik:

- a két szélén 4–4 kisebb téglalap áll, melyek a választható menüpontokat és a funkcióbillentyűk nevét mutatják;
- a 4–4 téglalap egy nagyobbat fog közre, mely a főképernyő. Itt történnek a fontosabb adatbevitel;
- legalul az információs lap található. Ez tartalmazza az X, Y koordinátákat, azt a funkciót, amiben vagyunk, a szabad memóriát, a lapok számát (ezekről bővebben később).

Indítás után hét lehetséges funkciót választhatunk:

- F1 Mem.: Memória. Kisebb feljegyzéseket írhatunk és tárolhatunk itt.
- F2 Tel.: Telefon. Különböző személyek nevét, címét, telefonszámát írhatjuk be.
- F3 Eml.: Emlékeztető. Események dátumát, időpontját, nevét tárolhatjuk.
- F4 Idő: Az időt és a riasztási időt állíthatjuk be.
- F5 Kal.: Kalkulátor. Itt egy számológéppel végezhetünk műveleteket.
- F6 Ini.: Inicializálás. Törli a legfontosabb adatokat a memóriából.
- F7 Vége: Visszatérünk a BASIC-be.
- F8: Kihaszíratlan funkció.

Nézzük kicsit bővebben ezeket a funkciókat.

## F1 MEMÓRIA

Lenyomása után újabb 7 funkciót kapunk és már a főképernyőn is villog a kurzor. Ekkor szabadon írhatunk. A következő billentyűket használhatjuk:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

0123456789

!"#\$%&'()

+ - / \* ↑ =

, : ; ? < > [ ]

Space

3 karakter változott meg:

– a gyökjel első fele √

@ Ö

£ Ü

Ékezetes karaktereket a SHIFT+magánhangzóval írhatunk. Az információs lapon máris változásokat láthatunk. X; Y tartalmazza azt a koordinátát, ahol a kurzor áll a főképernyőn.

FU az adott funkció, jelen esetben ez MEM. Memória a szabad lapok számát tartalmazza (tetszőlegesen átdefiniálható). Lan az adott lapok számát, melyet most szerkesztünk. Ves-sünk néhány pillantást 7 új funkciókra:

## F1 ELŐ

Egy lappal előre ugrik; ha már az utolsó lapon állunk, a 0-ra ugrik. Csak akkor ugrik, ha azt a képet, ahol F1-et nyomtunk, elmentette a memóriába, tehát ugrás után az adatok nem vesznek el.

## F2 HÁT

Ugyanaz, mint az F1 funkciója, csak itt hátrafelé ugrik (0 után az utolsóra). Az adatok itt is ugyanúgy megmaradnak, mint F1-nél.

## F3 TÖL

Az előzőleg már lemezen tárolt lapokat tölti vissza.

## F4 MEN

Az adott funkció összes lapját elmenti lemezre. Ez 40 lapnál 24 blokk.

FIGYELEM! A mentés nem törli az előzőleg kimentett azonos funkciójú file-t, mert az egy másik funkció feladata. Mentés előtt tehát mindig töröljünk!

## F5 UKM

Ugrás Közvetlenül Más lapra.

Erre a funkcióra azért van szükség, mert a sima előhátra lapozás a BASIC-ből adódóan elég lassú, főleg rossz ez olyankor, ha pl. az 5. lapról a 23-ra akarunk eljutni. Így csak lenyomjuk az F5-öt, a program elmenti a képet a memóriába, és már lapozhatunk is a +, – billentyűkkel. Hogy jelenleg melyik lapon vagyunk, azt az információs lapból tudhatjuk meg. Ha a kívánt lapon vagyunk, nyomjunk SPACE-t vagy RETURN-t.

## F6 FFT

Funkció File Törlése

Ez a funkció egy, már korábban elmentett adatfile-t töröl a lemezről, hogy az új adatok újra lemezre menthetők legyenek. Erre azért van szükség, mert a MEM funkció mindig MEM-ként tárolja adatait, és ha újból mentenénk, két MEM file lenne a lemezen, amit a meghajtó nem néz jó szemmel...

## F7 VÉGE

Visszatérhetünk a főmenübe úgy, hogy a szerkesztett adatokat a program a memóriába elmenti.

## F2 TELEFON

Teljes egészében azonos a Memóriával, de itt a főképernyőn szövegek is lesznek, amit a gép ír ki (NÉV, TEL., CÍM). Ezekre lehetőleg ne írjunk, mert bár a program lemezre és a memóriába is így menti el, de újra megjelenéskor a szövegek alatti karaktereket nem fogjuk látni.

## F3 EMLÉKEZTETŐ

Ez is teljes egészében azonos a memóriával, és erre is érvényes, hogy a gép által kiírt szövegek alatti karakterek elvesznek.

## F4 IDŐ

A főképernyőn a pontos időt láthatjuk (ami most még nem pontos; ahhoz be kell állítani), alatta pedig a riasztási időt (alapállapotban 99:99:99). Itt további három funkcióhoz jutunk:

## F1 IBE

```

0 rem -----+++++++-----
1 rem - . -
2 rem - kalkulator szimulator -
3 rem + +
4 rem + c64-re +
5 rem + . +
6 rem + irta : +
7 rem - -
8 rem - zsila zsolt 1993.viii. -
9 rem -----+++++++-----
100 rem +- inicializalas +-
110 gosub 1940: rem +- karakter +-
120 gosub 2000: rem +- kezdokep +-
130 gosub 2180: rem +- valtozok +-
140 rem +- foprogram +-
150 gosub 2000: i=0: gosub 2300
160 for x1=0 to 15: for y1=0 to 7: poke sz+x1+y1*40,12: next y1,x1
170 if (val(ti$)=>val(eb$)) and c3=0 then gosub 2360
180 get a$: if a$="" then 170
190 a=asc(a$): if a<133 then 170
200 on a-132 goto 320,340,1420,230,360,1080,290,230
210 goto 170
220 rem +- v e g e +-
230 print "[Clr][8:Dwn]"
240 print tab(12)"[Blk]viszont lAtAsra !"
250 for t=0 to 1500: next t
260 sys 64738
270 end
280 rem +- i n i t +-
290 gosub 2200
300 goto 150
310 rem +- m e m o +-
320 fu$="mem.": fu=1: goto 370
330 rem +- e m l . +-
340 fu$="eml.": fu=3: goto 370
350 rem +- t e l . +-
360 fu$="tel.": fu=2
370 i=1: gosub 2300
380 gosub 2470
390 gosub 2520
400 if fu=1 then 490
410 if fu=2 then 460
420 x1=0: y1=0: u$="dAtum ": gosub 2590
430 x1=0: y1=2: u$="id[s@] ": gosub 2590
440 x1=0: y1=4: u$="emlE. ": gosub 2590
450 goto 490
460 x1=0: y1=0: u$="nEv ": gosub 2590
470 x1=0: y1=2: u$="telsz.": gosub 2590
480 x1=0: y1=4: u$="cIm ": gosub 2590
490 aa=peek(k+x+40*y): gosub 2470
500 if (val(eb$)=<val(ti$)) and c3=0 then gosub 2360
510 gosub 2610
520 get a$: if a$="" then 500
530 poke k+x+40*y,aa
540 a=asc(a$): if a-132<1 then 560
550 on a-132 goto 710,870,1000,770,740,780,970,770
560 gosub 2670
570 if a>63 and a<96 then a=a-64: goto 620
580 if a>31 and a<64 then 620
590 mu=-1: for i=1 to 7: if a=asc(mid$(a$,i,1)) then mu=i: i=7
600 next : if mu=-1 then 490
610 a=ke(mu)
620 poke k+x+40*y,a: x=x+1: if x=16 then x=0: y=y+1: if y=8 then y=0
630 goto 490
640 rem +- kep vissza toltese +-
650 ss$="": for y1=0 to 7: for x1=0 to 15
660 ss$=ss$+chr$(peek(k+x1+40*y1)+128): next x1,y1
670 if fu=1 then me$(la)=ss$
680 if fu=2 then te$(la)=ss$
690 if fu=3 then em$(la)=ss$
700 return
710 gosub 650
720 la=la+1: if la>sz(fu) then la=0
730 goto 380
740 gosub 650
750 la=la-1: if la=-1 then la=sz(fu)

```

## Idő Beállítás

A program kéri az időt ÓÓPPMM alakban. Egy vonal mutatja, hogy melyik számot írjuk most át. A vonalat a kurzor jobbra, kurzor balra-val lehet mozgatni. A számokat a számbillentyűkkel kell beírni. Ha készen vagyunk, nyomjunk RETURN-t.

F2 RBE

## Riasztás Beállítás

Különösebb magyarázatot nem igényel, mert ugyanúgy kell beállítani a Riasztást, mint az időt. Ha új időt vagy riasztási időt adunk meg, a riasztás újból aktív.

F3 VÉGE

Vissza a főmenübe.

## F5 KAL

## Számológép

Elsőnek vigyük be az első számot. Ez hasonlóan történik, mint az idő beállítása, csak itt 8 számjegyes szám van, és használhatjuk a tizedes pontot is. Ha kész vagyunk, nyomjunk RETURN-t. Ekkor már a műveletek között válogathatunk a kurzor jobbra, kurzor balra-val. Ha a megfelelő műveleten vagyunk, nyomjunk RETURN-t.

A + - / x választása után leolvassuk a második számot és megkapjuk az eredményt. „A” után egyből az eredményt kapjuk meg. A „C” a bevitt előlről kezd. Az OF-ot választva visszajutunk a főmenübe.

## F6 INI

## Inicializálás

Törli a TI\$-t, az X-et, az Y-t, a FU\$-t, a memória-, emlékeztető és telefonlapokat. A riasztást 99:99:99-re állítja.

## F7 VÉGE

A program végrehajt egy RESET-et.

Üdvözlettel:

Zsila Zsolt

```

○ 760 goto 380
770 gosub 650: goto 150
780 gosub 650
○ 790 open 1,8,2, fu$+",s,w"
800 print#1,sz(fu)
○ 810 for i=0 to sz(fu)
820 on fu gosub 840,850,860
830 for j=0 to 7: s1$=mid$(ss$,j*16+1,16): print#1,s1$: next j,i: close 1: goto 370
○ 840 ss$=me$(i): return
850 ss$=te$(i): return
860 ss$=em$(i): return
○ 870 gosub 650
880 open 1,8,2, fu$+",s,r"
890 input#1,sz(fu)
○ 900 for i=0 to sz(fu): ss$="": for j=0 to 7
910 input#1,s1$: ss$=ss$+s1$: next j
920 on fu gosub 940,950,960
○ 930 next i: close 1: goto 370
940 me$(i)=ss$: return
○ 950 te$(i)=ss$: return
960 em$(i)=ss$: return
970 open 15,8,15,"s:"+fu$
○ 980 print#15,"v": close 15
990 goto 490
1000 gosub 650
○ 1010 gosub 2470
1020 get a$: if a$="" then 1020
1030 if a$="+" then la=la+1: if la>sz(fu) then la=0
○ 1040 if a$="-" then la=la-1: if la<0 then la=sz(fu)
1050 if a$=chr$(13) or a$=chr$(32) then 370
○ 1060 goto 1010
1070 rem +- i d o . +-
1080 a$="[Clr]": gosub 2710: i=2: gosub 2300: fu$="id[s@].": gosub 2470
○ 1090 gosub 2800
1100 if (val(ti$)=>val(eb$)) and c3=0 then gosub 2360
1110 get a$: if a$="" then 1090
○ 1120 a=asc(a$): if a<133 then 1110
1130 on a-132 goto 1150,150,150,150,1170,150,150,150
○ 1140 goto 1090
1150 aa$=ti$: gosub 1190
1160 ti$=aa$: goto 1080
○ 1170 aa$=eb$: gosub 1190
1180 eb$=aa$: goto 1080
1190 a$="[Clr]": gosub 2710
○ 1200 x1=2: y1=0: u$="kErem az id[s@]t": gosub 2590
1210 x1=1: y1=2: u$="OOppmm alakban": gosub 2590
1220 x1=5: y1=6: u$=aa$: gosub 2590: x=0
○ 1230 poke k+285+x,99
1240 get a$: if a$="" then 1240
○ 1250 poke k+285+x,32
1260 if a$="[Rgt]" then x=x+1: if x=6 then x=5
1270 if a$="[Lft]" then x=x-1: if x<0 then x=0
○ 1280 if a$=chr$(20) or a$=chr$(13) then 1400
1290 a=asc(a$): if a<48 or a>57 then 1230
1300 a=a-48: on x+1 goto 1330,1360,1380,1310,1380,1310
○ 1310 x1=5+x: y1=6: u$=a$: gosub 2590: a$="[Rgt]": goto 1260
1320 goto 1230
1330 if a>2 then 1230
○ 1340 if peek(k+246)>51 and a=2 then 1230
1350 goto 1310
1360 if peek(k+245)=50 and a>3 then 1230
○ 1370 goto 1310
1380 if a>5 then 1230
1390 goto 1310
○ 1400 aa$="": for i=0 to 5: aa$=aa$+chr$(peek(k+245+i)): next i: c3=0: a$="[Clr]": goto 2710
1410 rem +- k a l . +-
○ 1420 a$="[Clr]": gosub 2710: x1=1: y1=1: u$="+ - / * _ [=T]c of": gosub 2590
1430 i=3: gosub 2300: fu$="kal.": gosub 2470
1440 gosub 2470
○ 1450 po=0: jj=0: gosub 1680: a1=val(aa$)
1460 if (val(ti$)=>val(eb$)) and c3=0 then gosub 2360
○ 1470 gosub 1570: if mo=6 then 150
1480 if mo=5 then 1420
1490 if mo=4 then gosub 1890: goto 1540
○ 1500 if (val(ti$)=>val(eb$)) and c3=0 then gosub 2360
1510 po=0: jj=1: gosub 1680: a2=val(aa$)
1520 if (val(ti$)=>val(eb$)) and c3=0 then gosub 2360

```

```

1530 on mo+1 gosub 1810,1830,1850,1870
1540 x1=7-int(len(str$(er))/2): y1=7: u$=str$(er): gosub 2590
1550 get a$: if a$<>chr$(32) and a$<>chr$(13) then 1550
1560 goto 1420
1570 x=0
1580 poke k+2*x+41,peek(k+2*x+41) or 128
1590 poke k+2*x+42,peek(k+2*x+42) or 128
1600 get a$: if a$="" then 1600
1610 if a$=chr$(13) then mo=x: return
1620 if a$="[Rgt]" then gosub 1650: x=x+1: if x=7 then x=0
1630 if a$="[Lft]" then gosub 1650: x=x-1: if x<0 then x=6
1640 goto 1580
1650 poke k+2*x+41,peek(k+2*x+41) and 127
1660 poke k+2*x+42,peek(k+2*x+42) and 127
1670 return
1680 x1=4: y1=3+jj*2: u$="[8:0]": gosub 2590: x=0
1690 poke k+164+jj*80+x,99
1700 get a$: if a$="" then 1700
1710 poke k+164+jj*80+x,32
1720 if a$="[Rgt]" then x=x+1: if x=8 then x=0
1730 if a$="[Lft]" then x=x-1: if x<0 then x=7
1740 if a$=chr$(20) or a$=chr$(13) then 1800
1750 a=asc(a$): if a<46 or a>57 and a<>46 then 1690
1760 a=a-48: if a=-2 then a=46: if po=1 then 1690
1770 if a=46 then po=1
1780 x1=4+x: y1=3+jj*2: if peek(k+x1+40*y1)=46 and a<>46 then po=0
1790 u$=a$: gosub 2590: a$="[Rgt]": goto 1720
1800 aa$="": for i=0 to 7: aa$=aa$+chr$(peek(k+124+jj*80+i)): next : return
1810 er=a1+a2
1820 return
1830 er=a1-a2
1840 return
1850 er=a1/a2
1860 return
1870 er=a1*a2
1880 return
1890 er=sqr(a1)
1900 return
1910 end
1920 rem +- rutinok +-
1930 rem +- karakter +-
1940 print "[Clr][Blk][12:Dwn]"tab(14)"ke[Up][Lft].[Dwn]rem va[Up][Lft].[Dwn]rjon!"
: poke 56334,0: poke 1,51
1950 for i=0 to 1023: a=peek(53248+i): poke 53248+i,a: poke 54272+i,255-a: next
1960 for j=0 to 9: read b: b1=53248+b*8: b2=54272+b*8
1970 for i=0 to 7: read a: poke b1+i,a: poke b2+i,255-a: next i: next j
1980 poke 1,55: poke 56334,1: poke 53272,21: poke 657,128: return
1990 rem +- kezdokep +-
2000 poke 53280,11: poke 53281,11: poke 650,128: poke 56576,196: poke 648,196
2010 print "[Clr][Wht][38:=@]"
2020 print " [Blk][=G][Gry2][7: ]kalkulAtor szimulAtor[8: ][Wht][=M]"
2030 print " [Blk][=G][Gry2][5: ]Irta : zsila zsolt 1993.[6: ][Wht][=M]"
2040 print " [Blk][38:=T]"
2050 print tab(11)"[Wht][18:=@]"
2060 for i=0 to 7: print tab(11)"[Blk][=G]";spc(16)"[Wht][=M]": next
2070 print tab(11)"[Blk][18:=T]"
2080 print "[Hom][3:Dwn]": for i=0 to 3
2090 print " [Wht][9:=@]";tab(30)"[9:=@]"
2100 print " [Blk][=H][7: ][Wht][=N]";tab(30)"[Blk][=G][7: ][Wht][=N]"
2110 print " [Blk][9:=T]";tab(30)"[9:=T]": next : print
2120 print " [Wht][38:=@]"
2130 print " [Blk][=G][Gry2] *** x: 000 y: 000 fu: [4:..] *** [Wht][=M]"
2140 print " [Blk][=G][Gry2] [4:*] memOria: 000 lap: 000 [4:*] [Wht][=M]"
2150 print " [Blk][38:=T]"
2160 return
2170 rem +- valtozok +-
2180 dim me$(40),te$(40),em$(40),f$(3,7)
2190 for j=0 to 3: for i=0 to 7: read f$(j,i): next i,j
2200 ti$="[6:0]": eb$="[6:9]": x=0: y=0: fu$="[4:..]"
2210 a$="[16:sSpc]": rem 16 db. shift+space !!!!
2220 as$="AEIOU[s@][s\]": ke(1)=65: ke(2)=69: ke(3)=73: ke(4)=79: ke(5)=85: ke(6)=122
2230 ke(7)=105
2240 for i=0 to 40
2250 me$(i)=a$+a$+a$+a$+a$+a$+a$+a$
2260 te$(i)=me$(i): em$(i)=me$(i)
2270 next
2280 sz(1)=40: sz(2)=40: sz(3)=40: k=50388: sz=55508: return

```



```

○ 2290 rem +- funk. kiiras +-
○ 2300 print "[Hom][4:Dwn][LBlu]"
○ 2310 for j=0 to 3
○ 2320 print "[2:Rgt][RvOn]";j+1;"[3:Lft]f[2:Rgt][RvOf]";f$(i,j);
○ 2330 print tab(29)"[2:Rgt][RvOn]";j+5;"[3:Lft]f[2:Rgt][RvOf]";f$(i,j+4)
○ 2340 print : print : next : return
○ 2350 rem +- riasztas +-
○ 2360 gosub 2390: gosub 2390
○ 2370 get a$: if a$=chr$(32) then c3=1: return
○ 2380 for i=0 to 500: next : goto 2360
○ 2390 for i=0 to 24: poke 54272+i,0: next
○ 2400 poke 54273,130
○ 2410 poke 54277,9
○ 2420 poke 54287,30
○ 2430 poke 54296,15: poke 54276,17
○ 2440 for i=0 to 100: next : poke 54276,16
○ 2450 return
○ 2460 rem +- lap kitoltes +-
○ 2470 print "[Hom][17:Dwn]"
○ 2480 a=x: gosub 2500: print tab(12)a$;: a=y: gosub 2500: print spc(4)a$;spc(5)fu$
○ 2490 a=sz(fu): gosub 2500: print tab(18)a$;: a=la: gosub 2500: print spc(7)a$: return
○ 2500 a$=right$("000"+right$(str$(a),len(str$(a))-1),3): return
○ 2510 rem +- egy lap irasa +-
○ 2520 if fu=1 then ss$=me$(la)
○ 2530 if fu=2 then ss$=te$(la)
○ 2540 if fu=3 then ss$=em$(la)
○ 2550 for y1=0 to 7: for x1=0 to 15
○ 2560 a=asc(mid$(ss$,y1*16+x1+1,1))-128: poke k+x1+40*y1,a: next x1,y1
○ 2570 return
○ 2580 rem +- uzenet +-
○ 2590 print "[Hom]";: for i=1 to y1+5: print : next : print tab(x1+12)u$: return
○ 2600 rem +- kurzor +-
○ 2610 c1=c1-1: if c1>0 then 2650
○ 2620 if c2=0 then poke k+x+40*y,aa
○ 2630 if c2=1 then poke k+x+40*y,100
○ 2640 c1=5: c2=1-c2
○ 2650 return
○ 2660 rem +- mozgás +-
○ 2670 if a$="[Rgt]" then x=x+1: if x=16 then x=0: y=y+1: if y=8 then y=0
○ 2680 if a$="[Lft]" then x=x-1: if x=-1 then x=15: y=y-1: if y=-1 then y=7
○ 2690 if a$="[Up]" then y=y-1: if y=-1 then y=7
○ 2700 if a$="[Dwn]" then y=y+1: if y=8 then y=0
○ 2710 if a$="[Clr]" then for x=0 to 15: for y=0 to 7: poke k+x+40*y,32: next y,x: a$="[Hom]"
○ 2720 if a$="[Hom]" then x=0: y=0
○ 2730 if a=13 then x=0: y=y+1: if y=8 then y=0
○ 2740 if a=20 then 2760
○ 2750 return
○ 2760 x=x-1: if x=-1 then x=15: y=y-1
○ 2770 if y=-1 then y=7
○ 2780 poke k+x+40*y,32: return
○ 2790 rem +- ti$ & eb$ tordelese +-
○ 2800 x1=4: y1=1: u$=left$(ti$,2)+": "+mid$(ti$,3,2)+": "+right$(ti$,2): gosub 2590
○ 2810 x1=4: y1=3: u$=left$(eb$,2)+": "+mid$(eb$,3,2)+": "+right$(eb$,2): gosub 2590
○ 2820 return
○ 2830 rem +- adatok +-
○ 2840 rem +- karakter +-
○ 2850 data 65,12,24,60,102,126,102,102,0
○ 2860 data 69,12,24,126,96,120,96,126,0
○ 2870 data 73,12,24,60,24,24,24,60,0
○ 2880 data 79,12,24,60,102,102,102,60,0
○ 2890 data 85,12,24,102,102,102,102,60,0
○ 2900 data 0,102,0,60,102,102,102,60,0
○ 2910 data 122,51,102,60,102,102,102,60,0
○ 2920 data 28,102,0,102,102,102,102,60,0
○ 2930 data 105,51,102,0,102,102,102,60,0
○ 2940 data 31,3,3,102,102,60,60,24,0
○ 2950 rem +- funkcio +-
○ 2960 data "mem.", "tel.", "eml.", "id[s@].", "kal.", "ini.", "vEge", "[4:]"
○ 2970 data "el[s@].", "hAt.", "t@l.", "men.", "ukm.", "fft.", "vEge", "[4:]"
○ 2980 data "ibe.", "rbe.", "vEge", "[4:]", "[4:]", "[4:]", "[4:]", "[4:]"
○ 2990 data "[4:-]", "[4:-]", "[4:-]", "[4:-]", "[4:-]", "[4:-]", "[4:-]"
○ 3000 rem +- v e g e +-
○ 3001 rem ----- checksum -----
○ 3002 poke 65,peek(122): poke 66,peek(123)
○ 3003 v=peek(43)+peek(44)*256: read c$: p=0
○ 3004 n=peek(v+2)+peek(v+3)*256: if n=3001 then end
○ 3005 p=p+1: if p>len(c$) then read c$: p=1

```

```

3006 v=v+4: s=0: print n"[Up]"
3007 b=peek(v): v=v+1: if b then s=s+b: goto 3007
3008 if (s and 31)<>asc(mid$(c$,p))-64 then print "error in"n
3009 goto 3004
3010 data "\e]evsoik\vc_ef@w]e^ja@xbqj@mq_gwexlaxzvmkft@ftmh[_vbe\^cmv["
3011 data "vufyr]@hbnhydhgdahmnhrt[thhzeakst[tadfhzpqvckcjw]widskeuo_]v"
3012 data "rkt^rikoa]o\gnfn_nn_wcza]@\@]c]mwwpzuwtsabwsunhyulopoubqrf_xn"
3013 data "yn[nznfn@enf^vdvjemjc_x^q@xb@zda_nsc[xx]@tpebm^nda_pjggov\ep@"
3014 data "ineqhpjk@hbxndtdfdn^mnjaysgyofvz[pr@nmn_ndq[kfrquhk_]w@"

```

ready.

