

mikrovilág MAGAZIN

1992. szeptember 8.

8. évfolyam 8. szám

Ára: 96 Ft

+32 oldal!

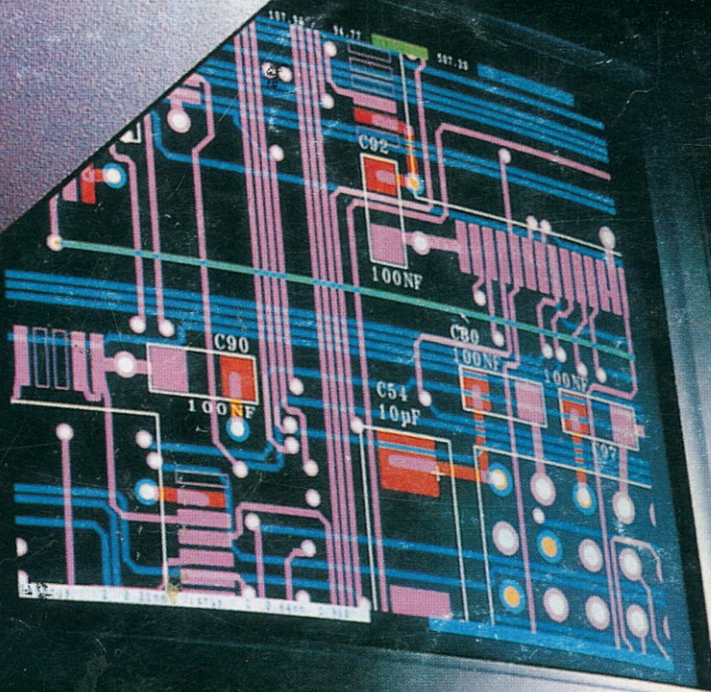
FLOPPY-ROSTRA

**Most a
legolcsóbb ...!**

**PC-SULI
Programozzuk
Pascalban!**

Játék-monitor

**C++ – avagy
a szükséges
plusz!**



**100 + 200 gramm program
IBM PC-re, C – 64 re,
Plus/4-re**

*Egy, két, hár... és!
Mány az AMIGA VILÁG!*

Már harmadszor *Bécsben, a Messepalastban* az



Az AMIGA Világegyetem itt a szomszédban:

akciók,

workshopok,

titkok,

csodák!

AMIGA WORLD:

a baráti PC-k és a PC-barátok világa!

Az irodai és szabadidős célú rendszerorientált PC-vásár, ahol az egész család jól szórakozhat!

Bemutatók,

árusítás,

óriási árkedvezmények:

most megveheti, megújíthatja, továbbfejlesztheti kedvenc AMIGA-ját!

Az ön családját is várja

szeptember 24-27-ig

az AMIGA család.

Szervező: ECI Expoconsult International Ges.a.b.H., Bécs

Es még valami:

Ne álljon sorba a bejáratnál: vegye meg belépőjét elővételben, forintért, a magyarországi képviselőnél (a napijegy 745 forint, gyermekeknek 14 éves korig 450 forint):

presentex

Vásárképviseleti Kft. Messevertretungs Gmbh. Fair Representation Ltd.

**Vásárképviseleti Kft.
Budapest, Vásárváros B pav. I.em.
Telefon: 157-4280, 178-0352**

Levél az olvasóhoz

Hát itt a 96 oldalas, amit júliusban ígértünk: másfélszeres terjedelem változatlan áron, előfizetőinknek a szokásos kedvezményrel, külön megrendelhető floppy-melléklettel, a megszokott hardver- és szoftverismertetővel, vásárlási tanácsokkal.

A programgyűjtemény hasonlít a tavaly szeptemberben megjelent **100 gramm program** című kiadványunkhoz, amely nagy sikert aratott, újra bebizonyítva, hogy a kezdőknek pótolhatatlan segítséget jelent, ha „beleláthatnak” egy kész, jól működő programba, de a haladók között is rengetegen vannak, akik szívesen pötyögnek be rövidebb programlistákat. Most 46 oldalnyi Plus/4-re, C-64-re és PC-re készült programot közlünk: van közöttük játék- és oktatóprogram, grafikai, zenei demo és jó néhány komolyabb felhasználói miniszoftver. A könnyebb felhasználóság érdekében a programokat az újság közepén, külön borítóval ellátott „kivehető ajtó”, mellékletként helyeztük el, és itt kapott helyet a két programozástechnikai összeállítás is.

A szoftvergyűjtemény most is megvásárolható lemezen, sőt az újságban két megrendelőlap is szerepel: az egyikkel a 46 oldal mellékletet lehet megrendelni C-64-re, Plus/4-re

vagy PC-re (ráadásként azokat a hosszabb programokat kínáljuk, amelyek terjedelmük miatt nem fértek a lapba, így csak a tartalomjegyzékben szerepelnek), a másikkal pedig a tavalyi, 100 gramm program című kiadványunk programlemezére lehet szert tenni (C-64 és Plus/4). Mindkét gyűjtemény kizárólag 5,25"-es lemezen (kazettán nem!) és csakis a megrendelőlap visszaküldésével, utánvétellel kapható. A programlemez ár 200 forint + postaköltség.

Előfizetőink újabb kedvezményt kapnak: ingyen küldjük el címükre a programlemezt, így részben kártalanítva őket a kimaradt lapszámokért. Csupán annyit kérünk, hogy 81. oldalon található **megrendelőlap**on tüntessék fel, mióta előfizetői a Mikrovilág Magazinnak, és ne felejtsek el megjelölni, milyen gépen (IBM PC vagy Commodore) kívánják futtatni a programokat.

Terveink szerint legközelebb decemberben, a karácsonyi ünnepek előtt jelentkezünk hasonló terjedelmű, erősen kibővített programfüzettel. Előfizetőinket pedig természetesen kártalanítani fogjuk a kieső lapszámokért, illetve más IDG kiadványok kedvezményes előfizetését ajánljuk fel kárpótlásként.

8. évfolyam 8. szám 1992. szeptember 8.

Monitor		
	Sok kicsi – sokba kerül None-bookok, ha összejönnek	4
Játék-monitor		
	Kézi készülékek a Game Boyhoz	6
	Tiny Toon	6
	Egy sötét kacsa	7
	Tini Nindzsa Teknőcök 4.	7
Piac		
	Hogyan vásároljunk számítógépet? Most a legolcsóbb!	8
Szoftver		
	Basic-bővítések 11.	12
Interjú		
	COCOM – korlátok nélkül	14
200 GRAMM PROGRAM		
	Plus/4-, C-64-, PC-programok	15
	Kezdőlépések kezdő programozóknak	17
	IBM PC-programok	57
PC-suli		
	Programozzunk Pascalban!	41
	C++ – avagy a szükséges plusz	49
	Kezdődjön a C-CÓ!	53



Nemzetközi informatikai magazin

Szerkeszti:

Bognár Ákos (-bá)

Guttray László (-ray)

Művészeti szerkesztő:

Kalocsainé Doór Vilma

Szerkesztőségi titkár:

Martek Istvánné

Fotó:

Matz Károly

Kiadja:

az IDG Magyarországi Lapkiadó Kft.

Felelős kiadó:

Bíró István ügyvezető igazgató

Művészeti vezető:

Lévai András

Műszaki vezető:

Mészáros Tibor

A szerkesztőség és a kiadó címe, és a hirdetések gondozása:

Budapest I., Krisztina krt. 99.

Levél cím: 1536 Budapest, Pf. 386

Telefon: 156-9122

Telefax: 202-5565

HU ISSN 1215-8968

Formakészítés:

IDG Magyarországi Lapkiadó Kft.

Nyomja: Ságvári Nyomda

Budapest XIII., Váci út 73.

920552

Felelős vezető:

Szilágyi Tamás igazgató

Előfizethető:

(fél, illetve egész évre) közvetlenül a kiadónál, továbbá bármely hírlapkézbesítő postahivatalnál, a hírlapkézbesítőknél, a hírlapüzletekben és a Hírlapelőfizetési és Lapellátási Irodánál (HELIR, 1900 Budapest XIII., Lehel u. 10/a) közvetlenül vagy átutalással a HELIR 215-96162 pénzforgalmi jelzőszámra.

Lapszámonkénti ára: 96 Ft

Előfizetési díj egy évre: 960 Ft,

fél évre: 480 Ft

A Mikrovilág az amerikai központú IDG (International Data Group) Communications cégnek, a világ legnagyobb számítástechnikai kiadójának egyik folyóirata. Az IDG Communications közel százötvenöt számítástechnikai kiadványt jelent meg a világ több mint ötven országában. A kiadó sajtótermékeit körülbelül harmincmillióan olvassák. Az IDG Communications tagvállalatai valamennyien hozzájárulnak az IDG nemzetközi hírszolgáltatáshoz, amely online módon, naponta szolgáltatja a nemzetközi számítástechnikai híreket. A hálózathoz átvett híreket IDG-vel jelöljük.



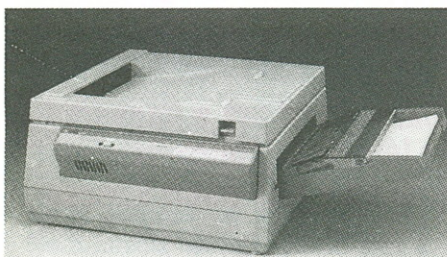


Floppyposta

Megrendelőlapok **81**

Hardver

Munkaállomások
Grafika a Silicon-völgyből **83**



Alkalmazás

Minisztériumi szolgáltatások
Adatbázis – megújított információkkal **84**

On-LINE és CD-ROM fejlesztések
Új technológiák békés harca **86**

Atari-piac

Fontos és forintos árak **88**

Teszt

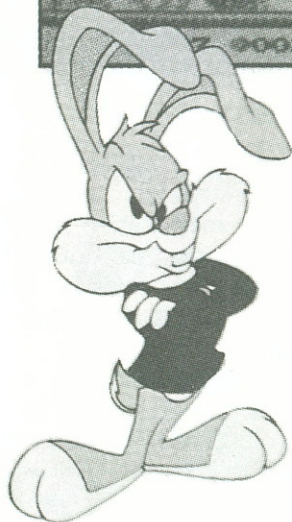
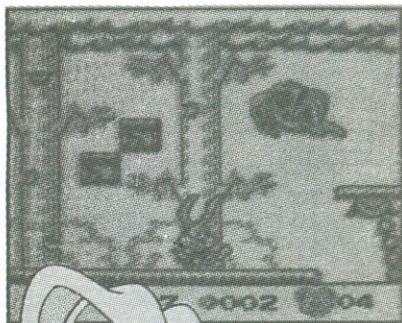
ÉKSzer kontra WordPerfect
Jó a szöveg? **90**

História

A számítástechnika története 7.
A Szilícium-völgy **93**

Postabontó

A PC-akció reakciója **95**



Sok kicsi – sokba kerül

Notebookok, ha összejönnek

Állítólag a század eleji cirkuszos világában volt egy illuzionista, aki képes volt kitalálni, milyen tárgyat rejtettek el előle. A módszer igencsak egyszerű volt: a „transzba ejtett” médium zsebre vágott például egy noteszt, majd feltette a következő kérdést: „No tessék kitalálni, mi van a zsebemben?”

Ennyi bevezető után tessék csak kitalálni, mit tartanak a jövő gépének? Igen, a notebookot, amelynek eredeti jelentése: notesz, jegyzetfüzet. Csakhogy amióta a számítergyártók a szupermini korszakot is célnak jelölték, azóta a szótárba bekerült egy új értelmezés, amire a magyar nyelvben a noteszgépet használjuk.

Ma már (szinte) mindent tud, amit egy jó képességű asztali PC-től várhatunk, legnagyobb korlátja a felhasználó, akinek ujjai többnyire az írógépek klaviatúráját kezelik kényelmesen.

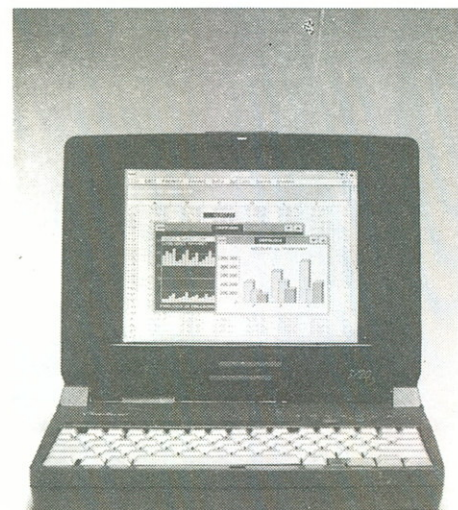
Az összeállításunkban szereplő noteszgépek talán sejtetik, hol tart ma a hagyományos irányú fejlesztés – de közben ne felejtsük el: ma már léteznek „touch-screen”, azaz képernyőérintéssel vezérelhető számítógépek, és terjednek a toll-papír kapcsolatot utánozó komputerek is.

A COMPAQ paklijából...

No tessék csak megnézni, ez a pakligép nem hordozható ugyan zsebben, de a közeljövőben nem is lesz senkinek egy paklira való, azaz 32 darab belőle, talán még a kereskedőcégek is nehezen hoznák össze a „kismaccsot”. Igaz, ami itt látható, az maga a pakli ásza, a vadonatúj csoda, ugyanis színes, aktív mátrixos (!) így aztán megjelenítőjén olyan tiszta, színgazdag képet láthatunk, mintha a kijelző helyére valami gyönyörűen hibátlan képesűség egyik lapját csúsztattunk volna. A Compaq LTE Lite/25c kijelzője

a 4096-os színkészletből egyidejűleg 256 színt mutat 640 × 480 ponton, ami a VGA felbontásnak felel meg.

Főbb jellemzői bármelyik nagy tudású asztali géppel felveszik a versenyt, a 4 MB-os alaplap 20 megára bővíthető, 84 vagy 120 MB-os winchesterére „odakészítették” az MS-DOS 5.0 operációs rendszert és a Windows 3.1-est is. Az új szolgáltatások közül leghasznosabb talán az, amelynek eredményeként grafikuson jelzi a gép, mennyi ideig használhatjuk akkumulátorról a kis színtest. Vadonatúj kialakítású az az egérpótló trackball, amelyet a kijelző



mellé helyeztek, és eltűntek a túlsúlyos billentyűzet miatti kényeszerből használhatatlan „soros” kurzorgombok, ugyanúgy fordított T alakban találjuk a négy nyilacsok, mint az asztali nagytestvéreken.

A másik Compaq-újdonság a Contura, amelyben 20, illetve 25 megahertzes 386SL dobog, operatív memóriája 4–12 MB-ig növelhető, merevlemezein pedig 40–120 MB-nyi információ tárolható.

Commodore, a megújuló

Amikor a hazai Commodore-őrület kiteljesedett, senki sem gondolta volna, hogy a PC-s korszakban ugyanez a cég képes lesz létezését tartani. Végül is a C 386 SX-LT-ben semmi világraszóló újdonság nincs, az viszont már a korábbi időszakban is érezhető volt, hogy Commodore-ék egyáltalán nem kívánnak a C-64-es babérjain csücsülni (ezt bizonyítja az Amiga-sorozat sikere is), hanem képesek követni a fejlődést. A 80386 SX processzorral, 16 MHz-es órajellel, 20 MB-os winchesterrel, 3,5 inches lemezmeghajtóval megépített notebookjuk VGA felbontású mono kijelzővel kerül forgalomba, alapszoftverei



Monokróm kijelzőjén alaphelyzetben (640 × 480 pont) 16 szürkeárnyalat látszik, 320 × 200-as felbontás esetén pedig 64 árnyalatot különböztethetünk meg.

Wyse, a csúcsgép

Azért az eddigi noteszgépek sem voltak sem lassúak, sem feledhetők – de a Wyse mégis-

Sharp, a praktikus

Nagy huncut ez a Sharp cég! Nem hozott semmi igazán újat a PC-3000-es szériával, aztán mégis van benne valami csábító. Notebooknak kicsi, palmtopnak meg nagy, súlya még fél kiló sincs, RAM-ja azonban 1000 kiló, azaz 1 megabájt. A tudása igazán a felhasználó mindennapos igényeinek megfelelő (nyolc funkciója között van a szövegszerkesztő és a táblázatkezelő is), leginkább a menedzserkalkulátor és a pici PC ötvöze. Kijelzője 640 × 200 pont felbontású, ami CGA-nak felel meg. A központi processzor egy 10 megahertzes 80C88A jelű proci, amitől ugyan nem lesz sebességi rekorder, de vegyük figyelembe, hogy „még növésben van”. 512 kB-os memóriakártyára tárolhatjuk az adatokat, de csatlakoztatható 3,5 inches lemezmeghajtó is.



csak új számot írhat be a történelembe.

Azért, mert a Wyse Decision-Mate 486SLC mégiscsak 486-os... Alaplapja 4–8 MB-os, winchestere 80–120 MB-os, kijelzője pedig 640 × 480-as felbontású 32 szürkeárnyalattal. Külön beépíthető kártyával 2400 baud/sec gyorsaságú faxmodemként is használható.

közül pedig kiemelik a „Notebook-manager” hagyományosnak is mondható szolgáltatásait. A modem-csatlakozások között azonban ott az igazi Commodore-specialitás: a BTX-hálózathoz való csatlakozás lehetősége. Ez a hálózat nemcsak elterjedtségéről híres, hanem arról is, hogy minden Commodore típusal egyszerűen lehet rákapcsolódni.



Kézikészülék a Game Boyhoz

Új kiegészítővel gazdagíthatjuk Game Boy készletünket: a Handy Kit a Nintendo Entertainment System legújabb fejlesztése.

A Handy Kit kinézetre is jobban hasonlít egy úrhajó vezérlőpultjára, mint egy játékgépre. A 25 dollárért kínált (körülbelül 2000 forintnak megfelelő összeg) kiegészítő dobozt egyszerűen a Game Boy tetejére csatlakoztatva sok hasznos funkcióval találkozunk:

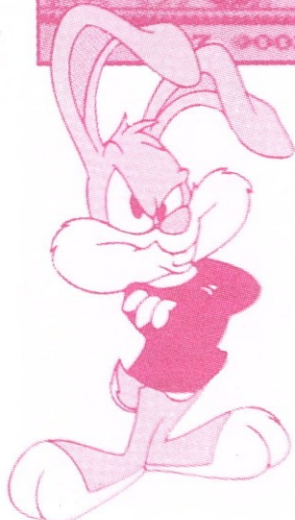
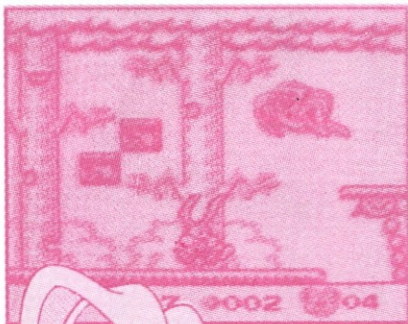
A játék képernyőjét mintegy 50 százalékkal felnagyítja, és gondoltak a rakoncátlan kölykökre is. A háttérrel is megvilágítja – így sötétben, akár a paplan alatt is játszhatunk. Akik kényesek a hanghatások és a zenék minőségére, azoknak bizonyára nagy öröme szolgálna, hogy a sztereohangokat külső „hangfalakon” keresztül hallhatják. A mini-joystick és a nagyobb méretű tűz-gomb megkönnyíti a gyors reakciót köve-



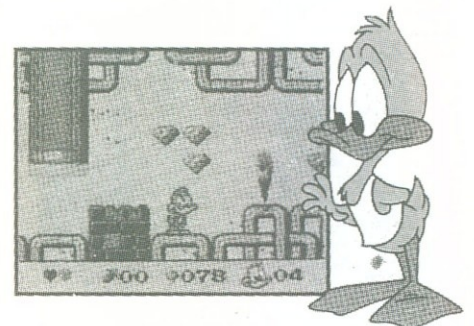
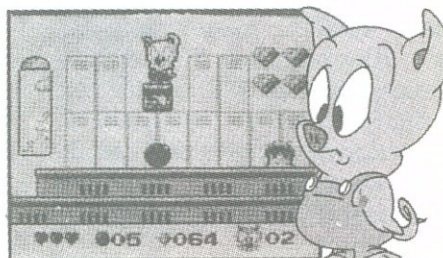
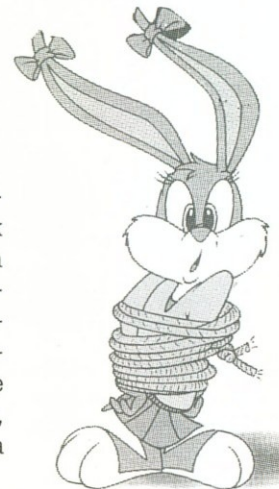
telő játékok végigjátszását. A Handy Kit tényleg kezes, így egészen picire összecsomagolható: a képen látható, hogy a képernyő és a két hangszóró be-

hajtható. A beépített vállpánt megkönnyíti a „szállítást”, de rövidesen elkészül a Handy Kit táskája is, amely a megszokott ötletekkel tarkított hordtáska lesz.

Tiny Toon

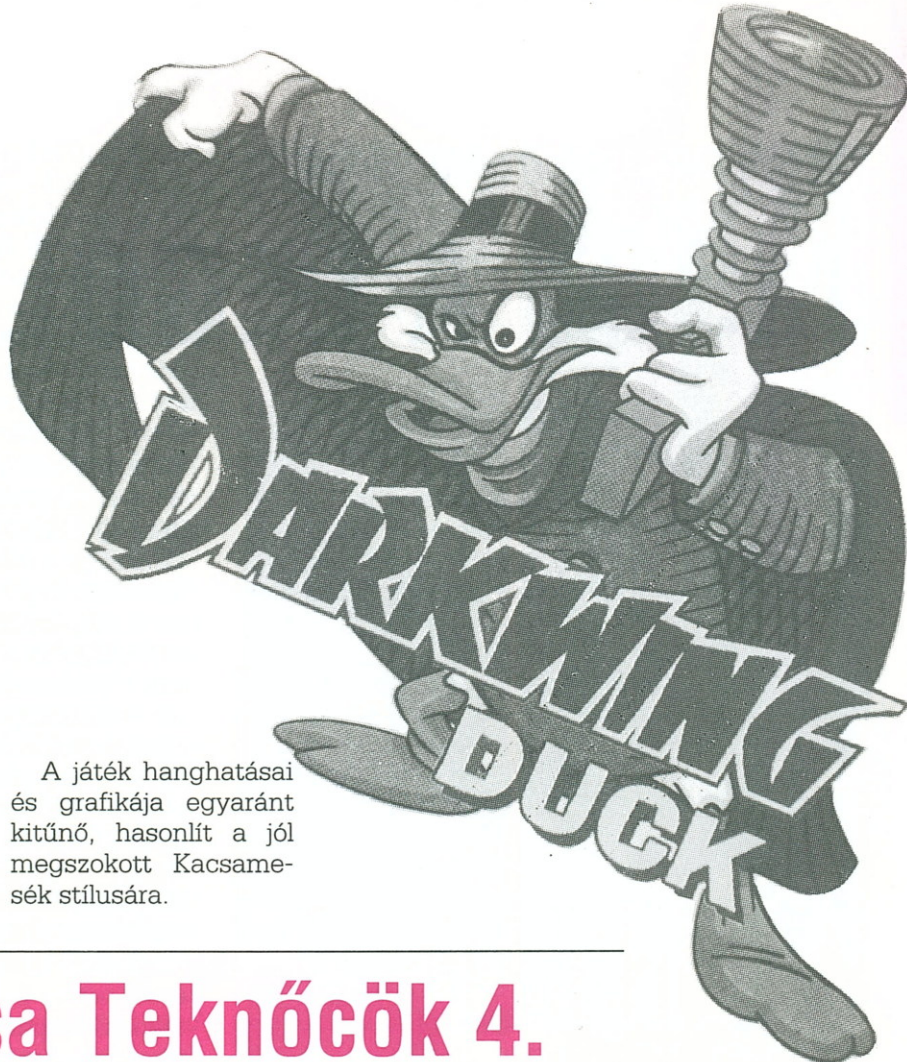


A kedves nyuszi, a kacska és a röfi jól ismert figurák a gyermekmeséket kedvelők körében. A Game Boy ügyességi játékokban bármelyik állatkéval játszhatunk. A küldetés során sok jól ismert rajzfilmfigurával találkozunk. A rendkívül aranyos játék hangja és grafikája egyaránt megfelel a Game Boy képességeinek – az élmény tökéletes, sokszor azt hisszük, hogy mi rendezzük a rajzfilmet.



Egy sötét kacsza

A F.O.W.L. nevű csapat elcsórt egy igen értékes festményt, majd a darabjait a város különböző pontjain rejtette el. Az elloptott részeket Darkwing Duck-nak kell felkutatnia, ellenkező esetben kedves „libuskáját” megfőzik! A Turbografx-16-on futó játékprogram igazi kacsavadászat. Ötféle képernyőn (melyek különböző irányokba scrollozódnak) kell küldetését teljesítenie. A meglelt képdarabkákat puzzle-szerű játékban kell összeilleszteniük.



A játék hanghatásai és grafikája egyaránt kitűnő, hasonlít a jól megszokott Kacsame-sék stílusára.

Tini Nindzsa Teknőcök 4.



Újabb játékkal találkozhatnak a teknőcimádó fiatalok. A Super NES-re készült Turtles in Time című játékban kis zöld hőseink valamelyikével kell végigverekednünk magunkat a pályákon, hogy visszaszerezzük az elloptott amerikai Szabadság-szobrot. Igazi akciójáték – verekezés közben sokszor veszítünk az energiánkból, természetesen az úton talált pizzák felszedésével újra teljes egészségnek örvehetünk. Néhány bonus pályán gördeszkán száguldozhatunk, és persze a pályák végén mindig meg kell küzdenünk valamelyik ellenfelünkkel – legutoljára marad a gonosz Krang, a Szabadság-szobor tolvaja.



Hogyan vásároljunk számítógépet?

MOST A LEGOLCSÓBB!

A következő oldalakon segítséget kívánunk nyújtani mindazoknak, akik mostanában szeretnének számítógépet vásárolni. Megmutatjuk, hogy miért állja meg a helyét még ma is az olcsó „ósvi”, és miképpen tud terjeszkedni a „vadi új”, ám drága masina.



Amiga 500

Sajnos azt már a legelején le kell szögeznünk, hogy a hazai gyártmányú gépek korszaka lejárt. Jó néhány számítógép teljesen eltűnt a piacról, és valószínűleg a felhasználók asztaláról is. Ilyen a Primo és a suliból jól ismert HT-1080Z iskolaszámítógép, és bizony lassan hasonló sorsra jut a Videoton TV Computer is. Amióta a Commodore-ok nevével alacsony áron „beszabadultak” Magyarországra – emlékezzünk csak a nagy bécsi bevásárolótúrákra –, azóta a hazai gépeknek nincs esélyük. Nem is annyira a gépek tudása miatt, hiszen a TVC sokban veri a Commodore 64-et, hanem inkább a szoftverellátás különbözősége miatt. Épp a minap olvastuk a Run-ban, a C-64 tulajdonosok magazinjában, hogy egy bizonyos szoftvercég még ma is évente több mint 100 új programot dob piacra! Ezek a programok özönlnek be az országba, ezzel persze boldog perceket szerezve a játék-örülteknek, szép summát a szoftverkalózoknak és szomorú pillanatok a szoftverfejlesztőknek. Ezzel el is érkeztünk a kiskiszámítógépek nagy esélyeséhez, a Commodore-csaphoz.

A rút kiskacsa

A Commodore több, mint tíz éve jelentette meg az első Commodore 64-et, azt a kis, barna, púpos masinát, amely kezdetben egy kisebb vagyonba került idehaza, de még Németországban is meglehetősen borsos volt az ára (majd' 800 márkába került, és a hozzá való drive is hasonlóan drága volt). A kis gép viszonylag nagy tudását sokan úgy használták ki, hogy hatalmas cégek adatnyilvántartásait próbál-

HELIX
computer

SZERVIZ

1133 Budapest, Kárpát u. 7/A
Telefon: 149-7909
Nyitva: hétfőtől csütörtökig: 9-17 óráig
pénteken: 9-15 óráig

PC XT/AT AMIGA COMMODORE ENTERPRISE TVC

számítógépek és perifériák javítása
(általában 1-3 nap alatt)
Kérésre házhoz megyünk!

ták elvégeztetni vele – kisebb-nagyobb sikerrel. A Commodore cég hamar rájött, hogy a C-64 sok hibával rendelkezik: nehézkes a programozása, a Basic nyelve tökéletesítésre szorul. A Plus/4-es már megfelelt a várakozásnak, de valahogy mégsem örvend akkora népszerűségnek, mint a 64-es.

Megemlíthetjük a C-16-ot is, de kis memóriája miatt ez sem terjedt el igazán. A sláger mindmáig a kis „púpos”, a C-64-es maradt. Ma már persze biztosan kevesen ismerik ebben a formában, ugyanis 2-3 éve új formát kapott, amely leginkább a Commodore 128-asra hasonlít, és manapság mindenki csak Commodore 64 II-nek emlegeti.

Becsületére válik a Commodore-os fejlesztőknek, hogy nem felejtették el a régi jó masinát, így sok új kiegészítővel „tuningolhatjuk” fel a számítógépet. A C-64 II szériához tartozik a 1541-es drive új formája is; jóval kisebb méretű (a tápegység külső dobozba került) és szebb is lett. Az 5,25"-es lemezeket megszoktuk már, de vásárolhatunk 3,5"-es drive-ot is. Ma már igen széles a kiegészítők skálája: rengeteg memóriabővítő, RAM-drive, egér, sőt, hard drive közül választhatunk.

Aki olcsó, de színvonalas, elsősorban játékokra alkalmas gépet szeretne, kevés adatnyilvántartással, annak kiválóan alkalmas a Commodore 64-es. Nem kell attól tartania, hogy elavult gépet vesz: a Commodore 64-es eladásának legjobb éve (a keleti piac megnyílása miatt) a tavalyi volt, és ki tudja, mit hoz a jövő?



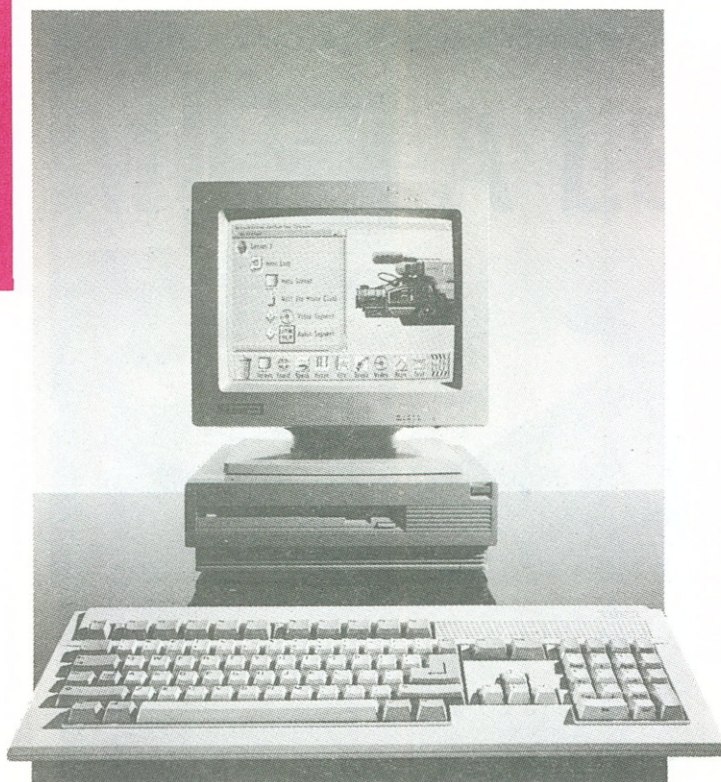
Commodore 64 II

Egy igazi jó barát

A Commodore egyik legjobb húzása (a C-64-es fejlesztése és eladása után) az Amiga számítógépek gyártási jogainak megvásárlása volt (kevesen tudják, hogy az Amiga nem eredeti Commodore, csak egy megvásárolt licenc továbbfejlesztése). A kezdetben sikertelen Amiga 1000-est nyomban az 500-as követte, amely az egész világon ismertté

Konfiguráció:	Fogyasztói ár:	Forgalmazó:
Commodore Amiga 500 Commodore Amiga 500 Plus Commodore Amiga 2000 Philips 8833 II monitor Alpha Data külső drive	44 900 Ft 54 990 Ft 87 500 Ft 32 900 Ft 9 900 Ft	ACOMP Kft. 1141 Budapest Álmos vezér útja 17. Tel.: 183-1817
Atari 1040 STE + egér Atari Mega STE 2 + egér – Atari SM 144 monitorral Atari TT030/2 + egér – Atari PCT 1426 monitorral	49 990 Ft 184 980 Ft 296 980 Ft	Atari Center 1061 Budapest, Andrássy út 40. Tel.: 112-3675
Commodore Amiga 500 Commodore Amiga 500 Plus Commodore 1084S monitor Commodore 64 II Commodore 1541 II meghajtó	44 500 Ft 56 000 Ft 33 000 Ft 14 900 Ft 15 900 Ft	Haár & Haár Bt. Tel.: 173-2008 111-5716
Commodore 64 II Commodore 1541 II meghajtó Commodore 1802 színes monitor Commodore Amiga 500 Commodore 1084S monitor	14 600 Ft 16 600 Ft 27 000 Ft 49 990 Ft 32 000 Ft	Novotrade 2C Kft. 1136 Budapest, Balzac u. 35 Tel.: 140-2954

A feltüntetett árak fogyasztói árak, tehát tartalmazzák az áfát.



