

### BÚCSÚZUNK!

A BIT-LET több mint négy éve kezdte meg „működését”. Az 51. szám – ez, amit most ön a kezében tart – mérföldkő lapocskánk történelmében. Ezzel a számmal ugyanis megszűnünk.

### VAN JÖVŐNK?

Remélhetőleg ez a megszűnés egyben valaminek a kezdetét is jelenti. A tervek, elképzelések szerint ugyanis csak most lesz igazán BIT-LET. Jövőre ugyanis önálló lapként szeretnénk megjelenni a piacon. Az első időkben aligha leszünk havilap. Célunk persze nyilván ez.

### ELŐRE!

Igyekszünk majd, hiszen már magát a tény is előrelépésnek tekintjük, hogy kiléphetünk anyalapunk kebeléből. De a lap profiljában is sokféle változást, bővülést tervezünk. Nyitni szeretnénk a profil felé is. Mert úgy gondoljuk, hogy amit nem tehetünk meg eddig havi 16 oldalon, azt megtehetjük 32–40 oldalon.

### SOSEM HÁTRA!

Ez a jelző azt is jelenti, hogy ami jó volt a 16 oldalas BIT-LET-ben, azt természetesen megtartjuk. Sőt, bővíteni kívánjuk sikeres rovatainkat, továbbra is nagyszájúak akarunk maradni, továbbra is célunk az amatőrök érdekeit szolgálni.

### KÖSZÖNET

Azoknak, akik lapunk eddigi pályafutása során bizalmat szavaztak nekünk, s törzsolvasóinkká váltak, őket szeretnénk megtartani a jövőben is. Köszönet azoknak, akik programokat küldtek lapunknak, akik cikkeikkel járultak hozzá a 16 oldal elkészítéséhez. Munkájukra az új lap létrejötté esetén is igényt tartunk. Kérjük tehát, hogy ne veszítsenek szem elől bennünket.

### JOGFOLYTONOSSÁG

Szeretnénk mindazok türelmét kérni, akiknek cikke, programja vár nálunk közlésre. Az új lap – a jelenlegi folytatása – szerkesztése során mindazokat a közlésre érdemes anyagokat szeretnénk felhasználni, amelyek ma fiókunkban vannak.

### BIZONYTALANSÁG

Persze a dolgok természetéből következik, hogy ma még nem tudjuk pontosan megmondani, hogy mikor és hány oldalon, valamint, hogy milyen áron jelenik majd meg az új lap. Mindazoknak, akik előzetes információra tartanak igényt, fölhívjuk a figyelmét a 30. oldalon található kis értesítő kártyánkra. Igérjük, hogy nem felejtkezünk el az előzetes értesítésről.

### VÉGÜL

Végül búcsúzik a szerkesztő, boldog új évet kíván minden BIT-LET olvasónak, s első ízben jelzi, hogy a jövő havi megjelenés dátumát hiába keresik a melléklet 16. oldalán.

### BELÜLRŐL

- 26 **Hiroldal** – amelyben ezúttal is találnak érdekességeket és kevésbé érdekességeket.
- 28 **Rotary** – aki ismeri ezt a szellemes játékot, annak azért, aki nem ismeri, annak azért érdekes a játék. A szabályokat nem ismétljük, megjelent szeptemberi számunkban.
- 31 **Mátrix 26** – egy táblázatkezelő program Spectrumra – nem kevés programozói tanulsággal.
- 34 **A GEOS-ről** – azoknak, akik a GEOS-t még nem ismerik, s azoknak, akik a GEOS-t már jól ismerik.
- 35 **Szoftverötlet** – kör Laser BASIC-ben. Egy spectrumos bővítés apró hiányosságát szünteti meg e program.
- 36 **Programajánlat** – gépi kódú programok DATA sorokban – mindez Primora.
- 37 **Posta** – néhány praktikus kérdés – néhány praktikus válasz – valamint néhány provokatív kérdés – válasz nélkül.
- 38 **Könyvmoly** – a megjelent új könyvek közt két érdekességre bukkantunk. Végre találtunk egy Atari könyvet, s mi végre találtunk egy ránk hasonlító könyvet.
- 40 **Tapasztalatnyerő** – Ennek a pályázatnak az eredményeivel még adósok voltunk. A beérkezett programok közül is közlünk részletet az egyikből. (Cseles egy darab.)

www/  
Béha

# HIRODAL



## KALIUMNIÓBÁT

gy kutatói program keretében a zürichi műszaki egyetem szilárdtest-fizikai laboratóriumában sikerrel növesztenek káliumnióbát-kristályokat. Ezeknek a nagy értékű mesterséges kristályoknak az előállítását — kutatók véleménye szerint — márföldkő az optikai rendszerek fejlesztésében. Ezek segítségével a számítógép elektronikus vezérlését majd káliumnióbát kristályokat a világon eddig még el nem ért tisztaságban előállítani, és ezeknek nagyon jók az optoelektronikai tulajdonságai.

## TELEFONKÖNYV

Másfél évvel ezelőtt döntött a Magyar Posta arról, hogy a korábbi bontás helyett megyénként adja ki a telefonkönyveket. Az első hat megyei telefonkönyv elkészülte után a Posta megállapodást kötött az Idegenforgalmi Propaganda és Kiadó Vállalat TD képviselőivel, s az egyetemes változást hozta a névsorok előállításában. A megállapodás alapján ugyanis az év márciusa óta az IPV TD számítógépe végzi a telefonkönyv-tördelést, szedését. A gép körülbelül öt percenként állít össze egy oldalt, amiről lézermásoló készít nyomdakész levonatot. Ezt lépezetesen a számítógép folyamatosan fotózza, és a filmről sokszorosítja a telefonkönyv lapjait. A számítógép folyamatosan ellenőrzi az adatokat és jelez, ha hibára akad. Előny az új módszernek az is, hogy jóval gyorsabb, mint a hagyományos eljárás: egy óha több mint egy hasábnyi kézi tördeléshez szükséges. A gyors átfutási időnek köszönhetően a legfrissebb adatváltozások is szerepelnek az új kötetekben.

## SZOLFÉZS

A zenetanítás egyik alaptárgya a szolfézs, az alapos elsajátítására viszont a tapasztalatok szerint a zeneiskolákban rendelkezésre álló heti két óra nem elegendő. Az otthoni és iskolai gyakorlást teszi most hatékonyabbá és könnyebbé az a számítógépes programcsomag, melyet alkotott az Egeri Állami Zeneiskola tanára alkotott meg és a Tudományszervezési és Informatikai Intézet forgalmaz. A Commodore 64 és 128 típusú számítógépeken futtatható programok használatát nem igényeljük, a technikai jártasságot, ám segítségükkel a hangközökkel, hangzatokkal, hangnemekkel, és fejleszthetők dallamirás, készség-

## NDK CHIP

Új chip gyártását kezdték meg az NDK-ban. A saját fejlesztésű, 256 kilobit kapacitású integrált áramkörök készítésére ma még csak néhány vezető tőke ország képes. Az NDK-ban jelenleg 120 ezren dolgoznak a mikroelektronikai alkatrészek, részegységek kutatásában, fejlesztésében és gyártásában.

## AUTOMATÁBÓL

Az Egyesült Államokban szoftvert is lehet kapni automatából. A Computerland cég 550 üzletben olyan automatákat állított fel, amelyekből a vevő 1000-nál több programot választhat ki. A programokat optikai lemezes memóriában tárolják. A vevőnek először bemutatták a szoftvert ismertető rövid videofilmet, azután megvizsgálhatja a kiválasztott programot és ha megfelel, készíthet róla hálékony lemezre másolatot. Ezt egy rövid leírással együtt megkapja. Az eredeti kézikönyv néhány nap múlva postán érkezik.

## VIVAS

1987. október 1-jétől a Datacoop lett a Magyar Vívó Szövetség hivatalos támogatója. Ami az információban a legfontosabb és leglényegesebb: a számítástechnikai kiszóvetkezést havonta százezer forinttal támogatja a szövetség, s miután a megállapodás öt évre szól, a szövetség szerint a prosperáló cég számára kitűnő reklámhordozónak tűnik. Azért kötötték az 1992. december 31-ig érvényes megállapodást. A szerződés természetesen a szövetségi eredményt és a magyar vívóválogatot, amely az utóbbi időkben több jelentős nemzetközi eredményt ért el, tartalmaz. Vállalta, hogy a szövetség és a kadett válogatot tagjai — az Olimpia és a hazai nemzetközi viadalokon melegen és a hazai nemzetközi viadalokon helyen viselik a Datacoop feliratot. Ez a jelenlegi szabályok szerint három, tízszáz-tíz cm-es vívóruhán, és a trikón is elhelyezik az új szponzor nevét. Abban is megállapodtak, hogy az MVSZ kötelezi a nemzetközi és hazai viadalokon döntőbe jutott, nem válogatott versenyzőket is a Datacoop felirat viselésére.

## TÖRÖLHETŐ LEMEZ

A Sony eredetileg informatikai adattároló céljára fejlesztette ki törölhető és újraintható optikai lemezét. A 13 cm átmérőjű lemez oldalanként 300 millió byte-ot tárol — ez 200 000 gépelt oldalal egyenértékű — szemben a kompaktlemez egymillió byte-jával. A lemez nem jelent konkurenciát a videoszalagnak, a mozgókép egyébként is hatalmas mennyiségű információ tárolását igényli. Komoly versenytársává válhat azonban a kompaktlemeznek.

## JAPÁN ROBOTOK

Japánban napjainkban már százezer teljesen automatizált robot működik. A gyárakban az összes szerelési munka 20 százalékát az ezek végzik. Ezzel szemben például az Egyesült Államokban csak húszezer robotot foglalkoztatnak, s ezek felét is a gépjárműgyártásban. A Fanuc japán robotgyártó, ezen az évben már havi ezer robotot gyárt, ezeken kívül az egész világon készülő különböző géptípusokhoz különleges szerzőm gépeket s központi vezérlő berendezéseket. A kutatólaboratóriumokban pedig olyan gépeket dolgoznak ki, amelyek képesek lesznek önállóan a tűzoltók. s az atomerőművekben is dolgozni tudnak.

CLN

**GYÓGYSZER**

A Szovjetunió Gyógyszerészeti Tudományos Kutató Intézetében kidolgozták a Gyógyszer nevű számítógépi rendszert, amelytől az orvosok naprakész információt kaphatnak a forgalomban lévő gyógyszerekről, azok klinikai alkalmazásáról, arról, hogy mely készítmények helyettesíthetik egymást és – összeférhetetlenség miatt – melyek nem adhatók egyidejűleg a betegnek. A rendszer felvilágosítást ad a gyógyszerek ellenőrzésének és tárolásáról, minőségéről is. A még az idén gairól, minőségéről is. A még az idén üzembe helyezésre készültek majd az új nyát rendszeresen kiegészítik majd az új információk megjelenő bel- és külföldi gyógyszerekészítmények adataival. Az új információs rendszer előnyeit nemcsak a moszkvai egészségügyi intézményekben használhatják, hanem a fővároson kívüli központokban is, ahová az adatokat mágnesszalagon juttatják el.

