

Az Országos Commodore Egyesület lapja

**újság**

VIII. évfolyam

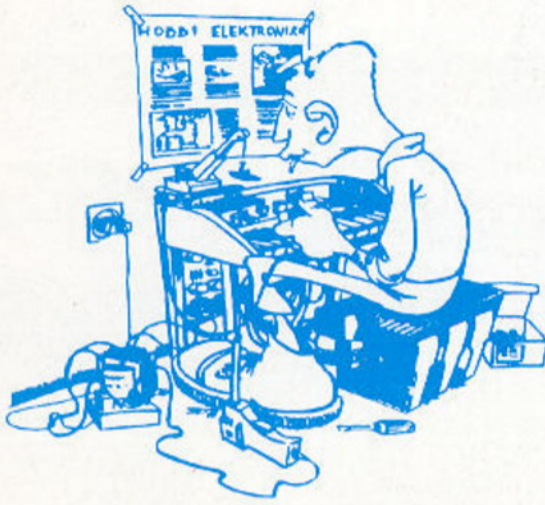
**1993 / 3**



**Játék-  
pályázat**

***Bővebb terjedelem  
Díjnyertes programok***





# HOBBI ELEKTRONIKA



**Urbán István mérnök áramköreinek szaküzlete**

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

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

## KÍNÁLATUNKBÓL:

### SZÁMÍTÁSTECHNIKA:

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

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

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

### ZENE - HANGTECHNIKA:

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

### VIDEÓS TÉMÁK:

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

### FÉNYTECHNIKA:

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

### EGYÉB HOBBI TÉMÁK:

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

**Az árak a 25% ÁFA-t tartalmazzák!**

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

### MŰSZERVÁSÁR

Bontott anyagok,  
használt műszerek  
bőngészője



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 irodájában (1025 Budapest, Vöröstorony utca 29. Telefon: 1-76-22-57), vagy átutalással az MNB 217-98 292, OTP 565-3610-8 számlára lehet befizetni. Megrendelés esetén számlát küldünk.

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

A **DEÁKPÁHOLY** tagjai minden hónapban megkapják a C-újságot, a tagsági díj egy évre 890 Ft.

A **PLUSZPÁHOLY** tagjai minden hónapban megkapják a C-újságot, és kapnak havonta 3 db vásárlási utalványt. A tagsági díj egy évre 2050 Ft.

A **SZUPERPÁHOLY** tagjai havonta 15 példányt kapnak a C-újságból, és ezzel havonta 15x3 db vásárlási utalványt is. Az éves tagsági díj 24 000 Ft.

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

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

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

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

**Kedvezményes ár! Tagoknak olcsóbb!**

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

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

### VIDÉKEN TOVÁBBI INFORMÁCIÓK KAPHATÓK:

Baja, AXIS Kft.,

Győri Bartók Béla Művelődési Ház,

Jászberényi Városi Könyvtár,

Kecskemét, SZIGMA—BIT,

Pécsi Apáczai Csere János Gimnázium,

Zalaegerszegi Ságvári Endre Gimnázium.

Az Országos Commodore Egyesület módszertani kiadványa

Egyesületi iroda és szerkesztőség:

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

Felelős kiadó: Horváth Judit, az egyesület elnöke

Főszerkesztő: Rados Péter, az OCE főtítkára

Felelős szerkesztő: dr. Horváth András

Művészeti szerkesztő: Bausz Sándor

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

Index: ISSN 0237-756 X

Terjeszti a Magyar Posta

Megvásárolható a hírlapárusoknál

92.0305 MSZH Nyomda és Kiadó Kft., Budapest

Felelős vezető: Nagy László

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

### tulajdonosok!

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

Cím: 1181 Budapest, Kondor Béla sétány 8.

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

Legközelebb:

**február 21-én,**

**március 21-én és**

**április 18-án.**

## FELHÍVÁS

Tisztelt Tagtársunk, Olvasóink!

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

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

Köszönettel:

Országos Commodore Egyesület

## Tisztelt Szerkesztőség

Régi és hűséges olvasója vagyok a C-Újságnak. Sok jó ötlettel találkoztam már cikkeikben, többek között a most általam közölt témákkal is.

Az első, pár soros programom a DATA adatok begépelését könnyíti meg. Igaz, első megközelítésre furának tűnik, de ennél nem kell különféle kezdő- és végcímetek állítgatni, meg POKE-olni. Egy abszolút kezdő is könnyen használhatja.

Egyébként ennek elbírálását Önökre bízom.

A másik egy ugyancsak régen lerágott csont: a magnó hanggossá tétele. Sokat foglalkoztatott a dolog, ugyanis a legelső gépem egy PRIMÓ volt. Ismerős ugye? Hangos a billentyű leütése, és hangos a szalag beolvasása, sőt beolvasásnál figyelemmel lehetett kísérni a rekordok számát, valamint a felvételi hibát.

Bizony ezek a „kis” pluszok a jóval drágább rokonoknál hiányoznak. Így jutottam el a különféle próbálgatások során a közölt ötlet megvalósításához.

Mióta ezt a kis csipogót beszereltem a magnóba, nem gond megtalálni a szalagon a programok elejét, végét, illetve az üres helyeket.

E két kis ötletet szeretném közreadni amennyiben ezt Önök is jónak látják.

Tisztelettel:

Kadlicskó József



# DATA beíró

A program használata:

Indítás után a gép kérésére beírjuk azt a sorszámot, ahonnan a DATA kezdődik.

A négy egymásba ágyazott I, J, K és D ciklusok segítségével csak a DATA-ba kerülő adatokat kell beírni, minden mást a gép végez el.

Tulajdonképpen az történik, hogy parancs módba kerül kiírásra a sorszám, a ? (CHR\$(63)), idézőjel CHR\$(34) idézőjelben a sorszám és a DATA (A\$) s az adatok végére újabb idézőjel.

Ha van a DATA végén ellenőrző összeg, akkor az a D ciklussal ugyancsak beírható, ha nincs akkor az 55–65 sorra nincs

szükség, viszont az 50. sor folytatásaként be kell írni: CHR\$(34).

Nagyon ügyeljünk a programban lévő ;'-re.

Amikor a ciklus végére értünk, a képernyő alján lévő utasítást kövessük. A kurzor mindig a sorszám előtti üres helyekre kerül, itt nyomjuk meg mindig a <RETURN> billentyűt.

Majd >RUN kezdő sorszám. Ekkor kiíródnak a sorszámmal ellátott DATA sorok. Ezeket ismét RETURN-al vigyük be a gépbe.

Így folytassuk mindaddig míg a végére nem érünk a DATA adat bevitelének, természetesen a kezdő sorszámok mindig az előző adatok folytatásaként következnek.

Ha minden adatot beírtunk, akkor a programot töröljük s helyére beírjuk a DATA beolvasó-, ellenőrző programot.

Így leírva talán kicsit bonyolultnak látszik, de a kezelése annál egyszerűbb.

```

1 REM *****
2 REM *
3 REM * GYORS DATA-BEÍRÓ *
4 REM * C= 64 +4 *
5 REM *
6 REM * KADLICKÓ JÓZSEF PÉCS *
10 PRINT CHR$(147):A$="DATA"
15 INPUT"QQ DATA ELSŐ SORSZÁMA:";A:"?"SH/HOME|HOME|";
20 FOR I=A TO A+10
25 PRINT I CHR$(63)CHR$(34)I;A$ " ";
30 :: FOR J=1 TO 8
35 :::FOR K=1 TO 2
40 :::GET V$:IF V$="" THEN 40
45 :::PRINT V$;:NEXT K:PRINT ", ";
50 :::NEXT J
55 FOR D=1 TO 4
60 GET V$:IF V$="" THEN 60
65 PRINT V$;:NEXT D:PRINT CHR$(34)
70 NEXT I
75 PRINT" 1. <HOME> SORSZÁMONKÉNT <RETURN>"
80 PRINT" 2. <TÖRLÉS> - >RUN "A:"?"6 UP"
85 END
1000 PRINT"Q <HOME> SORSZÁMONKÉNT <RETURN>"

```

# Hangos DATASZETT

Sok gondot okoz a COMMODORE gépek ún. „némasága”, különösen amikor szalagról töltünk be programot. Ezzel kapcsolatban már sok cikk jelent meg pro és kontra.

Most egy általam már hosszú idő óta alkalmazott ötletemet szeretném megosztani C-os társaimmal.

Egyszerűen — bárki által — megoldható az ötlet, nem kell hozzá más mint a kisebb műszaki alkatrészboltokban beszerezhető HPE-200 típusú piezo sugárzó. Ára: 120,— Ft. Pécsen a Király u. 37. alatti műszaki boltban kapható.

Szerelési útmutató: A magnó csatlakozó dugóját szét kell szedni a kis csavar eltávolításával. A csatlakozó fedelének egyik oldalán van egy kis 1 mm-es nyílás, ezt kell pár tized mm-rel nagyobbra kivágni.

A sugárzó piros színű vezetékének lecsupaszított végét dugjuk be a fehér színű vezeték forrasztási helye mellé. A bátrabbak

oda is forraszthatják. Ugyanígy a fekete színű végét a kék színű vezeték beforrasztott vége mellé. A négyes érintkezők közül a kék a szélső, a fehér a 3.

A fedeleken lévő furatokat 3 mm-re bővítsük ki s kerítsünk egy az eredeti csavarnál kb. 4 mm-rel hosszabbat, s ezzel rögzítsük a sugárzót a csatlakozó C-jelű oldalára. Előtte azonban a vezetékeket a kibővített nyíláson hozzuk ki. Ezzel kész is a kis csipogó.

A szerelésben jártasabbak elhelyezhetik a magnó belsejében is a csipogót a piros ledet rögzítő csavar alá.

Egy 20×5×1 mm-es bakelit lemez mindkét végére készítünk 3 mm Ø furatot. Az egyik végére a csipogót, a másik végére pedig a led-et rögzítő csavart erősítsük. A huzalokat meg kell hosszabbítani úgy, hogy azok a motor mellett elérjenek a megfelelő színű kivezetésekhez.

A kék színű vezeték a panel mellett lévő kis kapcsolónál van, a fehér pedig a panel hátsó végénél. A kékhez történő forrasztást megkönnyíti, ha panelt rögzítő két kis csavart eltávolítjuk és a panelt kissé elfordítjuk. Forrasztás után a panelt ismét a helyére rögzítjük.



**Tisztelt Szerkesztőség!**

A mellékelt lemezen található Gárdonyi Gergely „Programkönyvtár” című programjának (C-Újság, 1988/1.) általam kissé bővített változata és külön a bővítés listája.

A bővítéssel a program nyomtató nélkül is jól használható. A tartalom az F1 billentyű lenyomásával laponként a képernyőre listázható.

A módosítást felajánlom a C-Újságban való közlésre.

Üdvözlettel:

Haraszti Árpád

```

0 REM *****
1 REM *      C=UJSAG  SORSZAM    092      *
2 REM *
3 REM *      PROGRAMKONYVTAR          *
4 REM *
5 REM *      JAVITOTT VALTOZAT        *
6 REM *
7 REM *      PROGRAM:  GARDONYI GERGELY *
8 REM *      JAVITAS:  HARASZTI ARPAD  *
9 REM *****
230 PRINT:PRINT"LISTAZAS KEPERNYORE(TOVABB 'F1') "
661 IF (F/20=INT(F/20))AND(F<>DB-1)AND(F<>0)AND(F<>1)THENGOSUB2500
800 PRINT#4,"NEV"CHR$(16)"43"HOSSZ ID";
810 PRINT#4,"NEV"CHR$(16)"69"HOSSZ ID";
870 PRINT#4,"N$(F)P$"19"H(F)P$"24"I$(F)"N$(A+F)P$"44"H(A+F)P$"49"I$(A+F);
880 PRINT#4,"N$(B+F)P$"70"H(B+F)P$"75"I$(B+F)
2500 GETA$:IFA$=" "THEN RETURN
2510 GOTO 2500
READY.

```

**Tisztelt Szerkesztőség!****Datásító C+4-re**

```

0 goto62990
1 data 4c7c0c484a4a4a4a20120c6829,313
2 data 0f20120c6020210cc90a900318,278
3 data 6907693091d460e6d4d002e6d5,715
4 data 60bd640d1865d491d420210ca5,536
5 data d5690091d4e8bd640de8c900f0,75a
6 data 0920210c91d4c900d0ef60a200,545
7 data 20280ca5d120030ca5d020030c,39d
8 data 203a0ca5d7200c0ca5d620030c,3c4
9 data 203a0c20420c20210c20280c20,195
10 data 420ce0b090f360788d3fffa00a,6ae
11 data b9590d91d48810f8a5d0a4d148,746
12 data 9848a90085d685d7a5d4852b18,681
13 data 690b48a00091d4c8a5d5852c69,61d
14 data 0091d485d56885d4a5d4186925,69f
15 data a00091d4c8a5d5690091d4c8e6,7c3
16 data d6d002e6d7a5d691d4c8a5d791,91a
17 data d4c8a98391d4c8a92091d4a5d4,89c
18 data 1869059002e6d585d4a20da900,584
19 data 85d985daa000b1d0481865d990,70c
20 data 02e6da85d96820030ce6d0d002,63f
21 data e6d1cad0e520210ca92c91d4a5,762
22 data da200c0ca5d920030c20210ca9,3b5
23 data 0091d420210ca5d1c5d39087a5,67c
24 data d0c5d290816885d16885d0204c,75f
25 data 0c20210ca90091d420210ca900,35d
26 data 91d420210ca5d4a6d5852d862e,60c
27 data 8d3eff584c03870c1000008936,3d3
28 data 33303030002118f6e83a53b2d1,4ea
29 data 28220022293a4bb2303a51b2d1,40a
30 data 2822002229000f19f68149b231,360
31 data a4513a4cb230000c1af6874124,465
32 data 3a874224001f1bf6814ab230a4,4a8
33 data 31323a42b2d128ca2841242c32,43f

```

Nem mindenki szereti, ha kedvenc számítógépes lapjában hexdump listákkal találkozunk, aminek bevitele elég nehézkes és ha elgépeljük, senki sem szól ránk, hogy hibásan írtunk be valamit. S ha ezek után a program még fut is, az a legbosszantóbb, hiszen bármikor előjöhethet a nem kívánt programhiba. Ilyenkor mindig a programozót szokták szidni, holott szegény nem tehet róla, hogy nincs egy jó BASIC DATA sorokat készítő programja.

Nos, kedves programozók, íme az áhított program. Igazán nem hosszú. „Na persze,” mondhatnák most, „öt perc alatt írok egy datásítót BASIC-ben.” Ennyi idő alatt ezt a programocskát is be lehet gépelni. De most jön a lényeg: a BASIC DATA-készítő jó esetben 10–15 PERC alatt végez a programmal, és még mindenféle számításokat is kell végezni, hogy a bevétel is helyes legyen. Az itt bemutatott DATA-készítőt a Wash & Go samponhoz hasonlíthatnám, csak beindítjuk és kész: a leghosszabb bevihető programmal (14 Kbyte, ennél több DATA sorok formájában be sem fér a memóriába) 2,8 MÁSODPERC alatt kész van, és a BASIC program azonnal menthető, nincs szükség a feleslegessé vált DATA-készítő sorok törlésére.

Rekordsebessége mellett a program még ellenőrző kód készítését is elvállalta: a BASIC program indítása után azonnal ordít, és a hiba helyét is megmondja. Idegeskedő természetű emberek számára lefelé számol a gépi kód bevitele alatt. Amint kiírta a „Készen vagyok” üzenetet, a gépi kód indítható, ellenkező eset-











```

83 next i: close 8
84 rem ----- 4.
85 if q=0 then print "every checked file is ok.": goto 108
86 print "assign the files to scratch!"
87 print "nr filename          typ len": print u$
88 c=0: for i=0 to n: if e%(i)=0 then 96
89 gosub 124: if i=0 then print : goto 96
90 print "tab(36)"
91 if n$(i)="" then 95
92 gosub 142: if v=89 then print "yes": c=c+1: goto 96
93 if v=3 then print : goto 110
94 if v<>78 then 92
95 print "no": e%(i)=0
96 next i: if c=0 then 108
97 if z then c=0: goto 102
98 print "validate after scratching?": c=0
99 gosub 142: if v=89 then c=1: goto 102
100 if v<>78 then 99
101 rem ---
102 print "Scratching -": print#15,"i0": gosub 144: if e then 110
103 for i=1 to n: if e%(i)=0 then 106
104 print "i". tab(7): v=n$(i): gosub 138: print
105 print#15,"s0:"n$(i): input#15,e
106 next i: if c then print " - validating -": print#15,"v0": gosub 144
107 rem ----- end
108 print " - end -"
109 close 8: close 15: poke 650,0: poke 788,49: end
110 print "program aborted": goto 109
111 rem ===== b-r
112 if t=0 then return
113 get v$: if v$=" " then v=3: return
114 c=c+1: v$=right$(str$(t),2)+" "+right$(str$(s),2)+" "+right$(c,3)
115 print "tab(15)v$"
116 print#15,"u1:"s;t:s: input#15,e: if e then return
117 if b%(t,s) then e=400: return
118 b%(t,s)=1: a=t: b=s: gosub 140: t=x: gosub 140: s=x: goto 112
119 rem ===== line1
120 print right$(" "+str$(i),3)
121 v$=left$(n$(i)+s$,16): gosub 138
122 print "t$(t%(i) and 7)"right$(" "+str$(abs(1%(i))),3): return
123 rem ===== line2
124 print right$(" "+str$(i),3) ": gosub 121: print : e=e%(i)
125 if e=100 then print ", "type error": return
126 if e=200 then print ", empty name": return
127 if e<>300 then 129
128 print ", incorrect length "mid$(str$(int(t%(i)/8),2)": return
129 print " ": if 1%(i)<0 then print "#";
130 print right$(" "+str$(int(t%(i)/8),3).blk ";
131 print right$("0"+mid$(str$(a%(i,0),2),2)"-";
132 print right$("0"+mid$(str$(a%(i,1),2),2)";
133 print " ("right$("0"+mid$(str$(int(w%(i)/256),2),2)"-";
134 print right$("0"+mid$(str$(w%(i) and 255),2),2)" ";
135 if e=400 then print "overlap": return
136 print "err:"right$("0"+mid$(str$(e%(i),2),2): return
137 rem ===== print name
138 poke 216,16: for y=1 to 16: print mid$(v$,y,1): next y: poke 212,0: return
139 rem ===== get#
140 get #8,x$: x=asc(x#+z$): return
141 rem ===== get
142 wait 197,64: poke 198,0: wait 198,15: get v$: v=asc(v$): return
143 rem ===== error
144 input#15,e,e$: if e=0 then return
145 print "disk:e;e$": return
146 rem =====
147 data del,seq,prg,usr,rel," * ",???

```



# Lexikon

## XIII. rész

### RESET

A reset alatt egy olyan folyamatot értünk, amely a számítógépet (pl. a C64-est), vagy a perifériákat alapállapotba viszi. Alapállapot alatt a bekapcsolás utáni állapotot értjük. A C64-es processzora erre a célra egy külön reset vezetéssel rendelkezik, amelynek használatával (resetjel) elvégezhető a kívánt művelet. A C64-es esetében ilyenkor bizonyos tárolócellák egy meghatározott értékre lesznek beállítva. Beállítjuk az úgynevezett bekapcsolási színeket a képernyőn és a csatlakoztatott készülékek felé is küldünk egy reset jelet. Akinek van egy úgynevezett reset kapcsolója (ezt a szaküzletekben kereshetjük), megspórolhatja a számítógép állandó ki-be kapcsolgatását azzal, hogy megnyomja az adott gombot. Ez a módszer ráadásul kíméletesebb is, mert ekkor nem lép föl a bekapcsoláskor az építőelemeket (igen rövid ideig érő) indukciós „sokk”. Így jobban védjük a C64-es elemeit.

A resetet általában akkor kell kiváltanunk, amikor a rendszer valamilyen oknál fogva lemerevedett, „kiakadt”. Mivel a reset során a C64-es esetében csak a fontos mutatók beállítása és a kazettapuffer törlése történik meg, ez hasznos eszköz a programozásban is. Gépi kódú programok vagy grafikák ugyanis változatlanul a tárolóban maradnak (kivéve az említett kazettapuffert).

### STEPPERMOTOR

Ezt léptetőmotornak is nevezik. A 1541-es meghajtómotorjával ellentétben a léptetőmotor az író/olvasó fej mozgatójáért felel. Attól függően hogy hova kell írni az adatokat a lemezre, húzza vissza vagy viszi előre a léptetőmotor a fejet. Ez a felelős a régi 1541-esnél a lemezformálásakor hallható kattogásért, mivel ekkor a

motor a fejet ütközésig visszahúzza, hogy a műveletet egy meghatározott nullaponttól kezdje el. A fej meghajtása nem közvetlenül történik, hanem egy fémszalag segítségével, amely átviszi a mozgásokat a fejre. Erre a szalagra van a fej tartója felerősítve, amely síneken fekszik, így a fej két mozgásirányra szilárdan rögzítve van.

### ÍRÓ/OLVASÓ FEJ

A floppynek eme alkotórésze veszi át a lemezreírás tulajdonképpeni feladatát. Az egész elem két részből áll, magából a fejből és egy kis filcből, amely a lemezt főnről a fej irányába nyomja. Természetesen egy lemezre nem a szó hagyományos értelmében írunk. Sokkal inkább az történik, hogy a fej a lemeznek azon részét mágnesezi, ahonnan később az adatokat olvasni szeretnénk. Az olvasásnál tehát az írási művelet megfordul. Ez azt jelenti, hogy ekkor az író/olvasó fej azt ellenőrzi, mágneses-e az éppen vizsgált rész. Innen kerülnek azután az adatok a számítógépbe. A fej szilárdan rögzül a léptetőmotor fémszalagjára, és ezen kívül két sínen csúszik. Így a fejet nem lehet sem balra, sem jobbra kitéríteni. Az ellennyomást kifejtő filc gondoskodik emellett a lemez és az író/olvasó fej megfelelő kapcsolatáról.

### TÁROLÓ MÉDIUMOK I.

Hogy adatokat tartósan tárolhassunk, több lehetőség is a rendelkezésünkre áll. Ma már szinte történelmi tárolóeszköz a hagyományos magnószalag. Az adatokat itt a zenéhez hasonlóan visszük föl egymás után. Ennek a módszernek a legnagyobb hátránya a lassú sebesség. Jelenleg az információ tárolásának legnépszerűbb elemei a mágneslemezek. Különböző méretekkel találkozunk, amelyek mögött kü-

lönöző tárolókapacitás áll. Eredetileg 8 collos lemezeket használtak, de ezek kezelése túl bonyolult lett. Ezekre a lemezekre megabyte nagyságrendű adatmennyiségeket lehet fölvenni. A 1541-es által használt 5 1/4 collos lemezek manapság a legáltalánosabban elterjedt információhordozók. Ezekre a használt meghajtótól függő kapacitással lehet adatokat írni. A 1541-es floppy például 166 kbyte adatot képes ezekre írni.

### TÁROLÓ MÉDIUMOK II.

A lemezpiacon a legfiatalabb lemez méret a 3 1/2 coll. Ezeket például a Commodore Amiga használja. Ezeknél is sokkal nagyobb kapacitást jelentenek az úgynevezett fixlemezek. Ezeket azonban általában csak a PC-khez kínálják, bár az USA-ban létezik egy fixlemez a C64-hez is. A fixlemezek manapság gigabyte nagyságrendű információk befogadására képesek, ezeket azonban csakis a nagygépes számítóközpontokban alkalmazzák. A legújabb, ma még kissé kiforratlan tárolóeszköz a Compact Disc. Ezekre, a HiFi világából ismert szeletekre igen nagy mennyiségű adatot lehet fölvenni. Ugyanakkor az ezek olvasásához szükséges készülékek csakis a PC-k számára elérhetők. Emellett ez a technika még nem nevezhető beérettnek, s az alkalmazása igen drága is.

#### RAM operációs rendszer védelme

Ha kimásoltuk az operációs rendszert a RAM-ba, s ezután el akarjuk kerülni, hogy egy (RUN/STOP) + (RESTORE) kikapcsolja ezt, elég, ha beadjuk a POKE 64982,53 utasítást.

#### Directory listavédelem

Aki a lemezeit szeretné listázás ellen védeni, a jövőben a következő utasítással formátálja azokat: OPEN1,8,15,"N:név(shift-l)":close1  
Az ID helyén beadott <SHIFT> <L> meggátolja a directory kilistázást.

#### Dump-master

Ez a kis rutin megakadályozza a 1541/70/71-es floppyknál az író/olvasófej ütődését:

```
OPEN15,8,15,"M-W" CHR$(106) CHR$(10)
CRS$(1) CHR$(128):CLOSE15
```



# Képernyővarázslat — a TED

## Hatodik rész: A memória és a billentyűzet programozása

Nos, végre eljutottunk a cikksorozat utolsó részéhez. Persze ez még csak a cikk eleje, és remélem, tudok új információkat nyújtani a TED-ről, valamint hogy még mindig vannak, akik ezt elolvassák.

### A billentyűzet

Mint azt már az első részben leírtam, a billentyűzet és joystick kezelése is eme chip feladata. A TED K0—K7 bitjeivel figyeli ezeket. Nézzük meg, hogyan!

Az I/O területen (\$FD00-\$FEFF) belül az \$FD30-\$FD3F terület a billentyűzet-mátrix számára fenntartott hely. Az FPLA (Field Programmable Logical Array — programozható logikai hálózat; ezt a gépbe készre égetve, programozva szerelik be) a címbusz felső 12 bitjét figyel (A4—A15), így 16 bájtos blokkokkal képes megkülönböztetni. Ha a \$FD3× címek bármelyikét találja a címbusz (ezek között semmi különbség nincs), aktivál egy jelet. Ez a KEYBOARD CHIP-SELECT (billentyűzet chip kiválasztás) jel egy 6529B típusú IC-be fut be. Ez az IC kapcsolódik az adatbusz mind a nyolc bitjére. A KEYBOARD CHIP-SELECT vételekor eltárolja az adatbusz állapotát, a kimenetére helyezi, és a következő írásig megtartja azt. Erre a kimenetre csatlakozik maga a mátrix. Ez valójában egy 8\*8-as hálózat. Eme hálózat 64 metszéspontjának a gép 64 különböző billentyűje felel meg (a két SHIFT és a SHIFT-LOCK, valamint a két CONTROL [csak PLUS/4-en] egy-egy billentyűnek számít). Nyolc sorát a 65229B kimenetéről érkező jelek alkotják, míg nyolc oszlopa a TED K0—K7 bitjeibe fut be. Ezek a bitek ellenállásokkal +5 V tápfeszültségre vannak felhúzva. Egy billentyű lenyomásakor a hálózat azon sorának állapota, amelyikben a lenyomott billentyű van, a TED K0—K7 bitjeinek valamelyikébe íródik. Hogy melyikbe, az a billentyű mátrixbéli helyétől függ, vagyis attól, hogy melyik sor és oszlop metszéspontjában áll. Ha az adatbuszon ehhez a sorhoz „egyes” bitet ren-

delünk, a TED nem érzékeli a lenyomást, hiszen az ellenállások eleve „magas” szinten tartották a TED bemeneti bitjét. Ha azonban „alacsonyra” állítottuk a sor-kiválasztó jelet, a billentyű a TED felé vezető jelet is lehúzza. Ez az, amit a TED \$FF08 regiszterében érzékelhetünk. A két joystick a mátrix után, közvetlenül a TED bemenő bitjeire kapcsolódik, így ha ezeket kívánjuk figyelni, és a billentyűzetet nem, az \$FD30 cím írását kihagyhatjuk. Nézzük most meg mindezt programban (1. lista)! A SEI utasítás ilyen esetekben elengedhetetlen, ugyanis ha a gépi megszakító rutin — mely ugyanezt a módszert használja — átírja az általunk beállított értéket, hamis értéket olvasunk be. A rendszer ROM-ban található egy hasznos szubrutint. Hívási címe: \$DB70/73. Itt az alábbiakat találhatjuk:

DB70 STA \$FD30; billentyűzet is  
DB73 STA \$FF08; csak joystick  
DB76 LDA \$FF08  
DB79 RTS

Mint említettem, a gépi megszakító rutin is ezt a módszert használja, majd a kapott kódokból ASCII-kódot készít. Ehhez konverziós táblákat használ, melyekből négy darab van, és az \$E026-\$E128 területen helyezkednek el.

billentyűk magukban: \$E026-\$E065  
billentyűk SHIFT-tel: \$E67-\$E0A6  
billentyűk C=-ral: \$E0A8-\$E0E7  
billentyűk CTRL-lal: \$E0E9-\$E128

Ez a táblázat a ROM-ban foglal helyet, tehát nem változtathatjuk meg. Ha mégis szükség van arra, hogy az egyes billentyűkhöz rendelt ASCII-kódokat megváltoztassuk, átírhatjuk a dekódoló rutint. Bár ez szintén ROM-ban van, de kezdetére egy vektor mutat, mely a \$0545—\$0546 címen helyezkedik el. Alapértéke \$DB7A.

\$EC-\$ED: aktuális billentyű-dekódertábla kezdete  
\$EE: aktuális billentyűzet-mátrix kód  
\$EF: a billentyűzet-pufferben lévő karakterek száma  
\$0527—\$0530: billentyűzet-puffer  
\$053F: billentyűzet-puffer hossza  
\$0540: billentyűismétlés kapcsoló  
-00: kurzorvezérlők

-\$40: egyik sem  
-\$80: mindegyik  
\$0541: billentyűismétlés számláló  
\$0542: billentyűismétlési sebesség  
\$0543: SHIFT, CTRL, C= jelző  
\$0547: SHIFT/C= letiltás/engedélyezés  
-\$00: engedélyezett  
-\$FF: letiltott  
\$07EB: utoljára lenyomott karakter  
\$07F6: billentyűzet-dekódertábla mutatója

Ennyit kívántam a billentyűzetről szólni, és most rátérnék a cikk másik témájára: a memóriakezelésre. A RAM-okat már az első részben elég részletesen tárgyaltuk, így most inkább a ROM-okat helyezném előtérbe.