Amiga 3000

vált. Eleinte úgy hittük, hogy alacsony ára és nagy tudása miatt teljesen felváltja majd a C-64-est. Sokan nyergeltek át Amigára, de szerencsére mindkét gép életképesnek bizonyult.

Az Amiga 500-as fél megabájtos memóriájával, gyönyörű grafikájával és sztereó hangjával hamar kedvencé vált a játékosok körében. A sok jó tulajdonság miatt a játékprogramok egyre közelebb kerülhettek az életszerűséghez, a szimulátorok szinte tökéletesen adják vissza a valóságot – tényleg élvezet a játék. Beépített 3,5"-es drive-ja a

drágább, de nagyobb kapacitású kislemezeket fogadja, amelyekre 880 kilobájtnyi adatot rögzíthetünk. Egyre jobban elterjed a HD-s meghajtó (High Density, amely 1,44 megabájt kapacitású), a nagyobb gépekbe (Amiga 3000-es család) már ilyeneket szerelnek.

Az Amiga operációs rendszere két részre bontható: a CLI-ből (Command Line Interface) és a Workbenchből indíthatjuk programjainkat. Az előbbi viszonylag kényelmetlen (a parancsokat a megfelelő paraméterezési formában be kell gépelnünk), ezért sokan választják a Workbenchet. Az új, 2.0-s kickstart (a 3000-es családban és az 500 Plus-ban ilyen ROM található) sok segítséget nyújt, és akinek hard diskje is van, az hamar megkedveli az Amiga Workbenchet, ami leginkább a Macintosh-ok rendszerére hasonlít, de nem áll távol a PC-s Windows-tól sem (megjegyezzük, az Amiga Workbench messze gyorsabb és kényelmesebb, mint a Windows 3.0).

Az Amiga – még ha a legkisebbet, az 500-ast vesszük is – kiválóan alkalmas játéokra, zenére, de némi kiegészítővel (hard disk, memóriabővítő) a grafikával, rajzolással, feliratozással, adatbázis-kezeléssel foglalkozók is szívesen használják.

A másik világ

A PC-kompatibilis számítógépek csak nemrégiben törtek be az otthoni számítógépek világába. Áruk jelentősen csökkent (alaposan megközelítette a házi számítógép kategóriát), s ezért mind többen vásárolnak PC-t. A nagyvállalatoktól levedlett XT ma már szinte senkinek sem kell – a minimum az AT. Az XT egyik legnagyobb hátránya – mi más lenne – a sebesség. Ha volt már alkalmunk valamilyen AT-n dolgozni – még ha csak kisebb teljesítményűn is –, garantáltan utálni fogjuk az XT-t. Lassú,

Konfiguráció:	Fogyasztói ár:	Forgalmazó:
Best 2400EB MNP5 belső modem Artec B29 optomechanikai egér Artec A20DLX optikai egér Artec A256C színes kézi scanner Minden egérre örök garancia!	16 125 Ft 1 512 Ft 5 400 Ft 41 625 Ft	Humansoft Kft. 1149 Budapest, Angol u. 24/b. Tel.: 163-2789 Fax : 183-1789
AT286 20MHz, 1MB RAM, 1.2 drive, 40MB winch., mono monitor és bill. AT 386 33MHz, 2MB RAM, 1.2 drive, 40MB winch., mono monitor és bill. Discovery 2400, MNP 5 belső modem	64 000 Ft 96 375 Ft 16 500 Ft	Qwerty 1117 Budapest, Orlay u. 4. Tel.: 166-3098 Fax : 185-2687
AT 386SX 16MHz, 2MB RAM 1.2 és 1.44 MB drive 42MB winchester, 800x600 VGA port SVGA monokróm monitor 101 gombos tasztatúra	139 000 Ft	Szinva Net Kft. 3525 Miskolc, Vologda u. 3. Tel.: (46)346-634 Fax : (46)355-894

A feltüntetett árak fogyasztói árak, tehát tartalmazzák az áfát.

ebből kifolyólag tényleg csak csak egyszerűbb feladatokra alkalmas igazán. Persze ez így nem igaz, hiszen majd' minden AT-re ajánlott program hibátlanul fut – feltéve, ha van türelmünk kivárni. A másik probléma a meghatározó: mindössze 360 kilobájtnyi adat fér el egy lemezen, ez pedig a mai világban egyszerűen nem elég (a CorelDraw 3.0 installálásakor – nehogy megpróbálja valaki XT-n! – 10 darab HD-s lemezről kell dolgoznunk, ami egyenként 1,44 megabájt tömörített adatot tartalmaz). Egyszerűen nem ajánljuk az IBM PC/XT-kompatibilis gépek megvásárlását.

Egy kisebb AT konfigurációt ma már úgy 50 000 forintért is meg lehet kapni: ez alkalmas szöveg-

es felbontásához és színeihez szokott hozzá, a Soundblaster hangkártya használata után pedig a C-64-es fenemód hatásos digitalizált hangját spektrumos pittyegésnek fogjuk hallani.

Aztán ha már elérkezett az idő a továbblépésre, akkor meg lehet kóstolni a 386-ost. Ennek két nagy „fajtájával” találkozhatunk: az SX és a DX jelzésűekkel. Az előbbieket lassabbak, mivel a külső felépítésük „csak” 16, míg a DX jelzésűeké kívül-belül 32 bites. Az SX-szel akár 20–25 MHz-es sebességet is elérhetünk (vigyázat, a 20 MHz-es 286-os és 386-os között nem csekély a különbség!). Ilendő egy nagyobb winchestert tenni a gépbe (úgy 120 megásat), és talán most már ne a monokróm monitort válasszuk. Igazán szép grafikát egy megabájtos grafikus memóriával rendelkező S-VGA kártyák produkálnak – ehhez hozzá való monitor is szükséges.

A non plusz ultra a 486-os. Egy nagyobb teljesítményű gép 50 MHz-en is ketyeghet, a processzorba épített cache és aritmetikai társprocesszor tovább növeli a gép teljesítményét. Nagy felbontású monitort, több száz megás winchestert „köthetünk” rá, vagy CD-ROM-mal, lézerprinterrel egészíthetjük ki, s nem utolsó sorban serverként is kitűnően megállja a helyét – így szinte bármilyen munkára befogható. Ez tényleg a másik véglet, ugyanis egy ilyen konfigurációt már aligha úszunk meg félmillió forint alatt.



Kaiyo SL-200

szerkesztésre, kisebb adatbázisok kezelésére, bonyolultabb számítások végzésére, programozásra. Ekkor elegendő a monokróm monitor, egy kisebb, mondjuk 40 megás winchester, és valamilyen egyszerűbb nyomtató. A billentyűzet természetesen a konfigurációhoz tartozik, és persze az 5,25"-es meghajtó is be van építve a gépbe.

Ha komolyabb grafikai dolgokra, játékokra, zenére szeretnénk fogni a számítógépet, akkor bizony mélyebben a zsebünkbe kell nyúlnunk. Aki Commodore 64-ről szeretne áttérni PC-re, az kezdetben megelégedhet egy 286-os, 16 MHz-es AT-vel; ezt viszonylag jól ki lehet bővíteni. VGA kártyával és monitorral a grafikus megjelenítés sokkal szebb lesz, pláne, ha azt vesszük, hogy szemünk a C-64-

ACOMP Számítástechnikai Kft.

1141 Budapest
Álmos vezér útja 17.
Tel.: 183-1817

Commodore Amiga 500	35920 Ft
Commodore Amiga 500 Plus	43992 Ft
Commodore Amiga 2000	69999 Ft
Philips 8833 II monitor	26320 Ft
Alpha Data külső drive	7920 Ft
512 kB memóriabővítő	4000 Ft
1 MB bővítő Amiga 500 Plushoz	7920 Ft
Commodore 64 II	11680 Ft
Commodore 1541 II	13280 Ft
Commodore 1802 monitor	19920 Ft
Quickshot II Plus joystick	712 Ft
Quickshot II Turbo joystick	552 Ft
Boot Selector	1032 Ft
Handy Scanner 400 dpi	10392 Ft
MIDI Interface	2072 Ft

Áraink az áfá-t nem tartalmazzák!

Nyitvatartás: hétköznap 9–18 óráig
szombaton 9–12 óráig

Basic bővítések 11.

Az előző részben példát láttunk arra, hogyan lehet feldolgozni egy numerikus paramétert. Most lássuk, mi a teendő sztringparaméter esetén.

Példaprogramunk legyen az előző részben közöltek ellenkezője. Tehát alakítson át egy négykarakteres füzért – amely egy hexadecimális számot tartalmaz – tízes számrendszerbeli valós számmá. A szintaxis legyen a következő:

!TRING)

Még mielőtt elkezdenénk a rutin megírását, észre kell vennünk még egy problémát. A függvény végrehajtása ugyanis két olyan hiba miatt is leállhat, amelyre nincs üzenet a BASIC-ben. Az egyik akkor lép fel, ha a megadott füzér hossza nem négy karakter, a másik pedig akkor, ha a sztring olyan karaktert is tartalmaz, amely nem értelmezhető tizenhatos számrendszerbeli számként. Ezért át kell írunk a BASIC melegindítást és ki kell bővítenünk két hibaüzenettel.

Most pedig lássuk a rutint:

```
c000      lda #<FVVRH
c002      ldy #>FVVRH
c004      sta $30a
c007      sty $30b
c00a      lda #<NEWERR
c00c      ldy #>NEWERR
c00e      sta $300
c011      sty $301
c014      rts
c015 NEWERR txa
c016      bmi NINCS
c018      cmp #31
c01a      bcs MYERR
c01c      jmp $a43a
c01f NINCS jmp $a474
c022 MYERR sbc #31
c024      asl
c025      tax
c026      lda ÜZENET,x
c029      sta $22
c02b      lda ÜZENET+1,x
c02e      jmp $a445
```

```
c031 FVVRH  lda #0
c033      sta $d
c035      jsr $73
c038      cmp #„!”
c03a      beq ÚJFV
c03c      jsr $79
c03f      jmp $ae8d
c042 ÚJFV  jsr $73
c045      cmp #„d”
c047      bne SYNTAX
c049      jsr $73
c04c      jsr $ae1
c04f      jsr $b782      a füzér át-
c052      ;              vétele
c052      cmp #4        ha nem 4 ka-
c054      bne BADSTR   rakter hosz-
c056      ;              szú, hiba
c056      ldy #0        a sztring
c058      jsr ÁTVÁLT   átalakítása
c05b      sta $62      16 bites
c05d      iny          számmá,
c05e      jsr ÁTVÁLT
c061      sta $63
c063      ldx #$90     és átalakí-
c065      sec         tása lebegő-
c066      jmp $bc49   pontossá
c069 ÁTVÁLT lda ($22),y AC=a sztring
c06b      ;              következő
c06b      ;              karakterének
c06b      jsr SZÁMÍT  átváltása
c06e      ;              4 bites
c06e      ;              számmá
c06e      asl         feltolása
c06f      asl         a felső
c070      asl         4 bitbe
c071      asl
c072      sta $14     tárolása
c074      iny          AC=a sztring
c075      lda ($22),y következő
c077      ;              karaktere
c077      jsr SZÁMÍT  átváltása
c07a      ora $14     az alsó 4
c07c      ;              bit rásorolá-
c07c      ;              sa a felső
c07c      ;              4-re
c07c      rts
c07d SZÁMÍT jsr $b113   AC=betű ASC
c080      bcc NEMBETŰ kódja?
c082      cmp #„g”   igen, ha „g”-
```

```

c084      bcs NOTHEX  nél nagyobb,
c086      ;          hiba
c086      sbc #54     AC=AC-55
c088      rts
c089 NEMBETŰ cmp #,"0"  ha AC<,"0"
c08b      bcc NOTHEX ASC kódja,
c08d      ;          hiba
c08d      cmp #,"."  ha AC>,"."
c08f      bcs NOTHEX ASC kódja,
c091      ;          hiba
c091      sbc #52f    AC=AC-52f
c093      rts
c094 BADSTR ldx #31
c096      .byte $2c
c097 NOTHEX ldx #32
c099      jmp $a437
c09c SYNTAX jmp $af08
c09f ÜZENET .byte <ÜZENET1
c0a0      .byte >ÜZENET1
c0a1      .byte <ÜZENET2
c0a2      .byte >ÜZENET2
c0a3 ÜZENET1 .asc „string lonG”
c0ae ÜZENET2 .asc „not hex character”
          <c000-c0bf>

```

A rutin \$c04f-ig nem szorul magyarázatra. Itt átvesszük a beolvasott füzért az interpretertől. Ekkor a rendszer megszünteti a füzérveremben az erre a sztringre vonatkozó bejegyzéseket. A füzér címe átkerül \$0022-\$0023-ba, a hosszt AC-ban és YR-ben egyaránt visszkapjuk, a típuskapcsoló pedig

numerikusra állítódik. A rutin első parancsa egy füzérelőnézés, ezért nekünk ezt nem kell külön megtennünk.

Ezt követően ellenőrizzük, hogy a sztringhossz valóban 4 karakter-e. Ha nem, ?STRING LONG hibaüzenetet adunk ki, ellenkező esetben átalakítjuk a sztringet tizenhat bites számmá, amit \$0062-\$0063-on tárolunk, végül átalakítjuk valóssá. A rutin többi része értelemszerű.

A JSR \$B113 parancs ellenőrzi, hogy az AC-ban lévő karakter egy betű ASC kódja-e. Amennyiben igen, a C bit magasra állítódik, egyébként törlődik. Érdekes még az új hibaszámok betöltése. A NOTHEX címkére ugrásnál egyértelmű, de a BADSTR címkénél van egy kis turpisság a dologban. Idekerülve 31-et töltünk XR-be. A következő parancs a .BYTE \$2C-n kezdődik. A \$2c a BIT xxxx kódja, így az utána kezdődő LDX #32-t a BIT operandusaként értelmezi a rendszer, ily módon végrehajtható egy BIT \$A220 parancs, ami nem változtatja meg XR tartalmát. Így amikor a vezérlés a JMP \$A437 parancsra kerül, XR a helyes hibakódot tartalmazza. Ha a rutin a sztringben hexa kódként nem értelmezhető karaktert talál, ?NOT HEX CHARACTER hibaüzenetet kapunk.

Ezzel végére értünk az utasításvégrehajtó rutinon keresztüli bővítések magyarázatának. Akik folyamatosan követték sorozatunkat, azok várják türelemmel a sorozat folytatását.

Írta: **Lukács Krisztián**

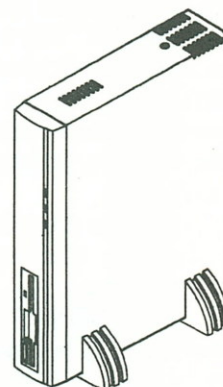
UNITRON

COMPUTER

Amikor a méret is számít

Smartbook: 260x190x45mm, 1 kártyahely
SuperSlim: 300x340x60mm, 2 k. hely
Bobcat II: 310x380x80mm, 3 k. hely
MSC: 390x390x80mm, 5 k. hely

AT 286, 386, 486, mono, vagy színes VGA,
102 gombos billentyűzet



HUMANsoft Elektronikai Kft.
 1149 Bp. Angol u. 24/b.
 T: 163-2879 F: 183-1789

COCOM-korlátok nélkül

Amikor az IBM Magyarországi Kft. nyár közepén, nem sokkal a COCOM-tiltások feloldása után sajtótájékoztatót tartott, a nyitott rendszerekben (OPEN SYSTEMS) is alkalmazható RISC 6000-es nagyszámítógépek voltak a központban. Szóba került azonban két olyan kérdéskör is, amelyről igaz csak sokat beszélnek az üzleti életben. Ugyanis a külföldi kereskedők szemében meghatározó a partner-ország üzleti tisztessége, éppen ezért tudni akarják: milyen a COCOM-listán szereplő országokba irányuló export ellenőrzése, és milyen a szellemi termékek védelme nálunk?

Dr. Straub Elektől, az IBM Magyarországi Kft igazgatójától mindenekelőtt azt kérdeztük: *mi változott azzal, hogy a részleges COCOM-korlátozást is megszüntették?*

– Átkerültünk egy másik csoportba, ahol a korábban már megismert ellenőrzési rendszer betartásával a legfejlettebb technológiák célországja is lehetünk. Ez konkrétan azt jelenti, hogy a legnagyobb teljesítményű IBM gépek is bekerülhetnek, természetesen a regisztrálási előírások megtartása mellett. Ugyanakkor ezek után nekünk kell kötelezettséget vállalni arra, hogy a tilalmi listán szereplő országokba rajtunk keresztül nem juthatnak el ezek a termékek.

– Ezek szerint az ellenőrzés kiterjed a magyar viszonteladók esetleges exportjára is?

– Igen, a COCOM-bizottság előírásainak értelmében vizsgálják, hogy az importált csúcstechnológia a magyar határokon ne kerülhessen tovább. Mindehhez meg kellett változtatni a törvényi szabályozást, és ki kellett dolgozni egy nagyon precíz helyi ellenőrzési rendszert, amelynek betartását rendszeresen vizsgálják. Ennek értelmében minden ellenőrzött technologiaelemet regisztrálni kell, és gondoskodni

kell arról is, hogy ne kerülhessen tovább a tiltott országokba. A szabályozás kidolgozása után, de még azelőtt, mielőtt lekerültünk volna a tilalmi listáról, a COCOM-bizottság tételesen is megvizsgálta egy-egy termék sorsát, és Magyarország minden tekintetben „megbízhatónak” minősült. Ez a gyakorlatban azt jelenti, hogy a regisztrált végfelhasználónál alkalmanként helyszíni ellenőrzés tartható, amikor megvizsgálják, hogy a komputerrendszer valóban ott van-e, és arra használják-e, amire az engedély szól.

– Ilyenkor bizonyára vizsgálják azt is, hogy a felhasználó legálisan használja-e a szoftvereket. Nyílt titok, hogy nálunk elég nagyarányú a „feketemasolatok” alkalmazása; mit szólnak ehhez a szoftverházak képviselői?

– A szellemi jogok védelme a számítástechnikában különös súllyal jelentkezik. A szoftvermásolás azonban nemcsak nálunk alakult ki, világszerte folyamatosan használnak illegálisan megszerzett programokat – bár ma már a legtöbb kelet-ázsiai klóngyártó is megveszi a licenz-jogokat, elmúlt már a kezdeti „kalóz-korszak”. De visszatérve Magyarországra, egy olyan cég, mint az IBM, csak akkor tud bizonyos tevékenységekbe belekezdeni, ha a jogrendszerben is megvannak a garanciái annak, hogy az itt alkalmazott, vagy itt létrehozott szellemi termékek ugyanolyan jogi védelmet kapnak, mint nyugaton. Tehát biztosak lehessünk abban, hogy akik használják a programjainkat, fizetnek is érte – illetve az illegális felhasználókkal szemben jogunk legyen eljárni, legyenek törvénybe foglalt szankciók-lehetőségek.

– Sajnos a kialakult gyakorlat szerint bocsánatos bűnnek számít a szoftvermásolás. Bár mindenki tudja, hogy jogtalan, amit tesz, és könnyen kimondják, hogy a jó programcsomagok drágák, amire a cégeknek sincs elég

pénzük, a magánembereknek pedig még annyi sem.

– Ez azonban nem fogadható el mindenható érvként. Gondoljuk csak végig: a szoftverfejlesztés drága üzleti tevékenység, jelentős kezdeti befektetésre van szükség. Ha viszont nincs lehetőség arra, hogy a szoftverfejlesztésbe fektetett munka profitot is hozzon – mert a felhasználók kézen-közön szerzik be a programokat – akkor senki sem fog szoftvert fejleszteni. Ha ez így van, akkor nálunk elsorvad a fejlesztés, ami nyilvánvalóan nagy kárt okoz egy olyan területen, ahol közismerten jó adottságaink vannak, hiszen a szoftvereseink világszerte elismertek. Ráadásul egy megbízhatatlan piacra a nagy szoftverházak sem adnak majd eladási jogokat, kizárjuk magunkat a szoftvervilágból. Rövid távon az az egyik legnegatívabb következménye az illegális alkalmazásnak, hogy gyakorlatilag lehetetlenné teszi a magyar szoftveripar kialakulását. Azokban az országokban, ahol korábban hasonlóan nagymértékű volt a szoftverkalózkodás, ma már jogi és pénzügyi eszközökkel, célzott reklámmal igyekeznek drasztikusan csökkenteni a feketealkalmazásokat, mert rájöttek arra, hogy nemzetközi elismertségüket, gazdasági érdekeiket veszélyezteti az a helyzet. Nekünk sem szabad elfogadni, hogy a gazdasági korlátokra hivatkozva elfogadható legyen az illegális alkalmazás. Szerencsére javul a helyzet, ma már – ha a PC-s alkalmazásokat nem nézzük – a korábbinál sokkal korrektebbek a felhasználók, a közepes és a nagygépek felhasználói között fel sem merül az illegális szoftverhasználat.

Így legyen. Ha már a COCOM-listáról lekerültünk, kialakulóban van a szellemi termékeket rendesen védő törvény, nem lenne szerencsés „olcsó kis szoftver-lopások” miatt eljátszani a fejlett világ bizalmát.

–ray

200 GRAMMI PROGRAMM

64 oldal



PROGRAM-TARTALOMJEGYZÉK

Plus/4- és C-64-programok

Grafikus terület invertálása	18
Hullámszás	18
Label Basic	19
Két cica	21
Dator	24
Disill	25
Set in Sid	27
Bűvös négyzetek	27
Prompt	27
IBM kurzor	28
Exchange floppy	28
Szabad blockok	28
Grafik picture on	29
Egyszerű magnókezelés	29
Filenév-kereső	29
Autóverseny	80
Sprite-kereső	80

CSAK LEMEZEN!!!

Áramkör-tervező	30
Karaktergenerátor	30
Szókitaláló	30
Telefonkönyv	30
Görög betűk	30
Órarendkészítő	30
Kincsesláda	30
Csapdában	30
Iskolás gyerek	30

PC-programok

Kezdőlépések	32
Négyzetek és körök	33

Számok egész számú osztói	34
Inverzió	35
Példák inverzió segítségével megoldható feladatokra	58
Axonometrikus kép rajzolása	60
Cikloisok	61

CSAK LEMEZEN!!!

Betűíró	64
Zenebeíró	64
Kúpszeletek simulókörei	64
Ellipszis-szerkesztések	64
Tesoszámok	64
Számkiíró	64
Dátumköz-számoló	64

C-64-programok

Rajzok	65
Rajzoló	66
Sűrítő	66
Music voices	67
Color	68
Csillagászati jelek	68
Sprite Basic	69
Betöltő	70
Invertáló	70
Kurzor help	71
Monitor	71
Örökélet	72
Dyinp	73
Villogás	74
Animator	75
Diagram	77
Monitor Plus	77
Grafika	78

Kezdőlépések kezdő programozóknak

Az alábbiakban azoknak a Commodore-os olvasóknak kívánunk segíteni, akiknek nincs elegendő számítógépes gyakorlatuk.

Kezdjük talán a programok begépelésével. A villogó négyzet – ezt kurzornak nevezzük – azt a helyet jelöli, ahol a következő betű megjelenik a képernyőn. A program minden sora sorszámmal van ellátva. A programot soronként gépeljük be, ügyelve arra, hogy a begépelést a sor elején kezdjük, és minden sor végén megnyomjuk a RETURN billentyűt (egy sor két képernyősornyi betűből és számból áll – ha ezt túllépjük, a Basic értelmező már új sornak veszi a további gépelést, ezért vigyázni kell a sorok hosszával). Ekkor a kurzor a következő sor elejére ugrik, és már kezdhetjük is a következő sor beírását.

A sorokat karakterenként, pontosan úgy gépeljük be, ahogy az a listában áll. Ha egy sort elrontunk, gépeljük újra, vagy ha megtaláltuk a hibát, a kurzorgombokkal a megfelelő helyre állva javítsuk a pontatlanságot.

Ha befejeztük a beírást, vagy kíváncsiak vagyunk programunk listájára, töröljük le a képernyőt – tartsuk nyomva valamelyik SHIFT billentyűt, és üssük le a CLR/HOME feliratú gombot –, majd gépeljük be a LIST parancsszót, és üssük RETURN-t. Ekkor a képernyőn legördül a program listája. A listázás a CONTROL (régebbi gépek esetén CTRL) billentyű lenyomásával lassítható. Ha a listázást meg akarjuk szakítani, akkor a kívánt helyen nyomjuk meg a RUN/STOP billentyűt.

Ha befejeztük a begépelést, vagy félbe szeretnénk hagyni, a gép kikapcsolása előtt tárolnunk kell („el kell mentenünk”) a programot – a számítógép ugyanis kikapcsolás után elfelejti a benne lévő programot.

A programok tárolása:

A program elmentése kazettás egység esetén a következő lépésekből áll:

Helyezzünk a kazettás egységbe egy kazettát, és tekerjük oda, ahova a programot rögzíteni szeretnénk. Írjuk be a SAVE„NÉV”,1 szöveget. A NÉV helyére mindig a program neve kerüljön – amit jegyezzünk is meg a számláló állásával együtt. Ezután nyomjuk meg a RETURN billentyűt. Ekkor két sorral lejjebb megjelenik a PRESS PLAY & RECORD ON TAPE szöveg. Nyomjuk le a kazettás egységen a REC és a PLAY gombokat egyszerre. A képernyőn megjelenik a következő sorban a SAVING NÉV szöveg – természetesen a NÉV helyén az általunk adott programnév áll –, és a képernyő a keret színére vált a kimentés ideje alatt.

Kimentés a lemezegegyeséggel floppyra:

Helyezzünk egy lemezt címdalával felfelé a meghajtóba, majd zárjuk rá a meghajtó ajtaját. (Ha a lemez teljesen új, még senki sem használta, akkor először formattálnunk kell. Ehhez adjuk ki a következő parancsot: OPEN-15,8,15,,N:NÉV,ID”: CLOSE15 – a NÉV helyére a lemez neve kerüljön – azon szabályok megtartásával, amelyeket a kazettás egységnél leírt NÉV-re adtunk. Az ID helyére két karakteres azonosító kerüljön, de vigyázzunk arra, hogy minden lemezünknek más legyen az ID kódja. Ha ezzel megvagyunk, üssük le a RETURN gombot. Ekkor a meghajtó úgy másfél percig kattogva dolgozik a lemezen, majd megjelenik a READY felirat a kurzorral.)

Ha a lemezt már használta valaki (vagy elvégeztük a fent említett műveletet), írjuk be a SAVE „NÉV”,8 paran-

csot és üssük le a RETURN-t. Ekkor a meghajtó ismét munkába kezd, és megjelenik a SAVING NÉV szöveg. Amikor befejezte az adatok elmentését, a READY üzenetet és a kurzort látjuk.

Ha el akarjuk indítani a programot, akkor gépeljük be a RUN parancsszót, majd üssük RETURN-t. Ha minden helyesen gépeltünk be, a program hibátlanul le fog futni. Hiba esetén a hibaüzenet (amely önmagában is utal a hiba fajtájára, ezek leírását a számítógéphez kapott könyvben találjuk meg) mögött van egy sorszám, amely a téves sort jelzi. Gépeljük be újra a sort, vagy javítsuk ki a hibát. Fontos, hogy csak teljes egészében begépelte programot indítsunk el!

A programok visszatöltése:

Ha elmentett programot akarunk újra használni, vagy egy félig begépelte folytatni, akkor azt vissza kell tölteni a gép memóriájába. Vigyázzunk, mert ilyenkor az éppen a gépben lévő program teljesen elvész!

A betöltés kazettás egységről:

Behelyezzük a programot tartalmazó kazettát az egységbe, és pörgessük egy-két fordulattal a program elé. Írjuk be a LOAD„NÉV”,1 parancsot – ahol a NÉV helyén pontosan ugyanannak a névnek kell állnia, mint amit a kimentéskor használtunk –, majd üssük RETURN-t. Ekkor megjelenik a PRESS PLAY ON TAPE szöveg. Nyomjuk le a magnón a PLAY billentyűt, minek hatására kiíródik a SEARCHING FOR NÉV szöveg, a képernyő a keret színére vált, és addig nem vált vissza, amíg a gép egy programot nem talál, és a képernyőn meg nem jelenik a FOUND NÉV felirat. A NÉV helyén a megtalált program neve áll. Nyomjuk meg a C= jelű billentyűt, így a kép újra a keret színére vált. Ha a NÉV helyén nem a LOAD-nál megjelölt név áll, a gép tovább keresi azt, ha viszont igen, akkor még a keretszínre váltás előtt kiíródik a LOADING szöveg. Abban az esetben, ha nem adtunk meg nevet (egyszerű „csupas” LOAD parancsnál), akkor a szalagon soron következő program töltődik be. Amikor a gép befejezte a töltést, visszakapjuk a képernyőt, megjelenik a READY felirat, és a kurzor. Az így betöltött programot RUN utasítással indíthatjuk el.

Lemezegegyeségen egyszerűbb a helyzet. A meghajtóba helyezzük azt a lemezt, amelyen a betölteni kívánt program van. (Ha kíváncsiak vagyunk arra, hogy milyen programok vannak a meghajtóban lévő lemezen, írjuk be a LOAD„\$”,8 parancsot és üssük RETURN-t. Amikor a töltés befejeződött, gépeljük be a LIST parancsot és most is üssük RETURN-t. Vigyázzunk, mert a katalógus is programként töltődik, és törli a bent lévő programot!) Írjuk be a LOAD„NÉV”,8 parancsot – a NÉV-re ugyanaz vonatkozik, mint kazettás egység esetén –, majd üssük RETURN-t. Ekkor megjelenik a SEARCHING FOR NÉV üzenet – a NÉV helyén a LOAD-nál megadott név áll. Ha a program megtalálható a lemezen, megjelenik a LOADING felirat, majd a töltés befejeztével a READY. és a kurzor. Amennyiben a program nincs rajta a lemezen, ?FILE NOT FOUND ERROR hibaüzenetet kapunk. A sikeresen betöltött program futtatása a RUN paranccsal indítható. Végül, ha új programot akarunk beírni, de már van egy a memóriában, akkor töröljük le a képernyőt, írjuk be a NEW parancsszót, és üssük RETURN-t. Ekkor a memóriában található program „eltűnik”.

Grafikus terület invertálása

A program tetszőlegesen kijelölt grafikus terület invertálását végzi. A képernyőrészt úgy jelölhetjük ki, hogy megadjuk a bal és a jobb oldalán változatlanul maradó oszlopok számát (ez egyébként X0 és X1), majd a felül és alul maradó oszlopokat (az előbbihez hasonlóan Y0 és Y1). Az erre vonatkozó megköteéseket a program tartalmazza.