**TRÓFEABÍRÁLAT**

A számítástechnika rohamos elterjedése következtében életünk egyre több területén találkozhatunk számítógépekkel. Tavaly az Országos Trófeabíráló Bizottság Szolgálat központja és a MÉM Erdőrendezési Szolgálat arra az elhatározásra jutott, hogy számítógépet használ a bírálóbizottságok munkájának könnyítésére és pontosítására. A számítógépes bírálat esetében a méretfelvételeket ugyanúgy el kell végezni, mint a hagyományos módon. A méretek itt is törzkönyvbe kerülnek, de a számítógép pillanatok alatt kiszámolja a trófea pontszámát, és az adatokat mágneslemezen tárolja. A mágneslemezes tárolás előnye, hogy sok adatot lehet nagyon kis helyen tárolni, és ki lehet küszöbíteni a számológép segíti az adatfeldolgozást is. A gép a mágneslemezen tárolja az adatokat a meghatározott program szerint, és azokat gyorsan feldolgozza. Ebben az esetben is sok mechanikus számológép az OTB mellett a győri, a keszthelyi, és a keszthelyvári bírálókat is bevezetik a számítógépes bírálatot. A bíráló- és adatfeldolgozó programok egységesebbé teszik a trófeabírálo bizottságok mágneslemezen tárolt adatait az OTB-nál gyorsan és pontosan lehet feldolgozni. Így az országos adatfeldolgozás is jóval gyorsabban végezhető el, mint a hagyományos módon. Számítógép segít a vadfajok világranglistájának és azok naprakész listájának elkészítésében és azok naprakész vezetésében is. Az OTB tervezi, hogy a számítógép segítségével tájékoztatni tudja a vadászatókat a vidéki trófeabírálo bizottságoknál bemutatott külföldi és hazai vadászó álltal eléjett vadfajok trófeáinak adatairól is.

ú!



**AST PREMIUM/386**

Az amerikai AST Research Corporation cég legújabb produkciója az Intel 80386 típusú mikroprocesszorra alapozott, AST Premium/386 névre keresztelt számítógépe. Az első 20 MHz-es, 32 bites számítógép memóriája 2-től 13 Megabyte-ig bővíthető. 40, 90 és 150 Megabyte-os fix lemezegységek kapcsolható hozzá. Beépített floppy egysége 5 1/4 inches, 1,2 Megabyte-os. Kétúdon üzemeltethető MS OS/2 és XENIX operációs rendszerek alatt. Az új számítógép különösen előnyösen alkalmazható CAD/CAM grafikus feladatokra és hálózati rendszerekben.

**ORVOSKÉPZÉS**

A budapesti Semmelweis Orvostudományi Egyetemen az orvostanhallgatók képzésében már rendszeresen alkalmaznak olyan számítástechnikai programokat, amelyek segítségével a medikusok alaposabban elsajátíthatják a tananyagot. A személyi számítógépekre alkalmas programokból pedig huszonötöt készítettek a Biofizikai Intézet munkatársai, s ezek közül többnek az összeállításában a hallgatók is részt vettek. A programok segítségével tanulják meg a medikusok többek között az EKG-görbék keletkezését, s modellezik a vérkeringést. Megismerkednek az orvostudományban alkalmazott legfontosabb műszerek felépítésével, működésével, valamint a számítógépek gyógyászati felhasználásának lehetőségeivel. A hallgatók a programokat a laboratórium gyakorlatokon hetente rendszeresen tanulmányozzák, de ezenkívül önszorgalomból bármikor újból áttekinthetik. Az eddigi tapasztalatok szerint a szemléletes módszerrel a medikusok könnyebben sajátítják el a tananyagot.

**NYELVTANULÁS**

Nyolchetes, bentlakásos angol nyelvtanfolyamot indított október közepén a TIT soproni szervezete. A tanfolyam hallgatói hazánkban egy eddig még nem alkalmazott módszerrel tanulják meg az idegen nyelv alapjait. Az új módszer az OTB-nál gyorsan és pontosan lehet feldolgozni. Így az országos adatfeldolgozás is jóval gyorsabban végezhető el, mint a hagyományos módon. Számítógép segít a vadfajok világranglistájának és azok naprakész listájának elkészítésében és azok naprakész vezetésében is. Az OTB tervezi, hogy a számítógép segítségével tájékoztatni tudja a vadászatókat a vidéki trófeabírálo bizottságoknál bemutatott külföldi és hazai vadászó álltal eléjett vadfajok trófeáinak adatairól is.



meghaladták a tartalom előállításának költségeit stb.) ”

Gratulálunk Nagy Lászlónak a szerintünk is igen magas szintű, gondolkodtató, élvezetes és épp ezért jobb sorsra érdemes játékhoz. Nagyon sajnáljuk, hogy ez a játék is a magyar találmányok nagy részének sorsára jutott. Szerintünk a játék ára a mai játékarakhoz viszonyítva nem volt kiugróan magas, valóban a reklám hiánya (és a csomagolás) lehetett a kudarc egyik fő oka, a másik talán a kitarítás hiánya (mármint a MÜTEX részéről), ugyanis tapasztalataink szerint a magyar játékpiacon nem ritka jelenség, hogy egy játék csak 2-3 évvel árusításának megkezdése után fut be igazán (ez a késési idő még a Bűvös kockánál is kb. két év volt). Nagy László igen jól játszó programját a Commodore újság szeretné közölni.



## A PROGRAM

Mi most **Dévényi Imre** programját közöljük (kisebbs változtatásokkal), mely ugyan egy kicsit gyengébben játszik a másik kettőnél, képernyőképe sem olyan szép, viszont csak fele olyan hosszú, mint azok.

A program a játék tábláját egy 8x8-as T% tömbben tárolja (ábra a köv. o.), melynek sorai a tábla jobbra felfelé haladó vonalainak, oszlopai pedig a tábla vízszintes sorainak felelnek meg. A tömb egy elemének értéke 0, ha a megfelelő helyen nincs korong, +1 ill. -1, ha ott fekete, ill. fehér korong van. Ennek a +-1-es tárolásnak hatalmas előnye, hogy egy üres lyuk 6 szomszédjának a tartalmát összeadva az eredmény előjele megmutatja, hogy ki forgathat ott. A kiindulási helyzet a 4-14-es DATA sorokban van leírva. A megjelenítés karaktergrafikával történik, s emiatt a tábla képe egy kicsit torz, viszont a megjelenítés nagyon gyors. Mivel (főleg a karaktergrafika miatt) a program listája „tele van” grafikus karakterekkel, ezért a megfelelő sorok végére megjegyzésként odaírtuk, hogy az ott szereplő grafikus karakterek hogyan állíthatók elő. Ha egy jel több egymás utáni sorban is szerepel, akkor csak az első sorba írtuk ki a jelentését; a C < a commodore-billentyűt jelöli. Természetesen ezeket a jelmagyarázó REM részeket begépeléskor érdemes elhagyni.

A program játékstratégiája elég egyszerű. Először is a tábla egyes mélyedéseihez az ábrán látható alapértékeket rendeli. Ezek az alapértékek azt mutatják, hogy mennyire fontos az a mező, mennyire kell törekednünk az illető mező golyóval történő elfoglalására. Az állások kiértékeléséhez egy értékelő függvényt használ, mely egy játékállást úgy értékel ki, hogy összeadja az egyes mezők tartalmának és alap-értékének a szorzatát. (Emlékezzünk rá: -1-et akkor tartalmaz egy mező, ha az adott helyen fehér golyó van, +1-et pedig akkor, ha fekete; 0 akkor szerepel táblázatunkban, ha üres a mező.) Ugyanezt végrehajthatjuk úgy is, hogy megszámloljuk, mennyivel több fekete golyó van a 2-es értékű lyukakban,

mint fehér. (Ez a különbség lehet negatív is.) Az így kapott számot két-tővel megszorozzuk, majd ugyanezt kiszámítjuk a 4-es, 8-as, ill. 33-as értékű lyukakra is, - itt, persze, értelem-szerűen a 4-gyel, 8-cal, ill. 33-mal szorozzuk meg a különbséget, és ezután a négy beszorzott értéket összeadjuk.

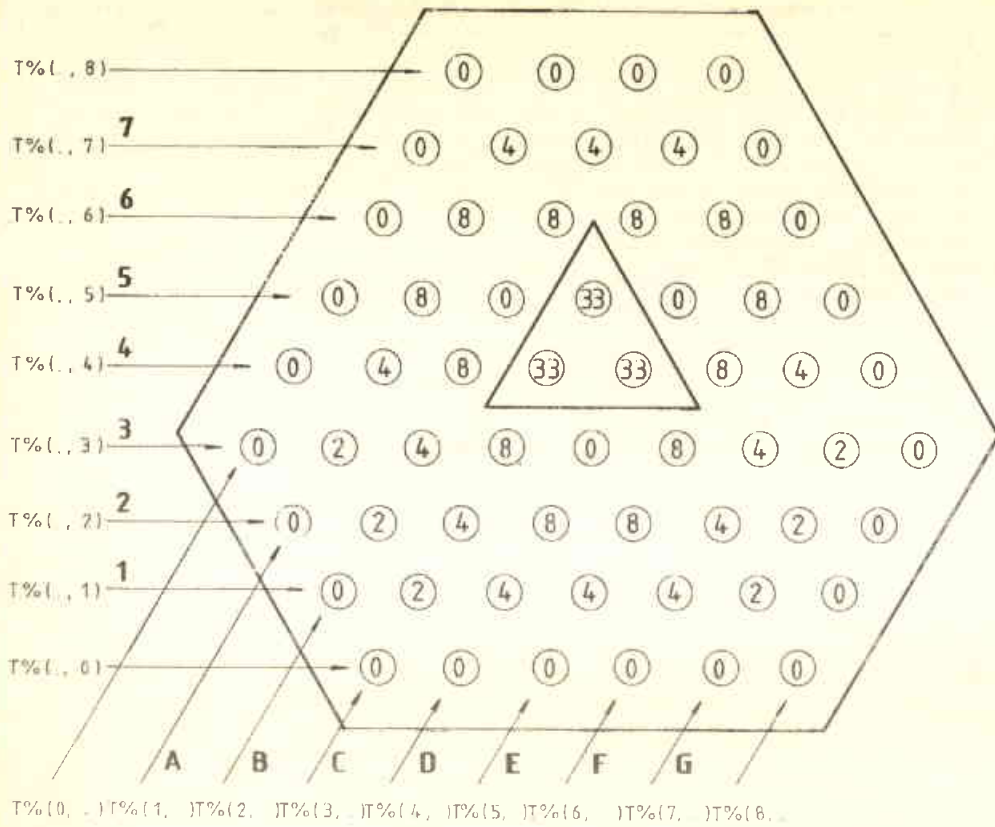
Mivel a gép (ha játszik) a világos golyókat kezeli, és ezeket a T% tömbben -1 jelzi, így a lehetséges lépések közül a gép azt fogja lépni, amely a lehető legkisebb értékű helyzetet állítja elő. (Ez a végigpróbálgatás a 302-336 sorokban található, egy helyzet kiértékelését a 358-362 rutin végzi.)