#### Megjegyzés:

A lista begépelése előtt olvassuk el az 1994/1. számban megjelent BASIC LISTÁZÓ című cikket. Az ott leírtakhoz annyit kell hozzátenni, hogy a használt nyomtatón megjelenő

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

AZ 5-RE VEGZODO SORSZAMU SOROKAT NEM KELL BEIRNI!

```

10 REM *****
20 REM * TOTO-LOTTO *
30 REM * KESZITETTE: *
40 REM * LOOS JOZSEF *
50 REM * COMMODORE PLUS/4 GEPRE. *
60 REM *****
70 DIM X(16):DIM K(5100):SOUND 1,999,2
80 PRINT "*****":VOL 7
85 REM [CLR] 3X[LE]
90 PRINT "*****"
95 REM [CTR/9] 6X[JOB] 25X[SPC] [CTR/0]
100 PRINT "*****"
105 REM [CTR/9] 6X[JOB] [CTR/0]
110 PRINT "*****"
115 REM [CTR/9] 6X[JOB] [CTR/0]
120 PRINT "*****"
125 REM [CTR/9] 6X[JOB] [CTR/0]
130 PRINT "*****"
135 REM [CTR/9] 6X[JOB] 25X[SPC] [CTR/0]
140 PRINT "*****"
145 REM [CTR/9] 6X[JOB] [CTR/0]
150 PRINT "*****"
155 REM [CTR/9] 6X[JOB] [CTR/0]
160 PRINT "*****"
165 REM [CTR/9] 6X[JOB] [CTR/0]
170 PRINT "SOOLSOFT"
175 REM 4X[LE] 13X[SPC]
180 PRINT "*****"
185 REM [LE] 16X[SPC]
190 FOR ZZ=1 TO 2000:NEXT:SOUND 2,89,3
200 PRINT "*****MIRE AKAR TIPPELNI ?"
205 REM [CLR] 2X[LE] 10X[JOB]
210 PRINT "*****"
215 REM 9X[JOB] 22X[=]
220 PRINT "*****TOTO (1)"
225 REM 3X[LE] 2X[JOB] 12X[SPC]
230 PRINT "*****6-OS LOTTO (2)"
235 REM [LE] 2X[JOB] 6X[SPC]
240 PRINT "*****LOTTO (3)"
245 REM [LE] 2X[JOB] 11X[SPC]
250 GETKEY A$:SOUND 1,890,2
260 IF A$="1" THEN GOTO 710
270 IF A$="2" THEN GOTO 300
280 IF A$="3" THEN GOTO 1300
290 IF A$<>"1" OR A$<>"2" OR A$<>"3" THEN 200
300 REM *****6-OS LOTTO*****

```

## TOTÓ-LOTTO

- A program kezelése:*  
Menüpontok kiválasztása.
- A program célja:*  
Totó-Lottó tippek adása.
- A program műfaja:* Szerencsejáték.
- A program leírása:*  
A program képes arra, hogy totó-, 6-os lottó- és lottótippeket adjon. A totószelvények száma maximum 315 db, a 6-os lottóé 850 db, a lottóé 1000 db lehet. A számítógép memóriájában maximum ennyi fér bele. A program indítása, majd a menüpontok közül való választás után, a szelvények számát kell megadni. A gép ezután a számokat generálja véletlenszerűen. A számokat vizsgálja, sorrendbe teszi és kiírja a képernyőre. Lehetőség van a szelvényszámok kinyomtatására is. Új tippkéréskor, a menüre tér vissza a program. A tippekhez sok szerencsét kívánhatunk.
- Programozási feladatok megoldása:*
  - tippszámok generálása
  - tippszámok vizsgálata (két egyforma nem lehet)
  - számok nagyságszerinti sorbatétele
  - totótippeknél háromfajta tippre való alakítás
  - tippszelvények nyomtatása, több példányban is.
- Megjegyzés:*  
Ha a tippekkel nem nyer valaki, akkor nem a programozó, illetve nem a számítógép a hibás.

Loós József

```

310 PRINT "-----6-OS LOTTO TIPP":PRINT "-----"
315 REM [CLR] 2X[LE] 12X[JOB] 11X[JOB] 17X[=] 2X[LE]
320 PRINT "HANY SZELVENNYEL AKAR JATSZANI ?":INPUT "(MAX.850)";Q:SOUND 1,900,2
325 REM [LE] [JOB] 3X[JOB]
330 IF Q>850 THEN GOTO 320
340 PRINT "":F=6:YI=0:D=1
345 REM 2X[LE]
350 FOR Y=1 TO Q:PRINT "Y";" ";*";
355 REM 4X[JOB] [BAL]
360 FOR I=1 TO 6:SOUND 1,560,1
370 X(I)=INT(RND(X)*45)+1
380 GOTO 520
390 NEXT I
400 GOSUB 610
410 FOR I=1 TO 6
420 PRINT X(I);
430 NEXT I
440 FOR I=1 TO 6:YI=YI+1
450 K(YI)=X(I)
460 NEXT I
470 PRINT TAB(34);"*"
480 PRINT "-----":SOUND 1,980,1
485 REM 7X[JOB] 28X[-]
490 NEXT Y
500 F=6:GOSUB 950
510 GOTO 1520
520 REM *****SZAM VIZSGALATA*****
530 B=I:X(B)=X(I)
540 FOR R=1 TO F+1:SOUND 1,990,1
550 IFR=B THEN GOTO 590
560 IF X(R)=X(B) AND F=6 THEN I=I-1:GOTO 370
570 IF X(R)=X(B) AND F=5 THEN I=I-1:GOTO 1370
580 NEXT R
590 IF D=3 THEN 1390
600 GOTO 390
610 REM *****SORREND*****
620 FOR W=1 TO F-1:SOUND 2,230,1
630 FOR L=W+1 TO F
640 IF X(L)>X(W) THEN GOTO 680
650 S=X(L)
660 X(L)=X(W)
670 X(W)=S
680 NEXT L
690 NEXT W
700 RETURN
710 REM ***TOTO***
720 PRINT "-----TOTO TIPP":PRINT "-----"
725 REM [CLR] 2X[LE] 14X[JOB] 13X[JOB] 11X[=] 2X[LE]
730 PRINT "HANY SORRAL AKAR JATSZANI (MAX.315)?":INPUT " ";Q:SOUND 1,890,2
735 REM [LE] [JOB] 3X[JOB]
740 IF Q>315 THEN GOTO 730
750 PRINT "":YI=0:T1=3:T2=30:T3=1
755 REM 2X[LE]
760 FOR Y=1 TO Q:PRINT Y;" ";I";
765 REM [BAL] [SHF/B]
770 FOR I=1 TO 16:SOUND 1,600,1
780 IF I=14 THEN PRINT "I";
785 REM [SHF/B]
790 X(I)=INT(RND(X)*3)+1
800 IF X(I)=3 THEN PRINT "X ";:GOTO 830
810 IF X(I)=1 THEN PRINT "1 ";:GOTO 830
820 IF X(I)=2 THEN PRINT "2 ";
830 NEXT I
840 FOR I=1 TO 16:YI=YI+1
850 K(YI)=X(I)
860 NEXT I
870 IF Y<99 AND Y>9 AND T3=1 THEN T1=T1+1:T2=T2+1:T3=2:GOTO 890
880 IF Y>99 AND T3=2 THEN T1=T1+1:T2=T2+1:T3=3
890 PRINT:PRINT TAB(T1);" ";TAB(T2);" "
900 PRINT "-----":SOUND 2,990,1
905 REM 3X[JOB] 35X[-]
910 NEXT Y
920 F=4:GOSUB 950
930 GOTO 1520
940 REM *****NYOMTATAS !!*****
950 OPEN 4,4
960 INPUT "KER NYOMTATAST (I/N)";P$:SOUND 1,780,2

```

```

965 REM 4X[JOBBS]
970 IF P$="I" THEN GOTO 1000
980 IF P$="N" THEN CLOSE 4:RETURN
990 IF P$<>"I" OR P$<>"N" THEN GOTO 960
1000 IF F=6 THEN GOTO 1030
1010 IF F=5 THEN GOTO 1060
1020 IF F=4 THEN GOTO 1090
1030 PRINT#4,"*=====*"
1035 REM 32X[=]
1040 PRINT#4,"***** 6-OS LOTTO TIPPEK *****"
1045 REM 7X[*] 8X[*]
1050 PRINT#4,"*=====*" :GOTO 1120
1055 REM 32X[=]
1060 PRINT#4,"*=====*"
1065 REM 32X[=]
1070 PRINT#4,"***** LOTTO TIPPEK *****"
1075 REM 7X[*] 6X[SPC] 8X[*]
1080 PRINT#4,"*=====*" :GOTO 1120
1085 REM 32X[=]
1090 PRINT#4,"*=====*"
1095 REM 32X[=]
1100 PRINT#4,"***** TOTO TIPPEK *****"
1105 REM 7X[*] 6X[SPC] 8X[*]
1110 PRINT#4,"*=====*"
1115 REM 32X[=]
1120 Y=0:YI=0
1130 Y=Y+1:IF Y=Q+1 THEN GOTO 1250
1140 PRINT#4,"*Y.";"* ";
1150 IF F=4 THEN GOTO 1190
1160 FOR I=1 TO F:YI=YI+1
1170 PRINT#4, K(YI);
1180 GOTO 1240
1190 FOR I=1 TO 16:YI=YI+1
1200 IF I=14 THEN PRINT#4," I";
1205 REM [SPC] [SLEFT$/B]
1210 IF K(YI)=3 THEN PRINT#4,"X ";;GOTO 1240
1220 IF K(YI)=1 THEN PRINT#4,"1 ";;GOTO 1240
1230 IF K(YI)=2 THEN PRINT#4,"2 ";;
1240 NEXT:PRINT#4:GOTO 1130
1250 PRINT#4,"*=====*"
1255 REM 32X[=]
1260 PRINT#4,"**** SOK SZERENCSET KIVANOK ! ****"
1265 REM 4X[*] 4X[*]
1270 PRINT#4,"*=====*"
1275 REM 32X[=]
1280 PRINT#4
1290 SOUND 2,780,2:GOTO 960
1300 REM ****LOTTO*****
1310 PRINT"*****LOTTO TIPP":PRINT"*****"
1315 REM [CLR] 2X[LE] 12X[JOBBS] 9X[JOBBS] 15X[=] 2X[LE]
1320 PRINT"*****HANY SZELVENNYEL AKAR JATSZANI ?":INPUT"*****"(MAX.1000)";Q:SOUND 1,890,2
1325 REM [LE] [JOBBS] 3X[JOBBS]
1330 IF Q>1000 THEN 1320
1340 PRINT"*****":F=5:YI=0
1345 REM 2X[LE]
1350 FOR Y=1 TO Q:PRINT"*****Y.";"* ";
1355 REM 4X[JOBBS] [BAL]
1360 FOR I=1 TO 5:SOUND 1,560,1
1370 X(I)=INT(RND(X)*90)+1
1380 D=3:GOTO 520
1390 NEXT I
1400 GOSUB 610
1410 FOR I=1 TO 5
1420 PRINT X(I);
1430 NEXT I
1440 FOR I=1 TO 5:YI=YI+1
1450 K(YI)=X(I)
1460 NEXT I
1470 PRINT TAB(33);"*"
1480 PRINT"*****":SOUND 1,990,1
1485 REM 7X[JOBBS] 27X[-]
1490 NEXT Y
1500 F=5:GOSUB 950
1510 REM *****END*****
1520 INPUT"*****KER UJ TIPPET (I/N)";B$:SOUND 2,780,2
1525 REM [LE] 4X[JOBBS]
1530 IF B$="I" THEN D=1:GOTO 200

```

```

1540 IF B$="N" THEN GOTO 1560
1550 IF B$<>"I" OR B$<>"N" THEN GOTO 1520
1560 PRINT" *=====*"
1565 REM 2X[SPC] 32X[=]
1570 PRINT" **** SOK SZERENCSET KIVANOK ! ****"
1575 REM 2X[SPC] 4X[*] 4X[*]
1580 PRINT" *=====*" :SOUND 1,890,3:END
1585 REM 2X[SPC] 32X[=]

```

MIRE AKAR TIPPELNI ?  
=====

TOTO (1)  
6-OS LOTTO (2)  
LOTTO (3)

# Rattle-Copy II. III. IV. V.

A program ShareWare. A készítője: Peter van Campen. Hollandia.

A Rattle Copy egy kicsit szegényesnek és furcsának tűnhet a közismert X-Copy és a D-Copy után, de rengeteg olyan tulajdonsága van, amely miatt megéri áttérni erre a másolóra.

Nézzük, mit nyújt ez a másoló:

Még az elején meg kell állapítanom, hogy a Rattle-ban a cél-, (SOURCE), illetve a forrás- (TARGETS) lemezek kijelölése automatikusan történik. A behelyezett lemezek írásvédettsége alapján dönti el a copy, hogy melyik lemezről van szó. Ennek előnyei és hátrányai egyaránt vannak természetesen. Előnyét talán nem is kell részleteznem. Hátrány az, hogy a nagy drive-t (5,25 collos) használóknál bajok lesznek a write protect kapcsolóval — ha van — (folyton kapcsolgatni kell). További hátránya (vagy előnye?) a másolónak, hogy ha a Verify (ellenőrzés) be van kapcsolva, akkor az összes céllemezt így írja meg. Véleményem szerint ez még mindig jobb, mint az, hogy egy bizonytalan és néha hibás Verify szolgáltatást élvezünk.

A másolót elindítva egy nem túlcicomázott képernyőt kapunk a Rattle II-es esetén. A Rattle III. érdekessége, hogy aki únja a másolást, az a programba beépített Pacman játékkal töltheti el a másolás unalmas perceit (óráit). A Rattle IV-es verziója hasonló a II-es verzióhoz, de a felhasználói felület már sokkal „Amigásabb” lett. Márciusban jelent meg a Rattle V-ös verziója. Ami véleményem szerint a legszebbre sikeredett. A bal oldali felső sarokban van a copy aktuális utasítását jelképező ablak. Ez akkor, ha csak egy drive-t használunk, illetve ha csak a forráslemezt raktuk be READ feliratot mutat. Ha két lemezt is behelyeztünk, akkor COPY feliratra vált át. A RAM-ba történő másolásnál a beolvasás után vált át WRITE feliratra.

MODE: — Másolási módok —

Rattle: Csak a használt trackokat (sávokat) másolja át, a többi csak formázza a céllemezen. Ha olyan lemezt másolunk amely már ezzel lett másolva (RATTLE mód), akkor egy speciális lementett tábla alapján a copy gyorsabban végzi el feladatát. (Ezt a speciális Bitmap-ot a copy a 0-ás trackra menti el, az eredeti adatok sérülése nélkül!)

Bitmap: A DOS foglaltsági táblája alapján másol. Itt jó lesz vigyázni, mert csak standard DOS lemezek esetén lehet használni. A foglalt területeket átmásolja, az üreseket formázza a céllemezen.

Range: A legegyszerűbb másolási mód, a kijelölt területet egy az egyben átviszi a céllemeze.

Format: Nem nagyon kell megmagyaráznom. A behelyezett írható lemezt megformattálja. Ha nem adjuk meg a lemez nevét, akkor EMPTY (üres) néven kerül megformálásra a lemez.

Nibble: Sajnálatos módon csak két drive esetén élvezhetjük ezt a szolgáltatását a Rattle-nak. Vigyázzunk, mint általában, itt sincs Verify! A Nibble móddal, a nehezen beolvasható (általában sérült, vagy védett sávokat másolhatunk át) úgy, hogy a céllemeze is az az adat kerül, mint ami a forráslemezen van.

VERIFY: A funkció be- illetve kikapcsolása.

FORMAT Off: Nem formattálja a kimaradt területeket (Rattle, Bitmap módban).

Fast: Nincs verify a formázáskor.

Verify: Itt már viszont van.

A képernyő jobb felső sarkában láthatjuk az időkijelzést, ha akarjuk átválthatjuk track kijelzésre.

NIBBLE a nibble mód beállítása.

Atari formátumú lemezek esetén: Sync 4489

Macintosh formátumú lemezek esetén: Sync 4489

MS-DOS formátumú lemezek esetén: Index

A beállítás módjától függően akár gyári védett AMIGA lemezek is másolhatók. — Biztonsági másolat —

SYNC: A Nibble módhoz kapcsolódik, itt lehet próbálkozni a beállítással.

SIDE: Atari, illetve MS-DOS lemezek másolásához szükséges.

Both — alsó-felső oldala a lemeznek

Up — csak a felső oldala

Down — csak az alsó oldala

NAME: Amikor a másoló forráslemezt érzékel, akkor ide írja ki a lemez nevét. Meg is lehet változtatni, igény szerint.

BOOT: A forráslemez bootblockját mutatja meg. Rögtön látni lehet ha a lemezen vírus van!!!! Változtatni is lehet rajta.

Imported: Standard bootblockot másol a vírus helyett a céllemeze.

Buffer: Egy előzőleg beolvasott bootblockot másol ki.

USER: A felhasználó neve.

START: Az induló track és oldal száma.

END: Az utolsó track és oldal száma.

POS: A kurzor pozíciója a TRACKMAP-on. (Lásd később!)

ALL: Standard beállítás kijelölése. (0,0—79,1.)

SOURCE: A fellelt forráslemez száma.

TARGETS: A fellelt cél lemez(ek) száma.

A középen található nagy tábla, a TRACKMAP. Itt lehet beállítani, hogy melyik területet akarjuk másolni. A beállítás úgy történik, hogy a sárga kis kockát a kezdő track-ra, illetve oldalra állítjuk, majd a bal sárga kis kockát a kezdő track-ra, illetve oldalra állítjuk az utolsó track-ot illetve oldalt jelképező kockáig és ott engedjük el. A beállítást a START illetve az END kiírásán láthatjuk folyamatosan.

EXAMINE BOTT: Egy új képernyőt nyit a másoló.

View: A kiválasztott bootblockot nézhetjük meg.

Kill: A veszélyes boot-ok kiirtása. (Lehetőleg csak vírusok esetén!)

Decode: Sok vírus kódolja a boot-ot, hogy ne lehessen ráismerni. Ez a menüpont ebben segít.

Text: Kódolt szöveg keresése.

Rattlecopy: Visszatér a másolóhoz.

INSTALL: A kiválasztott bootblockot felírja az összes céllemeze.

ABOUT: A készítő neve, címe és néhány üzenete a felhasználóknak.

RENAME: A cél lemezek nevét módosítja.

BITMAP: Csak olyanok használják, akik értenek is hozzá! A BLOCKMAP-on a bal egérgomb lefoglal a bal pedig felszabadít egy block-ot.

Drive: A kiválasztott drive száma. A jobb, illetve a bal egérgombbal lehet beállítani.

Show: Megmutatja az aktuális block-ot, amin a pointer áll.

Read: Beolvassa a foglaltsági térképet a lemezről.

Write: Újra írja a foglaltsági térképet.

All: Minden block kijelölése a lemezen.

Clear: Minden block foglaltságának törlése.

Undo: Az utoljára végzett műveletet visszaállítja.

Rattlecopy: Vissza a másolóhoz.

Kopácsi Lajosné Éva

# Quasi tömbváltozók. [dBASE III plus.]

A BASIC nyelvben (NEM-től!!) „megőszült” programíró sértődötten konstatálja, hogy a dBASE III+ nem ismeri sem a FOR/NEXT ciklust, sem a tömbváltozó fogalmát. (A többi „anomáliáról” most ne essék szó!)

Ha mégis felmerülne olyan feladat, amely legkönnyebben az említett hiátusokkal lenne megoldható, csak némi „furfang” segítségével boldogulhatunk.