Írta: **Erdős András**

```

0 REM***** <AA
  ****
1 REM***** <CA
  ****
2 REM* GRAFIKUS KEP MERETEZETT INVER <F6
  ZE *
3 REM***** <0B
  ****
4 REM***** <2B
  ****
5 DATA A0,00,A5,E0,85,D4,A5,E1 <AD
6 DATA 85,D0,A5,E2,85,D1,20,60 <82
7 DATA 06,20,40,06,B1,D6,45,95 <B6
8 DATA 91,D6,18,A5,D0,69,01,85 <FF
9 DATA D0,A5,D1,69,00,85,D1,C5 <80
10 DATA E6,D0,E3,A5,D0,C5,E5,D0 <5E
11 DATA DD,E6,D4,A5,D4,C5,E4,D0 <9F
12 DATA CD,60,00,00,00,00,00,00 <6F
13 DATA 38,A9,07,E5,E8,AA,A9,01 <7D
14 DATA E0,00,F0,04,0A,CA,D0,F8 <3A
15 DATA 85,95,60,00,00,00,00,00 <9D
16 DATA 0,0,0,0,0,0,0,0 <45
17 DATA A9,0,85,D6,A9,20,85,D7 <F5
18 DATA 18,A5,D6,65,D0,85,D6,A5 <A7
19 DATA D7,65,D1,85,D7,A5,D4,29 <C5
20 DATA 7,18,65,D6,85,D6,A5,D7 <9C
21 DATA 69,0,85,D7,A5,D0,29,7 <B2
22 DATA 85,E8,38,A5,D6,E5,E8,85 <A0
23 DATA D6,A5,D7,E9,0,85,D7,A5 <38
24 DATA D4,4A,4A,4A,85,DB,A9,0 <D8
25 DATA 85,DC,EA,EA,EA,EA,EA,EA <D9
26 DATA EA,EA,EA,6,DB,26,DC,6 <9D
27 DATA DB,26,DC,6,DB,26,DC,6 <43
28 DATA DB,26,DC,6,DB,26,DC,A2 <35
29 DATA A,18,A5,D6,65,DB,85,D6 <32
30 DATA A5,D7,65,DC,85,D7,CA,D0 <76
31 DATA F0,60,P <4E
32 FOR T=0 TO 221:READ A$:IF A$="P" T <A6
  HEN 34
33 A=A+DEC(A$):NEXT T <07
34 IF A<>28744 THEN PRINT "[CTRL/9][F <55
  LASH-ONJADATHIBA!!![CTRL/0][FLASH-
  OFF]":STOP
35 RESTORE :FOR T=0 TO 209:READ A$:PO <6F
  KE 1536+T,DEC(A$):NEXT T
36 GRAPHIC 1,1 <8A
37 CIRCLE 1,160,100,150,90:PAINT ,100 <90
  ,100
38 REM***** <75
  ****
39 REM* Y0=$E0 X0<=$E1 X0>=$E2 <5C
  *

```

```

40 REM* Y1=$E4 X1<=$E5 X1>=$E6 <5C
  *
41 REM***** <D4
  ****
42 POKE 224,20:POKE 225,40:POKE 226,0 <8D
43 POKE 228,180:POKE 229,40:POKE 230, <28
  1
44 GET KEY F$:SYS 1536:GOTO 44 <D5

```

Hullámzás

Az alábbi Plus/4-es program segítségével a nagyfelbontású és a szöveges képernyőket hullámoztat-hatjuk vízszintesen. A hullámzás a 1606-os címen változtatható. Ez, illetve a rutin ki- és bekapcsolása funkcióbillentyűkkel is elérhető.

Írta: **Erdős András**

```

0 REM***** <F7
1 REM***** <14
2 REM** HULLAMZAS ** <C9
3 REM***** <54
4 REM***** <74
5 DATA AD,09,FF,8D,09,FF,A5,D0 <F1
6 DATA C9,01,F0,5A,C9,02,F0,5B <F3
7 DATA C9,03,F0,5C,C9,04,F0,5D <25
8 DATA C9,05,F0,5E,C9,06,F0,5F <8C
9 DATA C9,07,F0,60,C9,08,F0,5C <E1
10 DATA C9,09,F0,53,C9,0A,F0,4A <96
11 DATA C9,0B,F0,41,C9,0C,F0,38 <D2
12 DATA C9,0D,F0,2F,C9,0E,F0,26 <0F
13 DATA A9,08,8D,07,FF,20,11,DB <B1
14 DATA AD,0B,FF,69,10,8D,0B,FF <5E
15 DATA E6,D0,A5,D0,C9,10,D0,04 <C8
16 DATA A9,00,85,D0,4C,C3,FC,0 <91
17 DATA 0,0,0,0,0,0,A9,08 <BA
18 DATA 4C,3C,06,A9,09,4C,3C,06 <7C
19 DATA A9,0A,4C,3C,06,A9,0B,4C <7F
20 DATA 3C,06,A9,0C,4C,3C,06,A9 <D4
21 DATA 0D,4C,3C,06,A9,0E,4C,3C <F7
22 DATA 06,A9,0F,4C,3C,06,78,A9 <AB
23 DATA FA,8D,14,03,A9,05,8D,15 <7A
24 DATA 03,58,60,78,A9,0E,8D,14 <C2
25 DATA 03,A9,CE,8D,15,03,58,A9 <C0
26 DATA 08,8D,07,FF,60,0 <A4
27 FOR T=0 TO 180:READ A$:IF A$="@" T <86
  HEN 29
28 RT=RT+DEC(A$):NEXT <E3
29 IF RT<>18265 THEN PRINT "ADATHIBA! <9D
  !!":STOP
30 RESTORE :FOR T=0 TO 172:READ A$:PO <37
  KE 1530+T,DEC(A$):NEXT
31 KEY 1,"P[SH/0]1606,P[SH/E](1606)+1 <6C
  "+CHR$(13)
32 KEY 2,"P[SH/0]1606,P[SH/E](1606)-1 <F0
  "+CHR$(13)
33 KEY 3,"SYS1672"+CHR$(13) <29
34 KEY 8,"SYS1685"+CHR$(13) <57
35 GRAPHIC 1,1:SYS 1672 <79
36 CIRCLE 1,160,100,20,99 <A7
37 PAINT 1,160,100 <D8
38 CIRCLE 1,60,50,50,30,,40:PAINT ,6 <B4
  0,50
39 CIRCLE 1,260,50,50,30,,310:PAINT <95
  ,260,50
40 GET KEY S$:SYS 1685:GRAPHIC 0,1 <CC

```

Label Basic

Bonyolultabb programok írásakor mindenkinek komoly problémát jelent a szubrutinok kezdőcímeinek fejbentartása. Sőt, Renumber kiadása után még ezeket a számokat is újra kell tanulnunk. A következő program ebben segít a felhasználóknak. Új – egybetűs – utasításokkal bővíti a Basicet (a gép nem téveszti őket össze a változó nevekkkel, hiszen azok csak a kifejezésben vagy az egyenlőségjel bal oldalán szerepelhetnek).

A bővítés új utasításai:

@<címkenév> – a programsor elején kell állnia, utána kettősponttal elválasztva egyéb utasítás is lehet. Az így megjelölt sorokra való hivatkozás módja: Goto után vagy bárhol, ahol sorszámnak kellene állnia, a címke nevét adjuk meg, elérve egy karaktert;

W – kiírja a programban szereplő összes címkét,

T – futás közben kiírja azokat a címkéket, amelyeknél éppen jár a program. Kikapcsolását az O utasítással végezzük;

R – minden hivatkozó (például ugró) utasítás és az utána következő címke közé beszúrja a megfelelő sorszámot;

L@<címkenév> – a címke sorától folyamatosan listázza a programot;

L<címke> – kiír minden olyan programsort, ahol hivatkozás van a megadott címkére;

C – a Label Basicben megírt programot közönséges Basic-programmá fordítja le. Ezután bővítés nélkül is használhatjuk programunkat.

A programot követő játék (Két cica) Label Basicben íródott.

Írta: **Koszper Vilmos**

```

10 OPEN 1,8,2,"LABEL BASIC,PRG,W" <13
20 PRINT#1,CHR$(1);CHR$(16); <3B
30 FOR X=4097 TO 6808 <6C
40 READ A$:A=DEC(A$) <3E
50 PRINT#1,CHR$(A);:E=E+A <01
60 NEXT X <0E
70 CLOSE 1 <7B
80 IF E<>254589 THEN PRINT "HIBA":END <D3
90 DLOAD "LABEL BASIC" <03
100 DATA 0B,10,01,00,9E,34,31,32 <36
110 DATA 38,00,00,00,00,00,00,A9 <68
120 DATA 22,85,2C,20,41,45,49,4F <56
130 DATA 55,4C,69,10,4C,83,10,4C <EC
140 DATA A0,10,4C,8C,10,4C,99,10 <2C
150 DATA FF,AD,29,10,F0,06,CE,29 <85
160 DATA 10,4C,42,CE,A9,7F,20,70 <8C
170 DATA DB,C9,DB,D0,F4,78,A9,FF <68
180 DATA 8D,29,10,BA,BD,05,01,85 <AB
190 DATA 39,85,A1,BD,06,01,85,3A <7D
200 DATA 85,A2,20,43,13,58,20,4F <C2
210 DATA FF,0D,3F,3F,20,49,4E,20 <5D
220 DATA 24,00,20,FB,FA,4C,D2,8C <8F

```

```

230 DATA 20,8C,10,78,A9,EF,A2,16 <74
240 DATA 8D,02,03,8E,03,03,A9,B7 <43
250 DATA A2,16,8D,08,03,8E,09,03 <43
260 DATA 58,60,78,20,CE,F2,20,17 <23
270 DATA 81,58,60,78,A9,2A,A2,10 <D2
280 DATA 8D,12,03,8E,13,03,58,60 <D7
290 DATA 78,A9,42,A2,CE,D0,F1,20 <97
300 DATA 67,C5,20,6C,10,20,4F,FF <2C
310 DATA 0D,20,20,4C,41,42,45,4C <60
320 DATA 20,42,41,53,49,43,2C,20 <0C
330 DATA 4B,4F,53,5A,50,45,52,20 <32
340 DATA 56,49,4C,4D,4F,53,20,31 <7B
350 DATA 39,39,30,00,A9,97,A2,1A <F4
360 DATA 85,2B,86,2C,20,7B,8A,4C <58
370 DATA 7E,86,00,00,00,00,00,00 <27
380 DATA 00,00,00,00,00,00,00,00 <7C
390 DATA 00,00,00,00,00,00,00,00 <3F
400 DATA 00,00,00,00,00,00,00,00 <CF
410 DATA 00,00,00,00,00,00,00,00 <8E
420 DATA 00,00,00,00,00,00,00,89 <8E
430 DATA 8A,8C,8D,D5,D6,D7,CB,A7 <55
440 DATA A5,2B,A4,2C,85,3B,84,3C <78
450 DATA 60,A5,3B,20,3A,11,A5,3C <16
460 DATA 4C,3A,11,20,47,11,85,3C <28
470 DATA 20,47,11,85,3B,60,00,00 <B9
480 DATA 00,00,00,00,00,00,00,00 <D5
490 DATA 00,AC,2F,11,C0,0A,B0,17 <D6
500 DATA 99,30,11,C8,D0,09,AC,2F <0C
510 DATA 11,F0,1F,88,B9,30,11,8C <B2
520 DATA 2F,11,60,A0,00,F0,F8,20 <BA
530 DATA 4F,FF,3F,3F,53,54,41,43 <56
540 DATA 4B,20,46,55,4C,4C,00,4C <16
550 DATA 81,86,20,4F,FF,3F,3F,53 <5E
560 DATA 54,41,43,4B,20,45,4D,50 <C8
570 DATA 54,59,00,4C,81,86,A9,01 <EC
580 DATA 2C,A9,02,2C,A9,03,2C,A9 <E0
590 DATA 04,18,65,3B,85,3B,90,02 <E5
600 DATA E6,3C,60,A0,00,2C,A0,02 <2D
610 DATA 2C,A0,04,A5,3B,99,FF,10 <BC
620 DATA A5,3C,99,00,11,60,20,73 <F0
630 DATA 04,90,FB,60,20,73,04,F0 <CD
640 DATA 13,C9,2C,F0,0F,C9,22,D0 <2D
650 DATA F3,20,73,04,C9,22,F0,EC <4B
660 DATA C9,00,D0,F5,60,A0,00,4C <65
670 DATA A5,04,A2,00,DD,08,11,F0 <83
680 DATA 06,E8,E0,09,90,F6,18,60 <09
690 DATA 00,00,00,00,00,00,A2,00 <19
700 DATA 68,9D,D9,11,E8,E0,06,D0 <11
710 DATA F7,20,9A,8A,A2,05,BD,D9 <43
720 DATA 11,48,CA,10,F9,60,00,00 <EF
730 DATA 00,A0,FF,A2,00,8A,F0,12 <93
740 DATA A0,00,A9,FF,AA,D0,0B,A9 <00
750 DATA 00,AA,A8,F0,05,A9,00,A8 <37
760 DATA A2,FF,8C,F9,11,8D,F8,11 <20
770 DATA 8E,F7,11,20,11,11,20,31 <7F
780 DATA 14,D0,03,4C,BA,12,2C,F7 <EF
790 DATA 11,10,03,20,18,14,20,88 <4B
800 DATA 11,20,79,04,F0,25,C9,83 <37
810 DATA F0,0E,C9,8F,D0,06,20,0B <16
820 DATA 8E,4C,32,12,C9,40,D0,06 <94
830 DATA 20,B0,8D,4C,32,12,C9,22 <E0
840 DATA F0,18,29,FF,30,21,20,73 <26
850 DATA 04,D0,F3,C9,3A,F0,05,20 <57
860 DATA 7F,11,90,BA,20,73,04,4C <0C
870 DATA 35,12,20,73,04,C9,22,F0 <7E
880 DATA E5,C9,00,D0,F5,F0,E8,20 <82
890 DATA CB,11,90,DA,20,C4,12,90 <9E
900 DATA 06,20,79,04,4C,5A,12,2C <AA
910 DATA F7,11,30,03,4C,2E,13,2C <19
920 DATA F8,11,10,06,20,54,11,4C <C7
930 DATA D2,8C,20,4F,FF,3F,4C,42 <8E
940 DATA 4C,20,4D,49,53,53,49,4E <8F
950 DATA 47,00,20,DD,14,20,20,16 <4F
960 DATA 20,43,13,20,54,11,4C,8F <C0
970 DATA 8D,2C,F8,11,10,01,60,4C <64
980 DATA 4D,13,00,C9,CB,D0,07,20 <DF
990 DATA 73,04,C9,A4,D0,3B,8D,C3 <F1
1000 DATA 12,20,94,11,20,A7,11,C9 <94
1010 DATA 5C,D0,1B,20,97,11,20,AD <00

```

1020	DATA	11, 20, 9A, 11, 2C, F7, 11, 30	<C4	1800	DATA	D0, 01, 68, 9D, DB, 10, 4C, 3B	<03
1030	DATA	22, 2C, F9, 11, 30, 23, 20, 56	<4C	1810	DATA	15, 38, A5, 3B, ED, 06, 15, A5	<24
1040	DATA	13, 90, 17, 20, 79, 04, C9, 2C	<DD	1820	DATA	3C, ED, 07, 15, 60, A2, 00, BD	<5B
1050	DATA	D0, 0F, AD, C3, 12, C9, A4, F0	<FC	1830	DATA	DB, 10, DD, E9, 10, D0, 05, E8	<22
1060	DATA	D0, C9, 89, F0, CC, C9, 8D, F0	<D1	1840	DATA	E0, 0E, 90, F3, 60, F0, 03, 4C	<24
1070	DATA	C8, 38, 60, 20, C3, 15, 4C, F2	<47	1850	DATA	A1, 94, 20, DE, B6, 20, 1A, 11	<F5
1080	DATA	12, 20, DC, 18, 4C, F2, 12, 20	<81	1860	DATA	20, 3F, 14, 20, 08, 12, 20, 0E	<D6
1090	DATA	4F, FF, 3F, 53, 54, 42, 2E, 20	<9B	1870	DATA	12, 20, 24, 11, 20, 18, 88, 20	<02
1100	DATA	00, 20, DD, 14, 20, 43, 13, 20	<AA	1880	DATA	4B, 88, 4C, DF, 11, AD, FF, 10	<73
1110	DATA	54, 11, 4C, 4C, CC, 20, 4F, FF	<80	1890	DATA	AE, 00, 11, 85, 3B, 86, 3C, A0	<7C
1120	DATA	3F, 58, 58, 58, 20, 00, 20, DD	<96	1900	DATA	00, B9, 00, 01, F0, 05, C8, 91	<A3
1130	DATA	14, 20, 43, 13, 20, 54, 11, 4C	<0F	1910	DATA	3B, D0, F6, C8, 20, A5, 04, C9	<17
1140	DATA	81, 86, A9, FF, 8D, F3, 04, A9	<1F	1920	DATA	20, F0, F8, C9, 3A, B0, 0A, C9	<C5
1150	DATA	D0, 8D, 13, FF, 20, 18, 88, 20	<B5	1930	DATA	30, 90, 06, A9, 20, 91, 3B, D0	<92
1160	DATA	4B, 88, 4C, DF, 11, 20, 04, 14	<F5	1940	DATA	EA, 60, AD, 01, 11, AE, 02, 11	<A7
1170	DATA	AD, 06, 11, D0, BA, 38, A9, 06	<31	1950	DATA	85, 3B, 86, 3C, 20, 28, 15, 2C	<24
1180	DATA	ED, 05, 11, B0, 02, A9, 00, 8D	<59	1960	DATA	F8, 11, 30, 22, 20, 11, 11, 20	<83
1190	DATA	07, 11, F0, 0B, 20, E9, 13, 20	<7E	1970	DATA	90, 14, D0, 02, 18, 60, AE, FD	<63
1200	DATA	F8, 13, 90, 96, 20, 83, 13, 20	<1F	1980	DATA	10, AD, FE, 10, 20, 13, 16, 20	<D7
1210	DATA	F3, 18, AD, 07, 11, 20, 8A, 11	<12	1990	DATA	96, 15, AD, 03, 11, AE, 04, 11	<D8
1220	DATA	38, 60, A5, 61, 48, A5, 62, 48	<8B	2000	DATA	85, 3B, 86, 3C, 38, 60, 20, 66	<E7
1230	DATA	A0, 00, 20, DC, 13, D0, 07, 68	<34	2010	DATA	15, D0, EF, AD, F9, 10, AE, FA	<1A
1240	DATA	85, 2E, 68, 85, 2D, 60, 78, 8D	<2B	2020	DATA	10, 85, 14, 86, 15, 20, 3D, 8A	<55
1250	DATA	3F, FF, 20, D0, 13, 90, 04, A9	<00	2030	DATA	20, 02, 8B, 20, E1, FF, D0, DA	<4F
1260	DATA	20, D0, 02, B1, 2D, 91, 61, 8D	<CE	2040	DATA	18, 60, 85, 62, 86, 63, A2, 90	<B5
1270	DATA	3E, FF, 58, 20, B5, 13, 20, BE	<53	2050	DATA	38, 20, CE, A2, 4C, 71, A4, A9	<1F
1280	DATA	13, 4C, 8B, 13, A5, 2D, D0, 02	<6D	2060	DATA	5C, D0, 02, A9, 40, 48, A9, 00	<43
1290	DATA	C6, 2E, C6, 2D, 60, A5, 61, D0	<37	2070	DATA	8D, 6C, 16, A9, 20, 20, D2, FF	<3A
1300	DATA	02, C6, 62, C6, 61, 60, A5, 3B	<FA	2080	DATA	68, 20, D2, FF, A2, 00, BD, DB	<9E
1310	DATA	D0, 02, C6, 3C, C6, 3B, 60, 38	<D2	2090	DATA	10, 30, 1D, C9, 22, F0, 0D, 20	<9D
1320	DATA	AD, FF, 10, E5, 2D, AD, 00, 11	<00	2100	DATA	D2, FF, C9, 20, F0, 05, E8, E0	<C9
1330	DATA	E5, 2E, 60, A5, 61, CD, FF, 10	<4E	2110	DATA	0E, D0, EB, 60, 48, AD, 6C, 16	<87
1340	DATA	D0, 05, A5, 62, CD, 00, 11, 60	<C1	2120	DATA	49, FF, 8D, 6C, 16, 68, D0, E7	<F8
1350	DATA	18, A5, 2D, 6D, 07, 11, 85, 61	<24	2130	DATA	2C, 6C, 16, 30, DE, 8E, 6B, 16	<39
1360	DATA	A5, 2E, 69, 00, 85, 62, 60, 38	<85	2140	DATA	AA, 20, 41, 18, AE, 6B, 16, 4C	<89
1370	DATA	A5, 37, E5, 61, A6, 38, CA, 8A	<2D	2150	DATA	47, 16, 00, 00, 20, 79, 16, 20	<46
1380	DATA	E5, 62, 60, 38, AD, 01, 11, ED	<40	2160	DATA	24, 16, A9, DB, A2, 10, D0, 04	<8A
1390	DATA	FF, 10, 8D, 05, 11, AD, 02, 11	<5F	2170	DATA	A9, E9, A2, 10, 8D, 38, 16, 8E	<48
1400	DATA	ED, 00, 11, 8D, 06, 11, 60, A0	<DC	2180	DATA	39, 16, 60, 20, 84, 9D, 48, C9	<B1
1410	DATA	03, 20, A5, 04, 99, F7, 10, 88	<8A	2190	DATA	8D, F0, 17, C9, 89, F0, 13, C9	<0D
1420	DATA	10, F7, 60, A0, 03, 20, A5, 04	<E9	2200	DATA	CB, F0, 04, 68, 4C, A1, 94, 68	<D6
1430	DATA	99, FB, 10, 88, 10, F7, 60, 00	<A7	2210	DATA	20, 73, 04, C9, A4, D0, F5, A9	<55
1440	DATA	20, C6, 11, 8D, 30, 14, C8, 20	<07	2220	DATA	89, 48, C6, 65, D0, 07, 68, 20	<40
1450	DATA	A5, 04, 0D, 30, 14, 60, 20, 11	<3A	2230	DATA	6A, 17, 4C, 2D, 17, 20, AD, 11	<22
1460	DATA	11, 20, 31, 14, F0, A1, 20, 18	<61	2240	DATA	C9, 2C, F0, EE, 68, 60, 20, 73	<51
1470	DATA	14, 20, 88, 11, 20, 79, 04, C9	<DB	2250	DATA	04, F0, 30, 20, 6A, 17, A8, 90	<F9
1480	DATA	00, F0, 30, C9, 40, D0, 31, 20	<37	2260	DATA	1D, E0, 06, B0, 23, 48, 20, 1A	<F1
1490	DATA	28, 15, 20, F3, 14, 20, 90, 14	<7B	2270	DATA	11, 20, 73, 04, F0, 16, E0, 00	<1B
1500	DATA	F0, 26, 20, 4F, FF, 0D, 3F, 44	<3E	2280	DATA	D0, 0B, C9, 40, F0, 0E, C9, 5C	<28
1510	DATA	4F, 55, 42, 4C, 45, 20, 4C, 42	<EA	2290	DATA	F0, 0A, 20, 24, 11, 68, 20, 25	<89
1520	DATA	4C, 00, 20, DD, 14, A9, 2C, 20	<30	2300	DATA	8C, 4C, EC, 16, 20, 24, 11, 68	<2B
1530	DATA	D2, FF, 20, E6, 14, 20, 24, 16	<55	2310	DATA	20, 11, 17, 4C, DC, 8B, A2, FF	<6E
1540	DATA	4C, B1, 12, 20, 7F, 11, 90, B9	<D4	2320	DATA	86, 3A, 20, 5A, 88, 86, 3B, 84	<0B
1550	DATA	20, F3, 14, 4C, 42, 14, 00, A9	<2F	2330	DATA	3C, 20, 73, 04, AA, F0, 0B, 90	<06
1560	DATA	00, 8D, 8F, 14, 20, 31, 14, F0	<DF	2340	DATA	0C, 20, 53, 89, 20, 79, 04, 4C	<7A
1570	DATA	20, 20, 24, 14, 20, 88, 11, 20	<D0	2350	DATA	BA, 16, 4C, 0F, 87, 4C, 2E, 87	<DA
1580	DATA	79, 04, C9, 00, F0, 14, C9, 40	<DC	2360	DATA	2C, EB, 02, 10, 17, 24, 81, 10	<6C
1590	DATA	D0, 15, 20, 2E, 15, 2C, 8F, 14	<8D	2370	DATA	13, 48, 8A, 48, A9, 5B, 20, B2	<9E
1600	DATA	30, 13, 20, 66, 15, D0, 08, A0	<F9	2380	DATA	90, 20, 5B, A4, A9, 5D, 20, B2	<27
1610	DATA	80, 60, 20, 7F, 11, 90, D0, 20	<8C	2390	DATA	90, 68, AA, 68, 8A, 0A, A8, B9	<42
1620	DATA	FB, 14, 4C, 95, 14, 20, 6D, 16	<89	2400	DATA	4B, 17, 48, B9, 4A, 17, 48, 4C	<56
1630	DATA	20, E1, FF, D0, F2, 60, 20, 1A	<1C	2410	DATA	73, 04, 4C, 52, 57, 43, 54, 4F	<7B
1640	DATA	11, 20, 11, 11, A9, FF, 20, 92	<A9	2420	DATA	91, 40, 89, 8D, CB, 8A, 8C, D7	<79
1650	DATA	14, 4C, 24, 11, AD, F9, 10, AE	<94	2430	DATA	8B, 8C, 18, 75, 15, CE, 14, 61	<45
1660	DATA	FA, 10, 4C, EC, 14, AD, FD, 10	<F6	2440	DATA	19, 78, 17, 7E, 17, 83, 16, 6B	<9A
1670	DATA	AE, FE, 10, 85, 39, 86, 3A, 4C	<AF	2450	DATA	18, C2, 17, A3, 17, BB, 17, 8C	<53
1680	DATA	53, A4, AD, F7, 10, AC, F8, 10	<63	2460	DATA	17, F4, 17, 17, 18, 19, 1A, C2	<BC
1690	DATA	D0, 06, AD, FB, 10, AC, FC, 10	<D7	2470	DATA	C5, A2, 00, DD, 3B, 17, F0, 06	<95
1700	DATA	85, 3B, 84, 3C, 60, 00, 00, A5	<DA	2480	DATA	E8, E0, 10, D0, F6, 18, 60, 00	<A2
1710	DATA	3B, 48, A5, 3C, 48, 20, AD, 11	<2F	2490	DATA	D0, 06, A9, FF, D0, 07, F0, 03	<FC
1720	DATA	A5, 3B, A6, 3C, 8D, 06, 15, 8E	<5B	2500	DATA	4C, A1, 94, A9, 00, 8D, 78, 17	<5E
1730	DATA	07, 15, 68, 85, 3C, 68, 85, 3B	<41	2510	DATA	60, 4C, 8F, 8D, F0, 0F, 20, 9A	<0F
1740	DATA	A9, FF, 8D, 27, 15, 60, FF, A9	<31	2520	DATA	8A, 20, 79, 04, 20, C3, 17, 20	<3C
1750	DATA	DB, A2, 10, D0, 04, A9, E9, A2	<76	2530	DATA	20, 8D, 4C, DC, 8B, 4C, BC, 8B	<73
1760	DATA	10, 8D, 55, 15, 8E, 56, 15, 20	<59	2540	DATA	4C, A1, 94, A0, 05, 20, 05, 89	<96
1770	DATA	08, 15, EE, 27, 15, AE, 27, 15	<5B	2550	DATA	88, B9, 38, 00, 91, 7C, 88, D0	<65
1780	DATA	E0, 0E, B0, E1, 20, 73, 04, 48	<4F	2560	DATA	F8, A9, 8D, 91, 7C, 20, 79, 04	<AE
1790	DATA	20, 5A, 15, 90, 05, 68, A9, 20	<05	2570	DATA	4C, C3, 17, C9, A4, D0, C1, 20	<DD
				2580	DATA	73, 04, 90, 08, F0, 06, 08, C9	<4F

2590 DATA 5C, F0, 04, 28, 4C, 4D, 8D, 28	<AA	3040 DATA 60, A0, 00, 20, A5, 04, 91, 61	<22
2600 DATA 20, 1A, 11, 20, 28, 15, 20, 11	<E1	3050 DATA 20, 28, 19, 90, 23, 20, 5B, 19	<25
2610 DATA 11, 20, 90, 14, F0, 10, 20, 24	<E2	3060 DATA 20, 7F, 11, 4C, 34, 19, 38, AD	<5D
2620 DATA 11, AD, FD, 10, AE, FE, 10, 85	<79	3070 DATA 03, 11, ED, 05, 11, 8D, 03, 11	<AD
2630 DATA 14, 86, 15, 4C, 50, 8D, 20, 24	<B1	3080 DATA AD, 04, 11, ED, 06, 11, 8D, 04	<65
2640 DATA 11, 4C, 8F, 8D, F0, 1E, B0, 1C	<E9	3090 DATA 11, 60, E6, 61, D0, 02, E6, 62	<B1
2650 DATA 20, 1A, 11, 20, A7, 11, C9, 5C	<39	3100 DATA 60, F0, 03, 4C, A1, 94, 20, 7B	<93
2660 DATA D0, 0C, 20, 24, 11, 20, 79, 04	<70	3110 DATA 15, 20, FA, 11, 20, 11, 11, 20	<08
2670 DATA 20, 9A, 8C, 4C, AD, 11, 20, 24	<FE	3120 DATA 31, 14, D0, 0C, 20, 18, 88, 20	<E3
2680 DATA 11, 20, 79, 04, 4C, 9A, 8C, F0	<05	3130 DATA 4B, 88, 20, DF, 11, 4C, 7E, 86	<23
2690 DATA 24, 90, 06, 20, 3A, B4, 4C, A1	<82	3140 DATA 20, 85, 11, 20, 73, 04, C9, 40	<87
2700 DATA 94, 20, 1A, 11, 20, A7, 11, C9	<7E	3150 DATA F0, 4D, C9, 00, F0, 36, C9, 3A	<C4
2710 DATA 5C, D0, 0C, 20, 24, 11, 20, 79	<5A	3160 DATA F0, F1, C9, A7, F0, ED, C9, D5	<C0
2720 DATA 04, 20, 2B, B4, 4C, AD, 11, 20	<7A	3170 DATA F0, E9, C9, 22, F0, 2C, C9, 57	<B7
2730 DATA 24, 11, 20, 79, 04, 4C, 2B, B4	<4A	3180 DATA F0, 64, C9, 54, F0, 60, C9, 4F	<CC
2740 DATA A0, 81, 84, 23, A0, 8E, 84, 22	<29	3190 DATA F0, 5C, EA, 20, 73, 04, C9, 00	<02
2750 DATA A0, 00, CA, 10, 0F, B1, 22, 48	<E2	3200 DATA F0, 12, C9, 22, F0, 14, C9, 3A	<C4
2760 DATA E6, 22, D0, 02, E6, 23, 68, 10	<3B	3210 DATA F0, C9, C9, A7, F0, C5, C9, D5	<14
2770 DATA F4, 30, EF, C8, B1, 22, 30, 06	<DD	3220 DATA F0, C1, D0, E7, 20, 7F, 11, 4C	<1E
2780 DATA 20, B2, 90, D0, F6, 60, 29, 7F	<7D	3230 DATA 70, 19, 20, 73, 04, C9, 00, F0	<68
2790 DATA 4C, B2, 90, 2C, 78, 17, 10, 19	<E1	3240 DATA F3, C9, 22, D0, F5, F0, D3, 20	<0C
2800 DATA 20, 1A, 11, 20, B4, 4C, 13, A9, 3C	<92	3250 DATA 97, 11, 20, AD, 11, 20, 9A, 11	<9B
2810 DATA 20, B2, 90, 20, 2E, 15, 20, 6D	<0D	3260 DATA 20, 79, 04, C9, 00, D0, 0C, 20	<E4
2820 DATA 16, A9, 3E, 20, B2, 90, 20, 24	<3A	3270 DATA 12, 19, A0, 00, A9, 3A, 91, 61	<AE
2830 DATA 11, 4C, B0, 8D, D0, 03, 4C, A1	<45	3280 DATA 20, 11, 1A, 20, DC, 18, AD, 01	<8C
2840 DATA 94, 20, DE, B6, C9, 40, D0, 2C	<50	3290 DATA 11, AE, 02, 11, 85, 3B, 86, 3C	<5E
2850 DATA 20, 1A, 11, 20, 28, 15, 20, 11	<AE	3300 DATA 20, 79, 04, 4C, AF, 19, 20, 97	<54
2860 DATA 11, 20, 90, 14, 08, 20, 24, 11	<A2	3310 DATA 11, 20, 73, 04, F0, CF, D0, 9E	<FE
2870 DATA 20, AD, 11, 28, D0, 03, 4C, 8F	<23	3320 DATA EE, 01, 11, D0, 03, EE, 02, 11	<67
2880 DATA 8D, AD, FD, 10, AE, FE, 10, 85	<C8	3330 DATA 60, 20, 2C, 93, 20, 79, 04, C9	<F6
2890 DATA 14, 86, 15, 20, 3D, 8A, 20, ED	<C1	3340 DATA 89, F0, 05, A9, A7, 20, 93, 94	<56
2900 DATA AE, 4C, 02, 8B, C9, 5C, D0, C6	<0A	3350 DATA A5, 61, D0, 1F, 20, B0, 8D, A0	<95
2910 DATA 20, 1A, 11, 20, 2E, 15, 20, 01	<FD	3360 DATA 00, 20, A5, 04, F0, 0D, 20, 73	<24
2920 DATA 12, 20, 24, 11, 4C, AD, 11, 00	<E7	3370 DATA 04, C9, D5, D0, EF, 20, 73, 04	<15
2930 DATA 00, 00, 00, 20, FE, 18, AD, 06	<07	3380 DATA 4C, 4C, 1A, 20, C1, 8D, D0, 03	<5E
2940 DATA 11, F0, 03, 4C, 18, 13, 20, 12	<D1	3390 DATA 4C, B3, 8D, 20, 79, 04, B0, 03	<F4
2950 DATA 19, 20, 1D, 19, 20, 32, 19, 20	<82	3400 DATA 4C, C3, 17, 08, 48, C9, 5C, D0	<36
2960 DATA 47, 19, AD, 03, 11, AC, 04, 11	<2F	3410 DATA 05, 68, 28, 4C, C3, 17, 68, 28	<F8
2970 DATA 85, 3B, 84, 3C, 60, 38, AD, 03	<4D	3420 DATA F0, 2D, 20, 6A, 17, A8, 90, 1D	<2B
2980 DATA 11, ED, 01, 11, 8D, 05, 11, AD	<6C	3430 DATA E0, 06, B0, 20, 48, 20, 1A, 11	<D4
2990 DATA 04, 11, ED, 02, 11, 8D, 06, 11	<EC	3440 DATA 20, 73, 04, F0, 13, E0, 00, D0	<4D
3000 DATA 60, AD, 01, 11, 85, 61, AD, 02	<D7	3450 DATA 08, C9, 40, F0, 0B, C9, 5C, F0	<0F
3010 DATA 11, 85, 62, 60, AD, 03, 11, 85	<5C	3460 DATA 07, 20, 24, 11, 68, 4C, 25, 8C	<B7
3020 DATA 3B, AD, 04, 11, 85, 3C, 60, 38	<55	3470 DATA 20, 24, 11, 68, 4C, 11, 17, 60	<77
3030 DATA A5, 2D, E5, 3B, A5, 2E, E5, 3C	<F0	3480 DATA 41, 45, 49, 4F, 55, 00, 00, 00	<A6

Két cica

Néhány éve az Élet és Tudomány Herczeg-Bizám: „A gondolkodás iskolája” rovatában olvashattuk O'Brain amerikai professzor játékát.