**Fontos észrevétel:** mivel azt a lépést keressük, amely a pillanatnyi helyzet értékén minél többet

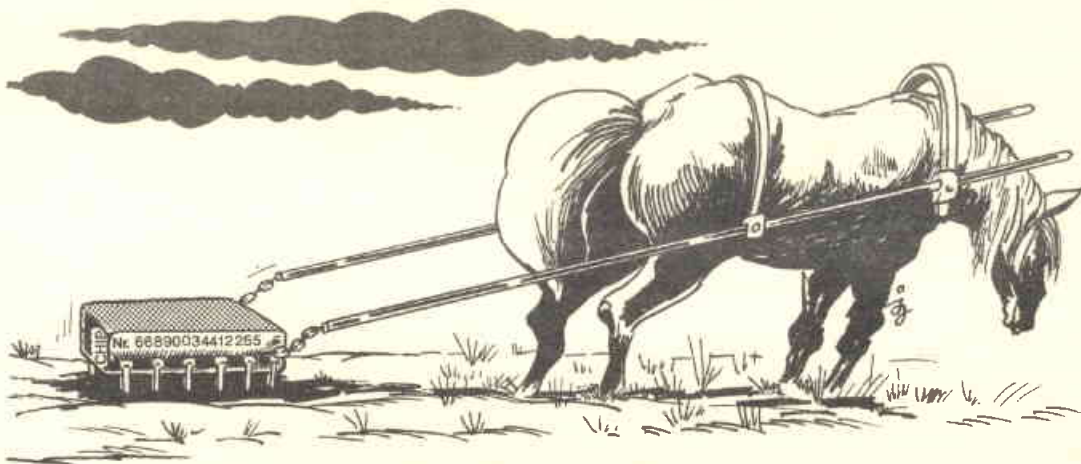
```

100000 CHAR,35,25,B4+"-
100001 GETKEYB$:IF B$C"0"OR B$C"8"THEN100
100002 IF B$="0"THENCHAR,35,25," "GOTO150
100003 CHAR,37,23,B$(F=VAL(B$))
100004 FORI=1 TO F
100005 DWSUB 228
100006 GOSUB252: SOUND 1,776+20*I,3:NEXTI
100007
100008 FORI=1TO208:NEXTI
100009 GOSUB290: GOSUB276
100010 CHAR,8,22,BL$(CHAR,8,25,BL$
100011 GOSUB292:REM KIERTÉKELÉSHEZ
100012 IF GY THEN CHAR,2,22,B$(GY+2)+ " JATEKOS NYERT "GOTO154
100013 G=5+-1:GOTO136
100014 VOLT:FORI=1TO4: SOUND1,500+50*I,10+2*I:NEXTI
100015 CHAR,2,23,"AKKORZ UJRA JATSZANI? (Y/N)"
100016 GETKEYB$:IF B$="Y"THEN206
100017 IF B$C"N" THEN156
100018 CHAR,8,23,BL$
100019 CHAR,8,23,"RR NEM HAT NEKI":END
100020 RESTORE:FORI=8TO2:FORJ=3TO8:READR
100021 TX(I,J)=R:NEXTJ,I
100022 GOSUB136:GOTO134
100023
100024 REM *** LETEHETO? ***
100025 N=0:IF+Y<4OR+Y<11THENRETURN
100026 IF TX(CX,Y)<0THENRETURN
100027 SZ=TX(CX,Y+1)+TX(CX,Y-1)+TX(CX-1,Y)+TX(CX+1,Y)+TX(CX+1,Y-1)
100028 IF SZ<0AND G=1THEN N=1:RETURN
100029 IF SZ<0AND G=-1 THEN N=1:RETURN
100030 RETURN
100031 ZK=TX(CX,Y+1):REM ***FORGATHO***
100032 TX(CX,Y+1)=TX(CX-KF,Y-(KF=1))
100033 TX(CX-KF,Y-(KF=1))=TX(CX-KF,Y+(KF=-1))
100034 TX(CX-KF,Y+(KF=-1))=TX(CX,Y+1)
100035 TX(CX,Y-1)=TX(CX+KF,Y-(KF=1))
100036 TX(CX+KF,Y-(KF=1))=TX(CX+KF,Y+(KF=-1))
100037 TX(CX+KF,Y+(KF=-1))=ZK
100038 RETURN
100039 REM *** UJ ALLAPOT RAJZOLAS ***
100040 CHAR,CX(CX,Y+1),2*(9-(Y+1)),6*(TX(CX,Y+1)+2)
100041 CHAR,CX(CX-1,Y+1),2*(9-(Y+1)),6*(TX(CX-1,Y+1)+2)
100042 CHAR,CX(CX-1,Y),2*(9-Y),6*(TX(CX-1,Y)+2)
100043 CHAR,CX(CX,Y-1),2*(9-(Y-1)),6*(TX(CX,Y-1)+2)
100044 CHAR,CX(CX+1,Y-1),2*(9-(Y-1)),6*(TX(CX+1,Y-1)+2)
100045 CHAR,CX(CX+1,Y),2*(9-Y),6*(TX(CX+1,Y)+2)
100046 RETURN
100047 CHAR,8,9,CHR$(129)+LEFT$(F$,17)+ " "+LEFT$(LF$,18)
100048 CHAR,8,10,LEFT$(LF$,17)+ " "+LEFT$(F$,18)
100049 CHAR,16,16,LEFT$(F$,17)+ " "+LEFT$(LF$,8)
100050 CHAR,16,11," "+CHR$(144) *REM "-"=SHIFT C
100051 RETURN
100052 CHAR,CX(CX,Y),2*(9-Y),CHR$(144)+R0$:RETURN
100053 CHAR,CX(CX,Y),2*(9-Y),CHR$(5)+R0$:RETURN
100054 CHAR,CX(CX,Y),2*(9-Y),CHR$(129)+R0$:RETURN
100055 SZ=TX(C,4)+TX(C,5)+TX(C,4):REM ERT
100056 IF SZ=8 THEN GY=1:GOTO300
100057 IF SZ=-8THEN GY=-1:GOTO300
100058 GY=0
100059 RETURN
100060 CC=-999:KF=1:PR=36
100061 RESTORE 28
100062 CHAR,2,23,"RZ EN LEPSEN? " *REM "-"=CC 2
100063 IF CC6THENPR=30
100064 FORI=1TOPR:READCX,Y:GOSUB216
100065 IFN=0 THEN330
100066 GOSUB358:PE=E
100067 FORJ=1TO5:GOSUB228
100068 GOSUB358:C=PE-E:IF CC=05 THEN 326
100069 R=X:B=Y:F=J:CS=C
100070 GOSUB292:IFGY=-1THENI=37
100071 NEXTJ
100072 GOSUB228
100073 NEXTI
100074 IF F=3 THEN F=5-F:KF=-1
100075
100076 K=R:Y=B:B$=CHR$(64+R0)+" "+STR$(CY)
100077 CHAR,16,23,B$
100078 IF KF=1 THENB$=" JOBBRA"
100079 IF KF=-1 THENB$="BALRA"
100080 CHAR,21,23,"FORGATHO "+B$+STR$(CF)
100081 FORI=1TO300:NEXTI
100082 GOSUB288
100083 FORI=1TO1500:NEXTI
100084 GOTO172
100085 E=P(CX,Y+1)+TX(CX,Y+1)+P(CX+1,Y)+TX(CX+1,Y)+P(CX+1,Y-1)+TX(CX+1,Y-1)
100086 E=E-P(CX,Y-1)+TX(CX,Y-1)+P(CX-1,Y)+TX(CX-1,Y)+P(CX-1,Y+1)+TX(CX-1,Y+1)
100087 RETURN

```



csökkent, így csak a két helyzet értékének a különbségére vagyunk mindig kíváncsiak. Ehhez pedig elég az éppen próbált letevés körüli értékeket kiszámítani – hiszen a tábla többi mezőjének értéke úgyszemint változik. Tehát egy forgatás akkor jó, ha a körülötte álló fehér golyókat minél nagyobb, a feketéket pedig minél kisebb értékű lyukakba forgatja át. Pl. a 2. ábrán a C3 lyuk körüli részt látjuk, az alapértékeket a lyukak mellé írtuk. Ha most a forgatót C3-ra helyezzük, s egyet jobbra forgatunk, akkor a fehér golyók a 8, 0, 4 értékű lyukakból a 33, 8, 4 értékűekbe vándorolnak, míg a feketék a 33 és 8 értékűből a 0 és 4 értékűbe, így az állás értéke  $33-8+8-0+4-4-(0-33+4-8)=70$ -nel javul (azaz a gépben 70-nel csökken). A program kezelése igen egyszerű, s futás elején ki is íródik a képernyőre. A korong letevésének helyét az ábrán vastag betűkkel szedett koordináták segítségével adhatjuk be – ezek játék közben a képernyőn is láthatók. Az elforgatást az irány (B=balra, J=jobbra) és egy 1 és 3 közötti szám segítségével adhatjuk be, mely szám azt mutatja, hogy hányszor hatvan fokkal akarjuk a korongot elforgatni. Mindenkinek jó játszadozást kívánunk!



**ÖRTE  
TESÍTŐ ÉRTE  
ÉRTESÍTŐ ÉRT  
SÍTŐ ÉRTESÍTŐ  
TESÍTŐ ÉRTES  
SÍTŐ ÉRTESÍTŐ  
ÉRTESÍTŐ É**

Lapunk törzsolvásóinak fölajánljuk azt a lehetőséget, hogy előzetes értesítést küldünk az önálló BIT-LET megjelenése előtt. Az értesítésben jelezzük majd a várható utcára kerülés időpontját, a lap pontos árát, s némi előzetest az első lapszám tartalmából. Amennyiben igényt tart erre az előzetesre, kérjük töltsse ki ön is az alábbi cédulát, s küldje el az Ötlet szerkesztőség címére: Ötlet-BIT-LET Budapest 1986

Név: .....

Város: .....

Cím: .....

Irányítószám: .....



# ZX SPECTRUM

TÁBLÁZATKEZELŐ PROGRAM ZX SPECTRUMRA

zor, és mindig látható marad. Megszüntetése B újbóli megnyomásával lehetséges.

**C:** újrászámítja a teljes táblázatot.

**D:** sor vagy oszlop kitörése a táblázatból.

**F:** a számok kijelzési formátumának beállítása.

**G:** ugrás az A1-Z26 box bármelyikébe.

**H:** help

**I:** sor vagy oszlop beszúrása

**J:** táblázat betöltése kettétről.

**R:** az aktuális box tartalmának ismétlése vízszintes vagy függőleges irányban abszolút vagy relatív módon.

**S:** a táblázat kimentése kettétrá.

**Z:** a képernyő kinyomtatása ZX printerre.

**SZINTAXIS:**

**Box megadása képletben:** X(nn) Pl: A(01)

**Függvények bevitole** a funkcióbilleentyükkel lehetséges (CAPS-SYM SH).

```
...
FOR I=1 TO 20
  FOR J=1 TO 20
    ...
  
```

```
...
PRINT "TÁBLÁZATKEZELŐ PROGRAM ZX SPECTRUMRA"
...