Az RUT. DEMO listán látható a megoldás, amelyhez azonban némi magyarázat szükségeltetik. A program első soraiban hozzuk létre a „tömbváltozókat”, és különböző karakterláncokat rendelünk hozzájuk. (Az első értékadás BASIC-ben így nézne ki: VALT\$(1) = "11111").

Az így létrehozott változókból, a ciklusos kiírás során készítünk „tömbváltozókat”. A fellelhető egyetlen ciklusutasításnak, a DO WHILE-nek nincsen ciklusváltozója. Így azt magunknak kell létrehozni és lépésenként növelni. A létrehozás módja: p = 1.

A "SOR = 'VALT'+STR(P,1)" utasítással, a „tömb” P-edik „elemét szereljük össze”, majd a "@ 8,0 SAY "" sorral történő pozicionálás után a "@ ROW +2, V SAY &SOR" írja ki az aktuális sztringet. NE feledkezzünk meg a „&” karakterről. Ha nem írjuk ki, akkor a változó értéke helyett a nevét kapjuk vissza.

A bármilyen (ASCII) szövegszerkesztővel megírt rutint (kiterjesztése: PRG) a dBASE III PLUS-ból futtathatjuk. Némi szerencsével, még működni is fog!

Most jutottunk el oda, hogy akár tetszik, akár nem, „töreldelmes vallomást” kell tennem! Megszállott menü párti vagyok, ez készített a fenti furfangok alkalmazására.

A PROG.ram MENU című lista egy menürutint mutat be, amely a megszokott módon vezérelhető. (<Le/Fel> és <ENTER>). Az aktuális menüpont inverzben látszik, a választás <ENTER>-rel történik. A program a választott menüpont sorszámával (P) tér vissza.

Szász Sándor

```

***** RUT. DEMO *****
set talk off
  valt1='11111'
  valt2='22222'
  valt3='33333'
  valt4='44444'
p=1
v=38
  clear
    @ 8,0 say ""
    do while p<>5
      sor='valt'+str(p,1)
      @ row()+2,v say &SOR
      p=p+1
    loop
  enddo
set talk on
return
  
```

```

***** PROG. MENU *****
set talk off
  valt1=' Adatkeresés '
  valt2=' Adatnyomtatás '
  valt3=' Adatbevitel '
  valt4=' Adattörlés '
  valt5=' BEFEJEZÉS '
  pos=.f.
p=2
v=32
  clear
    @ 8,0 say ""
    do while p#6
      sor='valt'+str(p,1)
      @ row()+2,v say &SOR
      p=p+1
    loop
  enddo
@ 7,31 to 17,47 double
sor='valt'+str(1,1)
set color to n/w
@ 8,v say &SOR
store 1 to p,w
f=8
do while w#13
  w=inkey()
  do case
    case w=5 .and. p>1
      pos=.t.
      p=p-1
      f=row()-2
    case w=24
      pos=.t.
      p=p+1
      f=row()+2
    if p>5
      store 1 to p,w
      f=8
  endif
endcase
if pos
  set color to w/n
  @ row(),v say &SOR
  set color to n/w
  sor='valt'+str(p,1)
  @ f,v say &SOR
  pos=.f.
endif
loop
enddo
set talk on
set color to w/n
return
  
```

# ABC rendezések vizsgálata

## 1. A program kezelése:

Bármelyik billentyű lenyomásával kezelhető.

## 2. A program célja:

ABC rendezési fajták összehasonlítása..

## 3. A program műfaja: Felhasználói, demo program.

## 4. A program leírása:

A program négyfajta rendező algoritmus szerinti rendezést végez. Így a felcseréléses, a buborék, a Shell-Metzner és a „gyors” fajták szerinti rendezést hajtja végre.

A rendezések eredményeit a program egy olyan diagramban ábrázolja, aminek a vízszintes tengelyén a rendezett elemek számát, a függőleges tengelyén a rendezések alatt eltelt időt (secundumban) mutatja.

Az elemek száma maximum 60. A vizsgálat úgy történik, hogy rendezés fajtánként, kettő elemtől a maximumig az elemeket rendezi a program. Minden elemszám-rendezés alatt eltelt időt a gép mér, és a diagramon ezt jelzi. Az így kialakult diagramokból (görbéből) jól látható, hogy az elemszámok növekedésével az ehhez tartozó idők hogyan alakulnak.

Összehasonlítva a rendező algoritmusokat a diagramok alapján, mindenki levonhatja a következtetéseket.

## 5. Programozási feladatok megoldása:

- rendező algoritmusok kitalálása (a szerző érdeme)
- idő mérések
- diagramban való ábrázolás.

## 6. Megjegyzések:

- a program működése közben a diagram-pontok rajzolása alatt, részidőket is láthatunk a képernyőn. (Rendezett elemszám, és a rendezés alatt eltelt idő.)
- A Shell-Metzner rendező nagyon lassú a diagram értékelhetősége miatt, így ennél nem a maximum elemszámig történik a vizsgálat. Ebben az esetben a maximum idő a határ (22 secundum). Ez idő alatt rendezett maximum elem számot jelzi a gép.
- Az egész program működése alatt eltelt idő viszonylag hosszú. Perceket is igénybe vesz a végig futása. De ez természetes is, hiszen a rendező algoritmusok működése alatt is idő szükségeltetik.

A végeredmények remélem kárpótolják a türelmes felhasználót.

Loós József

```

10 REM *****
20 REM * ABC RENDEZESSEK VIZSGALATA *
30 REM * KESZITETTE: *
40 REM *      LOOS JOZSEF *
50 REM * COMMODORE PLUS/4 GEPRE. *
60 REM *****
70 REM -----
80 DIM A$(61):VOL 6
90 DIM ST$(LOG(61)/LOG(2)+4),1)
100 PRINT "C"
110 PRINT:PRINT:PRINT
120 PRINT "  **** RENDEZO PROGRAMOK ****"
130 PRINT "      MEGFIGYELESE"
140 PRINT "XOX FELCSERELESESE....."
150 PRINT "  BUBOREK....."
160 PRINT "  SHELL-METZNER....."
170 PRINT "  GYORS....."
180 PRINT "-----"
190 PRINT "  OSSZEHASONLITO"
200 PRINT "  VIZSGALATA....."
210 PRINT "  VEGE....."
220 PRINT "XOXXMEZERLES BARMELYIK BETUVEL !"
230 GETKEY P#
240 PRINT "XOXXKIS TURELMET KEREK !"
250 D=0
260 FOR I=60 TO 1 STEP -1
270 D=D+1
280 A$(D)=STR$(I)
290 NEXT I:N=2:X1=0
300 REM -----RAJZOLAS 1-----
310 GRAPHIC 1,1
320 DRAW 1,20,30 TO 20,170 TO 185,170
330 DRAW 1,18,35 TO 20,30 TO 22,35:DRAW 1,180,167 TO 185,170 TO 180,173
340 FOR Y=30 TO 120 STEP 30
350 DRAW 1,16,170-Y TO 21,170-Y
360 NEXT Y
370 FOR X=50 TO 150 STEP 50
380 DRAW 1,20+X,169 TO 20+X,174
390 NEXT X
400 CHAR 1,12,0,"REND. ELEM:"
410 CHAR 1,12,1,"===== "
420 CHAR 1,25,4,"FELCSER. REND."
430 CHAR 1,25,7,"BUBOREK REND."
440 CHAR 1,25,10,"SHELL-MT REND."
450 CHAR 1,25,13,"GYORS REND."
460 CHAR 1,0,1,"IDŐ":CHAR 1,0,2,"(SEC)"

```

```

470 CHAR 1,1,21,"0":CHAR 1,1,17,"6":CHAR 1,0,13,"12":CHAR 1,0,9,"18"
480 CHAR 1,0,6,"24"
490 CHAR 1,8,22,"25":CHAR 1,14,22,"50":CHAR 1,20,22,"75"
500 CHAR 1,25,22,"ELEM"
510 CHAR 1,25,23,"SZAMA"
520 REM --- FELCSERELES RENDEZES ---
530 TT=TI
540 FOR J=1 TO N-1
550 FOR K=J+1 TO N
560 IF A$(J)<A$(K) THEN GOTO 580
570 TEMP$=A$(J):A$(J)=A$(K):A$(K)=TEMP$
580 NEXT K
590 NEXT J
600 Q1=((TI-TT)/60)*5
610 GOSUB 1150:GOSUB 1170:GOSUB 1250
620 IF N=61 THEN CHAR 1,18,5,"F",1:CHAR 1,30,6,STR$(N-1),1:N=2:X1=0:GOTO 640
630 GOTO 520
640 REM --- BUBOREK RENDEZES -----
650 CHAR 1,24,0," "
660 TT=TI
670 FOR J=N-1 TO 1 STEP -1:FIN=-1
680 FOR K=1 TO J
690 IF A$(K)<A$(K+1) THEN GOTO 720
700 FIN=0:TE$=A$(K):A$(K)=A$(K+1)
710 A$(K+1)=TE$
720 NEXT K:IF NOT FIN THEN NEXT J
730 Q2=((TI-TT)/60)*5
740 GOSUB 1150:GOSUB 1190:GOSUB 1250
750 IF N=61 THEN CHAR 1,19,20,"B",1:CHAR 1,30,9,STR$(N-1),1:N=2:X1=0:GOTO 770
760 GOTO 660
770 REM --- SHELL-METZNER REDEZES ----
780 CHAR 1,24,0," "
790 TT=TI
800 M=N
810 M=INT(M/2)
820 IF M=0 THEN Q3=((TI-TT)/60)*5:GOTO 910
830 J=1:K=N-M
840 I=J
850 L=I+M
860 IF A$(I)<A$(L) THEN GOTO 880
870 TE$=A$(I):A$(I)=A$(L):A$(L)=TE$
880 I=I-M:IF I>0 THEN 850
890 J=J+1:IF J>K THEN GOTO 810
900 GOTO 840
910 GOSUB 1150:GOSUB 1210:GOSUB 1250
920 IF Q3/5>22 THEN CHAR 1,30,12,STR$(N),1:N=2:X1=0:CHAR 1,10,5,"S",1:GOTO 940
930 GOTO 790
940 REM --- GYORS-RENDEZES -----
950 CHAR 1,24,0," "
960 TT=TI
970 REM DIM ST((LOG(N)/LOG(2)+4),1)
980 S=1:ST(1,0)=1:ST(1,1)=N
990 L=ST(S,0):R=ST(S,1):S=S-1
1000 J=L:K=R:X$=A$((L+R)/2)
1010 IF A$(J)>X$ THEN J=J+1:GOTO 1010
1020 FOR V=1 TO 1E6:IF A$(K)>X$ THEN K=K-1:NEXT
1030 IF J=K THEN J=J+1:K=K-1:GOTO 1010
1040 IF J<K THEN:TEMP$=A$(J):A$(J)=A$(K):A$(K)=TEMP$:J=J+1:K=K-1:GOTO 1010
1050 IF J<R THEN S=S+1:ST(S,0)=J:ST(S,1)=R
1060 R=K
1070 IF L<R THEN 1000
1080 IF S>0 THEN 990
1090 Q4=((TI-TT)/60)*5
1100 GOSUB 1150:GOSUB 1230:GOSUB 1250
1110 IF N=61 THEN CHAR 1,19,15,"GY",1:CHAR 1,30,15,STR$(N-1),1:N=2:X1=0:
GOTO 1130
1120 GOTO 960
1130 SOUND 1,890,5:GETKEY P$:GRAPHIC 0:PRINT"JAJZOLAS VEJELT, KEGE":END
1140 REM -----RAJZOLAS 2-----
1150 X=N
1160 CHAR 1,24,0,STR$(N),1:RETURN
1170 DRAW 1,20+X1+X,170-Q1: SOUND 1,790,1
1180 CHAR 1,25,5,STR$(Q1/5):RETURN
1190 DRAW 1,20+X1+X,170-Q2: SOUND 1,790,1
1200 CHAR 1,25,8,STR$(Q2/5):RETURN
1210 DRAW 1,20+X1+X,170-Q3: SOUND 1,790,1
1220 CHAR 1,25,11,STR$(Q3/5):RETURN
1230 DRAW 1,20+X1+X,170-Q4: SOUND 1,790,1
1240 CHAR 1,25,14,STR$(Q4/5):RETURN
1250 N=N+1:X1=X1+1:RETURN

```

READY.



# PC bővítés

## EPROMÉGETŐ IBM XT/AT SZÁMÍTÓGÉPEKHEZ

Ma a piacon kapható és régről a fiók mélyén fellelhető sokféle EPROM olyan égetőt kíván, ami szinte mindenféle algoritmus szerint képes égetni és ezek az algoritmusok szükség esetén szoftver útján változtathatók. Emellett hatékony védelemmel rendelkezik, amely mind az EPROM-ot, mind pedig az égetőt a lehetőség szerint megvédi a károsodástól, valamint biztosítja, hogy az EPROM programozókészülékbe helyezése és kivétele feszültségmentes állapotban történjen. Az áramkör működésének ismertetése előtt röviden, a teljesség igénye nélkül nézzük meg néhány gyakran használt EPROM típus égetési algoritmusát, hogy képet kapjunk arról, hogy mit csinál az égető. A programozáskor először a tápfeszültséget kell rákapcsolni, amely típustól függően lehet 5 V, vagy 6 V, amit a gyártó programozáskor előír. Ezután kapcsoljuk rá a címet, adatot és az Upp-t, ami szintén típus és gyártó függő, értéke 12,5 V, 21 V vagy 25 V lehet. Végül rákapcsoljuk a programozó impulzust, ami az égetés elvégzését biztosítja. Ez egy meghatározott formájú és időzítésű impulzus, ami szintén gyártótól függő katalógus adat. A programozás ezzel kész, következik az összehasonlítás, amikor visszaellenőrizzük, hogy az eredeti és az égetett adat megegyezik-e. Ezzel lezárul egy bájt beégetése, aminek időtartama a programozási algoritmustól függően 0,1  $\mu$ S–50 mS lehet. Az ismertebb EPROM típusok lábkiosztása az 1. ábrán látható, ezek égetési algoritmusai néhány szóban a következők:

**2716:** Ennél a típusnál az égetőfeszültséget rákapcsoljuk és az OE bemenetet magasra húzzuk, majd az adat és címvonalat aktivizáljuk. Az égetés a CE 50 ms-ig történő magasra állítása alatt valósul meg.

**2732:** Ennél a típusnál az égetőfeszültség és az OE bemenet egyetlen lábba került. Az OE-re csak olvasáskor van szükség. Ennek megfelelően a 20. lábba az égetőfeszültséget kapcsoljuk, majd ezt követi az adatok és a címek beállítása. Az

égetés a CE 50 ms-ig tartó L-re húzásával történik.

**2764:** Négydel több lába van, mint az előző típusoknak. Az égetőfeszültség rákapcsolása, OE magasra állítása után az adat és címvonalakat aktivizáljuk. A CE-t L-re húzzuk, majd a PGM 50 ms-ig tartó alacsonyra állításával égetünk. Többféle égetőfeszültséggel gyártják.

**27128:** Égetése megegyezik a 2764-gyel. Többféle égetőfeszültséggel gyártják, leggyakoribb a 12,5 V-os változat.

**27256:** Az égetése alapvetően más, mint az előzőké. A tápfeszültséget az égetés alatt 6 V-ra kell növelni, majd az égetőfeszültséget is rákapcsoljuk. Ezután aktivizáljuk a cím- és adatvonalakat. Az égetés 1 ms-os lépésekben történik a CE L-re húzásával. Az égetésnél minden címen ellenőrizzük a beírt bájt. Sikeres visszaolvasás után még háromszor annyi ideig ráégetünk, mint amennyi az első sikeres égetéshez kellett.

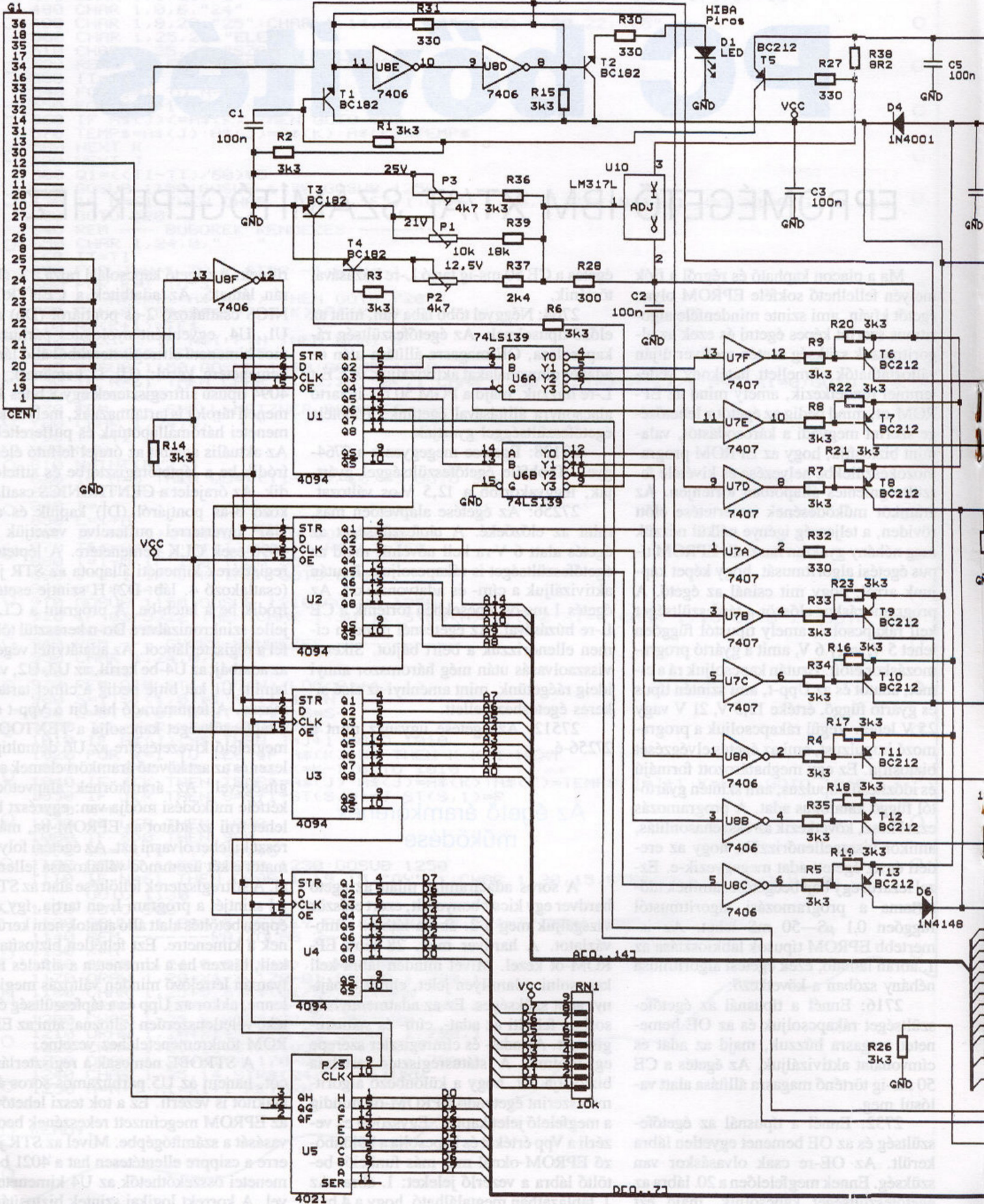
**27512:** Az égetése ugyanaz mint a 27256-é.