Egy 5-ször 5-ös játémezőn két cica bújt el, ezeket kell megkeresnünk. Ha nem találtuk el egyik cica helyét sem, akkor a kérdező – jelen esetben a gép – megadja valamelyik távolságát (ha már az egyiket megtaláltuk, akkor természetesen a következőkben mindig a másikat).

A találgatás addig ismétlődik, amíg nincs meg mindkét cica.

A program Label Basic nyelven íródott, de aki nek nincs türelme begépelnie azt, szintén könnyen átírhatja a programot – a megfelelő sorszámok megadásával.

Írta: **Koszper Vilmos**

10 REM 2 CICA (LABEL BASIC NYELVEN)	<01
20 :	<55
30 COLOR 1,7,7:COLOR 0,1:COLOR 4,6,5	<90
40 :	<B6
50 VOL 5	<AB
60 :	<18
70 AX=0:AY=0:REM AZ EGYIK ELREJTETT CICA HELYE	<64
80 BX=0:BY=0:REM A MASIK ELREJTETT CICA HELYE	<E1
90 VP=0:REM VALOSZINUSEGOSZTO. MINDEN KERDEZGETES ELOTT KIJELOLI,HOGY A	<6E
100 REM VALASZOK MILYEN VALOSZINUSEGGE L VONATKOZNAK MAJD AZ EGYIK CICARA S A	<DF
110 REM MASIKRA	<C5
120 :	<78
130 DIM TC(5,5):REM CICA TABLAZAT	<6B
140 REM 1=AZ "A" CICA	<BF
150 REM 2=A "B" CICA	<3C
160 REM 10=EGY KERDES	<10
170 REM 20=KET KERDES	<E1
180 REM ES IGY TOVABB	<DD
190 :	<BD
200 VL=0:REM A VALASZOLAS LEHETOSEGE	<BC
210 REM VL=0 --> BARMELYIK CICARA SZAB AD VALASZOLNI	<30
220 REM VL=1 --> AZ "A" CICARA NEM SZABAD VALSZOLNI,MIVEL MAR MEG VAN OL DVA	<E9

```

230 REM VL=2 --> A "B" CICARA NEM SZAB <04
    AD VALASZOLNI,MIVEL MEG VAN OLDVA
240 REM VL=3 --> VEGE ENNEK A PARTINAK <C1
250 W$="[C=/I]":WF$="[CTRL/9][C=/I][C <C0
    TRL/0]":WB$="[C=/K]":WJ$="[CTRL/9]
    [C=/K][CTRL/0]"
260 W1$="[CTRL/9][C=/D][CTRL/0]" <95
270 W2$="[CTRL/9][C=/F][CTRL/0]" <2E
280 W3$="[CTRL/9][C=/C][CTRL/0]" <80
290 W4$="[CTRL/9][C=/V][CTRL/0]" <D2
300 WQ$="[DOWN][3LEFT]" <C7
310 EW$=W1$+WF$+W2$+WQ$+WB$ <9C
320 UW$=WJ$+WQ$+W3$+WA$+W4$ <7F
330 UR$=EW$+" "+UW$ <1B
340 CC$=EW$+"*" +UW$ <AA
350 : <07
360 REMUR$="...[DOWN][3LEFT]. .[DOWN][ <18
    3LEFT]..."
370 REM CC$="...[DOWN][3LEFT].*.[DOWN] <06
    [3LEFT]..."
380 : <49
390 DIM BE$(5) <11
400 FOR J=1 TO 5 <50
410 BE$(J)="[3SPC][DOWN][3LEFT]" +CHR$ <36
    (64+J)+" [DOWN][3LEFT][3SPC]"
420 NEXT J <50
430 DIM SZ$(5) <C2
440 FOR J=1 TO 5 <55
450 SZ$(J)="[3SPC][DOWN][3LEFT]" +CHR$ <B0
    (48+J)+" [DOWN][3LEFT][3SPC]"
460 NEXT J <53
470 : <3F
480 NR=1:REM A JATEK SORSZAMA <0A
490 R0=10000:REM ALSO REKORD <C7
500 R9=0:REM FELSO REKORD <20
510 : <90
520 GOTO #MAIN <BF
530 : <12
540 @KEP <78
550 FOR J1=1 TO 5 <B9
560 FOR J2=1 TO 5 <F8
570 IF J2=AX AND J1=AY THEN #K1 <17
580 IF J2=BX AND J1=BY THEN #K2 <03
590 @K3:IF TC(J2,J1)=0 THEN #K4 <63
600 WW$=EW$+CHR$(TC(J2,J1)+48)+UW$ <07
610 GOTO #K5 <6F
620 @K1:IF NOT (VL=1 OR VL=3) THEN #K3 <BA
630 @K6:WW$=CC$ <F5
640 GOTO #K5 <91
650 @K2:IF NOT (VL=2 OR VL=3) THEN #K3 <4D
660 GOTO #K6 <F3
670 @K4:WW$=UR$ <E1
680 @K5 <49
690 CHAR 1,J2*3,(6-J1)*3,WW$ <04
700 NEXT J2 <CE
710 CHAR 1,18,(6-J1)*3,SZ$(J1) <83
720 NEXT J1 <FF
730 FOR J2=1 TO 5 <62
740 CHAR 1,J2*3,18,BE$(J2) <97
750 NEXT J2 <11
760 GOSUB #LEPSZ.IR <6C
770 RETURN <16
780 : <E2
790 @CELHELVEZES <F1
800 FOR J1=1 TO 5 <F6
810 FOR J2=1 TO 5 <39
820 TC(J1,J2)=0 <1A
830 NEXT J2 <46
840 NEXT J1 <16
850 @CELHELY1 <15
860 AX=INT(RND(8)*5+1) <80
870 AY=INT(RND(8)*5+1) <B7
880 BX=INT(RND(8)*5+1) <C6
890 BY=INT(RND(8)*5+1) <F5
900 IF AX=BX AND AY=BY THEN #CELHELY1 <FA
910 IF VP<>0 THEN #CELH2 <3E
920 VP=RND(8) <70
930 @CELH2:Q$=STR$(VP)+"000" <ED
940 LE=LEN(Q$):Q$=RIGHT$(Q$,LE-1) <42
950 Q$="0"+LEFT$(Q$,4) <A4
960 CHAR 1,25,8,"VALOSZ:"+Q$ <76
970 VL=0:REM MELYIK CICAT TALALTA MEG <A2
980 LS=0:REM A LEPES SORSZAMA <EE
990 RETURN <24
1000 : <DE
1010 @INPUT <0A
1020 GG=28 <D2
1030 CHAR 1,0,22,"[36SPC]" <AD
1040 CHAR 1,3,22,"HOL A CICA? (BETU, SZ <12
    AM)"
1050 DO <13
1060 GET Q$ <F3
1070 IF Q$<>" ". THEN CHAR 1,GG,22,Q$:EXI <34
    T
1080 VILL= NOT VILL <33
1090 IF VILL THEN CHAR 1,GG,22,"[CTRL/9 <15
    ] [CTRL/0]":ELSE CHAR 1,GG,22," "
1100 LOOP <02
1110 IF Q$<"A" OR Q$>"E" THEN #INPUT <F2
1120 IB=ASC(Q$)-64:IB$=Q$ <C4
1130 DO <D8
1140 GET Q$ <37
1150 IF Q$<>" " THEN CHAR 1,GG+2,22,Q$:E <71
    XIT
1160 VILL= NOT VILL <74
1170 IF VILL THEN CHAR 1,GG+2,22,"[CTRL <42
    /9] [CTRL/0]":ELSE CHAR 1,GG+2,22,
    " "
1180 LOOP <89
1190 IF Q$<"1" OR Q$>"5" THEN #INPUT <6B
1200 IS=ASC(Q$)-48:IS$=Q$ <C7
1210 RETURN <8A
1220 : <6D
1230 @LEPSZ.IR <83
1240 CHAR 1,21,1,"LEPESSZAM:" <99
1250 PRINT LS <62
1260 RETURN <9E
1270 : <A1
1280 @JATEK <8A
1290 CHAR 1,1,1,"JATEK:" <F5
1300 PRINT NR; <B2
1310 CHAR 1,25,3,"MAX:" <38
1320 PRINT R9; <54
1330 CHAR 1,25,5,"MIN:" <B8
1340 PRINT R0; <C0
1350 RETURN <34
1360 : <36
1370 : <07
1380 @MAIN:SCNCLR :PRINT :PRINT :PRINT <91
    :PRINT
1390 PRINT "[3SPC]VALOSZINUSEG 0.500 LE <55
    GYEN.....F"
1400 PRINT :PRINT "[3SPC]VALOSZINUSEG T <D8
    ETSZOLEGES (GEPI)..G"
1410 GET KEY Q$:IF Q$="G" THEN VP=0:GOT <C0
    0 1440#MAIN1
1420 IF Q$="F" THEN VP=.5:GOTO 1440#MAI <61
    N1
1430 GOTO #MAIN <BE
1440 @MAIN1:SCNCLR :GOSUB #JATEK <3F
1450 GOSUB #CELHELVEZES <44
1460 @KEZD:GOSUB #KEP <03
1470 IF VL=3 THEN #UJJATEK <5F
1480 GOSUB #INPUT <06
1490 GOSUB #TJ TO RL <D0
1500 LS=LS+1 <1E
1510 @L1:IF AX=IB AND AY=IS THEN #L3 <B0
1520 IF BX=IB AND BY=IS THEN #L4 <8F
1530 IF VL=1 OR VL=3 THEN #L6 <44
1540 IF VL=2 THEN #L8 <C4
1550 P=RND(8) <BE
1560 IF P>VP THEN #L8:ELSE #L6 <1F
1570 @L3:IF VL=1 OR VL=3 THEN #L6 <B6
1580 VL=VL OR 1 <F8
1590 @L9:GOSUB #TALALATJELZES <DE
1600 TA=0 <C8
1610 GOTO #L99 <59

```



```

1620 @L4:IF VL=2 OR VL=3 THEN #L8
1630 VL=VL OR 2
1640 GOTO 1590#L9
1650 @L6:GOSUB #TAVII.
1660 GOTO #L99
1670 @L8:GOSUB #TAVI.
1680 @L99
1690 TC( IB, IS)=TA
1700 GOSUB #TAVOL 'SAG
1710 GOTO #KEZD
1720 @TALALATJELZES
1730 CHAR 1,22,10,"[CTRL/9]ITT A CICA!["
CTRL/0]"
1740 RETURN
1750 :
1760 @TJ TO RL
1770 CHAR 1,22,10,"[12SPC]"
1780 RETURN
1790 :
1800 @UJJATEK

```

```

<B1 1810 : <22
<ED 1820 CHAR 1,21,14,"[4SPC]UJ JATEK?" <14
<E2 1830 SOUND 1,700,5:SOUND 1,600,5 <97
<A5 1840 GET KEY Q$ <FF
<4D 1850 IF R0>LS THEN R0=LS <B2
<19 1860 IF R9<LS THEN R9=LS <40
<AF 1870 NR=NR+1 <B9
<89 1880 GOTO #MAIN <CE
<AC 1890 @TAVOL SAG <53
<F4 1900 CHAR 1,21,12,"TIPP: "+IB$+" "+IS$ <A4
<73 1910 CHAR 1,21,14,"TAVOLSAG:"+STR$(TA)+ <86
<7F " "
1920 RETURN <48
<7F 1930 @TAVI. <78
<FF 1940 TA=ABS(AX-IB)+ABS(AY-IS) <62
<7E 1950 RETURN <DA
<76 1960 : <5C
<20 1970 @TAVII. <6D
<02 1980 TA=ABS(BX-IB)+ABS(BY-IS) <FC
<58 1990 RETURN <9F

```

FLOPPY-POSTA A programok lemezen is megrendelhetők!

PC - AKCIÓ !

Gépvásárlási lehetőség 10 havi kamatmentes részletfizetésre!

Vásárlói csoportokat szervezünk a következő kategóriákban:

- I. AT számítógép monitorral, már 4.300,- Ft-os havi részletre
- II. 25 Mhz-es 286-os, 40 MB HDD, 14"-os mono VGA monitorral csak 7.100,- Ft/hó
- III. 386-os professzionális konfiguráció, Super VGA monitorral 9.900,- Ft/hó

Lehetőség egyéni kiegészítésekre!
12 hónap garancia!

Áraink az ÁFA-t már tartalmazzák!



További információ:
PC VÁSÁRLÓI KLUB
1149 Budapest,
Nagy L. kir. út 141.
Tel.: 183-1982



Rubin

APARTMAN-HOTEL

BUDAPEST

MÁR A KÜLSEJE IS ÉLMÉNY...

...a Rubin Apartman-Hotelnek, mely az utóbbi évek egyik legeredetibb építészeti alkotása.

DE HA BELÉP,

kényelmes, otthonos szálloda.
Biopton gyógycentrum, kozmetika
manager-club
kitűnő étterem,
söröző
1 korsó Gösser sör 100 Ft
vegetáriánustál 210 Ft
borjú bécsi 220 Ft
erőleves 60 Ft

bowling tekepálya 300 Ft/óra
biliárdszoba 300 Ft/óra
Fitness-center uszodával
100 Ft/óra v. 250 Ft/nap
uszoda (B épület)
70 Ft/óra v. 200 Ft/nap
teniszpálya 200 Ft/óra
mini-ABC várja Önt



H-1118 Budapest XI., Dayka Gábor u. 3.

Telefon: 166-6811, 269-8240
Telefax: 166-5193 Telex: 22-2215

APARTMAN KFT.

Dator

A következő program egy Basic-bővítésként működő szuper data-maker. Futtatása után létrehoz a lemezen egy gépi kódú file-t, amit ezután bármikor használhatunk.

A program által implementált új utasítás, a * Toolkit jellegű, tehát csak parancs módban használható. Szintaxisa: * A,B,C,D. A paraméterek a következők: A – kezdőcím, B – végcím, C – ha nulla, akkor a ROM-ból olvas, ha nem, akkor a RAM-ból, D – ha nulla, akkor decimális számokat ír, ha nem, akkor hexát. A paraméterek helyén tetszőleges numerikus kifejezés állhat. A lehetséges hibaüzenetek:

(?Break error után ...)

S*? – a datásítást a Stop billentyűvel megszakították;

(?Out of memory után ...)

A*? – nincs elegendő Basic munkaterület;

B*? – túl nagy az utolsó sorszám;

C*? – elfogyott a Basic munkaterület (230 és 255) között;

D*? – elérte a legnagyobb sorszámot.

A data sorok a program végéhez kapcsolódnak. Az új utasítás kiiktatása: SYS 4115, illetve ismét üzembe helyezése: SYS 4112.

Írta: **Koszper Vilmos**

```

10 OPEN 1,8,2,"DATOR,PRG,W" <39
20 PRINT#1,CHR$(1);CHR$(16); <3B
30 FOR X=4097 TO 4860 <AB
40 READ A$:A=DEC(A$) <3E
50 PRINT#1,CHR$(A);:E=E+A <01
60 NEXT X <0E
70 CLOSE 1 <7B
80 IF E<>70673 THEN PRINT "HIBA":END <41
90 DLOAD "DATOR" <B2
100 DATA 0B,10,01,00,9E,34,31,30 <F5
110 DATA 39,00,00,00,4C,16,10,4C <56
120 DATA 2D,10,4C,44,10,38,A9,00 <D5
130 DATA 8D,FC,12,8D,FD,12,8D,FE <A6
140 DATA 12,A9,FD,A2,12,85,2B,86 <EC
150 DATA 2C,20,7B,8A,A9,D0,A2,11 <5D
160 DATA 8D,08,03,8E,09,03,A9,DC <31
170 DATA A2,11,8D,02,03,8E,03,03 <98
180 DATA 4C,7E,86,A9,D6,A2,8B,8D <75
190 DATA 08,03,8E,09,03,A9,12,A2 <EE
200 DATA 87,D0,E7,00,00,00,00,00 <E9
210 DATA 00,00,00,00,00,00,00,A0 <A8
220 DATA 00,2C,5A,10,10,04,78,8D <BD
230 DATA 3F,FF,B1,3B,8D,3E,FF,58 <F8
240 DATA E6,3B,D0,02,E6,3C,29,FF <0E
250 DATA 60,3B,AD,5E,10,E5,3B,AD <DF
260 DATA 5F,10,E5,3C,60,AE,5E,10 <08
270 DATA AD,5F,10,4C,9F,10,20,60 <39
280 DATA 10,20,74,CD,4C,A2,10,00 <E0

```

```

290 DATA AE,5C,10,AD,5D,10,20,5F <7D
300 DATA A4,A0,00,8C,98,10,AC,98 <5C
310 DATA 10,B9,00,01,F0,08,20,4C <FA
320 DATA 11,EE,98,10,D0,F0,60,2C <D8
330 DATA 5B,10,30,03,4C,99,10,AD <C1
340 DATA 5D,10,20,CF,10,AD,5C,10 <E3
350 DATA 4C,CF,10,20,60,10,48,20 <50
360 DATA 10,FB,68,20,20,FB,20,4C <A9
370 DATA 11,8A,20,4C,11,60,2C,5B <27
380 DATA 10,30,E8,4C,8F,10,2C,5B <40
390 DATA 10,30,03,4C,86,10,AD,5F <9A
400 DATA 10,20,CF,10,AD,5E,10,4C <13
410 DATA CF,10,20,18,88,A0,00,8C <7A
420 DATA 58,10,8C,59,10,A5,2B,A6 <BE
430 DATA 2C,85,62,86,63,C8,20,30 <B6
440 DATA 11,D0,01,60,8D,54,10,C8 <E1
450 DATA 20,30,11,8D,58,10,C8,20 <E1
460 DATA 30,11,8D,59,10,A0,00,20 <ED
470 DATA 30,11,AE,54,10,D0,DA,78 <83
480 DATA 8D,3F,FF,B1,62,8D,3E,FF <DD
490 DATA 58,60,A5,37,E5,2D,A5,38 <C2
500 DATA E5,2E,60,A9,20,2C,A9,2C <22
510 DATA 2C,A9,00,A0,00,91,5F,E6 <A8
520 DATA 5F,D0,02,E6,60,8D,2D,D0 <D3
530 DATA 02,E6,2E,60,EE,58,10,D0 <A7
540 DATA 03,EE,59,10,60,20,18,88 <AD
550 DATA A5,22,A6,23,85,5F,86,60 <51
560 DATA 20,8E,11,A9,8F,20,4C,11 <4B
570 DATA 20,44,11,20,B8,10,A9,2D <F0
580 DATA 20,4C,11,20,D2,FF,20,E7 <09
590 DATA 10,20,4A,11,60,20,5D,11 <23
600 DATA A9,01,20,4C,11,20,4C,11 <B5
610 DATA AD,58,10,20,4C,11,AD,59 <36
620 DATA 10,20,4C,11,20,3E,90,60 <64
630 DATA 20,8E,11,A9,83,20,4C,11 <7E
640 DATA 20,44,11,A9,08,8D,54,10 <55
650 DATA 20,DF,10,20,EB,12,F0,0B <8E
660 DATA CE,54,10,F0,06,20,47,11 <D5
670 DATA 4C,B9,11,20,4A,11,60,20 <6E
680 DATA 73,04,08,C9,AC,F0,23,28 <95
690 DATA 4C,D9,8B,A2,FF,86,3A,20 <4B
700 DATA 5A,88,86,3B,84,3C,20,73 <68
710 DATA 04,AA,F0,EF,B0,03,4C,2E <E1
720 DATA 87,20,53,89,20,79,04,4C <8B
730 DATA D3,11,28,20,02,12,4C,DC <5B
740 DATA 8B,A9,12,48,A9,0A,48,4C <7C
750 DATA 73,04,F0,30,20,E1,9D,8C <02
760 DATA 5C,10,8D,5D,10,20,79,04 <5D
770 DATA F0,22,20,DE,9D,8C,5E,10 <8F
780 DATA 8D,5F,10,20,79,04,F0,14 <17
790 DATA 20,91,94,20,84,9D,8E,5A <7B
800 DATA 10,F0,09,20,91,94,20,84 <83
810 DATA 9D,8E,5B,10,20,DE,B6,A5 <8B
820 DATA 3B,A6,3C,8D,56,10,8E,57 <3C
830 DATA 10,AD,5C,10,AE,5D,10,85 <F1
840 DATA 3B,86,3C,A2,FF,AD,5A,10 <2C
850 DATA F0,03,8E,5A,10,AD,5B,10 <E1
860 DATA F0,03,8E,5B,10,20,3B,11 <82
870 DATA F0,41,20,FB,10,20,F0,12 <04
880 DATA B0,3C,20,66,11,20,3B,11 <EB
890 DATA F0,37,20,F0,12,B0,35,20 <62
900 DATA A9,11,20,E1,FF,D0,11,20 <38
910 DATA 4F,FF,0D,53,2A,3F,00,20 <F3
920 DATA D4,12,20,9A,8A,4C,D2,8C <B6
930 DATA 20,EB,12,F0,05,20,7A,10 <60
940 DATA B0,D3,20,D4,12,20,9A,8A <87
950 DATA 4C,7E,86,A9,41,2C,A9,42 <75
960 DATA 2C,A9,43,2C,A9,44,8D,98 <43
970 DATA 10,A9,0D,20,D2,FF,AD,98 <59
980 DATA 10,20,D2,FF,20,4F,FF,2A <5C
990 DATA 3F,00,20,D4,12,20,9A,8A <5C
1000 DATA 4C,81,86,20,4A,11,20,4A <26
1010 DATA 11,20,18,88,20,4B,88,AD <F0
1020 DATA 56,10,AE,57,10,85,3B,86 <36
1030 DATA 3C,60,A5,3B,05,3C,60,38 <6C
1040 DATA AD,58,10,E9,FE,AD,59,10 <35
1050 DATA E9,F9,60,00,00,00,00,00 <25

```

Disill

Ha Plus/4-es gépünkön lefuttatjuk a következő programot, akkor egyetlen Basic utasítással egy disassemblert hívhatunk meg. Ennek érdekessége, hogy a nem szabványos – úgynevezett illegális – kódok nagy részét is képes megjeleníteni a többihez hasonlóan, hárombetűs mnemonikonok formájában. A disill hívása: / A,B,C,D. Az A és a B paraméterek a kezdő- és végcímek. Ha C=0, akkor a ROM-ból olvas, különben a RAM-ból. Ha D=0 akkor nem fordítja az illegális kódokat, a szokásos módon ???-et ír ki. Mindegyik paraméter helyén tetszőleges numerikus kifejezés állhat.

Az illegális kódok mnemonikonjai:

NOP, SBC – ugyanazt csinálják, mint szabványos megfelelőik;

CRA – (crash) leállítja a mikroprocesszort;

JR1 – átugorja az utána következő bájtot;

JR2 – átugorja az utána következő két bájtot;

LAX – LDA és LDX egymás után;

DCP – DEC és CMP;

ISC – INC és SBC;

RLA – ROL és AND;

RRA – ROR és ADC;

SLO – ASL és ORA;

SRE – LSR és EOR;

ANR – AND# és LSR A;

XAN – TXA és AND#.

Az előbbi utasításokban a flagek az elvárható módon változnak.

ANC – AND, majd C az N állapotát veszi fel;

XSB – az AND X értékét X-be teszi, majd kivonja belőle az operandust. Ez ismét az X-be kerül. A kivonás során a processzor C-t nem veszi figyelembe!

ANX – Az A AND X értéke az operandus címre kerül.