A 64K RAM-mal rendelkező gépeknél \$8000-\$FCFF-ig és \$FF40-\$FFFF-ig a memória „több rétegű”. Ez alatt azt értem, hogy „egymáson” helyezkedik el RAM és ROM terület. Hogy egy olvasó-ciklus ezen a területen honnan olvas, az attól függ, hogy melyiket választottuk ki („lapoztuk be”) utoljára. Ezt a célt a TED két regisztere szolgálja: az \$FF3E, melyre bármit írva a ROM lapozódik be, és az \$FF3F, melynek írása a RAM kiválasztását jelenti. Ez a két regiszter úgynevezett „fantomregiszter”, mert csak írásuknak van jelentősége, a beírt adat milyensége közömbös, és olvasásuk hatástalan (nem a beírt adatot olvassuk vissza).

Ha abban a szerencsés helyzetben vagyunk, hogy nemcsak BASIC és KERNAL ROM van gépünkben, tovább bonyolódik a helyzet. Ilyenkor azt is meg kell adni, melyik ROM-ot kívánjuk használni. Erre a célra külön lapozóelektronikát építettek a gépbe. Ez az elektronika az \$FDD0-\$FDDF címeiken érhető el.

FIGYELEM! Nem mindegy, melyikbe írunk! Ez most nem olyan, mint a billentyűzetnél volt, inkább a RAM/ROM lapozáshoz hasonlít. Ezek a címek nem a TED regiszterei, az I/O területen vannak. A RAM és a TED kapcsolatáról már volt szó. Ha ROM-ot választottunk ki, és az átfedő területről olvasunk, a RAM-ok nem kapnak CAS jelet, tehát náluk egy plusz frissítési ciklus játszódik le. Aktiválódik a TED CS0 vagy CS1 jele, attól függően, hogy melyik 16K-s tartományt használjuk (\$8000-\$BFFF vagy \$C000-\$FFFF). Hogy ezek után melyik ROM IC kap chip-select jelet, az a lapozóelektronika dolga. Ismét a FPLA a dolgok fő irányítója. Ha az \$FDD0-\$FDDF címekre hivatkozunk, az FPLA aktiválja az ADDR CLK



(ADDRESS CLOCK — cím órajel; kicsit furcsán hangzik, de a tároló aktiváló jelét órajelnek hívják) jelet. Ennek hatására — írás esetén — eltárolódik a címbusz alsó négy (A0—A3) bitje egy 74LS175-ös típusú IC-ben. Ebből a dekódoló hálózat a CS0 vagy CS1 jelek hatására aktiválja egyik vagy másik kimenetét, és ezzel valamelyik ROM-ot. Az az adatot az adatbuszra teszi, és innen az a proceszorba kerül. A logikai hálózatnak összesen nyolc darab kimenő jele van, tehát összesen  $8 \cdot 16 = 128K$  ROM-ot képes kezelni egy PLUS/4. Ezek:

BASIC	(\$8000-BFFF)
KERNAL	(\$C000-FFFF)
FUNCTION LOW	(\$8000-BFFF)
FUNCTION HIGH	(\$C000-FFFF)
CARTRIDGE1 LOW	(\$8000-BFFF)
CARTRIDGE1 HIGH	(\$C000-FFFF)
CARTRIDGE2 LOW	(\$8000-BFFF)
CARTRIDGE2 HIGH	(\$C000-FFFF)

Természetesen \$FD00-\$FEFF-ig mindig az I/O terület lapozódik be. Közös ezen túl az \$FC00-\$FCFF-ig terjedő terület. Ha ide hivatkozunk, mindig a KERNAL lapozódik be. Ennek oka az, hogy itt helyezkednek el a ROM-okat egymással összekapcsoló rutinok (pl. LONG JUMP — egyik ROM-ból a másikba ugrik), és itt kezdődik és végződik a megszakító rutin is. Ennek hardver megvalósítása az FPLA KERN jelével történik, amely nem a neves színészt, hanem a lapozó-

elektronika azon funkcióját aktiválja, hogy függetlenül az utoljára kiválasztott

ROM-tól a KERNAL kapjon chip-select jelet. A címek és ROM-ok kapcsolata:

Cím	\$8000-BFFF	\$C000-FFFF
\$FDD0	BASIC	KERNAL
\$FDD1	FUNCTION LOW	KERNAL
\$FDD2	CARTRIDGE 1 LOW	KERNAL
\$FDD3	CARTRIDGE2 LOW	KERNAL
\$FDD4	BASIC	FUNCTION HIGH
\$FDD5	FUNCTION LOW	FUNCTION HIGH
\$FDD6	CARTRIDGE 1 LOW	FUNCTION HIGH
\$FDD7	CARTRIDGE2 LOW	FUNCTION HIGH
\$FDD8	BASIC	CARTRIDGE1 HIGH
\$FDD9	FUNCTION LOW	CARTRIDGE1 HIGH
\$FDDA	CARTRIDGE1 LOW	CARTRIDGE1 HIGH
\$FDDB	CARTRIDGE2 LOW	CARTRIDGE1 HIGH
\$FDDC	BASIC	CARTRIDGE2 HIGH
\$FDDD	FUNCTION LOW	CARTRIDGE2 HIGH
\$FDDE	CARTRIDGE1 LOW	CARTRIDGE2 HIGH
\$FDDF	CARTRIDGE2 LOW	CARTRIDGE2 HIGH

Nézünk most egy programot, mely felhasználja a cikk teljes anyagát: egy ROM-COPY-t (2. lista).

Úgy hiszem, eljött a búcsú pillanata. A sorozat itt véget ér. Szeretnék köszönetet mondani önöknek türelmükért, valamint Nagy Attilának és Tamásnak felbecsülhetetlen értékű segítségükért. Remélem a jövőben még viszontlátjuk egymást az újság hasábjain.

#### Felhasznált irodalom:

COMMODORE+4 felhasználói kézikönyv — (c) COMMODORE ELECTRONICS LIMITED, 1984. Babán Gábor—Masa István: Gépi kódú programozás kezdőknek és haladóknak — NOVOTRADE, 1987. Tóth Viktor: A COMMODORE 16-os belső felépítése — NOVOTRADE, 1986. Wilhelm Besenthal—Jens Muus: PLUS/4-kézikönyv az összes tudnivalóval — NOVOTRADE, 1989.

```

00001 0000 ;kukac
00002 0000 * = $7000
00003 7000 x1 = $d0 ;atmeneti
00004 7000 y1 = x1+1 ;koordinatak
00005 7000 ciml = y1+1 ;olvasas/
00006 7000 cimh = ciml+1 ;/irashoz
00007 7000 bteszt = $db70 ;billentyuteszt
00008 7000 irqveg = $ce0e ;gepi megszakitorutin
00009 7000 keret = $ff19
00010 7000 sorl = $d802 ;tablazat sorok
00011 7000 sorh = $d81b ;kezdocimeivel a romban
00012 7000 a2 00 ldx #0
00013 7002 a9 a0 inck lda #$a0 ;init ciklus
00014 7004 9d 00 0c sta $0c00,x ;kepernyo
00015 7007 9d 00 0d sta $0d00,x ;inverz
00016 700a 9d 00 0e sta $0e00,x ;space
00017 700d 9d f8 0e sta $0ef8,x ;feltoltes
00018 7010 a9 10 lda #$10
00019 7012 9d 00 08 sta $0800,x ;fekete szin
00020 7015 9d 00 09 sta $0900,x ;beallitasa
00021 7018 9d 00 0a sta $0a00,x
00022 701b 9d f8 0a sta $0af8,x
00023 701e e8 inx
00024 701f d0 e1 bne inck
00025 7021 a9 80 lda #$80 ;fekete
00026 7023 8d 19 ff sta keret
00027 7026 a2 07 ldx #7
00028 7028 a9 14 tabc lda #20 ;kepernyo kozepe
00029 702a 9d f9 70 sta xtab,x
00030 702d a9 0d lda #13
00031 702f 9d 01 71 sta ytab,x
00032 7032 ca dex

```



○	00033	7033	10	f3		bpl	tabc		○
	00034	7035	78			sei			
○	00035	7036	a2	44		ldx	#<mexak		○
	00036	7038	a0	70		ldy	#>mexak		○
○	00037	703a	8e	14	03	stx	\$0314		○
	00038	703d	8c	15	03	sty	\$0315		○
	00039	7040	58			cli			
○	00040	7041	4c	41	70	vege	jmp vege		○
	00041	7044	ad	00	71	mexak	lda xtab+7	;modositando	○
	00042	7047	85	d0		sta	x1	;koord-ak	○
○	00043	7049	ad	08	71		lda ytab+7		○
	00044	704c	85	d1		sta	y1		
○	00045	704e	a9	df		kle	lda #\$df	;k.le	○
	00046	7050	20	70	db	jsr	bteszt	;figyeles	○
	00047	7053	29	01		and	#1	;teszt	○
○	00048	7055	d0	07		bne	kfel	;nincs lenyomva	○
	00049	7057	a5	d1		lda	y1	;y1=y1+1	○
	00050	7059	18			clc			
○	00051	705a	69	01		adc	#1		○
	00052	705c	85	d1		sta	y1		
○	00053	705e	a9	df		kfel	lda #\$df	;k.fel	○
	00054	7060	20	70	db	jsr	bteszt	;figyeles	○
	00055	7063	29	08		and	#8	;teszt	○
○	00056	7065	d0	09		bne	kjobb	;nincs lenyomva	○
	00057	7067	a5	d1		lda	y1	;y1=y1-1	○
	00058	7069	38			sec			
○	00059	706a	e9	01		sbc	#1		○
	00060	706c	30	02		bmi	kjobb	;ha negativ	○
○	00061	706e	85	d1		sta	y1		○
	00062	7070	a9	bf		kjobb	lda #\$bf	;k.jobb	○
○	00063	7072	20	70	db	jsr	bteszt	;figyeles	○
	00064	7075	29	08		and	#8	;teszt	○
○	00065	7077	d0	07		bne	kbal	;nincs lenyomva	○
	00066	7079	a5	d0		lda	x1	;x1=x1+1	○
	00067	707b	18			clc			
○	00068	707c	69	01		adc	#1		○
	00069	707e	85	d0		sta	x1		
○	00070	7080	a9	bf		kbal	lda #\$bf	;k.bal	○
	00071	7082	20	70	db	jsr	bteszt	;figyeles	○
	00072	7085	29	01		and	#1	;teszt	○
○	00073	7087	d0	09		bne	check	;nincs lenyomva	○
	00074	7089	a5	d0		lda	x1	;x1=x1-1	○
	00075	708b	38			sec			
○	00076	708c	e9	01		sbc	#1		○
	00077	708e	30	02		bmi	check	;ha negativ	○
○	00078	7090	85	d0		sta	x1		○
	00079	7092	a5	d0		check	lda x1		
○	00080	7094	c9	28		cmp	#40		○
	00081	7096	90	04		bcc	chy		○
○	00082	7098	a9	27		lda	#39		○
	00083	709a	85	d0		sta	x1		○
○	00084	709c	a5	d1		chy	lda y1		○
	00085	709e	c9	19		cmp	#25		○
○	00086	70a0	90	04		bcc	utolso		○
	00087	70a2	a9	18		lda	#24		
○	00088	70a4	85	d1		sta	y1		○
	00089	70a6	a2	00		utolso	ldx #0	;utolso	○
○	00090	70a8	20	e2	70	jsr	cimsz	;karakter	○
	00091	70ab	a9	10		lda	#\$10	;feketere	○
	00092	70ad	91	d2		sta	(ciml),y		
○	00093	70af	a2	00		ldx	#0		○
	00094	70b1	bd	fa	70	masc	lda xtab+1,x	;tablazat	○
	00095	70b4	9d	f9	70		sta xtab,x	;elemenkenti	○
○	00096	70b7	bd	02	71		lda ytab+1,x	;hatramozditas	○
	00097	70ba	9d	01	71		sta ytab,x		
○	00098	70bd	e8			inx			○
	00099	70be	e0	07		cpx	#7		
○	00100	70c0	d0	ef		bne	masc		○
	00101	70c2	a6	d0		ldx	x1		
	00102	70c4	a4	d1		ldy	y1		



```

00103 70c6 8e 00 71      stx xtab+7      ;uj koord
00104 70c9 8c 08 71      sty ytab+7      ;beirasa
00105 70cc a2 00          ldx #0
00106 70ce 20 e2 70      focik jsr cimsz      ;cim szamitas
00107 70d1 8a            txa
00108 70d2 0a            asl a           ;szinkod
00109 70d3 0a            asl a           ;eloallitasa
00110 70d4 0a            asl a           ;szin=16*x+1
00111 70d5 0a            asl a
00112 70d6 09 01        ora #1
00113 70d8 91 d2        sta (ciml),y
00114 70da e8            inx
00115 70db e0 08        cpx #8
00116 70dd d0 ef        bne focik
00117 70df 4c 0e ce      irqveg jmp irqveg      ;vege
00118 70e2 bd f9 70      cimsz lda xtab,x      ;x koord.
00119 70e5 48            pha            ;mentes
00120 70e6 bd 01 71      lda ytab,x      ;y koord.
00121 70e9 a8            tay
00122 70ea b9 02 d8      lda sorl,y      ;sorkezdet
00123 70ed 85 d2        sta ciml        ;cime
00124 70ef b9 1b d8      lda sorh,y      ;rom
00125 70f2 29 0b        and #$0b        ;tablazatbol
00126 70f4 85 d3        sta cimh
00127 70f6 68            pla
00128 70f7 a8            tay
00129 70f8 60            rts
00130 70f9              xtab * = *+8
00131 7101              ytab * = *+8

```

end of assembly, error count = 00000

```

bteszt db70      check 7092      chy 709c      cimh 00d3
ciml 00d2      cimsz 70e2      focik 70ce      inck 7002
irqveg ce0e     kbal 7080      keret ff19      kfel 705e
kjobb 7070     kle 704e      masc 70b1      mexak 7044
sorh d81b     sorl d802     tabc 7028      utolso 70a6
vege 7041     x1 00d0      xtab 70f9      y1 00d1
ytab 7101

```

```

00124 1275 000001 0000      ;rom-copier
00125 1277 000002 0000      ;
00126 1278 000003 0000      * = $1100
00127 1279 000004 1100      ;
00128 1279 000005 1100      mon = $ff52
00129 1279 000006 1100      rom = $d0
00130 1279 000007 1100      bank = $fdd0
00131 1279 000008 1100      plot = $fff0
00132 1279 000009 1100      primm = $ff4f
00133 1279 000010 1100      scnclr = $d888.
00134 1284 000011 1100      cursl = $ff0d
00135 1284 000012 1100      cursh = $ff0c
00136 1882 000013 1100      matrix = $db70
00137 128e 000014 1100      innen = $d0
00138 000015 1100      ide = innen+2
00139 000016 1100      border = $ff19
00140 000017 1100      hely = $0ea5
00141 000018 1100      ;
00142 000019 1100      ;
00143 000020 1100      ;
00144 000021 1100      ;
00145 000022 1100      ;
00146 000023 1100 78      sei
00147 000024 1101 20 88 d8    jsr scnclr
00148 000025 1104 18      clc

```



```

00026 1105 a2 03          ldx #3          ;4. sor
00027 1107 a0 0b          ldy #11         ;12. oszlop
00028 1109 20 f0 ff          jsr plot
00029 110c 20 4f ff          jsr primm
00030 110f          ;
00031 110f          ;
00032 110f          ;a menu szovege
00033 110f          ;
00034 110f          ;
00035 110f          52 4f 4d 2d 43 4f 50 49 45 52 20 42 59 20 4b 4f 46 41 0d
          .byt 'rom-copier by kofa',13
          0d 0d 0d 41 20 50 52 4f 47 52 41 4d 20 41 20 4b 49 56 41 4c 41
          53 5a 54 4f 54 54 20 52 4f 4d 2d 4f 54 20 52 41 4d 2d 42 41 0d
00036 1122          .byt 13,13,13,'a program a kivaltasztott rom-ot ra
-BE',13
          20 20 20 20 20 20 4d 41 53 4f 4c 4a 41 20 24 32 30 30 30 2d 54
          4f 4c 20 4b 45 5a 44 4f 44 4f 45 4e 2e
00037 114c          .byt 'masolja $2000-tol kezdoden.'
          0d 0d
00038 116e          .byt 13,13
          31 20 2d 20 53 59 53 54 45 4d 20 52 4f 4d 20 28 42 41 53 49 43
          20 26 20 4b 45 52 4e 41 4c 29 0d
00039 1170          .byt '1 - system rom (basic & kernal)',13
          32 20 2d 20 49 4e 54 45 52 4e 41 4c 20 45 58 50 41 4e 53 49 4f
          4e 20 28 46 55 4e 43 54 49 4f 4e 29 0d
00040 1190          .byt '2 - internal expansion (function)',13
          33 20 2d 20 43 41 52 54 52 49 44 47 45 31 0d
00041 11b2          .byt '3 - cartridge1',13
          34 20 2d 20 43 41 52 54 52 49 44 47 45 32 0d 0d 0d
00042 11c1          .byt '4 - cartridge2',13,13,13
          20 20 20 49 52 44 20 42 45 20 41 20 4b 49 56 41 4c 41 53 5a 54
          4f 54 54 20 52 4f 4d 20 53 5a 41 4d 41 54 21 00
00043 11d2          .byt 'ird be a kivaltasztott rom szamat!',0
00044 11f7          ;
00045 11f7          ;
00046 11f7          ;a szam bekerese
00047 11f7          ;
00048 11f7          ;
00049 11f7 a2 a5          ldx #<hely     ;kurzor a kepre
00050 11f9 a0 0e          ldy #>hely
00051 11fb 8e 0d ff          stx cursl
00052 11fe 8c 0c ff          sty cursh
00053 1201 a9 7f          figyel lda #$7f   ;7. sor
00054 1203 20 70 db          jsr matrix
00055 1206 c9 fe          cmp #$fe      ;0. oszlop
00056 1208 f0 1d          beq system   ;1-est nyomtak
00057 120a c9 f7          cmp #$f7     ;3. oszlop
00058 120c f0 15          beq expans   ;2-est nyomtak
00059 120e a9 fd          lda #$fd     ;1. sor
00060 1210 20 70 db          jsr matrix
00061 1213 c9 fe          cmp #$fe      ;0. oszlop
00062 1215 f0 08          beq cart1    ;3-ast nyomtak
00063 1217 c9 f7          cmp #$f7     ;3. oszlop
00064 1219 d0 e6          bne figyel   ;nem 4-es
00065 121b a2 34          ldx #$34     ;4-es kepernyokod
00066 121d d0 0a          bne ok
00067 121f a2 33          cart1 ldx #$33  ;3-as kepernyokod
00068 1221 d0 06          bne ok
00069 1223 a2 32          expans ldx #$32 ;2-es kepernyokod
00070 1225 d0 02          bne ok
00071 1227 a2 31          system ldx #$31 ;1-es kepernyokod
00072 1229          ;
00073 1229          ;
00074 1229          ;return-t vagy del-t kerunk
00075 1229          ;
00076 1229          ;
00077 1229 8e a5 0e          ok stx hely    ;a szam kiirasa
00078 122c ee 0d ff          inc cursl    ;kurzor jobbra
00079 122f a9 fe          var lda #$fe   ;0.sor
00080 1231 20 70 db          jsr matrix

```



```

00081 1234 c9 fd          cmp #$fd          ;1.oszlop->return
00082 1236 f0 0f          beq masol
00083 1238 c9 fe          cmp #$fe          ;0.oszlop->del
00084 123a d0 f3          bne var
00085 123c a9 20          lda #$20          ;space
00086 123e 8d a5 0e       sta hely          ;szam torles
00087 1241 ce 0d ff       dec cursl         ;kurzor vissza
00088 1244 4c 01 12       jmp figyel        ;uj szam bekerese
00089 1247                  ;
00090 1247                  ;
00091 1247                  ;a bank-kod eloallitasa
00092 1247                  ;
00093 1247                  ;
00094 1247 ca          masol dex          ;valasztott rom szama-1
00095 1248 8a          txa
00096 1249 29 03        and #$03          ;ertek levalasztasa
00097 124b 85 d2        sta ide           ;atmeneti tar
00098 124d 0a          asl a             ;ottel valo szorzas
00099 124e 0a          asl a
00100 124f 18          clc
00101 1250 65 d2        adc ide
00102 1252 aa          tax               ;attoltes
00103 1253                  ;
00104 1253                  ;
00105 1253                  ;masolas
00106 1253                  ;
00107 1253                  ;
00108 1253 a9 0b          lda #$0b
00109 1255 8d 06 ff       sta $ff06         ;kep ki
00110 1258 9d d0 fd       sta bank,x        ;rom belapozasa
00111 125b a0 00          ldy #0            ;mutatok beallitasa
00112 125d a9 80          lda #$80          ;$8000-tol
00113 125f a2 20          ldx #$20          ;$2000-re
00114 1261 84 d0          sty innen
00115 1263 85 d1          sta innen+1
00116 1265 84 d2          sty ide
00117 1267 86 d3          stx ide+1
00118 1269 b1 d0          ciklus lda (innen),y
00119 126b 91 d2          sta (ide),y
00120 126d ee 19 ff       inc border
00121 1270 c8          iny
00122 1271 d0 f6          bne ciklus
00123 1273 e6 d3          inc ide+1
00124 1275 e6 d1          inc innen+1
00125 1277 d0 f0          bne ciklus        ;$ffff-ig ok
00126 1279                  ;
00127 1279                  ;
00128 1279                  ;befejezes
00129 1279                  ;
00130 1279                  ;
00131 1279 8d d0 fd       sta bank          ;system vissza
00132 127c a9 1b          lda #$1b          ;kep be
00133 127e 8d 06 ff       sta $ff06
00134 1281 20 88 d8       jsr scnlr
00135 1284 20 4f ff       jsr primm
          0d 0d 20 20 20 41 20 4d 41 53 4f 4c 41 54 20 4b 45 53 5a 20 21
          0d 00
00136 1287                  .byt 13,13,' a masolat kesz !',13,0
00137 129e 4c 52 ff       jmp mon

end of assembly, error count = 00000

bank      fdd0      border    ff19      cart1    121f      ciklus    1269
cursh     ff0c      cursl    ff0d      expans   1223      figyel    1201
hely      0ea5      ide      00d2      innen   00d0      masol     1247
matrix    db70      mon      ff52      ok       1229      plot      fff0
primm     ff4f      rom      00d0      scnlr    d888      system    1227
var       122f

```



```

00001 0000 * = $2000
00002 2000 mexak = $fffe ;mexakitorutin-vektor
00003 2000 irq = $ff09 ;interrupt erzekeles
00004 2000 keret = $ff19
00005 2000 ;
00006 2000 ;
00007 2000 ;
00008 2000 ;
00009 2000 ;
00010 2000 78 start sei ;interrupt ki
00011 2001 ee 19 ff csik inc keret ;keret csikozasa
00012 2004 2c 09 ff bit irq ;tortent mexak?
00013 2007 10 f8 bpl csik ;nem
00014 2009 a9 20 lda #>tovabb ;pc high
00015 200b 48 pha
00016 200c a9 17 lda #<tovabb ;pc low
00017 200e 48 pha
00018 200f 08 php ;sr
00019 2010 68 pla ;break-bit
00020 2011 29 ef and #$ef ;torlese
00021 2013 48 pha ;vissza
00022 2014 6c fe ff jmp (mexak) ;irq kezeles
00023 JMP START ;UJRA
end of assembly, error count = 00000

csik 2001 irq ff09 keret ff19 mexak fffe
start 2000 tovabb 2017

```

## Szövegszeletelő

SYS49152 hatására 5 karakter vastagságú szöveget görgeti és közben elvágja. SYS49179 hatására a legfelső sort jobbról balra görgeti.

```

100 REM" |-----| "
110 REM" | SZOVEGSZELETELO C-64 | "
120 REM" |-----| "
130 REM" | , , | "
140 REM" | KESZITETTE: BRUCE MAESTRO | "
150 REM" |-----| "
160 :
170 DEFFNA(X)=ASC(MID$(A$,X,1))-48+7*(MID$(A$,X,1)>"@")
180 FOR I=49152 TO 49202 STEP 8:S=0
190 :FOR J=0 TO 7
200 :READ A$
210 :A=16*FNA(1)+FNA(2)
220 :POKE I+J,A
230 :S=S+A
240 :NEXT
250 :READ A$
260 :A=16*FNA(1)+FNA(2)
270 IF(SAND255)=ATHENNEXT:PRINT"OK":GOTO290
280 PRINT"HIBA:";PEEK(63)+256*PEEK(64)
290 PRINTCHR$(147);POKE53280,0:POKE53281,0:POKE646,5
300 PRINT"75ZOVEGSZELETELO...BY.BRUCE.MAESTRO....."
310 GOSUB330
320 FORI=0T090:NEXT:SYS49179:SYS49152:GOTO320
330 PRINT" | T T T T T | "
340 PRINT" | | | | | | "
350 PRINT" | | | | | | "
360 PRINT" | | | | | | "
370 PRINT" + + + + + + "
380 RETURN
390 DATA A9,00,A2,00,AC,27,04,EA,0C
400 DATA CA,BD,27,04,E8,9D,27,04,62
410 DATA CA,E0,00,D0,F3,8C,27,04,24
420 DATA 60,C8,00,A2,00,AC,00,04,7A
430 DATA EA,E8,BD,00,04,CA,9D,00,FA
440 DATA 04,E8,E0,39,D0,F3,8C,27,7B
450 DATA 04,60,FF,FF,FF,FF,FF,5E
460 *****
470 * SYS49152 : A SZOVEGET SZELETELI *
480 * *
490 * SYS49179 A FELSO SORT GORGETI + *
500 *****

```



# Még többet ésszel!

## X. rész

Az első kérdésem úgy szólt, hogy mért jó a négybyte-os előolvasás kétszeri szerepeltetése a 180. és 250. sorokban. A válasz: semmiért, sőt.

Ilyen előolvasásnak akkor van értelme, ha egy ciklus feltételének része a rekordok első részének vizsgálata, vagy valami másért fel akarjuk borítani az olvasás-feldolgozás rendszer egyensúlyát, az olvasás javára. Időnként előfordul ilyen helyzet, amelyet lehetne ugyan formulákba öntve tipizálni, de enélkül is felismerhetők. Itt például úgy tűnik, hogy a „\$” nevű, a tartalomjegyzék kicsit kódolt képét tartalmazó file végén van négy felesleges byte. Mivel a (háttultesztelő) ciklus feltétele a file végének el nemérése, ezért ilyen esetben valóban csak előolvasással akadályozható meg a file végén túlról való olvasás.

Csak hogy a „\$” file végén nincsenek felesleges byte-ok. A múltkor elmondtam, hogy az adatokat 32 byte hosszú egységekben kapjuk, és az utolsó (a szabad terület) üzenet után a file-nak vége van. Az ST értékét tehát a 32-es blokkok után kell megnézni, és helytelen dolog a file-t „túlolvasni”. Ezzel ugyanis kockáztatjuk, hogy valami hiba vagy egy spéci lemezturbó program miatt az ST értéke már elveszíti a „file vége” értelmű 64-es tartalmat. Egyébként tudni illik, hogy az ST nyolc bitje különböző üzeneteket hordoz. Mivel előfordulhat, hogy a file vége elérésével egyidőben beüt valami adatátviteli hiba is, ezért file végét jelezheti a például a 65, a 66 vagy a -64 is. Ilyen esetre jobb az „ST = 64” helyett az „ST AND 64” vagy az „ST” feltételek használata. Ez utóbbival a ciklust minden nem nulla ST megszakítja, de ez általában úgyszólván célszerű.

Elég tehát minden rekord elején kiolvasni a négy byte-ot (1. pl.): De vegyük észre, hogy a 190-es sorban így egymás után beolvasunk 4 majd 28 byte-ot. Ezt értelmesnek látszik felváltani egy 32 byte-os olvasási ciklussal, aminek csak az az ára, hogy a rekordon belül minden pozícióhoz négyet kell adnunk.

Van itt egy kis probléma. Ez a megoldás elméletben nagyon klassz, de kipróbálva valamiért nem működik. Néhány kísérlet a gyanút egyértelműen az ominózus négyesre tereli, ezek olvasásakor valami nem stimmel. Hadd mutassak be az újoncoknak egy közismert Nagy Titkot, amelyről soha többé nem szabad elfeledkezni.

A BASIC hibás. Nem ez az egyetlen ilyen pontja, de talán a legkínosabb. A GET utasítás, ha egy nullás kódú (CHR\$(0)) karaktert olvas el, akkor nem ezt, hanem egy üres karakterláncot (""") ad vissza értéként. Ha tehát egy 32 byte hosszú rekordot olvasok be, amelyben valahol van nullás karakter is, akkor a kapott karakterlánc hossza kisebb lesz 32-nél, és persze a későbbi karakterek mind balra tolódnak. Sajnos csak az a megoldás, ha a GET helyett a 2. példához hasonló szubrutint használunk, vagy valami gépi kódú szubrutint írunk e célra.

Most már igazán nem illik tovább titkolnom, hogy mit tartalmaz a bajkeverő négy byte. Az első kettő csak töltelék, fejléc esetében \$01/04, egyébként \$01/01 lehet. A másik két byte a file hossza, alsó/felső byte alakban. Aki még nem ismeri ezt a szóhasználatot, annak megmagyarázom: a hossz értékét két byte-on ábrázoljuk, és a kisebb helyiértékű (alsó) byte van előbb. Így például egy 258 blokk hosszú file esetében a négy byte értéke \$01/01/02/01. Látszik, hogy ha a hossz egyik byte-ja nulla (pl. 256 blokknál rövidebb a file), akkor a GET utasítás itt hibázik.

Miért beszélek én végig GET utasításról, és miért nem GET#-ről? File-okat az utóbbival kell olvasni, hogy működhessen mégis a program? — szólt a második kérdés. A kulcs a 180. és 250. sorok SYS utasítása, amelyek meghívják a CHKIN és CLRCHN nevű Kernal szubrutinokat.

Talán kezdjük az elején. A gép használata során úgy tűnhet, hogy a PRINT utasítás a képernyőre ír, és a GET vagy INPUT a billentyűzetről olvas. Ehelyett a gép mindig az ún. elsődleges kiviteli eszközre ír, és az elsődleges bemeneti eszköztől olvas. Ezek az eszközök persze általában megegyeznek a képernyővel és a billentyűzettel, de a kapcsolat megváltoztatható. A tárban a DFLTO és DFLTN nevű rekeszekben (C64-en ezek címei 154 és 153) van a megfelelő elsődleges eszközök száma, bekapcsolás után 3 (képernyő) és 0. A PRINT mindig arra az eszközre ír, amelynek a száma a DFLTO rekeszben található. Ha a nyomtatót felkészítjük adatfogadásra és ezt a számot 4-re cseréljük, akkor onnantól minden PRINT eredménye a nyomtatóra megy. Körülbelül ez lesz egy OPEN és egy CMD utasítás együttes hatása is. Átállítható a bemeneti eszköz is, csak ehhez nincs a CMD-hez hasonló utasításunk.