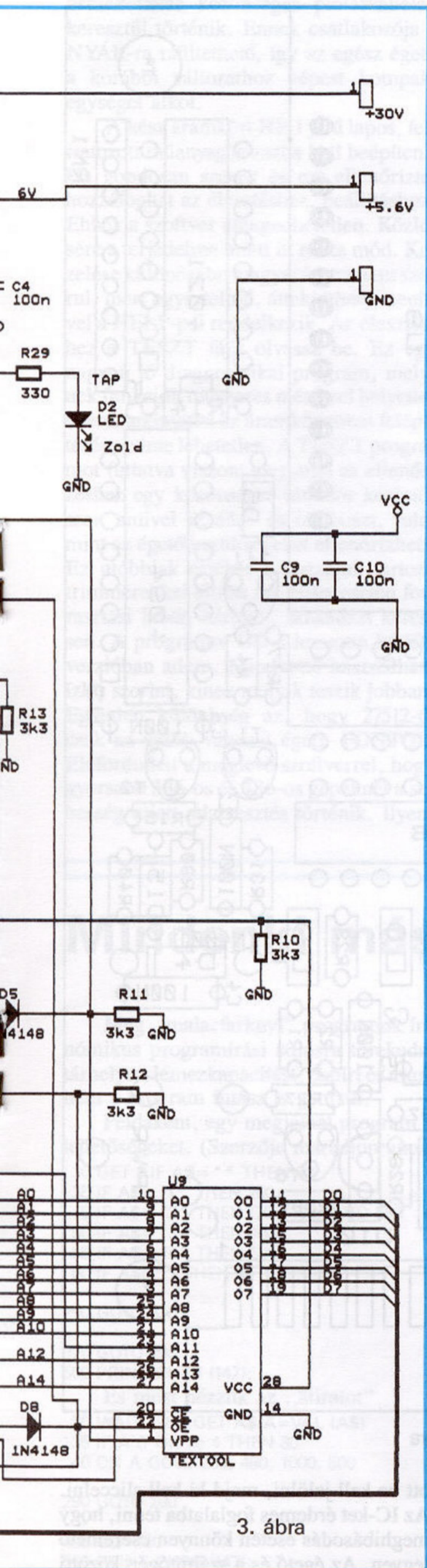
### Az égető áramkörének működése

A soros adatáramlás miatt az égető hardver egy kicsit bonyolult, ezért először vizsgáljuk meg a 2. ábrán látható tömbvázlatot. A hardver max. 28 lábú EPROM-ot kezel. Mivel minden lábba kell kapcsolni valamilyen jelet, ehhez 4 bájtnyi adat szükséges. Ez az adatmennyiség sorban feltölti az adat-, cím- és státuszregisztert. Az adat- és címregiszter szerepe egyértelmű. A státuszregiszter tartalma biztosítja azt, hogy a különböző algoritmus szerint égetendő EPROM-ok mindig a megfelelő jelet kapják. Egyszerűen ez vezérli a Vpp értékét és kapcsolja a különböző EPROM-oknál más-más funkciót betöltő lábba a vezérlő jeleket: 1. ábra. Az 1. táblázatban megtalálható, hogy a 4 bájt egyes bitjei az EPROM melyik lábára ke-

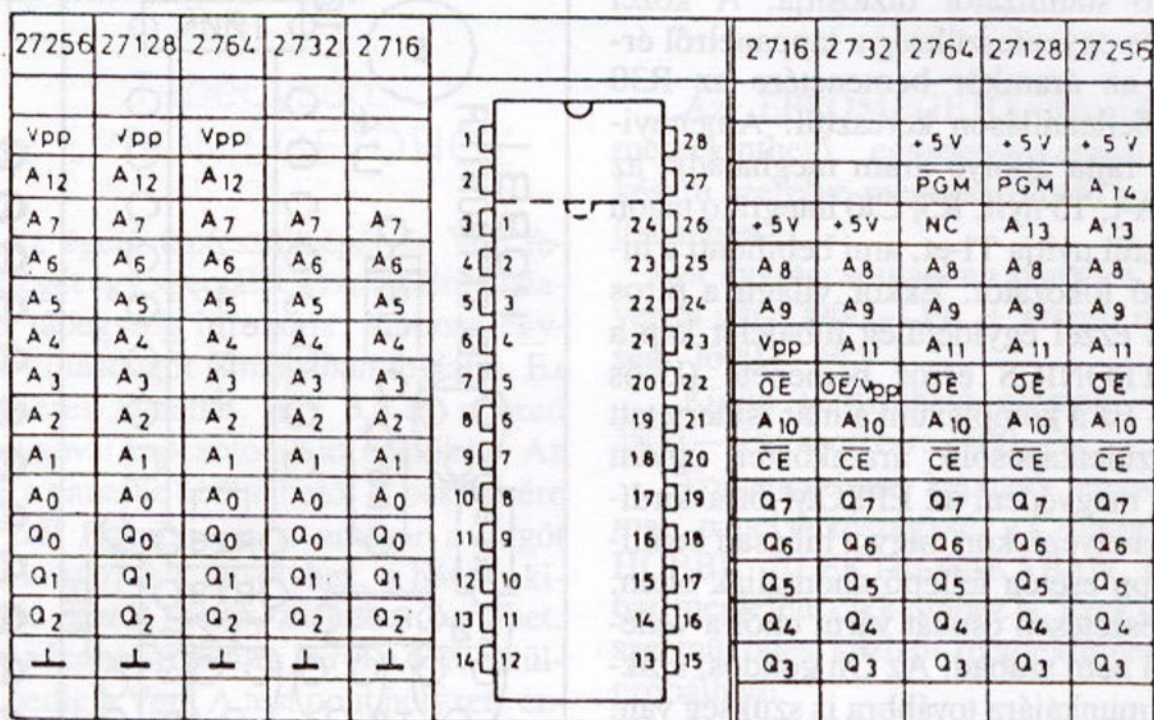
rülnek. Az égető kapcsolási rajza a 3. ábrán látható. Az adatbitek a CENTRONICS csatlakozó 2-es pontjáról (Do) az U1..U4, egyenként nyolcbites párhuzamos kimenetű siftregiszterekből álló lánc bemenetére kerül: (U1 D bemenet). A 4094 típusú siftregiszterek egy 8 bites átmeneti tárolót is tartalmaznak, melyek kimenetei háromállapotúak és pufferelek. Az aktuális adatbit az órajel felfutó élére íródik be a léptetőregiszterbe és siftelődik. Az órajelet a CENTRONICS csatlakozó 3-as pontjáról (D1) kapjuk és az U8F inverterrel puffereelve vezetjük a 4090-esek CLK bemenetére. A léptetőregiszterek kimeneti állapota az STR jel (csatlakozó 4. láb: D2) H szintje esetén íródik be a latch-be. A program a CLK jellel szinkronizálva a Do-n keresztül tölti fel a regiszterláncot. Az adatátvitel végén az adatbájt az U4-be kerül, az U3, U2, valamint U1 két bitje pedig a címet tartalmazza. A fennmaradó hat bit a Vpp-t és a tápfeszültséget kapcsolja a TEXTTOOL megfelelő kivezetéseire az U6 demultiplexer és az azt követő áramköri elemek segítségével. Az áramkörnek alapvetően kétféle működési módja van: egyrészt be lehet írni az adatot az EPROM-ba, másrészt ki lehet olvasni azt. Az égetési folyamatot e két üzemmód váltakozása jellemzi. A siftregiszterek feltöltése alatt az STR jel szintjét a program L-en tartja, így az éppen betöltés alatt álló adatok nem kerülnek a kimenetre. Ezt feltétlenül biztosítani kell, hiszen ha a kimeneten a siftelés folyamán létrejövő minden változás megjelenne, akkor az Upp és a tápfeszültség értéke véletlenszerűen változna, ami az EPROM tönkremeneteléhez vezetne.

A STROBE nemcsak a regiszterláncot, hanem az U5 párhuzamos-soros átalakítót is vezérli. Ez a tok teszi lehetővé az EPROM megcímzett rekeszének beolvasását a számítógépbe. Mivel az STR jel erre a csipre ellentétesen hat a 4021 bemenetei összeköthetők az U4 kimeneteivel. A korrekt logikai szintek biztosítása érdekében az U4 és U5 közösített vonalait

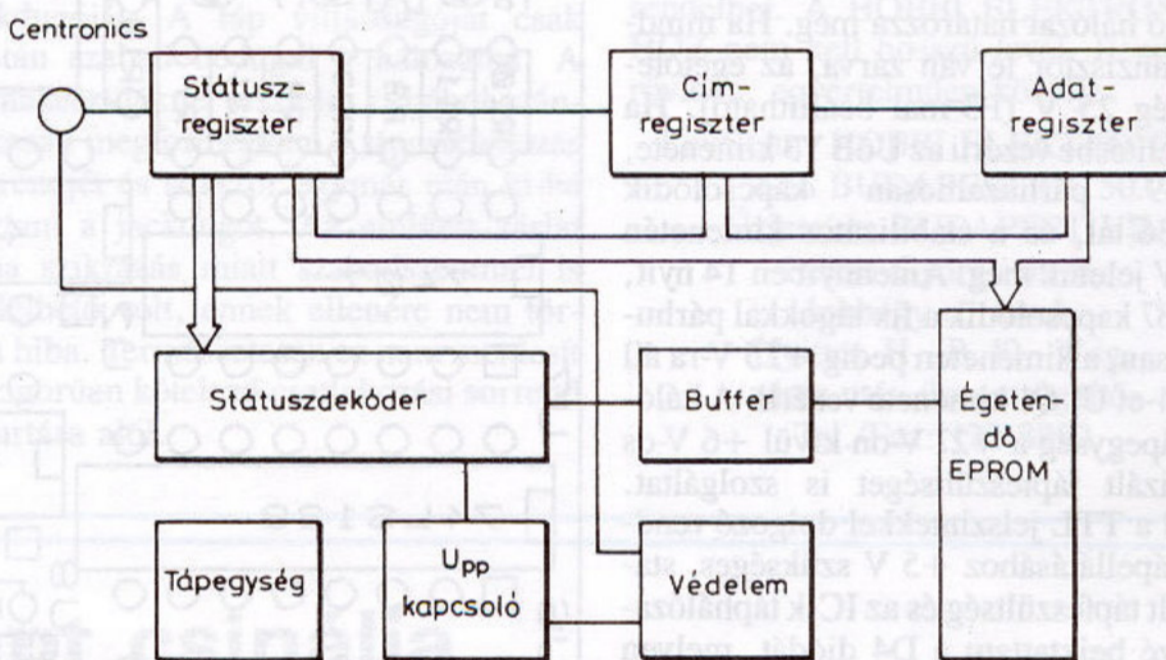




3. ábra



1. ábra



2. ábra

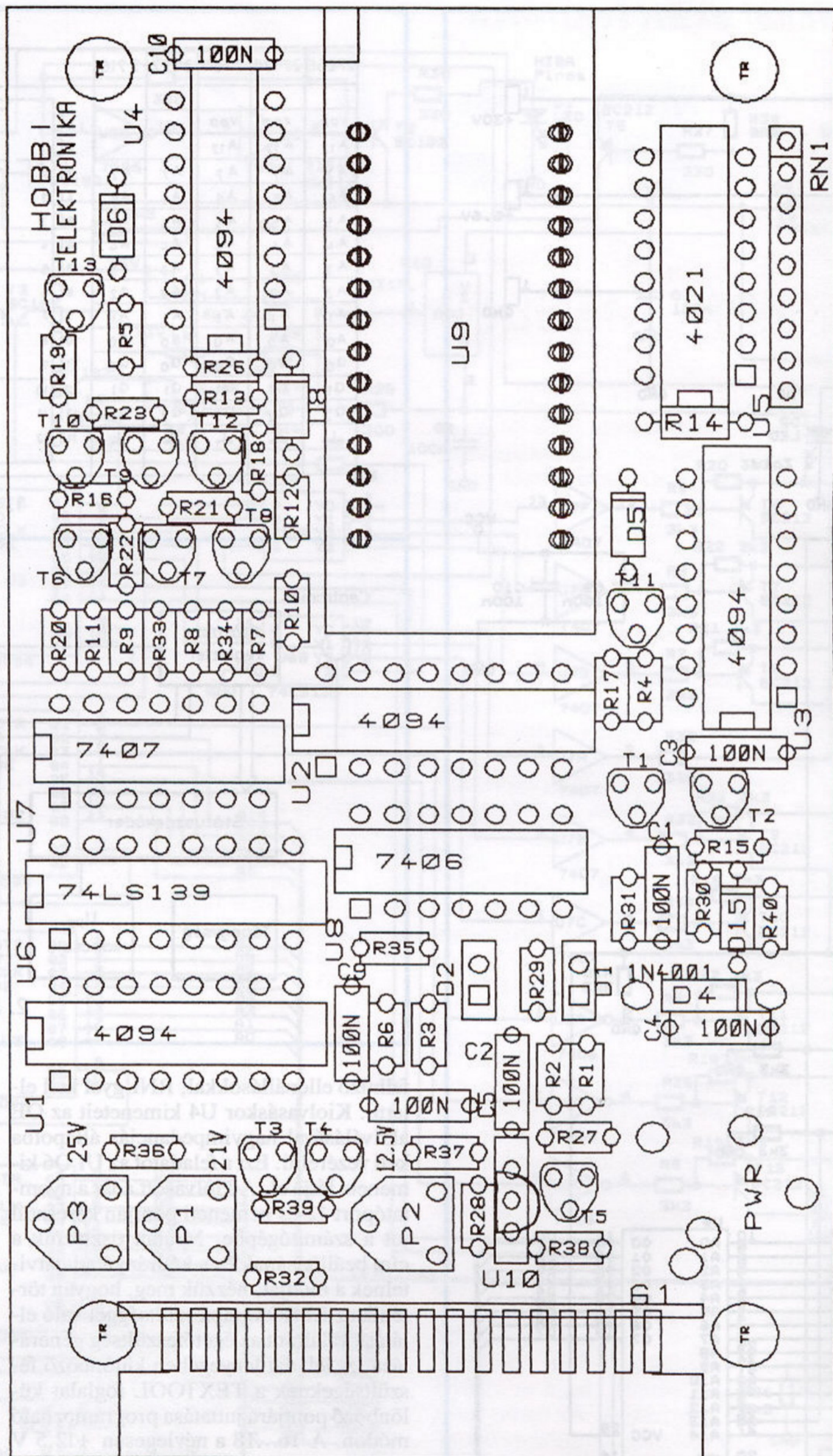
felhúzó ellenállásokkal, RN1-gyel kell ellátni. Kiolvasáskor U4 kimeneteit az OE aktiválásával nagyimpedanciás állapotba kell vezérelni. Ezt a feladatot az U1 Q6 kimenete látja el. A kiolvasott adat a nyomtatóport 12-es bemeneti pontján keresztül jut a számítógépbe. Miután tisztáztuk a cím beállításának és a kétirányú adatátvitelnek a módját, nézzük meg, hogyan történik az EPROM tápfeszültséggel való ellátása valamint az égetőfeszültség generálása. A feladat lényegében különböző feszültségeknek a TEXT00L foglalat különböző pontjára juttatása programozható módon. A T6...T8 a névlegesen +12,5 V +21 V, +25 V-os Vpp-t kapcsolja. Ezek közül egyszerre mindig csak az egyik aktív, erről az U6 gondoskodik. A tranzistorok meghajtása az U7 nyitott kollektoros nem-invertáló pufferein keresztül történik. A multiplexer ezen egységének 1G engedélyező bemenetén keresztül az

áramkör letiltható. Erre áramköri hiba esetén kerülhet sor. Az 1G vonalat a T1, T5, U8E, U8D és passzív hálózatukból felépített rövidzárvédelmi áramkör emeli H-ra, ha az Upp-vonal túlterhelődik. A túlterhelést a T2-vel meghajtott D1 jelzi. Mivel normális esetben a védelem inaktív, a sifregiszterek feltöltése alatt U6A Yo kimenete L szintű, így azt vezérlésre nem használjuk. Az LS 139 B fele az EPROM tápfeszültségét kapcsolja az előzőekhez hasonló módon, puffereken át vezérelt T9, T10 segítségével. T9 az EPROM behelyezésénél, illetve eltávolítása során megkövetelt feszültségmentes állapotot biztosítja. T10 feladata a 27256 és 27512 esetében a +6 V-ra növelt tápfeszültség kapcsolása. T11...T13 pufferként funkcionálnak, ugyanis azok az EPROM-kivezetések, amelyeket a különböző típusoknál kapcsolgatni kell, TTL kimenetről közvetlenül nem hajthatók meg. Az égetőfe-

szükséget az U10 „programozható” át-eresztő stabilizátor biztosítja. A közel 27 V-os egyenfeszültség a táppanelről érkezik az áramkör bemenetére az R30 figyelőellenálláson keresztül. Amennyiben a rajta átfolyó áram meghaladja az 50 mA-t, T5 nyit, R5, C10 integráló tagon keresztül nyitja T1-et, ami bebillenti a hibajelző fokozatot. Ekkor világít a piros led és ezzel egyidejűleg hibajelet kap a CENTRONICS error bemenete (15-ös pont). Ez a komplexum a már ismertett tápfeszültségkapcsoló áramkörrel együtt képes megvédeni az EPROM-ot a fordított behelyezéskor, vagy a hibásan beállított Vpp esetén fellépő anomáliák ellen. Természetesen csodát várni ettől a védelemtől nem szabad! Az Ön gondos, szakember munkájára továbbra is szükség van. A stabilizátor IC kimenőfeszültségét a fix értékű R28-ból és a T3, T4 tranzisztorok vezérlésével átkapcsolható R36, R37, R38, P1...P3 elemekből felépített visszacsatoló hálózat határozza meg. Ha mindkét tranzisztor le van zárva, az égetőfeszültség 25 V (P3-mal beállítható). Ha T3-t telítésbe vezérli az U6B Y3 kimenete, P1-R39 párhuzamosan kapcsolódik P3-R36-tal, és a stabilizátor kimenetén +21 V jelenik meg. Amennyiben T4 nyit, P2-R37 kapcsolódik a fix tagokkal párhuzamosan, a kimeneten pedig +25 V-ra áll be. T4-et U1 Q1 kimenete vezérli. A hálózati tápegység a +27 V-on kívül +6 V-os stabilizált tápfeszültséget is szolgáltat. Mivel a TTL jelszintekkel dolgozó rendszer tápellátásához +5 V szükséges, stabilizált tápfeszültség és az IC-k táphálózata közé beiktattam a D4 diódát, melyen 07 V esik.

### Az áramkör szerelése, élesztése

A kapcsolás bonyolultságából sejthető, hogy azt kétoldalas, furatgalvanizált NYÁK-on kell megépíteni. (NYÁK a szerző üzletében beszerezhető.) Az alkatrészek ültetését a 4. ábra szerint végezze el a NYÁK alapos ellenőrzése után! Az ültetés javasolt sorrendje a következő: helyezze a TEXTTOOL-t a panelre. A lábait igazgatva, ha szükséges egy picit megreszelve ültesse a helyére. Erre azért van szükség, mert a furat a galvanizálás során veszített a méretéből, ezért a TEXTTOOL lába szorosan illeszkedik a furatba. Ha ez megtörtént, a gyorscsatlakozót emelje ki, majd forrassa be a többi alkatrészt R—C—D—T—IC-foglalt sorrendben. Az alkatrészeket a lehető legrövidebb kivezetésekkel forrassa be, hogy a két LED kivételével egyik sem emelkedjen a TEXTTOOL foglalat szintje fölé, ugyanis az előlapon



4. ábra

vágott nyíláson és furatokon csak ez és a két LED bújik ki a dobozból. Fontos, hogy a gyorscsatlakozó működtető karja a doboz végén lehajtható legyen, ezért azt

ott be kell jelölni, majd ki kell sliccelni. Az IC-eket érdemes foglalatba tenni, hogy meghibásodás esetén könnyen cserélhető legyen. Az égető és a számítógép közötti

Az IBM EPROMÉGETŐ  
árájánlata

Az EPROMÉGETŐ mintadarabja megtekinthető, egységcsomagban vagy készre szerelve megvásárolható a szerző üzletében.

Ára egységcsomagban panellel, szoftverrel stb.: 7900,— Ft, + szerelt tápegység: 1600,— Ft.

Külön vásárolható elemek: panel: 1500,— Ft; szoftver: 1500, Ft.

Folyamatosan kapható egységcsomag, panel és részegység a C-újságban és HOBBY ELEKTRONIKÁBAN korábban megjelent cikkeimhez is. Ezek egy része működés közben megtekinthető, kipróbálható.

A vidéki olvasóknak segít a szerző levelező-egységcsomagküldő szolgáltatása. A megrendelt csomagot postán utánvétellel elküldöm. Telefonon és levélben is rendelhet. A HOBBY ELEKTRONIKÁHOZ nem kell hosszú levél. Rendelését röviden, egyértelműen közölje.

Levélcím: HOBBY ELEKTRONIKA  
1656 BUDAPEST Pf. 50.

Üzletcím: BUDAPEST VII.

Dózsa György út 16.

Jobbágy u. sarok.

Nyitva: H—P. 10—17-ig,  
zárás után üzenetrögzítő.

Tel./Fax: 122-8892

kor a gépet le kell lassítani. Erre minden alaplap lehetőséget biztosít.

Tápegység  
az EPROMÉGETŐHÖZ

Az égető tápfeszültségét és Vpp feszültségét egy speciális, erre a célra kialakított tápegység biztosítja. Doboza egyben a villásdugót is magában foglalja. Ez háromeres kábellel, egy 3,5-es sztereó jackdugóval kapcsolódik az égetőhöz. Az aljzat a panelbe beültethető. A bekötésére oda kell figyelni, mert amikor a dugót csatlakoztatjuk a hüvelyhez, a három kivezetés közül kettő zárlatba kerülhet. Ezért a legelső pontra kerül a 6 V, a külsőre pedig a Vpp. A test pont helyzete ezután egyértelmű. A fentiek miatt be kell tartani a készülékek csatlakoztatásának sorrendjét is. Először a számítógépet csatlakoztassa az égetőhöz, majd a tápegység jackdugóját. A táp villásdugóját csak ezután szabad bedugni a hálózatba. A mintakészüléknél tesztelés céljából szándékosan megfordítottam a tápcsatlakozás sorrendjét és sokszor egymás után ki-be húztam a jackdugót. Az említett zárlat néha szikrázás miatt szabadszemmel is észlelhető volt, ennek ellenére nem történt hiba. Természetesen ez nem mentesít a szigorúan kötelező csatlakozási sorrend betartása alól.

összeköttetés közönséges pinterkábelen keresztül történik. Ennek csatlakozója a NYÁK-ra ráültethető, így az egész égető a korábbi változathoz képest kompakt egységet alkot.

A kész áramkört HE 1 jelű lapos, feliratozott műanyagdobozba kell beépíteni. Ha gondosan szerelt és ezt ellenőrizte, hozzáfoghat az élesztéshez, beállításához. Ehhez a szoftver elengedhetetlen. Közlésére a terjedelme miatt itt nincs mód. Kezelése különösebb magyarázatra nem szorul, mert egyértelmű, áttekinthető menüvel a HELP-pel rendelkezik. Az élesztéshez a TESZT fájlt olvassa be. Ez egy nagyon jó diagnosztikai program, melynek funkcióit műszeres méréssel helyettesíteni, tekintettel az áramkör soros felépítésére szinte lehetetlen. A TESZT programot futtatva viszont elegendő az ellenőrzéshez egy közönséges mutatós kéziműszer, amivel az adat- és címbuszt, valamint az égetőfeszültségeket ellenőrizheti. Ez utóbbiak értékét a hozzájuk tartozó trimmerekkal állítsa be. Hiba esetén forrasztási hibát, zárlatot, szakadást keresen. A programot DD-s lemezen kétféle verzióban adom. Mindkettő használható ízlés szerint, kinek melyik tesz jobban. Egyetlen különbség az, hogy 27512-öt csak az újabb változat égeti. FONTOS! Előfordulhat a meglévő szoftverrel, hogy gyorsabb 386-os és 486-os gépeknél a sebesség miatt adatvesztés történik. Ilyen-

## Mindenként másként csinálja

Még „malacfarknyi” programok írásakor is érdemes ökonómikus programírási stílusra törekedni. Ezáltal nem csupán tárhelyet, lemezkapacitást, papírt és munkát takarítunk meg, hanem a program futása is gyorsul.