Írta: **Koszper Vilmos**

```

10 IF X<>0 GOTO 110 <DD
20 OPEN 1,8,2,"DISILL,PRG,W" <D8
30 PRINT#1,CHR$(34);CHR$(34); <08
40 FOR X=8738 TO 10377 <27
50 READ A$:A=DEC(A$) <4F
60 PRINT#1,CHR$(A);:E=E+A <52
70 NEXT X <BF
80 CLOSE 1 <0C
90 IF E<>132850 THEN PRINT "HIBA":END <29
100 LOAD "DISILL",8,1 <E0
110 SYS DEC("2222") <94
120 DATA 4C,2B,22,4C,56,22,4C,6D <8A
130 DATA 22,38,A9,00,E9,22,A9,40 <EC
140 DATA E9,22,90,15,A9,00,8D,89 <8C
150 DATA 28,8D,8A,28,8D,8B,28,A9 <14
160 DATA 8A,A2,28,85,2B,86,2C,D0 <02
170 DATA 08,A9,21,A2,22,85,37,86 <F8

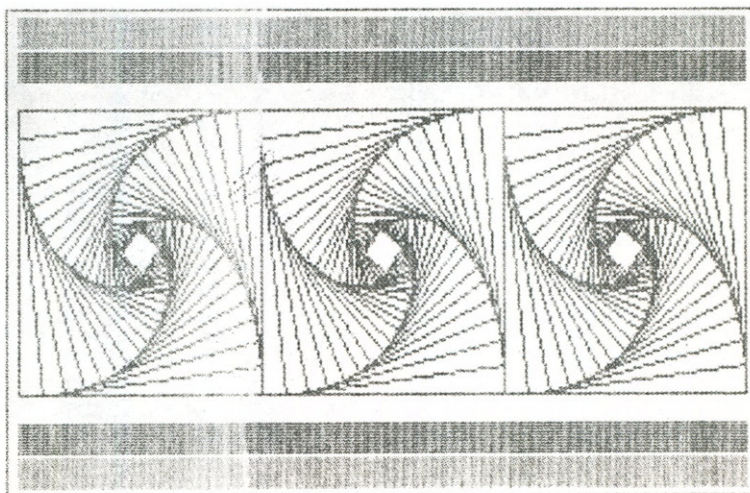
```

```

180 DATA 38,20,7B,8A,A9,19,A2,28 <EE
190 DATA 8D,08,03,8E,09,03,A9,25 <9B
200 DATA A2,28,8D,02,03,8E,03,03 <A1
210 DATA 4C,7E,86,A9,D6,A2,8B,8D <42
220 DATA 08,03,8E,09,03,A9,12,A2 <D9
230 DATA 87,D0,E7,80,FF,00,D0,00 <E9
240 DATA D3,00,00,00,00,00,00,00 <D9
250 DATA 00,00,80,8C,00,0C,00,87 <BB
260 DATA 87,07,80,86,81,06,00,83 <CB
270 DATA 83,03,82,8B,00,0B,00,88 <1D
280 DATA 88,08,80,85,00,05,00,84 <01
290 DATA 84,04,83,8C,00,0C,87,87 <40
300 DATA 87,07,80,86,81,06,83,83 <04
310 DATA 83,03,82,8B,00,0B,00,88 <2F
320 DATA 88,08,80,85,00,05,00,84 <53
330 DATA 84,04,80,8C,00,0C,00,87 <99
340 DATA 87,07,80,86,81,06,83,83 <01
350 DATA 83,03,82,8B,00,0B,00,88 <28
360 DATA 88,08,80,85,00,05,00,84 <5C
370 DATA 84,04,80,8C,00,0C,00,87 <9A
380 DATA 87,07,80,86,81,00,8A,83 <73
390 DATA 83,03,82,8B,00,0B,00,88 <35
400 DATA 88,08,80,85,00,05,00,84 <AE
410 DATA 84,04,00,8C,00,0C,87,87 <15
420 DATA 87,07,80,00,80,06,83,83 <FC
430 DATA 83,03,82,8B,00,00,88,88 <E7
440 DATA 89,09,80,85,80,00,00,84 <E7
450 DATA 00,00,86,8C,86,0C,87,87 <4D
460 DATA 87,07,80,86,80,00,83,83 <87
470 DATA 83,03,82,8B,00,0B,88,88 <06
480 DATA 89,09,80,85,80,00,84,84 <1D
490 DATA 85,05,86,8C,00,0C,87,87 <11
500 DATA 87,07,80,86,80,06,83,83 <D8
510 DATA 83,03,82,8B,00,0B,00,88 <92
520 DATA 88,08,80,85,00,05,00,84 <E6
530 DATA 84,04,86,8C,00,0C,87,87 <91
540 DATA 87,07,80,86,80,06,83,83 <C5
550 DATA 83,03,82,8B,00,0B,00,88 <9F
560 DATA 88,08,80,85,00,05,00,84 <E3
570 DATA 84,04,0E,2C,18,3B,23,2C <C2
580 DATA 06,3B,2E,2C,06,02,24,2C <84
590 DATA 06,3B,0D,2C,18,3B,23,2C <09
600 DATA 06,3B,11,2C,2B,3B,24,2C <FB
610 DATA 06,3B,25,03,18,31,0A,03 <AD
620 DATA 32,31,30,03,32,02,0A,03 <4A
630 DATA 32,31,0B,03,18,31,23,03 <87
640 DATA 32,31,38,03,2B,31,24,03 <B9
650 DATA 32,31,35,1D,18,3C,23,1D <3D
660 DATA 2A,3C,2D,1D,2A,04,22,1D <39
670 DATA 2A,3C,0F,1D,18,3C,23,1D <59
680 DATA 2A,3C,13,1D,2B,3C,24,1D <79
690 DATA 2A,3C,36,01,18,34,23,01 <C7
700 DATA 33,34,2F,01,33,00,22,01 <E8
710 DATA 33,34,10,01,18,34,23,01 <BD
720 DATA 33,34,3A,01,2B,34,24,01 <1F
730 DATA 33,34,23,3D,23,05,3F,3D <FE
740 DATA 3E,05,1C,23,43,46,3F,3D <20
750 DATA 3E,05,07,3D,18,00,3F,3D <C4
760 DATA 3E,05,45,3D,44,00,00,3D <DF
770 DATA 00,00,29,27,28,26,29,27 <47
780 DATA 28,26,41,27,40,00,29,27 <D0
790 DATA 28,26,08,27,18,26,29,27 <5E
800 DATA 28,26,14,27,42,00,29,27 <BD
810 DATA 28,26,17,15,23,19,17,15 <2E
820 DATA 1A,19,20,15,1B,47,17,15 <B6
830 DATA 1A,19,0C,15,18,19,23,15 <08
840 DATA 1A,19,12,15,2B,19,24,15 <4A
850 DATA 1A,19,16,37,23,21,16,37 <86
860 DATA 1E,21,1F,37,2B,37,16,37 <58
870 DATA 1E,21,09,37,18,21,23,37 <40
880 DATA 1E,21,39,37,2B,21,24,37 <2F
890 DATA 1E,21,3F,3F,3F,41,44,43 <CB
900 DATA 41,4E,43,41,4E,44,41,4E <9B
910 DATA 52,41,4E,58,41,53,4C,42 <DC
920 DATA 43,43,42,43,53,42,45,51 <38
930 DATA 42,49,54,42,4D,49,42,4E <64
940 DATA 45,42,50,4C,42,52,4B,42 <83
950 DATA 56,43,42,56,53,43,4C,43 <D7
960 DATA 43,4C,44,43,4C,49,43,4C <BF

```

970 DATA	56, 43, 4D, 50, 43, 50, 58, 43	<2E	1570 DATA	0A, 18, 6D, 87, 22, A8, A9, 02	<48
980 DATA	50, 59, 43, 52, 41, 44, 43, 50	<E4	1580 DATA	8D, C8, 25, B9, 8C, 24, 20, D2	<DB
990 DATA	44, 45, 43, 44, 45, 58, 44, 45	<01	1590 DATA	FF, C8, CE, C8, 25, 10, F4, 20	<F1
1000 DATA	59, 45, 4F, 52, 49, 4E, 43, 49	<89	1600 DATA	C3, 25, AD, 85, 22, D0, 22, 20	<1F
1010 DATA	4E, 58, 49, 4E, 59, 49, 53, 43	<81	1610 DATA	C3, 25, 18, A5, 3B, 6D, 86, 22	<E6
1020 DATA	4A, 4D, 50, 4A, 52, 31, 4A, 52	<26	1620 DATA	85, 3B, 90, 02, E6, 3C, 20, 0D	<0D
1030 DATA	32, 4A, 53, 52, 4C, 41, 58, 4C	<A6	1630 DATA	28, 90, 03, 4C, 03, 26, AD, 8A	<D2
1040 DATA	44, 41, 4C, 44, 58, 4C, 44, 59	<8C	1640 DATA	22, AE, 8B, 22, 85, 3B, 86, 3C	<5A
1050 DATA	4C, 53, 52, 4E, 4F, 50, 4F, 52	<C4	1650 DATA	60, C9, 01, D0, 09, 20, A5, 25	<DB
1060 DATA	41, 50, 48, 41, 50, 48, 50, 50	<1E	1660 DATA	20, C3, 25, 4C, C9, 26, C9, 02	<62
1070 DATA	4C, 41, 50, 4C, 50, 52, 4C, 41	<FC	1670 DATA	D0, 4B, 18, A5, 3B, 69, 02, 8D	<AE
1080 DATA	52, 4F, 4C, 52, 4F, 52, 52, 52	<63	1680 DATA	88, 22, A5, 3C, 69, 00, 8D, 89	<3B
1090 DATA	41, 52, 54, 49, 52, 54, 53, 53	<BD	1690 DATA	22, A0, 01, 20, 92, 25, 30, 12	<E2
1100 DATA	42, 43, 53, 45, 43, 53, 45, 44	<5D	1700 DATA	18, 6D, 88, 22, 8D, 88, 22, AD	<71
1110 DATA	53, 45, 49, 53, 4C, 4F, 53, 52	<9A	1710 DATA	89, 22, 69, 00, 8D, 89, 22, 4C	<CB
1120 DATA	45, 53, 54, 41, 53, 54, 58, 53	<06	1720 DATA	3E, 27, 49, FF, 8D, 87, 22, EE	<1F
1130 DATA	54, 59, 54, 41, 58, 54, 41, 59	<DC	1730 DATA	87, 22, 38, AD, 88, 22, ED, 87	<29
1140 DATA	54, 53, 58, 54, 58, 41, 54, 58	<68	1740 DATA	22, 8D, 88, 22, AD, 89, 22, E9	<5E
1150 DATA	53, 54, 59, 41, 58, 41, 4E, 58	<E1	1750 DATA	00, 8D, 89, 22, 20, EB, 27, 20	<98
1160 DATA	53, 42, 5C, 5C, 5C, 5C, 5C, 5C	<50	1760 DATA	C3, 25, 4C, C9, 26, C9, 03, D0	<FD
1170 DATA	5C, 5C, 5C, 5C, 5C, 5C, 5C, 5C	<17	1770 DATA	06, 20, DA, 27, 4C, 3E, 27, C9	<49
1180 DATA	5C, 5C, 5C, 5C, 5C, 5C, 5C, 5C	<C7	1780 DATA	04, D0, 09, 20, CA, 27, 20, EB	<6B
1190 DATA	5C, 5C, 01, 01, 02, 03, 03, 03	<E2	1790 DATA	27, 4C, 83, 27, C9, 05, D0, 09	<67
1200 DATA	02, 02, 02, 02, 03, 02, 02, 00	<19	1800 DATA	20, DA, 27, 20, EB, 27, 4C, B6	<23
1210 DATA	E6, 3B, D0, 02, E6, 3C, A0, 00	<F0	1810 DATA	27, C9, 06, D0, 09, 20, B7, 25	<85
1220 DATA	2C, 7E, 22, 10, 04, 78, 8D, 3F	<A2	1820 DATA	20, D1, 27, 4C, CA, 26, C9, 07	<EF
1230 DATA	FF, B1, 3B, 8D, 3E, FF, 58, 60	<84	1830 DATA	F0, F6, C9, 08, D0, 0C, 20, D1	<58
1240 DATA	A9, 0D, 2C, A9, 41, 2C, A9, 58	<E9	1840 DATA	27, 20, BD, 25, 20, AB, 25, 4C	<0D
1250 DATA	2C, A9, 59, 2C, A9, 28, 2C, A9	<06	1850 DATA	C9, 26, C9, 09, D0, 06, 20, D1	<F4
1260 DATA	29, 2C, A9, 2E, 2C, A9, 23, 2C	<85	1860 DATA	27, 4C, B6, 27, C9, 0A, D0, 0F	<FA
1270 DATA	A9, 24, 2C, A9, 2C, 2C, A9, 3E	<B6	1870 DATA	20, AE, 25, 20, DA, 27, 20, EB	<98
1280 DATA	2C, A9, 20, 4C, D2, FF, 00, A9	<95	1880 DATA	27, 20, B1, 25, 4C, C9, 26, C9	<11
1290 DATA	04, 2C, A9, 07, 2C, A9, 06, 8D	<1B	1890 DATA	0B, D0, 12, 20, AE, 25, 20, D1	<D5
1300 DATA	C8, 25, 20, C3, 25, CE, C8, 25	<23	1900 DATA	27, 20, B1, 25, 20, BD, 25, 20	<11
1310 DATA	D0, F8, 60, A5, 3B, A6, 3C, 8D	<F7	1910 DATA	AB, 25, 4C, C9, 26, 20, AE, 25	<6E
1320 DATA	8A, 22, 8E, 8B, 22, AD, 7F, 22	<8C	1920 DATA	20, D1, 27, 20, BD, 25, 20, AB	<EE
1330 DATA	AE, 80, 22, 85, 3B, 86, 3C, A2	<77	1930 DATA	25, 20, B1, 25, 4C, C9, 26, 20	<61
1340 DATA	FF, AD, 7E, 22, F0, 03, 8E, 7E	<C0	1940 DATA	BA, 25, 20, E2, 27, 4C, F7, 27	<13
1350 DATA	22, AD, 7D, 22, F0, 03, 8E, 7D	<0B	1950 DATA	A0, 02, 20, 92, 25, 8D, 89, 22	<D7
1360 DATA	22, 20, C0, 8C, 20, A2, 25, 20	<5A	1960 DATA	A0, 01, 20, 92, 25, 8D, 88, 22	<E9
1370 DATA	B4, 25, 20, C3, 25, A5, 3B, A6	<F7	1970 DATA	60, 20, BA, 25, AD, 89, 22, 20	<22
1380 DATA	3C, 8D, 88, 22, 8E, 89, 22, 20	<8B	1980 DATA	F7, 27, AD, 88, 22, 48, 4A, 4A	<84
1390 DATA	EE, 27, 20, C3, 25, 20, 90, 25	<78	1990 DATA	4A, 4A, 20, 02, 28, 68, 29, 0F	<25
1400 DATA	8D, 83, 22, 20, F7, 27, 20, C3	<6C	2000 DATA	C9, 0A, 90, 02, 69, 06, 69, 30	<A3
1410 DATA	25, AC, 83, 22, B9, 8C, 22, 8D	<42	2010 DATA	4C, D2, FF, 38, AD, 81, 22, E5	<7A
1420 DATA	84, 22, 30, 14, 2C, 7D, 22, 30	<F7	2020 DATA	3B, AD, 82, 22, E5, 3C, 60, 20	<ED
1430 DATA	0F, 20, CC, 25, A9, 01, 8D, 86	<DE	2030 DATA	73, 04, 08, C9, AD, F0, 23, 28	<B7
1440 DATA	22, A9, 00, 8D, 85, 22, F0, 5D	<F7	2040 DATA	4C, D9, 8B, A2, FF, 86, 3A, 20	<DB
1450 DATA	AC, 83, 22, B9, 8C, 22, 29, 0F	<E1	2050 DATA	5A, 88, 86, 3B, 84, 3C, 20, 73	<1D
1460 DATA	8D, 85, 22, AB, B9, 7C, 25, 8D	<8E	2060 DATA	04, AA, F0, EF, B0, 03, 4C, 2E	<3F
1470 DATA	86, 22, C9, 01, F0, 2B, C9, 02	<70	2070 DATA	87, 20, 53, 89, 20, 79, 04, 4C	<3E
1480 DATA	F0, 19, A0, 01, 20, 92, 25, 20	<5F	2080 DATA	1C, 28, 28, 20, 4B, 28, 4C, DC	<F2
1490 DATA	F7, 27, 20, C3, 25, A0, 02, 20	<C8	2090 DATA	8B, A9, 28, 48, A9, 53, 48, 4C	<F6
1500 DATA	92, 25, 20, F7, 27, 20, C3, 25	<87	2100 DATA	73, 04, F0, 30, 20, E1, 9D, 8C	<27
1510 DATA	4C, 8E, 26, A0, 01, 20, 92, 25	<43	2110 DATA	7F, 22, 8D, 80, 22, 20, 79, 04	<8D
1520 DATA	20, F7, 27, 20, C9, 25, 4C, 8E	<D2	2120 DATA	F0, 22, 20, DE, 9D, 8C, 81, 22	<49
1530 DATA	26, 20, CF, 25, 2C, 7D, 22, 10	<DE	2130 DATA	8D, 82, 22, 20, 79, 04, F0, 14	<E7
1540 DATA	0B, 2C, 84, 22, 30, 06, 20, C0	<A2	2140 DATA	20, 91, 94, 20, 84, 9D, 8E, 7E	<C3
1550 DATA	25, 4C, A1, 26, 20, C3, 25, AC	<DC	2150 DATA	22, F0, 09, 20, 91, 94, 20, 84	<14
1560 DATA	83, 22, B9, 8C, 23, 8D, 87, 22	<4D	2160 DATA	9D, 8E, 7D, 22, 4C, DD, 25, 00	<DD



Set in SID

A rutint hanghatások programozásakor használhatjuk, ugyanis a SID legfontosabb regisztereit beállító rutinnal bővíti a Basicet.

Írta: **Olasz Endre**

```
100 REM" [C=/A][10SH/*][SH/C][SH/*][6SH <0F
/C][10SH/*][C=/S]"
110 REM" [SH/-][7SPC]SET IN SID[7SPC]C- <37
64[SH/-]"
111 REM" [SH/-][7SPC]=====[11SPC][ <84
SH/-]"
120 REM" [SH/-][7SPC]HASZNALAT:[11SPC][ <80
SH/-]"
121 REM" [SH/-] SYS 49152 , HANG F. , H <36
ANG [SH/-]"
122 REM" [SH/-] F.2 , HANG INP. , HANG <B9
INP. [SH/-]"
123 REM" [SH/-]-2 , HULLAMF. , LECSENG <D6
I. , [SH/-]"
124 REM" [SH/-]JELENG I. , HANG SZUR. , <25
HAN-[SH/B]"
125 REM" [SH/-] GERO , ATALAK 1 , ATALA <BB
K'2 [SH/-]"
130 REM" [SH/-][2SPC],[2SPC],[22SPC][SH <55
/-]"
140 REM" [SH/-] KESZITETTE: BRUCE MAEST <D6
RO[2SPC][SH/-]"
150 REM" [C=/Z][2SH/*][SH/C][13SH/*][6S <47
H/C][6SH/*][C=/X]"
160 : <7B
180 DEF FN A(X)=ASC(MID$(A$,X,1))-48+7 <0A
*(MID$(A$,X,1))@"
190 FOR I=49152 TO 49268 STEP 8:S=0 <CD
200 ::FOR J=0 TO 7 <E4
210 :::READ A$ <D3
220 ::A=16*FN A(1)+FN A(2) <35
230 :::POKE I+J,A <F8
240 ::S=S+A <00
250 ::NEXT <69
260 ::READ A$ <D9
270 ::A=16*FN A(1)+FN A(2) <53
280 IF (S AND 255)=A THEN NEXT :PRINT <72
"OK":END
290 PRINT "HIBA:";PEEK(63)+256*PEEK(64 <89
)
300 DATA 20,73,00,EA,EA,EA,20,9E,0F <D9
301 DATA B7,B4,00,8E,00,D4,20,73,60 <2A
302 DATA 00,20,9E,B7,B4,00,8E,01,B8 <60
303 DATA D4,20,73,00,20,9E,B7,B4,90 <E2
304 DATA 00,8E,03,D4,20,73,00,20,18 <C0
305 DATA 9E,B7,B4,00,8E,04,D4,20,8F <D1
306 DATA 73,00,20,9E,B7,B4,00,8E,2A <35
307 DATA 05,D4,20,73,00,20,9E,B7,E1 <9B
308 DATA B4,00,8E,06,D4,20,73,00,AF <9C
309 DATA 20,9E,B7,B4,00,8E,17,D4,A2 <CA
310 DATA 20,73,00,20,9E,B7,B4,00,BC <E6
311 DATA 8E,18,D4,20,73,00,20,9E,CB <8A
312 DATA B7,B4,00,8E,19,D4,20,73,79 <E8
313 DATA 00,20,9E,B7,B4,00,8E,1A,D1 <FA
314 DATA D4,4C,AE,A7,FF,FF,FF,FF,71 <D2
```

Bűvös négyzetek

Diákok körében ismert a bűvös négyzetek előállításának egy trükkös módszere. Vegyünk egy páratlan oldalú négyzetet, majd kezdjük el beleírni a számokat 1-től n^2 -ig. Minden szám után a következő mezőnek a tőle jobbra fel találhatót vegyük (természetesen a tábla körkörös). Ha elakadunk, lépünk egyet lefelé, és már folytathatjuk is a feltöltést. Ha betelt a tábla, kész a bűvös négyzet. Az egyes sorok, oszlopok, illetve átlók összege állandó. Ha valaki ezt az összeget képlettel szeretné felírni, jól nézze meg a középső oszlopot! Valami furcsát fedezhet fel benne.

Írta: **Lantos Zoltán**

```
100 REM BUVOS NEGYZET <24
110 : <B7
120 PRINT CHR$(147);"BUVOS NEGYZET:" <74
130 IF (M AND 1)=0 THEN INPUT "MERET ( <8F
1-9 PARATLAN) ";M:GOTO 130
140 PRINT <C0
150 : <BA
160 DIM A(M,M) <5D
170 Y%=M/2+.5:X%=1 <AF
180 A(X%,Y%)=1:IF M=1 THEN 290 <C6
190 FOR X=2 TO M*M <E2
200 PX=X%:PY=Y% <85
210 X%=X%-1:IF X%=0 THEN X%=M <A7
220 Y%=Y%+1:IF Y%>M THEN Y%=1 <1C
230 IF A(X%,Y%)=0 THEN 260 <4E
240 X%=PX+1:IF X%>M THEN X%=1 <F8
250 Y%=PY <3A
260 A(X%,Y%)=X <D0
270 NEXT X <0E
280 : <13
300 FOR X=1 TO M:FOR Y=1 TO M <F7
310 Z=A(X,Y) <A0
320 PRINT TAB(Y*4-4-(Z<10));Z; <D2
340 NEXT Y:PRINT :PRINT :NEXT X <45
```

Prompt

A rutin futtatása után minden Return leütésekor a következő sor elején egy > jel jelenik meg promptként. Ez természetesen az input pufferre nincs hatással, így nyugodtan dolgozhatunk mellette.

Írta: **Fogarasi András**

```
1 IF A(>256 THEN POKE 49151+I,A:READ <AE
A:I=I+1:GOTO 1
2 DATA 169,12,162,192,141,2,3,142,3, <CF
3,96,255,169,62,32,210,255,32,131,
164,96,256
3 SYS 49152 <44
4 REM *** STARSOFT SOFTWARE *** <60
5 REM *** 1991.01.20. *** <42
6 REM *** GREETINGS TO BOGI SOFTWARE <A5
***
```

IBM kurzor

A program segítségével kurzorunk formáját állíthatjuk be (580-as sor). Külön érdekessége, hogy villogását is lelassítja, ez pedig egyes IBM Basic-ekhez hasonló hatást kelt. Villogásának sebességét egyébként a 49177-es címen módosíthatjuk.

Írta: **Olasz Endre**

```

100 REM***** <7B
120 REM* IBM CURSOR C-64* <A7
140 REM* , , * <1D
160 REM* KESZITETTE :BRUCE MAESTRO * <D7
180 REM***** <45
200 : <0D
220 FOR I=49152 TO 49269 <2D
240 ::READ A:POKE I,A:S=S+A <17
260 NEXT <61
280 IF S<>16332 THEN PRINT "HIBA" <AB
281 DATA 120,169,013,141,020,003,169 <06
282 DATA 192,141,021,003,088,096,032 <29
283 DATA 234,255,165,204,208,041,198 <99
284 DATA 205,208,037,169,020,133,205 <7F
285 DATA 164,211,070,207,174,135,002 <D6
286 DATA 177,209,176,017,230,207,133 <77
287 DATA 206,032,036,234,177,243,141 <E7
288 DATA 135,002,174,134,002,165,206 <41
289 DATA 073,128,032,028,234,076,097 <00
290 DATA 234 <0C
300 DATA 120,165,001,072,041,251,133 <0E
320 DATA 001,169,000,133,251,169,208 <F4
340 DATA 133,252,169,000,133,253,169 <3D
360 DATA 048,133,254,160,000,162,000 <D5
380 DATA 177,251,145,253,200,208,249 <8A
400 DATA 232,230,252,230,254,224,016 <FF
420 DATA 208,240,104,133,001,169,028 <36
440 DATA 141,024,208,088,096 <DC
460 SYS 49152:POKE 49177,0:SYS 49216 <F8
480 REM----- <A1
500 : <50
520 FOR I=12288+8*160 TO 12288+8*160+7 <98
540 READ A:POKE I,A <FA
560 NEXT <44
580 DATA 15,15,15,15,15,15,15,15 <45
600 END <7B

```

Exchange floppy

A program lehetővé teszi lemezegységünk átcímzését szoftverrel, például 8-asról 9-esre. Az egység azonnal visszakapja eredeti számát, ahogy a drive NMI (9-es user) vagy RESET (10-es user) hívást kap. VC 1541-nél a drive sebessége sem állítható a program futtatása után, bár ennek csak olyanoknál

van jelentősége, amelyek kifejezetten két különböző géptípusra készültek.

Írta: **Olasz Endre**

```

100 REM ***** <7B
110 REM * EXCHANGE FLOPPY C-64* <CE
120 REM * , , * <9C
130 REM * KESZITETTE: BRUCE MAESTRO * <14
140 REM ***** <46
150 : <BA
160 PRINT "[SH/CLR][4DOWN]:POKE 53280 <FB
,0:POKE 53281,0:POKE 646,5:PRINT "
CHANGE DEVICE# ...."
170 INPUT "[DOWN]OLD DEVICE NUMBER ";0 <68
180 IF 0<8 OR 0>15 THEN 170 <69
190 INPUT "NEW DEVICE NUMBER ";N <40
200 IF 0<8 OR 0>15 THEN 190 <3E
210 OPEN 15,0,15 <F6
220 PRINT#15,"M-W"CHR$(119)CHR$(0)CHR$ <15
(2)CHR$(N+32)CHR$(N+64)
230 CLOSE 15 <6A
240 END <A4

```

Szabad blockok

Az alábbi program tetszőleges lemez szabad blockjainak számát határozza meg. Előnye, hogy a BAM-tesztelő eljárásoknál lényegesen gyorsabb, és mindig azt az értéket adja, ami a lemezkatalógusból kiolvasható. Ha esetleg ezt az értéket átírtuk, akkor az új blockszámot adja eredményül.

Írta: **Lantos Zoltán**

```

10 REM 'SZABAD BLOCKOK SZAMA' <9E
15 REM LANTOS ZOLTAN <27
20 : <55
30 E$=CHR$(0):OPEN 1,8,0,"$" +E$:POKE <A2
781,1:SYS 65478:GET A$,A$
40 GET A$,A$:IF ST THEN SYS 65484:CLO <A0
SE 1:GOTO 80
50 GET H$,L$ <87
60 GET A$,B$:IF A$ THEN 60 <6C
70 GOTO 40 <15
80 F=ASC(H$+E$)+256*ASC(L$+E$) <A1
90 : <0A
100 PRINT F;"BLOCKS FREE." <17

```

Grafik picture on

A program futtatása után egyszerűbbé válik a képernyőmód beállítása. A rutin SYS 50000,X,Y,Z utasítással hívható, ahol X,Y és Z rendre az 53272, 53265 valamint 56576-os címek megfelelői.

Írta: **Olasz Endre**

```

100 REM"[C=/A][27SH/*][C=/S]" <3A
110 REM"[SH/-][4SPC]GRAFIK PICTURE ON[ <87
3SPC]C-64[SH/-]"
120 REM"[SH/-][27SPC][SH/-]" <C5
121 REM"[SH/-][3SPC]AKT:SYS50000,X,Y,A <33
[6SPC][SH/-]"
122 REM"[SH/-][3SPC]X=KEP.MEMORIA Y=HI <B2
.UZEM.[SH/-]"
123 REM"[SH/-][3SPC]A=A POR. ADAT.[10S <43
PC][SH/-]"
130 REM"[SH/-][2SPC],[2SPC],[21SPC][SH <52
/-]"
140 REM"[SH/-] KESZITETTE: BRUCE MAEST <9E
RO [SH/-]"
150 REM"[C=/Z][27SH/*][C=/X]" <D6
160 : <7B
180 DEF FN A(X)=ASC(MID$(A$,X,1))-48+7 <0A
*(MID$(A$,X,1))>"@")
190 FOR I=50000 TO 50045 STEP 8:S=0 <B9
200 ::FOR J=0 TO 7 <E4
210 :::READ A$ <D3
220 :::A=16*FN A(1)+FN A(2) <35
230 :::POKE I+J,A <F8
240 :::S=S+A <00
250 :::NEXT <69
260 :::READ A$ <D9
270 :::A=16*FN A(1)+FN A(2) <53
280 IF (S AND 255)=A THEN NEXT :PRINT <72
"OK":END
290 PRINT "HIBA: ";PEEK(63)+256*PEEK(64 <89
)
300 DATA 20,73,00,EA,EA,EA,20,9E,0F <D9
301 DATA B7,B4,00,8E,18,D0,20,73,74 <1A
302 DATA 00,EA,EA,EA,20,9E,B7,B4,E7 <8B
303 DATA 00,8E,11,D0,20,73,00,EA,EC <06
304 DATA EA,EA,20,9E,B7,B4,00,8E,8B <6E
305 DATA 00,DD,4C,AE,A7,66,66,66,B0 <30

```

Egyszerű magnókezelés

A program lefuttatása, majd a SYS 49152 kiadása után programjaink a magnó-turbókhöz hasonló módon <L, illetve <S parancsokkal tölthetők vagy menthetők.

Írta: **Olasz Endre**

```

100 REM"[C=/A][27SH/*][C=/S]" <3A
110 REM"[SH/-][4SPC]NEW LOAD & SAVE[4S <F6
PC]C-64[SH/-]"
120 REM"[SH/-][27SPC][SH/-]" <C5
130 REM"[SH/-] INIC: SYS49152[3SPC]AKT <A5
:<L[3SPC][SH/-]"
140 REM"[SH/-][22SPC]<S[3SPC][SH/-]" <3B
150 REM"[SH/-][2SPC],[2SPC].[21SPC][SH <1F
/-]"
160 REM"[SH/-] KESZITETTE: BRUCE MAEST <18
RO [SH/-]"
170 REM"[C=/Z][27SH/*][C=/X]" <57
180 : <FC
190 DEF FN A(X)=ASC(MID$(A$,X,1))-48+7 <CA
*(MID$(A$,X,1))>"@")
200 FOR I=49152 TO 49203 STEP 8:S=0 <8B
210 ::FOR J=0 TO 7 <24
220 :::READ A$ <92
230 :::A=16*FN A(1)+FN A(2) <F5
240 :::POKE I+J,A <38
250 :::S=S+A <C0
260 :::NEXT <AA
270 :::READ A$ <98
280 :::A=16*FN A(1)+FN A(2) <92
290 IF (S AND 255)=A THEN NEXT :PRINT <B2
"OK":END
300 PRINT "HIBA: ";PEEK(63)+256*PEEK(64 <F9
)
310 DATA A9,0B,8D,08,03,A9,C0,8D,42 <C4
320 DATA 09,03,60,20,73,00,C9,5F,27 <CC
330 DATA F0,06,20,79,00,4C,E7,A7,69 <AC
340 DATA 20,73,00,C9,4C,D0,03,20,9B <57
350 DATA 2D,C0,C9,53,D0,03,20,30,2C <4B
360 DATA C0,4C,08,AF,FF,4C,2C,F7,31 <2F
370 DATA 4C,6A,F7,FF,FF,FF,FF,FF,A8 <36

```

Filenév-kereső

A korábban megjelent file-kereső rutin kibővített változata. A program nemcsak teszteli, hogy a specifikált file a lemezen van-e, hanem meg is keresi azt, és a teljes nevét adja vissza.

Írta: **Lantos Zoltán**

```

10 INPUT "FILE-NEV ";N$ <66
20 IF N$="" THEN 140 <F2
30 : <E5
40 OPEN 1,8,0,"$"+N$ <39
50 GET#1,A$,A$ <04
60 E$="" :O=0 <B2
70 GET#1,A$,A$,A$,B$:IF ST THEN CLOSE <C9
1:END
80 GET#1,A$ <4A
90 IF A$<>" THEN 110 <FD
100 S=S+1:IF S=3 THEN CLOSE 1:GOTO 140 <C0
110 IF A$=CHR$(34) THEN O= NOT O:GOTO <66
80
120 IF O AND S=2 THEN E$=E$+A$:GOTO 80 <FF
130 GOTO 80 <AE
140 IF E$="" THEN PRINT "A FILE NINCS <DC
A LEMEZEN !":END
150 PRINT "A FILE NEVE :";E$ <59

```

CSAK LEMEZEN



Áramkörtervező

A következő program C-64-re, Graphics Basicben íródott.

Különböző kapcsolási rajzok elkészítéséhez ad jelentős segítséget. Használata előtt az áramkörfile-t le kell futtatni, ez hozza létre a szükséges – áramköri jeleket tartalmazó – állományokat (ezt érdemes egy üres lemezen végezni).

A program az elkészített kapcsolási rajzokat lemezen tárolja, illetve nyomtatóra is ki tudja rajzolni.

A következőkben az egyes funkciókat ismertetjük:

F1 – írás a grafikára.

A kurzor szerepét itt egy kis piros négyzet tölti be, ezzel írhatunk. A kurzormozgató billentyűkkel a már megszokott módon irányíthatjuk, de lehetőségünk van finomabb pozicionálásra is (a + és a – billentyűkkel a kurzor mozgásának lépésközét módosíthatjuk). A grafikus képre minden karakter kiírható, ami a billentyűzet tetején van (még a PI is), kivéve a <kukac> és a karaktereket.

Egyéb speciális jelek:

Á, É, Í, Ó	– shiftelve a megfelelő, ékezet nélküli betűk
Ü, Ű	– font, Shift/Font
Ö, Ő	– @, Shift/@
mikrojel	– Shift/*
nano	– Shift/N
piko	– Shift/P
m (méter)	– Shift/M
l (hossz)	– Shift/L
+	– shiftelve +, illetve –
ohm	– Shift/R

A Space törli a kurzorpozícióban levő karaktert. Az írás funkcióból az F1-gyel léphetünk ki. Programbővítési szempontból lényeges, hogy ezek a karakterek 5-ször 8 ponból állnak, formáik pedig datasorokban vannak megadva.

F2 – grafika törlése keretrajzolással,

F3 – elem kiválasztása.

Itt a kiválasztást értelem szerint a kurzorgombokkal végezzük, az 1, illetve 2 billentyűkkel pedig lapozhatunk.

A kiválasztott áramköri elem betöltődik lemezről. Ha nem akarunk választani, akkor az utolsó opcióval azonnal visszatérhetünk a főmenübe.

F5 – nyomtatás.

Lehetőségünk van a kinyomtatott rajz után

egyéb szöveg kiírására is. Ezt a nyomtatás előtt kell megadnunk egy kezdetleges szövegszerkesztővel. A sorokat számozza, az utolsó sorba pedig @ karaktert kell írunk. F1-gyel bármikor – természetesen nyomtatáson kívül – visszatérhetünk a főmenübe. A nyomtatás a Space leütésére indul.

F7 – grafika.

Itt rajzolhatjuk meg a tényleges áramkört. A kurzor az írásfunkcióhoz hasonlóan irányítható, a + és – itt is a mozgás lépésközét állítja.