```





# A GEOS-RÓL

## AZOKNAK, AKIK A GEOST MÉG NEM ISMERIK, ÉS AZOKNAK, AKIK A GEOST MÁR JÓL ISMERIK

1986 óta egy rendszerprogram hódította meg a C 64 kedvelőit, a GEOS.

Hogy mi is az a GEOS? Mint a legtöbb programnév, ez is betűszó: Graphics Environment Operating System, amelyet grafikus parancstáblával rendelkező grafikus operációs rendszernek fordíthatunk. A grafikus parancstábla alatt azt értjük, hogy nem kell parancsokat begépelni, hanem valamilyen ábra, felület kiválasztásával tudunk parancsot kiadni. Ezt vezérelhetjük joystickkal, egérrel és egyéb más bemeneti eszközökkel.

### MIÉRT JÓ A GEOS?

**1. Az összes lemezműveletet egyszerűen megvalósíthatjuk,** ez nagyon emberséggé teszi a gép kezelését. Jó példa erre a file-ok törlése:

**C 64 módban:** OPEN15,8,15,"0:program név":  
CLOSE15 parancssort kell kiadnunk.

**GEOS-ban:** a képernyő alján található egy szemetkosár rajz, amibe a kijelölt adatokat „kidobhatjuk”. Hátránya ennek a parancsszervezésnek, hogy korlátozza a drive lehetőségeinek kihasználását. Nem tudjuk letörölni például a lemezen levő összes H betűs filet egy parancssal, amelyet C 64 üzemban egyszerűen megtehetünk (print#15,"s0:H\*"), kénytelenek vagyunk egyenként törölni minden egyes filet.

**2. Egységes rendszerbe foglalták a különféle felhasználói programokat,** ezek meghatározott felületen illeszkednek egymással. Ez alatt a felület alatt az általános adatszerkezetet, és programozási szintet értjük, amelyet a VLIR fileszervezés, és az ugrótábla jelent. Így például a rajzolóprogram rajzait felhasználhatjuk a szövegszerkesztőben, a táblázatkezelő adatait átvihetjük a szövegszerkesztőbe, a rajzolóprogramba és így tovább...

Egy ilyen egységes rendszer következményei szinte elképzelhetetlenek, sok felesleges munkát tudunk vele megtagarítani.

Az eddigi Commodore programok közül a Print Box az egyetlen, amely kevés sikerrel, de próbálkozik az egységesítéssel. Bár ez cseppet sem hasonlítható ehhez a szintű általánosítási törekvéshez.

**3. A GEOS a fejlett programozási irányzat, a moduláris programozás jegyében készült.** A modulok leírása, az egész rendszer felépítése jól dokumentált, könnyen fejleszthető. Nem új ez a módszer, már az UNIX-ki-fejlesztését is így csinálták, csak eddig kispépen még nem talákoztunk hasonló megoldással. Általában a rendszerleírást késvé adja ki a gyártó cég, esetleg más által visszajevtve jut a nyilvánosság elé. Az UNIX-nál kezdődött, hogy a rendszerleírás tartalmazza a forrásprogramot és a felhasználó ezt már úgy alakíthatja, ahogy akarja. A GEOS is hasonló, bár speciálisan egy processzorra (6510) készült. A rendszer leírása azonnal megjelent, nem titkolóztak a programozói fogásokkal, a felhasználási lehetőségekkel.

**4. Új és gyors file-kezelést hoztak létre a VLIR file-ok előállításával.** A VLIR-file legnagyobb előnye, hogy az ember sokkal hosszabb programot írhat, mint a C 64-es tárkapacitása. Különálló részekre osztjuk fel a

programot, melyeket azután a GEOS szükség esetén betölt. Ha például rákérdezőnk egy C 64-es file-ra, és a GEOS azt az információt adja, hogy a file szekvenciális, akkor ez nem a megszokott értelemben vett soros elérésű adatfile, hanem csak arra utal, hogy szekvenciális tárfelhasználásról van szó. Ez olyan, mint ahogyan a BASIC-programok tárolódnak. A GEOS által használt VLIR-file-ok jelentése: „Variable Length Indexed Record” (változó hosszúságú indexelt rekordok)

A VLIR továbbfejlesztése a relatív tárkezelésnek. Egy file-t nem kell egy darabban betölteni, hanem több kisebb, különböző hosszúságú részletben. Előnye, hogy a felhasználó csak azokat a programrészeket tölti be a tárbba, melyekre feltétlenül szüksége van. Jó példa erre a GEO-PAINT, illetve a vele előállított képek. Tehát egy VLIR-mutatóblokk a különböző programrészekre mutat, melyeket csak szükség esetén tölt be, és a VLIR file-ok nemcsak programok, hanem adatfile-ok esetén is használhatók. Pl. a Notesz-file minden VLIR-mutatója a notesz egy-egy lapjára mutat. A VLIR mutatóblokk max. 127 rekordra mutathat. (256 Byte hosszú lemezblokkot feltételezve).

### A GEOS RUTINOK

Ha megismertük a GEOS mellé adott programokat (geoPaint, geoWrite) akkor vizsgáljuk meg, milyen rutinokból áll, és milyen a memóriafelosztás a GEOS alatt. A memóriafelosztás alapján láthatjuk, hogy a GEOS mellett csak gépi programok férnek el.

Ugyanis egy elég kicsiny memóriaterület áll a rendelkezésünkre (\$0400-\$5FFF). Ebben igazán jó programot csak úgy írhatunk, ha kihasználjuk a GEOS már bentlévő rutinjait.

**A rutinokat funkciójuk szerint több nagy csoportba oszthatjuk:**

- Grafikus rutinok
- I/O rutinok
- Belső rendszervezérlő rutinok

Ezeket kisebb csoportokra bonthatjuk:

**A grafikus rutinok:**

- **alapvető grafikus feladatok:** pont kigyújtás, eloltás, vizsgálat, vonalhúzás, törlés, invertálás stb. A két képernyő közötti műveletek.
- **ablakkezelés:** ablak felépítése, szöveg kiírása az ablakba, a kiválasztható ablakocskák (OK, YES stb.) kirajzolása, lekérdezése.
- **menükezelés:** a vízszintes, ill. függőleges menük felépítése, lekezelése.

**Az I/O rutinok:**

- **lemezműveletek** (direktory kezelése, validálás stb.)
- **file-műveletek** (törlés, átnevezés stb.)

**A rendszervezérlő rutinok:**

- **Hibavizsgálatok, üzenetek.**
- **Konfiguráció beállítása**

A rutinok vezérléséhez paraméterek kellene. Ezeket a paramétereket a regiszterekben, de az utasítást (rutin-hívás) követő táblázatban is átadhatjuk.

Mint tudjuk, a 6510-nek három regisztere van, és egy összetettebb funkció hívásához több mint három paraméter kell. Ezért a GEOS alatt létrehoztak 16 kiegészítő 16 bites regisztert a nullás lapon (S02-S22).

**A másik nagy változás a GEOS-ban a lemezen tapasztalható.** A normál DOS által üresen hagyott területeket használják ki a programok egyéb jellemzőinek (idő, dátum stb.) tárolására. Sajnos az egy programhoz tartozó információs blokk külön egy szektori helyet foglal, így kevesebb a felhasználható lemezterület. Ebben az információs szektorban van letárolva a filet jellemző ICON, töltési cím, az információs táblában található üzenetek.

**Memórfelosztás GEOS alatt:**

- \$000-\$007F GEOS KERNAL által használt nullás lap
- \$0080-\$00FF Az applikációk által használható nullás lap.
- \$0100-\$01FF Processzor stack
- \$0200-\$03FF Nagyrészt nem használt.
- \$0300-\$0333-ig a normál C 64 rendszer vektorok találhatóak itt. Ezeket a vektorokat a GEOS mindig meghatározott értékre állítja be.
- \$0400-\$05FF Application programok számára

fenntartott terület. Ha a DESK-TOP-hoz tartozó rutinokat is szeretnénk futtatni, használjuk a \$4570-\$496E területet.

**\$6000-\$7F3F 2. Képernyő.** Ide menti el a GEOS KERNAL az ablakok mögötti területet, vagy pedig a menük alatti részt.

**\$7F40-\$7FFF** Applikációk számára szabad hely.

**\$8000-\$80FF** Pufferterület a floppy adatoknak.

**\$8100-\$81FF 2.** Pufferterület (az aktuális info-sektort tartalmazza)

**\$8200-\$82FF 3.** Pufferterület (a GEOS által használt BAM)

**\$8300-\$83FF 4.** Pufferterület (a track-sector mutatókat tartalmazza töltés és tárolás esetén)

**\$8400-\$8BFF** GEOS KERNAL adatai (JOB címek, filenév, disk név, sprite adatok)

**\$8C00-\$8FE9** Szín RAM

**\$8FF8-\$8FFF** A sprite adatok mutatói

**\$9000-\$9FFD** GEOS KERNAL (a program)

**\$A000-\$BF3F 1.** képernyő, az, amelyet a képernyőn látunk.

**\$BF40-\$FFFF** GEOS KERNAL

*Honti Tamás*

## SZOFTVER ÖTLETEK



### 3 DIMENZIÓS TÖMBÖK ENTERPRISE-ON

Az Enterprise IS-BASIC-jének egyik hibája, hogy csak maximum kétdimenziós tömböket tud használni. Ez legtöbbször elég is, s ha nem, akkor sem túl bonyolult egy többdimenziós tömböt egy kétdimenziósba elpakolni (erre is mutat példát az alábbi program).

```

10 INPUT PROMPT "TÖMBMERETEK:"; H1, H2, H3
20 NUMERIC TOMB2 (H1, (H2+1)*(H3+1)-1)
30 DEF TOMB (I, J, K)
40 LET TOMB=TOMB2 (I, J*(H3+1)+K)
50 END DEF
60 DEF TOMB2 (I, J, K, TART)
70 LET TOMB2 (I, J*(H3+1)+K)=TART
80 END DEF
90 !
100 ! PÉLDA BEVITELRE, A TOMB (2,0,3)-BA
    AKARUNK 6*78-AT VINNI:
110 CALL TOMB2 (2,0,3,6*78)
120 ! PÉLDA TÖMBELEM KIOLVASÁSÁRA, A TOMB
    (2,0,3) ÉRTÉK FELÉRE VAGYUNK KÖVETKEZŐK
130 PRINT TOMB (2,0,3) / 2
    
```

Ha valaki azonban ragaszkodik pl. a háromdimenziós tömbökhöz (szép, áttekinthető programot akar írni), annak segítségül közöljük ezt az egyszerű programot. A három indexhatárt a H1, H2, H3 változókba rakva a 20-80 sorok lehetővé teszik, hogy a TOMB nevű háromdimenziós tömböt kényelmesen használjuk – példát a használatra a 110-es és a 130-as sorokban láthatunk. A példában a tömb indexei 0-tól kezdődnek. Ezt a két eljárást bárki könnyen átalakíthatja 3-nál több dimenziós vagy string tömbökre, ill. úgy, hogy az indexek 1-től kezdődjenek. Az ilyen tömbök használata persze kicsit lassítja a programot, a TOMB egy elemének elérése pl. 2,2-szer lassabb, mint a TOMB2 megfelelő elemének az elérése.