Példaként, egy megjelent program részletén mutatjuk be a lehetőségeket. (Szerzője maradjon anonim.)

```
30 GET $IF A$=" " THEN 30
35 IF A$="1" THEN 200
40 IF A$="2" THEN QQ=100:GOTO 500
50 IF A$="3" THEN 1000
55 IF A$="4" THEN 500
60 IF A$="" THEN 30
```

290 GOTO 370

370 GOTO 200

500 PRINT CHR\$(147);

És most nézzük az „átíratot”

```
30 WAIT 198,1:GET A$:A=VAL(A$)
35 IF A (1 OR A) 4 THEN 30
40 ON A GOTO 200, 490, 1000, 500
```

290 GOTO 200

490 QQ=100

500 PRINT CHR\$(147);

De ha már az ON X GOTO utasításnál tartunk... Ha nem is gyakran, de előfordulhat hogy a sorszámlista nem fér ki egyetlen logikai sorba. Az ilyen esetben alkalmazandó megoldáshoz tudni kell, hogy ha az X értéke nagyobb, mint a sorszámlista elemeinek száma, a vezérlés „leesik” a következő sorba, ott folytatódik a program. Ezért alkalmazható az alábbi megoldás.

```
50 ON X GOTO 100, 200, 300, 400, 500
60 ON X-5 GOTO 600, 700, 800, 900, 1000
```

De a vezérlés akkor is „leesik”, ha X=0. Nézzünk erre is egy példát. A „hagyományos” megoldás:

```
90 ON X GOTO 100, 200, 300, 400, 500
100 REM
```

A fentiek alapján a 90. sort így módosíthatjuk:

```
90 ON X-1 GOTO 200, 300, 400, 500
```

Mivel a szubrutinból, az ON X GOSUB utasítást követő sorra tér vissza a vezérlés, bonyolultabb megoldáshoz kell folyamodni:

```
200 IF X) 5 THEN 230
210 ON X GOSUB 10, 20, 30, 40, 50
220 GOTO 240
230 ON X-5 GOSUB 60, 70, 80, 90, 100
240 REM folytatás
```

Szász Sándor

# Utasítások

Minden Basic-programozó ismeri azokat az utasításokat és parancsokat, amiket a C-64 megért. De hogy mi mindent lehet azokkal csinálni, már sokkal kevesebben tudják. Emiatt itt most bemutatunk néhány speciális alkalmazást.

\*

Bizonyára közülünk nagyon sokan tudnak bánni a Basic-utasításokkal. Tudnak azokkal programokat írni. Ha valaki kezdő, akkor is hamar megtanulja azokat.

Azonban néhány utasítás egyes olyan problémák elegáns megoldására alkalmas, amelyekre nem mindenki jönne rá magától. Ezekben az esetekben gyakran szellemes műfogásokat (kunsztstikliket) is fel lehet fedezni. A legjobb, ha egy példán mutatunk be ezekből valamit.

A példa: Tételezzük fel, hogy egy programban FOR ... NEXT ciklust kell használnunk, amiben azonban sem a kezdő-, sem a végérték nincs meghatározva. Többnyire ezek olyan értékek, amiket a program futása során először be kell adni, vagy a ciklus futása előtt ezek kiszámítódnak. A mi esetünkben legyen ezen értékek változóinak a neve A és E. Most egyszerűen az alábbi programsorokra gondolhatunk.

```
10 INPUT "KEREK 2 ERTEKET  
BEADNI"; A, E
```

```
...  
100 FOR X = A TO E  
110 ...  
120 NEXT E
```

Így nézne ki a ciklus, nemde? De az ilyen látszólag egyszerű megoldásnak is megvan a maga alattomos hátulüttlője. Mi van pl. akkor, ha E kisebb, mint A? Akkor a ciklus pontosan csak egyetlenegyszer futna le, nem többször, mert X nagyobb, mint E. Ezt az eshetőséget tehát a programírásakor valahogy figyelembe kell venni.

Elég bonyolult módon ez az eset vizsgálatával eldönthető, majd két lehetséges esetre (A nagyobb/egyenlő E, illetve A kisebb mint E) külön-külön lehet ciklust írni. Elegánsnak ez a megoldás nem nevezhető, azonkívül feleslegesen foglal el memóriahelyeket is, mert nemcsak hogy két ciklus kell hozzá, hanem még egy IF vizsgálat is, ami az A és E változók egymáshoz viszonyított értékétől függően a megfelelő ciklusra irányítja a program futását.

Lényegesen rövidebb a probléma megoldása SGN-függvény alkalmazásával. Legyen őszinte, mondja meg; megoldott egyáltalán erre az utasításra?

Nézzük meg kissé pontosabban ennek az utasításnak a hatását:

$$V = \text{SGN}(\text{kifejezés})$$

értéke  $-1$ , ha a kifejezés értéke negatív,  $0$  ha a kifejezés értéke is nulla és  $1$ , ha a kifejezés értéke pozitív. Tehát pl.

$$V = \text{SGN}(4-12)$$

értéke  $-1$ , mert a kifejezés negatív szám.

A mi esetünkben a SGN-függvény arra szolgál, hogy eldöntsük, vajjon a ciklusváltozó értéke növekvő vagy csökkenő. Ehhez csupán a FOR ciklust kell kiegészítenünk egy STEP-pel, mondjuk a következő formában:

```
100 FOR X = A TO E STEP  
SGN(E-A)
```

Ha most az E (végérték) kisebb, mint az A (kezdőérték), a STEP értéke  $-1$  lesz. Ennek következtében a ciklus „visszafelé számlál”. Ellenkező esetben a STEP értéke  $+1$  marad, és a számlálást a megszokott módon, „felfelé számlálva” zajlik le.

Amint látható, az eredeti programhoz egyetlen kicsi kiegészítés kell ahhoz, hogy a ciklusunk univerzálissá váljon. A kétciklusos megoldáshoz képest jelentős mennyiségű memóriahelyet takarítottunk meg.

Ha  $1$ -től eltérő lépésközzel kell a ciklusnak lefutni, elegendő, ha a SGN függvény mögé írjuk a megfelelő szorzótényezőt:

```
100 FOR X = A TO E STEP  
SGN(E-A) * tényező
```

Ezt az utasítást tehát, amit többnyire csak matematikai feladatokban alkalmazva tételeznénk fel, mint látjuk, egészen másképp is lehet alkalmazni.

## Függvénydefiniálás

A következő utasítás, amit csak nagyon ritkán alkalmazunk, pedig nagyon sokra képes, a DEF FN. A szintaxisa pontosan:

DEF FN 1. Változó [2. Változó] = aritmetikai kifejezés. (A C-64-en csak egy független változó lehetséges!)

Röviden szólva a Basic 2.0 verziója ezzel lehetőséget nyújt kiegészítő függvények megfogalmazására, amit az igényünknek megfelelően tudunk megszerkeszteni, definiálni.

Ha pl. egy programban többször ki kell számítani az

$$A * B + (C/2)$$

kifejezést, akkor ezt a legcélszerűbben mindjárt a program elején definiáljuk függvényként a következőképpen:

```
DEF FN F(X) = A * B + (C/2)
```

Attól kezdve, ahogyan ezen az utasításon a program indítás után átfutott, bárhol hívhatjuk, hívhatkozhatunk rá egyszerűen, mint pl. a következő egyenletben:

$$\text{Változó} = \text{FN F}(X)$$

Ebben az esetben az X úgynevezett „vakváltozó”. Magának a függvénynek a kiértékeléséhez nem is szükséges. (Mint látjuk, példánk szerinti függvénydefiniációban nem is fordul elő!). Csak arra szolgál, hogy megakadályozza a Basic-interpreter akadékoskodását, „mekegését”. Természetesen előfordulhat ez a „független változó” a függvénydefiniáció jobb oldalán is, mint pl.:

```
DEF FN F(X) = X * X * PI
```

Ezzel a képlettel egy X-sugarú kör területét tudjuk kiszámítani. Ha ezek után a függvényt bárhol a programban felhívjuk pl. így:

```
PRINT FN F(4)
```

a képernyőn olvashatjuk a  $4$  cm sugarú kör területét

A Basic gyakran használt függvényeinek ezen kibővítési lehetőségét a programozók sajnos csak nagyon ritkán használják.

Fontos tudni, hogy a függvénydefiniációk maguk is tartalmazhatnak már korábban definiált függvényre való hivatkozást. Példaként mutassuk be a körhenger felületének a kiszámítását, amihez először ki kell számítani az alaplap, majd a palást felületét. Ez utóbbit az alaplap kerületének és a palást magasságának a szorzata adja. Így tehát a kör területének és a palást területének a képletei szükségesek, amiket korábban definiálni kell:

```
10 DEF FN F(X) = X * X * PI
```

```
20 DEF FN U(X) = 2 * X * PI
```

Ezek után már nem probléma a felület függvényének a definiálása:

```
30 DEF FN O(X) = 2 * FN F(X) +  
FN U(X) * H
```

ahol H a henger palásthossza ill. magassága.

No a geometriából egyelőre elég lesz ennyi. Mindenesetre gadagabbak lettünk három használható függvényvel.

A programok írása közben töprengve biztosan számos további függvénydefiniáció fog eszünkbe jutni, amiből idővel

egész szép kis számítási segédeszköz-könyvtár állítható össze. Mivel a programok így rövidebbek és okos névválasztás esetén áttekinthetőbbek is lesznek, egyre több alkalmazási lehetőséget is fogunk találni. Vegyük azonban figyelembe, hogy a függvénydefiníció és hívás csak program módban használható.

## Betűsaláta

Különösen adatbankoknál és hasonló programoknál csaknem mindig előadódik egy probléma: a szavak rendezése. A szavak lehetnek nevek, városok, árulista vagy bármi hasonló. Találjunk ki rá módszert, hogy gyorsan visszakaphassuk a szavakat „névsorba rendezett” lista vagy adatállomány formájában. Azonban hogyan lehet azt megvizsgálni, hogy egy adott szó a következő elé vagy után helyezendő el a névsorban?

Először nézzük csak meg, hogyan csinálnánk ezt a számokkal. Egészen egyszerűen a < és > összehasonlító operandusokkal. A számokat sorjában egyenként összehasonlítanánk az összes többivel. Ha a sorrend nem egyezik, a két számot felcseréljük. Érdekes módon a sztringekkel ez az összehasonlítás hasonlóképpen működik. Erre egy példa:

```
10 A$ = "MAGYAR"
20 B$ = "NEMETH"
30 IF A$ < B$ THEN PRINT
   "HELYES!"
```

Ez a kis program az adott esetben valóban a „HELYES!” szót írja ki. Próbáljunk ki egy másik esetet. Írjuk az első sorba:

```
10 A$ = "SZABO"
```

Most nem kapunk semmilyen feliratot a képernyőn.

Mivel többnyire s sztingösszehasonlítások esetében a szövegeket ASCII-kódba alakítja át a gép, majd azokat mint számokat kezelve végzi el az összehasonlítást, jó adag számítási munkát és időt takarítunk meg magunknak és a gépnek is. Ezen túlmenően a program is rövidebb és jobban olvasható lesz, amit úgy a tizedik javításnál már nagyon is tudunk értékelni.

Az 1. lista a rendező rutint mutatja, ami ugyan nem a leggyorsabb, de világosan követhető rajta, hogyan működik. A folyamatábra megvilágítja a lényeges lépéseket.

A rutin abból indul ki, hogy a rendezendő mező A\$(x), és x csak 0–10 közötti értékű lehet. Ha nagyobb kellene, a 15010 sorban a cilus felső határt kívánságunk szerint változtassuk meg. A rutin így működik:

Valahonnak a programból GOSUB

```

14090 rem egyszerű rendezo
14091 rem szamlalo nullazasa.
15000 f=0
15005 rem rendezociklus indul
15010 for a=0 to 9
15015 : rem helyes a sorrend ?
15020 : if a$(a) < a$(a+1) then 15070
15025 : rem ha nem, csere
15030 : b$a$(a)
15040 : a$(a) = a$(a+1)
15050 : a$(a+1) = b$
15055 : rem csere volt, szamlalo=1
15060 : f=1
15065 : rem következő pár vizsgálata.
15070 next a
15075 rem ha nem volt csere, kész
15080 if f=0 then 15110
15085 rem ha volt, szamlalot torolni
15090 f=0
15095 rem ciklust újra futtatni
15100 goto 15010
15110 return

ready.
```

15000 utasítással kell felhívni. A 15000 sor a F változót nullázza. Ez azt mutatja, „érezkeli”, hogy egy rendezési átfutás során két vagy több sztring kicsréldött-e, vagy nem történt csere.

Ezt követően a 15010 orban kezdődik meg a tulajdonképpeni rendezési munka. Ha 10-nél több sztringet tartalmazó mezőt akarunk rendezni, a FOR ... NEXT ciklus végértékét megfelelően változtatni kell (max. érték -1). A 15020 sor megvizsgálja, hogy az aktuális sztring kisebb-e, mint a következő. Itt a „kisebb” azt jelenti, hogy előbb kell álljon a sorrendben. Ha a relációra igen a válasz, a sztring a helyén marad, a 15070 számú sorban a ciklus sorszáma eggyel nő és a következő sztring vizsgálata kezdődik. Ha a relációvizsgálatra nem a válasz, akkor a 15030–15050 sorokban lezajlik a két sztring helycseréje. Mivel két sztringet egymással „felülír-

ni” nem lehet, az egyik közbülső, átmeneti tárolására a B\$-t használjuk.

Ha helycsere történt, a 15060 sorban F értéke 1-re változik. Amiatt a cilusnak legalább még egyszer le kell futnia mindannyiszor, amíg többé helycsere nem történik, vagyis F értéke 0 marad. Ha ez a helyzet, akkor a 15080 sorban F értéke 0-nak bizonyul, és a szubrutin a 15110 RETURN hatására visszatér. Ha mégis történt csere, akkor F nem 0, és a 15100 újra a ciklus elejére irányítja a futást. Ez a játék tehát mindaddig folytatódik, amíg a sztringek sorrendje hibátlan nem lesz.

Amint látható, a rendezőrutin viszonylag rövid. Hozzátehetjük, hogy vannak sokkal gyorsabb eljárások is, de azokról nem látható, hogyan is dolgoznak. Emiatt tűnt számunkra ez a példa demonstrációra alkalmasabbnak.

# Még többet ésszel!

## XXII. rész

Soros állományba írni roppant egyszerű dolog. Az első teendő persze az állomány megnyitása írásra, aztán a kiírás a PRINT# utasításokkal, végül az állomány lezárása. Vedd észre, hogy nem PRINT, hanem PRINT# utasítást említettem, ez két szempontból is lényeges lehet. Egy régebbi részben már elmondtam, hogy a CMD és PRINT együttes használatának buktatóit elkerülhetjük a kifejezetten file írására kitalált PRINT# utasítással. Továbbá ezzel hangsúlyoztam, hogy ez egy külön utasítás, másik tokenel, amelyben a "#" jelet nem szabad szóközzel elválasztani.

Az utasítás pontos leírása a következő: **PRINT#lf[,nyomatási kép]]**. Az lf most is a megnyitott file logikai száma, megegyezik az OPEN-ben használttal. Apropos, OPEN. Eddig elfelejtettem kiemelni, hogy ha egy bizonyos logikai számmal (lf) megnyitott állományt lezártunk, akkor ezután ugyanazt a számot bármilyen másik file megnyitásához szabadon felhasználhatjuk.

Aki már valaha is használta a PRINT utasítást, az tudhatja, hogy a „nyomatási kép” mit jelent: különféle kifejezéseknek tabulátorokkal (; és ,) elválasztott sorozata, már amennyire ezt szakvakban röviden meg lehet fogalmazni. A szögletes zárójelek azt jelzik, hogy a **nyomatási kép** lehet üres is, ennek is megvan a maga értelme. Ha viszont nem üres, akkor a fileszámtól egy vesszővel kell elválasztani.

A lemezre írás szabályai némileg eltérnek a képernyőre és a nyomtatóra írás szabályaitól, ezeket talán az 1. példa segítségével próbálhatjuk meg sorra venni. Általános szabály szerint a lemezre kiírt adatok után minden PRINT# utasítás egy Return (13-as) karaktert is elküld a gép. A 2. sor így tehát négy karaktert ír az állományba: a három betűt és a Returnt. A Returnnek a beolvasás során jut majd igazi főszerep. A Return kiírását a 3. sorban látható módon lehet elnyomni, a PRINT-nél megszokott módon, a nyomatási kép végére tett pontosvesszővel. Az ezután kiírt karakter közvetlenül az „ABC” szöveg után lesz tárolva.

A 4. sor hét karaktert ír a lemezre, köztük a vesszőt is, hiszen az a kiírt karakterlánc része. Az 5. sorban a vessző már úgynevezett tabulátor, így ha a kivitel a képernyőre szólna, a „DEF” elé annyi „kurzor jobbra” jel íródna ki, hogy a „D” betű a következő tizedes oszlopba kerüljön. Ezzel szemben a lemezre írva minden vessző tabulátor pontosan tíz szóköz kiírását eredményezi. Mivel ennek nem sok értelme van, nem szokás ilyet csinálni. A 6. sor hatása már a megszokott, azaz a hat betűt egymás után, határolójel nélkül írja az állományba.

Másképp néznek ki a numerikus kifejezések is. A 7. sorban két karakterlánc kiírása történik, az eddigiek szerint lemezre kerül a két számjegy („12”) és persze egy Return. A 8. sorban a számok már valódi numerikus kifejezések. Ezek kiírásakor a számjegyek elé kerül az előjel, ami nem-negatív szám esetén egy szóköz; a számjegyek után pedig egy újabb szóköz íródik ki. Ez a szerkezet megegyezik a nyomtatóra írás eredményével, de eltér a képernyőre íráskor (a szám után egy „kurzor jobbra”) és az STR\$ függvény által elért (a szám után nincs semmi) képtől.

A 9. és 10. sorok hatása megegyezik. A **nyomatási kép** itt üres, vagyis itt csak a Return karakter kerül az állományba. Más

néven ezt az üres kiírást soremelésnek hívjuk, mintha csak a képernyőre küldenénk egy PRINT-et.

Van a nyomtatási képek szerkesztésének egy apró, nem túl ismert, de néha igen lényeges fortélya. A 11–13. sorokban háromféle módon írok ki két kifejezést. Az első módszer nem hatásában, hanem a végrehajtás módjában különbözik a másik kettőtől.

A 12. sorban látható módszert, azaz a kifejezéseket elválasztó pontosvesszők elhagyását, elsősorban helytakarékoság okán gyakran alkalmazzák, én szintúgy. Az ezzel egyenértékű, 13. sorbeli változatban a plusz jel a kifejezést egyetlen kifejezéssé fűzi össze (konkatenálja); a 12. sorban ugyanez történik, külön kérés nélkül. Az interpreter először kiértékeli a teljes kifejezést, és ennek eredményét írja csak ki. Ha a kifejezés bármelyik tagja valamiért hibás, akkor az egésznek az előállítása és kiírása elakad, vagyis a lemezre semmi sem kerül belőle. Ellenben a 11. sorban a kiértékelés először csak a pontosvesszőig halad, az eredményt a gép az állományba írja, és csak ezután veszi elő a második kifejezést. Ha ez hibás, az első már akkor is elment.

Egy másik, veszélyesebb különbség is van. Előfordul néha, hogy egy szuszra olyan hosszú adagot írunk az állományba, amely teljes hossza összesen meghaladja a 255 karaktert. Az első módszer, amíg a kifejezések külön-külön ennél rövidebbek, hibátlanul működik. Ha viszont a második módszert használjuk, a gép előbb összefűzi a karakterláncokat, és mint részeredményt a memóriába tárolja. Ebből adódóan ha az együttes hossz túl nagy, hibaüzenetet kapunk, szemre teljesen indokolatlanul. Adatbázis kezelésekor mi sem rosszabb egy véletlen leállásnál.

Rövid karaktorsorozatok lemezre írásakor, de olvasásakor is megfigyelhető, hogy a lemez hol megindul, hol megáll, miközben a gép nagyjából folyamatosan küldi az adatokat a lemezre. Ez azért van, mert a géppel nem a lemezre írunk! Az állományba elküldött adatok a lemezegység memóriájában levő egyik átmeneti tárolóba, műszóval pufferbe kerülnek. Ezt a puffert írtuk-olvastuk a közvetlen lemezműveletekkel is. Íráskor ha a puffer megtelik, a lemezegység automatikusan kimenteti a tartalmát a soron következő lemezblokkba, a puffert kiüríti, és tovább fogadja az érkező adatokat. Olvasáskor a lemezegység egyszerre egy blokkot tölt be a pufferbe, és onnan adagolja az adatokat a kiolvasás által diktált ütemben. Azért, hogy ezeken a feladatokon levegye a vállunkról, kapott a lemezegység egy önálló proceszort.

Ez a szakaszos tárolás azért szükséges, mert az adatok lemezre rögzítése sokkal lassabban megoldható műveletet, mint az adat átvitele a soros adatkábelén (pedig az sem villámgyors). Ha minden byte lemezre írását meg kéne várnia a gépnek, mielőtt a következőt elküldhetné, bizony, egy blokk kiírása egy percig is eltarthatna. Ehelyett ugyanaz történik, mint a még lassabb nyomtató esetében is: a külső eszköz először összegyűjt egy adagra való az adatokból (lemezen ez 256 byte), majd azt a saját behangolt rendszerével gyorsan kiírja, egyszerre.

Miért érdekes ez most számunkra? Eddig többnyire valahogy úgy fogalmaztam, hogy ezt vagy azt a lemezre, az állományba írjuk. Nos, ha az írás közben a lemezt kivesszük, beüt egy áramszünet, vagy más módon megszakítjuk a csatornát, akkor mindaz még nem rögzült a lemezen, amit a lemezegység még éppen a pufferében gyűjtögetett. Ez szerencsére nem lehet több 255 byte-nál, de ennyi adat elvesztése is használhatatlanná tehet egy adatbázist. Tehát számítani kell rá, hogy amit a lemezre állo-



mányba elküldtünk, talán még nincs biztonságban. Az állomány lezárásakor (CLOSE utsítás) a csatorna szétkapcsolása előtt a lemezegység kiírja a még a pufferében levő adatokat. Csak ezután alszik ki a kis piros lámpa, ha nincs más adatcsatorna nyitva, és a lemez csak ezután vehető ki.

Említettem, hogy a programfile is egy egyszerű soros állomány, amelyben akár adatok is tárolhatók. Az eddig tárgyalt módszer helyett másképp is írhatunk bele, nevezetesen a SAVE paranccsal, amellyel speciálisan a tár egy részének tartalma menthető el, akár adathalmaz, akár egy program. Magában foglalja az állomány megnyitását, az adatok kiírását és az állomány lezárását is. A parancs alakja: **SAVE fn,dn[,sa]**. Sorrendben a paraméterek: filenév (karakteres kifejezés), eszközszám és másodlagos cím. Ez utóbbi használata megengedett, de semmilyen értéknek semmilyen látható hatásáról nem tudok. A programfile első két byte-ja a kimentett társzakasz kezdőcíme lesz. Ha ezt az állományt OPEN-nel olvassuk, akkor az első két byte-ot ennek tudatában kell kezelni. Ha pedig a programfile-t LOAD-dal akarjuk betölteni és OPEN-nel létrehozni, akkor ezt a két byte-ot saját magunknak kell kiírni, mert a LOAD számítani fog rá. A többi tudnivalót a XIV. részben már elmeséltem, vagy megegyezik a többi soros állomány írásának szabályaival.