A külső egységekkel való kapcsolatteremtést végző szubrutinok, az operációs rendszer részeként, a memória valamely területén ROM-ba vannak „beégetve”. Ezeknek a szubrutinoknak az összefoglaló neve a Kernal. A név eredetét nem ismerem, talán rövidítés, de a „kernel” angolul magot jelent, itt esetleg az operációs rendszer magját. Tudomásom szerint a Kernal minden Commodore gépen úgy van elhelyezve, hogy belépési pontjai (a meghívható címek) valami jellegzetes helyen csoportosulnak. Talán mert így könnyebb azokat megjegyezni, vagy megtalálni. Nem igazán BASIC tudnivaló, de a legfontosabb Kernal szubrutinok úgy vannak előkészítve, hogy helyettük egyszerű módon beilleszthetjük saját szubrutinjainkat, és még a belépési pontokat sem kell az azokat felhasználó programokban megváltoztatni.

A Kernal szubrutinoknak saját nevük van, amelyet a Commodore cég szabványosított a gépei számára. Ezért mostantól nem említem meg a belépési pontokat, mivel az minden gépen más lehet. A megfelelő számértékek minden tisztességes kézikönyvből kikereshetők.

A Kernal azért érdekes a számunkra, mert BASIC programból is elindíthatók a szubrutinjai, és ily módon néha olyan elemi periféria-műveleteket végezhetünk, amelyek BASIC utasításokkal nem valósíthatók meg. Például a LOAD szubrutin meghívásával betölthető egy file anélkül, hogy a BASIC főprogram előlről kezdődne, lásd LOAD utasítás.

A meghívásra a SYS utasítást vagy az USR függvényt lehet használni. Ekkor a BASIC program értelmezése félbeszakad, elindul a kijelölt gépi kódú program. A RETURN gépi kódú megfelelőjének (assemblyben RTS) hatására a gép visszatér a BASIC programhoz, mint egy rendes szubrutin esetében. A kézikönyvekben olvasható, hogy melyik Kernal szubrutin milyen bemeneti paramétereket kíván, ezek átadásáról híváskor szintén gondoskodnunk kell. A SYS végrehajtásakor a gép a tár 780–783. számú (C64) rekeszeiből feltölti az A, X, Y és SP nevű regisztereket (ezt most már nem magyarázom el), ezek megfelelő értékét tehát nekünk kell a hívás előtt beállítanunk.

A Kernal felhasználására egy másik példa a 180. sorban látható alkalmazás. A C64-ben a 65478-as címen a CHKIN nevű



szubrutin kezdődik. Lényegében ez a CMD utasítás bemeneti megfelelője. Ennek köszönhetően a bemeneti eszköz átváltozik az 1 logikai számú file-ra, a számot a 781. rekeszben adjuk át az X regiszternek, ahogy kell. A továbbiakban a GET utasítás már nem a billentyűzetet olvassa, hanem az 1. számú file-t. Ezt az állapotot szünteti meg a 250. sorban a CLRCHN meghívása, amely nem kíván paramétert.

Kérdés lehet, hogy miért jó, ha az elsődleges eszközök ilyen átirányítását választjuk a file-ok (másodlagos) közvetlen olvasása helyett. Sajnos én nem tudok meggyőző választ adni. Ennek a módszernek valami történeti oka lehet, a kevésbé fejlett nyelvek és gépek archaikus korszakából. Talán azt a lehetőséget igyekszik szolgálni, hogy a kiviteli utasítás egységesen a PRINT lehessen, és a mindenkorai célt szabadon állíthatassuk CMD utasításokkal. Hasonlóan kell értelmezni a beviteli eszközre vonatkozóan is.

Nem hibás ez a módszer, de őszintén ajánlom az elkerülését. A ki- és beviteli eszközöket a rendszer önmaga is gyakran állítja, ezért könnyen kersztezhetjük a géppel egymás szándékát. Az átirányított adatforgalomba, mivel azt nem szabályozzuk közvetlenül, bekerülhet oda nem illő részlet, a gép rendszerüzemetei, az INPUT promptja, és még ki tudja, mi minden. Nagy bosszakovás oka lehet, ha a PRINT szövegében feledékenyen eljuttat színvezérlő karakter a nyomtatáskor például ledob egy lapot vagy valami ördögös nyomtatási módba vált. Szándékosan utoljára említettem meg az, hogy egy GET# utasításról nyomban

```
(1.)
180 ... SYS 64748
190 GET A$, A$, A$, A$: ...
250 IF ST=64 ...

(2.)
1000 GET A$: A$=CHR$(ASC(A$+CHR$(0))) :
      RETURN
```

látszik, hogy valamilyen file-ból olvasok vele, és akkor ehhez tartom magam a program többi részében is (nyitás, zárás, ST, hibaüzenet stb.). Egyetlen olyan I/O utasítás van, amelynek nincs másodlagos eszközre irányuló párja a LIST. Hagyjuk meg tehát a CMD-t és Kernal társait a programlistázás igénytelen feladatára, és a file-ok írását-olvasását végezzük erre kitalált utasításokkal. Vonatkozik ez a mostanáig vizsgált kis programra is, amelynek elemzegetését ezennel befejeztem.

Hódi Gyula

## IRQ sprite

SYS49152 hatására a II-es tárolóban lévő sprite jobbról balra görög. A spritében tárolhatunk üzenetet vagy jegyzetet.

```
100 REM *****
110 REM *      IRQ SPRITE      C-64 *
120 REM *      =====      *
130 REM * AKT:49152  LEALL.:RUN/STOP*
140 REM *      ,      ,      *
150 REM * KESZITETTE: BRUCE MAESTRO *
160 REM *****
170 :S=0:E=0
180 :FOR I= 49152 TO 49241
190 : :READA:POKEI,A:S=S+A
200 : :NEXT
210 : :IF S<> 12533 THEN PRINT"HIBA"
220 : :POKE53288,1
230 : :FOR I= 704TO 766
240 : :READA:POKEI,A:E=E+A
250 : :NEXT
260 :IF E<> 3613 THEN PRINT"HIBA"
270 DATA 120,169,021,141,020,003,169
280 DATA 192,141,021,003,160,000,140
290 DATA 084,192,140,086,192,088,096
300 DATA 169,011,141,249,007,169,003
310 DATA 141,021,208,169,120,141,003
320 DATA 208,238,002,208,234,234,234
330 DATA 234,234,234,234,234,234,234
340 DATA 234,234,234,234,234,234,234
350 DATA 234,234,234,234,234,234,234
360 DATA 169,000,141,032,208,141,033
370 DATA 208,200,192,030,208,002,160
380 DATA 000,140,086,192,076,049,234
390 DATA 000,000,004,006,009,011
400 REM *****
410 REM *      A SPRITE ADATAI      *
420 REM *****
430 DATA 000,000,000,000,000,000,000
440 DATA 000,000,255,255,255,000,000
450 DATA 000,151,161,015,148,033,009
460 DATA 148,033,009,247,161,009,148
470 DATA 033,009,148,033,009,151,189
480 DATA 239,000,000,000,255,255,255
490 DATA 000,000,000,000,000,000,000
500 DATA 000,000,000,000,000,000,000
510 DATA 000,000,000,000,000,000,000
```



# JACKIE UTILITITES C-64

## II. rész

A C64-esben lévő VIC-II grafikus chip rasztermegszakítási lehetőségeit BASICból is (!) kihasználhatjuk a 4. listán található „RATER HELP” rutinnal. A képernyő 0–255-ig terjedő rasztersor tartományában tetszőleges számú, de maximum 255 db megszakítást hozhatunk létre, ahol a keret és a háttér színét megváltoztathatjuk. Használata: *SYS 49152, tömb 1 % (0), tömb 2 % (0), tömb 3 % (0)*. Az egész típusú, azonos elemszámú, egydimenziós tömböket előre definiáljuk és töltjük föl a megfelelő értékekkel. De vigyázzunk, csak akkorára dimenzionáljuk, amekkorára részt ki is használunk teljesen belőlük! A „tömb 1”-be a megszakítási helyek rasztersor számát (0–255) kell beírni. Az egész helyek közt legalább 4 rasztersornak, a két legalsó közt pedig (ha a billentyűzetet is használjuk közben) legalább 34-nek (!) kell lennie! A „tömb 2” a keret, a „tömb 3” a háttér színének kódjait tartalmazza, ami akkor íródik be a megfelelő VIC-II regiszterekbe, ha a nála eggyel kisebb indexű „tömb 1” elem által definiált helyen jön létre a megszakítás. A rutint a *SYS 49440* paranccsal állíthatjuk le.

Az 5. lista „IRQ MUSICER” programja a BASIC-ben programozó zenekedvelőknek nyújt nagy segítséget. Akár három szólamú zenét is képes eljátszani a BASIC programunk futása vagy írása közben is. A megszakítás-vezérelt programot a következőképpen bírhatjuk munkára: *SYS 49152, lassítás, F1 % (0), K1 % (0), F2 % (0), K2 % (0), F3 % (0), K3 % (0), H % (0)*. Ismét azonos elemszámú, egydimenziós egész típusú tömböket kell használnunk és kötelezően csak akkorára definiálhatjuk őket, amennyit ki is használunk, mert különben a rutin hibásan működhet! A számozott „F” tömbök az egyes szólamokhoz tartozó frekvenciatáblázatot tartalmazzák. A „K”-kban a szólamok kontrollregisztereinek értékét kell megadni (SID 4, 11., 18. vagy) végül pedig a „H” tömb elemeinek első byte-ja a SID 24., felső byte-ja pedig a SID 23. regiszterébe íródik be minden, a lassításnak (1–255) megfelelő számú IRQ-ban. Amennyiben nem kívánjuk kihasználni mind a három szólamot, a megfelelő tömböket nullákkal töltjük föl!

„MULTI SPRITE POSITIER” a neve a 6. lista programjának. Intróknban, meóknban és a játékok címképernyőinek látványos megalkotásához rendkívül jól felhasználható, ugyanis segítségével egyszerre mozgatható a C64-es által biztosított 8 db sprite egy előre — a felhasználó által — definiált pályán, tetszőleges követési távolsággal. Indítsa tehát: *SYS 49152, X % (0), y % (0), lassítás, I % (0)*. Az első két paraméterenként megadott azonos (max. 254!) elemszámú, kötelezően egydimenziós, Integer típusú tömbök tartalmazzák a szellemek pályájának koordinátáit rendre, x, y sorrendben. A „lassítás” funkciója egyértelmű, de ha nullának vesszük, a rutin kikapcsolódik. Az utolsó paraméter egy 8 eleműre (0–7!) definiált integer tömbváltozó, melyen minden szellemhez külön-külön úgy kell adni a követési távolságot. A kipróbálásban segít a „MULTI POSITIER HELP” rutin (7. lista), amelynek segítségével egy ellipszis pályát határozhatunk meg a szellemeknek. Ha a középpontot 172;140-nek, az x;y számokat pedig 148, ill. 90-nek vesszük, a sprite-ok a látható területen a legnagyobb ellipszis pályán fognak mozogni. És mindez természetesen raszter megszakításban! Ha netán egy-egy helyen villanást észlelünk az objektumokban, azt a POKE (140), PEEK (140)-1 paranccsal korrigálhatjuk.

```

100 REM" "
110 REM" | RASTER HELP |"
120 REM" | COMMODORE 64 |"
130 REM" |-----|"
140 REM" |IRTA: JACKIE MAESTRO |"
150 REM" |"
160 :
170 K=49152:V=49463
180 DEFFNA(X)=ASC(MID$(A$,X,1))-48+7*(
(MID$(A$,X,1)>"@")
190 FOR I=K TO V STEP 8:S=0
200 ::FOR J=0 TO 7
210 ::READ A$
220 ::A=16*FNA(1)+FNA(2)
230 ::POKE I+J,A:PRINT "J";I+J
240 ::S=S+A
250 ::NEXT
260 ::READ A$
270 ::A=16*FNA(1)+FNA(2)
280 IF(SAND255)=ATHENNEXT:PRINT"OK":END
290 PRINT"HIBA:";PEEK(63)+256*PEEK(64)
300 DATA 20,BE,C0,A0,06,B1,47,8D,C9
301 DATA AA,02,20,FF,C0,A2,00,A0,CD
302 DATA 00,B1,47,C9,00,F0,03,4C,00
303 DATA 48,B2,C8,B1,47,9D,00,CD,24
304 DATA 20,F1,C0,E8,EC,AA,02,D0,21
305 DATA E6,20,BE,C0,20,FF,C0,A2,05
306 DATA 00,A0,01,B1,47,9D,00,CE,04
307 DATA 20,F1,C0,E8,EC,AA,02,D0,21
308 DATA F0,20,BE,C0,20,FF,C0,A2,0F
309 DATA 00,A0,01,B1,47,9D,00,CF,05
310 DATA 20,F1,C0,E8,EC,AA,02,D0,21
311 DATA F0,AD,00,CF,8D,21,D0,AD,97
312 DATA 00,CE,8D,20,D0,A9,01,8D,82
313 DATA A9,02,A9,00,8D,0E,DC,78,43
314 DATA AD,00,CD,8D,12,D0,AD,11,A7
315 DATA D0,29,7F,8D,11,D0,A9,81,10
316 DATA 8D,1A,D0,A9,8F,8D,14,03,53
317 DATA A9,C0,8D,15,03,58,60,AD,73
318 DATA 19,D0,8D,19,D0,AE,A9,02,B3
319 DATA 6D,00,CF,8D,21,D0,BD,00,C7
320 DATA CE,8D,20,D0,8D,00,CD,8D,62
321 DATA 12,D0,EE,A9,02,E8,EC,AA,F9
322 DATA 02,D0,08,A9,00,8D,A9,02,BB
323 DATA 4C,31,EA,4C,BC,FE,20,FD,8A
324 DATA AE,20,8B,B0,38,A5,47,E9,16
325 DATA 07,85,47,A5,48,E9,00,85,2E
326 DATA 48,A0,00,B1,47,0A,B0,05,9F
327 DATA A2,16,4C,3A,A4,C8,B1,47,A2
328 DATA 0A,90,F5,A0,04,B1,47,C9,F4
329 DATA 02,90,05,A2,12,4C,3A,A4,75
330 DATA 60,18,A5,47,69,02,85,47,9B
331 DATA A5,48,69,00,85,48,60,A0,23
332 DATA 05,B1,47,C9,00,F0,03,4C,05
333 DATA 48,B2,C8,B1,47,CD,AA,02,33
334 DATA D0,F5,18,A5,47,69,07,85,BE
335 DATA 47,A5,48,69,00,85,48,60,CA
336 DATA 78,A9,31,8D,14,03,A9,EA,89
337 DATA 8D,15,03,A9,00,8D,1A,D0,C5
338 DATA A9,01,8D,0E,DC,58,60,00,D9
339 REM =====

```

.RFANY.



```

100 REM" "
110 REM" MULTI SPRITE POSITIER I"
120 REM" | COMMODORE 64 |"
130 REM" |-----|"
140 REM" |IRTA: JACKIE MAESTRO |"
150 REM" |-----|"
160 :
170 K=49152:V=49553
180 DEFFNA(X)=ASC(MID$(A$,X,1))-48+7*
(MID$(A$,X,1)>"@")
190 FOR I=K TO V STEP 8:S=0
200 ::FOR J=0 TO 7
210 :::READ A$
220 :::A=16*FNA(1)+FNA(2)
230 :::POKE I+J,A:PRINT"J";I+J
240 :::S=S+A
250 ::NEXT
260 ::READ A$
270 ::A=16*FNA(1)+FNA(2)
280 IF(SAND255)=ATHENNEXT:PRINT"OK":END
290 PRINT"HIBA:";PEEK(63)+256*PEEK(64)
300 DATA 20,27,C1,A0,06,B1,47,8D,33
301 DATA AA,02,20,68,C1,A2,00,A0,37
302 DATA 00,B1,47,C9,02,90,03,4C,A2
303 DATA 48,B2,9D,00,C7,C8,B1,47,1E
304 DATA 9D,00,C6,20,5A,C1,E8,EC,72
305 DATA AA,02,D0,E3,20,27,C1,20,87
306 DATA 68,C1,A2,00,A0,00,B1,47,63
307 DATA C9,00,F0,03,4C,48,B2,C8,CA
308 DATA B1,47,9D,00,C8,20,5A,C1,98
309 DATA E8,EC,AA,02,D0,E6,20,FD,53
310 DATA AE,20,9E,B7,86,8E,AD,AA,78E
311 DATA 02,85,8C,A9,08,8D,AA,02,FD
312 DATA 20,27,C1,20,68,C1,A2,00,F3
313 DATA A0,00,B1,47,C9,00,F0,03,54
314 DATA 4C,48,B2,C8,B1,47,9D,00,A3
315 DATA C9,20,5A,C1,E8,E0,08,D0,A4
316 DATA E7,A5,8E,D0,17,78,A9,31,53
317 DATA 8D,14,03,A9,EA,8D,15,03,DC
318 DATA A9,00,8D,1A,D0,A9,01,8D,57
319 DATA 0E,DC,58,60,A9,00,85,8F,5F
320 DATA 78,A9,FF,8D,12,D0,AD,11,4D
321 DATA D0,29,7F,8D,11,D0,A9,81,10
322 DATA 8D,1A,D0,A9,00,8D,0E,DC,97
323 DATA A9,C4,8D,14,03,A9,C0,8D,07
324 DATA 15,03,58,60,AD,19,D0,8D,F3
325 DATA 19,D0,E6,8F,A5,8F,C5,8E,E5
326 DATA F0,03,4C,31,EA,A9,00,85,88
327 DATA 8F,A9,00,85,02,8D,10,D0,2C
328 DATA A6,02,BD,00,C9,AA,A5,02,7F
329 DATA 0A,A8,BD,00,C6,99,00,D0,9E
330 DATA BD,00,C8,99,01,D0,8D,00,AC
331 DATA C7,F0,0B,A4,02,B9,89,C1,6B
332 DATA 0D,10,D0,8D,10,D0,E6,02,42
333 DATA A5,02,C9,08,D0,D2,A2,00,BC
334 DATA FE,00,C9,BD,00,C9,C5,8C,9E
335 DATA D0,05,A9,00,9D,00,C9,E8,CC
336 DATA E0,08,D0,EC,4C,31,EA,20,2B
337 DATA FD,AE,20,8B,80,38,A5,47,2A
338 DATA E9,07,85,47,A5,48,E9,00,92
339 DATA 85,48,A0,00,B1,47,0A,B0,1F
340 DATA 05,A2,16,4C,3A,A4,C8,B1,60
341 DATA 47,0A,90,F5,A0,04,B1,47,72
342 DATA C9,02,90,05,A2,12,4C,3A,9A
343 DATA A4,60,18,A5,47,69,02,85,F8
344 DATA 47,A5,48,69,00,85,48,60,CA
345 DATA A0,05,B1,47,C9,00,F0,03,59
346 DATA 4C,48,B2,C8,B1,47,CD,AA,7D
347 DATA 02,D0,F5,18,A5,47,69,07,3B
348 DATA 85,47,A5,48,69,00,85,48,EF
349 DATA 60,01,02,04,03,10,20,40,DF
350 DATA 80,00,00,00,00,00,00,00,80
351 REM =====

```

READY.

```

100 REM" "
110 REM" | IRG MUSICER |"
120 REM" | COMMODORE 64 |"
130 REM" |-----|"
140 REM" |IRTA: JACKIE MAESTRO |"
150 REM" |-----|"
160 :
170 K=49152:V=49626
180 DEFFNA(X)=ASC(MID$(A$,X,1))-48+7*
(MID$(A$,X,1)>"@")
190 FOR I=K TO V STEP 8:S=0
200 ::FOR J=0 TO 7
210 :::READ A$
220 :::A=16*FNA(1)+FNA(2)
230 :::POKE I+J,A:PRINT"J";I+J
240 :::S=S+A
250 ::NEXT
260 ::READ A$
270 ::A=16*FNA(1)+FNA(2)
280 IF(SAND255)=ATHENNEXT:PRINT"OK":END
290 PRINT"HIBA:";PEEK(63)+256*PEEK(64)
300 DATA 20,FD,AE,20,9E,B7,8E,A8,76
301 DATA 02,20,78,C1,A0,05,B1,47,F8
302 DATA C9,00,F0,03,4C,48,B2,C8,CA
303 DATA B1,47,8D,AA,02,20,B9,C1,CB
304 DATA A2,00,A0,00,B1,47,9D,00,D7
305 DATA C6,C8,B1,47,9D,00,C5,20,08
306 DATA AB,C1,E8,EC,AA,02,D0,EA,A6
307 DATA 20,78,C1,20,B9,C1,A2,00,95
308 DATA A0,00,B1,47,C9,00,F0,03,54
309 DATA 4C,48,B2,C8,B1,47,9D,00,A3
310 DATA C7,20,AB,C1,E8,EC,AA,02,D3
311 DATA D0,E6,20,78,C1,20,B9,C1,A9
312 DATA A2,00,A0,00,B1,47,9D,00,D7
313 DATA C9,C8,B1,47,9D,00,C8,20,0E
314 DATA AB,C1,E8,EC,AA,02,D0,EA,A6
315 DATA 20,78,C1,20,B9,C1,A2,00,95
316 DATA A0,00,B1,47,C9,00,F0,03,54
317 DATA 4C,48,B2,C8,B1,47,9D,00,A3
318 DATA CA,20,AB,C1,E8,EC,AA,02,D6
319 DATA D0,E6,20,78,C1,20,B9,C1,A9
320 DATA A2,00,A0,00,B1,47,9D,00,D7
321 DATA CC,C8,B1,47,9D,00,CB,20,14
322 DATA AB,C1,E8,EC,AA,02,D0,EA,A6
323 DATA 20,78,C1,20,B9,C1,A2,00,95
324 DATA A0,00,B1,47,C9,00,F0,03,54
325 DATA 4C,48,B2,C8,B1,47,9D,00,A3
326 DATA CD,20,AB,C1,E8,EC,AA,02,D9
327 DATA D0,E6,20,78,C1,20,B9,C1,A9
328 DATA A2,00,A0,00,B1,47,9D,00,D7
329 DATA CE,C8,B1,47,9D,00,CF,20,1A
330 DATA AB,C1,E8,EC,AA,02,D0,EA,A6
331 DATA A9,00,8D,A7,02,8D,A9,02,17
332 DATA 78,A9,0D,8D,14,03,A9,C1,3C
333 DATA 8D,15,03,58,60,EE,A7,02,F4
334 DATA AD,A7,02,CD,A8,02,F0,03,C0
335 DATA 4C,31,EA,A9,00,8D,A7,02,46
336 DATA AE,A9,02,BD,00,C5,8D,00,68
337 DATA D4,BD,00,C6,8D,01,D4,BD,76
338 DATA 00,C7,8D,04,D4,BD,00,C8,B1
339 DATA 8D,07,D4,BD,00,C9,8D,08,83
340 DATA D4,BD,00,CA,8D,0B,D4,BD,84
341 DATA 00,CB,8D,0E,D4,BD,00,CC,C3
342 DATA 8D,0F,D4,BD,00,CD,8D,12,99
343 DATA D4,BD,00,CE,8D,17,D4,BD,94
344 DATA 00,CF,8D,18,D4,EE,A9,02,E1
345 DATA AD,A9,02,CD,AA,02,D0,05,A6
346 DATA A9,00,8D,A9,02,4C,31,EA,48
347 DATA 20,FD,AE,20,8B,80,38,A5,03
348 DATA 47,E9,07,85,47,A5,48,E9,09
349 DATA 00,85,48,A0,00,B1,47,0A,6F
350 DATA B0,05,A2,16,4C,3A,A4,C8,5F
351 DATA B1,47,0A,90,F5,A0,04,B1,DC
352 DATA 47,C9,02,90,05,A2,12,4C,A7
353 DATA 3A,A4,60,18,A5,47,69,02,AD
354 DATA 85,47,A5,48,69,00,85,48,EF
355 DATA 60,A0,05,B1,47,C9,00,F0,B6
356 DATA 03,4C,48,B2,C8,B1,47,CD,D6
357 DATA AA,02,D0,F5,18,A5,47,69,DE
358 DATA 07,85,47,A5,48,69,00,85,AE
359 DATA 48,60,00,00,00,00,00,00,A8
360 REM =====

```

READY.



```

100 REM" ┌───────────────────────────────────┐ "
110 REM" |   MULTI POSITIER  HELP   | "
120 REM" |           COMMODORE 64   | "
130 REM" |-----| "
140 REM" |   KESZITETTE: JACKIE   | "
150 REM" └───────────────────────────────────┘ "
160 :
170 DIM X%(254),Y%(254),I%(7)
180 POKE 53280,.:POKE 53281,.
190 POKE 646,7:POKE 650,255
200 FOR I=0 TO 7
210 ::I%(I)=10*I
220 ::POKE 53287+I,3
230 ::POKE 2040+I,13
240 NEXT
250 FOR I=832 TO 896
260 ::POKE I,255
270 NEXT

```

```

280 POKE 53275,255: POKE 53269,255
290 PRINT CHR$(147)
300 INPUT "KOZEPPONT X-KOORDINATA":KX
310 INPUT "KOZEPPONT Y-KOORDINATA":KY
320 INPUT "X-IRANYU SUGAR NAGYSAGA":SX
330 INPUT "Y-IRANYU SUGAR NAGYSAGA":SY
340 J=0
350 FOR I=0 TO 2*PI STEP (PI/127)
360 ::X%(J)=COS(I)*SX+KX
370 ::Y%(J)=SIN(I)*SY+KY
380 ::J=J+1
390 NEXT
400 POKE 53280,.:POKE 53281,.
410 SYS49152,X%(0),Y%(0),1,I%(0)
420 END
430 REM =====
READY.

```

```

100 REM *****
110 REM *   SPRITE OSSZEALLITO   C-64*
120 REM *   =====*
130 REM *   , , , , , *
140 REM * KESZITETTE: BRUCE MAESTRO *
150 REM *   (OLASZ ENDRE) *
160 REM *****
170 :
180 PRINT"J":POKE53280,0:POKE53281,0:POKE646,5
190 FORI=53287TO53295:POKEI,1:NEXT
200 POKE650,255
210 :
220 FOR I= 16320 TO 16382
230 ::READA:POKEI,A:S=S+A
240 NEXT
250 IF S<> 1056 THEN PRINT"HIBA"
260 DATA 000,000,000,000,000,000,000
270 DATA 000,000,000,000,000,000,000
280 DATA 000,000,000,000,000,004,000
290 DATA 000,012,000,000,024,000,000
300 DATA 048,000,000,096,000,024,192
310 DATA 000,025,128,000,027,000,000
320 DATA 030,000,000,031,192,000,031
330 DATA 192,000,000,000,000,000,000
340 DATA 000,000,000,000,000,000,000
350 POKE2040,255:F=0:A=0:B=0:C=0:D=0:R=0
360 POKE53269,255
370 X=24:Y=50
380 PRINTTAB(8)"
390 PRINT"1.... 1. SPRITE SHAPE +
400 PRINT"2.... 1. SPRITE SHAPE -
410 PRINT"3.... 2. SPRITE SHAPE +
420 PRINT"4.... 2. SPRITE SHAPE -
430 PRINT"5.... 3. SPRITE SHAPE +
440 PRINT"6.... 3. SPRITE SHAPE -
450 PRINT"7.... 4. SPRITE SHAPE +
460 PRINT"8.... 4. SPRITE SHAPE -
470 PRINT"9.... 5. SPRITE SHAPE +
480 PRINT"0.... 5. SPRITE SHAPE -
490 PRINT"+.... 6. SPRITE SHAPE +
500 PRINT"-.... 6. SPRITE SHAPE -
510 PRINT"K.... END
520 POKE53250,100:POKE53251,70
530 POKE53252,121:POKE53253,70
540 POKE53254,100:POKE53255,91
550 POKE53256,121:POKE53257,91
560 POKE53258,100:POKE53259,113
570 POKE53260,121:POKE53261,113
580 GETA#
590 IFA#="1" THENA=A+1:X=90:Y=55
600 IFA#="2" THENA=A-1:X=90:Y=55
610 IFA#="3" THENB=B+1:X=150:Y=55
620 IFA#="4" THENB=B-1:X=150:Y=55

```

## Sprite összeállító

Ebben a programban maximum 6 spritét állíthatunk össze. Egy nyíl mutatja, hogy hányas számú spritét lapozgatjuk.

1. billentyű lenyomására az első spritét lapozzuk, 2. billentyű lenyomására az első spritét visszafelé lapozzuk.

3., 4. billentyű második sprite lapozás előre, hátra.

5., 6. billentyű harmadik sprite lapozás előre, hátra.

7., 8. billentyű negyedik sprite lapozás előre, hátra.

9., 0. billentyű ötödik sprite lapozás előre, hátra.

+, -. billentyű hatodik sprite lapozás előre, hátra.