*Király Zoltán*

### KÖR LASER BASICBEN

Az LSI kiadásában nemrégiben megjelent a SINCLAIR SPECTRUM JÁTÉK ÉS PROGRAM III c. könyv. Ebben olvasható a LASER BASIC leírása. Szerencsémre 4-5 hónapja találkoztam, és mindenféle leírás nélkül kísérletezgettem vele. Szerintem is nagyon hatékony BASIC bővítés, főleg grafikai programok írásához. Egyetlen hibáját vettem észre, pontosabban a hozzá tartozó SPRITE GENERATOR-ét. Sok esetben jó lenne,

ha lenne benne körrajzoló rutin is. Sajnos ezt viszont kifejeztették belőle, de könnyedén beírható. Mivel a program törzse BASIC, így BREAK-kel leállítható, majd az alábbi módosításokat kell elvégezni:

```

115 IF AC=94 THEN GOSUB 5300
.
.
5300 REM KÖRRAJZOLÁS
5310 LET KOZ=B*(XS+BX)+XC+CX: LET EPE=167-(B*(YS+SY)+YC+CY)
5320 INPUT "SUGAR = ";RAD
5330 IF KOZ+RAD>247 OR KOZ-RAD<128 OR EPE+RAD>167 OR EPE-RAD<48 THEN PRINT AT 21,0;"ERROR - KILOGNA A KEPBOL"
    : GO TO 5310
5340 LET INV=0: GO SUB 5400
5350 PRINT AT 21,0;"MEGFELEL? Y/N ": GO SUB BTAC
5360 IF AC=09 THEN RETURN
5370 IF AC=78 THEN LET INV=1:
5400 CIRCLE INVERSE INV; KOZ,EPE,RAD: RETURN
    
```

Tehát ennyi kis kiigazítással megoldható a körrajzolás. Először a középpontot kell a kívánt pozícióba állítani mind a spriteképernyőn, mind a munkatáblán. A rutin hívása a SYMBOL SHIFT H lenyomásával történik. Ezután kell megadni a sugár nagyságát, majd kirajzolódik a kör. Ha úgy sikerült, ahogy gondoltuk, akkor a kérdésre y-nal válaszolva térhetünk vissza a főprogramba. Ha netán nem úgy sikerült, akkor az n-nel válaszolva, az elhibázott kísérlet letörölődik és megjelenik egy újabb választási lehetőség, y lenyomásával újra próbálkozhatunk, n-nel visszatérünk a főprogramba. A másik bővítési lehetőség Spectrum tulajdonosok számára. Az eredeti program a kurzor-nyilakra nem reagál, pedig sokkal kényelmesebb a használatuk. Ezek használata a következő módosításokkal érhető el:

```

110 IF AC=8 OR AC=9 OR AC=10 OR AC=11 OR AC=53 OR AC=54 OR
    AC=55 OR AC=56 OR AC=48 OR AC=57 OR AC=34 OR AC=44 THEN
    GO TO 1000
1015 IF AC=56 OR AC=9 THEN LET XC=XC+1: GO TO 1200
1025 IF AC=54 OR AC=10 THEN LET YC=YC+1: GO TO 1200
1030 IF AC=55 OR AC=11 THEN LET YC=YC-1: GO TO 1200
1035 IF AC=56 OR AC=8 THEN LET XC=XC-1: GO TO 1200
1045 IF AC=57 OR AC=44 THEN LET ON=65: GO TO 1210
1050 IF AC=48 OR AC=34 THEN LET ON=112: GO TO 1210
    
```

*Németh Gábor*

# PROGRAMMA. IÁNLAT:

## GÉPI KÓDÚ PROGRAMOK DATA SOROKBAN - PRIMO

A program legfőbb erénye, hogy a gépi kódú program olyan BASIC-változatát készíti el, amely a legkisebb változtatás nélkül is azonnal indítható. A program ugyanis a betöltő programot is elkészíti, mégpedig úgy, hogy az az eredeti gépi kódú programmal azonos helyre tölt be, így tehát azzal egyenértékű működést biztosít. Ezen felül a program első soraként REM sort készít, ahova beírhatjuk kimentés előtt a program nevét.

### Egyéb jellemzők, szolgáltatások:

- a gépi kódú program nagysága elvileg nem korlátozott
- a készítendő BASIC-program tárban való optimális elhelyezéséről a program gondoskodik
- a gépi kódú program kezdő- és végcímének ismeretében a DATA program lefutása után a rendszerváltozók automatikusan átrásra kerülnek úgy, hogy csak a létrehozott új program válik elérhetővé, amelyet most már szalagra lehet rögzíteni.
- indításkor az aktuális gépi kódú program kezdő és végcímének decimális értékeit kell beírni, ezután kiírásra kerül a művelethez szükséges időtartam.
- ha az aktuális gépi kódú program és a DATA program részben vagy egészben azonos területre kerül, akkor a program futása még a művelet megkezdése előtt üzenet kiírással leáll. A készített BASIC-program kimentése után a rendszerváltozók alaphelyzetbe állítását a gép rövid időre történő kikapcsolásával végezzük el.

### A DATA program működése a következő:

- 10-130** nem kíván magyarázatot
- 140-160** felmérik a működő és a készítendő programok tárigényét
- 170-180, 200-210** sorok a működő és a gépi kódú programtárban elfoglalt helyzetétől függően kijelölik a DATA file kezdőcímét
- 190** leállítja a program futását, ha a felmért tárigény és a gépi kódú program között átfedés jön létre
- 220** tapasztalati képlettel számolja és kiírja a művelet idejének közelítő értékét
- 240** az "1 DELETE1" sort készíti. Erre azért van szükség, mert a DATA program lefutása után a létrehozott BASIC-program magától indul.
- 260** a REM sort készíti el
- 260-410** A DATA sorokat készítik, ezenbelül a **350**-es sor a gépi kódú program végét figyeli
- 410-510** sorok készítik el a betöltőprogramot
- 520** átírja a rendszerváltozókat és JP 1B5D-vel NEW utasítás rutinjára lép. Ez indítja el a 43A4 címen kezdődő BASIC programot, melyről a **240**-es sornál esett szó, de a rendszerváltozók beállítása miatt erre szükség van.
- 540-570** a sorszámozáshoz készített szubrutin az **580.** ill **600**-as sorszámon kezdődő szubrutinok a V és K1 változók értékétől függően állítják be "P" értéket, amely adott esetben az említett változók értékeit kettes komplement kódra konvertálja.

**Szekeres Antal** Orosháza, Huba u. 4.

## CSERÉBÉRE

**COMMODORE 16, PLUS/4** felhasználói és játékprogramokat cserélek! Márkatársak írjatok!

Egri Imre, 5561 Békésszentandrás, Pérd u. 2.

**ATARI cseretársakat keresek.**

Habányi Tamás, 8200 Veszprém, Egy u. 45/c

**C 16, PLUS/4** gépre angol, német és orosz nyelvű oktatóprogramok eladók. Játék- és felhasználói programok cseréje.

Kálmán Albert, 3300 Eger, Rákóczi út 31. III/11. Tel.-üzenet: 143-031, 330-345 (Bp.)

**Kb. 120 programmal rendelkezem TVC-n,** játék és felhasználói programokat cserélnék.

Dobrovics Zsolt, 9400 Sopron, Vörösmarty u. 7.

**PRIMO programokat cserélnék.** Megvannak: Monitor, Pascal, Assembler, Disassembler, Z80 Monitor, Graphics '85, Pallas. Játékok közül: Vil, a bányász, Dobjép, Ördögmotoros, UFO, HAMM, Párbaj, Kigyók, Keljfeljancsi, Turbo, Chess 1.

Pogonyi Zsolt, 7150 Bonyhád, Perczel kert 2/A

```

10 REM *****
20 REM *      Gépi kódú programok      *
30 REM *      DATA sorokká alakítása,  *
40 REM *      betöltő program készítés  *
50 REM *****
60 REM * 1987.03.15. * Szekeres Antal *
70 REM *****
80 CLS
90 PRINTCHR$(2)" DATA FILE : BETÖLTŐ CHR$(
10) "
      program készítése":PRINT:P
RINT:PRINT
100 REM változókknak helyfoglalás
110 D=0:P=0:G1=0:G2=0:HE=0:BA=0:SZ=0:I=0:K
=0:V=0:K1=0:SO=0:A=-1:J=0:L=0:A$="" :A$=0:
FB=0
120 INPUT"Gépi kódú program kezdőcíme : ";
G1
130 INPUT"Gépi kódú program vége : ";G2:PR
INT:PRINT:PRINT
140 HE=(G2-G1)*A+(G2-G1)/8*6+170
150 BA=PEEK(16637)+256*PEEK(16638)
160 SZ=BA+HE
170 IF G1>G2 THEN K=BA:GOTO00220
180 IF PEEK(16548)+256*PEEK(16549)>G2THEN0
0200
190 CLS:PRINT"A gépi kódú program azonos te
rületre esik a BASIC programmal!":PRINT:P
RINT"Írja át a BASIC terület kezdőcímét és
is- mételje meg a program betöltést.":PRI
NT:PRINT:PRINT:END
200 IF G1>17386+HE THEN K=17386:GOTO00220
210 K=PEEK(16637)+256*PEEK(16638)
220 PRINT"Türelem, kb. ";INT(HE*.3/60)/10:"
percig dolgozom.":PRINT:PRINT
230 V=K
240 K1=V:V=V+2:SO=1:GOSUB00580:POKEV-P,SO-
256*INT(SO/256),INT(SO/256):V=V+2:GOSUB005
80:POKEV-P,182,49,0:V=V+3:GOSUB00600:POKEK
1-P,V-256*INT(V/256),INT(V/256):SO=0
250 GOSUB00540:POKEV-P,147,32,65,32,112,11
4,111,103,114,97,109,32,110,101,118,101,32
,58,0:V=V+19:GOSUB00600:POKEK1-P,V-256*INT
(V/256),INT(V/256)
260 K1=V:V=V+2:GOSUB00580
270 SO=SO+10
280 POKEV-P,SO-256*INT(SO/256),INT(SO/256)
,136,32:V=V+4
290 FORI=1TO9
300 A=A+1:IFG1+A>32767THEN0=65536ELSE0=0
310 A$=STR$(PEEK(G1+A-Q)):L=LEN(A$)
320 FORJ=2TOL:GOSUB00580
330 POKEV-P,ASC(MID$(A$,J,1)):V=V+1
340 NEXT:GOSUB00580
350 IF G1+A=G2 THEN POKEV-P,0:V=V+1:GOSUB0
0600:POKEK1-P,V-256*INT(V/256),INT(V/256):
GOTO00410
360 IF I<8THENPOKEV-P,44:V=V+1
370 NEXT:GOSUB00580
380 POKEV-P,0:V=V+1:GOSUB00600
390 POKEK1-P,V-256*INT(V/256),INT(V/256)
400 GOTO00260
410 GOSUB00540
420 POKEV-P,71,49,213:V=V+3:GOSUB00580:A$=
STR$(G1):L=LEN(A$):FORI=2TOL:POKEV-P,ASC(M
ID$(A$,I,1)):V=V+1:GOSUB00580:NEXT:POKEV-P
,58:V=V+1:GOSUB00580
430 POKEV-P,71,50,213:V=V+3:GOSUB00580:A$=
STR$(G2):L=LEN(A$):FORI=2TOL:POKEV-P,ASC(M
ID$(A$,I,1)):V=V+1:GOSUB00580:NEXT:POKEV-P
,58,143,71,49,212,51,50,55,54,55,202,81,21
3,54,53,53,51,54,58,149,81,212,48,0
440 V=V+24:GOSUB00600:POKEK1-P,V-256*INT(V
/256),INT(V/256)
450 GOSUB00540
460 POKEV-P,129,73,213,71,49,189,71,50,58,
143,73,214,51,50,55,54,56,202,74,213,73,58
,149,74,213,73,206,54,53,53,51,54,0:V=V+33
:GOSUB00600:POKEK1-P,V-256*INT(V/256),INT(
V/256)
470 GOSUB00540
480 POKEV-P,139,65,58,177,74,44,65,0:V=V+8
:GOSUB00600:POKEK1-P,V-256*INT(V/256),INT(
V/256)
490 GOSUB00540
500 POKEV-P,135,73,58,65,213,193,40,71,49,
206,81,41,0:V=V+13:GOSUB00600:POKEK1-P,V-2
56*INT(V/256),INT(V/256)
510 GOSUB00580
520 POKEV-P,0,0:GOSUB00600:POKEK1-P,V-256*
INT(V/256),INT(V/256):POKE16548,K-256*INT(
K/256),INT(K/256):V=V+2:POKE16722,V-256*IN
T(V/256),INT(V/256):POKE16755,42,82,65,34,
249,64,195,93,27:CMD
530 END
540 K1=V:V=V+2:GOSUB00580
550 SO=SO+10
560 POKEV-P,SO-256*INT(SO/256),INT(SO/256)
:V=V+2:GOSUB00580
570 RETURN
580 IFV>32767THENP=65536ELSEP=0
590 RETURN
600 IFK1>32767THENP=65536ELSEP=0
610 RETURN