P billentyű – a kurzorból radírt csinál,

G – a kurzor mozgás közben rajzol,

R – az előbbi két funkció megszüntetése,

* – nagyobb terület törlése. Leütése után a kisebb nyíllal meg kell adnunk a törölni kívánt terület két átellenes sarkát. A T billentyű törli a blockot vagy a Space-szel kiléphetünk törlés nélkül.

<pont>, <vessző> – a nyíl irányának beállítása.

Akkor lehet fontos, ha rajzolásnál nem akarjuk, hogy a nyíl eltakarjon valamit.

Space – visszatérés a főmenübe.

Shift/X és X, illetve Shift/Y és Y – elmentik, illetve visszatöltik a kurzor X vagy Y irányú koordinátáit.

S – az F3-mal kiválasztott áramköri elem beszerkesztése. A jel az 1..4 billentyűkkel forgatható, Return leütésére felkerül a rajzra.

Vonal rajzolása:

A rajzolni kívánt vonal két végpontján Returnt kell ütnünk, majd a két pontot a V vagy az L leütésére a gép összeköti. V-nél sarkozással (szögletesen), L-nél pedig ferde vonallal. Ha valamelyik kijelölt pont nem felel meg, T-vel törölhetjük azt. Ha a vonal vízszintes vagy függőleges, akkor a program automatikusan meghúzza.

M – kép betöltése vagy lemezre mentése,

1, 2 – kép betöltése, kimentése,

3 – vissza a főmenübe.

A kép nevének beírása közben F1-gyel azonnal a főmenübe juthatunk.

D – lemezkatalógus.

Ha a lemezkatalógus hívása során Dos-hiba miatt a gép lefagy, Run/Stop+Restore után tetszőleges funkcióbillentyű leütésére a program újra indul. Az áramkör nem törlődik!

F – lemez formázása.

Hasonlóan az előbbi input menükhöz, innen is F1-gyel léphetünk ki. Csak a lemez nevét kell megadnunk.

Írta: **Kocsis Attila**

Karaktergenerátor

Az újságban már nemegyszer találkozhattunk karakterkészletet tervező programmal. Ez azonban amellet, hogy a teljes készlet módosítását lehetővé teszi, még datasorokba is rendezi a kiválasztott karaktereket. Természetesen lehetőségünk van rollozásra, invertálásra, tükrözésre is.

Szókitaláló

A program – a TV műsorából már ismert „Ötből ötöt”-höz hasonló – szókitaláló játék.

Teljes szavakat kell beírni, a program pedig a megfelelő helyen levő, talált betűket értékeli. A szót adott számú próbálkozásból kell kitalálnunk.

Telefonkönyv

Ez egy pár száz név tárolására alkalmas telefonkönyv, amelyben ismerőseink lakás- és munkahelyi telefonszámát tárolhatjuk relatív fájlban.

Görög betűk

A program a görög kis- és nagybetűk használatát biztosítja. Mivel az új karakterek a grafikus jelek helyén jönnek létre, így az eredeti karakterkészlet is használható marad.

Órarendkészítő

A program órarend készítéséhez ad segítséget. Megszerkeszthetünk 1, illetve 2 hét periódusú órarendet is, amit utána akár el is menthetünk vagy printerre nyomtathatunk. A program órák permutálását nem végzi, így inkább diákok, mint iskolavezetőség részére készült.

A fenti öt programot írta:
Várady Tamás



Kincsesládá

A következő lövöldözős játékban tankkal kell megszereznünk egy kincsesládát. A játék port 2-ről irányítható.

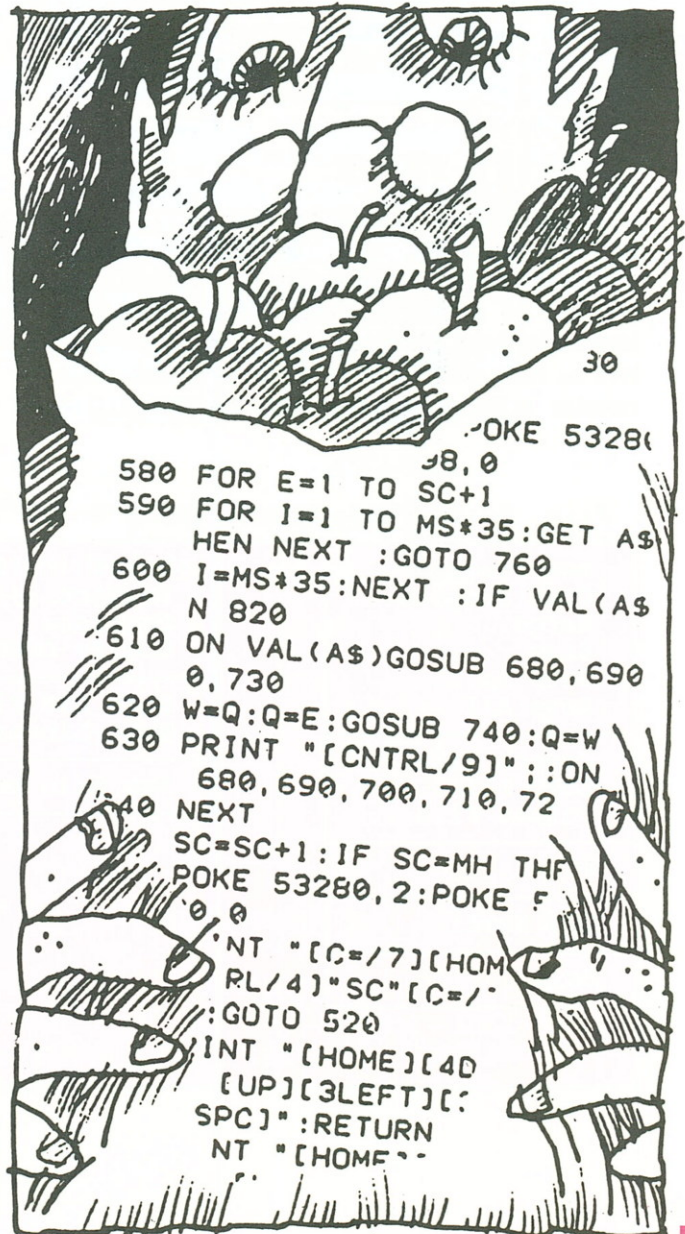
Cspadában

A klasszikus Purple Turtles játék – ezúttal barlangban. Teknőcök hátán ugrálva kell átjutnunk a tó túloldalára. Részletesebb magyarázat a programban található.

Iskolás gyerek

A játékban egy iskolás gyerekkel kell rendetlenkednünk a suliban. Irányítás: joystick port 2.

Az utolsó három programot írta:
Gábryel



Kezdő lépések

A most következő oldalakon IBM PC-kompatibilis számítógépen futtatható programokat találunk. Ezek oktató jellegűek (van köztük magyarázó-, számoló- és bemutatóprogram) – valamennyien a matematika egy-egy részének bemutatására szolgálnak.

A programok Quick Basicben készültek, ezért beírásukhoz szükséges a Basic nyelv (akik a programokat megrendelik lemezen, azoknak átmásoljuk a Quick Basicet). A programok grafikus utasításokat használó sorai miatt minimum CGA monitorra (és kártyára) van szükségünk.

Néhány szó a Quick Basic használatáról azoknak, akik még nem foglalkoztak efféle programozással. A program elindítása után (QB, majd Enter) a program Basic szerkesztőjébe lépünk. Ha van egerünk, akkor könnyű a dolgunk: a felső legördülő menürendszerből egyszerűen és gyorsan választhatunk. Itt az első oszlopból választhatjuk ki a kimentési és a betöltési funkciókat. Egér hiányában az Alt gomb, majd az Enter egyszeri megnyomására szintén lehívhatjuk a menüt, ahol ezek

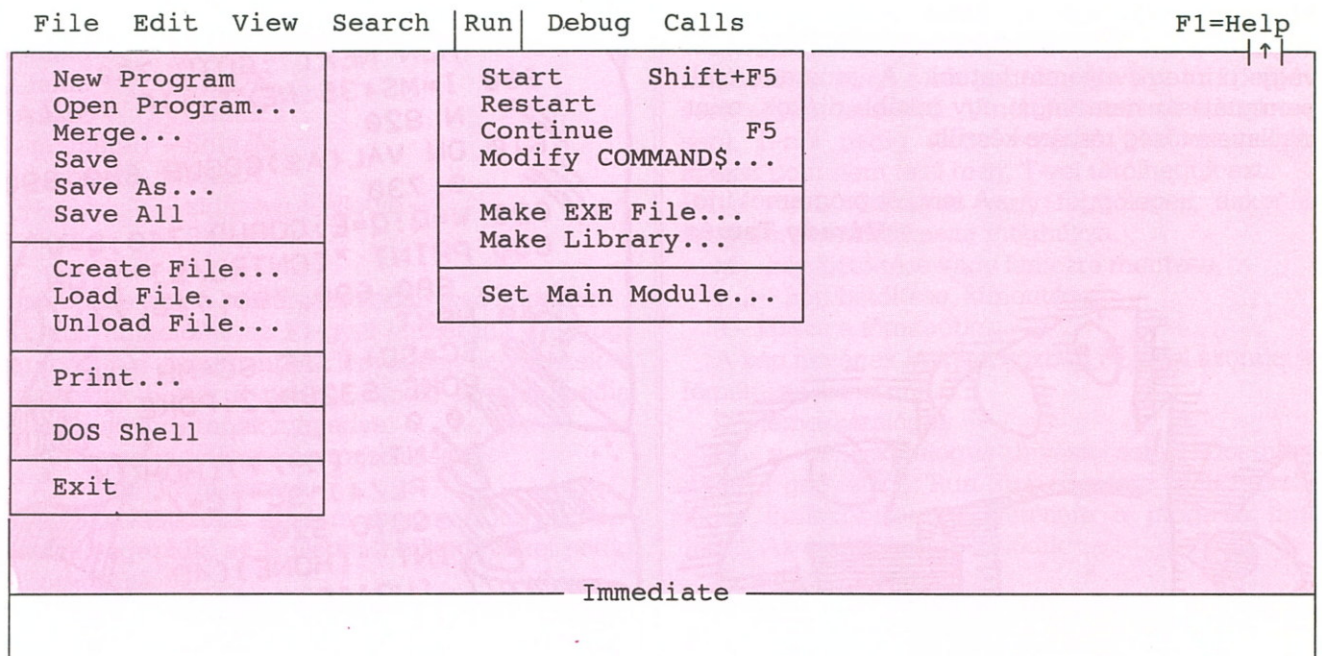
után a kurzorbillentyűkkel mozoghatunk, és az Enterrel választhatunk. A Yes/No kérdéses és lemezkezelői ablakokban a Tab gombbal ugrálhatunk az egyes ablakrészek között.

Ha elkészültünk a programlista bepötyögésével, akkor a menü Save As (kimentés) pontját választjuk ki; ezek után be kell írunk a program nevét, majd az OK felirat megnyomásával lemezre (vagy winchesterre, ekkor az alkönyvtárak kiválasztása is fontos) mentjük.

Az elmentett programot a Shift+F5 gomb megnyomásával (vagy a legördülő menüből a Start kiválasztásával) indíthatjuk el. Ugyanígy járunk el akkor is, ha a megvásárolt lemezről egyenként töltjük be a kívánt programokat.

A Quick Basic (kilépéskor, vagy ha a következő programot kívánjuk betölteni) sokszor megkérdezi, hogy a kilépés vagy a következő program betöltése előtt elmentse-e a gépben lévőket. Ekkor Yes (igen), No (nem) és Cancel (elvet) feliratok közül kell választanunk.

(Az IBM PC gépeken futó programokat Pethes Endre írta)



Négyzetek és körök

```

SCREEN 0
COLOR 12, 1, 11
CLS
pi = 3.14159
y = 2: x = 2: z = 20: GOSUB keret
LOCATE 12, 2: PRINT CHR$(199): LOCATE 12, 79: PRINT CHR$(182)
FOR i = 1 TO 76: LOCATE 12, 2 + i: PRINT CHR$(196): NEXT
COLOR 14, 1, 11
LOCATE 3, 8: PRINT "Legyen adott egy kör és egy négyzet. Tételezzük fel, hogy vagy:"
LOCATE 5, 18: PRINT "1) kerületeik egyenlőek, vagy"
LOCATE 7, 18: PRINT "2) területeik egyenlőek."
LOCATE 9, 8: PRINT "Adjuk meg a közös kerületet, illetve területet. Számítsuk ki ebből a"
LOCATE 11, 8: PRINT "négyzetoldalt, a körsugarat és a hiányzó kerületet, vagy területet."
COLOR 15, 1, 11
ujra:
COLOR 11, 1, 11
LOCATE 14, 8: PRINT "Mit választ: közös kerület (1), közös terület (2):"
LOCATE 14, 66: PRINT "(0=vége)": LOCATE 14, 60: INPUT v
IF v = 1 THEN GOTO ker
IF v = 2 THEN GOTO ter
IF v = 0 THEN END
ker:
LOCATE 16, 8: INPUT "Kérem a közös kerület hosszát: "; k
r = k / (2 * pi): o = k / 4: kt = r ^ 2 * pi: nt = o ^ 2
COLOR 15, 1, 11
LOCATE 18, 8: PRINT "A négyzet oldala ="; o; "a négyzet területe ="; nt
LOCATE 20, 8: PRINT "A kör sugara ="; r; ", a kör területe ="; kt
GOSUB all
CLS : GOTO ujra
ter:
LOCATE 16, 8: INPUT "Kérem a közös terület nagyságát: "; t
r = SQR(t / pi): o = SQR(t): kk = 2 * r * pi: kn = 4 * o
COLOR 15, 1, 11
LOCATE 18, 8: PRINT "A négyzet oldala ="; o; "a négyzet kerülete ="; kn
LOCATE 20, 8: PRINT "A kör sugara ="; r; ", a kör kerülete ="; kk
GOSUB all
CLS : GOTO ujra
END
all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN
keret:
t = 40 - x: x1 = 40 + t: v = x1 - x: m = z: y1 = y + m + 1
LOCATE y, x: PRINT CHR$(201)
LOCATE y, x1 + 1: PRINT CHR$(187)
FOR i = 1 TO v: LOCATE y, x + i: PRINT CHR$(205): NEXT
LOCATE y1, x: PRINT CHR$(200)
LOCATE y1, x1 + 1: PRINT CHR$(188)
FOR i = 1 TO v: LOCATE y1, x + i: PRINT CHR$(205): NEXT
FOR i = 1 TO m: LOCATE y + i, x: PRINT CHR$(186): NEXT
FOR i = 1 TO m: LOCATE y + i, x1 + 1: PRINT CHR$(186): NEXT
RETURN

```

Nem kell megjedni, nem találtuk fel a kör négyszögesítését! Az azonban előfordulhat, hogy egy kör és egy négyzet egymás mellett szerepel, s a feladat szerint a síkidom kerülete vagy területe egyenlő. A program ilyen esetekre meghatározza a síkidomok adatait.

```

Prog:
SCREEN 0
COLOR 5, 0, 6
CLS
LOCATE 2, 9: FOR i = 1 TO 62: PRINT CHR$(220); : NEXT
FOR j = 0 TO 1
FOR i = 1 TO 9: LOCATE 2 + i, 10 - j: PRINT CHR$(219): NEXT
FOR i = 1 TO 9: LOCATE 2 + i, 69 + j: PRINT CHR$(219): NEXT: NEXT
LOCATE 12, 9: FOR i = 1 TO 62: PRINT CHR$(223); : NEXT
COLOR 6, 0, 6
LOCATE 19, 1: FOR i = 1 TO 80: PRINT CHR$(220); : NEXT
LOCATE 24, 1: FOR i = 1 TO 80: PRINT CHR$(223); : NEXT
COLOR 1, 15, 6
FOR i = 0 TO 3
LOCATE 20 + i, 2: PRINT "
NEXT
COLOR 4, 15, 6
LOCATE 3, 11: PRINT "
LOCATE 4, 11: PRINT "          SZAMOK EGESZ SZAMU OSZTOI
LOCATE 5, 11: PRINT "
LOCATE 6, 11: PRINT " A program egy adott szam eseteben megkeresi azokat az
LOCATE 7, 11: PRINT " egesz szamokat, amelyekkel az adott szam maradek nelkul
LOCATE 8, 11: PRINT " oszthato. A szamokat kiirja, s ha nem talal tobbet, ezt
LOCATE 9, 11: PRINT " jelzi. (Jelzeset varjuk meg !)
LOCATE 10, 11: PRINT "
LOCATE 11, 11: PRINT "
COLOR 14, 0, 6
LOCATE 13, 9: PRINT CHR$(201)
FOR i = 0 TO 59: LOCATE 13, 10 + i: PRINT CHR$(205): NEXT
LOCATE 13, 70: PRINT CHR$(187)
FOR i = 0 TO 3: LOCATE 14 + i, 9: PRINT CHR$(186); : NEXT
FOR i = 0 TO 3: LOCATE 14 + i, 70: PRINT CHR$(186); : NEXT
LOCATE 18, 9: PRINT CHR$(200)
FOR i = 0 TO 59: LOCATE 18, 10 + i: PRINT CHR$(205): NEXT
LOCATE 18, 70: PRINT CHR$(188)
COLOR 4, 3, 6
LOCATE 14, 11: PRINT "
LOCATE 15, 11: PRINT "          Kere'm, adja meg a szamot :
LOCATE 16, 11: PRINT "          (Ha 0-t ad, a program megall.)
LOCATE 17, 11: PRINT "
LOCATE 15, 53: INPUT s
IF s = 0 THEN END
COLOR 1, 15, 6
FOR i = 0 TO 3
LOCATE 20 + i, 2: PRINT "
NEXT
b = 1: x = 20: y = 3
GOTO szamol
szamol:
c = s / b: c1 = INT(c)
IF c = c1 THEN GOTO kiir2 ELSE GOTO kiir3
kiir2:
LOCATE x, y: PRINT b; : b = b + 1: y = y + 7
IF y > 70 THEN x = x + 1: y = 3
IF b > s THEN PRINT " Nincs tobb": GOSUB all: GOTO prog
GOTO szamol
kiir3:
b = b + 1: GOTO szamol
all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN

```

Számok egész számú osztói

A programlistában olvasható a bővebb magyarázat.

```

SCREEN 1
WINDOW (0, 0)-(639, 479)
COLOR 0, 2
LINE (25, 395)-(625, 185), , B
LINE (25, 180)-(625, 165), , BF
LINE (25, 162)-(625, 155), 2, BF
LINE (25, 150)-(625, 145), 1, BF
LINE (30, 390)-(319, 190), 1, B: LINE (321, 390)-(620, 190), 2, B
LINE (45, 380)-(65, 380), 1: LINE -(65, 260), 1: LINE -(45, 260), 1: LINE -(45, 380), 1
LINE (85, 320)-(105, 320), 1: LINE -(145, 240), 1: LINE -(145, 320), 1
LINE -(165, 320), 1: LINE -(165, 200), 1: LINE -(145, 200), 1
LINE -(105, 280), 1: LINE -(105, 200), 1: LINE -(85, 200), 1: LINE -(85, 320), 1
LINE (165, 380)-(191, 260), 1: LINE -(219, 260), 1: LINE -(245, 380), 1
LINE -(225, 380), 1: LINE -(205, 300), 1: LINE -(185, 380), 1: LINE -(165, 380), 1
LINE (245, 320)-(305, 320), 1: LINE -(305, 300), 1: LINE -(265, 300), 1
LINE -(265, 270), 1: LINE -(285, 270), 1: LINE -(285, 250), 1
LINE -(265, 250), 1: LINE -(265, 220), 1: LINE -(305, 220), 1
LINE -(305, 200), 1: LINE -(245, 200), 1: LINE -(245, 320), 1
PAINT (35, 385), 1
LINE (330, 380)-(370, 380), 2: LINE -(390, 370), 2: LINE -(390, 320), 2
LINE -(370, 300), 2: LINE -(390, 260), 2: LINE -(370, 260), 2
LINE -(350, 300), 2: LINE -(350, 260), 2: LINE -(330, 260), 2: LINE -(330, 380), 2
LINE (350, 360)-(364, 360), 2: LINE -(370, 356), 2: LINE -(370, 324), 2
LINE -(364, 320), 2: LINE -(350, 320), 2: LINE -(350, 360), 2
PAINT (353, 353), 2
LINE (410, 320)-(470, 320), 2: LINE -(470, 300), 2: LINE -(430, 220), 2
LINE -(470, 220), 2: LINE -(470, 200), 2: LINE -(410, 200), 2: LINE -(410, 220), 2
LINE -(450, 300), 2: LINE -(410, 300), 2: LINE -(410, 320), 2
LINE (490, 380)-(510, 380), 2: LINE -(510, 260), 2: LINE -(490, 260), 2
LINE -(490, 380), 2
LINE (530, 300)-(550, 320), 2: LINE -(590, 320), 2: LINE -(610, 300), 2
LINE -(610, 220), 2: LINE -(590, 200), 2: LINE -(550, 200), 2
LINE -(530, 220), 2: LINE -(530, 300), 2
LINE (556, 300)-(584, 300), 2: LINE -(590, 294), 2: LINE -(590, 226), 2
LINE -(584, 220), 2: LINE -(556, 220), 2: LINE -(550, 226), 2
LINE -(550, 294), 2: LINE -(564, 300), 2
PAINT (570, 280), 2
PAINT (335, 200), 2
LOCATE 21, 7: PRINT "Pethes Endre, Budapest. 1992."
GOSUB all: CLS
SCREEN 0: COLOR 14, 1, 7: CLS
LOCATE 3, 3: PRINT "Ha adott egy O középpontú R sugarú"
LOCATE 5, 2: PRINT "kör, a sík egy P pontjának inverze az"
LOCATE 7, 2: PRINT "OP félegyenesnek az a P' pontja, a-"
LOCATE 9, 2: PRINT "melyre az"
LOCATE 11, 25: PRINT "2"
LOCATE 12, 14: PRINT "OP * OP' = R"
LOCATE 15, 2: PRINT "összefüggés fennáll."
LOCATE 17, 2: PRINT "Az O pont az inverzió pólusa, a kör az"
LOCATE 18, 2: PRINT "inverzió alapköre, R² pedig az inver-"
LOCATE 19, 2: PRINT "zió hatványa."
COLOR 11, 1, 7
LOCATE 23, 2: PRINT "Szerkesszük meg egy pont inverzét:"
GOSUB all: CLS
SCREEN 1

```

Inverzió

Az inverzió olyan geometriai leképezés, amely egy – a síkban adott tetszés szerinti – kör segítségével, annak középpontja és a sík különböző pontjai között létesít

megfeleltetést. Egy pont inverzének megszerkesztése körzével-vonalzóval elég munkaigényes, a sok pontból álló alakzatoké pe-

dig igencsak türelempróbáló munka.

A számítógép másodpercek alatt meghatározza és a képernyőre rajzolja a geometria ismert alakzatainak – egyenesek, körök és kúpszeletek – inverzeit. A program futás közben lépésenként megáll, hogy a kialakuló ábrát megnézhessük, a szövegeket elolvashassuk. Ilyenkor bármely gomb megnyomásával tovább léphetjük.

```

WINDOW (0, 0)-(319, 225)
COLOR 0, 2
LINK (5, 5)-(315, 220), 2, B
o1 = 100: o2 = 100: r = 65: pi = 3.14159: u = pi / 180: z1 = o1: z2 = o2: GOSUB 9
LOCATE 14, 14: PRINT "O": GOSUB 11: LOCATE 11, 14: PRINT "R"
p1 = 260: p2 = 165: z1 = p1: z2 = p2: GOSUB 9: LOCATE 9, 33: PRINT "P"
v1 = p1 - o1: v2 = p2 - o2: v = SQR(v1 ^ 2 + v2 ^ 2): v3 = v1 / v: v4 = v2 / v
d = r * r / v: p3 = o1 + d * v3: p4 = o2 + d * v4: u1 = -v4: u2 = v3
d1 = SQR(r ^ 2 - d ^ 2): p5 = p3 + d1 * u1: p6 = p4 + d1 * u2
p7 = p3 - d1 * u1: p8 = p4 - d1 * u2: a1 = p1: a2 = p2: b1 = o1: b2 = o2
GOSUB 12: a1 = p5: a2 = p6: b1 = p1: b2 = p2: GOSUB 12
a1 = p7: a2 = p8: b1 = p1: b2 = p2: GOSUB 12
a1 = p5: a2 = p6: b1 = o1: b2 = o2: GOSUB 12
a1 = p7: a2 = p8: b1 = o1: b2 = o2: GOSUB 12
a1 = p7: a2 = p8: b1 = p5: b2 = p6: GOSUB 12
LOCATE 14, 17: PRINT "P": z1 = p3: z2 = p4: GOSUB 9
GOSUB all
LOCATE 14, 29: PRINT "ugyanis:"
LOCATE 16, 29: PRINT "OP=R:OP'"
LOCATE 19, 29: PRINT "OP*OP'=R"
LOCATE 18, 37: PRINT "2"
GOSUB all: CLS
SCREEN 0
COLOR 14, 1, 7: CLS
LOCATE 3, 2: PRINT "Az inverzio alapvető tulajdonságai:"
LOCATE 7, 2: PRINT "Az alapkör pontjának inverze önmaga."
LOCATE 9, 2: PRINT "Külső pont inverze belső pont"
LOCATE 10, 2: PRINT "és fordítva."
LOCATE 12, 2: PRINT "O-nak nincs inverze."
LOCATE 14, 2: PRINT "Ha P inverze P', akkor P' inverze P."
GOSUB all
SCREEN 1
WINDOW (0, 0)-(639, 479)
COLOR 0, 2: CLS
LOCATE 1, 11: PRINT "Egyenesek inverzei"
GOSUB all
o1 = 319: o2 = 239: r = 165: u = 3.14159 / 180
z1 = o1: z2 = o2: GOSUB 9
CIRCLE (o1, o2), r
GOSUB all
LOCATE 23, 2: PRINT "e1"
FOR i = -1900 TO 0 STEP 30: p1 = i: p2 = 30
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 0 TO 639 STEP 3: p1 = 0 + i: p2 = 30: PSET (p1, p2), 2
d = r * r / v: p3 = o1 + (d * v3): p4 = o2 + (d * v4)
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 640 TO 1900 STEP 30: p1 = i: p2 = 30
GOSUB 10: PSET (p3, p4), 1: NEXT
LOCATE 16, 2: PRINT "e2"
FOR i = -1900 TO 0 STEP 30: p1 = i: p2 = 165: GOSUB 10
PSET (p3, p4), 1: NEXT
FOR i = 0 TO 640 STEP 3: p1 = i: p2 = 165: PSET (p1, p2), 2
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 640 TO 1900 STEP 30: p1 = i: p2 = 165: GOSUB 10
PSET (p3, p4), 1: NEXT
LOCATE 4, 2: PRINT "e3"

```

```

FOR i = -1900 TO 0 STEP 30: p1 = i: p2 = 404: GOSUB 10
PSET (p3, p4), 1: NEXT
FOR i = 0 TO 640 STEP 3: p1 = i: p2 = 404: PSET (p1, p2), 2
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 640 TO 1900 STEP 30: p1 = i: p2 = 404: GOSUB 10
PSET (p3, p4), 1: NEXT
LINE (2, o2 - 1)-(639, o2 - 1), 2: LINE (2, o2 + 1)-(639, o2 + 1), 1
LOCATE 12, 2: PRINT "e4=e4"
GOSUB all: CLS
SCREEN 0: COLOR 14, 1, 7: CLS
LOCATE 5, 2: PRINT "Az 0 ponton áthaladó egyenes inverze "
LOCATE 7, 2: PRINT "önmaga.(Az 0 pontot kivéve!)"
LOCATE 10, 2: PRINT "Minden más egyenes inverze kör."
GOSUB all: CLS
SCREEN 0
COLOR 14, 1, 7
LOCATE 2, 15: PRINT "Körök inverzei"
LOCATE 5, 2: PRINT "Az alapkör inverze önmaga."
LOCATE 7, 2: PRINT "Az alapkörrel koncentrikus kör inverze"
LOCATE 8, 2: PRINT "az alapkörrel koncentrikus másik kör."
LOCATE 10, 2: PRINT "Ha a kör tartalmaz két olyan pontot,"
LOCATE 11, 2: PRINT "amelyek egymás inverzei, akkor minden"
LOCATE 12, 2: PRINT "pontjának inverze a kör pontja lesz."
LOCATE 13, 2: PRINT "Az ilyen kör is önmaga inverze."
LOCATE 15, 2: PRINT "Az alapkör középpontján átmenő kör in-"
LOCATE 16, 2: PRINT "verze egyenes. Más körök inverze az"
LOCATE 17, 2: PRINT "eredetitől különböző kör. Középpont-"
LOCATE 18, 2: PRINT "ja nem az eredeti középpont inverze."
GOSUB all: CLS
LOCATE 4, 2: PRINT "Bemutatjuk azt az esetet, amikor a kör"
LOCATE 6, 2: PRINT "két olyan pontot tartalmaz, amelyek"
LOCATE 8, 2: PRINT "egymás inverzei."
LOCATE 10, 2: PRINT "A kör az alapkört merőlegesen metszi."
LOCATE 12, 2: PRINT "(A kör inverze önmaga!)"
GOSUB all: CLS
SCREEN 1
WINDOW (0, 0)-(319, 225)
COLOR 0, 2
o1 = 100: o2 = 100: r = 65: pi = 3.14159: u = pi / 180
z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
LOCATE 14, 12: PRINT "O"
LOCATE 14, 18: PRINT "P"
LOCATE 14, 32: PRINT "P'"
p1 = 240: p2 = 100: z1 = p1: z2 = p2: GOSUB 9
v1 = p1 - o1: v2 = p2 - o2: v = SQR(v1 ^ 2 + v2 ^ 2): v3 = v1 / v: v4 = v2 / v: d = r * r / v
p3 = o1 + (d * v3): p4 = o2 + (d * v4): u1 = -v4: u2 = v3
d1 = SQR(r ^ 2 - d ^ 2): p5 = p3 + d1 * u1: p6 = p4 + d1 * u2
p7 = p3 - d1 * u1: p8 = p4 - d1 * u2
a1 = p1: a2 = p2: b1 = o1: b2 = o2: GOSUB 12
a1 = p5: a2 = p6: b1 = p1: b2 = p2: GOSUB 12
a1 = p7: a2 = p8: b1 = p1: b2 = p2: GOSUB 12
a1 = p5: a2 = p6: b1 = o1: b2 = o2: GOSUB 12
a1 = p7: a2 = p8: b1 = o1: b2 = o2: GOSUB 12
a1 = p7: a2 = p8: b1 = p5: b2 = p6: GOSUB 12
z1 = p3: z2 = p4: GOSUB 9
f1 = (p1 + p3) / 2: f2 = o2: z1 = f1: z2 = f2: GOSUB 9