K. kilépés

```

630 IFA#="5" THENC=C+1:X=90:Y=80
640 IFA#="6" THENC=C-1:X=90:Y=80
650 IFA#="7" THEND=D+1:X=150:Y=80
660 IFA#="8" THEND=D-1:X=150:Y=80
670 IFA#="9" THENR=R+1:X=90:Y=105
680 IFA#="0" THENR=R-1:X=90:Y=105
690 IFA#="+" THENF=F+1:X=150:Y=105
700 IFA#="-" THENF=F-1:X=150:Y=105
710 IFA#="K" THENPOKE53269,0:END
720 IFAC<0 THENA=0
730 IFAD>255 THENA=255
740 IFC<0 THENB=0
750 IFC>255 THENB=255
760 IFC<0 THENC=0
770 IFC>255 THENC=255
780 IFD<0 THEND=0
790 IFD>255 THEND=255
800 IFF<0 THENF=0
810 IFF>255 THENF=255
820 IFR<0 THENR=0
830 IFR>255 THENR=255
831 POKE53248,X:POKE53249,Y
840 POKE2041,A:POKE2042,B:POKE2043,C:POKE2044,D:POKE2045,R:POKE2046,F
845 POKE2040,255
850 GOTO580

```



# C-64 bővítések

## C-64 IC teszter

Különböző C-64 bővítők, logikai áramkörök építése, tesztelése, javítása során gyakran találkozunk olyan problémával, amikor egy TTL IC paramétereit működőképességét kellene meghatározni. Elvileg ez nem nagy gond. Mivel ezek katalógus áramkörök, egyértelműen adott a működésük igazságtáblázata, ami alapján a teszt elvégezhető. A gyakorlati megvalósítás azonban komoly korlátokba ütközik. Gondoljuk meg, hogy ezen IC-k nagy többsége 14, 16, 18 vagy 20 lábú tokban helyezkedik el. Vegyük alapul a 14 lábú tokot és ennek egy gyakran használt reprezentációját a 7400-át. A bekötését és igazságtáblázatát itt most szándékosan nem adom meg, mert mindenkinek van olyan katalógusa, amelyben ez szerepel. Ha ezt az IC-t tesztelni akarjuk hagyományos módszerrel az igazságtáblája szerint, tápot adunk neki és jelkombinációt kapcsolunk a bemenetekre, közben figyeljük a kimenet válaszát. A fixen felkötött két tápvezetékét leszámítva 12 lábat kell „végigtapogatni”. A gyakorlati megvalósítása ennek nem is olyan egyszerű, pedig itt 4×3-as csoportban végezhetjük el a mérést. A nagyobb gond akkor van, ha nincs igazságtáblánk vagy olyan IC kerül a kezünkbe, amit nem tudunk azonosítani. A saját gyakorlatomból tudom, hogy a szovjet IC-k nagyrésze ilyen és ezeknél a megbízhatóság sem a legjobb. Nem elhanyagolható az a tény sem, hogy egyre több helyen kerülnek selejtezésre a 70-es, 80-as évek berendezései, amelyek tele vannak ilyen TTL áramkörökkel. Ezeket kár lenne szemétté dobni, hiszen nagy részük újra hasznosítható, ha biztosak vagyunk benne, hogy kiforrasztáskor nem sérültek meg. E pár indok alapján kirajzolódik, hogy jó lenne egy olyan „táblázat”, amely tartalmazza sok TTL IC igazságtábláját és kellene egy olyan hardver, ami biztosítja, hogy a táblázat adata minden IC lábára eljusson. Erre a feladatra számítógépet célszerű használni. A működés így már világossá válik. A szoftver tartalmazza min-

den egyes IC igazságtábláját, amit sorban egymás után rákapcsol a katalógusban megadott lábakra és a katalógusban jelzett kimeneteken nézi a válaszfüggvényt. Ahol ez a válaszfüggvény megegyezik a táblázatával az a keresett IC adata, ami alapján kijelölhető a név és a paraméter. Ezután vizsgáljuk meg részletesen, hogy hogyan működik az az áramkör, ami erre a feladatra készült.

### Az IC teszter hardverének működése

Már említettem, hogy a leggyakrabban alkalmazott áramkörök 14, 16, 18 vagy 20 lábúak. Ezeket megvizsgálva kiderül, hogy az esetek nagy többségében

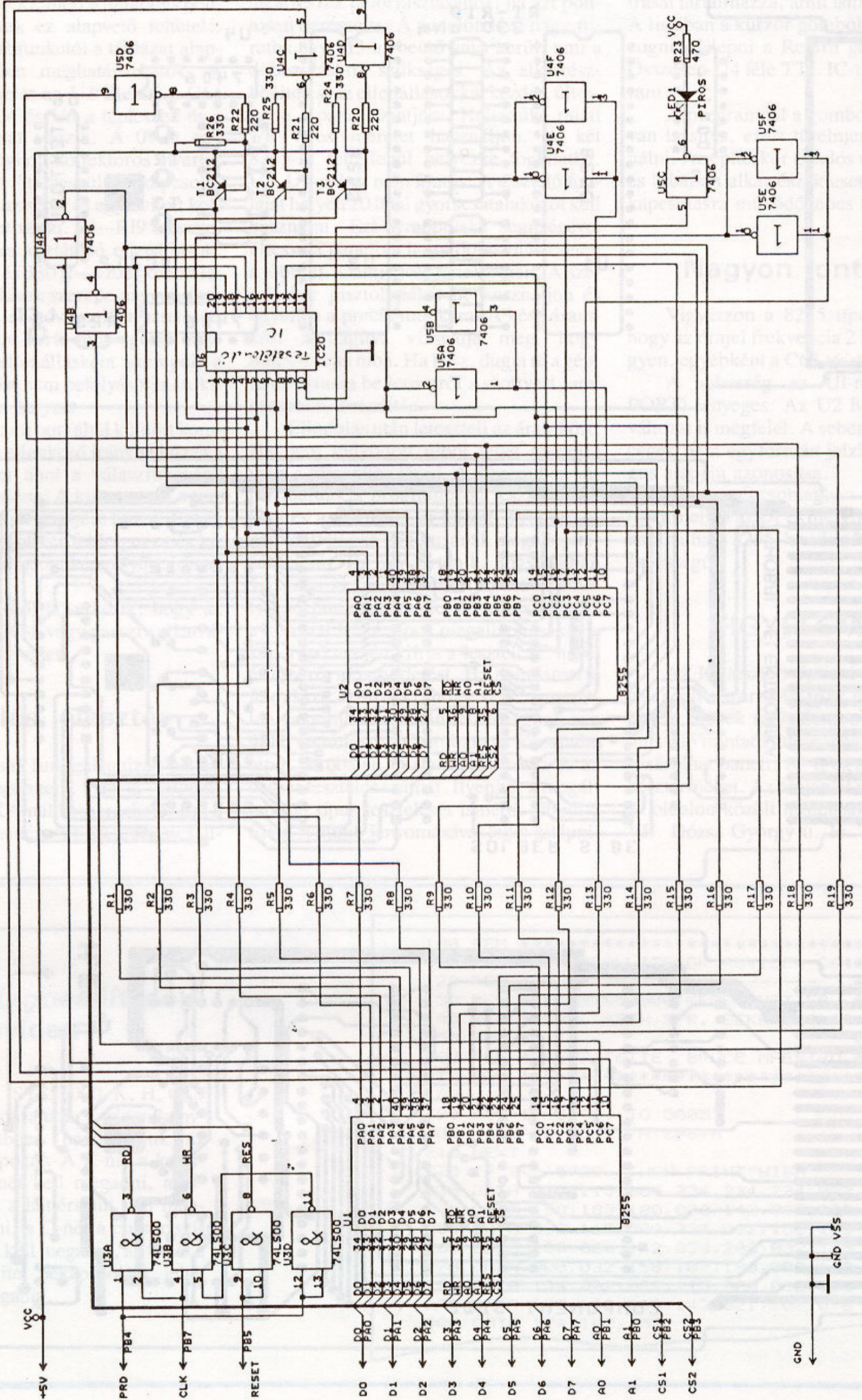
a tápfesz lábak azonos helyen vannak, de néhány kivétel miatt szükséges más helyre is tápot kapcsolni. Így az áramkör földet kapcsolhat a 7, 8, 9, 10, 16, 17 lábakra. +5 V-ot kapcsolhat a 4, 5 és 20 lábakra. A táppontok közül egy, a 10. láb fixen földre köthető, a többi tápellátása csak szoftver útján, kapcsolófokozatokkal lehetséges. Az 1. ábra mutatja a kapcsolási rajzot. Az áramkör lényeges eleme a két db 8255 típusú IC, amely egyenként három 8 bites portot tartalmaz. U1 esetében mindhárom kimeneti üzemmódban dolgozik. U2-nél a C-port 3., 7. bitjei kimeneti, az összes többi bemeneti üzemmódban van. A beállított üzemmód a működés során sosem változik.

Kövessük végig a kimeneti portok lábait az 1. táblázat és a kapcsolási rajz sze-

### A portfunkciók táblázata

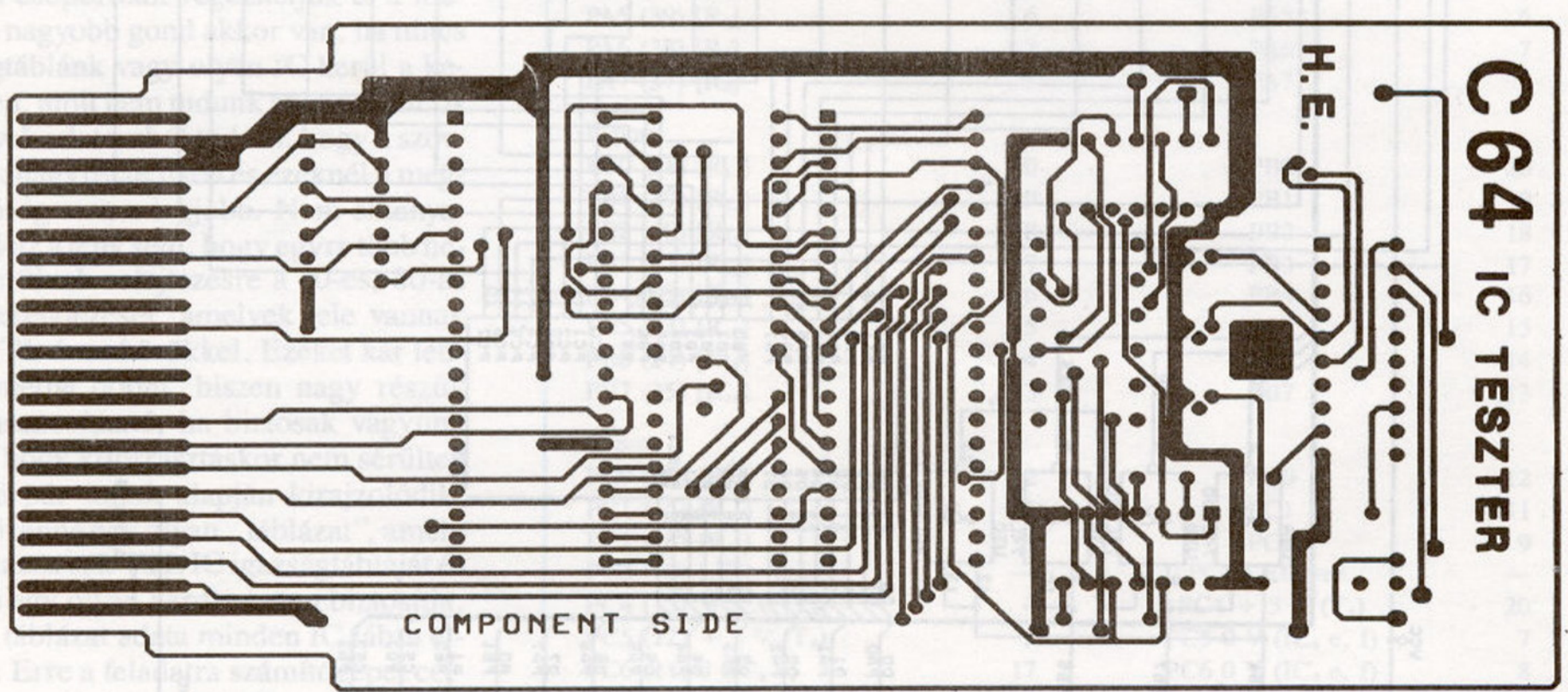
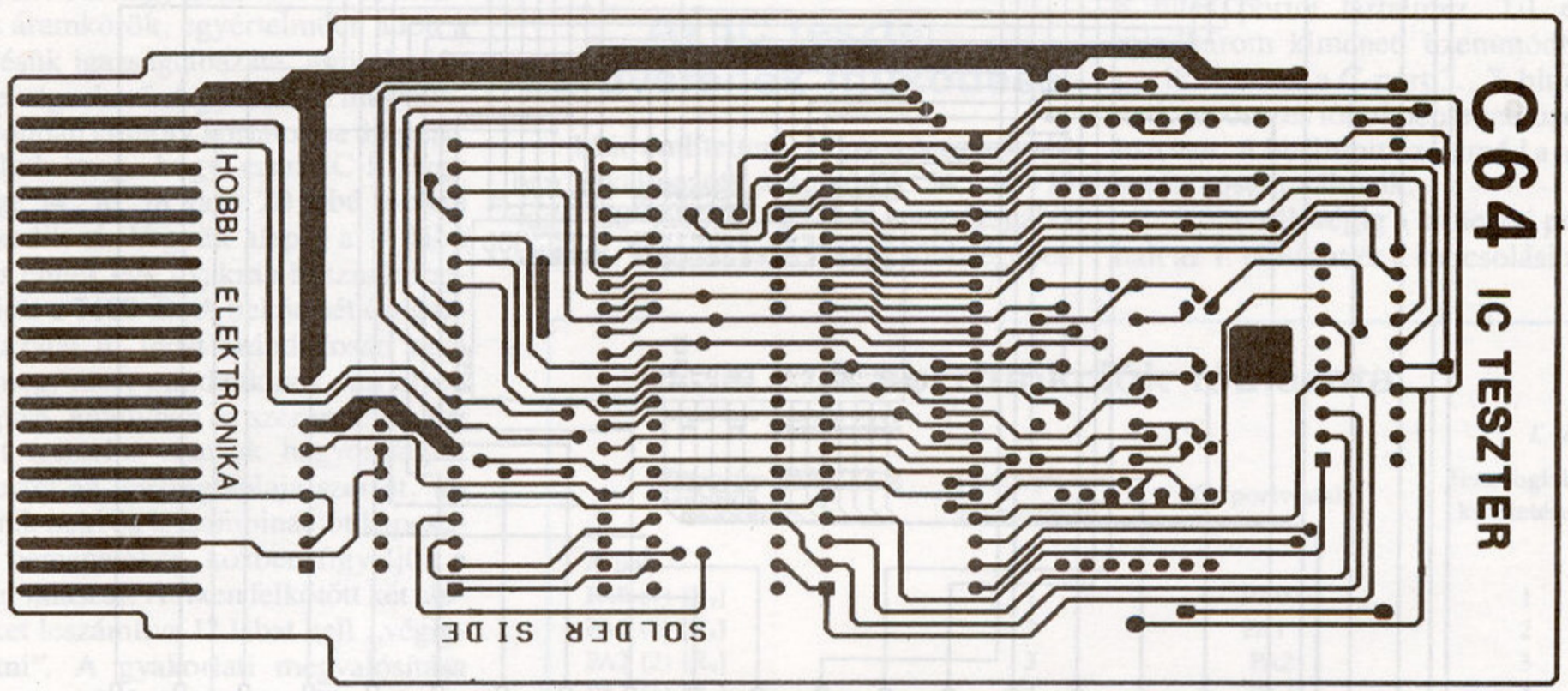
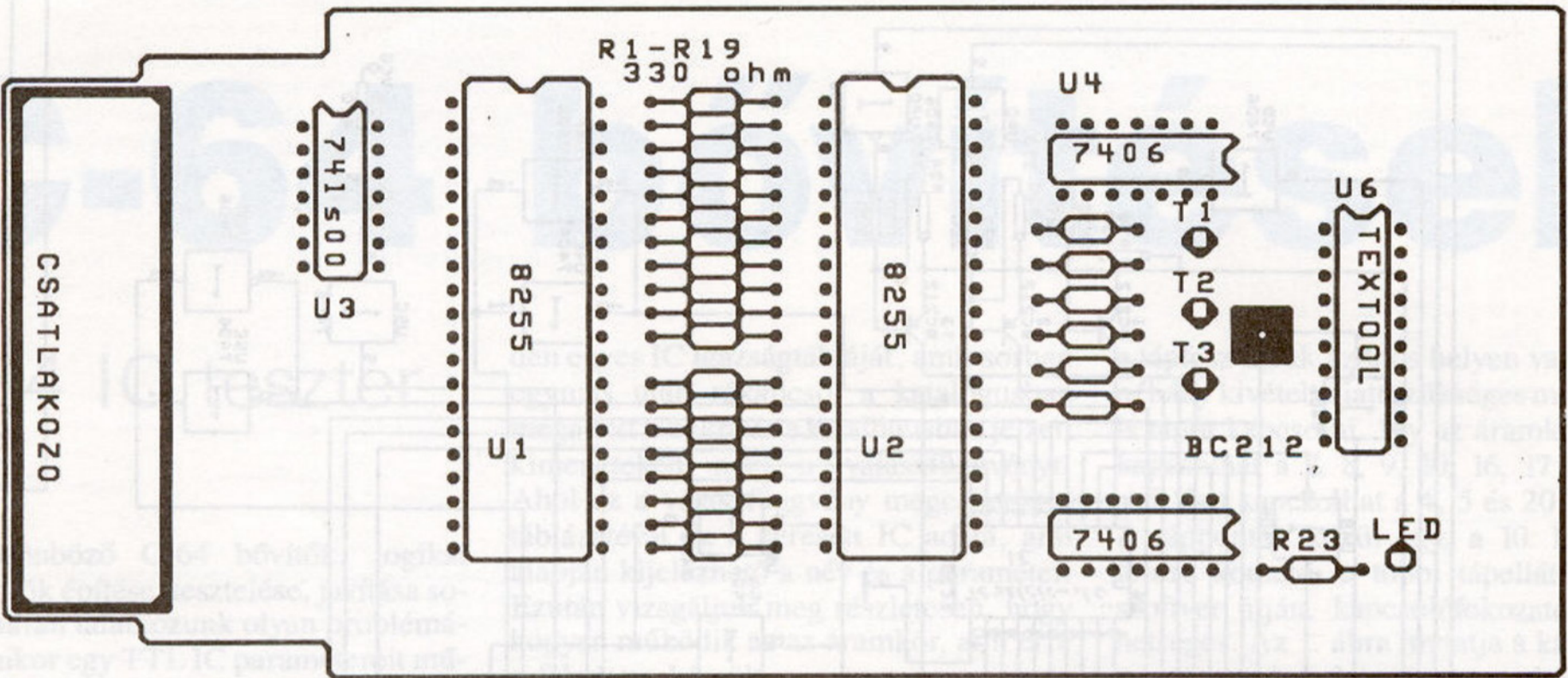
IC <sub>1</sub> portvonalai (lábszám)	Tesztfoglat kivezetéssel	IC <sub>2</sub> portvonalai	Tesztfoglat kivezetései
<i>1. táblázat</i>			
A port			
PA0 (4) [R <sub>7</sub> ]	1	PA0	1
PA1 (3) [R <sub>6</sub> ]	2	PA1	2
PA2 (2) [R <sub>4</sub> ]	3	PA2	3
PA3 (1) [R <sub>2</sub> ]	4	PA3	4
PA4 (40) [R <sub>1</sub> ]	5	PA4	5
PA5 (39) [R <sub>3</sub> ]	6	PA5	6
PA6 (38) [R <sub>5</sub> ]	7	PA6	7
PA7 (37) [R <sub>8</sub> ]	8	PA7	8
B port			
PB0 (18) [R <sub>12</sub> ]	20	PB0	20
PB1 (19) [R <sub>13</sub> ]	19	PB1	19
PB2 (20) [R <sub>14</sub> ]	18	PB2	18
PB3 (21) [R <sub>19</sub> ]	17	PB3	17
PB4 (22) [R <sub>18</sub> ]	16	PB4	16
PB5 (23) [R <sub>17</sub> ]	15	PB5	15
PB6 (24) [R <sub>16</sub> ]	14	PB6	14
PB7 (25) [R <sub>15</sub> ]	13	PB7	13
C port			
PC0 (14) [R <sub>9</sub> ]	12	PC0	12
PC1 (15) [R <sub>19</sub> ]	11	PC1	11
PC2 (16) [R <sub>11</sub> ]	9	PC2	9
PC3 (17) LED vezérlés	—	PC2 LED vez.	—
PC4 (13) + 5 V (T <sub>3</sub> )	4	PC4 + 5 V (T <sub>1</sub> )	20
PC5 (12) + 5 V (T <sub>2</sub> )	5	PC5 0 V (IC <sub>5</sub> e, f)	7
PC6 (11) 0 (IC <sub>4</sub> b)	17	PC6 0 V (IC <sub>4</sub> e, f)	8
PC7 (10) 0 (IC <sub>4</sub> a)	16	PC7 0 V (IC <sub>5</sub> a, b)	9





Title	IC tester C64-hex
Size	B
Document Number	BJ881225-07
REV	1
Date	January 4, 1991
Sheet	1 of 1







rint! Látszólag ez semmitmondó lépés, felesleges időtöltés lenne, de most semmi esetre se hagyjuk ki, mert a működés pontos megértésének ez alapvető feltétele.

A portok lábfunckciói a táblázat alapján egyértelműen meghatározhatók. A portok vezérlőjelét az U3 illeszti a C64 buszjeleihez. U4 és U5 a tápfeszek és a LED kapcsolását végzik. A 0V-ot egy vagy két 7406 nyitott kollektoros inverter biztosítja. A „+” tápfeszültség kapcsolását egy PNP tranzisztor és egy nyitott kollektoros inverter végzi. R1—R19 a teszteléshez szükséges korlátozó ellenállások, értékük azonos és 330 Ω—470 Ω között lehet. Az ellenállások szerepe nagyon fontos, ezek teszik lehetővé az IC tesztelését. Ha ez bemenetre kerül nincs gond, egyszerű felhúzó ellenállásként szerepel, a logikai működést nem befolyásolja. A kimenetnél más a helyzet.

Az ellenállás a port által küldött kombináció alapján ellenkező irányba húzza a kimenetet, mint ahol a válaszfüggvény szerint lenni kellene. A kimenet ezt a terhelést képes a saját szintjére húzni. Ezt az U2 érzékeli, a szoftver feldolgozza és kiértékelve az eredményt képernyőre írja az IC nevét.

A működésből következik, hogy az áramkört hibás IC-k vagy passzív alkatrészek nem veszélyeztetni.

## Szerelés, élesztés

A teszter csak furatgalvanizált panelre szabad megépíteni. Ez a HOBBI ELEKTRONIKÁ-nál beszerezhető. Ezt a kontúrvonal mentén vágjuk körbe és il-

lesszük pontosan a C64 bővítő portjának csatlakozójához. Csak azután kezdjen az alkatrészek beforrasztásához, ha azt pontosan bereszelte. A panelon lévő nagy furatba egy M3-as beütő anya kerül, ami a dobozolásához szükséges. Az alkatrészbeültetést az ellenállásokkal kezdje, ültesse le a panel szintjére. Helyszűke miatt 1/8 W-os méretet használjon. A két 8255-öt feltétlenül helyezze foglalatba, U3, U5-nél ez nem fontos. A tesztelő foglalat helyén 20 lábú gyorscsatlakozót kell használni. Ezt wrapptüske segítségével ültesse a panelbe, hogy dobozolás esetén a többi IC szintje fölé emelkedjen. A szereléskor pisztolypákát ne használjon és ügyeljen a precíz munkára. A kész áramkört átvilágítva vizsgálja meg, hogy nincs-e rajta hiba. Ha kész, dugja rá a gépre és olvassa be lemezről a szoftvert, ami önturbós, önindító.

Elindulás után leteszteli az áramkört. Ha nem működött újból lehet tesztelni vagy a főmenübe lépni. A főmenüben választhatunk a programozott tesztelés, IC lista és a teszter teszt között. Programozott tesztelés az IC típusának megállapítását jelenti. Ez úgy történik, hogy sorban vizsgálja az összes ismert típust és amelyiknél pontosan megfelelő választ kap, az a keresett IC. A típus megállapításakor a képernyő tetején kiírja a tesztelt IC nevét és egy rövid ismertetést. Ha nem ismerte fel, akkor azt írja ki, hogy az IC ismeretlen vagy hibás. Ezután lehetőségünk van új IC tesztelésre, IC próbára és főmenübe lépni. Próbánál kiírja a jó, a rossz és az összes tesztelés számát. Ilyenkor a megállapított típus tesztelését ismétli. Ebből a SPACE gomb lenyomásával lehet kilépni.

A főmenü következő pontja az IC lista. Ez az összes olyan IC nevét és rövid leírását tartalmazza, amit ismer a program. A listában a kurzor gombokkal lehet mozogni. Kilépni a Return gombbal lehet. Összesen 174 féle TTL IC-t ismer a program.

A programnál a gombok figyelése legyen lassítva, ezért türelmesen kell használni. Az építéskor gondos munka mellett és hibátlan alkatrészek esetében első bekapcsolásra működőképes az áramkör.

## Nagyon fontos!!!

Vigyázzon a 8255 típusára, fontos, hogy az órajel frekvencia 2 MHz felett legyen, egyébként a C64 teszter hibát jelez.

A sebesség az U1-nél (kimeneti PORT) lényeges. Az U2 helyén lassúbb változat is megfelel. A sebességet a gyártó cégek nem egyformán jelzik, azt katalógus alapján azonosítsa.

Az egységcsomagban megjelöltem, hogy melyik az U1 és melyik az U2 helyén használható 8255, ha a két IC nem azonos sebességű.

## FIGYELEM!!!

Az IC tesztet építeni szándékozók-nak segít a szerző HOBBI ELEKTRONIKA üzletének szolgáltatása. Az üzletben működő mintadarabok megtekintése után vásárolhat panelt, IC-t, egységcsomagot, készterméket. Az árajánlat megtalálható a 2. oldalon közölt hirdetésben. Cím: Bp. VII. Dózsa György u. 16. T.: 122-8892.

## Alapbeállítások röviden

SYS50000, K, H, C, S kiadása után a gépen könnyebben beállíthatjuk az alapokat. A K-nál a keretszínét kell megadni, a H-nál a háttérszínét kell megadni, a C-nél a cursor színét kell megadni, az S-nél a sprite bekapcsolást lehet megadni.

```

100 REM *****
110 REM *ALAPBEALLITASOK ROVIDEN-C64*
120 REM *-----*
130 REM *AKT:SYS50000,KERET,HATTER,*
140 REM *CURS.SZIN,SPR.BEKAPCSOLAS*
150 REM * , , *
160 REM * KESZITETTE: BRUCE MAESTRO *
170 REM *****
180 :
190 FOR I= 50000 TO 50051
200 : :READA:POKEI,A:S=S+A
210 NEXT
220 IF S<> 5720 THEN PRINT"HIBA"
230 DATA 032,115,000,234,234,234,032
240 DATA 158,183,180,000,142,032,208
250 DATA 032,115,000,234,032,158,183
260 DATA 180,000,142,033,208,032,115
270 DATA 000,032,158,183,180,000,142
280 DATA 134,002,032,115,000,032,158
290 DATA 183,180,000,142,021,208,076
300 DATA 174,167,153

```



# Pattogó golyók

Olvastam pályázati felhívásukat és szeretnék benevezni a mellékelt programmal. Remélem megfelel a pályázati feltételeknek. A feldolgozott játék eredeti nevét nem ismerem, ezért adtam neki a „Pattogó golyók” fantázianevet. A játék látszólag a dámajátékra hasonlít, azonban itt a játék célja, hogy a lehető legkevesebb lépésből saját bábuinkat átlósan mozgatva, a kiindulási mezőkkel ellentétes oldal utolsó 2 sorába juttassuk. A bábukat kizárólag a sárga üres mezőkre lehet mozgatni átlós irányban úgy, hogy mind a saját, mind az ellenfél bábuait átugorjuk. Az győz, aki hamarabb eléri összes bábujaival az utolsó 2 sort. Ha valamelyik játékos nem tud lépni, akkor kimarad, ha egyik játékos sem, akkor döntetlen a parti. Csak előre lehet haladni a sakktáblaszerű játéktéren, amely  $8 \times 8$  mezőből áll. A programban a gép ellen játszhatunk 2 fokozatban. A fokozatok között „csupán” annyi a különbség, hogy a második fokozatban nem fordulhat elő, hogy a gép ne tudja befejezni a játszmát. A gép az 1. fokozatban 2 stratégiát valósít meg, amelyeknek fontossági sorrendjük van. A legfontosabb szabály a gép számára: azzal a bábuval lépjen, amellyel a legtöbb lépést lehet megtenni, ha több ilyen lépés van, akkor a 2. szabály lép érvénybe: az a figura lép, ame-

lyik a haladási irányhoz viszonyítva a leghátsó pozícióban van. Ha ez is megegyezik, akkor a RND függvény dönti el a lépést. Mivel előfordulhat olyan lépéssorozat, amelyről nem lehet a játszmát befejezni, szükséges volt egy új szabályt bevezetni, amely a „célmező elérési szabály” nevet viseli. Ez a szabály hosszabb számolást igényel, ezért került a 2. fokozatba. Vagyis a lépés csak akkor kivitelezhető, ha ezzel a lépéssel be lehet fejezni a partit. Ez a szabály tehát a legmagasabb rendű, de a 2. fokozatban is csak a 9. lépéstől lép érvénybe. Ez azért van, mivel 8 lépésből még nem lehet olyan szituációt előidézni, amely előnytelen pozícióhoz vezet. Erre és a szabállyal kapcsolatos egyéb megállapításokra „próbálgatásokkal” jöttem rá.