```

# P O S T A



A Primoknál, az "A" típusnál ('84 1. verzió) mint köztudomású, nincs kivézetve semmiféle csatlakozó (kivéve a szükségeseket). A billentyűzetet pedig nem éppen a játékprogramokhoz tervezték, s így előbb-utóbb szükségessé válik a botkormány.

Mondjuk ügyel-bajjal összehozok valamiféle megoldást. Akkor vajon bármilyen joystick csatlakoztatható lesz hozzá? (a hardver füzetben ez áll: „megfelelően megépített”)  
Kovács Zoltán, 1196 Bp., János F. u.147.

A teljes kiépítésű Primon a joystick részére rendelkezésre áll az egyik kimeneti regiszter egyik bitje és az egyik bemeneti regiszter két (azaz kettő) bitje. Ebből mindjárt kitűnik, hogy nemcsak szoftverre, hanem speciális hardverre (illesztőegység, "interface") is szükség van. Vagyis lenne. Az ós-Primokból ugyanis hiányzik néhány IC, a csatlakozókról nem is beszélve.

Elkeseredésre azonban nincs ok:

1. A *µMagazin*ban (1986. október) közölték a botkormány interface kapcsolási rajzát és a hozzátartozó gépi szubrutint. (31. oldal, Gerencsér Sándor)
2. *µMagazin* 1985/2. 28. oldal: dr. Simonyi Endre vállalja, hogy 50 Ft-ért bármilyen gyári botkormányt átalakít olyanná, hogy Primohoz is lehessen használni. (Gondolom, akkori 50 Ft-ért.) ...Vajon a hiányzó IC-eket is beszereli?
3. Érdeklődésünkre a Fehér Hajó utcai GELKA-ban azt mondták:
  - a) Vállalják a Primo hiányzó integrált áramköreinek beszerelését a csatlakozókkal együtt.
  - b) Készítenek joystickhoz illesztőegységet (kivánság szerint egy vagy két botkormányhoz), a számítógép dobozába beszerelik, és így Commodore joysticket lehet hozzá használni. Árakat nem mondtak, éppen most változnak...

**Darázs Tibor**, Debrecen, István út 19. Primo tulajdonos olvasónk egy sor kérdést tett föl. Igyekeztünk válaszolni.

1. *Lehetőség van-e P64 tip. gépbe a soros interface kialakításához szükséges elemek házilagos beépítésére, és egy ilyen átalakítással alkalmassá válik-e a gép a Commodore-lemezmeghajtó kezelésére?*

Igen, a Hardver-füzet 1. és 2. sz. mellékletei alapján minden további nélkül. Ismét igen.

2. *Értesülésem szerint kapható kazettán a lemezegység kezelési feladatait ellátó (CDOS című) operációs rendszer. Hol kapható és mennyiért?*

Láttuk árusítani a Keravillnál, az Elektromodul árjegyzékén a kiskereskedelmi ára 1986-ban 1103 Ft volt.

3. *Hol szerezhető be a Commodore lemezmeghajtó soros csatlakozójának el- lendarabja?*

Tudtunkkal sehol. De tudja, ilyenkor mindig azt szoktuk tanácsolni, hogy próbálkozzon a Bécsi út túlsó végén, ott sem biztos hogy sikerül.

4. *A PRIMO Szoftverleírás szerint a 00161-es címen található a képkiváltásra várakozó (VBLANK) rutin, amely a képkiváltás kezdetekor adja vissza a vezérlést a programnak. Azonban: az Y = call (161) módon meghívva, a gép alapállapotba kerül és a program elszáll. Mi az oka? A cím nem 161?*

A képkiváltásra várakozó rutin a 12449 (30A1H) címen kezdődik, bár a "Szoftverfüzet" mást állít.

BASIC-ben egyáltalán nincs értelme meghívni, gépi kódú programoknál is csak akkor, ha valami hipersuperprofi programot akarunk csinálni.

5. *A képkiváltás 7 ms-nyi ideje alatt hogyan lehet gyorsan megváltoztatni a képernyő tartalmát?*

A képernyőhöz tartozó memóriaterület minden Primonál az utolsó 6K. (A kimeneti regiszter módosításával ezt 8K-val alacsonyabb címre lehet átkapcsolni.) A képkiváltás ideje alatt a gyors gépi kódú rutinokkal erre a területre tetszőleges adatokat lehet írni (betölteni).

Ha a képernyőmódosítást megfelelően szervezzük meg, a VBLANK rutint – tapasztalatom szerint – főleges alkalmazni, a kép gyors módosítását a szem úgy érzékeli, mintha az egyszerre történne – éppen úgy, mint egy filmen.

Példa a célszerű szervezésre: a számításokat és a képmódosításokat a RAM-nak egy másik területén végeztetjük el és a kész képet egy ciklussal (sőt! egy utasítással [!]) átvisszük a képernyőmemóriába.

Egy másik jó módszer az, ha OUT utasítással kapcsolatjuk a két képernyőt, és mindig éppen arra a képernyőre rajzolunk, amelyik nem látszik. Ez utóbbi esetben jól megvagyunk mindenféle gépi szubrutinok nélkül is.

*Kérem továbbítsák kérdéseimet a Videoton illetékeseknek; és közöljék, hogy a TV-Computer leírásában miért "titkolták el":*

– a "BASIC corrupted hibaüzenetet, – az IO függvényt (argumentuma 0 és 255 közt kell legyen, értéke általában nulla, 87 és 106 közötti értékre viszont különböző számokat ad ismeretlen "logika" szerint), (tokenje sincs)

– azt a tényt, hogy a DEF F(X) ... utasítással definiált függvények argumentuma nem kezdődhet "-" jellel pl.: 10DEF FN(X) = X

RUN  
PRINT FN(2)

2  
PRINT FN(-1)  
\*\*\*Not understood  
PRINT FN (0-1)

–1  
Lőrinczy Zsigmond, 1025 Bp., Cimbalom u. 1.  
Kérdéseire nem tudunk válaszolni. Tartunk tőle, hogy az illetékesek sem tudnának. De örülünk az információknak.

**Szilágyi Balázs**, 1125 Bp., Nógrádi u. 2/B olvasónk több kérdéssel fordult hozzánk, amelyekkel megkerestük Gergely Jánost a Videoton Vevőszolgálatán, akitől a következő felvilágosítást kaptuk.

1. *Létezik-e még Intelligent Software nevű cég? Ha van, mi a címe?*

A cég létezik, de senki (a céggel szoros kapcsolatban álló VIDEOTON-t is beleértve) nem hajlandó címüket kiadni. (Magyarzatot sem kaptunk.)

2. *Milyen jogok védettek a VIDEOTON, az Enterprise GMBH vagy az IS Enterprise, vagy TV-Computer*

– ROM listájának közreadása (saját megjegyzéseimmel)

– kapcsolási rajz publikálása  
– egyéb szoftver-, ill. hardverismeretek nyomtatása

A TVC-ről tudunk felvilágosítást szerezni:

– ROM listát a VIDEOTON nem publikálhatja, de annak nincs akadálya, hogy más, a saját megjegyzéseivel ezt megtegye.

– Kapcsolási rajz publikálása folyamatban; van a Novotrade-ben készül egy 3 darabból álló sorozat:

- Operációs rendszer ismertetése
- Hardverismertetés (kapcs. rajzzal)
- BASIC példatár

Időpontot nem tudtak mondani, de minden könyvesboltban kapható lesz.

3. *Ha van angol és német billentyűzettel ellátott Enterprise, akkor miért nem csinálnak olyat, amit a TV-Computerhez hasonló dobozba tesznek? (Úgy jobban vonzana.)*

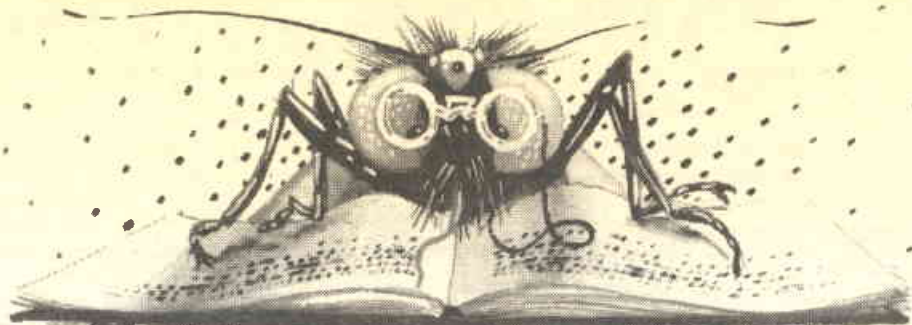
Nem világos, miért kellene az Enterprise-t TVC dobozba tenni?