```

```

r3 = SQR((f1 - p3) ^ 2 + (f2 - p4) ^ 2)
l = 2: FOR i = 0 TO 360 - 1 STEP 1: p1 = f1 + r3 * COS(u * i)
p2 = f2 + r3 * SIN(u * i): PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS
SCREEN 1
WINDOW (0, 0)-(319, 224)
COLOR 0, 2
LOCATE 2, 5: PRINT "Az alapkör középpontján átmenő kör"
LOCATE 3, 5: PRINT "inverze egyenes."
o1 = 160: o2 = 125: r = 60: pi = 3.14159: u = pi / 180
z1 = o1: z2 = o2: GOSUB 9
GOSUB 11: o3 = 220: o4 = 80: r1 = 75: z1 = o3: z2 = o4: GOSUB 9
l = 2: FOR i = 1 TO 360 - 1 STEP 1: x = o3 + r1 * COS(u * i)
y = o4 + r1 * SIN(u * i): x1 = o3 + r1 * COS(u * (i + 1))
y1 = o4 + r1 * SIN(u * (i + 1)): LINE (x, y)-(x1, y1), 2: x = x1: y = y1: NEXT
LOCATE 15, 36: PRINT "K"
l = 3: FOR i = 1 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i): GOSUB 10
PSET (p3, p4), 1: NEXT: LOCATE 7, 31: PRINT "K"
GOSUB all: CLS
SCREEN 1
WINDOW (0, 0)-(319, 224)
COLOR 0, 2
LOCATE 2, 6: PRINT "Egyeb körök inverzei"
o1 = 160: o2 = 108: r = 50: pi = 3.14159: u = pi / 180
z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
o3 = 120: o4 = 123: r1 = 30: z1 = o3: z2 = o4: GOSUB 9: GOSUB 11
CIRCLE (o3, o4), r1, 2
l = 2: FOR i = 0 TO 340 - 1 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 - r1 * SIN(u * i): GOSUB 10
PSET (p3, p4), 1: NEXT
o3 = 260: o4 = 100: r1 = 50: z1 = o3: z2 = o4: GOSUB 9
CIRCLE (o3, o4), r1, 2
l = 4: FOR i = 0 TO 360 - 1 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 - r1 * SIN(u * i): GOSUB 10
PSET (p3, p4), 1: NEXT
GOSUB all: CLS
SCREEN 1
WINDOW (0, 0)-(319, 224)
COLOR 0, 2
LINE (5, 5)-(315, 220), 1, B
LOCATE 8, 11: PRINT "KUPSZELETEK INVERZEI"
GOSUB all: CLS
SCREEN 1
WINDOW (0, 0)-(319, 226)
COLOR 0, 2
LOCATE 1, 11: PRINT "Ellipszis inverze"
f = 1.42
o1 = 160: o2 = 100: r = 60: z1 = o1: z2 = o2
GOSUB 9: GOSUB 11
k1 = 160: k2 = 80: a = 110: b = 50: l = 2: FOR i = 0 TO 360 - 1 STEP 1
p1 = k1 + a * COS(u * i): p2 = k2 + (b * SIN(u * i)) / f
PSET (p1, p2), 2
GOSUB 10
PSET (p3, p4), 1: NEXT
GOSUB all: CLS

```



```

LOCATE 1, 11: PRINT "Ellipszis inverze"
o1 = 160: o2 = 90: r = 60: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11.
k1 = 160: k2 = 80: a = 110: b = 50: l = 2
FOR i = 0 TO 360 - 1 STEP 1: p1 = k1 + a * COS(i * u): p2 = k2 + b * SIN(u * i)
PSET (p1, p2), 2: GOSUB 10
PSET (p3, p4), 1: NEXT
GOSUB all: CLS

LOCATE 1, 11: PRINT "Ellipszis inverze"
o1 = 180: o2 = 110: r = 80: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
k1 = 140: k2 = 110: a = 40: b = 30: l = 2
FOR i = 2 TO 360 - 1 STEP 1: p1 = k1 + a * COS(i * u): p2 = k2 + b * SIN(u * i)
PSET (p1, p2), 2: GOSUB 10
PSET (p3, p4), 1: NEXT
GOSUB all: CLS

LOCATE 1, 11: PRINT "Ellipszis inverze"
o1 = 160: o2 = 90: r = 70: z1 = o1: z2 = o2: GOSUB 9
GOSUB 11
k1 = 160: k2 = 125: a = 110: b = 35: l = 2: FOR i = 0 TO 360 - 1 STEP 1
p1 = k1 + a * SIN(u * i): p2 = k2 - b * COS(u * i)
p3 = k1 + a * SIN(u * (i + 1)): p4 = k2 - b * COS(u * (i + 1))
LINE (p1, p2)-(p3, p4), 2: p1 = p3: p2 = p4: NEXT
FOR i = 15 TO 345 - 1 STEP 1: p1 = k1 + a * SIN(u * i): p2 = k2 - b * COS(u * i)
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS

LOCATE 1, 11: PRINT "Ellipszis inverze"
o1 = 190: o2 = 100: r = 70: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
o3 = 90: o4 = 100: a = 80: b = 30: l = 4: FOR i = 0 TO 360 - 1 STEP 1
p1 = o3 + b * SIN(u * i): p2 = o4 + a * COS(u * i): PSET (p1, p2), 2
GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS

LOCATE 1, 11: PRINT "Ellipszis inverze"
o1 = 160: o2 = 100: r = 70: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
o3 = 160: o4 = 100: a = 70: b = 40: l = 4: FOR i = 0 TO 360 - 1 STEP 1
p1 = o3 + b * SIN(u * i): p2 = o4 + a * COS(u * i): PSET (p1, p2), 2
GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS

LOCATE 1, 11: PRINT "Parabola inverze"
o1 = 190: o2 = 100: r = 70: z1 = o1: z2 = o2: GOSUB 9
GOSUB 11
a1 = 80: a2 = 100: FOR i = -300 TO -100 STEP 3: p1 = a1 + (i ^ 2 / 40): p2 = a2 + i
GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 80: a2 = 100: a = 99: FOR i = 0 TO (2 * a)
p1 = a1 + ((a - i) ^ 2) / 40: p2 = a2 - (a - i): PSET (p1, p2), 2: GOSUB 10
PSET (p3, p4), 1: NEXT
a1 = 80: a2 = 100: FOR i = 200 TO 400 STEP 3: p1 = a1 + ((a - i) ^ 2) / 40
p2 = a2 - (a - i): GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS

LOCATE 1, 1: PRINT " A parabola fókuszja az 0 pont."
o1 = 210: o2 = 100: r = 80: z1 = o1: z2 = o2: GOSUB 9
GOSUB 11
a1 = 150: a2 = 100: FOR i = -300 TO -100 STEP 3: p1 = a1 + (i ^ 2) / 120
p2 = a2 + i: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 150: a2 = 100: a = 99: FOR i = 0 TO (2 * a): p1 = a1 + ((a - i) ^ 2) / 120
p2 = a2 - (a - i): PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 150: a2 = 100: FOR i = 200 TO 400 STEP 3: p1 = a1 + ((a - i) ^ 2) / 120
p2 = a2 - (a - i): GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS

```

```

LOCATE 1, 11: PRINT "Parabola inverze"
o1 = 160: o2 = 110: r = 80: z1 = o1: z2 = o2: GOSUB 9
GOSUB 11: a1 = 80: a2 = 110: FOR i = -300 TO -100 STEP 3
p1 = a1 + (i ^ 2) / 60: p2 = a2 + i: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 80: a2 = 110: a = 99: FOR i = 0 TO (2 * a): p1 = a1 + ((a - i) ^ 2) / 60
p2 = a2 - (a - i): PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 80: a2 = 110: FOR i = 200 TO 400 STEP 3: p1 = a1 + ((a - i) ^ 2) / 60
p2 = a2 - (a - i): GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS
LOCATE 1, 11: PRINT "Parabola inverze"
o1 = 100: o2 = 110: r = 90: z1 = o1: z2 = o2: GOSUB 9
GOSUB 11: a1 = 101: a2 = 110: FOR i = -300 TO -100 STEP 3
p1 = a1 + (i ^ 2) / 60: p2 = a2 + i: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 101: a2 = 110: a = 99: FOR i = 2 TO (2 * a): p1 = a1 + ((a - i) ^ 2) / 60
p2 = a2 - (a - i): PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 101: a2 = 110: FOR i = 200 TO 400 STEP 3: p1 = a1 + ((a - i) ^ 2) / 60
p2 = a2 - (a - i): GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS
LOCATE 1, 11: PRINT "Parabola inverze"
o1 = 100: o2 = 100: r = 70: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
a1 = 170: a2 = 100: FOR i = -300 TO -100 STEP 3: p1 = a1 + (i ^ 2) / 80
p2 = a2 + i: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 170: a2 = 100: a = 99: FOR i = 0 TO (2 * a) STEP 2: p1 = a1 + ((a - i) ^ 2) / 80
p2 = a2 - (a - i): PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
a1 = 170: a2 = 100: FOR i = 200 TO 400 STEP 3: p1 = a1 + ((a - i) ^ 2) / 80
p2 = a2 - (a - i): GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS
LOCATE 1, 12: PRINT "Hiperbola inverze"
o1 = 160: o2 = 100: r = 70: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
k1 = 160: k2 = 100: a = 70: b = 60: l = -1
FOR i = 440 TO 120 STEP 10 * l: p1 = k1 + i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 120 TO a STEP 2 * l: p1 = k1 + i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
l = 2: FOR i = a TO 120 STEP l: p1 = k1 + i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 120 TO 420 STEP 5 * l: p1 = k1 + i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
l = -1: FOR i = 440 TO 120 STEP 10 * l: p1 = k1 - i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 120 TO a STEP 2 * l: p1 = k1 - i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
l = 2: FOR i = a TO 120 STEP l: p1 = k1 - i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 120 TO 420 STEP 5 * l: p1 = k1 - i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
LOCATE 3, 5: PRINT "Az inverz gorbe most lemniszkata"
GOSUB all: CLS
SCREEN 1
WINDOW (0, 0)-(319, 224)
COLOR 0, 2
LOCATE 1, 12: PRINT "Hiperbola inverze"
o1 = 230: o2 = 100: r = 70: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
k1 = 110: k2 = 100: a = 30: b = 50
l = -1: FOR i = 440 TO 60 STEP 5 * l: p1 = k1 + i

```

(Folytatás az 57. oldalon.)

Programozzunk Pascalban!

A következő – első látásra kissé hosszúnak látszó – összeállításban a Pascal programnyelv bemutatásával foglalkozunk. Bízvást hihetünk abban, hogy olvasóink különösebb alapismeretek nélkül is alkalmazói szintre juthatnak el, hiszen szerzőnk, Sasvári Gyula évek óta foglalkozik a programnyelvekkel, ráadásul a SZÁMALK Miskolci Tagozatának gyakorlatvezetőjeként tanári tapasztalatokat is szerzett.

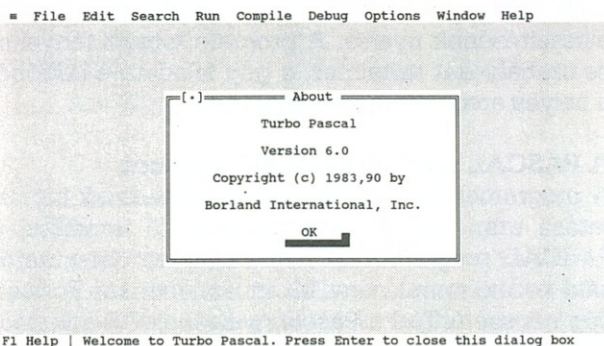
A „rendszerelő elmék” számára különösen fontos lehet az a fejezet, amelyből röviden megismerhetik a programozási nyelvek történetét, majd az IBM gépeken használható Turbo Pascal fordítóprogramot. Az összeállítás második részében már a Pascal gyakorlati alkalmazásáé a főszerep. Jó néhány példával segítjük a könnyebb tanulást, de még több gyakorlati feladatokkal találkozhat az az olvasónk, aki lemezen is megrendeli programmellékletünket.

A bittől a Pascal nyelvig

Bit, byte, memóriacím, utasításkód, relatív ugráscím – ezek a szavak ismerősnek tűnhetnek azoknak, akik ma a számítógépek „belső lelkével” vagy a gépi kódú programozással foglalkoznak. A bevezető szavai manapság már sok programozó számára is idegenül hangzanak, de a négy évtizeddel ezelőtti programozók számára ez volt az egyetlen mód a számítógépekkel való „párbeszédre”. A megtervezett programrészleteket papíron, kódtáblázatok, számológépek segítségével lefordították gépi kódra, majd a programbevitel lassú, nehézkes és hibákkal teli művelete kezdődött. Az egyszerű feladatok – amiket ma akár negyedóra alatt elkészíthetünk – néha csak hetek, hónapok alatt születtek meg. A számítógépes szakemberekben így hamar felmerült az igény a programozási feladatok egyszerűsítésére, felgyorsítására és érthetőbbé tételére. A gépi kódú utasításokat könnyen megjegyezhető – és az utasítás funkciójára utaló – rövidítésekkel (ún. mnemonikokkal) helyettesítették, és egyszerű programok segítségével értelmeztették az így már az ember számára jobban érthető, de a gép számára értelmetlen kódsorokat. Azonban az összetettebb feladatok elkészítése még mindig rengeteg (gyakran alig különböző) lépésből állt. A programozási nyelvek fejlődésének következő lépése így az összetett lépések lehetséges egyszerűsítésével, általánossá tételével folytatódott.

Bábeli (programozási) nyelvzavar

A programozás egyszerűsítése során hamar kiderült, hogy bizonyos műveletek általánosítása sokszorosan felgyorsítja és lerövidíti a programozási munkát. Ilyen műveletek voltak az adatbevitel és -kivitel utasításai, ciklusok, elágazások létrehozása. Az egyes műveleteket már a programban csak



A Pascal bejelentkezik

egyetlen utasítássor helyettesítette. Az így keletkező utasítások nevei már gyakran egyértelműen azonosították a mögöttük rejlő műveletet, nem volt szükség a rövidítések használatára. A használt szavak, kifejezések – igaz, hogy angol nyelven – néha szinte folyamatosan olvashatóvá tették a programok listáit.

A hatvanas évekre a számítástechnika egyre szélesebb körben terjedt el. A rengeteg felhasználási terület új, specializált programozási nyelvek kifejlesztését várta el. Ebben az időszakban több száz féle „nyelvjárás” létezett. Matematikai, statisztikai, gépészeti, adatkezelési nyelvek mellett léteztek úrkutatási, oktatási, orvosi sőt kulturális célú feladatok megoldását támogató programozási nyelvek is. A nyelvek bábeléből napjainkra szinte minden speciális célra írt nyelv kiveszett. A ma használatos ún. általános célú programozási nyelvek sokoldalúsága szükségtelessé tette a nagyszámú egyedi „eszköz” elsajátítását és használatát. A mai nyelvek sem egyformák, használhatóságuk a feladatunktól függ.

Évtizedünk programozási nyelvei

Nézzük át röviden a napjainkban használt programozási nyelvek egyedi vonásait, jellegzetességeit:

BASIC: Általános célú, kezdők számára javasolt, igen elterjedt programozási nyelv. 1968-ban indult el fejlődése, ma szinte minden géptípuson elérhető nyelv.

C: Általános célú nyelv, de bonyolult, gépközel feladatok megoldására használják elsősorban. Segítségével ma főként más programozási nyelveket, operációs rendszereket készítenek.

DBASE: Adatbázis-kezelésre specializálódott nyelv. Sok – vele szinte teljesen kompatibilis – követője alakult ki, közepes bonyolultságú adatkezelési feladatokhoz kiválóan használható.

FORTH: 1970-ben kifejlesztett nyelv. A szokásostól eltérő szerkezete miatt nehezen terjedt el. Kifejlesztője – egy amerikai csillagász – berendezések vezérlésére találta ki a nyelvet.

PASCAL: 1968-ban fejlesztették ki a zürichi egyetemen, pontosan definiált felépítése és szerkezete miatt könnyen elsajátítható, oktatási nyelvként is gyakran alkalmazzák. Napjainkra az egyik legjobban elterjedt általános célú programozási nyelv.

PROLOG: Logikai feladatok, szakértői rendszerek elkészítésének nyelve. A program ismert tényeket és szabályokat tartalmaz, a gép feladata e kettőből a helyes eredmények megtalálása.

A PASCAL nyelv az IBM PC gépeken

A programozási nyelvek rövid történetének bemutatása után térjünk rá sorozatunk fő témájára, a PASCAL programozási nyelv alapvető bemutatására és megtanulására. Mi itt az amerikai Borland cég népszerű Turbo Pascal nyelvének 6.0-s verzióját fogjuk használni, de természetesen szinte bármilyen PC-s Pascal fordítóprogram használható. A fordítóprogram szolgáltatásainak bemutatása után a sorozat többi részében csak a Pascal nyelv megismerésével foglalkozunk.

Induljunk el!

A Turbo Pascal az XT és AT gépeken egyaránt használható, de a 6.0-s verzióhoz már merevlemez (winchester) is szükséges. A fordítóprogramot a \TP nevű könyvtárban találhatjuk meg, s a TURBO névre hallgat. Így elindítását az alábbi két parancsral végezhetjük el:

```
cd c:\tp
turbo
```

A program betöltődése után például az alábbi bejelentkező képet láthatjuk:

A Turbo Pascal „akcióban”

The screenshot shows the Turbo Pascal 6.0 environment. At the top is a menu bar: File, Edit, Search, Run, Compile, Debug, Options, Window, Help. Below the menu bar, there are several windows:

- Editor Window (KEPLOPO.P):** Contains code:


```
{ $F+ }
{ $M 8192,0,8192 }

Uses Dos;

... \...\APP.

{*****}
{
  Turbo Pascal
  Turbo Vision
}

ASCIITAB.PAS 3
{*****}
{
  Turbo Pascal 6.0
  Turbo Vision Demo
  Copyright (c) 199
}
```
- Help Window (Integer types):** Contains the text: "Turbo Pascal provide predefined Integer t denotes a specific s whole numbers, as sh". Below it is a table:

Type	
Shortint	-
Integer	-3276
Longint	-214748364
Byte	
Word	
- Options Window:** Shows settings for Compiler, Memory sizes, Linker, Debugger, Directories, Environment, and Preferences.
- System Information Window:** Shows CPU information: CPU 4, DX 0000, BX 0000, CS 0000, DS 0000, ES 0000, SS 0000, s=0 o=0, a=0 d=0.

At the bottom, a command prompt shows: C:\CIKKSOR>keplopo.exe and C>c:\tp\turbo

A fordítóprogram az alábbi szolgáltatásokat tartalmazza:

Beépített szövegszerkesztő: egy időben több szöveget szerkeszthetünk vele.

Fordítóprogram: A lefordított program memóriába vagy mágneslemezre kerül.

Hibakereső: Segítségével a programot sorról sorra tanulmányozhatjuk, megnézhetjük működését.

HELP: Bárhol segítséget kérhetünk az <F1> billentyű lenyomásával. Az utasítások táblázata a <SHIFT F1> lenyomására jelenik meg.

A program használható egerrel is. Segítségével az ablakok tetszőleges méretűre állíthatók és mozgathatók.

A program főmenüje az alábbi fontosabb lehetőségeket tartalmazza:

File – Forrásszövegek és könyvtárak kezelése. Itt lehet a programlistákat betölteni, kimenteni, könyvtárat választani, szöveget nyomtatni, kilépni a programból.

Edit – Szövegszerkesztési funkciók. A szövegdarabok másolását, törlését segítő menü.

Search – Szövegkeresés és kicserélés.

Run – Programfuttatás és nyomon követés. Futtathatjuk az elkészült programot, vagy elindíthatjuk a hibakeresési, nyomonkövetési folyamatot.

Compile – Program lefordítása. A programok lefordítását, szerkesztését végző menüpont.

Debug – Nyomon követés. A nyomon követés során megnézhetjük az egyes változók értékeit, törléspontokat tehetünk a programokba, stb.

Options – Opciók beállítása. Beállíthatjuk a fordítóprogram egyes paramétereit, a szövegszer-

F1 Help | Specify desktop settings

kesztő és eger működését, szabályozhatjuk a program színeit stb.

Window – Ablakkezelés. A menü segítségével lehet az egyes ablakokat lezárni, elrendezni, mozgatni stb.

Help – Segítségkérés. A menü többféle szempontból ad segítséget, tájékoztatást a programról.

A „programozott oktatás” szempontjai szerint most a „tréning” következne. Érdeemes gyakorolni a fordítóprogram kezelését, használatát, hogy a tanuláskor bemutatott programok begépelése, kipróbálása már ne jelentsen problémát.

A Pascal program szerkezete

Egy Pascal nyelvű program szerkezete az alábbi lehet:

Adattípusok,	A program bevezető szakaszában határozzuk meg a szükséges adattípusokat, változókat, konstansokat.
Konstansok,	
Változók	A második szakasz az eljárások, függvények (alprogramok) leírásának helye.
Alprogramok	
Főprogram	A tényleges főprogram a programlista legvégén található meg.

A programok felépítése során fontos szabály, hogy csak azokat a változókat, alprogramokat használhatjuk, amiket már meghatároztunk.

Változók

A programokban használt változóknál előre meg kell határozni a változó nevét és típusát. A névben állhatnak betűk és számok, de betűvel kell kezdődnie. A Pascal az alábbi (alapértelmezett) típusokat tartalmazza (a* jelűek nem minden fordítóprogramban szerepelnek!):

- string** – max. 255 betűből álló szöveg
- string[n]** – max. n betűből álló szöveg (0 < n <= 255)
- integer** – egész szám -32767 és +32768 között
- word** – egész szám 0 és 65535 között
- byte** – egész szám 0 és 255 között
- char** – egy betű
- real** – valós szám
- boolean** – logikai igaz vagy hamis érték
- * **shortint** – egész szám -127 és +128 között
- * **longint** – 2147483648 és +2147483647 közötti egész
- pointer** – memóriacím

Ezekkel az adattípusokkal minden létező feladat megoldható. A típusok közüli választást megelőzi a feladat pontos áttanulmányozása. Mi csak a három leggyakoribb adattípussal foglalkozunk, ezek a String, Integer és Real típusú adatok.

Első programunk

Írjunk egy rövid programot, mely két számot összead, majd tájékoztat az eredményről:

```

Var
  A: Integer;
  B: Integer;
  C: Integer;
BEGIN
  A:= 12;
  B:= 23;
  C:= A + B;
  WriteLn('Az összeg:',C);
END.

```

A program elején létrehozunk három változót. A változókat létrehozó szakaszt a Var szó vezeti be. A Var szó használata a változók meghatározása előtt kötelező, ha elfelejtjük kiírni, hibaüzenetet kapunk.

A három változó rendre az A, B és C nevekre hallgat, s mindegyikben egész számot fogunk tárolni.

A főprogram a BEGIN szónál kezdődik. A BEGIN szintén elhagyhatatlan, ez jelzi a ténylegesen végrehajtható programsorok kezdetét.

A következő két sor az A és B változók induló értékét határozza meg. Figyeljük meg, hogy az értékadásnál az ún. „legyen egyenlő” jelet (:=) használtuk. Ne keverjük össze az egyenlő (=) jellel!

Az A és B értékének meghatározása után kiszámoljuk az A + B értékét, s ez a C változóba kerül.

A három értékadás után már csak az eredményt kell kiírni a képernyőre. Ezt a WriteLn utasítás végzi el. Az utasítás szövegeket és számokat is meg tud jeleníteni. A szövegeket egyszeres idézőjelbe kell tenni ('). Ha egyszerre több dolgot szeretnénk megjeleníteni, az egyes kiírandókat vesszővel válasszuk el egymástól. A programunkban szereplő WriteLn('Az összeg:',C); sor így az alábbi sort írja ki:

```
Az összeg:35
```

Programunkat az END szóval fejeztük be. Ez a sor egyben a program vége is, ezért még egy pont (.) is követi, így lesz teljes a program végét jelző END. szó. Az END – a Var ill. BEGIN szavakhoz hasonlóan – kötelező használatú szó.

Figyeljük meg, hogy a Var, BEGIN és END. sorának kivételével valamennyi sort a ; jel zárt le. A Pascal programokban utasításainkat mindig a pontosvessző fejezi be.

A program futtatása a Run főmenüpont Run menüpontjának választásával, vagy a Control-F9 billentyű-kombináció használatával indítható el. Hibás programlista esetén a gép jelzi a hiba helyét és okát. Ha a programunk befejezte működését, az ALT-F5 gombbal tudjuk megtekinteni a kapott eredményt.

Próbáljuk meg most átalakítani első programunkat! A kiírt összegből nem tudjuk megállapítani, hogy melyik két számot adtuk össze. Javítsuk ki a WriteLn utasítást úgy, hogy a két kiinduló számot és az összeget is megjelenítse. Ehhez a WriteLn sorát az alábbira kellene kicserélni:

```
Writeln(A,' és ',B,' összege:',C);
```

Jó lenne az is, ha a program kérdezné meg, hogy milyen két számot akarunk összeadni. Ehhez az A és B értékadását kell kijavítani: a két számot a billentyűzetről gépeljük be, ne pedig állandó értékek legyenek. Az adatok manuális bevitelét a ReadLn(változó); utasítás végzi el. A programban az alábbi módosítások kellene:

```
A:= 12; helyett a ReadLn(A); utasítás, és a
B:= 23; helyett a ReadLn(B); utasítás.
```

A program futásakor semmi nem tájékoztat arról, hogy adatokat kellene beírunk, ezért még bővítjük ki programunkat az alábbi módon:

```
Write('Mennyi az A értéke?'); ReadLn(A);
Write('Mennyi a B értéke?'); ReadLn(B);
```

Láthatjuk, hogy egy sorba – pontosvesszővel elválasztva – több utasítást is írhatunk.

Szövegek

Az alábbi kis program – az eddig bemutatott utasítások segítségével – megkérdezi a nevünket, majd üdvözl:

```
Var
Neved: String;
BEGIN
Neved:= ' ';
Write('Mi a neved?'); ReadLn(Neved);
Writeln('Üdvözöllek, ', Neved, '!');
END.
```

Láthatjuk, hogy a Neved változó szöveges típusú, s az adatbevitel és adatkiírás utasítása ugyanúgy használható szövegek és számok esetén is. A Neved változó kezdeti értékadása azt jelenti, hogy a változó semmilyen szöveget nem tartalmaz.

Figyeljük meg a Write / WriteLn ill. a Read / ReadLn közötti különbséget! Így lehet vezérelni a kiíró és a beolvasó utasítások képernyőn való elhelyezkedését. A WriteLn és ReadLn végrehajtása után a kiírás illetve beolvasás új képernyősorban kezdődik, míg a Write és Read esetén nem.

A sorozat következő részéig gyakorlásként készítsünk egyszerű programokat például az alábbi feladatokra:

- Kör kerületének és területének kiszámolása.
- Három szám átlagának meghatározása.
- Az $((A-3) * B) + (C + 3)$ képlet kiszámolása.

Most, hogy túl vagyunk az első program elkészítésén, ismerkedjünk meg egy gyakori programvezérlési eszközzel: a feltételes elágazással és annak Pascal-beli formájával.

Vizsgálatok, elágazások

A már elkészített kis programunkat fogjuk átalakítani, kibővíteni úgy, hogy megismerhessük a feltételes elágazásokat.

Mi történik ebben a programban, ha valaki nem ír be semmit, csak lenyomja az ENTER billentyűt? A Neved változó értéke nem fog változni. Hogy

lehetne felismerni, hogy begépelünk-e valamilyen szöveget, vagy sem?

Nézzük meg az alábbi – módosított – programot!

```
Var
Neved: String;
BEGIN
Neved:= ' ';
Write('Mi a neved?'); ReadLn(Neved);
if Neved <> ' ' then
  Writeln('Üdvözöllek, ', Neved, '!');
Writeln('Vége. ');
END.
```

Ez a program már tartalmaz egy kis vizsgálatot is. Ha nem írtunk be semmilyen nevet, nem fog minket üdvözölni. A vizsgálatot az

```
if Neved <> ' ' then ...
utasítás végezte el. Így lehetne olvasni:
Ha a Neved nem üres, AKKOR csinálunk valamit;
```

A fenti program tehát csak akkor hajtja végre az üdvözlő sort, ha az if után szereplő (logikai) kifejezés igaznak bizonyul. A „Vége” szöveg viszont minden esetben megjelenik a képen. Ez azt jelenti, hogy az elágazás igaz esete csak egy sort tud végrehajtani (a példában a Writeln('Üdvözöllek, ', Neved, '!'); sort).

Hogyan lehetne megoldani, hogy ha a feltétel igaz, több sorból álló programrészletet is végrehajthassunk? A megoldást a következő egyszerű részlet mutatja meg:

```
if A < 0 then BEGIN
  Writeln('A változó értéke negatív!');
  Write('Írj be egy új értéket!');
  ReadLn(A);
END;
```

Látható, hogy csak annyi a dolgunk, hogy a végrehajtható sorokat egy BEGIN – END szópár közé helyezzük. Az utasítások ilyen összefogására gyakran lesz szükségünk.

Alakítsuk tovább kis programunkat az alábbi módon: cseréljük ki a vizsgálatot és elágazást így:

```
if Neved = ' ' then BEGIN
  Writeln('Neveletlen!');
  Writeln('Még a nevét sem mondja meg...');
END
else BEGIN
  Writeln('Üdvözöllek ', Neved, '!');
  Writeln('Jó munkát!');
END;
```

Próbáljuk ki a programot úgy, hogy beírunk egy nevet, és úgy is, hogy nem...

Az elágazás teljes szerkezete tehát így néz ki:

```
if feltétel then
  Csinálunk valamit, ha igaz volt else
  Csinálunk mást, ha nem volt igaz;
```

Jegyezzük meg, hogy ilyen esetben az else szó előtti utasítás (vagy END) mögött nincs pontosvessző!

Feladatok az elágazások és feltételek gyakorlására:

- Válasszuk ki három szám közül a legkisebbet!
- Adott az A, B és C változó. Döntsük el, hogy a C az A és B között van-e.

Többirányú elágazások

Néha szükség lehet arra, hogy programunk ne csak két irányba ágazhasson el – például ha a hét minden napjához más és más tennivalót rendelünk. Az egyes napok szerinti elágazást az eddigiek szerint így írhattuk volna meg:

```
if MaiNap = Hetfo then Amit_Hetfon_Csinalunk;
if MaiNap = Kedd then Amit_Kedden_Csinalunk;
...
if MaiNap = Vasarnap then Amit_Vasarnap_Csinalunk;
```

Ez így működik ugyan, de nem túl elegáns. Lehet máshogy is? Természetesen lehet, a többirányú elágazással:

```
case MaiNap of
Hetfo: Amit_Hetfon_Csinalunk;
Kedd: Amit_Kedden_Csinalunk;
...
Vasarnap: Amit_Vasarnap_Csinalunk;
END;
```

Ez a szerkezet áttekinthetőbb, és jobban megmutatja a többirányú elágazás lényegét. A szerkezet általános alakja az alábbi:

```
case változó of
Ertek1: Valami1;
Ertek2: Valami2;
...
ErtekN: ValamiN;
else EgyikSemVolt
END;
```

A „változó” csak egész szám, vagy betű típusú karakter lehet. Ha az elágazás végrehajtásakor a „változó” értéke egyik megadott elágazáshoz sem jó, akkor az else utáni sorok hajtódnak végre. (Az else ág elhagyható.) Nézzünk meg egy másik példát:

```
Var
A: Integer;

BEGIN
Write('Írj be egy számot:'); ReadLn(A);
case A of
0: Writeln('Nullát írtál be');
1: Writeln('Szerintem egyes volt');
2: Writeln('Ez egy kettes!');
```

```
else Writeln('Valami mást kaptam!');
end;
END.
```

Itt az ideje, hogy megismerkedjünk a ciklusokkal, és ismétlési lehetőségekkel. Itt három módját tanulhatjuk meg a ciklusok elkészítésének.

Ciklusok, ismétlések

Eddig csak olyan programokat készíthettünk, melyek úgynevezett lineáris lefutásúak, vagyis a program első sorától az utolsóig minden utasítás legfeljebb egyetlen alkalommal hajtódott végre. Ha valamit többször kell megismételni, és a tevékenység megkezdése előtt már tudjuk azt, hogy hányszor, használhatjuk az alábbi ciklusszervező utasítást:

```
for változó:= kezdet to vég do
Csinálunk valamit;
```

A végrehajtás menete a következő lesz:

- 1.) A „változó” egyenlő lesz a „kezdet” értékével.
- 2.) Ha a „vég” nagyobb, vagy egyenlő mint a „változó” értéke, végrehajtódik az, amit ismételni szeretnénk.
- 3.) Ezután a „változó” értéke eggyel nő, s ismét a 2.ponttól folytatódik a végrehajtás.

Nézzük meg ezt egy rövid programban:

```
Var
A: Integer;
B: Integer;
C: Integer;
BEGIN
Write('A számolás kezdete:'); ReadLn(A);
Write('A számolás vége:'); ReadLn(B);
for C:= A to B do begin
Writeln('C értéke most ',C);
Writeln('—————');
end;
END.
```

Ha a cikluson belül csak egyetlen utasítást akarunk ismételni, a végrehajtandó utasítást övező BEGIN – END elhagyható.