Ezután pedig a programleírás következik. A játékeret  $T\%(9,9)$  tömbben ábrázoltam, ahol az első index a tábla X koordinátája, a 2. index az Y. Az origó a bal alsó sarok. Mind a gép, mind a játékos figuráinak aktuális pozícióját külön tömbben tároltam.  $F\%(16,2)$  tömbben azoknak a figuráknak a pozíciói vannak tárolva, amelyek egy bizonyos lépéshosszal rendelkeznek, ahol  $F\%(X, 0)$  a lépés iránya. Ezek a definíciók a program 1000–1050 soraiban találhatóak. Ezután egy iracionális rész

van a programban, amely beállítja az alapértékeket. A  $T\%$  tömbben 0 a tiltott mezőt, 1 a játékos bábuait, 2 a gép bábuait és 3 az üres mezőket szimbolizálja. A gép lépése a 2000 sortól van elhelyezve. Mivel 8 bábu van, ezért egy ciklus indul 1-től 8-ig, és egy bábu elméletileg 2 irányban tud lépni, ezért a ciklus magja gyakorlatilag 16-szor fut le. Az első fázisban a ciklus meghívja a lépésgenerátort, melynek feladata 1. fokozatban az aktuális lépés hosszának meghatározása  $F\%(0, 0)$ -ban. 2. fokozatban ezenkívül eldönti a gép, hogy a lépéssel befejezhető-e a parti, ha nem, lépésszámnak 0-át ad át. A 2020–2030 sorban a program mindig a leghosszabb lépést jegyzi meg. Ha ezzel megegyező hosszúságú lépés is létezik, akkor ezt is megjegyzi. Tehát ez a rész valósítja meg az 1. szabályt. A 2. szabályt, vagyis a hátrébb lévő figurákat a program 2050–2100 sora válogatja ki, majd ha még mindig több lépés marad, a 2110 sor dönt. Ezután megtörténik a lépés mind a képernyőn, mind a tömbökben, majd a 2180 sorban megvizsgálja a gép, hogy nem nyerte-e meg utolsó lépésével a partit; ha nem, akkor átadja a vezérlést a játékos lépésével foglalkozó résznek 1500 sortól. Szintén az 1500-as sorra lép a program, ha a maximális lépésszám 0, vagyis nincs szabályos lépés, de ebben az esetben a döntetlenjelző magas lesz. A lépésgenerátor úgy működik, hogy a program „képzeletben” addig lépteti a figurát, amíg üres mezőre nem jut vagy 0-ra, de ez azt jelenti, hogy erre már nem léphet. 9600-as sortól van a 2. fokozat főszabálya, az „elérési szabály”. Ez nem számtömbbel dolgozik, hanem a tábla  $t\%$ -ba másolódik soronként. Ezzel a fogással gyorsítottam a programot, mivel a stringműveletek gyorsabbak. A lelke a 9800-tól kezdődő szubrutin, amely kiszámítja, hány figura éri el az aktuális álmezőt. Abból a megfontolásból, hogy az álmezőt csak azok a figurák érik el, amelyek a célmezőtől fölfelé benne vannak abban a háromszög alakban, amelynek csúcsa a célmező. Ezután a program összehasonlítja az egyes célmezőket minimálisan elérő bábuk számával. Ha ennél kevesebb éri el a célmezőt, nem lehet a játszmát befejezni. Ezeket a számokat szintén próbálgatással állapítottam meg. Remélem, ez a programleírás és a programban lévő kommentek elegendőek a program megértéséhez.

Tisztelettel:

Gaszó János

```

10 GOSUB5000:GOTO500
100 GETQ#:IFQ#=""THEN100
110 PRINT"PARTI? I=IGEN"
120 GETQ#:IFQ#=""THEN120
130 IFQ#<"I"THENSYS64738:REM BYE BYE
140 GOTO1100
300 PRINT"KISZABALYT? I=IGEN"
310 GETQ#:IFQ#=""THEN310
320 IFQ#<"I"THEN1000
330 PRINT"KISZABALY"
340 PRINT"KISZABALYT A PIROS GOLYUKAT VEZETI."
350 PRINT"KISZABALYT A FEKETE GOLYUKAT VEZETI."
360 PRINT"SLK KAT SORBA JUTTATNI, A LEGKEVESEBB LEPESSEL."
370 PRINT"PARSOLASZABALY: A GOLYUKAT CSAK FERDIRE LEHET MOZGATNI."
380 PRINT"LEGKEVESEBBI SORBA MEZORE JUTTATNI."
390 PRINT"PARSOLASZABALY: A GOLYUKAT CSAK FERDIRE LEHET MOZGATNI."
400 PRINT"PARSOLASZABALY: A GOLYUKAT CSAK FERDIRE LEHET MOZGATNI."
410 PRINT"TOZTATNI, HA KISZABALYT VAGYUNK AKKOR RETURN"
412 PRINT"CSAK ELRE LEHET HALADNI, HA VALAMELYIK GOLYUK"
414 PRINT"FERDIRE NEM TUD LEPNI AKKOR A MASIK KIVETKIVEN"
416 PRINT"KEZIK, HA EGYIK FERDIRE SEM TUD LEPNI AKKOR"
418 PRINT"DONTETLEN. SOK SZERENCSET!"
420 GETQ#:IFQ#=""THEN420
430 GOTO1000
500 PRINT"PARTI: PATTOGASZABALY"
510 PRINT"PARTI: PATTOGASZABALY"
520 K1=2:Y1=7:X0=8:Y0=1:GOSUB5000
530 GETQ#:IFQ#=""THEN530

```

Játék-  
pályázat











```

6040 POKE54276,129:RETURN:REM HANGHATAS
8000 Y0=9-Y0:Y1=9-Y1:REM LEPESE JATEKOS
8010 POKE53248,X0*24:POKE53249,50+(Y0-1)*24:POKE53287,2:REM SPRITE KOORD
8020 POKE53250,X0*24+10:POKE53251,65+(Y0-1)*24
8030 POKE53269,3:PRINT"X";:IFY0<1THENFORI=1TO((Y0-1)*3):PRINT:NEXT
8035 PRINTTAB((X0-1)*3)"TTT#####TTT#####TTT"
8040 FORI=1TOY0-Y1:GOSUB6000
8050 FORJ=1TO24:POKE53248,PEEK(53248)+(X1-X0)/ABS(X1-X0)
8060 POKE53249,PEEK(53249)-1
8070 POKE53250,PEEK(53250)+(X1-X0)/ABS(X1-X0)
8080 IFJ<7THEN9110
8090 IFJ>18THENPOKE53251,PEEK(53251)-1
8100 POKE53251,PEEK(53251)-1
8110 NEXTJ:GOSUB6020:NEXTI
8115 PRINT"X";:IFY1<1THENFORI=1TO((Y1-1)*3):PRINT:NEXT
8120 PRINTTAB((X1-1)*3)"T";P1#;"#####";P2#;"#####";P3#;POKE53269,0:RETURN
9000 Y0=9-Y0:Y1=9-Y1:REM LEPESE GEP
9010 POKE53248,X0*24:POKE53249,50+(Y0-1)*24:POKE53287,6:REM SPRITE KOORD
9020 POKE53250,X0*24+10:POKE53251,65+(Y0-1)*24
9030 POKE53269,3:PRINT"X";:IFY0<1THENFORI=1TO((Y0-1)*3):PRINT:NEXT
9035 PRINTTAB((X0-1)*3)"TTT#####TTT#####TTT"
9040 FORI=1TOY1-Y0:GOSUB6000
9050 FORJ=1TO24:POKE53248,PEEK(53248)+(X1-X0)/ABS(X1-X0)
9060 POKE53249,PEEK(53249)+1
9070 POKE53250,PEEK(53250)+(X1-X0)/ABS(X1-X0)
9080 IFJ<7THEN9110
9090 IFJ>18THENPOKE53251,PEEK(53251)+1
9100 POKE53251,PEEK(53251)+1
9110 NEXTJ:GOSUB6020:NEXTI
9115 PRINT"X";:FORI=1TO((Y1-2)*3):PRINT:NEXT:PRINT"X";
9120 PRINTTAB((X1-1)*3)P1#;"#####";P2#;"#####";P3#;POKE53269,0:Y1=9-Y1:RETURN
9500 REM LEPESEGENERATOR
9510 A1=B%(FI,1):A2=B%(FI,2):A3=A1:A4=A2:F%(0,0)=0:REM F%(0,0)=LEPESSZAM
9520 A3=A3+IR:A4=A4-1:IFT%(A3,A4)=1ORT%(A3,A4)=2THENF%(0,0)=F%(0,0)+1:GOTO9520
9530 IFT%(A3,A4)=0THENF%(0,0)=0:RETURN
9540 IFT%(A3,A4)=3THENF%(0,0)=F%(0,0)+1
9550 IF F0<20RLS<8THENRETURN
9555 IF F%(0,0)<F%(0,1)THENRETURN
9560 REM2.FOKOZAT
9590 B%(FI,1)=A3:B%(FI,2)=A4
9600 T#=T3#:FORI=1TO8:X=B%(I,1):Y=B%(I,2):K=(Y-1)*8+X
9610 T#=LEFT$(T#,K-1)+CHR$(48+I)+RIGHT$(T#,64-K):NEXTI:REM VIZSGALANDO ALLAS=T#
9620 B%(FI,1)=A1:B%(FI,2)=A2
9630 A6=8:GOSUB9800:IFEZ(8)<1THEN9790:REM A 8. CELMEZOT NEM ERI EL FIGURA!
9640 A6=1:GOSUB9800:IFEZ(1)<2THEN9790
9650 A6=7:GOSUB9800:IFEZ(7)<3THEN9790
9660 A6=4:GOSUB9800:IFEZ(4)<1THEN9790
9670 A6=3:GOSUB9800:A5=ASC(MID$(T#,1,1)):A6=ASC(MID$(T#,3,1))
9680 IF(A5<48AND A6<48)ANDEZ(3)<2THEN9790
9690 IF(A5<48AND A6=48)ANDEZ(3)<3THEN9790
9700 IF(A5=48AND A6<48)ANDEZ(3)<3THEN9790
9710 IF(A5=48AND A6=48)ANDEZ(3)<4THEN9790
9720 RETURN
9790 F%(0,0)=0:RETURN:REM NEM NYERD LEPESE!
9800 EZ(A6)=0:X=A6:Y=1+ABS(A6/2=INT(A6/2)):REM A6 CELMEZO KOORDINATAI
9810 FORI=0TO3:X1=X:Y1=Y+I*2:GOSUB9950:NEXTI:REM A CELMEZO MÖGÖTTI MEZOK VIZSG.
9820 IFX=1THEN9880
9830 FORI=1TOX-1:REM A CELMEZOTOL BALRA LEVO MEZOK
9840 IFY+I>8THEN9870
9850 FORJ=1TOINT(((9-(Y+I))+1)/2):REM X.OSZLOPBAN LEVO MEZOK
9860 X1=X-I:Y1=J*2+I-1:GOSUB9950:NEXTI
9870 NEXTI:IFX=8THENRETURN
9880 FORI=1TO8-X:REM CELMEZOTOL JOBBRA
9890 FORJ=1TOINT(((9-(Y+I))+1)/2)
9900 X1=I+X:Y1=J*2+I-1:GOSUB9950:NEXTI:RETURN
9950 K=(Y1-1)*8+X1:A5=ASC(MID$(T#,K,1))-48
9960 IFA5<0THENEZ(A6)=EZ(A6)+1
9970 RETURN

```

READY.

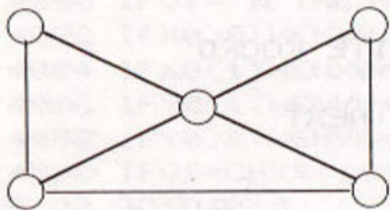


# Beszorítós (DO-GUTI)

## A játék szabályai

A Beszorítós egy kétszemélyes, táblás játék. A játék eredeti neve a hinduk ősi nyelvén egyszerűen párbjátékot jelent. Lényege az ellenfél mozgásának megakadályozása. Mindkét játékosnak 2 bábuja van.

A tábla:



– Az egyik játékos a bábuait a négyzet két alsó sarkába, a másik a két felső, vonallal nem összekötött sarkába helyezi. (Az aszimmetria ellenére a kiinduló helyzet nem kedvez egyik játékosnak sem.)

– A bábuk a vonalak mentén tolatók át az egyik szomszédos mezőre. Bármelyik bábuval lehet kezdeni a játszmát. Az a játékos győz, aki ellenfelét beszorítja, vagyis megakadályozza a további lépéseit.

– Megjegyezzük, hogy két, egymással szimmetrikus helyzet létezik, amelyben bábuinkkal az ellenfél két bábuját beszoríthatjuk.

– A győzelem tévesztésen alapul, azaz ha két jó játékos elég figyelmesen játszik, a végtelenségig tologathatják a bábukat (ez az eset ebben a programban akkor fordulhat elő, ha a nehézségi fokozat kettő (azaz nehéz, és a játékos a SEGÍTSÉG opció szerint teszi meg lépéseit).

## A program kezelése

A program betöltése lemezről: LOAD „BESZORÍTÓS”,8. Indítása: RUN parancssal.

Először a nehézségi fokozatot (1 – könnyű, 2 – nehéz) kell megadni, majd a kezdés jogát (1 – játékos, 2 – gép). Ezután láthatóvá válik a tábla a bábukkal, a menürész és egy nyíl, amit a 2. portba dugott joystickkal mozgathatunk.

A lépés következőképpen történik: Vigyük a nyilat valamelyik saját bábunkra (□ alakú), és nyomjuk meg a tüzgombot. Erre a bábu arra a helyre kerül, ahol eddig az üres mező volt. Ezután automatikusan a gép lép,

majd ismét mozgathatjuk a nyilat egészen addig, amíg valamelyik fél meg nem nyeri a játékot. Ekkor a program kiírja, hogy nyertünk vagy veszítettünk-e és leáll. Újabb játszmahoz egyszerűen gépeljük be a RUN parancsot.

## A menürész

Három menüpont áll rendelkezésre, amit úgy aktiválhatunk, ahogy a lépést (tüzgomb megnyomása az adott szónál).

– Vége menüpont: A játék befejezése. Vigyázat, az állás örökre elvész! Újraindítás lehetséges a RUN parancssal.

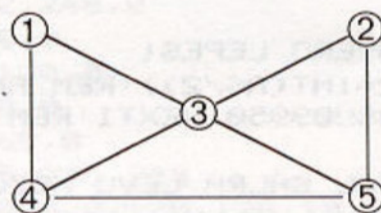
– SEGÍTSÉG menüpont: Aktiválása után az egyik bábunk villogni kezd. Ez a program által javasolt bábu a következő lépéshez, amelyet meglépve nem veszhetjük el a játékot.

– A JÁTÉKRÓL menüpont: Egy rövid ismertetőt ad a játékról és a szabályokról. A tüzgombbal vagy egy tetszőleges billentyű lenyomásával térhetünk vissza a játékba.

## A stratégia

A gép stratégiáját a nehézségi fokozat befolyásolja. Könnyű fokozatban a gép véletlenszerűen választja ki a megfelelő bábút, így könnyű megvenni.

Az öt mezőt megszámoztam 1-től 5-ig a következőképpen:



A gép a nyereshez a következő logikát használja:

Mivel a játékban csak egy üres mező (továbbiakban: luk) létezik, csak azt kell meghatározni, hogy melyik bábuja lépjen a lukra. Az esetek többségében azonban nem is kell gondolkodni, ugyanis csak egy bábuja léphet ilyenkor. Ekkor megkeresi, melyik ez a bábu, és meglépi.

Amikor két bábuval léphet, a gépnek azzal kell lépnie, amellyel a következő lépésben nem lesz beszorítható. Ezt úgy érheti el, ha elkerüli a két bábuja

egymás fölé kerülését. Ez a gyakorlatban azt jelenti, hogy nem azt a lépést lépi, amellyel a bábu a 1. és 4., illetve a 2. és 5. pozícióba kerülnének. Ezenkívül akkor sem egészen mindegy, hogy mit lép, amikor első ránézésre mindegynek tűnik. Ugyanis előbbre hozhatjuk a nyeresést, ha a bábuk a felső terület felé mozognak. Így a gép akkor is számol, ha a luk a 3-as mezőre esik.

Itt jegyzem meg, hogy a program a SEGÍTSÉG menüpont választásakor ugyanilyen módszer szerint számolja ki a megfelelő bábút a játékos számára.

Nagyjából ennyit a stratégiáról. Bővebben a Részletes leírás megfelelő részében olvashatunk erről.

## Részletes programleírás

A program leírásakor megpróbáltam a strukturált programozás elveit követni (már amennyire ezt a 2.0-ás BASIC lehetővé teszi). Így elkülöníthetők egyes szubrutinok (amiket GOSUB-bal hívnak meg) és igyekeztem kevés GOTO-t használni.

A játék ciklusába (130–300. sorok) az SZ (szerva) változó függvényeként lépünk be (játékos vagy gép lépése).

A játékciklus a játékos lépésével kezdődik (a 130. sortól). Először beállítjuk (aktualizáljuk v. frissítjük) az E\$( ) tömböt (2000), majd megállapítjuk, hogy melyik mezőn van a luk (3000). Az eredmény a LU változóba kerül. (Értéke 1-től 5-ig mehet.)

Ezután megnézzük, hogy a játékos melyik bábuja (esetleg mind a kettő) léphet egyáltalán (3200), és ha egyik sem (mindkét KI( ) értéke 0), kiírjuk, hogy veszett, majd befejezzük a program futását (11000. sor).

Ha léphet a játékos, tovább fut a program (140. sor). Meghívunk egy olyan szubrutint, amely a tüzgomb megnyomására tér vissza, egyébként meg a nyilat mozgatja a JOY-nak megfelelően (2500). Ha benyomták a tüzgombot, jön a következő szubrutin (5000). Ez előállít egy OB (object) nevű változót, aminek értéke 0-tól 8-ig terjedhet, és azt mutatja, hogy hol történt a tüzgomb lenyomása (vagyis hova mutatott a játékos):

- 1–5 esetén az egyes mezők valamelyikén,
- 6 esetén a VÉGE menüpontra,
- 7 esetén a SEGÍTSÉG menüpontra,
- 8 esetén pedig az A JÁTÉKRÓL menüpontra.




 Játék-  
pályázat

```

1 REM *****
2 REM *      BESZORITOS      *
3 REM * BY ALEXAY PETER (AXP) *
4 REM *      1992.X.12.      *
5 REM *      FOR THE C=UJSAG  *
6 REM *****
7 :
100 GOSUB 1000 :REM BEALLITASOK
110 GOSUB 1500 :REM TABLA KI
115 GOSUB 1600 :REM SPRITE-OK BE
120 IFSZ=2THEN200 :REM SZERYA A GEPNEL
122 :
124 REM A JATEKOS LEP
126 REM *****
129 :
130 GOSUB 2000 :GOSUB 3000
134 J#="J":GOSUB 3200 :REM KI LEPHET ?
136 IFKI(1)=0ANDKI(2)=0THEN11000
138 : REM * VESZTETT *
140 GOSUB 2500 :REM VARAS A FIRE-RA
145 GOSUB 5000 :REM OB KISZAMOLASA
150 IF OB=0 THEN 140
155 IF OB>5 THEN 400 :REM MENURE MENT
160 IF E$(OB)<>"J" THEN 140
165 IFKI(2)=0THENIFKI(1)=0THENHN=OB:HV=LU:GOTO180
170 IFKI(2)THENHN=OB:HV=LU:GOTO180
175 GOTO 140
180 GOSUB4000 :REM LEPESE
190 :
192 REM GEP JON
194 REM *****
196 :
200 GOSUB 2000 :REM FRISSITES
210 GOSUB 3000 :REM HOL A LUK ?
220 J#="G"
230 GOSUB 3200 :REM MELYIK BABUVAL ?
240 IFKI(2)=0THENIFKI(1)=0THEN10000
245 : REM * A GEP ELVESZTETTE *
250 IFKI(2)=0THENHN=KI(1):GOTO280
255 HN=0
260 IF FO=2 THEN J#="G":GOSUB3400
265 IF HN THEN280
270 VE=INT(RND(1)*2)+1
275 HN=KI(VE)
280 HV=LU :REM A LUKRA LEPUNK
290 GOSUB 4000 :REM LEPESE
300 GOTO 130 :REM A JATEKOS JON
399 :
400 IFOB=6THEN21000 :REM VEGE
410 IFOB=7THEN440 :REM SEGITSEG
420 GOSUB8000 :REM A JATEKRQL
430 GOTO140 :REM VISSZA
439 :
440 S=4:GOSUB7500 :REM INVERZ BE
450 GOSUB2000:GOSUB3000 :REM FRISSITES
460 IFKI(2)=0THENHN=KI(1):GOTO490
470 J#="J":GOSUB3400 :REM GONDOLKOZZ !
480 IFHN=0THENHN=KI(INT(RND(1)*2)+1)
490 GOSUB7000 :REM VILLOSTATAS
500 S=4:GOSUB7530 :REM INVERZ KI
510 GOTO140 :REM VISSZA
989 :
990 REM BEALLITASOK
994 REM *****
999 :
1000 V=53248:RESTORE
1005 DIMSX(5),SY(5),XA(8),XF(8),YA(8),YF(8),MS(8)
1007 DIMM$(5),ES$(4)
1010 POKE53280,0:POKE53281,0
1015 POKE780,52
1020 PRINT"J"CHR$(8)CHR$(142)
1030 FORI=1TO5:READSX(I),SY(I)
1040 NEXTI
1050 DATA48,74,208,74,128,146,48,218,208,218
1060 FORI=1TO8
1070 :READXA(I),XF(I),YA(I),YF(I),MS(I)
1080 NEXTI
1085 DATA32,88,58,106,0
1090 DATA192,248,58,106,0

```

A többi helyre való mutatáskor az OB értéke 0 lesz. Ezután OB kiértékelése folyik.

Először azt vizsgáljuk meg, hogy 0 volt-e. Ha igen, visszaugrunk a várakozási részre (140).

Ha OB értéke nagyobb 5-nél (tehát menüpont), akkor a vezérlést átadjuk a 400-as sornak (lásd a következő bekezdést!)

Ezután már biztos, hogy egy mezőre mutatott a játékos. Megnézzük, hogy azon a mezőn, ahová mutatott, az ő bábuja van-e (160. sor). Ha nem, visszaugrunk a várakozási részre. Most már tudjuk, hogy a játékos valamelyik saját bábuja mutatott. A 4000. sortól kezdődő „LÉPÉS” szubrutin két paramétert vár. Az egyik (HN, azaz honnan) az, hogy melyik mezőről (1–5), a másik (HV, azaz hová) pedig az, hogy melyik mezőre mozgassa a megfelelő bábút.

Ha a játékos egyetlen lehetséges lépését tette meg, így HN az OB értékét veszi fel, HV pedig természetesen LU (azaz a luk) értékét (165. sor). Ugyanez a helyzet, ha a játékos két lépése közül választott (170. sor). Ha csak egy bábuval léphetett, de a másikra mutatott, visszaugrunk a várakozásra (mivel ezzel nem tud lépni). Ezután már minden kész a lépéshez, tehát ez következik (4000). Ezzel a játékkal foglalkozó részt befejeztük.

A 200. sortól kezdődik a GÉP LÉPÉSE rész. Ez ugyanúgy kezdődik, ahogyan a JÁTEKOS LÉPÉSE rész kezdődött (E\$( ) frissítése; LU meghatározása; KI (1) és KI (2) meghatározása, majd ezek ellenőrzése.

Ezután megnézzük, hogy egy vagy két lehetőségünk van-e a lépésre (250. sor). Ha csak egy, úgy HN az első lehetőség, KI (1) értékét veszi fel. Ha két eset van, úgy a fokozat szerint véletlenszerűen vagy gondolkozással előállítjuk, hogy melyik bábuval lépünk.

Ha nehéz fokozatra van beállítva, a program meghívja a MIVEL LÉPJÜNK szubrutint (3400). Ez a rutin beállítja HN értékét a megfelelő mezőre; vagy 0-át ad vissza, ha mindegy, hogy melyikkel lépünk. Ez esetben az RND függvény segítségével választjuk ki, hogy KI (1) vagy KI (2) legyen-e a meglépendő bábu.

Ha készen van HN értéke, HV fölveszi LU értékét (a lukra lépünk), és meghívódik a LÉPÉS rutin (4000). Ezután visszaugrunk a játékciklus elejére.

A 400-tól 510-ig terjedő rész a menüpontokkal foglalkozik:

– „VÉGE” esetén a 21000. sorra ug-



runk, ami elkészön és befejezi a programot.

– „SEGÍTSÉG” esetén a „MIVEL LÉPJÜNK” szubrutin segítségével az előzőekhez hasonló módon kiválasztunk egy mezőt (a HN-ben), majd meghívjuk a VILLOGTATÁS szubrutint (GOSUB 7000), és végül visszaugrunk a várakozási részre (GOTO 140).

– A maradék esetben (ez csak az „A JÁTÉKRÓL” lehet) egy szubrutin segítségével (8000) kiírjuk a magyarázat szöveget és várunk a tűzgombra (vagy egy billentyűre), majd visszaugrunk a 140-es sorra.

Ennyit a főprogramról, és most következzen néhány szubrutin leírása részletesebben.

### Frissítés (2000–2080)

Bemenő paraméterek: — ; kimenő paraméterek: E\$ (1–5). Ez a fontos és gyakran használt szubrutin a táblán szereplő bábukat leíró E\$( ) tömböt tölti fel „J” vagy „G” értékkel. Ezt a SPRITE-ok, azaz az egyes bábuk mindenkor helyzetéből állapítja meg. Az SX (1–5) és az SY (1–5) nevű konstans tömbökben az egyes mezőkhöz tartozó SPRITE-koordináták értéke szerepel.

Az ES\$ (1–4) menü konstans tömb elemei rendre: „G”, „G”, „J”, „J”; azt mutatják, hogy egy bizonyos sorszámú SPRITE melyik fél bábuját jelenti. (A 0-ás sprite a Nyíl, az 1–2. sprite-ok a gép bábui, a 3–4. pedig a játékos bábui).

Az E\$( ) tömb egyik helyére nem kerül adat; itt az első sorban (2000.) elvégzett „0”-val való feltöltés eredményeképpen „0” fog állni (azaz itt van a luk).

### Luk meghatározása (3000–3040)

Bemenő paraméterek: E\$ (1–5); kimenő paraméterek: LU.

### Ki mehet a lukra (3200–3290)

Bemenő adatok: J\$, LU, E\$( ); kimenő adatok: KI (1–2). Ezt a szubrutint talán lépésgenerátornak is hívhatnánk, ugyanis megmondja, hogy egy adott fél mely bábui léphetnek a következő lépésben. Ezeket a lehetséges lépéseket a KI (1–2) változóknak kapjuk meg (mező-értékekben, azaz 1–5 ér-

```

○ 1095 DATA112,168,130,178,0
○ 1100 DATA32,88,202,250,0
○ 1105 DATA192,248,202,250,0
○ 1110 DATA8,40,66,74,1
○ 1115 DATA8,72,82,90,1
○ 1120 DATA8,88,98,106,1
○ 1130 FORI=1T05:READM$(I):NEXTI
○ 1140 DATA34,35,1245,135,234
○ 1150 FORI=1T04:READES$(I):NEXTI
○ 1160 DATA6,G,J,J
○ 1165 :
○ 1170 POKE19,1:INPUT"OFOKOZAT (1-2) = ";FO:POKE19,0:PRINT
○ 1175 IF(FO<>1)AND(FO<>2)THEN1170
○ 1180 INPUT"KI KEZD (1=JATEKOS, 2=GEPE) ";SZ
○ 1185 IF(SZ<>1)AND(SZ<>2)THEN1180
○ 1199 :
○ 1200 FORI=832T0294
○ 1210 :READA:POKEI,A:NEXTI
○ 1215 :
○ 1220 REM SP1-2 ; 13.BLOKK
○ 1225 REM *****
○ 1230 DATA0,16,0,0,56,0,0,56,0
○ 1235 DATA0,124,0,0,124,0,0,254,0
○ 1240 DATA0,255,0,1,255,0,1,255,128
○ 1245 DATA3,255,128,7,255,192,7,255,224
○ 1250 DATA15,255,224,15,255,248,31,255,240
○ 1255 DATA31,255,248,63,255,252,63,255,252
○ 1260 DATA127,255,254,127,255,254,255,255,255
○ 1265 REM MOST JON A 14.BLOKK,SP3-4
○ 1270 REM *****
○ 1270 FORI=896T0958
○ 1280 :POKEI,255:NEXTI
○ 1290 FORI=704T0736
○ 1300 :READA:POKEI,A:NEXTI
○ 1305 :
○ 1310 REM SP0 ; 11.BLOKK
○ 1320 REM *****
○ 1330 DATA254,0,0,252,0,0,248,0,0
○ 1335 DATA252,0,0,254,0,0,220,0,0
○ 1340 DATA143,128,0,7,192,0,3,224,0
○ 1345 DATA1,192,0,0,128,0
○ 1350 FORI=737T0766
○ 1360 :POKEI,0:NEXTI
○ 1370 RETURN
○ 1492 :
○ 1493 REM TABLARAJZOLAS
○ 1495 REM *****
○ 1499 :
○ 1500 T1$=" |-----| ":T2$=" |-----| "
○ 1505 T3$=" |-----| ":T4$=" |-----| "
○ 1510 T5$=" |":T6$=" \":T7$=" /"
○ 1515 S5$=" " :S13$=" "
○ 1520 S6$=" " :S4$=" "
○ 1525 T8$=" |-----| ":T9$=" |"+S5$+T5$+S13$+T5$+S5$+T5$
○ 1527 U1$=S4$+T5$+S6$+T5$+S5$+T5$+S6$+T5$
○ 1530 PRINT " ";T1$;S13$;T1$
○ 1535 PRINTT9$:PRINTT9$:PRINTT9$:PRINTT9$
○ 1540 PRINT " ";T4$;S13$;T4$
○ 1545 PRINTS4$;T5$;" ";T6$;" ";T7$;" ";T5$
○ 1550 PRINTS4$;T5$;S4$;T6$;" ";T7$;S4$;T5$
○ 1555 PRINTS4$;T5$;S5$;T6$;" ";T7$;S5$;T5$
○ 1560 PRINTS4$;T5$;S6$;T1$;S6$;T5$
○ 1565 PRINTU1$:PRINTU1$:PRINTU1$:PRINTU1$
○ 1570 PRINTS4$;T5$;S6$;T2$;S6$;T5$
○ 1575 PRINTS4$;T5$;S5$;T7$;S6$;" ";T6$;S5$;T5$
○ 1580 PRINTS4$;T5$;S4$;T7$;S6$;" ";T6$;S4$;T5$
○ 1585 PRINTS4$;T5$;" ";T7$;" ";T6$;" ";T5$
○ 1590 PRINT " ";T3$;S13$;T3$
○ 1595 PRINTT9$:PRINTT9$
○ 1600 PRINT " ";T5$;S5$;" |":T8$;" |":S5$;T5$
○ 1605 PRINTT9$
○ 1610 PRINT " ";T2$;S13$;T2$
○ 1619 :
○ 1620 PRINT"
○ 1625 PRINTTAB(30);" , "
○ 1630 PRINTTAB(30);"VEGE"
○ 1635 PRINTTAB(30);" , , "
○ 1640 PRINTTAB(30);"SEGITSEG"

```



téssel). Ha csak egy bábuval lehetséges a lépés, akkor a KI (2) értéke 0 lesz; ha egyáltalán nem léphet az illető fél (azaz beszorították), akkor mind a két változó 0 lesz. Az M\$ (1–5) nevű, változó hosszúságú elemekből álló konstans tömbben azt tartjuk nyilván, hogy egy adott mezőre mely mezőkről léphetnek bábuk. A kérdéses mezőt az indexben adjuk meg, és értéként egy olyan sztringet kapunk, amely számokból áll. Például, ha a luk a 2-es mezőn van, akkor ugyebár a 3-as és az 5-ös mezőkről léphetünk rá. Ennek megfelelően M\$ (2) értéke „35”. Ha pedig a 3-as, középső mezőre esik a luk, akkor M\$ (3) értéke = „1245”, ugyanis ide bármely mezőről léphetünk. A rutinnak egy fontos jelzőkaraktert kell még átadni: J\$. Ez azt jelzi, hogy melyik fél bábu a kérdésesek. Így J\$-t a rutin hívása előtt fel kell tölteni „J” vagy „G” értékkel. Ezáltal elértük, hogy ezt a rutint mindkét fél használhassa.