A folytatás már az olvasásról, az ST változó értékeiről és a katalógusfile beosztásáról fog szólni. Találkozunk.

Hódi Gyula

```

○ ( 1. ) ○
○
○ 1 OPEN 2, 8, 3, "@:PROBA, S, W" ○
○ 2 PRINT#2, "ABC" ○
○ 3 PRINT#2, "ABC" ; ○
○ 4 PRINT#2, "ABC, DEF" ; ○
○ 5 PRINT#2, "ABC", "DEF" ; ○
○ 6 PRINT#2, "ABC" ; "DEF" ; ○
○ 7 PRINT#2, "1" ; "2" ○
○ 8 PRINT#2, 1 ; 2 ○
○ 9 PRINT#2, ○
○ 10 PRINT#2 ○
○ 11 PRINT#2, CHR$(34) ; "ABC" ○
○ 12 PRINT#2, CHR$(34) "ABC" ○
○ 13 PRINT#2, CHR$(34) + "ABC" ○
○ 14 CLOSE 2 ○
○
○
○

```

# IBM PC

## Softwareújdonságok

### Képűjság

Sokakban talán már felmerült a gondolat, hogy mi is lenne, ha ők is készítenének egy kis házi képűjságot. Ezt akár BASIC-ben is meg lehetne írni, de van ennél egy sokkal kényelmesebb megoldás is, a TELETEXT V3.0, hiszen a program könnyen kezelhető, menüvezérelt, és a vele elkészített műsor minősége megegyezik a TV képűjsággal. Mint már mondtam a program menüvezérelt, így ezért a leírást is menükénti csoportokra bontjuk.

**Képszerkesztő:** Ebben a menüben az alábbi almenük közül választhatunk:

**Szerkesztés:** A legfontosabb rész a programban. Itt tudjuk az egyes oldalakat megrajzolni. Lehetőségünk van különböző betűméretre (F1+szám 1 és 4 között), különböző karakterkészletekre (F2, F4, F6, F8). Az alapszint az F3-mal állíthatjuk be (F3 után pedig CTRL vagy C = + szám). A módokat az F5 billentyű segítségével állíthatjuk be. Az 1. karakterkészlet

(F2) a 10, 11, a 2. (F4) 20, 21, a 3. (F6) 30, 31, és végül a 4. karakterkészlet (F8) a 40-, illetve a 41-es módban lehet. A páros számú módokban a betűk megegyeznek a billentyűzetten található karakterekkel, csak az alakjuk más, míg a páratlan módokban (11, 21, 31, 41) új karaktereket is használhatunk. A következő párok ezeket az átdefiniált karakterkészletet mutatják be. Az első karakter a billentyűzetten található meg, míg ennek párja az új betű: lá; 2é; 3í; 4ó; 5ö; 6ő; 7ú; 8ü; 9ű; 0C = ;+vessző; - két pont lent; fontjel két vessző lent; szorzás fa; hatványjel alma; kettőspont umlaut A; pontosvessző C betű bekarikázva; vessző idézőjel lent; pont idézőjel fent. F7-tel a szavakat villogtathatjuk, ami például kiemelésre szolgál. Kilépést pedig a RUN/STOP billentyű biztosít.

**FONTOS!!!** Minden képet, amit megrajzoltunk ki kell menteni, mert később csak úgy tudunk műsort szerkeszteni, ha összefűzzük őket.

**Bemutatás:** Az elkészült képet mutatja be.

**Diszkezelés:** Itt tudjuk a képeket ki-

menteni, betölteni, vagy egyéb műveleteket elvégezni.

**Nyomtatás:** Az adott képernyő tartalmának kinyomtatása. A központi menüben a *Műsorszerkesztő* almenüben tudunk a már kimentett képekkel egy műsort összehozni:

**Szerkesztés:** Beállíthatjuk, hogy a képek milyen sorrendben kövessék egymást.

**Listázás:** A már megszerkesztett műsorról kapunk egy kis információt, ami természetesen ki is nyomtatható.

A *Műsorsugárzó* menü segítségével már futtathatjuk is a képűjságot, de előtte beállíthatjuk a színeket, a dátumot, de még az időt is.

Az utolsó menü pedig a *Kilépés a programból*:

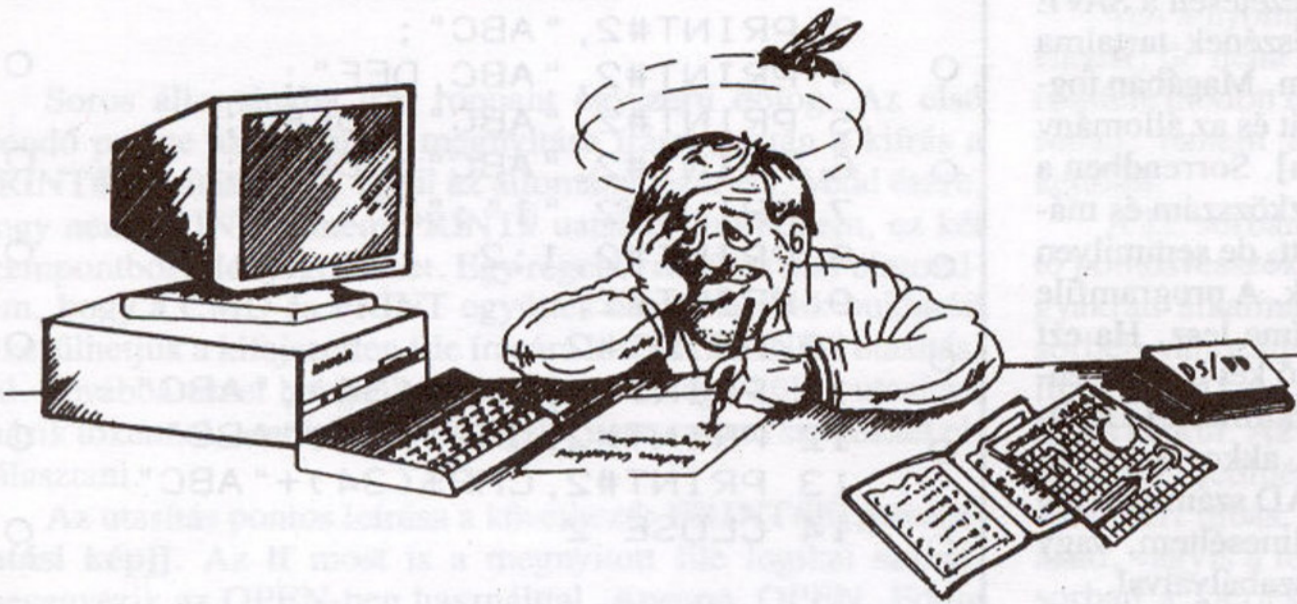
Bizonyára mindenki tudja, hogy mit jelenthet ez az opció. Kilépés előtt a program még kétszer megerősítést kér, de ha véletlenül mind a kétszer megnyomjuk az I-t, akkor se essünk pánikba, mert a RESTORE lenyomása után ismét visszakerülünk a főmenübe.

Ha viszont a RESET gombot nyomtuk volna meg, akkor sem törődik a programunk, hiszen a RESTORE segítségével ismét visszakerülünk a főmenübe (még akkor is, ha már egy programot betöltöttünk!!!).

Ennyit lehetett elmondani erről a PROFIL TELETEX-SZERKESZTŐRŐL.

Kun Szilárd

## FRIEND TWO CREW



## JURASSIC PARK

Steven Spielberg legújabb mesterműve hatalmas népszerűségnek számított az elmúlt évben. A film egyik szereplője John Hammond amerikai milliommós a világ legkiválóbb genetikusai és paleontológusai segítségével valóra váltja élete álmát, borostyánkőbe zárt dinoszauruszvérből génebézési eljárással újraalkotja a jurakorban élt őslényeket. A természetel azonban az ember sem tud szembe szállni, egy végzetes hiba során elszabadulnak a legveszedelmesebb dinoszauruszok... Spielberg magasfokú számítástechnikát használt a lények megteremtéséhez, mozgatásához és montírozásához. A film élvezhetőségét, izgalmasságát a Dolby Surround szisztemű hang is nagymértékben befolyásolja. Steven S. a következő szereposztásban készítette a J. P.-ot: Sam Neill, Laura Dern, Jeff Goldblum, Richard Attenborough. Élethűség, félelem, poén, stressz, fordulatosság jellemzi a munkát. A dinóláz már javában tombolt a Földön, amikor 1993 őszén Magyarországra is megérkezett. A mozik zsúfolásig megteltek, megjelentek a Jurassic emblémás kitűzők, képeskönyvek, játékok, pótlók. Vállalkozó szellemű egyének éttermüket, szórakozóhelyüket dinoszauruszfigurákkal színesítették.

A film megjelenésével egyidőben az Ocean szoftvercsoport megkapta a számítógépes verziók elkészítésének a feladatát és jogát. A program PC-re (4HD), Amigára (5DD, 5DD AGA), valamint más cégek által Segára, Gameboyra és Amiga CD 32-es konzolra is elkészült. Az

A500-as verzió jóval kevesebb színnel, gyengébb grafikával rendelkezik, mint az A1200-as AGA formátum. Ez főleg a 3D-os részeknél vehető észre. Az alkotók kénytelenek voltak a zenét is mellőzni néhány pályán. A szinteken hallható zenét, amelyek hűen tükrözik az adott pálya hangulatát és a hangeffekteket, Dean Evans készítette.

Elterjedt jónéhány rossz AGA verzió, a hiba általában a 3D-os részeknél jelentkezik, a program változó időközönként egyszerűen kifagy. A másik jellemző hiba a megszerzett kódok beírásának a lehetlenségén alapszik. A PC verzió majdnem megegyezik az 1200-aséval.

A játék két fő pályatípusból épül fel, egyik az ún. izometrikus, ahol felülnézetből látjuk a területet. Ezeken a szinteken kisebb-nagyobb feladatokat kell elvégezni. A másik típusú pályán, amivel először a harmadik részben találkozunk, 3D-os szemszögből látjuk a folyosókat. Ezután a látogatóközpontban, a katakombákban már végig 3D-os pályákon játszhatunk. A játék története Michael Crichton novellája alapján íródott, azonban megértéséhez elegendő belelapozni a film forgatókönyvébe is.

A cselekmény dr. Alan Grant és felesége régészeti helyszínéről indul, ahol ősi dinoszaurusz csontokat kutatnak. Itt jelenik meg John Hammond, aki a kutatásokat támogatja. John-nak van egy kutatócsoportja, akik szenzációs eredményt értek el. Borostyánkőben a fagyantába ragadt szúnyogokból dinó vért szívtak ki.

Hogyan került a szúnyog a gyantába? Erre a kérdésre egyszerű a válasz. A szúnyog rárepül a dinoszauruszra, a vérből szívogat, majd balszerencséjére véletlenül lecsorgó fagyantába ragad. A fa kidőlt elkorhad, a gyanta viszont megőrizte a szúnyogot. Ezeket a kis gyantacseppeket, a borostyánkőveket a bányászok megtalálták és eljuttatták Hammond csoportjához. A borostyánkővek a föld alatt több millió évig pihentek, a dinó vérben a DNS láncokon ez idő alatt hiányosságok keletkeztek, ezeket a lyukakat béka DNS-sel pótolták. A kutatások színhelyén a Velociraptor átszállítása közben egy ember meghal.

Hammond a tragédia tuán felkereste dr. Alan Grant, híres paleontológust és felkérte, hogy vizsgálja meg az őslények birodalmát. Grant, felesége, egy stresszkutató és Hammond az ügyvédjével elindultak a Costa Ricához közel egy szigetre, hogy megnézzék a Jurassic Parkot. A látogatóközpont felé menet lélegzetelállítva nézték az ősi növényevőket. Hammond a központban részletes tájékoztatást adott a parkról és a kutatási eredményekről. Bemutatta két unokáját, Tim-et és Lex-t. A park biztonságára néhány számítógépező ügyelt. Hammond hogy részletesen bemutassa vendégeinek a dinoszauruszokat, két robotvezérrel autóba ültette őket és unokáit. Azonban egy édesszájú, pénzsovár számítógépező, Dennis Nedry egy Dobson nevű kereskedő megbízásából jó pénzért megpróbál dinó embriókat a parkból kicsempészni. Ehhez viszont kénytelen volt kikapcsolni az egész számítógépes rendszert. Amíg a túrakocsik kint voltak, Dennis megpróbál a parkból kiiskoljni. Balszerencséjére (eső, kidőlt útjelző) eltévedt, és gazdag ember helyett egy Dilophosaurus zsákmányaként végezte. Közben Grant, Hammond és unokái és kísérőik a parkban autóztak, amíg a masinák le nem álltak. Épp a Tyrannosaurus kerítésénél jártak, amikor Dennis áramtalanítása életbelépett. Az elszabadult T-Rex támadásait sikerült valahogy elhárítaniuk. Az ügyvédet kivéve, életben maradtak. A T-Rex az autót Tim-mel együtt a szakadékba lökte. Lex és Grant leereszkedtek hozzá. Tim megmentése után elindultak a látogatóközpont felé. Tulajdonképpen ekkor kapcsolódunk a történetbe. A filmmel ellentétben a novellában csak a két gyerek indul vissza, vagyis csak ők tűnnek el.

A játékban az első pályán tehát a Tyrannosaurus Rex kerítése előtt állunk. Ezen a pályán néhány egyszerűbb feladatot kell teljesítenünk. Keressünk egy kameratartó oszlopot, és kapcsoljuk be az alatta lévő számítógépet. Ez az Enter billentyűvel lehetséges. A megjelenő menük

közül válasszuk ki a Paddoc System-et (3-as). Nyissuk ki a bunker ajtaját, Bunker Door Lock (1-gyes). Ugyanitt információt kaphatunk a szinteken mozgó dinókról, valamint a főbb objektumokról. A bunker a pálya (vagy a térkép) bal alsó részében található, ahol található egy szerszámosládát. Balra egy új fegyvert, jobbra Tim-et láthatjuk. Középen egy üregben feszítsük fel a csatornanyílás rácsát a szerszámokkal. Másszunk be. Keressük meg az ID kártyát (rögtön balra), majd a faládát (a kártya felett, a bejárattól a második elágazás balra). A ládát lökjük be a szennyvízbe, majd ússzunk vissza a ládával együtt a bejárathoz. Jobbra az első elágazásnál a hosszú folyosón ütközésig toljuk a ládikót, majd fel koppanásig, jobbra, fel, jobbra, fel, és balra. Itt üldöggél Lex. Tolvaj a ládát közvetlenül elé, akkor felkel és ráül a szállítóeszközre. Az előző útvonalon vigyük vissza őt a bejárathoz (kijáráshoz). Másszunk ki a nyíláson. Tim és Lex így már megkerült, az ID kártya segítségével a jobb felső kapun távozhatunk a pályáról.

Utunk a Triceratops-ok területére vezetett. A Triceratops békés növényevő, méretei az elefántéhoz hasonló, fejéből kétoldalt és elől szarvak állnak ki. Ránk különösebb veszélyt nem jelentenek, maximum összetaposhatnak. Az előző pályán, és most is felszedhetünk néhány tárgyat: a mentőláda energiát, a fegyveresláda lőszert, vagy egy új fegyvert ad. Az esetleg összegyűjtött fegyverek között az I billentyűvel válogathatunk. Célszerű az elektromos nagyfeszültséggel működő pisztolyt használni. Az F gombbal pedig a játékot lehet megállítani (pause). Tehát a Triceratops-ok területén vagyunk.

Gyűjtsünk össze annyi bogyót, amíg a bal alsó sarokban megjelenő ikon telítettséget nem jelez. Balra haladva a kidőlt kerítés után felül található egy döglött Triceratops-t, tőle jobbra egy élőt láthatunk. Fussunk lefelé. Ha elegendő bogyót gyűjtöttünk, a dinó egy kis ideig azt eszegeti, így gyorsabban eliszkolhatunk a jobbra vezető úton.

Ismét jobbra haladva még egy Triceratops-szal találkozunk. Vegyük fel az összes mentőládát. Irtsuk ki a sziklákról köpködő dinókat, majd ugorjunk le. Alul egy kőfal zárja el az utat. Álljunk elé, ha a dinoszaurusz támad mindig ugorjunk el a megfelelő ütemben balra, vagy jobbra. Rövid idő után nagytestű ellenfelünk kitöri a falat. Szaladjunk lefelé és másszunk ki a hasadékon. Balra találunk egy ID kártyát, segítségével a jobboldali bunkerén keresztül távozhatunk a pályáról. A bunker folyosóin az előzőktől eltérő 3D-os grafikával találkozunk. Térképet az M billentyűvel használhatunk. A bejárattal szemben felül automata ajtón léphetünk egy kisebb csarnokba. Rövid töltés következik, a megjelenő ajtó egy új rendszerbe vezet. Alul két labirintust találunk, a baloldali végén a kijárat van. Egy hatalmas növényevő a Stegosaurus területén vagyunk. Ezek a dinók csak farkuk csapkodásával okozhatnak kisebb energiavesztést. A pályán mozgatható köveket kell ide-oda tologatnunk, és a megfelelő helyekre lökdösnünk. A kőtologatás közben Pterodactylusok súlyos köveket próbálnak a fejünkre dobni. Vigyázzunk, hogy a sziklákról véletlenül se ugorjunk le.

A következő szinten a Gallimimusokkal találkozunk. Első feladatunk a balra lévő fennsík végéhez gyalogolni. Ott egy

hordót találunk. Lökjük le függőlegesen a szikláról. Kerüljünk egy nagyot. Átérve a hasadékon fáklyákat találhatunk. Néhány pillanat múlva egy veszedelmes szörnyeteg, a Tyrannosaurus (az egyik legnagyobb húsevő, zsákmányát egészben lenyeli) fog ránk vadászni. Amikor vészesen közeledik, dobjunk a hátunk mögé egy égő fáklyát. A dinó néhány másodpercre megtorpan (a T-Rex csak a fényt, a mozgást érzékeli). Utunk folytatása lehetetlen lenne, ha a mélyedésbe előzőleg nem gurítottuk volna bele fenntről a hordót. Másszunk át rajta, majd egy fáklyát dobjunk a kiömlött olajra. Keressük meg az ID kártyát, és engedjük ki a katakombából Hammond unokáit, majd balra távozunk.

A Dilophosaurusok területén vagyunk. (Dilophosaurus húsevő őslény, áldozatát a száján keresztül kilövellő méreg bénítja meg. Méricskél.) Bújunk át a kerítés alatti lyukon, majd nem messze ússzunk át a folyó másik oldalára. Ülünk be a gumicsónakba, és gyorsan evezünk le a folyón. A Pterodactylusoknál a nagyobb vízeséshez közeledve ugorjunk ki a gumicsónakból, majd ha átért ismét szálljunk vissza. A Brachiosaurusoknál megtaláljuk Dennis Nedry terepjáróját, és az embriókat, valamint egy ID kártyát.

A pályát két helyen hagyhatjuk el. Célszerű a balalsó számítógépes vezérlésű kapun távozni. Ha a főkapunál vagyunk Tim-mel és Lex-szel, menjünk a látogatóközpontba. Most talán a játék legszebb 3D-os része következik. A film forgatókönyvében tulajdonképpen Hammondék ekkor már mindent áramtalanítottak, ezzel elszabadultak a Velociraptorok. A számítógépes rendszer újraindításakor a központi biztosíték lecsapódott. Nekünk csak annyi a feladatunk, hogy megtaláljuk a főkapcsolót, és azt visszakapcsoljuk. Ezután valahogy el kell jutnunk a többiekhez. A Velociraptorok támadásaira számíthatunk. (A Velociraptor a valaha élt legveszedelmesebb dinoszaurusz. Intelligenciája igen magas, gyorsan fut. Karmai elérhetik a 15 cm-t is.) Az M billentyűvel egy térképet hívhatunk le. Keressük meg a gépház legfelső részében az infrás szemüveget, ezzel az alsó szinteken is jól láthatunk a sötétben. A térkép segítségével keressük meg a többieket. A játék végén egy rövid befejezést láthatunk. Könnyítésül ismertetünk néhány pályakódot: A1200-as AGA verzió második pálya B5A48352, a harmadik E54C67AA. A500-as verzió 8EB75C3D, DE5FB8C5, EEE7740D. A program megrendelői most Jurassic Parkos matricát kapnak ajándékba.

F. Péter



# KIADVÁNYSZERKESZTÉS C64-EN, PC-N



tetszetős grafikákkal és feliratokkal is díszíthetünk.

A következő bemutatásra szánt program a Banner. E programmal transzparenszeket készíthetünk. Fantasztikus karakterkészlet és ötletes jelölések állnak rendelkezésünkre. A méreteket kicsinyíthetjük, illetve nagyíthatjuk, s egy normál állapotban elkészült transzparens hosszát is módosíthatjuk, ami leporelló papíron kb. 5–6 méter is lehet.

A következő téma az újságszerkesztés lenne. Erre szintén van egy nagyszerű program, ami C64-en a Newsroom, PC-n pedig a News címet kapta.

A két program lényegében ugyanazt tudja, viszont a PC-s programban használhatók az ékezetes karakterek is. E programok elvégzik a szöveg teljes tördelését, fejléceket szerkeszthetünk, s választhatunk a többszáz grafikából is, amelyeket a szerzők készítettek.

A szövegeket hasábonként kell megszerkeszteni, majd az elkészült hasábokat össze kell fűzni, s ki lehet nyomtatni az elkészült újságot.

Léteznek ezektől komolyabb kiadványszerkesztő programok is, például a Ventura vagy Windows alá az Aldus PageMaker, de ezekről a hely hiánya végett már nem írhatok.

Remélem, e kis leírással mindenkinek kedvet csináltam a számítógépes kiadványszerkesztéshez.