4. *Miért nem olvashatók a BIT-LET-ben statisztikák a hazai mikroszámítógéptulajdonosokról? Hányan vannak? Melyik gép a legnépszerűbb?*

Mert erre csak becslések vannak, durván 100%-os hibával, és így az adatok nem sokat mondanak. (Azt meg úgys mindenki tudja, hogy C 64-ből van a legtöbb).

5. *A TVC drive leírása 250 példányban készült. Ezután érdekel, hogy vajon hányan vannak az Enterprise és a TV-Computer tulajdonosok.*

A TVC drive leírás nem 250, hanem 2500 példányban készült legalább. Minden drive-hoz jár. Így azt hisszük, hogy elegendő.



# K Ö N Y V M O L Y

**Peter Norton: Fedezzük fel az IBM PC-t!** – Műszaki Könyvkiadó, 349 o., 110 Ft. (Az IBM-könyvek és -programok neves szerzője e kötetében saját, közvetlen tapasztalatai alapján írja le a gép megismerésének folyamatát.)

**Theisz György: BASIC-tanácsadó C-16, Plus/4**

(A didaktikusan felépített kötet a gépek iskolai felhasználóinak – tanároknak és diákoknak – nyújt segítséget.)

**Csepai János-Quittner Pál: BIT-LES – Számítástechnikai lemezlovas – SZÁMALK, 223. o., 125. Ft.**

Ha lapunk rendszeres olvasói egy könyvesboltban találkozunk ezzel a kötettel, bizonyára felmerül bennük a kérdés: „Mi ez? Talán valami újfajta SZUPER BIT-LET?”

Már előljáróban meg kell jegyeznünk, hogy a könyvnek és a BIT-LET-nek semmi köze nincs egymáshoz. A szerzők és a kötetet kiadó SZÁMALK arra hivatkoznak, hogy a címet a Beatles együttes nevéből asszociálva ötlötték ki. Erre utalna a „Számítástechnikai lemezlovas” alcím, a számítógépen zenélő fiatalokat ábrázoló címkép, valamint a belső címlap a Sárga tengeralattjáró c. film figurájával. Szerintük így a cím ötlete mindenki számára nyilvánvaló – de erre konkrét utalást a kötetnek szinte a végén, a 201. oldalon tesznek csupán. Ahogy elmondták, csak két héttel a könyvcím kitalálása után jutott eszükbe, hogy igencsak nagy az egybeesés lapunk címével.

Már csak a fentiek miatt is érdemes megnéznünk, hogy milyen színvonalú a könyv, hogy mennyiben szól lapunk olvasóközönségéhez, egyszóval válalható-e számunkra a cím hasonlósága. A szerzők a bevezetésben 12 és 99 év között bárkinek ajánlják kötetüket. Ahogy az előszóból („A szerzők mentegetőzése”) kiderül, a könyvet azért osztották 69, néhány oldalas fejezetre, és azért választották a könnyed hangvételt, hogy bárki kedve szerint lapozgathasson benne, illetve bármikor könnyen megtalálja az őt érdekítő témakör rövid ismertetését.

Ennek a szerkesztési módszernek azonban két nagy hátránya is van. Most mindkettőre csak egy-egy példát említünk – de hasonlóak, sajnos, hemzsegenek a kötetben.

Az egyik az, hogy a 69 kis fejezet mindegyike a számítástechnika egy-egy önálló témakörét igyekszik bemutatni a rendelkezésre álló két-három oldalon – ami szinte megoldhatatlan feladat. A szerzők sem képesek erre, így azután érvényesül a „sokat markol, keveset fog” elve: szinte mindenről szólnak, de nem mondanak szinte semmit. (Sajnos, igaz, amit a szerzők a 153. oldalon önironikusan megjegyeznek: „... önmagában attól nem válnak valakik íróvá, számítástechni-

fogalmát, a második a hardverrel, a harmadik a szoftverrel, a negyedik a programozással, az ötödik pedig a számítástechnika alkalmazásával foglalkozik. A gond itt az – és ez a második, amit a szerkesztési koncepció kapcsán meg kell említenünk –, hogy az öt rész között rengeteg az átfedés, illetve, hogy az egymáshoz kapcsolódó témakörök nem követik logikus rendben egymást. Az átfedésekre ugyan a szerzők már az Előszóban felhívják a figyelmet. A fejezetek sorrendjében azonban már elhibázottnak tűnik az, hogy például a számítógépes rendszerek biztonságáról, az adatvédelemről szólnak a 39. és a 40. fejezetben – ahol sci-fi filmből és a valóságból vett eseteket mutatnak be az illegális adatfelhasználásra, és az adatvédelmi eljárásokat ismertetik –, valamint a 63. és 65. fejezetben, ahol a rejtjelezéssel, kódolással, és a katasztrófák elhárításával foglalkoznak. Sokkal közérthetőbb és használhatóbb lett volna, ha ez a négy fejezet egymás mellé kerül.

E kifogások mellett még azt is meg kell említenünk, hogy az öt fő rész színvonal, illetve ismereteik szintje sem egyenletes. Az első a laikusoknak szól. A második is így indul, de itt már sok az olyan fogalom, amit az amatőrök meg sem értenek. A harmadik rész szinte csak olyan fogalmakkal foglalkozik, amivel az otthoni felhasználó nem is találkozik – a szakmabeli számára viszont túl leegyszerűsítőek a magyarázatok. A negyedik és ötödik rész ismét csak a kezdőkhöz szólna – de rendszertelenül kapkod a témakörök között, szinte esetlegesen változtatva.

A könyv utolsó előtti – 68. – fejezete terjedelmesebb, mint a többi: „Nagy nevek, kis történelem” címmel a számítástechnika kiemelkedő egyéniségeinek munkásságáról olvashatunk 11 oldalon. Ez tulajdonképpen már a független részének tekinthető, hiszen a 69. fejezet rövid epilógusa után mellékletként egy tárgymutatót, valamint egy számítástechnikai kiszótárt is találunk a kötetben.

Külön kell még szólnunk a kötet grafikáiról, melyeket Halász Géza készített. A karikatúrákat – ahogy az 54.



kusokká, vagy egyidejűleg mindkettővé, mert megírtak egy számítástechnikával foglalkozó könyvet.”) Így a mindössze másfél oldalas 11. fejezet úgy indul, mintha meg akarná magyarázni a bináris számok fogalmát és használatuk módját, azután egy váratlan fordulattal a Boole-algebra bemutatására tér rá, végül egy furcsa átvezetés után a gépi kód félooldal ismertetésével zárul.

Van azért persze öt nagyobb szerkezeti egysége is a könyvnek: az első általánosságban igyekszik ismertetni a számítógép és a számítástechnika

fejezetben ő maga írja – az Art Studio Spectrumos változatával rajzolta. Nem lehetett kis munka, hiszen a kötetben három–négy oldalanként találunk illusztrációt. Időnként zavaró, hogy a rajzokban szinte minden zárt terület fel van töltve valamilyen rasztermintával – de talán a grafika a kötet legszínvonalasabb része.

Hogy válaszoljunk a kritika elején föl-tett kérdésre: nem örülünk, hogy a könyv ezzel a címmel látott napvilágot!

**Tallér József**

**A. Hettinger–A. Heinz: ATARI-BASIC kezdőknek – Műszaki Könyvkiadó, 130 o., 60 Ft.**

Nagy örömmel vettem észre a könyvesbolt polcán A. Hettinger–A. Heinz: ATARI-BASIC kezdőknek c. könyvét. Végre! Úgy tűnt, már annyira „leírták” az ATARI 400-as és 800-as típusait, hogy semmilyen irodalmat nem adnak ki róla, hozzá. Amikor azonban kézbe vettem a könyvet, kicsit meglepődtem: egy kezdőknek szóló könyvet vastagabbnak képzeltem. A tartalomjegyzéket szemlélve már erősen csodálkoztam: az ATARI-BASIC-et egyetlen fejezet (24 oldal) ismerteti. Ugyanakkor (ez előnyös) a grafikát és a hanggenerátor használatát külön fejezetben tárgyalja. Sajnos erős a gyanúm, hogy egy kezdő, aki ebből a könyvből akarja a programozást megtanulni, nehezen boldogul.

Mindenekelőtt túl rövidnek és felületesnek érzem a magyarázatokat. A program lényegét például így foglalják össze a szerzők: „Nem jó, ha a számítógép minden parancsot azonnal teljesít” (13. o.) Persze értem, de biztos, hogy ez egy kezdő számára is világos? A magyar fordítás lektora, Fried Katalin ugyan sok hasznos kiegészítő lábjegyzettel próbálta a könyv hiányosságait pótolni, ám azért a kiegészítések mégsem vehetik át a könyv szerepét. Pl. a lábjegyzetből kissé bővebben értesülünk a program mibenlétéről, itt van leírva először a sorok törlése (14. o.), kizárólag itt kapunk képet a változók fogalmáról (12. o.), a szerzők ugyanis rögtön a tömbök ismertetésével kezdik. Nem szerepel azonban – még lábjegyzetben sem – a változó-név helyesírása (betűvel kezdődő betűszám kombináció) stb.

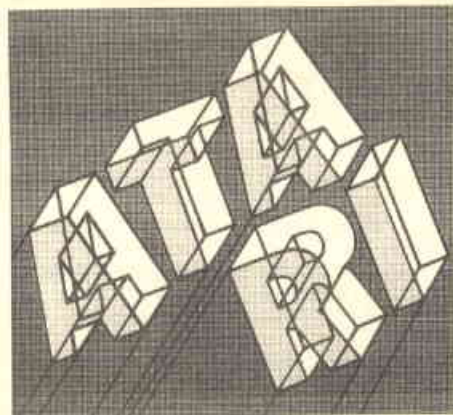
Elvi tévedések is előfordulnak. Csak néhány példa:

„A sorok számozása egytől ötjegyű számokig terjedhet...” – 32767-nél nagyobb sorszám nem írható!

... a program futását a BREAK vagy SYSTEM-RESET billentyű lenyomásával lehet megszakítani. A megszakított programokat a CONT parancsral lehet folytatni.” – SYSTEM-RESET után a CONT nem működik (pontosabban a program **második** sorára ugrik)! (29. o.)

„Írjori egy programot, amely a következő színregisztereket állítja be: Keret: zöld, Háttér: vörös, Szöveg: fekete” – 0. grafikai módban (melyről itt szó

A. HETTINGER–A. HEINZ



van) a háttér és a szöveg csak azonos színű lehet! (38. o.)

A 128. oldal utolsó bekezdése teljes tévedés. A GET, ill. PUT egy byte-ot mozgat, az INPUT sem ötöt, hanem annyit, ahány karakter hosszú az adat. A gép egyébként a számokat nem öt, hanem hat byte-on tárolja.

A helyzetet bonyolították a didaktikai hibák, elírások, rendszertelenségek. Például: a PRINT ugyan külön alcím-ként szerepel, a PRINT-vevessző és -pontosvevessző használata azonban a STEP ill. az INPUT ismertetésekor van leírva – ki tudja, miért?