### Mivel lépünk (3400–3510)

Bemenő adatok: LU, J\$, E\$}; kimenő adat: HN.

Ez a rutin tartalmazza a játék által alkalmazott stratégiát. Feladata: egy megadott luk-állás esetére meghatározza, hogy mely mezőről érdemes lépni. A választott mezőszámot (1–5) a HN nevű változóba rakja. Ez a rutin lényegében csupa IF-THEN-ből áll. A J\$ paraméter itt azt jelenti, hogy melyik fél a kérdéses.

A legjobb lépést a luk (LU) 1, 2, 4 és 5 értékénél nem nehéz meghatározni. Csak arra kell figyelni, hogy ne keletkezzen olyan helyzet, ahol a két bábu függőlegesen egymás fölött áll. Tehát pl. ha a luk az 1. mezőre esik, akkor a két lehetséges lépés (4. és 3. mező) közül csak a 4. jó, mivel ha ezt lépjük, akkor az eddig 4-esen álló bábu kerül az 1. mezőre és az 1–3. mezők nem függőlegesek.

Akkor bonyolultabb a helyzet, ha a luk a 3. mezőre, azaz középre esik. Ilyenkor két állást kell megfontolni: Ha az egyik bábu a másik fölött van, akkor ez a nyeresé előtti utolsó állás, ugyanis a felsőt betolva középre a másik játékos nem tud lépni.

A másik megfontolandó állás, ha a két bábu egymással ellentétes oldalon helyezkedik el, azaz az 1. és 5., illetve a 2. és 4. mezőket foglalja le. Ekkor az alsóval érdemes lépni, mert így megdő az esély a nyeresésre. Ugyanis, ha az alsót feltoljuk középre, akkor a másik já-

```

1645 PRINTTAB(30); "      , , "
1650 PRINTTAB(30); "A JATEKROL"
1652 RETURN
1653 :
1655 REM SPRITE-OK BEKAPCSOLASA
1657 REM *****
1658 :
1660 POKEV+6, SX(4):POKEV+7, SY(4)
1665 POKEV+8, SX(5):POKEV+9, SY(5)
1670 POKEV+16,0
1675 POKEV+39,1:POKEV+40,5:POKEV+41,5
1680 POKEV+42,6:POKEV+43,6
1685 POKE2040,11:POKE2041,13:POKE2042,13
1690 POKE2043,14:POKE2044,14
1695 POKEV,120:POKEV+1,100
1700 POKEV+2, SX(1):POKEV+3, SY(1)
1705 POKEV+4, SX(2):POKEV+5, SY(2)
1710 POKEV+6, SX(4):POKEV+7, SY(4)
1715 POKEV+8, SX(5):POKEV+9, SY(5)
1720 POKEV+21,31 :REM BEKAPCSOLAS
1730 RETURN
1990 :
1994 REM FRISSITES
1996 REM *****
1999 :
2000 FORI=1TO5:EF(I)="0":NEXTI
2005 FORJ=1TO5
2010 :FORI=V+2TOV+10STEP2
2020 : :IFPEEK(I)>SX(J)THEN2060
2030 : :IFPEEK(I+1)>SY(J)THEN2060
2040 : :EF(J)=EF((I-V)/2)
2050 : :GOTO2070
2060 :NEXTI
2070 NEXTJ
2080 RETURN
2490 :
2494 REM JOY-KEZELES
2496 REM *****
2499 :
2500 XA=0:YA=0
2510 A=PEEK(56320)
2520 IF (A AND 16)=0 THEN RETURN
2530 IF (A AND 8)=0 THEN XA=4
2540 IF (A AND 4)=0 THEN XA=-4
2550 IF (A AND 2)=0 THEN YA=4
2560 IF (A AND 1)=0 THEN YA=-4
2570 IF XA=-4 THEN 2690
2580 D1=PEEK(V)+XA:JE=0
2590 IF D1<64 THEN 2610
2600 IF (PEEK(V+16) AND 1) THEN 2740
2610 IF D1>255 THEN JE=1:D1=D1-256
2620 POKEV, D1
2630 IF JE<1 THEN 2740
2640 D2=PEEK(V+16) AND 1
2650 D2=1-D2
2660 POKEV+16, PEEK(V+16) AND 2540 OR D2
2670 GOTO2740
2680 D3=PEEK(V)+XA:JE=0
2690 IF D3<24 THEN 2710
2700 IF (PEEK(V+16) AND 1)=0 THEN 2740
2710 IF D3<0 THEN JE=1:D3=D3+256
2720 POKEV, D3
2730 IF J THEN 2640
2740 D4=PEEK(V+1)+YA
2750 POKEV+1, D4
2760 IF D4>48 THEN 2780
2770 POKEV+1, 50
2780 IF D4<230 THEN 2800
2790 POKEV+1, 228
2800 GOTO2500
2990 :
2994 REM LUK MEGHATAROZASA
2996 REM *****
2999 :
3000 FORI=1TO5
3010 :IFE(I)="0" THEN 3030
3020 NEXTI
3030 LU=I
3040 RETURN

```



tékos a bent lévő bábujaival ráléphet az imént keletkezett üres helyre, s így már el is vesztette a játszmat.

Programtechnikailag a legegyszerűbb módon, egymás utáni IF-ekkel valószínűleg meg a HN megkeresése.

A gyorsabb működés érdekében az AND logikai kapcsolat helyett egy sorban történő IF...THEN IF...THEN...utasításokat alkalmaztam. (Például: IF E\$(1) = J\$ THEN IF E\$(4) = J\$ THEN HN = 1.)

Végül még annyit, hogy mint az iménti példán is látszik, ez a rutin is független a játszó féltől. A SEGÍTSÉG opció ugyanis ugyanezt a rutint használja a megfelelő bábu megkeresésére. Ilyenkor elég csak J\$ értékét „J”-re állítani és meghívni a rutint.

### A bábu mozgatása (4000–4080)

Bemenő adatok: HN, HV. Kimenő adat: nincs.

A rutin egy bábut tol el egyik helyről a másikra. A HN mutatja, hogy honnan (1–5), a HV pedig, hogy hová tolja a bábu sprite-ot. A sprite-koordinátákat az SX() és SY() konstans tömbökből veszi, majd a megfelelő sprite-koordinátával egyszerűen felülírja a régit, ezáltal a bábu az új helyre kerül, helyén meg luk támad. Ezért van szükség ezután még a tábla „frissítésére”, ugyanis ez a rutin az E\$( ) tömbhöz nem nyúl.

### Mire mutatott? (Object [OB] előállítás) (5000–5130)

Be —; ki: OB.

Erről a rutinról már volt szó a játékciklus elemzésekor. Még annyit, hogy az XA (1–8), XF (1–8), YA (1–8), YF (1–8) és MS (1–8) tömbkonstansok SPRITE-koordinátát jelölnek. Azt mutatják, hogy az indexben megadott objektum (mező, ill. menüszöveg) mint ablak milyen koordinátákkal bír.

Egy négyzet két átellenes pontjának koordinátáiról és a SPRITE legfelső (MSB) bitjének állásáról van szó. E négyzeten belüli tűzgomb lenyomást a program az adott objektumra való rámutatásként értelmezi.

Azt hiszem, a többi rutin nem igényel különösebb magyarázatot, s az előzőek és a programlista alapján megérthető.

Alexay Péter

```

3190 :
3194 REM KI MEHET A LUKRA ?
3196 REM *****
3199 :
3200 X=1:KI(1)=0:KI(2)=0
3210 A#=M$(LU)
3220 A=LEN(A#)
3230 FORI=1TOA
3240 :B#=MID$(A#,I,1)
3250 :C#=E$(VAL(B#))
3270 :IFC#=J$THENKI(X)=VAL(B#):X=X+1
3280 NEXTI
3290 RETURN
3390 :
3394 REM MIVEL LEPJUNK ?
3396 REM *****
3399 :
3400 IFLU=1THENHN=4:GOTO3510
3410 IFLU=2THENHN=5:GOTO3510
3420 IFLU=3THEN3460
3430 A=LU-3
3440 IFE$(A)=J$THENHN=A:GOTO3510
3450 HN=0:GOTO3510
3460 IFE$(1)=J$THENIFE$(4)=J$THENHN=1:GOTO3510
3470 IFE$(1)=J$THENIFE$(5)=J$THENHN=5:GOTO3510
3480 IFE$(2)=J$THENIFE$(5)=J$THENHN=2:GOTO3510
3490 IFE$(2)=J$THENIFE$(4)=J$THENHN=4:GOTO3510
3500 HN=0
3510 RETURN
3990 :
3994 REM BABU MOZGATASA
3996 REM *****
3999 :
4000 FORI=V+2TOV+10STEP2
4010 :IFPEEK(I)<SX(HN)THEN4050
4020 :IFPEEK(I+1)<SY(HN)THEN4050
4030 :SK=I-V
4040 :GOTO4060
4050 NEXTI
4060 POKEV+SK,SX(HV)
4070 POKEV+1+SK,SY(HV)
4080 RETURN
4990 :
4992 REM MIRE MUTATOTT ?
4994 REM /OBJECT (OB) ELDALLITASA/
4996 REM *****
4999 :
5000 X=PEEK(V):Y=PEEK(V+1)
5010 MS=PEEK(V+16)AND1
5020 OB=0
5030 FORI=1TO8
5040 :IFX<XAK(I)THEN5110
5050 :IFY<YF(I)THEN5110
5060 :IFX<YF(I)THEN5110
5070 :IFY<YF(I)THEN5110
5080 :IFMS<MS(I)THEN5110
5090 :OB=I
5100 :RETURN
5110 NEXTI
5120 OB=0
5130 RETURN
5990 :
5992 REM SEGITSEG
5994 REM *****
5999 :
7000 FORI=V+2TOV+10STEP2
7010 :IFPEEK(I)<SX(HN)THEN7050
7020 :IFPEEK(I+1)<SY(HN)THEN7050
7030 :SK=(I-V)/2
7040 :GOTO7060
7050 NEXTI
7060 A=2+SK
7070 A1=255-A
7080 FOR JA=1 TO 4
7090 :FORQ=1TO100:NEXTQ
7100 :POKEV+21,PEEK(V+21)ANDA1
7110 :FORQ=1TO100:NEXTQ
7120 :POKEV+21,PEEK(V+21)ORR
7130 NEXT JA
7140 RETURN

```







# Zsírozás

A „Zsírozás v 1.3” nevű játékprogram az azonos nevű gyermek-kártyajáték mikroszámítógépes változata Commodore 64-es számítógépre. A programban a számítógép egyfajta játék-stratégiát valósít meg.

A játékot egy személy játszhatja a számítógéppel. A játékszabályok egyszerűek. A 32 lapos magyar kártya összekevert paklijából négy-négy lapot kell kiosztani. A játékot kezdő játékos egy lapot játszik ki. A kijátszott kártya azonos rangú lappal vagy hetessel üthető, egyébként a kijátszott, asztalon fekvő lapokat a kijátszó viszi el. Ha a kijátszott lapot az ellenfél azonos rangú kártyával (pl. alsót alsóval, ászt ásszal stb.) vagy hetessel megütötte, az asztalon lévő lapok szintén azonos rangú lappal vagy hetessel tovább üthetők. Az asztról csak páros számú lapot lehet elvinni. A következő kijátszás joga azé, aki az asztalon lévő lapokat (az ütést) hazavitte.

A játék célja az, hogy az ütésekben minél több tízes és ász (zsír) legyen. Egy-egy elvitt zsír 10 pontot ér. Ha a nyertes 40...70 pontot gyűjtött össze, 1 mérkőzéspontot kap. Kétszeresen számít a győzelem, ha a nyertes 80 pontot ért el, de a vesztes — akit ilyenkor „kopasznak” neveznek — legalább egy, bármilyen ütést hazavitt. Háromszoros a nyerés értéke, ha a vesztes egyetlen ütéssel sem rendelkezik, azaz „csupasz” lett.

A játék során a kártyalapok színét (tök, zöld, makk és piros) nem kell figyelembe venni, így csak a lapfajta — a hetes (7), a nyolcas (8), a kilences (9), a tízes (10), az alsó (A), a felső (F), a király (K) és az ász (ÁS) — játszanak lényegi szerepet a játékban és a stratégia kialakításában is. Hála az egyszerű szabályoknak, a kártyajáték viszonylag könnyen algoritmizálható és a stratégiai egyszerűsítésekkel akár 16K tárkapacitású gépen is megvalósítható.

A gép a bejelentkezést követően a (SPACE) billentyű lenyomására a játékszabályokat ismerteti dióhéjban. Minden más gomb lenyomásával a játék kezdődhet meg. A gép leosztja a lapokat a képernyőre, a bal oldalra a saját lapjait fonákjukkal felfelé, a jobb oldalra a játékos lapjait, látható módon. A játékos a kiválasztott lapot a lap mellett felüntetett billentyűjel — a (Q), a (W), az (E), illetve az (R) — szerinti billentyű megnyomásával játszhatja ki. A kiját-

szott lapok a képernyő közepén jelennek meg, a kijátszás sorrendjében sorszámozva. A pakliból, a talonból még ki nem játszott lapok számát a legfelső képernyősor közepén megjelenő számláló mutaja. A játékban elért pontokat a kártyamezők alatti számlálók számolják. Egy-egy lejátszott partit követően szöveges értékelés következik, majd bármely nyomógomb lenyomására új parti játszható mindaddig, amíg a gépet ki nem kapcsoljuk. A játékot hangeffektusok és egyszerű zene színesítik.

## Programáttekintés

A játékprogram három, egymástól fizikailag is elkülönülő alprogramból áll. Magát a játékot a „C” alprogram tartalmazza. Az „A” és „B” alprogramok előkészítő feladatokat látnak el. A játékot az „A” alprogram behívásával és futtatásával lehet elindítani.

Az „A” és a „B” alprogramokban alkalmazott programozási fogások (érdekesebb POKE-ok, önműködő programbehívás, karaktergenerátor — akár kibővítve vagy átalakítva is) más BASIC programokban is felhasználhatók.

## A „C” alprogram

Mielőtt a vezérlés a játék gerincét képző programsorokra kerülne, az alprogram a fentebb már vázolt módon ismételtellen ellenőrzi a BASIC-munkaterület átkapcsolását. Ha ezt rendben lévőnek találja, elvégzi a játékhoz szükséges további előkészítéseket (a tömbkö dimenzionálásait, a képernyőkép kialakításához szükséges X és Y változók beállítását, a hanggenerátor beállítását stb.) és az elsőnek bejelentkező képernyőképet előállító rutinra adja a vezérlést. A bejelentkezést követően jut a vezérlés az 1000...1400 sorokban látható vezérlő részbe.

A kártyajátékban felmerülő különböző részfeladatok végrehajtására — a moduláris programozás elveit követve — különböző egymástól jól elkülönülő rutinokat készítettem. Ezeket a szükséges sorrendben a vezérlő rész hívja meg.

A játék rutinjai a következő feladatokat látják el:

10000 — BEVEZETÉS

A rutin feltölti a kártyapaklit mo-

dellező P (32) tömböt 1-től 32-ig terjedő számokkal és képernyőre írja a Q-W-E-R szimbólumokat.

11000 — LAPOSZTÁS

A rutin a P (32) tömbből véletlenszerűen válogatva kiosztja a lapokat a gép és a játékos számára is. A stratégiai döntésekhez szükséges szinten a lapok a G1 (4) és a K1 (4) tömbökbe kerülnek. A képernyőn való megjelenítéshez szükséges adatokat a G2\$ (4) és a K2\$ (4) tömbök tartalmazzák. Az 1...32-ig terjedő számokból a két szintnek megfelelő átkódolást 30000 sorszámánál kezdődő rutin végzi el.

A laposztás rutinja felrajzolja a képernyőre a kiosztott lapokat szimbolizáló ábrákat és a talonban maradt lapok számlálóját is.

12000 — KÉZI LAP KIJÁTSZÁSA

A rutin a Q-W-E-R billentyűk valamelyikének lenyomásával kiválasztott lap kijátszásáról, a K1 (0) változóba helyezéséről gondoskodik. Egyúttal a lapot szimbolizáló ábrát törli a képernyő jobb oldaláról.

A rutin renonsz — téves kijátszás — elleni védelmet is tartalmaz. A „tovább ütés” játékhelyzetben ugyanis csak az asztalon lévő legalsó lapfajta vagy az adut képviselő hetes kijátszása szabályszerű. A védelem megakadályozza másfajta lap kijátszását.

13000 — KÖZÉPMEZŐ RAJZOLÁS

A rutin az asztalra kijátszott lapot jeleníti meg a képernyő középrészén.

14000 — A GÉP ÉRTÉKES LAPJAI

A rutin feladata a gép stratégiájában a jelentős, értékes lapok számbavétele. A HE változó a hetesek, a TZ a tízesek, az AS az ászok számát tárolja.

15000 — A GÉP VÁLASZA ÉRTÉKTELEN LAPRA

A rutin a gép stratégiáját tartalmazza arra az esetre, ha az ellenfél értéktelen lapot, vagy esetleg hetest játszott ki.

16000 — A GÉP VÁLASZA ÉRTÉKES LAPRA

A rutin a gép stratégiáját arra az esetre tartalmazza, amikor az ellenfél tízest vagy ászt játszott ki.

17000 — A GÉPI LAP KIJÁTSZÁSA AZ „A” TÖMBBE

A rutin feladata a gép által kijátszott lap elhelyezése az asztalt jelképező A(8) tömbbe és a kijátszott lapot szimbolizáló ábra törlése a képernyő bal oldaláról.

18000 — KÉZ VISZI

A rutin feladata a játékos által elütött lapok közül az értékesek számbavétele, az asztal A(8) tömbjének és a



képernyőn megjelenő ábrájának törlése. A rutin az ütésért járó pontokat a játékos lapjai alatt tünteti fel.

#### 19000 – GÉP VISZI

A rutin feladata lényegében az előzővel azonos, avval a különbséggel, hogy a szükséges műveleteket a gép által elütött lapokra végzi el.

#### 20000 – TOVÁBBÜTÉS VIZSGÁLATA

A rutin „kibic”-ként működik, annak érdekében, hogy az emberi tévedés lehetőségét kiszűrje. Előbb megjeleníti a „Tovább üti (i/n)?” kérdést, majd megfelelő válasz esetén gondoskodik a szöveg törléséről. Téves válasz esetén a „Sajnos önnek nincs ütőlapja!” üzenettel utal a hibás ütési szándéokra.

#### 21000 – GÉPI KEZDÉS

Amennyiben a kijátszott lapokat a gép ütötte, vagy a partit a gép nyerte meg, a következő, illetve a kezdő kijátszás joga őt illeti. Ez a rutin a gép stratégiája erre az esetre.

#### 22000 – SZABAD VÁLASZTÁS

A „szabad választás” játékhelyzetében a gépnek csak akkor kell lapot kijátszania, ha meg tudja ütni az ellenfél által korábban kijátszott lapot. A gép csak akkor üti tovább a kijátszottakat, ha van ugyanolyan lapja, illetve ha a kijátszott lap értékes, akkor is továbbüti,

ha hetessel rendelkezik. A rutin ezt a feltételvizsgálatot végzi el.

#### 23000 – PARTI ÉRTÉKEKELÉS

A játszma végeztével ez a rutin értékeli ki a partit és az eredménynek megfelelő szövegváltozatot írja ki. A rutinból a vezérlés az 55000 sorban kezdődő zenei rutinba kerül át.

#### 30000 – LAPÁTKÓDOLÁS

Korábban már volt arról szó, hogy a stratégiában a lapok színe figyelmen kívül hagyható, ugyanakkor a képernyőn való megjelenítésben ez is fontos. A rutin ezt a kétfajta átkódolást végzi el. A P(32) tömbből kiosztott 1..32 számból előállítja a stratégiai döntésekhez szükséges értékeket (hetes = 1...ász = 8) valamint a képernyőn való megjelenítéshez a színt jelképező karakterrel (tök = [, zöld = font, makk = ], piros = felnyíl) együtt a lap szimbólumát (7, 8, 9, 10, A, F, K és balranyíl = ász).

#### 40000 – CRSR POZICIONÁLÁS

A képernyő különböző pontjainak eléréséhez a képernyőkép kialakításához szükséges cursor pozicionáló rutin. (A rutin a POKE781,X:POKE782,Y:SYS65520 gépi kódú rutinnal helyettesíthető.)

#### 50000 – KÖSZÖNTŐ

Két képernyőképet tartalmazó ru-

tin. A játék legelején bejelentkező szöveget és szükség szerint a játékszabályokat írja föl a képernyőre. A S\$ (24) tömbben elhelyezett szövegeket az 54000 sortól kezdődő kiírató rutin írógépszerűen jeleníti meg.

A rutinból a vezérlés szintén az 55000 sorban kezdődő zenei rutinba kerül át.

#### 55000 – ZENE

A rutin a kísérőzenét állítja elő.

A hangjegyeket és a megszólalási idő arányszámait az 56000 sortól kezdődő adatlista tartalmazza.