Kun Szilárd

Manapság már a tesztoleges kivitelű reklámlakátok, hirdetések és még az újságok szerkesztése is elképzelhetetlen lenne számítógépek nélkül. A tetszetős grafikákkal ellátott meghívók, üdvözlőkártyák is számítógéppel készülnek. Sokak talán azt gondolják, hogy ezeket csak bonyolult számítógépekkel és kimagaslóan jó minőségű lézernyomtatókkal lehet csak megvalósítani. Ők nagyon is tévednek, hiszen ilyen kiadványokat (ha a minőségük nem is vetekszik a komolyabb gépeken készütekkel) házi körülmények között is meg lehet valósítani. Még a legegyszerűbb számítógépekkel is (például Commodore 64) tetszetős nyomtatványokat készíthetünk. E leírásban a Commodore 64 és az IBM PC ezen képességeit mutatjuk be, s megpróbáljuk e két gép kiadványszerkesztő programjait összehasonlítani.

A legtöbben talán a Print Shop-ot, és a Print Master-t ismerik, hiszen e programcsomag mindkét gépre megjelent, s a PC-s verzió kiegészült egy Print Banner nevű, külön programmal is. E programokkal üdvözlőkártyákat, nagybetűs feliratokat, fejléceket, transzparenszeket, posztereket, naptárakat vagy éppen cégtáblákat nyomtathatunk. Lehetőségünk van különböző keretek választására, ami körül fogja kiadványunkat.

Ha a keretet kiválasztottuk, akkor jöhet a megfelelő karakterkészlet kijelölése is. Ha ez is megtörtént, akkor pedig, már

írhatjuk is a szöveget. Sajnos a programnak van egy hátránya is, mégpedig az, hogy nem használhatunk ékezetes karaktereket. Ha a szövegbevitel megtörtént, még válogathatunk a grafikák közül is, de ha nem tetszenek a gyártó által készített ötletes és humoros rajzok, akkor rajzolhatunk mi magunk is.

Ha naptárt akarunk készíteni, akkor meg kell adni az évet és a hónapot, s a programban található beépített öröknaptár már el is készítette naptárunkat, amit



# AZ AMIGA CSALÁD TÖRTÉNETE

A Commodore számítógépek fejlődésében óriási mérföldkönek számított az A1000-es megjelenése. Ez az első Amiga, amely új formátumával, „nagy” memóriakapacitásával, kiemelkedő zenei és grafikai teljesítményével a Commodore 64-es után hatalmas előretörést mutatott. Sajnos nem igazán terjedt el az igen borsos ára, és nagy vetélytárs az A500-as térhódítása miatt. Napjainkban Magyarországon csak néhány száz darabot találhatunk, a készülék értéke 40–60 ezer forint között mozog.

Az A1000-es winchesterrel, külön klaviatúrával, és 3.5 lemezmeghajtóval rendelkezik. A számítógép alapját egy Motorola 16 bites 68000-as mikroprocesszor képezi. A gép órajele, ami a gyorsaságra utal, 7.14 MHz. RAM memóriája 256 kb-ot tartalmaz egy ún. WOM-ot (Write Only Memory), amely a RAM és a ROM kombinációja.

Üzembehelyezése után lemezről tölthető az Amiga működéséhez szükséges Kickstart rendszerszoftver. Ez a Kickstart 1.1-es változata.

Az előzőekben két rövidítést használtunk, a RAM (Random Access Memory) szabad tárolóhely, a ROM (Read Only Memory) gyárilag telített memória, az alapműködéshez szükséges rutinok helye.

Az 1000-es pályafutása hamar kettőtört, amikor elkészült az Amiga 500-as számítógép. Billentyűzete együtt van a gép fontosabb részeivel. Ugyancsak 3.5-ös meghajtóval és 68000-as mikroprocesszorral rendelkezik. Órajele 7.14 MHz. Memóriája az A1000-eshez képest emelkedett, 512 kb-ot tett. Ebbe az Amigába a Kickstart már be van építve a ROM-ba, vagyis a gépben kikapcsolás után is jelen van. 1985–87 között 1.2-es, majd 1.3-as Kickstarttal került forgalomba. 8 bites sztereó HI-FI hangja, 4096 színből álló színekészlete van. Operációs rendszere ikonok, ablakok segítségével több program egyidejű futtatását teszi lehetővé. Az A500-as a PAL szisztémával teljesen kompatibilis. Ez a gép már rendelkezik blitterrel, ami nagy előnyt jelent az animációknál, valamint a gyors grafikus mozgásoknál. Az első Amiga számítógépek közül az 500-as méltán lett a Commodore 64-es véglegesen kikiáltott utóda. A Commodore GMBH eladási listáján a második helyen áll, ezt a helyet napjainkban egy jóval korszerűbb Amiga számítógép kezdi átvenni.

Fejlődési sorrendben az A2000-es következik. Sokan nevezik az 500-as

nagytestvérének. Képességét tekintve tulajdonképpen megegyezik elődjével. Felépítése a PC-khez hasonlóan moduláris, nagy mértékben bővíthető, és kiegészíthető. RAM memóriája 1 Mb-ot tartalmaz. Központi processzora az 500-assal megegyező. A gép négy kártyával bővíthető, az egyik kártyahelyen PC/XT modult kapcsolhat az alkotók, ezért ezek a készülékek PC-ként is működtethetők. A számítógép „dobozá” egy 5.25-ös lemezmeghajtót tartalmaz. A multitasking (párhuzamos) üzemmód is a 2000-es jellemzője. Egyszerre, párhuzamosan több feladatot képes elvégezni, egyidőben jelen lehet egy IBM és egy Amiga képernyő is. A számítógép egymás mellett tud PC és Amiga programokat futtatni.

A 2000-es követője az Amiga 2500 UX, amely elődje továbbfejlesztett változata. Ez már 68020-as processzorral készült, órajele 14.3 MHz. Az Amiga rendszerszoftverén kívül az ún. Unix rendszert is beépítették, ezért az Unix programok is futtathatók. RAM memóriája 3 Mb-ot tartalmaz. A készüléket turbókártyával is felszerelték.

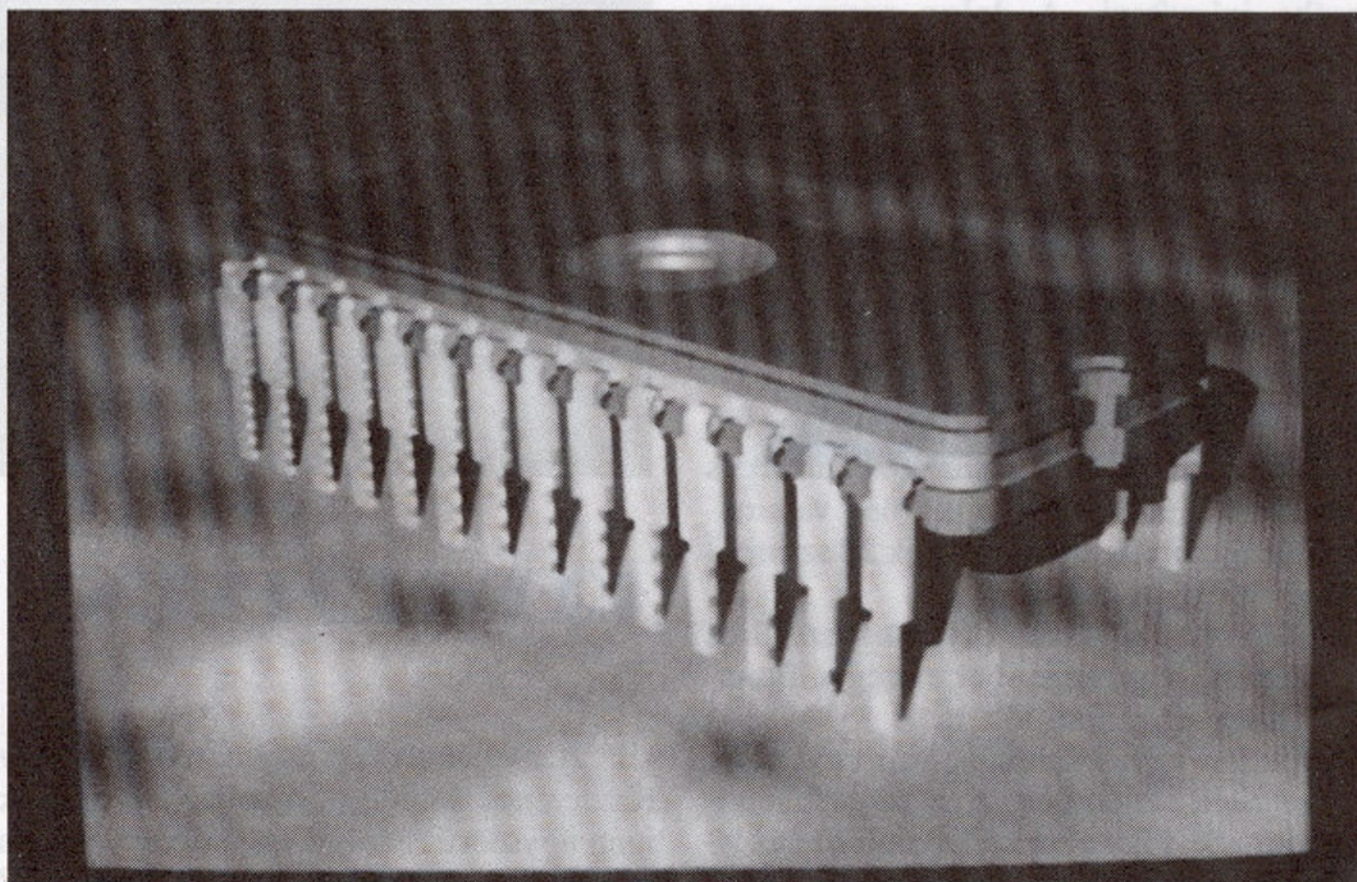
A fejlődés itt sem torpant meg, elkészült az Amiga 500+, az A600 és az A3000-es számítógép. Ezek a gépek az új 2.0-ás operációs rendszerrel és az ECS (Enhanced Chip Set) csomaggal lettek felszerelve. Az ECS lehetővé tette a nagyobb képfrekvenciát, a remegéstől mentes Multisync, VGA monitorok használatát, valamint a SuperHighRes dupla felbontású vízszintesen 1280 pixeles képernyőt.

Az A500+ az A500-as kiegészített változata. RAM memóriája 1 Mb-ot tartalmaz.

Az A600-as egy kisebb, olcsóbb számítógép. Memóriája ugyancsak 1 Mb-ot tartalmaz. A készülék egy IDE hardisk vezérlőt tartalmaz. A gépben a Motorola processzor felett akár egy 210 Mb-ot tartalmazó AT hardisk is elhelyezhető. Bővíthető még az alaplapra, és a PCMCIA csatlakozó segítségével is. Az A600-as 1991-es év terméke. Két kisebb hátrány, a numerikus billentyűzet hiánya, valamint 80%-os kompatibilitás jellemzi. Kickstartja 2.05-ös. A televízióval rendelkezők örömeire RF TV modulátorral is rendelkezik.

Az Amiga 3000-es számítógépet a Commodore cég nem igazán otthoni célokra tervezte. Egyik kiemelkedő újonság a kis méret. A készülék két változata ismert, az elsőben Motorola 32 bites 68030-as processzor, és 6881-es matematikai koprocesszor található. Órajele 16 MHz. A második 68030-as mellett 6882-es koprocesszorral rendelkezik. Órajele 25 MHz. Az A3000-es 1 Mb-ot tartalmazó chip és fast memóriával dolgozik, vagyis RAM-je 2 Mb-ot tartalmaz. A chip RAM az alaplapra 2 Mb-ig, a fast RAM a belső rekeszben 16 Mb-ig bővíthető. A beépített 3.5-ös meghajtón kívül még egy drive számára biztosított helyet. Egy SCSI hardisk vezérlőt is találhatunk az alaplapra, ez 32 bites DMA (Direct Memory Access) hivatkozást tesz lehetővé.

A gép winchestere 50 Mb-ot tartalmaz. Az A3000-es rendelkezik egy ún. Display



Enhancer-rel, ami a monitorképért, remegésmentességért felel. A készülékhez csatlakoztathatunk egy 9 pólusú monitor dugaszt, és egy 23 pólusút is. 9 pólusú csatlakozót PC VGA monitorok használnak. A 3000-es négy vízszintes helyzetű slottal rendelkezik, ahová AT kártyákat, 16 bites hangkártyákat, 24 bites grafikai videokártyát, vagy akár Macintosh Quadra 68040-es processzorkártyát helyezhetünk. A gép „doboz” asztali és torony kivitelben is kapható. Az Amiga 1200-as és az Amiga 4000-es a mai két legkorszerűbb személyi számítógép. A Commodore 1992–93-ban kidolgozta az AA chip-et és az AGA grafikát. 24 bites, 16.7 millió színekészletből 256 színt jeleníthetünk meg egy azon időben. A HAM8-as rendszerrel akár a 16.7 milliót is. A digitalizált képek, és ezek animált mozgása tökéletes. A fotók fénykép minőségűek.

Az A1200-es otthoni célokra lett kifejlesztve, AA chip készlettel, IDE hardisk vezérlővel, PCMCIA csatlakozóval van felszerelve. A gép egy 68EC020-as mikroprocesszort és 68881/68882-es lebegőpontos koprocesszort tartalmaz, amit a Motorola készített. Órajele

NTSC esetén 14.32 MHz, PAL esetén 14.19 MHz. ROM-ja 512 Kbájt. RAM memóriája 2 Mbájt. A Chip RAM grafikus és zenei adatok tárolóhelye, amely az A1200-es esetében 1,2 Mbájttal bővíthető. Bővítési lehetőség a 32 bites 1 Mbájtnál az alaplapon lehetséges. 32 bites 2 Mbájtos bővítőt a készülék belsejébe rakhatunk.

A Fast RAM, amit a 68EC020-as processzor tud csak elérni, 8 Mbájtig bővíthető. 4 Mbájtig 32 bites RAM-okkal a CPU slotban, a következő 4 Mbájt 16 bites RAM-mel a PCMCIA csatlakon.

További bővítési lehetőségek: A gépben az alkotók helyet biztosítottak egy 2.5-ös belső merevlemez számára. CD-ROM, külső merevlemez (winchester) csatlakoztatási lehetőség. Az 1200-as IDE (Intelligent Drive Electronics) merevlemez kontrollert tartalmaz.

Az A1230-as turbókártya és memóriabővítő 1993-ban került piacra. Ez 40 MHz-es 68030-as mikroprocesszort és 68882-es matematikai koprocesszort tartalmaz, 1 Mbájt Fast RAM-mal. Bővíthető 32 Mbájtig.

Az Amiga 4000-es Macintosh Quadra munkagép asztali, vagy torony kivitel-

ben. A számítógép 25 MHz-es Motorola 68040-es processzort tartalmaz. Olcsóbb kivitelben 68030-as processzorról is kapható. Ezek a mikroprocesszorok cserélhetőek. Az olcsóbb konfiguráció 68030-as processzorról, 120 Mbájtos winchesterrel, valamint 4 Mbájt RAM-mel rendelkezik. Ára a forgalmazónál 209 000 forint. A Commodore cég első, és az „Amiga család” legutolsó tagja a CD Amiga 32-es, játékgép. 1993-ban a világ legjobb konzolgépe. A készülék egy MT68020-as mikroprocesszort, AA chipkészletet, AGA grafikát, DSP chipet (a kép és a hang egymáshoz szinkronizálásához), PLANAR chipet, 2 Mbájt chip RAM-ot rejt. 32 bites konstrukció. Az egység megközelítőleg A/4-es méretű. A CD 32-es minden számítógépkedvelő álmát egyszerre megvalósította. Lehetővé tette a folyamatos, élethű mozgás kivitelezését. Idén 1994-ben a CD 32-es kiegészítőit is elkezdik gyártani. Az A5000-es és az A1400-es még sokat várat magára. Sajnos a Commodore GMBH bizonyos problémái miatt ezek a készülékek csak gondolatban léteznek.

Folk Péter

## NOVA

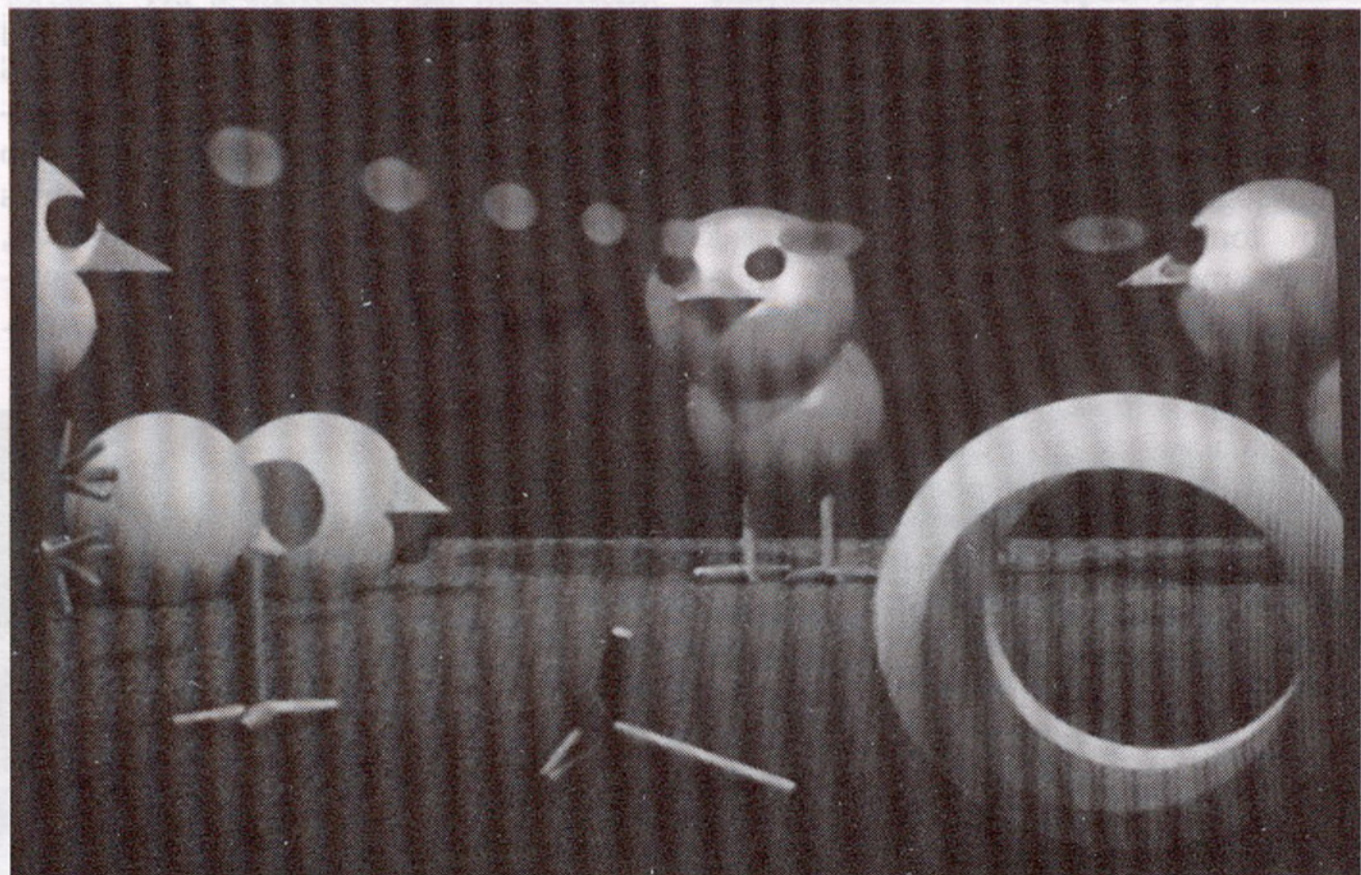
Ez a logikai program egyszerűen kiváló. A feladat a következő. Egy „bigyó” lézert fényt bocsát ki magából. Nekünk ezt a fényt kell különböző prizmákkal, lencsékkel és egyéb dolgokkal a gyűjtőlencsébe irányítani. A dolog persze nem ilyen egyszerű.

A fény nem keresztezheti önmagát, s nem mindig egy darab gyűjtőlencse várja a fényt. A felhasználható „szerszámok” is korlátozva vannak, sőt még a feladat megoldásához szükséges lépések is.

Nagy előnye a programnak, hogy nincs benne idő és élet. Addig játszunk, amíg van bennünk erő és kitartás újratekdeni egy elrontott szintet. Ha kíváncsiak vagyunk a fény útjára mielőtt bármit is csinálnánk, akkor nyomjuk meg valamelyik funkcióbillentyűt. A játék 28 szintből áll és végigjátszható.

Néha a megoldás olyan egyszerű és szellemes, hogy az ember elszégyelli magát, hogy erre nem jött rá.

Minden egyes szintnek van egy neve. Ezt a játék elején beírva az adott feladathoz jutunk, s így nem kell nyüglődni az ismerős szintekkel.



## KÉPREJTVÉNY

A mellékelt kép egy demo valamely részletének fotóját ábrázolja. Kérdésünk igen egyszerű. Melyik ez a demo, vagyis mi a neve, és melyik csapat készítette?

Megfejtéseket a rovat levélcímére lehet feladni:

FRIEND TWO CREW,  
KÉPREJTVÉNY

BUDAPEST, POSTAFIÓK 701/55.

A huszadik helyes megfejtő MARKSMAN márkájú formatervezett, digitális asztali órát kap ajándékba, amit postán fogunk elküldeni.



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

C-64-re legjobb lemez és magnós programok eladók. Iszonyúan olcsó minden, nem érdemes kihagyni! Érdeklődni levélben. Válaszborítékot küldjetek. Bagi Péter, 1204 Budapest, Maros u. 168.

Amiga 500/500+ -ra egér eladó! Irányár 1100 Ft. Érdeklődni személyesen vagy levélben. Bagi Péter, 1204 Budapest, Maros u. 168.

Régebbi C= újságok, összes 576, COV-ok, COV évkönyvek, CHIP, HETI-CHIPEK, Guruk stb. folyóiratok eladók, NES konzolkártyákra cserélhetők! Válaszborítékért lista! Más értelmes csere is érdekel! Írj feltétlenül! Seres Zoltán, 6723 Szeged, Építő út 5/a. II/6.

Programéhes Amiga 1200-as és 500-as tulajdonos leveleit várom. Gyurkó Zsolt, 4031 Debrecen, Krónikás u. 6. IV/40.

Keresem a C=64 „C” nyelvet floppy-n. Dr. Zsigmond Géza, 1086 Budapest, Lujza u. 22., tel.: 133—21—26.

C-16, +4-re programokat adok—veszek—cserélek 5 Ft/db. Válaszboríték! Soós Viktor, 7355 Nagymányok, Katona u. 11.