Az „A PRINT-parancs speciális alkalmazásai” címszó példái cseppet sem

speciálisak az addigiakhoz képest (pl. „HALLO”) 12. o.

A 38. oldal felső táblázata (3.1. tábl. Grafikus módok színregiszterei), az előtér (szöveg) színregisztere’ teljesen összevissza és hibás.

A 44. oldal ábrája hibás és zavaró.

A 71. oldal az 5. fejezet kezdete: BASIC-parancsok. Az első négy „parancs”: ABS, ADR, ASC, ATN. Ezek függvények! Parancsként beírva őket, hibajelzést kapunk. Ezenkívül az ADR függvény (a szöveggel ellentétben) csak alfanumerikus változóval használható. A BYE parancs leírásánál nem értem, mi történik automatikusan. 84. oldal: „A hang elhallgattatásának felélesztése”. Tessék?

Végül bizonyos dolgokat meg sem említ a könyv: az Editálásról, vagyis sorok beírásáról, javításáról, a vezérlő-billentyűk szerepéről; a relációs jelekről szó sem esik.

Sokáig folytathatnám a hibák felsorolását.

El kell ismernem, hogy a magyar fordítás lektori megjegyzései sokat javítanak és segítenek. Nem volt szerencsés éppen ezt a könyvet lefordítani. Ráadásul a magyar kiadáshoz fűzött Kiegészítésben is vannak hibák.

Összefoglalva az a benyomásom, hogy ez a könyv egy rosszul megválasztott olvasókönyv készült. Kezdőknek túl rövid és érthetetlen egy ilyen „gyors-talpaló” tanfolyam. A haladóbbak viszont nem fognak bepötyögni olyan példaprogramokat, amelyeknek fele végrehajtásra sem kerül. Legvalószínűbb, hogy egyszerűen átlapozzák az egészet és a játékprogramokat fogják beírni a gépbe.

Hadd idézzek egy (nem tőlem származó) kritikát: „Nagy előnye, hogy magyarul van”! Hát, igaz...

**Rieth József**



# lapasztalat

N Y E R O

## ÉRTÉKELÉS

A pályázatra (bűvös kocka) 20 megoldás érkezett, melyek közül hatban a beküldött programrészlet durva hibákat tartalmaz. A többi 14 megoldás lényegében jó, van néhány nagyon ötletes elgondolás, néhányan viszont jó ötletüket nagyon csúnya programmal valósították meg. Ezért most is súlyozott sorsolást tartottunk(!) a BIT-LET Karácsonyon. Mivel erre lapzárta után került sor, a nyertesek nevét még nem tudjuk. **A sorsolás rendszere a következő volt:**

**Az 1. díjat (3000 Ft-os Centrum-utalvány)**

**Bolgár László, Bp; Lévai Gábor, Albertirsa és Lipi Gábor, Bp. között,**

**A 2. díjat (1000 Ft-os Centrum-utalvány)**

**Blaschek József, Bp.; Kruzlicz Ferenc, Tótkomlós; Szalkai István, Devecser és Szirmai Ákos, Bp. között sorsoltuk ki.**

**A 3-7. díjak, egyenként 1-1 db 200 Ft-os Centrum-utalvány) sorsolásán a fentiekben kívül részt vehettek még:**

**Csillag Péter, Bp.; Csizmadia Tamás, Bp.; Hernyik András, Szentés; Kajári Gábor, Cegléd; Nagy Zoltán, Székesfehérvár; Peták Tamás, Szolnok; és Tőkés Mihály, Dunakeszi.**

A feladat az volt, hogy olyan struktúrát találjunk ki a bűvös kocka állapotainak tárolásához, mely egy esetleges visszaforgató-programhoz a lehető legjobban illeszkedik; s meg kellett írni az egy oldal elforgatását megvalósító rutint.

### Néhány észrevétel:

– mivel általában legfeljebb 72 elemszámú tömbökről van szó, ezért nem nagyon érdemes a memóriával takarékoskodni.

– egy kocka-visszaforgató algoritmusban úgyis elég sokat kell számolni, így egy forgatás megvalósítását nem érdemes a lehető leggyorsabbra készíteni, persze azért túl lassúra sem.

– egy oldal forgatásánál (bármilyen tárolásnál) 4-es ciklikus cseréket kell végezni, ezt nem érdemes FOR ciklussal csinálni. Főleg nem érdemes erre egy másik tömböt használni, s onnan visszamásolgatni. Sokkal egyszerűbb (és mellékesen gyorsabb is) a 4-es cserét az 5 értékadás leírásával megoldani.

– mivel minden visszaforgató algoritmusban nagy szerepe van a szemközi oldalpárokknak, így a szemközi oldalak kódjai között legyen egyszerű összefüggés (pl. dobókockaelv).

### A megoldások 2 fő csoportba oszthatók.

Volt, aki a kocka térbeli helyzetét tárolta (tehát az egyes kis kockák helyét és helyzetét), s volt, aki a lapokon látható négyzeteket. Egy visszaforgató algoritmus számára az első módszer tűnik jobbnak, azonban itt elég nehéz jól megoldani a kis kockák helyzetének táro-

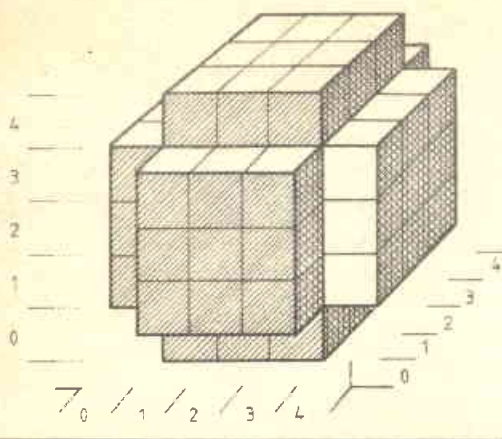
lását. Erre több ügyes megoldás érkezett, a legizgalmasabbat alább közöljük.

**A kis négyzetek tárolásánál az a gond,** hogy egy forgatás a 4 szomszédos oldal 3-3, a forgatandó lap felé eső kis négyzetét is mozgatja. Ezeket legtöbbször nagy segéd-tömbökkel és sok DATA-val határozzák meg, néhányan azonban ügyesebbek voltak. A megoldók e csoportjában Szirmai Ákos volt a legleleményesebb. Ő a kocka lapjainak kis négyzeteit úgy számozta be, hogy azok a kocka egy-egy lapján csavarvonalat alkotnak. Sikerült elérnie azt is, hogy a kocka bármely oldalát forgatva, az adatok hasonlóképpen rendeződnek minden lapon. Azaz ha a kocka bármelyik lapját fordítjuk el, a további négy kapcsolódó oldalon a mozgandó kis négyzetek sorszámai előre meghatározottak, a kiválasztott (forgatandó) oldal sorszámtól függetlenül. Mindig a szükséges 3 db 4-es ciklikus cserét hajtja végre a gép, azaz megfelelő helyekre az új színek kerülnek. Ekkor csak az egyes lapok 4 szomszédjának kódját kell megfelelő sorrendben DATA-ba írni. Egy másik igen jónak látszó ötlet Bolgár Lászlótól származik, aki az egyes helyeken nemcsak az illető négyzet színét tárolja, hanem annak a helynek a sorszámt is, ahová rendezés után az illető lapon kerül.

**A legötletesebb megoldás Lipi Gábor és Nyilas István László közös műve,** levelükből és programjukból a lényegyet kiragadtuk, s alábbiakban idézzük. Programjukból csak a forgatást végző rutint emeltük ki, s az egyszerűbb nyomtatás kedvéért „átcommodoreplusnégyesítettük”, (az eredeti TVC-re szőlt), bár ezzel nem sokat változtattunk rajta.

A Bűvös kocka állapotának tárolására egy 5\*5\*5-os tömböt használtunk. Azért esett erre a választásunk, mert így egyszerre lehet nyilvántartani a 27 kis kocka egymáshoz viszonyított helyzetét és az 54 színes négyzet laponkénti helyét. Az előbbi segítséget nyújthat egy olyan rendezőprogramnak, amely a felcserélődött kockákat viszi a helyükre; az utóbbi pedig a már helyén lévő kockák megfelelő irányba fordításához nyújt segítséget. Így egyben a kocka megrajzolását is könnyű végrehajtani és egy későbbi mintázattel- ismerő programrészt is nehézség nélkül kezelheti az állományt.

**És végül zárszóként: a Nyerő oldal az az búcsúzik olvasóitól, pályázóitól, hogy rémélhetőleg az új BIT-LET-ben is lesznek pályázatok, s lesznek kisorsolható gépek.**



Az ábrán az általunk használt háromdimenziós tömb térbeli szerkezete látható: A be nem jelölt részeket nem használjuk. Ezen az ábrán az oldalaknak megfelelő rekeszeket láthatjuk, ezekben tároljuk a különböző színeknek megfelelő kódokat 0-5-ig. Az ábrán látható testnek a belseje nem üres. Itt foglal helyet az a 3\*3\*3-as rész, amiben a kis kockák sorszámt tároljuk 0-26-ig. A forgatást úgy hajtjuk végre, mint a valóságban. A program forgatás szubrutinja a megfelelő laphoz tartozó két réteget a tömbben 90°-kal elforgatja.

```

10 DIM K(4,4,4)
50 INPUT "FORGATANDÓ LAP SORSZÁMA":L
60 INPUT "FORGATÁS MERTEKE (1-3)":M
70 IF L AND 1 THEN M=4-M
100 FOR MB=1 TO M:GOSUB 1000:NEXTMB:GOTO 50
1000 S=C1 AND L)*3-OR INT(L/2)+1 GOTO 1030,1040,1050
1030 FOR SK=S TO S+1:FOR I=(SK=0)-(SK=4) TO 1:FOR J=1 TO S-I
1032 Z=K(SK,I,J):K(SK,I,J)=K(SK,4-J,I):K(SK,4-J,I)=K(SK,4-I,4-J)
1034 K(SK,4-I,4-J)=K(SK,J,4-I):K(SK,J,4-I)=K(SK,I,4-J)
1040 FOR SK=S TO S+1:FOR I=(SK=0)-(SK=4) TO 1:FOR J=1 TO S-I
1042 Z=K(I,J,SK):K(I,J,SK)=K(4-J,I,SK):K(4-J,I,SK)=K(4-I,4-J,SK)
1044 K(4-I,4-J,SK)=K(I,4-I,SK):K(I,4-I,SK)=Z:NEXT J,I,SK:RETURN
1050 FOR SK=S TO S+1:FOR I=(SK=0)-(SK=4) TO 1:FOR J=1 TO S-I
1052 Z=K(I,SK,J):K(I,SK,J)=K(J,SK,4-I):K(J,SK,4-I)=K(4-I,SK,4-J)
1054 K(4-I,SK,4-J)=K(4-J,SK,I):K(4-J,SK,I)=Z:NEXT J,I,SK:RETURN
    
```