A játék v 1.1 verzióját a Texas Instruments TI99/4R típusú gépere, v 1.2 változatát a HT 1080Z típusú gépere készítettem. A C64-re adaptált változat magán viseli az elődök által meghatározott programszerkesztési követelmények nyomait. A program ezért nem tekinthető különösebben optimalizátnak, ez azonban nem jelenti a játékszínvonal lényeges romlását. Úgy hiszem, nemcsak az elektronikus zsugaasztal mellett lehet eltölteni kellemes perceket, de a program esetleges optimalizálásával is, illetve – kihasználva a C64-esnek az említett elődöknél nagyobb tárhelykapacitását – a gép stratégiájának további erősítésével is lehetne kísérletezni.

```

001  REM *****
002  REM *****
003  REM *****
004  REM *****
005  REM *****
006  REM *****
007  REM *****
008  REM *****
009  REM *****
010  REM *****
011  REM *****
012  REM *****
013  REM *****
014  REM *****
015  REM *****
016  REM *****
017  REM *****
018  REM *****
019  REM *****
020  REM *****
021  REM *****
022  REM *****
023  REM *****
024  REM *****
025  REM *****
026  REM *****
027  REM *****
028  REM *****
029  REM *****
030  REM *****
031  REM *****
032  REM *****
033  REM *****
034  REM *****
035  REM *****
036  REM *****
037  REM *****
038  REM *****
039  REM *****
040  REM *****
041  REM *****
042  REM *****
043  REM *****
044  REM *****
045  REM *****
046  REM *****
047  REM *****
048  REM *****
049  REM *****
050  REM *****
051  REM *****
052  REM *****
053  REM *****
054  REM *****
055  REM *****
056  REM *****
057  REM *****
058  REM *****
059  REM *****
060  REM *****
061  REM *****
062  REM *****
063  REM *****
064  REM *****
065  REM *****
066  REM *****
067  REM *****
068  REM *****
069  REM *****
070  REM *****
071  REM *****
072  REM *****
073  REM *****
074  REM *****
075  REM *****
076  REM *****
077  REM *****
078  REM *****
079  REM *****
080  REM *****
081  REM *****
082  REM *****
083  REM *****
084  REM *****
085  REM *****
086  REM *****
087  REM *****
088  REM *****
089  REM *****
090  REM *****
091  REM *****
092  REM *****
093  REM *****
094  REM *****
095  REM *****
096  REM *****
097  REM *****
098  REM *****
099  REM *****
100  REM *****
101  REM *****
102  REM *****
103  REM *****
104  REM *****
105  REM *****
106  REM *****
107  REM *****
108  REM *****
109  REM *****
110  REM *****
111  REM *****
112  REM *****
113  REM *****
114  REM *****
115  REM *****
116  REM *****
117  REM *****
118  REM *****
119  REM *****
120  REM *****
121  REM *****
122  REM *****
123  REM *****
124  REM *****
125  REM *****
126  REM *****
127  REM *****
128  REM *****
129  REM *****
130  REM *****
131  REM *****
132  REM *****
133  REM *****
134  REM *****
135  REM *****
136  REM *****
137  REM *****
138  REM *****
139  REM *****
140  REM *****
141  REM *****
142  REM *****
143  REM *****
144  REM *****
145  REM *****
146  REM *****
147  REM *****
148  REM *****
149  REM *****
150  REM *****
151  REM *****
152  REM *****
153  REM *****
154  REM *****
155  REM *****
156  REM *****
157  REM *****
158  REM *****
159  REM *****
160  REM *****
161  REM *****
162  REM *****
163  REM *****
164  REM *****
165  REM *****
166  REM *****
167  REM *****
168  REM *****
169  REM *****
170  REM *****
171  REM *****
172  REM *****
173  REM *****
174  REM *****
175  REM *****
176  REM *****
177  REM *****
178  REM *****
179  REM *****
180  REM *****
181  REM *****
182  REM *****
183  REM *****
184  REM *****
185  REM *****
186  REM *****
187  REM *****
188  REM *****
189  REM *****
190  REM *****
191  REM *****
192  REM *****
193  REM *****
194  REM *****
195  REM *****
196  REM *****
197  REM *****
198  REM *****
199  REM *****
200  REM *****
201  REM *****
202  REM *****
203  REM *****
204  REM *****
205  REM *****
206  REM *****
207  REM *****
208  REM *****
209  REM *****
210  REM *****
211  REM *****
212  REM *****
213  REM *****
214  REM *****
215  REM *****
216  REM *****
217  REM *****
218  REM *****
219  REM *****
220  REM *****
221  REM *****
222  REM *****
223  REM *****
224  REM *****
225  REM *****
226  REM *****
227  REM *****
228  REM *****
229  REM *****
230  REM *****
231  REM *****
232  REM *****
233  REM *****
234  REM *****
235  REM *****
236  REM *****
237  REM *****
238  REM *****
239  REM *****
240  REM *****
241  REM *****
242  REM *****
243  REM *****
244  REM *****
245  REM *****
246  REM *****
247  REM *****
248  REM *****
249  REM *****
250  REM *****
251  REM *****
252  REM *****
253  REM *****
254  REM *****
255  REM *****
256  REM *****
257  REM *****
258  REM *****
259  REM *****
260  REM *****
261  REM *****
262  REM *****
263  REM *****
264  REM *****
265  REM *****
266  REM *****
267  REM *****
268  REM *****
269  REM *****
270  REM *****
271  REM *****
272  REM *****
273  REM *****
274  REM *****
275  REM *****
276  REM *****
277  REM *****
278  REM *****
279  REM *****
280  REM *****
281  REM *****
282  REM *****
283  REM *****
284  REM *****
285  REM *****
286  REM *****
287  REM *****
288  REM *****
289  REM *****
290  REM *****
291  REM *****
292  REM *****
293  REM *****
294  REM *****
295  REM *****
296  REM *****
297  REM *****
298  REM *****
299  REM *****
300  REM *****
301  REM *****
302  REM *****
303  REM *****
304  REM *****
305  REM *****
306  REM *****
307  REM *****
308  REM *****
309  REM *****
310  REM *****
311  REM *****
312  REM *****
313  REM *****
314  REM *****
315  REM *****
316  REM *****
317  REM *****
318  REM *****
319  REM *****
320  REM *****
321  REM *****
322  REM *****
323  REM *****
324  REM *****
325  REM *****
326  REM *****
327  REM *****
328  REM *****
329  REM *****
330  REM *****
331  REM *****
332  REM *****
333  REM *****
334  REM *****
335  REM *****
336  REM *****
337  REM *****
338  REM *****
339  REM *****
340  REM *****
341  REM *****
342  REM *****
343  REM *****
344  REM *****
345  REM *****
346  REM *****
347  REM *****
348  REM *****
349  REM *****
350  REM *****
351  REM *****
352  REM *****
353  REM *****
354  REM *****
355  REM *****
356  REM *****
357  REM *****
358  REM *****
359  REM *****
360  REM *****
361  REM *****
362  REM *****
363  REM *****
364  REM *****
365  REM *****
366  REM *****
367  REM *****
368  REM *****
369  REM *****
370  REM *****
371  REM *****
372  REM *****
373  REM *****
374  REM *****
375  REM *****
376  REM *****
377  REM *****
378  REM *****
379  REM *****
380  REM *****
381  REM *****
382  REM *****
383  REM *****
384  REM *****
385  REM *****
386  REM *****
387  REM *****
388  REM *****
389  REM *****
390  REM *****
391  REM *****
392  REM *****
393  REM *****
394  REM *****
395  REM *****
396  REM *****
397  REM *****
398  REM *****
399  REM *****
400  REM *****
401  REM *****
402  REM *****
403  REM *****
404  REM *****
405  REM *****
406  REM *****
407  REM *****
408  REM *****
409  REM *****
410  REM *****
411  REM *****
412  REM *****
413  REM *****
414  REM *****
415  REM *****
416  REM *****
417  REM *****
418  REM *****
419  REM *****
420  REM *****
421  REM *****
422  REM *****
423  REM *****
424  REM *****
425  REM *****
426  REM *****
427  REM *****
428  REM *****
429  REM *****
430  REM *****
431  REM *****
432  REM *****
433  REM *****
434  REM *****
435  REM *****
436  REM *****
437  REM *****
438  REM *****
439  REM *****
440  REM *****
441  REM *****
442  REM *****
443  REM *****
444  REM *****
445  REM *****
446  REM *****
447  REM *****
448  REM *****
449  REM *****
450  REM *****
451  REM *****
452  REM *****
453  REM *****
454  REM *****
455  REM *****
456  REM *****
457  REM *****
458  REM *****
459  REM *****
460  REM *****
461  REM *****
462  REM *****
463  REM *****
464  REM *****
465  REM *****
466  REM *****
467  REM *****
468  REM *****
469  REM *****
470  REM *****
471  REM *****
472  REM *****
473  REM *****
474  REM *****
475  REM *****
476  REM *****
477  REM *****
478  REM *****
479  REM *****
480  REM *****
481  REM *****
482  REM *****
483  REM *****
484  REM *****
485  REM *****
486  REM *****
487  REM *****
488  REM *****
489  REM *****
490  REM *****
491  REM *****
492  REM *****
493  REM *****
494  REM *****
495  REM *****
496  REM *****
497  REM *****
498  REM *****
499  REM *****
500  REM *****
501  REM *****
502  REM *****
503  REM *****
504  REM *****
505  REM *****
506  REM *****
507  REM *****
508  REM *****
509  REM *****
510  REM *****
511  REM *****
512  REM *****
513  REM *****
514  REM *****
515  REM *****
516  REM *****
517  REM *****
518  REM *****
519  REM *****
520  REM *****
521  REM *****
522  REM *****
523  REM *****
524  REM *****
525  REM *****
526  REM *****
527  REM *****
528  REM *****
529  REM *****
530  REM *****
531  REM *****
532  REM *****
533  REM *****
534  REM *****
535  REM *****
536  REM *****
537  REM *****
538  REM *****
539  REM *****
540  REM *****
541  REM *****
542  REM *****
543  REM *****
544  REM *****
545  REM *****
546  REM *****
547  REM *****
548  REM *****
549  REM *****
550  REM *****
551  REM *****
552  REM *****
553  REM *****
554  REM *****
555  REM *****
556  REM *****
557  REM *****
558  REM *****
559  REM *****
560  REM *****
561  REM *****
562  REM *****
563  REM *****
564  REM *****
565  REM *****
566  REM *****
567  REM *****
568  REM *****
569  REM *****
570  REM *****
571  REM *****
572  REM *****
573  REM *****
574  REM *****
575  REM *****
576  REM *****
577  REM *****
578  REM *****
579  REM *****
580  REM *****
581  REM *****
582  REM *****
583  REM *****
584  REM *****
585  REM *****
586  REM *****
587  REM *****
588  REM *****
589  REM *****
590  REM *****
591  REM *****
592  REM *****
593  REM *****
594  REM *****
595  REM *****
596  REM *****
597  REM *****
598  REM *****
599  REM *****
600  REM *****
601  REM *****
602  REM *****
603  REM *****
604  REM *****
605  REM *****
606  REM *****
607  REM *****
608  REM *****
609  REM *****
610  REM *****
611  REM *****
612  REM *****
613  REM *****
614  REM *****
615  REM *****
616  REM *****
617  REM *****
618  REM *****
619  REM *****
620  REM *****
621  REM *****
622  REM *****
623  REM *****
624  REM *****
625  REM *****
626  REM *****
627  REM *****
628  REM *****
629  REM *****
630  REM *****
631  REM *****
632  REM *****
633  REM *****
634  REM *****
635  REM *****
636  REM *****
637  REM *****
638  REM *****
639  REM *****
640  REM *****
641  REM *****
642  REM *****
643  REM *****
644  REM *****
645  REM *****
646  REM *****
647  REM *****
648  REM *****
649  REM *****
650  REM *****
651  REM *****
652  REM *****
653  REM *****
654  REM *****
655  REM *****
656  REM *****
657  REM *****
658  REM *****
659  REM *****
660  REM *****
661  REM *****
662  REM *****
663  REM *****
664  REM *****
665  REM *****
666  REM *****
667  REM *****
668  REM *****
669  REM *****
670  REM *****
671  REM *****
672  REM *****
673  REM *****
674  REM *****
675  REM *****
676  REM *****
677  REM *****
678  REM *****
679  REM *****
680  REM *****
681  REM *****
682  REM *****
683  REM *****
684  REM *****
685  REM *****
686  REM *****
687  REM *****
688  REM *****
689  REM *****
690  REM *****
691  REM *****
692  REM *****
693  REM *****
694  REM *****
695  REM *****
696  REM *****
697  REM *****
698  REM *****
699  REM *****
700  REM *****
701  REM *****
702  REM *****
703  REM *****
704  REM *****
705  REM *****
706  REM *****
707  REM *****
708  REM *****
709  REM *****
710  REM *****
711  REM *****
712  REM *****
713  REM *****
714  REM *****
715  REM *****
716  REM *****
717  REM *****
718  REM *****
719  REM *****
720  REM *****
721  REM *****
722  REM *****
723  REM *****
724  REM *****
725  REM *****
726  REM *****
727  REM *****
728  REM *****
729  REM *****
730  REM *****
731  REM *****
732  REM *****
733  REM *****
734  REM *****
735  REM *****
736  REM *****
737  REM *****
738  REM *****
739  REM *****
740  REM *****
741  REM *****
742  REM *****
743  REM *****
744  REM *****
745  REM *****
746  REM *****
747  REM *****
748  REM *****
749  REM *****
750  REM *****
751  REM *****
752  REM *****
753  REM *****
754  REM *****
755  REM *****
756  REM *****
757  REM *****
758  REM *****
759  REM *****
760  REM *****
761  REM *****
762  REM *****
763  REM *****
764  REM *****
765  REM *****
766  REM *****
767  REM *****
768  REM *****
769  REM *****
770  REM *****
771  REM *****
772  REM *****
773  REM *****
774  REM *****
775  REM *****
776  REM *****
777  REM *****
778  REM *****
779  REM *****
780  REM *****
781  REM *****
782  REM *****
783  REM *****
784  REM *****
785  REM *****
786  REM *****
787  REM *****
788  REM *****
789  REM *****
790  REM *****
791  REM *****
792  REM *****
793  REM *****
794  REM *****
795  REM *****
796  REM *****
797  REM *****
798  REM *****
799  REM *****
800  REM *****
801  REM *****
802  REM *****
803  REM *****
804  REM *****
805  REM *****
806  REM *****
807  REM *****
808  REM *****
809  REM *****
810  REM *****
811  REM *****
812  REM *****
813  REM *****
814  REM *****
815  REM *****
816  REM *****
817  REM *****
818  REM *****
819  REM *****
820  REM *****
821  REM *****
822  REM *****
823  REM *****
824  REM *****
825  REM *****
826  REM *****
827  REM *****
828  REM *****
829  REM *****
830  REM *****
831  REM *****
832  REM *****
833  REM *****
834  REM *****
835  REM *****
836  REM *****
837  REM *****
838  REM *****
839  REM *****
840  REM *****
841  REM *****
842  REM *****
843  REM *****
844  REM *****
845  REM *****
846  REM *****
847  REM *****
848  REM *****
849  REM *****
850  REM *****
851  REM *****
852  REM *****
853  REM *****
854  REM *****
855  REM *****
856  REM *****
857  REM *****
858  REM *****
859  REM *****
860  REM *****
861  REM *****
862  REM *****
863  REM *****
864  REM *****
865  REM *****
866  REM *****
867  REM *****
868  REM *****
869  REM *****
870  REM *****
871  REM *****
872  REM *****
873  REM *****
874  REM *****
875  REM *****
876  REM *****
877  REM *****
878  REM *****
879  REM *****
880  REM *****
881  REM *****
882  REM *****
883  REM *****
884  REM *****
885  REM *****
886  REM *****
887  REM *****
888  REM *****
889  REM *****
890  REM *****
891  REM *****
892  REM *****
893  REM *****
894  REM *****
895  REM *****
896  REM *****
897  REM *****
898  REM *****
899  REM *****
900  REM *****
901  REM *****
902  REM *****
903  REM *****
904  REM *****
905  REM *****
906  REM *****
907  REM *****
908  REM *****
909  REM *****
910  REM *****
911  REM *****
912  REM *****
913  REM *****
914  REM *****
915  REM *****
916  REM *****
917  REM *****
918  REM *****
919  REM *****
920  REM *****
921  REM *****
922  REM *****
923  REM *****
924  REM *****
925  REM *****
926  REM *****
927  REM *****
928  REM *****
929  REM *****
930  REM *****
931  REM *****
932  REM *****
933  REM *****
934  REM *****
935  REM *****
936  REM *****
937  REM *****
938  REM *****
939  REM *****
940  REM *****
941  REM *****
942  REM *****
943  REM *****
944  REM *****
945  REM *****
946  REM *****
947  REM *****
948  REM *****
949  REM *****
950  REM *****
951  REM *****
952  REM *****
953  REM *****
954  REM *****
955  REM *****
956  REM *****
957  REM *****
958  REM *****
959  REM *****
960  REM *****
961  REM *****
962  REM *****
963  REM *****
964  REM *****
965  REM *****
966  REM *****
967  REM *****
968  REM *****
969  REM *****
970  REM *****
971  REM *****
972  REM *****
973  REM *****
974  REM *****
975  REM *****
976  REM *****
977  REM *****
978  REM *****
979  REM *****
980  REM *****
981  REM *****
982  REM *****
983  REM *****
984  REM *****
985  REM *****
986  REM *****
987  REM *****
988  REM *****
989  REM *****
990  REM *****
991  REM *****
992  REM *****
993  REM *****
994  REM *****
995  REM *****
996  REM *****
997  REM *****
998  REM *****
999  REM *****
1000  REM *****

```

Játék-  
pályázat



```

XS 1 REM *****
CH 2 REM *          ZSIROZAS V1.3          *
MM 3 REM *
BT 4 REM *          B                      *
GM 5 REM *
PC 6 REM *          LIST: 1-10450 LINES    *
XX 7 REM *          USED: 2634 BYTES      *
PM 8 REM *
XR 9 REM *****
JE 10 :
PD 1000 REM - BASIC TERULET ELLENORZESE
DE 1010 IF PEEK(44) < >16 THENQ =1
DB 1020 IF PEEK(43) < >1 THENQ =1
CM 1030 IF PEEK(4096) < >0 THENQ =1
DB 1040 IFQ =0 THEN2010
BH 1050 POKE546,0
JD 1060 POKE53280,2: POKE53281,2
QP 1070 PRINT CHR$(147); CHR$(19); "?LOAD[SPC][SPC]ERROR"
BA 1080 PRINT
XP 1090 PRINT")TURN[SPC]THE[SPC]COMPUTER[SPC]OFF[SPC]AND[SPC]ON[SPC]A
      GAIN."
RB 1100 PRINT")LOAD[SPC]THE[SPC]PROGRAMME[SPC]WHICH[SPC]IS[SPC]CALLED
      [SPC](A)."
KS 1110 POKE120,2:: REM = KILLER POKE
KF 1120 END
XH 1130 :
MF 2000 REM - KARAKTERGENERATOR
XA 2010 REM -- PARAMETEREZES
MH 2020 DIMD(8)
XF 2030 CR =53248::: REM = KARAKTER ROM KEZDOCIM
QD 2040 CW =2048::: REM = KARAKTER RAM KEZDOCIM
GX 2050 CT =2816::: REM = KARAKTER ATHELYEZO KARAKTER (SHIFT+SPACE)
      KEZDOCIME
AM 2060 POKE53272,18
ED 2070 :
EQ 3000 REM -- ASCII (->) POKE ATKODOLO
ER 3010 FORI =0 TO7: READD(I): NEXTI
PH 3020 DATA128,032,000,064,191,096,064,096
XP 4000 REM -- KARAKTER ROM MASOLAS A RAMBA
BG 4010 POKE56334,0: POKE1,51
MH 4020 FORI =0 TO2047 STEP512
KF 4030 : IFI =1024 THENP =255
BC 4040 : FORJ =0 TO511 STEP8
QA 4050 :: FORK =0 TO7
BX 4060 ::: POKECW +I +J +K, PEEK(CR +I +J +K)
CG 4070 :: NEXTK
CE 4080 : NEXTJ
MM 4090 NEXTI
SC 4100 P =0
CQ 4110 FORI =512 TO1536 STEP1024
HJ 4120 : IFI =1536 THENP =255
CX 4130 : POKECW +I +8,P
SR 4140 : FORJ =8 TO216
BF 4150 :: POKECW +I +J, PEEK(CR +I +J)
QP 4160 : NEXTJ
XB 4170 NEXTI
QB 4180 POKE1,55: POKE56334,1
RD 5000 REM -- KARAKTER BEHELYEZES
QX 5010 READA
KG 5020 IFA = -1 THEN6010
KF 5030 FORI =0 TO7
DP 5040 : READB
QS 5050 : POKECT +I,B
JF 5060 NEXTI
BS 5070 SH = INT(A /32):SL =A -8H *32
PA 5080 P =CW +(D(SH) +SL) *8
PS 5090 FORI =0 TO7
JP 5100 : POKEP +I, PEEK(CT +I)
RP 5110 NEXTI
AC 5120 GOTO5010
EC 6000 REM - KOVETKEZO PROGRAM HIVASA
PD 6010 A$ ="C"
JM 6020 PRINT CHR$(147);
QC 6030 PRINT"LOAD" CHR$(34)A$ CHR$(34)",B"
DQ 6040 PRINT"[DOWN][DOWN][DOWN][DOWN]RUN
KG 6050 POKE198,4

```



```

DQ 6060 POKE631,19
CE 6070 POKE632,13: POKE633,13: POKE634,13
MX 6080 :
FX 10000 REM - UJ KARAKTEREK ADATAI
GR 10010 DATA91: REM = ASC [
KR 10020 DATA024,126,255,255,255,000,126,024
EK 10030 DATA92: REM = ASC E
RC 10040 DATA036,060,060,126,255,254,082,024
BF 10050 DATA93: REM = ASC J
RH 10060 DATA024,060,060,060,060,126,024,012
DB 10070 DATA94: REM = ASC ↑
AF 10080 DATA066,231,255,255,126,060,024,000
RR 10090 DATA95: REM = ASC ←
CM 10100 DATA048,032,096,211,212,210,241,214
KK 10110 :
PS 10120 :
FS 10130 :
HX 10140 :
DC 10150 DATA35: REM = ASC #
XG 10160 DATA012,008,060,102,102,126,102,000
BS 10170 DATA36: REM = ASC $
EA 10180 DATA012,008,126,112,124,112,126,000
HM 10190 DATA37: REM = ASC %
BM 10200 DATA012,008,060,024,024,024,060,000
SP 10210 DATA38: REM = ASC &
AE 10220 DATA012,008,060,102,102,102,060,000
XH 10230 DATA39: REM = ASC '
AM 10240 DATA054,000,060,102,102,102,060,000
RH 10250 :
DH 10260 :
FD 10270 DATA42: REM = ASC *
MK 10280 DATA010,020,060,102,102,102,060,000
HX 10290 DATA43: REM = ASC +
DF 10300 DATA012,008,102,102,102,102,060,000
QM 10310 :
CM 10320 :
GH 10330 DATA60: REM = ASC <
HF 10340 DATA054,000,102,102,102,102,060,000
ED 10350 DATA61: REM = ASC =
DX 10360 DATA010,020,000,102,102,102,060,000
CX 10370 DATA62: REM = ASC >
BJ 10380 DATA060,066,153,161,161,153,066,060
QR 10390 :
FX 10400 :
KK 10410 DATA64: REM = ASC @
BQ 10420 DATA030,096,111,096,111,096,030,000
PX 10430 DATA-1
QD 10440 :
    
```

```

XG 1 XEN *
CH 2 XEN *
BH 3 XEN *
FX 4 XEN *
GB 5 XEN *
FE 6 XEN *
BK 7 XEN *
BH 8 XEN *
AX 9 XEN *
TE 10 :
BH 100 PRINT CHR$(147)
AF 110 REM - A BASIC TERULET ELLENORZESE
NS 120 IF PEEK(44) < >16 THEN0 =1
TF 130 IF PEEK(43) < >1 THEN0 =1
TH 140 IF PEEK(4096) < >0 THEN0 =1
SR 150 IF 0 =0 THEN300
UB 160 POKE546,0
RV 170 POKE53290,2: POKE53291,2
SR 180 PRINT CHR$(147); CHR$(19); "?LOAD)[SPC][SPC]ERROR"
TE 190 PRINT
AD 200 PRINT">TURN[SPC]THE[SPC]COMPUTER[SPC]OFF[SPC]AND[SPC]ON[SPC]AG
AIN."
DH 210 PRINT">LOAD)[SPC]THE[SPC]PROGRAMME[SPC]WHICH[SPC]IS[SPC]CALLED
[SPC](A)."
    
```



```

OK 220 POKE53265, PEEK(53265) OR16
FX 230 POKE120,2
NE 240 ENJ)
RP 250 :
FD 300 CLR
MR 310 PRINT CHR$(147)
FB 320 FORI =63: T0640: POKE1,0: NEXT
CG 330 POKE53265, PEEK(53265) OR16
EG 340 :
AR 400 DIMR(8),G1(4),G2$(4),K1(4),K2$(4),P(32),S$(24)
XE 410 X1 =4:X2 =X1:::X3 =00:X4 =21:X5 =03:X6 =23:::X7 =X6:::
ES 420 Y1 =4:Y2 =Y1 +25:Y3 =17:Y4 =00:Y5 =15:Y6 =Y2 +1:Y7 =Y1 +1:
JP 430 TX =1000
XK 440 NG =0:MK =0
PR 450 :
GS 500 HW =54272:HI =90:SI =7
HX 510 FORI =0 T024: POKEHW +1,0: NEXTI
HM 520 POKEHW +02,000: POKEHW +03,000: POKEHW +05,114: POKEHW +06,243
PG 530 POKEHW +24,015
FD 540 :
PA 600 P1 =13:P2 =13:P3 =0
SX 610 POKE53280,P1: POKE53281,P2: POKE646,P3
MH 620 :
FM 630 GOSUB51010: REM->KOSZONTO
HJ 640 IFG$ ="[SPC]" THEN GOSUB52010
DR 650 P1 =12:P2 =1:P3 =0:P4 =8:P5 =6:P6 =7:P7 =3
MM 660 :
JF 1000 REM - JATEKVEZERLES
PX 1010 GOSUB10010: REM->BEVEZETES
CS 1020 B =0
SB 1030 IFUP =1 THEN1200
GS 1040 GOSUB11010: REM->LAPOSZTAS
DB 1050 GOSUB12010: REM->KEZI LAP KIJATSZASA
SM 1060 KR =1
RM 1070 GOSUB13010: REM->KOZEPMEZO RAJZOLAS
SB 1080 GOSUB14010: REM->A GEP ERTEKES LAPJAI
DD 1090 ON(K1(0)) GOSUB15010,15010,15010,16010,15010,15010,15010,1601
0: REM->VALASZOK
JF 1100 GOSUB13010: REM->KOZEPMEZO RAJZOLAS
PJ 1110 IFA(B) =1 THEN1160
DD 1120 IFA(B) =A(1) THEN1160
HH 1130 GOSUB18010: REM->KEZ VISZI
QA 1140 IFVK >31 THEN1370
AK 1150 GOT01040
HC 1160 GOSUB20010: REM->TOVABBUTES VIZSGALATA
XX 1170 IFTK >0 THEN1050
JF 1180 GOSUB19010: REM->GEP VISZI
RE 1190 IFVK >31 THEN1370
DF 1200 GOSUB11010: REM->LAPOSZTAS
SS 1210 GOSUB14010: REM->A GEP ERTEKES LAPJAI
SE 1220 GOSUB21010: REM->GEPI KEZDES
EQ 1230 GOSUB13010: REM->KOZEPMEZO RAJZOLAS
DF 1240 GOSUB12010: REM->KEZI LAP KIJATSZASA
BG 1250 KR =1
RS 1260 GOSUB13010: REM->KOZEPMEZO RAJZOLAS
PJ 1270 IFA(B) =1 THEN1290
JP 1280 IFA(B) < >A(1) THEN1170
PK 1290 GOSUB22010: REM->SZABAD VALASZTAS
SA 1300 IFTG >0 THEN1320
MA 1310 GOT01130
DM 1320 GOSUB13010: REM->KOZEPMEZO RAJZOLAS
JA 1330 GOSUB12010: REM->KEZI LAP KIJATSZASA
JQ 1340 KR =1
EE 1350 GOSUB13010: REM->KOZEPMEZO RAJZOLAS
DJ 1360 GOT01270
FA 1370 GOSUB23010: REM->PARTI ERTEKELES
XE 1380 GOT01010
FH 1390 END
PJ 1400 :
FD 10000 REM - BEVEZETES
AE 10010 FORI =1 T032
HD 10020 :P(1) =I
GG 10030 NEXTI
HB 10040 PP =32:PK =32
PG 10050 POKE53280,P1: POKE53281,P2: POKE646,P3
DR 10060 X =05:Y =39: GOSUB40010: PRINT"0"

```



```

○ PS 10070 X =09: ::::: GOSUB40010: PRINT"W"
○ JG 10080 X =13: ::::: GOSUB40010: PRINT"E"
○ HR 10090 X =17: ::::: GOSUB40010: PRINT"R"
○ JH 10100 RETURN
○ KK 10110 :
○ HS 11000 REM - LAPOSZTAS
○ RM 11010 L = INT(32 * RND(0)) +1
○ KK 11020 IFPP <1 THEN RETURN
○ EM 11030 IFP(L) <0 THEN11010
○ FO 11040 ::
○ GX 11050 IFOK =1 THEN11240
○ DB 11060 FORI =1 TO4
○ CD 11070 : IFG1(I) >0 THEN11220
○ PH 11080 :Q =P(L)
○ SD 11090 : GOSUB30010
○ RD 11100 :G1(I) =Q1:Q2$(I) =Q$
○ EP 11110 :P(L) = -1
○ KS 11120 :PP =PP -1
○ GO 11130 : POKE646,P4
○ JJ 11140 :X =I *X1 +0:Y =Y1: GOSUB40010: PRINT"██████"
○ BB 11150 :X =I *X1 +1: ::::: GOSUB40010: PRINT"██████"
○ FX 11160 :X =I *X1 +2: ::::: GOSUB40010: PRINT"██████"
○ HP 11170 ::
○ SS 11180 :X =X3:Y =Y3: GOSUB40010: PRINT"[SPC][SPC][SPC]"
○ KB 11190 :X =X3: ::::: GOSUB40010: PRINTPP
○ RP 11200 :OK =1
○ XK 11210 : GOTO11020
○ AQ 11220 NEXTI
○ DJ 11230 RETURN
○ AE 11240 FORI =1 TO4
○ BE 11250 : IFK1(I) >0 THEN11400
○ FO 11260 :Q =P(L)
○ RG 11270 : GOSUB30010
○ KS 11280 :K1(I) =Q1:K2$(I) =Q$
○ SF 11290 :P(L) = -1
○ JB 11300 :PP =PP -1
○ DA 11310 : POKE646,P5
○ GD 11320 :X =I *X2 +0:Y =Y2: GOSUB40010: PRINT"┌───┐"
○ ED 11330 :X =I *X2 +1: ::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC]"
○ GG 11340 :X =I *X1 +2: ::::: GOSUB40010: PRINT"└───┘"
○ CO 11350 :X =I *X2 +1:Y =Y2 +1: GOSUB40010: PRINTK2$(I)
○ JB 11360 :X =X3:Y =Y3: GOSUB40010: PRINT"[SPC][SPC][SPC]"
○ ME 11370 :X =X3: ::::: GOSUB40010: PRINTPP
○ PG 11380 :OK =0
○ HX 11390 : GOTO11020
○ HK 11400 NEXTI
○ AF 11410 RETURN
○ XS 11420 :
○ PE 12000 REM - KEZI LAP KIJATSZASA
○ RE 12010 B =B +1
○ DF 12020 POKE646,P3
○ KS 12030 X =X4:Y =Y4: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
○ EC 12040 X =X4: ::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]N[SPC]
[SPC]JN[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC]"
○ KR 12050 ::
○ MX 12060 POKEHW +05,009: POKEHW +06,009
○ BJ 12070 POKEHW +01,058: POKEHW,138
○ DF 12080 POKEHW +04,017: FORI =1 TOHI #2: NEXTI
○ PS 12090 POKEHW +04,016: FORI =1 TOHI: NEXTI
○ PQ 12100 POKEHW +01,029: POKEHW,069
○ QF 12110 POKEHW +04,017: FORI =1 TOHI #2: NEXTI
○ AS 12120 POKEHW +04,016: FORI =1 TOHI: NEXTI
○ GS 12130 POKEHW +05,114: POKEHW +06,243
○ DH 12140 ::
○ ME 12150 GETG$
○ FX 12160 IFG$ ="" THEN12150
○ SG 12170 H =0
○ RR 12180 IFG$ ="0" THENH =1
○ ME 12190 IFG$ ="W" THENH =2
○ GC 12200 IFG$ ="E" THENH =3
○ XG 12210 IFG$ ="R" THENH =4
○ EK 12220 !FR =0 THEN12150

```



```

O CP 12230 IFTK =0 THEN12290
O KU 12240 REM -- RENONSZ ELLENI VEDELEM
O AS 12250 IF(K1(H) < >A(1)) AND(K1(H) < >1) THENH =0
O PF 12260 IFH =0 THEN12150
O AH 12270 REM -----
O AR 12280 ::
O FJ 12290 IFK1(H) <1 THEN12150
O KG 12300 K1(0) =K1(H)
O HP 12310 X =H *X2 +0:Y =Y2: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC]
[SPC][SPC]"
O AB 12320 X =H *X2 +1:Y =Y2: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC]
[SPC][SPC]"
O MX 12330 X =H *X2 +2:Y =Y2: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC]
[SPC][SPC]"
O FP 12340 X =X4:Y =Y4: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
O OB 12350 K(H) =0
O PB 12360 A(B) =K1(0)
O PD 12370 IFB <2 THEN RETURN
O KM 12380 K(0) =A(1)
O BC 12390 RETURN
O KK 12400 :
O RK 13000 REM - KOZEPMEZO RAJZOLAS
O HJ 13010 X =X5 +2 *B:Y =Y5: GOSUB40010
O EB 13020 IFKR =0 THEN POKE646,P4: PRINTB;"[LEFT],[SPC]";G2*(F):G1(F)
=0:G2*(F) ="[SPC]"
O EH 13030 IFKR =1 THEN POKE646,P5: PRINTB;"[LEFT],[SPC]";K2*(H):K1(H)
=0:K2*(H) ="[SPC]"
O BX 13040 ::
O GX 13050 PK =PK -1
O GP 13060 KR =0
O SP 13070 RETURN
O OF 13080 :
O RC 14000 REM - A GEP ERTEKES LAPJAI
O DA 14010 HE =0:TZ =0:AS =0
O NJ 14020 FORI =1 TO4
O KS 14030 : IFG1(I) =1 THENHE =HE +1
O HQ 14040 : IFG1(I) =4 THENTZ =TZ +1
O RP 14050 : IFG1(I) =0 THENAS =AS +1
O EG 14060 NEXTI
O OM 14070 RETURN
O SG 14080 :
O DJ 15000 REM - A GEP VALASZA ERTEKTELEN LAPRA
O KO 15010 FORF =1 TO4
O KI 15020 : IFG1(F) < >K1(0) THEN15050
O BF 15030 : GOSUB17010
O BP 15040 : RETURN
O KM 15050 NEXTF
O HP 15060 FORF =1 TO4
O BS 15070 : ON(G1(F) +1) GOTO15100,15100,15000,15000,15100,15000,15000
,15000,15100
O PA 15080 : GOSUB17010
O GH 15090 : RETURN
O PE 15100 NEXTF
O FF 15110 FORF =1 TO4
O DX 15120 : IFG1(F) < >1 THEN15150
O JC 15130 : GOSUB17010
O KB 15140 : RETURN
O RP 15150 NEXTF
O HP 15160 ON(TZ +1) GOTO15240,15170,15180,15180,15240
O GX 15170 E2 =4: GOTO15190
O DG 15180 E2 =8
O FC 15190 FORF =1 TO4
O JS 15200 : IFG1(F) < >E2 THEN15230
O HS 15210 : GOSUB17010
O CO 15220 : RETURN
O JR 15230 NEXTF
O DO 15240 F = INT(4 * RND(0)) +1
O XA 15250 IFG1(F) =0 THEN15240
O KH 15260 GOSUB17010
O DR 15270 RETURN
O RE 15280 :
O SC 16000 REM - A GEP VALASZA ERTEKES LAPRA
O MD 16010 IFK1(0) < >4 THENE1 =AS:E2 =0: GOTO16030

```



```

XC 16020 E1 =T2:E2 =4
KA 16030 IF9 < =1 THENE1 =3
PS 16040 !FPK < =8 THENE1 =3
BR 16050 ::
GF 16060 IFE1 <3 THEN16120
DJ 16070 FORF =1 T04
CD 16080 : IFG1(F) < >E2 THEN16110
HB 16090 : GOSUB17010
SM 16100 : RETURN
EP 16110 NEXTF
HB 16120 IF(E1 +HE) <3 THEN16180
DF 16130 FORF =1 T04
PE 16140 : IFG1(F) < >1 THEN16170
GC 16150 : GOSUB17010
PB 16160 : RETURN
DA 16170 NEXTF
FB 16180 FORF =1 T04
BX 16190 : ON(G1(F) +1) GOTO16220,16220,16200,16200,16220,16200,16200
      16200,16220
HQ 16200 : GOSUB17010
MD 16210 : RETURN
AC 16220 NEXTF
PX 16230 F = INT(4 * RND(0)) +1
HD 16240 IFG1(F) =0 THEN16230
BF 16250 GOSUB17010
PC 16260 RETURN
MM 16270 :
BK 17000 REM - GEPÍ LAP KIJÁTSZÁSA AZ 'A' TOMBBE
HE 17010 B =B +1
PR 17020 A(B) =G1(F)
KD 17030 FORI =1 TOTX: NEXTI
PM 17040 X =F *X1 +0:Y =Y1: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC]
      [SPC][SPC]"
KF 17050 X =F *X1 +1::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC]
      [SPC][SPC]"
EG 17060 X =F *X1 +2::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC]
      [SPC][SPC]"
GX 17070 RETURN
EF 17080 :
BK 18000 REM - KEZ VISZI
DK 18010 FORI =1 TOTX: NEXTI
HS 18020 FORI =1 T016
FD 18030 :X =X5 +I:Y =Y5: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
      [SPC][SPC][SPC][SPC]"
PX 18040 NEXTI
EJ 18050 FORI =1 T08
DD 18060 : IFA(I) =4 THENKP =KP +1
KG 18070 : IFA(I) =8 THENKP =KP +1
DP 18080 NEXTI
XH 18090 FORI =1 T08
GS 18100 : IFA(I) >0 THENVK =VK +1
EE 18110 :A(I) =0
QM 18120 NEXTI
EE 18130 POKE646,P5
PH 18140 X =X6:Y =Y6: GOSUB40010: PRINT"[SPC][SPC][SPC]"
AC 18150 X =X6::::: GOSUB40010: PRINTKP *10
EP 18160 KS =KS +1: IFVK =32 THENVK =34
RF 18170 B =0:TG =0:TK =0:OK =1
JF 18180 RETURN
KD 18190 :
PK 19000 REM - GEP VISZI
KJ 19010 FORI =1 TOTX: NEXTI
FA 19020 FORI =1 T016
HA 19030 :X =X5 +I:Y =Y5: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
      [SPC][SPC][SPC][SPC]"
BQ 19040 NEXTI
MM 19050 FORI =1 T08
MG 19060 : IFA(I) =4 THENGP =GP +1
PM 19070 : IFA(I) =8 THENGP =GP +1
KJ 19080 NEXTI
BQ 19090 FORI =1 T08
EX 19100 : IFA(I) >0 THENVK =VK +1
HM 19110 :A(I) =0
MF 19120 NEXTI
FG 19130 POKE646,P4
GE 19140 X =X7:Y =Y7: GOSUB40010: PRINT"[SPC][SPC][SPC]"
BP 19150 X =X7::::: GOSUB40010: PRINTGP *10