C-64-re! nyáktervező és elektronikai programok, poke- és GEOS-lexikon. Válaszborítékot kérek. A készlet korlátozott! Varsányi, Szombathely, Nagy László u. 11.

Eladó C-64-hez egy hang-digitalizáló, 2 db eprom-bank, továbbá 2 db játék Cartridge. Ugyanitt programcsere lemezen. Pázsit Tibor, 5650 Mezőberény, Várkonyi u. 1.

C-64 programok cseréje, eladása lemezen. Cseh Balázs, 8636 Balatonöszöd, Szabadság út 112.

Otthon végezhető papírmunkát ajánlok! 30 Ft-os bélyegért tájékoztató. Soós Viktor, 7355 Nagymányok, Katona u. 11.

C-64-re felhasználó, oktató, könyvelő, játékok 5 Ft-tól. Borítéklista. Földesné, 5000 Szolnok, Gyórfy J. u. 12. Tel.: 56/420-544.

C-64 monitorral, floppyval újszerű állapotban eladó. Ára: 18 000 Ft. Kóródi Csaba, 6050 Lajosmizse, Erkel Ferenc utca 1., tel.: 76/356—901.

C-64-re már 3 Ft/db ártól eladó játék-, felhasználói- és egyéb programok lemeze és kazettára. A választék óriási. Válaszborítékért listát küldök. Kovács Krisztián, 6750 Szeged-Algyő, Bartók B. u. 6.

Kedves olvasónk!

A1000, A500, A500+, A600, A1200, és egyéb Amiga számítógépekre, valamint C64, C128-re játék- és felhasználói prg-k, valamint PD demók megrendelhetők. Vízhatlan csomagolás, kedvezmények, számla. Bélyeggel ellátott válaszborítékért részletes tájékoztatót és programlistát küldünk. Amigán a lista mágneslemezen is kérhető egy 3.5-ös lemez, és 30 Ft-os bélyeg ellenében. A borítékokat géptípusonként a következő címekre lehet feladni:

AMIGA: FRIEND TWO CREW, 1399 Bp., Pf.: 701/55. Tel.: 135-12-89. (Ügyelet naponta 16—20 óra között)

C-64: BUMBLE BEE, 1116 Bp., Sáfrány u. 44. Tel.: 162-77-24.

Ha kivágta a megrendelőlapot, kérjük tegye felbélyegzett borítékba, címezze meg, és úgy küldje el.

Köszönjük!

## IGEEEN! Megrendelem az ACOMP Kft.

borító hátoldalon szereplő kínálatából a feladó címére postai utánvétellel az alábbi termékeket:

..... db	.....	Ára .....	Ft/db: összesen .....	Ft
..... db	.....	Ára .....	Ft/db: összesen .....	Ft
..... db	.....	Ára .....	Ft/db: összesen .....	Ft
..... db	.....	Ára .....	Ft/db: összesen .....	Ft
..... db	.....	Ára .....	Ft/db: összesen .....	Ft
..... db	.....	Ára .....	Ft/db: összesen .....	Ft
			Mindösszesen: .....	Ft

Tudomásul veszem, hogy a csomagküldés átfutási ideje 2-4 hét.  
50.000,- Ft feletti megrendelésre csomagküldést nem vállalunk!  
Az áraink a postaköltséget nem tartalmazzák!

## NÉGY MATEMATIKUS

## KERESZTREJTVÉNY

A megfejtéseket soronként adjuk!

**VÍZSZINTES:** 1. Ő írta „A magyarországi matematika története” című munkát. 2. I. — Alig áll a lábán. 3. Nagy Vince monogramja. — Ö. O. — Tam keverve. — Nádor Magda monogramja. — Tantál vegyjele. 4. Héjas gyümölcs. — Aszalt gyümölcs. — Gléda. 5. Zsalu. — Regény fajta. — Folyó. 6. Létezett. — Három különböző magánhangzó. — Duna menti város. 7. Kicsinyítőképző. — Füttyülő. — Nagy madár. — Kettős betű. 8. Megehető, jó. — Vadállat. 9. Magyarországi és olaszországi gépkocsik jelzései. — Helyhatározó. — Sav igéje. — Már egyszer szerepelt ez a szó. 10. Hangnem. — Kopasz. — Létezik ilyen tár is. 11. Sérülte fia, Dávid párthíve volt. — Gyümölcs. — Beszél. 12. Három egyforma betű. — Napszak. — Pusztít. 13. Névelők. — Férfinév. — A dögvész istene. — Fél tánc. 14. Vihar és más természeti elemek istene. — Madridi park. 15. Több tankönyv írója, Apáczai Csere János-díj tulajdonosa.

**FÜGGŐLEGES:** 1. Matematikai Kutató Intézet munkatársa. 2. Virág fajta. — Női név. 3. Fordított mutatószó. — Tüzet szüntet. — Végtelen művészet fajta. — Kétté (!). 4. Lángoló. — Dunántúli csatorna. — Kanalaz. 5. Ilyen elem is van. — Minden élőlény anyja, ő volt az, aki a vízözön után újrateremtette az emberiséget. — Van ilyen medve is. 6. Két magánhangzó. — Fordított kicsinyítőképző. —

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1			Z	E	N				B	A	R	N	A		
2			E	G	Y	I			I	N	O	G			
3	N	V		Ö	O		A	M	T		N	M		T	
4	D	I	O		M	A	Z	S	O	L	A		S	O	R
5		O	L			K	R	I	M	I			A	J	
6	E	L	T						I				V	A	C
7	K	A		S	I	P			S	A	S		S	Z	
8			F	I	N	O	M		A		A				
9	H	I		O	T	T			M	A	R		S	Z	
10	A	L	T				T	A	R				S	Z	Ó
11						B	A	N	A	N		S	Z	Ó	L
12					M		E	S	T				I	R	T
13	A	A			O							A		T	A
14			T		S		B		P			S	Ó		N
15			T		Ó	R	M					Z	L		

Feladó: .....

.....

.....

**ACOMP Kft.**  
Csomagküldő Szolgálat

**B u d a p e s t**

XIV. Álmos vezér park 20.

**1141**

Autó alkatrész. — Szeszes ital. 7. Szép Ernő egyik műve. — Kevert ital. — Berényi Miklós monogramja. 8. Jahve anyala, az Éden őre. — Az igazság és méltanyosság pártfogója. 9. Mosószer márka. — Visszatár (!). — Prazeodímum vegyjele. 10. Bizmut vegyjele. — Lítium vegyjele. — Kén és amerícium vegyjele. — Több homoksivatag a Szaharában. 11. Női név. — Szín hírnöke, ő bocsát álmat az emberekre. — ... Tivadar, színművész volt. 12. Omladék. — Vidéki vendégmarasztaló. — Ütőkártya. 13. Nagy Gábor monogramja. — Gyomor termeli. — Dohányzik. — Állati szállás. 14. Tyúk termék. — Hintett. 15. Grünwald-díjas matematikus.

(A rejtvényben a rövid és hosszú ékezetek között nem teszünk különbséget.)

A rejtvényt nem kell beküldeni, mert szórakoztató jellegű!

Mokos István





## NÉGY MATEMATIKUS

A megfejtéseket soronként adjuk!

**VÍZSZINTES:** 1. Ő írta „A magyarországi matematika története” című munkát. 2. I. — Alig áll a lábán. 3. Nagy Vince monogramja. — Ö. O. — Tam keverve. — Nádor Magda monogramja. — Tantál vegyjele. 4. Héjas gyümölcs. — Aszalt gyümölcs. — Gléda. 5. Zsalu. — Regény fajta. — Folyó. 6. Létezett. — Három különböző magánhangzó. — Duna menti város. 7. Kicsinyítőképző. — Fűtyülő. — Nagy madár. — Kettős betű. 8. Megehető, jó. — Vadállat. 9. Magyarországi és olaszországi gépkocsik jelzései. — Helyhatározó. — Sav ígéje. — Már egyszer szerepelt ez a szó. 10. Hangnem. 11. Létezik ilyen tár is.

**KE**ADE SZERVÍZ Kft. az alább felsorolt szervízeiben szervízszolgáltatás munkadíjából 10% kedvezményt biztosít egyesületi tagoknak.

	1	2	3
1			Z
2			E
3	N	V	
4	D	I	C
5		O	L
6	E	L	T

pest, Szigony u. 8. Tel.: 1343-153  
 Zolt, Fazekas u. 1-3. Tel.: 46/321-488  
 scsaba, Bartók B. u. 37. Tel.: 66/327-195  
 ed, Csongrádi sugárút 76. Tel.: 62/493-185  
 nbathely, Szalonak u. 31. Tel.: 94/314-519

Értékesítési és szervízszolgáltatásai  
 Értékesítési és szervízszolgáltatásai  
 Értékesítési és szervízszolgáltatásai

os Commodore Egyesület szolgáltatásai

**Novotrade**  
 SZERVÍZ Kft.

### 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

#### Egyesületi tagoknak 20% kedvezmény:

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

#### Egyesület tagoknak 30% kedvezmény:

Speeddos (átkapcsolható) operációs rendszer beépítése (C64 átalakítás, lemezegység átalakítás + párhuzamos kábel)	5000 Ft
1541 kompatibilis lemezegységbe elektronikus lemezlyukasztó beépítése	900 Ft
PAGEFOX magyar ékezetes kiadvány-szerkesztő cartridge (a teljes A/4-es oldal kinyomtatásához 640 pont/soros nyomtató szükséges minimum, pl. Citizen 120D)	5500 Ft
FASTLOAD cartridge (lemezes gyorstöltő, másoló, monitor)	1500 Ft
TTL IC-teszter cartridge + program	4300 Ft
288/256 Kbyte-os eprombank (vezérlő eprommal)	5000 Ft
Epromégető (2716-tól 27512-ig)	5000 Ft
C64-hez tároló oszcilloszkóp	8000 Ft
C64-bővítő-port elosztó (egyszerre 4 db cartridge lehet a gépben, melyeket gombnyomásra lehet kapcsolni)	7500 Ft
C64 USER — CENTRONICS nyomtatókábel (GEOS kábel)	1500 Ft
256 K RAM-diszk (256 Kbyte RAM-mal)	14000 Ft
256 K RAM-diszk (64 Kbyte RAM-mal)	9000 Ft
2x64 Kbyte-os cartridge igény szerinti programokkal feltöltve	4.300 Ft
64 Kbyte-os cartridge igény szerinti programokkal feltöltve	3000 Ft
Képűjság (teletext) dekóder C-64-re	10000 Ft
1764-es RAM bővítő GEOS-hoz is használható	11250 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az eprommal együtt egységesen	700 Ft

Árainkat az alkatrészarak változásai befolyásolhatják.

A fenti bővítések megrendelhetők levélben az O.C.E. címén, valamint személyesen a havonta rendezendő klubdélelőttön, ahol rendszeres bemutatót is tartunk.

Postázás esetén 100 Ft postaköltség kerül felszámításra.

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

## MÁJUS

### 60 Ft-os vásárlási utalvány

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

Érvényes: 1994. május 31-ig.

### HOBBI ELEKTRONIKA

## MÁJUS

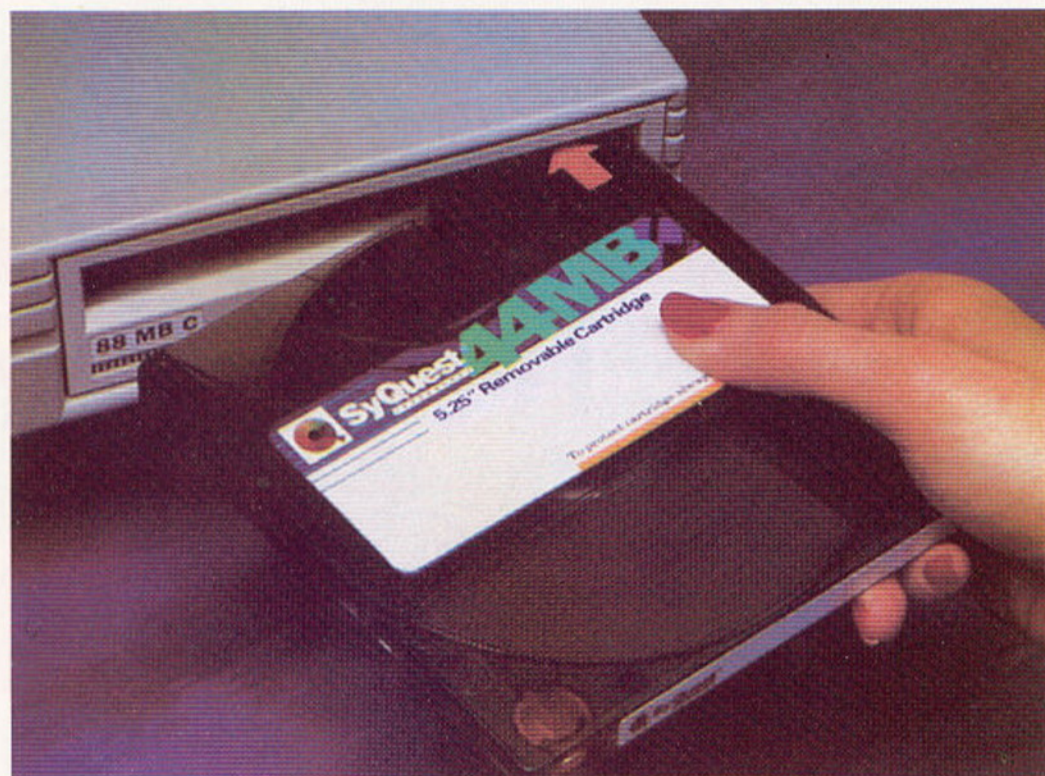
### vásárlási utalványa

Értéke:

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

Beváltható a Hobbi Elektronika Kft.-nél. Budapest VII., Dózsa György u. 16. Telefon: 122-8892

Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!



**SyQuest**  
TECHNOLOGY

Hivatalos magyarországi disztribútora:

**NOVOTRADE**

SZERVIZ Kft.

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

## Cserélhető lemezes winchester!

Szervizeinkben és szerződött viszonteladóinkon keresztül az alábbiakat forgalmazzuk:

SQ 555	44 MB-os 5 <sup>1</sup> / <sub>4</sub>	SCSI drive,	SQ 400 lemez	( 44 MB)
SQ 5110	88 MB-os 5 <sup>1</sup> / <sub>4</sub>	SCSI drive,	SQ 800 lemez	( 88 MB)
SQ 3105A	105 MB-os 3 <sup>1</sup> / <sub>2</sub>	AT-BUS drive,	SQ 310 lemez	(105 MB)
SQ 2542A	42 MB-os 2 <sup>1</sup> / <sub>2</sub>	AT-BUS drive,	SQ 240 lemez	( 42 MB)

## Számítógépszerviz !

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

**Várjuk jelentkezését az alábbi szervizeinkben:**

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

**NOVOTRADE SZERVIZ KFT.**



# ACOMP

## S Z Á M Í T Á S T E C H N I K A I K F T.

1125 Budapest, Királyhágó utca 2.  
1135 Budapest, Szent László út. 74./A.  
1191 Budapest, Katica utca 9.

Tel.: 156-6790  
Tel.: 149-6165  
Tel.: 280-4267

Fax: 251-2385  
Fax: 251-2385  
Fax: 177-9419

Nyitva: hétfőtől péntekig: 9—17 óráig, szombaton 9—13 óráig.

Commodore Amiga 600	34 990 Ft	Handyscanner fekete/fehér Amigához	17 990 Ft
Commodore Amiga 1200	52 990 Ft	Stereo hangdigitalizáló Amigához	6 990 Ft
Commodore Amiga 1200 Desktop Dynamite	59 990 Ft	Sound Enhancer Amigához	3 500 Ft
Commodore Amiga 2000	39 990 Ft	Trackball Amigához	3 500 Ft
Commodore Amiga CD-32	49 990 Ft	Rockey Advanced Video Keying for Amiga	24 900 Ft
A 600/A 1200-hoz 40 MB HD kábellel	21 990 Ft	G. V. P. A4000-040/40/4	259 000 Ft
60 MB HD kábellel	24 990 Ft	G. V. P. G-Lock genlock S-VHS	59 900 Ft
80 MB HD kábellel	28 990 Ft	G. V. P. A1230-II/50/00/4 + DPP-II	119 900 Ft
120 MB HD kábellel	36 990 Ft	G. V. P. A1291 SCSI Option	14 900 Ft
Commodore Amiga 4000/040-40MHZ/6MB/0MB	???	G. V. P. Impact Vision 24/CT-PAL	198 900 Ft
Commodore Amiga 4000/040-25MHZ/6MB/0MB	309 000 Ft	1.76 MB HD külső drive Amigához (OS2+)	16 990 Ft
Commodore Amiga 4000/030-25MHZ/4MB/0MB	169 000 Ft	3.5" Amiga slim külső drive	9 990 Ft
+ 4 MB RAM modul	27 990 Ft	2.5"-2.5" Hard disk kábel	2 990 Ft
Commodore 1942 Multisync Stereo Monitor	59 990 Ft	2.5"-3.5" Hard disk kábel	2 990 Ft
Commodore 1084s Stereo-Color monitor	35 990 Ft	Mouse pad	290 Ft
Commodore 1085s Stereo-Color monitor	35 990 Ft	Noris MB 80 3,5" lemeztartó	490 Ft
Commodore A-520 TV-Modulator (Amigával)	4 990 Ft	Noris DB 100 5,25" lemeztartó	490 Ft
Commodore Amiga - EuroScart kábel	1 290 Ft	Amiga Magazin (német) újság	490 Ft
Commodore C-64 Terminator + Datasette	15 990 Ft	Power Play (német) újság	490 Ft
Commodore C-64 TERMINATOR SET	14 990 Ft	Beyond the Minds Eye (Computer Animáció)	3 000 Ft
Commodore Datasette	2 990 Ft	Sound Blaster PRO-2 Delux	15 900 Ft
Sega Megadrive + 2 pad + 4 játék	23 990 Ft	Sound Blaster 16 ASP Delux	31 900 Ft
Sega Megadrive + 1 pad + 1 játék	15 990 Ft	Joystickok	
Sega MegaCD II. + 3 CD játék	51 990 Ft	Quickshot QS - 113 IBM	1 190 Ft
Sega MegaCD II. + Road Avenger	41 990 Ft	Quickshot QS - 123 Warrior IBM	1 390 Ft
Sega Game Gear + Columns	12 990 Ft	Quickshot QS - 151 Aviator5 IBM	3 790 Ft
512 Kb órás memóriabővítő	4 990 Ft	Quickshot QS - 172 Raider5 IBM	1 750 Ft
1.0 MB-os órás chip bővítő A600-ba	7 900 Ft	Quickshot QS - 189 Python5 IBM	1 390 Ft
Ram nélküli órás fast bővítő A1200-ba	11 990 Ft	Quickshot QS - 191 Starfighter5	1 290 Ft
1.0 MB-os órás fast bővítő A1200-ba	16 990 Ft	Quickshot QS - 101 I.	550 Ft
2.0 MB-os órás fast bővítő A1200-ba	23 990 Ft	Quickshot QS - 111A II. Turbo Mikrokap.	990 Ft
4.0 MB-os órás fast bővítő A1200-ba	37 990 Ft	Quickshot QS - 128 Maverick1	1 490 Ft
8.0 MB-os órás fast bővítő A1200-ba	62 990 Ft	Quickshot QS - 130F Python1	1 190 Ft
2.0 MB PCMCIA bővítő A600-ba, A1200-ba	19 990 Ft	Quickshot QS - 137F Python1M Mikrokap.	1 290 Ft
Noris Porvédő Amiga 600	990 Ft	Quickshot QS - 155 Aviator1 Mikrokap.	2 890 Ft
Noris Porvédő C-64 II.	790 Ft	Quickshot QS - 149 Intruder1	2 690 Ft
Noris Porvédő C-64 I.	150 Ft	Quickshot QS - 130N Pyton2 Nintendo	990 Ft
Noris üveg 14"-os földelhető monitorfilter	1 290 Ft	Dynamics Competition Pro IBM	3 990 Ft
NoName 3.5" DSDD lemez	550 Ft	Dynamics Competition Pro 5000	1 190 Ft
NoName 3.5" DSHD lemez	790 Ft	Dynamics Competition Pro 5000 Mini	1 690 Ft
NoName 5.25" DSDD lemez BULK	200 Ft	Dynamics Competition Star Mini	2 190 Ft
Maxell 3.5" MF2-HD lemez	1 390 Ft	Dynamics Competition Special Mini	1 690 Ft
BASF 5.25" DSHD lemez	790 Ft	Dynamics Competition Transparent Mini	1 690 Ft
Wonderline 5.25" DSHD lemez	490 Ft	Commodore CD-32 játékok	Egységáron: 4 990 Ft
Polaroid 3.5" DSDD lemez	690 Ft	Nigel Mansell's World Championship,	
Polaroid 3.5" DSHD lemez	1 190 Ft	The Labirinth of time, Zool, Morph, Alfred Chicken,	
Polaroid 5.25" DSDD lemez	450 Ft	Sensible Seccer, Deep Core, Liberation, D-Generation,	
Profex 3.5" DSDD lemez (11 db/Form)	690 Ft	Pinball Fantasies, Mean Arenas, Whale's Voyage,	
Profex 3.5" DSHD lemez (11 db/Form)	990 Ft	Troll, James Bond 2, Overkill & Lunar-C, Castles II.,	
Profex 5.25" DSHD lemez (11 db/Form)	390 Ft	Pirates Gold, Dangerous Streets, Seek & Destroy,	
Action Replay MK VI. Pro (C-64) + könyv	5 900 Ft	7 Gates of Jambala, Summer Olympix, Nick Falco's Golf	
C-64 midi szoftverrel	6 500 Ft	Chamber's of Shaolin, Lotus Trilogy, Twon with no name	
Swiftly Amiga Mouse (3 gombos)	2 500 Ft	SEGA MEGADRIVE,	
Mouse - Joystick automatikus kiválasztó	2 500 Ft	MEGA CD JÁTÉKOK: 2 990 Ft - 8 990 Ft	
Real Time Clok Amiga 1200-hoz	2 500 Ft		
Midi Amiga Interface	3 500 Ft		

Áraink mindenkori változtatásának a jogát fenntartjuk!

Áraink az 1 év garanciát és az ÁFA-t tartalmazzák!