```



```

XS 19150 GS =GS +1: IFVK =32 THENVK =33
EC 19170 B =0: TG =0: TK =0: OK =0
BP 19180 RETURN
DB 19190 :
FK 20000 REM - TOVABBUTES VIZSGALATA
RC 20010 POKE646,P3
CF 20020 FORH =1 TO4
MB 20030 : IFK1(H) =! THENTK =1: GOTO20000
GJ 20040 : IFK1(H) =A(1) THENTK =1: GOTO20000
SC 20050 NEXTH
PP 20060 TK =0
CB 20070 ::
FS 20080 IFH =8 THEN20330
EJ 20090 IFPK =0 THEN20330
QE 20100 X =X4: Y =Y4: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
AD 20110 X =X4: Y =Y4: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC]TOV#BB[SPC]<TI[SPC](I/N)?[SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
CR 20120 GETG$
MR 20130 IFG$ ="" THEN20120
AQ 20140 IFG$ < >"I" THENTK =0: GOTO20330
FD 20150 ::
PS 20160 IFTK >0 THEN20330
RS 20170 X =X4: Y =Y4: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
PP 20180 X =X4: Y =Y4: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
[SPC]SAJNOS[SPC]NNEK[SPC]NINCS[SPC]<T*LAPJA[SPC][SPC][SPC]
[SPC][SPC][SPC]"
RK 20190 ::
XB 20200 POKEHW +05,008: POKEHW +06,000
BJ 20210 FORI =1 TO2
KC 20220 : POKEHW +01,032: POKEHW,219
JP 20230 : POKEHW +04,129: FORJ =1 TOHI *2: NEXTJ
GQ 20240 : POKEHW +04,128: FORJ =1 TOHI: NEXTJ
SA 20250 : POKEHW +01,016: POKEHW,109
CQ 20260 : POKEHW +04,129: FORJ =1 TOHI *2: NEXTJ
AX 20270 : POKEHW +04,128: FORJ =1 TOHI: NEXTJ
PC 20280 NEXTI
SJ 20290 POKEHW +05,114: POKEHW +06,243
QE 20300 ::
CJ 20310 FORI =1 TOTX: NEXTI
AQ 20320 ::
HE 20330 X =X4: Y =Y4: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
PD 20340 RETURN
MK 20350 :
GX 21000 REM - GERI KEZDES
PJ 21010 E1 =TZ: E2 =4
RP 21020 ::
FX 21030 IFE1 >2 THEN21230
KX 21040 IFE1 <1 THEN21060
JM 21050 IF(E1 +HE) >2 THEN21230
FB 21060 IFE2 =0 THEN21100
MS 21070 E1 =AS: E2 =8
GG 21080 ::
AH 21090 GOTO21030
DD 21100 FORF =1 TO4
DP 21110 : ON(G1(F) +1) GOTO21140,21140,21120,21120,21140,21120,21120
,21120,21140
QP 21120 : GOSUB17010
PQ 21130 : RETURN
DQ 21140 NEXTF
DF 21150 IFAS >0 THEN21210
SP 21160 IFTZ >0 THEN21190
SB 21170 E2 =1
BS 21180 GOTO21230
FC 21190 E2 =4
JS 21200 GOTO21230
CS 21210 IFTZ >1 THEN21190
SX 21220 E2 =8

```





```

CJ 21230 FORF =1 TO4
HJ 21240 : IFG1(F) < >E2 THEN21270
OC 21250 : GOSUB17010
RF 21260 : RETURN
BF 21270 NEXTF
MR 21280 F = INT(4 * RND(0)) +1
JM 21290 IFG1(F) =0 THEN21280
EO 21300 GOSUB17010
XF 21310 RETURN
ER 21320 :
RR 22000 REM -- SZHBAH VALASZTAS
CF 22010 K1(0) =R(1)
RT 22020 FORF =1 TO4
MS 22030 : IFG1(F) < >K1(0) THEN22060
KB 22040 : GOSUB17010
DD 22050 :IG =1: RETURN
EC 22060 NEXTF
DR 22070 IFK1(0) =4 THEN22090
BR 22080 IFK1(0) < >8 THENIG =0: RETURN
SC 22090 FORF =1 TO4
MX 22100 : IFG1(F) < >1 THEN22130
OR 22110 : GOSUB17010
-H 22120 :IG =1: RETURN
FB 22130 NEXTF
GT 22140 :IG =0
ER 22150 RETURN
RT 22160 :
XR 23000 REM -- PARTI ERTEKELES
FA 23010 FORI =1 TOTX: NEXTI
FS 23020 PRINT CHR$(147)
HF 23030 POKE646,P3
PH 23040 FORI =0 TO24
AG 23050 :S$(I) ="[SPC]"
AC 23060 NEXTI
SB 23070 IFGP >KP THEN24010
CJ 23080 J=KP >GP THEN25010
FB 23090 IFVK =33 THEN24010
QB 23100 IFVK =34 THEN25010
HH 23110 ::
EK 24000 REM -- A GEP NYERT
FR 24010 POKE53280,P1: POKE53281,P6
CK 24020 MG =MG +1
RG 24030 IFGP =8 THENMG =MG +1::S$(0) ="[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]KOPASZ
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC]"
JH 24040 IFKS =0 THENMG =MG +1::S$(0) ="[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]CSUPASZ!
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC]"
JM 24050 I = INT(5 * RND(0)) +1
XR 24060 IFI =1 THENS$(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]# [SPC]L [SPC]O [SPC]M
[SPC] [SPC] [SPC] [SPC] [SPC] [SPC] [SPC] [SPC] [SPC] [SPC] [SPC] [SPC]
KX 24070 IFI =2 THENS$(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]BIZONY
[SPC]EZ[SPC]NEM[SPC]S[SPC]S[SPC]KERCL[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC]"
PB 24080 IFI =3 THENS$(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]NAH#T,
[SPC]NEM[SPC]H[SPC]TEM[SPC]VOLNA[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC]"
BK 24090 IFI =4 THENS$(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC]SZOMOR+[SPC]HXREM[SPC]VAN, [SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC]"
GP 24100 IFI =5 THENS$(1) ="[NE[SPC]ADJA[SPC]FEL[SPC]LEGK[SPC]ZELEBB[SPC]BIZTO
S[SPC]S[SPC]KERCL!"
DG 24110 X =06:Y =00: GOSUB40010: PRINTS$(1)
DF 24120 X =08::::: GOSUB40010: PRINTS$(0)
AB 24130 X =10::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC]EZT[SPC]A[SPC]PARTI[SPC]#N[SPC]NYERTEM[SPC][SPC]
[SPC][SPC][SPC][SPC][SPC]"
BK 24140 X =12::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC]A[SPC]M*RK*Z$S[SPC]#LL$A: [SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
QR 24150 X =16::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
PH 24160 PRINT STR$(MG) +"[SPC]:" + STR$(MK)

```





```

000000 00 24170  :
000000 01 24180 GP =0:GS =0:KP =0:KS =0:OK =0:UP =1:VK =0
000000 02 24190 REM -> ZENE
000000 03 24200 GOTO55010
000000 04 24210 :
000000 05 25000 REM -- A KEZ NYERT
000000 06 25010 POKE53280,P1: POKE53281,P7
000000 07 25020 MK =MK +1
000000 08 25030 IFKP =8 THENMK =MK +1::S*(0) ="[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]KOPASZ[SPC]LETTEM[SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
000000 09 25040 IFGS =0 THENMK =MK +1::S*(0) ="[SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]CSUPASZ[SPC]LETTEM[SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
000000 10 25050 I = INT(5 * RND(0)) +1
000000 11 25060 IFI =1 THENS*(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC]G[SPC]R[SPC]A[SPC]T[SPC]U[SPC]L[SPC]#[SPC]L[SPC]O[SPC]K
[SPC],[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
000000 12 25070 IFI =2 THENS*(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC]BIZONY
[SPC]EZ[SPC]NEM[SPC]SIKER<LT'[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC]"
000000 13 25080 IFI =3 THENS*(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC]NAH#T,
[SPC]NEM[SPC]HITTEM[SPC]VOLNA[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC]"
000000 14 25090 IFI =4 THENS*(1) ="[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC]SZOMOR+[SPC]VAGYOK,[SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
000000 15 25100 IFI =5 THENS*(1) ="[SPC][SPC]NEM[SPC]ADOM[SPC]FEL[SPC]LEGK'
ZELEBR[SPC]#[SPC]NYEREK[SPC]"
000000 16 25110 X =06:Y =00: GOSUB40010: PRINTS*(1)
000000 17 25120 X =08::::: GOSUB40010: PRINTS*(0)
000000 18 25130 X =10::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC]EZT[SPC]A[SPC]PARTIT[SPC]'N[SPC]NYERTE[SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC]"
000000 19 25140 X =12::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC]A[SPC]M#RK#Z#S[SPC]#LL#SA:[SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
000000 20 25150 X =16::::: GOSUB40010: PRINT"[SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]"
000000 21 25160 PRINT STR*(MG) +"[SPC]:" + STR*(MK)
000000 22 25170 :
000000 23 25180 GP =0:GS =0:KP =0:KS =0:OK =1:UP =0:VK =0
000000 24 25190 REM -> ZENE
000000 25 25200 GOTO55010
000000 26 25210 :
000000 27 30000 REM - LAP ATKODOLAS
000000 28 30010 Q2 =0:Q1 =Q2 -(8 * INT(Q2 /8)) +1
000000 29 30020 IFQ2 /8 >0 THENQ$ ="["
000000 30 30030 IFQ2 /8 >1 THENQ$ ="["
000000 31 30040 IFQ2 /8 >2 THENQ$ ="["
000000 32 30050 IFQ2 /8 >3 THENQ$ ="["
000000 33 30060 IFQ1 =1 THENQ$ =Q$ +"[SPC][SPC]7"
000000 34 30070 IFQ1 =2 THENQ$ =Q$ +"[SPC][SPC]8"
000000 35 30080 IFQ1 =3 THENQ$ =Q$ +"[SPC][SPC]9"
000000 36 30090 IFQ1 =4 THENQ$ =Q$ +"[SPC]10"
000000 37 30100 IFQ1 =5 THENQ$ =Q$ +"[SPC][SPC]A"
000000 38 30110 IFQ1 =6 THENQ$ =Q$ +"[SPC][SPC]F"
000000 39 30120 IFQ1 =7 THENQ$ =Q$ +"[SPC][SPC]K"
000000 40 30130 IFQ1 =8 THENQ$ =Q$ +"[SPC][SPC]←"
000000 41 30140 RETURN
000000 42 30150 :
000000 43 40000 REM - CURSR POZICIONALAS
000000 44 40010 X$ ="[DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN]
[DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN][DOWN]
[DOWN][DOWN][DOWN][DOWN][DOWN]"
000000 45 40020 Y$ ="[RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT][RIGHT]
[RIGHT]"
000000 46 40030 X$ = MID$(X$,1,X)
000000 47 40040 Y$ = MID$(Y$,1,Y)
000000 48 40050 X$ ="[HOME]" +X$ +Y$
000000 49 40060 PRINTX$;
000000 50 40070 RETURN

```

A program további részét (köszöntő, képernyőszöveg, zene) — mivel a működés megértéséhez nem elengedhetetlenül szükséges — terjedelmi okból nem tudjuk közölni. A teljes program a Pötyögőszolgálatról megrendelhető.







Tisztelt C-64 felhasználó!  
Örömmel értesítem, hogy elkészült a NARANCS 3 6200 nevű katalóguskészítő program! A program ára 500 Ft, amely tartalmazza a postaköltséget is. Borbély Balázs, 1116 Budapest, Sáfrány u. 44. III. 33. Tel.: 162-7724.

C-64-re programok eladók lemezre és kazettára, kedvezményel. Felbélyegzett válaszborítékért listát küldök. Papp Mihály, 4034 Debrecen, Sterbinszky u. 37.

Sürgősen eladó: C-64 + floppy + játékok (100 db lemez) + joy. Az árban meg egyezünk. Staub Ferenc, Tel.: 1-755-240.

AMIGA 500 és A500 Plus játék- & felhasználói programok nagy választékban, mérsékelt áron kaphatók ÉVÁ-NÁL! Listához, tájékoztatóhoz szükséges: 1 db 3,5-es lemez és 20 Ft-os bélyeg. Cím: Kopácsi Lajosné, H-1031 Budapest, Vizimolnár u. 2. X/95.

Sürgősen eladó: SEGA Mastersystem (tartozékokkal) + játék, 1 db WB III. 10.000 Ft-ért. Staub Ferenc, Tel.: 1755-240.

Vennék C-16 alapgépre játék, oktató és szövegszerkesztő programot kazetta cartridge. Lőrök Ernő, 3524 Miskolc, Hajós u. 4.

Eladó C-128 + 1571 floppy, valamint kb, 200 db játéklemez. Ár: 25.000 Ft. Hajdany Ádám, 1145 Budapest, Újvilág u. 39/b.

Keresem C-64-re 1 kazettára a Shoot'em un Construction Kit-et (Fizetek, mint a köles!). Cím: 2626 Nagymaros, Jókai u. 4. Tel.: (27) 54-342.

C-64-es programok eladók lemezen. Csere is érdekel. Érd: (Pád Softwer) 6772 Deszk, Alkotmány u. 4. Tel.: (62) 372-491.

Olcsón eladó 200 MByte játék-, ill. felhasználói program PC-re. Válaszborítékra tájékoztató. Nagy István, 1202 Budapest, Mártírok útja 151.

## NOVOTRADE—2C Kft. ÁRLISTA

### Hardverek

C-64 alapgép	13 990 Ft
VC 1541/II floppy drive	15 990 Ft
Datsette	2 990 Ft
Amiga 500 alapgép	39 990 Ft
Amiga 500 Plus alapgép	48 000 Ft
1351 Mouse (C-64-hez)	2 990 Ft
A 520 RF modulátor	3 900 Ft
Amiga tárbővítő	6 000 Ft
RocHard Hard disk illesztő	22 500 Ft
RocHard Hard disk illesztő + 40 Mbyte Winchester	50 000 Ft
RocGen Genlock RG300	10 750 Ft
Ethernet Card (A2000-hez)	35 000 Ft
MPS 1230 nyomtató	24 500 Ft
Philips 8833/II. (Amiga, C-64)	33 000 Ft

### C64 játékok

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

### Hardverkiegészítők

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

2C Áruház, Budapest XIII., Balzac u. 35.  
Tel.: 1402-954



Amiga-500 plusz eladó, vagy elcserélném. Kérek C-128 + sz.monitort. Tel.: 25-29-411.

C-64-hez jó állapotú magnó, floppy olcsón eladó. Szabó Tibor, 6300 Kalocsa, Kishalás sor 9. 4. em. 14.

Felhasználói és játékprogramok lemezen. Adás-vétel-csere. Válaszborítékért lista! 450 Ft. 1701 Budapest, Pf.: 229.

C-64 tulajdonosok. Pályázat indult. Díjak 35.000 Ft-ért. Két válaszborítékot küldj. Varsányi, Szombathely, Nagy L. u. 11.

Eladó C-64, Anitech monitor, magnó, joy, 3 super, Games, tartozékok, játékok. Nagy Csaba, 1399 Bp., Pf.: 701/893.

C-64 + 1541 floppy + magnó + 2 db joystick + Seikosha printer + 100 db programlemez + irodalom eladó. Tel.: Rácz Gyula, 153-4722/199 mellék.

C-64-es programokat cserélek és eladok. Lemezen 20 Ft/db. Felbélyegzett válaszborítékért listát küldök és kérek. Kázmér Attila, 6000 Kecskemét, Gizella tér 2. Tel.:(76) 486-257.

Eladó C-64/II. + 1541/II. floppy + Datasette + modul + lemezek + kazetták + szakkönyvek. Irányár: 27.000 Ft. Érdeklődni: az esti órákban a 99/28-826-os telefonon.

Eladó! Megkimélt C-64-II. + 1531 Datasette (10.000 Ft). Cím: Kápolnási János, 8200 Veszprém, Jutasi út 63/A. VI/38.

64-esek! Kazettára eladók felhasználói, utántöltős és egyéb programok, valamint egy ritka cartridge 4 programmal. Válaszboríték! Cím: Szilágyi Ákos, 9024 Győr, Répce út 17.

Eladó Commodore 128-as komplett számítógép. Érdeklődni lehet este 18 óra után. Tel.: 1-271-173.

C-64 bővítők 700-1600, Final III. 3000, epromégetők 2100 Ft-ért eladók. Válaszborítékért ismertető. Mikroklub, 8100 Várpalota, Pf.:65.

COVOX hangkártya PC-hez tetszetőleges kiserelésben (erősítővel, zeneszerkesztővel, modulokkal, digitalizálóval) eladó! Alap kiserelés (kovox + modulok) 2500 Ft. Erőlködővel 3500 Ft! Bármilyen menységben! 6 hónapos GARANCIA! Cím: 2626 Nagymaros, Jókai u. 4. Tel.: (27) 54-342.

Eladó 25 db új 3M lemez C-64 színvonalas játékprogramokkal 90 Ft/db áron, vagy egyben 2000 Ft-ért. Nagy István, 1202 Budapest, Mártírok útja 151.

Plus/4 programcsere és programeladás! Lemezen és kazettán! Válaszborítékért listát küldök. Több mint 1000 programból válogathatsz! Hartung Gábor, 6045 Ladánybene, Gödörállás 55.

Olcsón eladó 200 MByte játék-, ill. felhasználói program PC-re. Válaszborítékra tájékoztató. Nagy István, 1202 Budapest, Mártírok útja 151.

Videós élmények C-64-en digitalizálva. Olvassa a képűságot C-64-en 4000-féle Commodore-alkatrész. Tel.: 173-17-83.

Programok készítését vállalja a BCSB TEAM C-64-re és C+4-re. Cím: Sidló Csaba, 3242 Parádsasvár, Arany J. u. 33.

Eladók fekete-fehér és zöld monitorok 5000 Ft/db. Érdeklődni: Kázmér Attila, Tel.: (76) 486-257.

C-64-re kazettára, diskre oktatók, szótárak, könyvelői, felhasználói, játékprogramok 5 Ft-tól. Leírások 40 Ft/oldal. Egyedi programok készítését vállalom. Borítékért tájékoztatom. Földes Jánosné, 5000 Szolnok, Györffy J. u. 12. I/7.

C-64-re programokat eladok, csak lemezen! Válaszborítékért listát küldök. Czifra Zoltán, 5231 Fegyvernek, Kiss János út 18/a.

C-64 program 3 Ft-ért? Igen! Lemezre és kazettára, játékok, demók felhasználói csak nálam! Cím: Kovács Krisztián, 6750 Szeged-Algyő, Bartók Béla u. 6.

C-64 + floppy-1541 + 40 lemez + magnó + 20 db kazetta + 2 db joy. Ár: 24.000 Ft. Cím: 1191 Budapest, Hunyadi u. 21. fszt. 21. 14 óra után.

Keresek C-64-hez Artstudióhoz printconfig, GEOS-hoz eger meghajtó file, Printfox programot. Povisel I., 7135 Dunaszentgyörgy, Csapó u. 2.

C-64 programok eladása kazettán! Szuperprogramok nagyban olcsón! Írjatok! Büdi Norbert, 3900 Szerencs, Csalogány u. 34/a.

C-64-hez vadi új 1351 mouse csak 2000 Ft!!! ACTION Replay VII. 2200 Ft. 576 Kbyte 91/1-11 és 92/2-9-ig terjedő számai csak 400 Ft!!! Cím: Badenszky Csaba, 2800 Tatabánya, Fő tér 9.

Játék- és felhasználói programok eladók. Bélyeggel ellátott válaszborítékért listát és tájékoztatót küldünk. Kérjük, tüntesse fel a gépe típusát, és a hozzá tartozó perifériákat! (Vízhatlan csomagolás, kedvezmények!) (AMIGA) FRIEND TWO CREW, 1399 Bp., Pf.: 701/55.

(C-64) BORBÉLY BALÁZS-BUMBLE BEE, 1116 Bp., Sáfrány u. 44. (IBM) PROSZOLG, 1399 Bp., Pf.: 636.

Eladó 1541/II-es garanciaszalagos Commodore drive 12.500 Ft-ért, valamint 100-as lemeztartók 500 Ft/db áron. Érdeklődni lehet a 135-1289-es telefonszámon.

Az EPROM BT. A Commodore 64-re az ismert D&T cartridge családja mellett kifejlesztett egy teljesen új, két részből álló modulrendszert.

Részei:

**1./ BASIC MODUL**

Jelölése: BM.  
Helye: a C-64 ROM bővítő portja.  
Tartalma: - NYÁK csatlakozó, a cserélhető modulok számára,  
- RESET gomb.  
Fogadhat: 1 vagy 2 - csak cégünk által programozott - IC-t

A cserélhető modul helyes csatlakoztatása mechanikusan biztosított.

**2./ CSERÉLHETŐ MODUL**

Helye: a BASIC MODUL NYÁK csatlakozója.  
Megnevezése: játékprogramok esetén GAME MODUL.  
1 IC-vel 85-100 Kbyte terjedelem, 4 változat, 4-5 program/változat.  
2 IC-vel 170-200 Kbyte terjedelem, 6 változat 8-10 program/változat.  
Szoftveres menüváltás és modulkikapcsolás.

Minden termékre 6 hónap garancia + használati útmutató.

Árak: BASIC MODUL 1100 Ft,  
GAME MODUL 1000 Ft 1 IC-vel,  
2000 Ft 2 IC-vel.

Az árak az ÁFA-t tartalmazzák!

**Megrendelhető:**

**EPROM BT.**

**1046 Budapest, Török I. u. 25.**

**Tel.: (1) 1690-779 Postai utánvét!**



## ACOMP Kft.

### MÁRCIUSI

60 Ft-os

### vásárlási utalványa

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

Érvényes: 1993. március 31-ig.

Egy személy részére egyszeri vásárláshoz egy utalvány használható fel!

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

1083 Budapest, Szigony u. 9.	Tel.: 134-3153
3525 Miskolc, Fazekas u. 1-3.	Tel.: 46/321-488
5600 Békéscsaba, Bartók B. u. 37.	Tel.: 66/327-195
6724 Szeged, Csongrádi sugárút 76.	Tel.: 62/313-377
8000 Székesfehérvár, Rádió u. 15/A.	Tel.: 22/319-765
9700 Szombathely, Szalonak u. 31.	Tel.: 94/14-519

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

**NOVOTRADE**  
SZERVÍZ Kft.

## MAKROVILÁG

### utazási iroda

#### Beváltható utazás megrendelése esetén

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

## Az Országos Commodore Egyesület szolgáltatásai

### Egyesületi tagoknak 20% kedvezmény:

VC—20 memóriabővítés 3—27 kByte-os:	kiépitéstől függő
C—16, C—116 memóriájának bővítése 64 kByte-ra:	3500 Ft
C—16 belső 16 kByte-os EPROM bővítés:	1450 Ft
C—16 belső 32 kByte-os EPROM bővítés:	2900 Ft
C—16 belső 8 kByte-os SOFT—ROM bővítés:	2800 Ft
C—16 belső 32 kByte-os SOFT—ROM bővítés:	4000 Ft
C—16 8 kByte-ról 32 kByte-ra átalakítás:	2000 Ft
C—16 és 1541 kompatibilis lemezegység párhuzamosítása:	3200 Ft
SOFTROM modul 32K, kikapcsoláskor sem felejt C-16, C-116, +4	5000 Ft
FÉK C—16, C—116, +4 potméteres sebességváltóztatás	
0%-tól 100%-ig fokozatmentesen	2000 Ft
TTL IC-teszter (Cartridge+lemezen a program)	4300 Ft
+4, C—16, C—116 UNI—ROM modul különféle kiépitésekben:	
— 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
Képújság (teletext) dekóder C-64-re	10000 Ft
Epromok programozása meglévő programokkal, vagy saját hozott programok beégetésével 2716-tól 27512-ig az eprommal együtt egységesen	700 Ft
Árainkat az alkatrészárak változásai befolyásolhatják.	
A fenti bővítések megrendelhetők levélben az O.C.E. címen, valamint személyesen a havonta rendezendő klubdéllelőttön, ahol rendszeres bemutatót is tartunk.	
Postázás esetén 100 Ft postaköltség kerül felszámításra.	
A kedvezmény igénybevételéhez az O.C.E. tagsági igazolvány bemutatása szükséges.	

## MÁRCIUSI

60 Ft-os

### vásárlási utalvány

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

Érvényes: 1993. március 31-ig.

## HOBBI ELEKTRONIKA

### MÁRCIUSI

### 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!



# Elköltöztünk!

Értesítjük tisztelt ügyfeleinket,  
hogy a

**NAVOTRADE**  
SZERVIZ KFT.

Budapest V., Magyar utca 12—14. szám  
alatti szervize elköltözött.

## Új címünk:

1083 Budapest VIII.,  
Szigony utca 9.  
Telefon/Fax: 134-31-53





# ACOMP

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

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

Commodore Amiga 500	37 900 Ft	Noris MB 80 3.5" lemeztartó	690 Ft
Commodore Amiga 500 Plus	41 900 Ft	Noris DB 10 5.25" lemeztartó	150 Ft
Commodore Amiga 600	43 900 Ft	Noris DB 50 5.25" lemeztartó	490 Ft
Commodore Amiga 1200	67 900 Ft	Noris DB 100 5.25" lemeztartó	690 Ft
Commodore 1084 s Stereo-Color monitor	30 900 Ft	Noris Amiga 500 porvédő	990 Ft
Commodore A-520 TV-Modulator	3 500 Ft	Noris C-64 II porvédő	790 Ft
Commodore C-64 II	11 900 Ft	Noris MF 14 C 14" monitorfilter	590 Ft
Commodore C-128D	24 900 Ft	Noris Mouse pad	250 Ft
Commodore 1541 II Floppy	15 890 Ft	Sound Blaster Pro-2 basic	16 900 Ft
Commodore 1802 monitor	24 900 Ft	Midi Amiga Interface	2 090 Ft
Commodore 1352 mouse (eredeti Amiga)	5 990 Ft	Handyscanner Amigához	13 900 Ft
Commodore Datassette	2 500 Ft	Boot Selector Amigához	1 490 Ft
Philips 8833 II. Stereo-Color monitor	30 900 Ft	Stereo hangdigitalizáló Amigához	6 900 Ft
512 Kb órás memóriabővítő	3 200 Ft	Trackball Amigához	3 590 Ft
2.0 Mb órás memóriabővítő	14 900 Ft	Képdigitalizáló + RGB Splitter	12 900 Ft
1.0 Mb-os chip bővítő Amiga 500 Plus-ba	6 900 Ft	Rochard HD kontrollor A500/A500+	24 990 Ft
1.0 Mb-os órás chip bővítő Amiga 600-ba	7 900 Ft	(AT Bus, 0-8 Mbyte Ram)	
3.5" külső floppy drive	9 490 Ft	+ 40 Mbyte Winchesterrel	47 900 Ft
Roctec 5.25" külső floppy drive	12 900 Ft	+ 80 Mbyte Winchesterrel	50 900 Ft
Samsung 24 tűs printer	37 500 Ft	+ 1 Mbyte SIMM Ram	4 990 Ft
Quickshot II joystick	690 Ft	Amiga Magazin / Power Play újságok	450 Ft
Quickshot II plus joystick	890 Ft	NoName 3.5" DSDD lemez	550 Ft
Quickshot II Turbo Joystick	990 Ft	NoName 3.5" DSHD lemez	850 Ft
Quickshot QS - 113 analóg joystick	990 Ft	NoName 5.25" DSDD lemez	270 Ft
Quickshot QS - 123 analóg joystick	1 290 Ft	NoName 5.25" DSHD lemez	430 Ft
Action Replay MK III (Amiga) kézikönyvvel	9 990 Ft	3M 3.5" DSDD lemez	890 Ft
4 Player Adapter - 4 Joystick csatl.	1 890 Ft	Maxell 3.5" MF2-DD lemez	990 Ft
Action Replay MK VI (C-64) kézikönyvvel	5 900 Ft	Maxell 3.5" MF2-HD lemez	1 690 Ft
C-64 midi szoftverrel	6 500 Ft	Maxell 5.25" MD2-D lemez	590 Ft
C-64/c-128 mouse	2 500 Ft	Maxell 5.25" MD2-HD lemez	890 Ft
Swiftly Amiga/Atari Mouse	2 500 Ft	Profex 3.5" DSDD lemez (11 darabos)	790 Ft
Swiftly Amiga Mouse (3 gombos)	2 600 Ft	Profex 3.5" DSHD lemez (11 darabos)	1 390 Ft
Mouse/Joystick automatikus kiválasztó	1 990 Ft	Profex 5.25" DSDD lemez (11 darabos)	350 Ft
Noris MB 10 3.5" lemeztartó	150 Ft	Profex 5.25" DSHD lemez (11 darabos)	600 Ft
Noris MB 40 3.5" lemeztartó	490 Ft	Fuji 5.25" MD 2DD lemez	450 Ft

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

Nyitvatartás 9—18 óráig, szombaton: 9—13 óráig.

Vidéki vásárlóknak utánvételes csomagküldő szolgálat!

Az árajánlatban szereplő adatok, árak mindenkori változtatásának jogát fenntartjuk!