Az alábbiakat jegyezzük meg a for ciklusokkal kapcsolatban:

- 1.) A ciklusváltozó típusa csak egész számokat tartalmazó típus, vagy betű (Char) lehet.
- 2.) A ciklus mindig csak egyesével lépeget.

Két rövid példa a ciklusokra:

```
Var
C: Char;
BEGIN
Writeln('Kisbetűk:');
for C:= 'a' to 'z' do
Write(C);
Writeln;
END.
```

A fenti program bemutatta, hogyan lehet betű típusú változó a ciklus számlálója. A következő pedig a visszafelé számoló ciklust ismerteti.

```
Var
A: Integer;
BEGIN
for A:= 1 to 10 do
    Writeln(A);
for A:= 10 downto 1 do
    Writeln(A);
END.
```

A ciklus maga is utasításnak számít, így lehetőség van tetszőleges számú ciklus egybeépítésére. Értelemszerűen a ciklusok belseje tartalmazhat vizsgálatokat, elágazásokat is. A következő példa egy ilyet mutat be. Hogy mit csinál, annak megfejtését az olvasókra bízunk:

```
Var
i,j: integer;
BEGIN
for i:= 1 to 10 do begin
    for j:= 1 to 10 do
        if i < j then
            Write(' | ')
        else if i = j then
            Write('J ')
        else
            Write('- ');
    Writeln;
END;
END.
```

A program elemzése során néhány újdonságot is fel lehet fedezni. Az „újdonságok” az eddig leírtakból következnek, s bepillanthatunk velük a Pascal egyik kiváló tulajdonságába: az elágazások, ciklusok tetszőleges mélységben egymásba ágyazhatók. Ennek a kis programnak az alapos áttanulmányozását, teljes megértését külön is ajánlom, sokat lehet tanulni belőle.

Ciklusok másképpen

Gyakran előfordulhat, hogy valamit többször kellene megismételni, de előre nem tudjuk pontosan, hogy hány alkalommal. Mit lehet ilyenkor tenni? Az alábbi program bemutat egyet a használható lehetőségek közül:

```
Var
A: Integer;
BEGIN
Writeln('Írj be számokat! Ha 0-t írsz, vége!');
repeat
    Write('A szám?'); Readln(A);
    Writeln('Most ',A,'-t írtál');
until A = 0;
END.
```

A ciklus szerkezete az alábbi:

```
repeat
csinálj valamit
until feltétel
```

A ciklus addig működik, amíg a „feltétel” nem igaz. Figyeljük meg, hogy a repeat – until között akár több utasítás is szerepelhet, de itt nem kell a BEGIN – END pár használata (persze nem tilos). Természetesen ez a ciklus is tetszőlegesen egymásba ágyazható más ciklusokkal és elágazásokkal is. Ez a ciklus az úgynevezett „hátultesz-telő” ciklus. (A kilépés feltétele a ciklus végén található.)

Nézzük meg végezetül a harmadik lehetséges cikluskészítési módszert! Ezt is egy rövid programmal mutatjuk be:

```
Var
A, B: Integer;
BEGIN
A:= 0;
while A < 20 do begin
    WriteLn('Az aktuális összeg:',A);
    Write('Mennyit adjak hozzá?'); Readln(B);
    A:= A + B;
END;
Writeln('Az összeg nagyobb 20-nál!');
END.
```

A while .. do ciklus szerkezete az alábbi:

```
while feltétel do
Csinálunk valamit
```

Ez a ciklus az úgynevezett „előtesztelő” ciklus (a végrehajtás feltétele a ciklusba belépés előtt található), és addig hajtja végre az ismétlődő utasításokat, amíg a „feltétel” nem teljesül. Ha a feltétel hamissá válik, befejeződik a ciklus.

Eddig megismerhettük a programok vezérléséhez szükséges legfontosabb utasításokat, tudnivalókat. Most az adatok egyféle csoportosítását ismerhetjük meg, ami nélkülözhetetlen lesz két egyszerű, de érdekes és később hasznossá tehető program megírásához.

Összegyűjtött adatok

Tegyük fel, hogy a havi zsebpénzünket szeretnénk egy kis programmal nyilvántartani. Hogyan lehetne a 12 havi összegeket úgy tárolni, hogy később könnyen összeadhassuk azokat? Semmiképpen ne kezdjük így:

```
Var
Januar, Februar, Marcius ....
Oktober, November, December: Integer;
```

Elég nehéz lenne így kezelni az adatainkat. Használjunk inkább tömböket! A tömbökben

egy azonos név alatt több egyedi adatot tudunk kezelni. Nézzük meg az alábbi változó létrehozását:

```

Var
  Tomb: array[1..10] of Integer;
  Az így létrehozott tömb tíz darab egész számot
  tartalmaz majd. Az egyes számokra a Tomb[1],
  Tomb[2] ...módon hivatkozhatunk. A tömb létre-
  hozásának általános alakja az alábbi:

```

```

  Nev: array[alsóhatár .. felsőhatár] of adattípus;
  Nézzük meg tehát az egyszerű zsebpénz-nyil-
  vántartót!

```

```

Var
  Zsebpénz: array[1..12] of Real;
  Koltottem: Array[1..12] of Real;
  I: Integer;
  Ossz1: Real;
  Ossz2: Real;
BEGIN
  Writeln('Írd be havonként az adatokat');
  for i:= 1 to 12 do begin
    Write(i, ' hónapban a zsebpénzed?');
    ReadLn(Zsebpénz[i]);
    Write(i, ' hónapban elköltöttél?');
    ReadLn(Koltottem[i]);
  END;
  Ossz1:= 0;
  Ossz2:= 0;
  for i:= 1 to 12 do begin
    Ossz1:= Ossz1 + Zsebpénz[i];
    Ossz2:= Ossz2 + Koltottem[i];
  END;
  Writeln('Az évben ' ,Ossz1, ' Ft-od volt');
  Writeln('Az évben ' ,Ossz2, ' Ft-ot költöttel');
  if Ossz1 - Ossz2 > 0 then
    Writeln('Maradt: ' ,Ossz1 - Ossz2, ' Ft-od')
  else Writeln('Jól gazdálkodtál???');
END.

```

Második hasznos programunkkal ismerőseink neveit rendezhetjük ABC rendbe. A program megkérdezi a neveket majd rendezés után kiírja az eredményt. ...

```

Var
  Nevek: array[1..10] of String[20];
  Nev: String[30];
  Csere: String[30];
  I,J,S: Integer;
BEGIN
  Writeln('Írd be a neveket! (max. 10 lehet)');
  Writeln('Ha nem adsz nevet, kezdem a rendezést.');
```

```

  I:= 1;
  Write(I, ' név:'); ReadLn(Nev);
  while (I < 10) and (Nev <> ' ') do begin
    Nevek[I]:= Nev;
    I:= I + 1;

```

```

  Write(I, ' név:'); ReadLn(Nev);
END;
I:= I - 1;
S:= 1;
while S < I do BEGIN
  if Nevek[S] > Nevek[S+1] then begin
    Csere:= Nevek[S];
    Nevek[S]:= Nevek[S+1];
    Nevek[S+1]:= Csere;
    S:= 1;
  END
  else S:= S + 1;
END;
Writeln('A nevek rendezve:');
for J:= 1 to I do
  Writeln(Nevék[J]);
END.

```

Feladatként próbáljuk meg végiggondolni a program működését, esetleg papíron utánozzuk a rendezés folyamatát.

Természetesen lehetőségünk van táblázatok készítésére is. Ekkor egy adatot nem egyszerűen a sorszáma, hanem a táblázatbeli sor- és oszlopszáma határoz meg. Egy táblázat létrehozása lehet az alábbi:

```

Var
  Nevek: array[1..10, 1..5] of String[20];

```

Alprogramok

Az apró részletekből folyamatosan kialakuló „nagyobb” programokban gyakran előfordulnak egyforma ismétlődő, vagy egymástól csak kevésben különböző programrészletek is. Nézzük, hogyan lehet az ilyen, többször előforduló programrészletek unalmas megírását kikerülni. Az egyforma (vagy nagyon hasonló) részletek kigyűjtése, alprogrammá (vagy „Pascalosabb” nevén: eljárásokká) alakítása során időt és memóriát takaríthatunk meg. Vágjunk neki, nézzük meg rögtön az alábbi kis programot:

```

procedure VonalatRajzolok;
BEGIN
  Writeln ('—————');
END;
BEGIN
  VonalatRajzolok;
  Writeln('Ez a kis program');
  VonalatRajzolok;
  Writeln('vonalat húz');
  VonalatRajzolok;
  Writeln('az egyes sorok közé');
  VonalatRajzolok;
END.

```

Az újdonság az, hogy készítettünk egy egyszerű eljárást, ami a VonalatRajzolok nevet kapta. Az eljárás elkészítése az alábbi szerkezettel oldható meg:

```
procedure EljárásNév;
BEGIN
```

Itt írjuk le, hogy mit csinálunk az eljárásban.
END;

Innentől kezdve az elkészült eljárás bárhol használható, egyszerűen csak a nevét kell leírni.

Az előző példában szereplő eljárás mindig ugyanazt csinálja: húz egy rövid vonalat. Azonban arra is van mód, hogy kívülről beavatkozzunk az eljárás működésébe. Ezt paraméterezésnek nevezük. Például:

```
Var
I: Integer;
procedure BetuRajzolo(Mit: Char; Mennyit: Integer);
BEGIN
  for i:= 1 to Mennyit do
    Write(Mit);]
END;
BEGIN
  BetuRajzolo('A', 2);
  BetuRajzolo('B', 3);
  BetuRajzolo('C', 5);
  BetuRajzolo('-', 10);
END;
```

A program az alábbi eredményt adja:

```
AA
BBB
CCCCC
```

Láthattuk, hogy a BetuRajzolo alprogram két paramétert kapott: Melyik betűből hány darabot kell kiírnia. Az eljárás belsejében (de csak ott) a Mit és Mennyit változókat ugyanúgy kezelhetjük, mint a program elején létrehozott I változót.

A paramétereket a függvény létrehozásakor zárójelek között a nevük és típusuk megadásával soroljuk fel. Később, a függvény használatakor már csak a paraméterek felsorolása szükséges.

Előfordulhat, hogy egy gyakran használandó számolási műveletet (képletet) szeretnénk alprogrammá alakítani. Ekkor gondoskodnunk kell a kiszámolt eredmény „visszaadásáról”, vagyis arról, hogy a kapott eredmény a program tudomására jusson. Nézzük meg az alábbi rövid programot, amely a kör kerületét számolja ki alprogram, vagy pontosabb néven függvény segítségével:

```
Var
R: Real;
function KorKerulet(Sugar: Real): Real;
BEGIN
  KorKerulet:= 2 * Sugar * 3.1415;
END;
BEGIN
  repeat
```

```
Write('Írd be a kör sugarát! 0 = vége');
ReadLn(R);
if R <> 0 then]
  WriteLn('Kerület:',KorKerulet(R));
until R = 0;
END.
```

A függvény általános alakja az alábbi:
function függvénynév: eredménytípus;

```
BEGIN
  itt számolunk;
  függvénynév:= a kiszámolt eredmény;
END;
```

Természetesen a függvények is kaphatnak paramétereket, mint ahogy ez a példából kiderült. A függvény által visszaadott eredmény tetszőleges alapértelmezett típus lehet. Így írható olyan függvény, ami szám paramétert kap, de szöveget ad vissza:

```
Var
N: Integer;
function NapNev(Nap: Integer): String;
BEGIN
  case Nap of
    1: NapNev:= 'Hétfő';
    2: NapNev:= 'Kedd';
    3: NapNev:= 'Szerda';
    4: NapNev:= 'Csütörtök';
    5: NapNev:= 'Péntek';
    6: NapNev:= 'Szombat';
    7: NapNev:= 'Vasárnap';
    else NapNev:= ' ';
  END;
END;
BEGIN
  Write('Hányadik nap?'); ReadLn(N);
  WriteLn(NapNev(N));
END.
```

A Pascal nyelv alapismereteit röviden bemutató összeállítás végén sikeres programozást és tanulást kívánunk minden Olvasónak. Rengeteg lehetőséget és ötletet találhat bárki, aki kísérletező kedvvel és kis elszántsággal foglalkozik a Pascal nyelvvel. Ha bárkinek itt meg nem választott kérdése, problémája van, írjon Szerkesztőségünknek, s a felmerült kérdésekre az újságban alkalmanként visszatérünk. Ha pedig az alapismereteken túljutva az Olvasók érdeklődnek a Pascal komolyabb megismerése iránt, ezt is jelezzék nekünk. Ilyen lehetséges témák pl. a bonyolultabb adatszerkezetek használata, az adatállományok (file-ok) kezelése, a grafikus lehetőségek és még sok más.

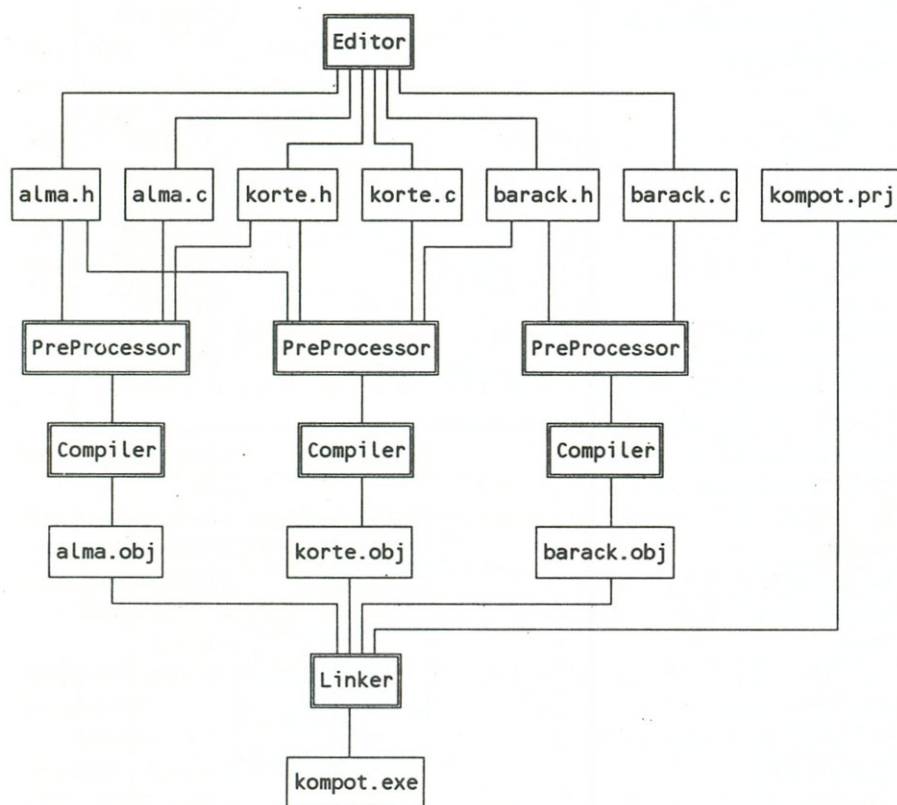
Sasvári Gyula

(Külön megrendelhető lemez mellékletünkön a fenti gyakorló példákon kívül részletesebb feladatokat is közlünk.)

C++ – avagy

a szükséges plusz

Sorozatunknak ebben a fejezetében a Borland szoftverház C++ 2.0 fordítóprogramjával ismerkedhetünk meg, amely a C programok írásához szükséges. Általános esetben egy C program elkészítésének folyamata a következő:

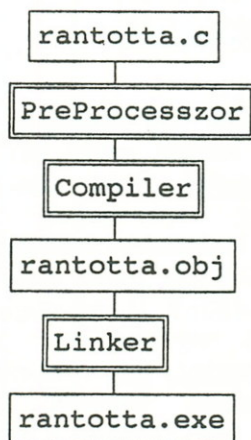


Az Editor, a Preprocessor, a Compiler és a Linker a fordító része. Az Editorral hozhatjuk létre a programszöveget .c és .h kiterjesztéssel. A Preprocessor és a Compiler object (.obj kiterjesztésű) file-okat hoz létre, amelyeket azután a Linker futtatható programmá szerkeszt össze. Nem kell azonban megjedni, kisebb programoknál egyszerűbben is eljáráhatunk.

A fordító egy menürendsze-

ren, illetve ablakokon keresztül kommunikál velünk. Néhány speciális esettől eltekintve, az ablakokban a programjaink szövege található, míg a tetejükön láthatjuk, melyik file van bennük. Ha több ablak van megnyitva, az aktuális (amelyikben éppen javíthatunk) kerete dupla és fényesebb a többinél. Mindegyik jobb felső sarkában egy szám is van; ablakváltáshoz az **ALT**-ot és a kiválasztott ablak számát kell egyszerre leütni vagy végig-sétálunk rajtuk az **F6** szorgalmas nyomogatásával. Még egy lehetőség van a váltásra: az **ALT+0** hatására egy menüből választhatunk a megnyitott és a már bezárt ablakok közül.

Ha új file-t akarunk megnyitni, akkor ezt legegyszerűbben az **F3** billentyű segítségével tehetjük meg, de a menürendszerben a file almenü open pontja is ugyanezt a hatást eredményezi. A felső menüt az **F10**-zel tehetjük aktívá, míg **ALT+f** hatására rögtön a file almenüben találjuk magunkat. Az ablakban lévő file tartalmát **F2**-vel vagy a file almenü save menüpontjával (a továbbiakban file/save file/open ...) menthetjük le. Ablakot becsukni **ALT+F3** Window/Close-zal, maximálisan kinyitni, illetőleg visszaállítani az eredeti nagyságát pedig az **F5** billentyűvel lehet. Az ablakokat tetszés szerint átméretezhetjük és tologathatjuk a képernyőn **CTRL+F5** után. (A nyilakkal lehet tologatni, a **SHIFT+** nyilakkal pedig átmére-



tezni.) Az ablakokat Window/Tile menüponttal egymás mellé, aWindow/Cascade-dal pedig egymás mögé tehetjük a képernyőn.

Az aktuális ablakban lévő programot a Run/Run menüponttal vagy csak egyszerűen **CTRL+F9**-cel indíthatjuk, ekkor a fordító automatikusan lefordítja, összeszerkeszti és elindítja a programot, ha hibát talált valahol benne, leáll és kiírja, hogy mi nem tetszik

neki. A fordító „van olyan kedves”, hogy nem csak a hibákért szól, hanem figyelmeztet is minket, ha valami gyanús dolgot műveltünk (általában igaza is van és tényleg nem úgy akartuk). Nagy segítség ez, hiszen amint ezt egy bölcs programozó megfogalmazta: „*C-ben mindent szabad, csak tévedni nem!*”

Ahhoz, hogy igazán urai legyünk a C fordítónak, ismerkedjünk meg elsősorban az Editorral, ugyanis ezzel lesz a legtöbb dolgunk.

Mozgás az ablakban és a file-ban

A kurzormozgató nyilak hatására a kurzor elmozdul a kívánt irányba;

PgUp, **PgDn** egy ablakmagasságnyi felfele vagy lefele ugrik a file-ban;

Home, **End** a kurzor a sor elejére, illetve a végére ugrik;

(CTRL+jobbra-), **balra nyíl** a kurzor a következő, illetve az előző szó elejére ugrik;

(CTRL+Home), **(End)** a kurzor az ablak tetejére, illetve aljára ugrik

Del egy karaktert töröl jobbra; **BackSpace** egy karaktert töröl balra;

Ins ki/bekapcsolja az Insert módot. Ha be van kapcsolva, akkor a begépelte szöveg beszűrődik a kurzor helyére, ha ki van kapcsolva, akkor a már ott lévő szöveg átíródik.

miközben a **SHIFT**-et lenyomva tartjuk.

(CTRL+K)+h a blokk kijelölésének megszüntetése;

(CTRL+K)+y blokk törlése;

(CTRL+K)+c blokk másolása;

(CTRL+K)+v blokk átmozgatása;

(CTRL+K)+w blokk kiírása egy file-ba;

(CTRL+K)+r blokk beolvasása a file-ból;

(CTRL+Ins) blokk másolása pufferba;

(SHIFT+Del) a blokk tartalmának törlése, pufferba helyezése;

(SHIFT+Ins) a pufferben tárolt blokk visszatöltése;

(CTRL+K)+i a blokk tartalmát egygel jobbra görgeti;

(CTRL+K)+u a blokk tartalmát egygel balra görgeti;

Törlés

(CTRL+y) egész sor törlése;

(CTRL+q)+y törlés a kurzortól a sor végéig;

(CTRL+t) törlés a kurzortól a következő szó elejéig;

(CTRL+q)+f szöveg keresése
(CTRL+l) a következő ilyen szöveg;

(CTRL+q)+a szöveg kicserélése más szövegre.

Opciók

Ahhoz, hogy a fordító úgy működjön, ahogy mi szeretnénk, nem pedig úgy, ahogy ő akarja, meg kell mondani neki, mit hogyan csináljon, be kell állítani az opcióit. (Ezt meg kell csinálni, ha nem akarjuk saját életünket megkeseríteni.) A sok megoldás közül a legcélszerűbbeket soroljuk fel, illetve mutatjuk be a következő képernyőábrákon:

[*] Advanced Code Generation

Floating Point	Options
() None	[X] Generate underbars
(o) Emulation	[X] Line numbers debug info
() 8087	[X] Debug info in OBJs
() 80287	[X] Fast floating point
	[] Fast huge pointers
Instruction Set	[] Generate COMDEFS
(o) 8088/8086	[] Automatic far data
() 80186	
() 80286	
	Far Data Threshold 32767

OK Cancel Help

[*] Entry/Exit Code Generation

Prolog/Epilog Code Generation

(o) DOS standard

() DOS overlay

() Windows all functions exportable

() Windows explicit functions exported

() Windows smart callbacks

() Windows DLL all functions exportable

() Windows DLL explicit functions exported

Calling Convention Stack Options

(o) C [X] Standard stack frame

() Pascal [X] Test stack overflow

OK Cancel Help

[*] Code Generation

Model	Options
() Tiny	[X] Treat enums as ints
() Small	[] Word alignment
() Medium	[] Duplicate strings merged
() Compact	[] Unsigned characters
() Large	[] Pre-compiled headers
(o) Huge	

Assume SS equals DS

(o) Default for memory model

() Never

() Always

Defines

More... OK Cancel Help

Blokkműveletek

Egy blokk egy kijelölt szöveget jelent, amivel aztán különböző műveleteket lehet végezni – másolni, átmozgatni, törölni, file-ba lementeni, betölteni...

(CTRL+K)+b blokk eleje,
(CTRL+K)+k blokk vége.

Blokkot egyszerűbben is ki lehet jelölni, ha a pozicionáló billentyűket (**nyilak**, **PgUp**, **PgDn**, **End**, **Home**) használjuk,

[*] C++ Options

C++ Virtual Tables Use C++ Compiler
 Smart CPP extension only
 Local C++ always
 External
 Public

Options
 Out-of-line inline functions
 Far virtual tables

OK Cancel Help

[*] More ANSI Violations

Case bypasses initialization of a local variable
 Goto bypasses initialization of a local variable
 Untyped bit field assumed signed int
 'ident' declared as both external and static
 Declare 'ident' prior to use in prototype
 Division by zero
 Initializing 'ident' with 'ident'
 This initialization is only partially bracketed

OK Cancel Help

[*] Optimization Options

Optimization Options
 Register optimization
 Jump optimization OK

Register Variables
 None
 Register keyword Cancel
 Automatic

Optimize For
 Size Help
 Speed

[*] C++ Warnings

Assignment to 'this' is obsolete
 Base initialization without a class name
 Functions containing 'ident' are not expanded inline
 Function 'ident' should have a prototype
 'ident' is both a structure tag and a name
 Temporary used to initialize 'ident'
 Temporary used for parameter 'ident'
 The constant member 'ident' is not initialized
 This style of function definition is now obsolete
 Use of 'overload' is now unnecessary and obsolete
 Obsolete syntax; use '::' instead
 Assigning 'ident' to 'ident'
 'ident' hides virtual function 'ident'
 Non-const function 'ident' called for const object

OK Cancel Help

[*] Source Options

Source Options OK
 Nested comments.

Keywords
 Borland C++ Cancel
 ANSI
 UNIX V
 Kernighan and Ritchie

Identifier Length 32 Help

[*] Frequent Errors

Function should return a value
 Unreachable code
 Code has no effect
 Possible use of 'ident' before definition
 'ident' is assigned a value which is never used
 Parameter 'ident' is never used
 Possibly incorrect assignment

More... OK Cancel Help

[*] Compiler Messages

Errors: Stop After 25
Warnings: Stop After 25
 Display warnings

Portability...
ANSI violations...
C++ warnings...
Frequent errors...

OK Cancel Help

[*] More Frequent Errors

Superfluous & with function
 'ident' declared but never used
 Ambiguous operators need parentheses
 Structure passed by value
 No declaration for function 'ident'
 Call to function with no prototype
 Restarting compile using assembly
 Unknown assembler instruction
 Function definition cannot be a typedef'ed declaration
 Ill formed pragma
 Array variable 'ident' is near

OK Cancel Help

[*] Portability Warnings

Non-portable pointer conversion
 Non-portable pointer comparison
 Constant out of range in comparison
 Constant is long
 Conversion may lose significant digits
 Mixing pointers to signed and unsigned char

OK Cancel Help

[*] Segment Names

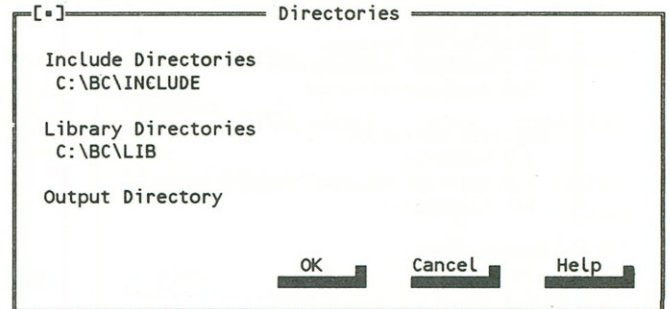
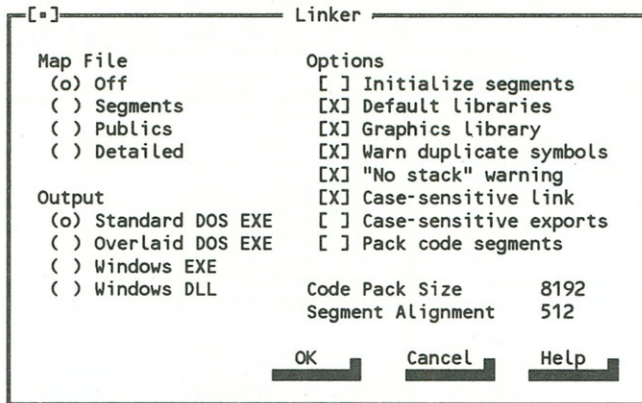
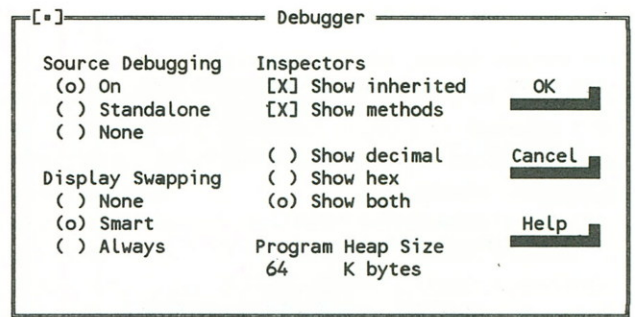
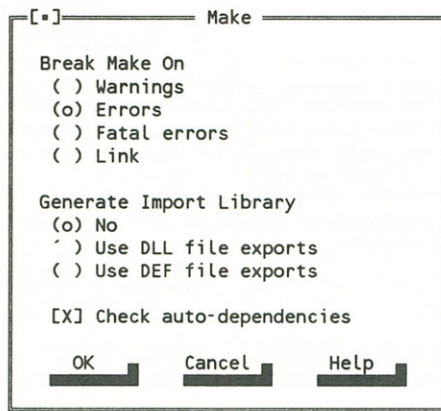
Code Segment *	BSS Segment *
Code Group *	BSS Group *
Code Class *	BSS Class *
Data Segment *	Far Data Segment *
Data Group *	Far Data Group *
Data Class *	Far Data Class *

OK Cancel Help

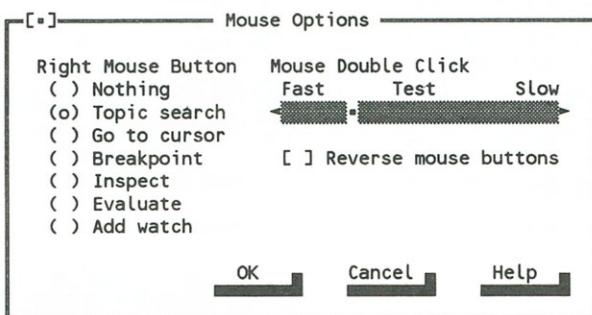
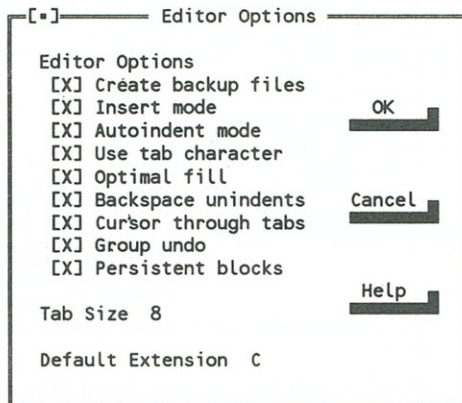
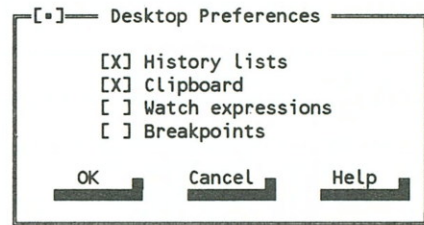
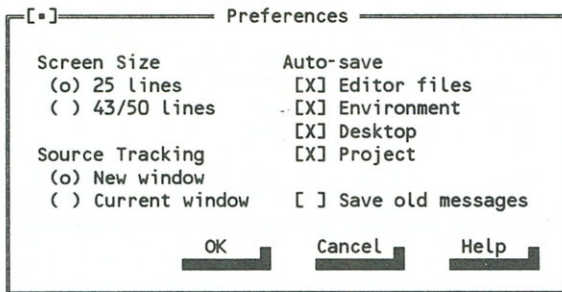
[*] ANSI Violations

Void functions may not return a value
 Both return and return of a value used
 Suspicious pointer conversion
 Undefined structure 'ident'
 Redefinition of 'ident' is not identical
 Hexadecimal value more than three digits

More... OK Cancel Help



A `c:\bc` helyére azt a pathot kell helyettesíteni, ahol a Borland C++ alkönyvtára van.



Most pedig írjunk egy rövidke programot, hogy lássuk, valóban működik-e még a fordítóprogram (minden mesterkedésünk ellenére):

```
suli
#include <studio.h>
void main()
{
    puts(„Na végre!”);
}
```

Ha ezt beírtuk, futtasuk le (nyomjunk **CTRL+F9**-et). A futás eredményét az **ALT+F5** billentyűvel nézhetjük meg.

Baros László
(MosaiC Szoftverstúdió)

Kezdődjön a C-có!

A C programozási nyelvet Dennis Ritchie tervezte több mint húsz évvel ezelőtt, és azóta is a profi programozók által leggyakrabban használt általános célú programnyelv. Köszönheti ezt annak, hogy gépközeli (sok olyan elemet tartalmaz, ami a legtöbb processzornak is csak egy-két utasítás), és ugyanakkor mégis gépfüggetlen (gyakorlatilag minden gépen használhatunk C-fordítót). Ez nagy dolog, mert ha valaki gyors és kicsi programot akar írni, nem kell minden héten egy-egy új mikroprocesszor assembly (gépi kódú) nyelvét megtanulnia, hanem mindegyiken használhatja a C-t. Ez volt az első olyan magasszintű programnyelv, amelyiken operációs rendszert is írtak. (Érdekességként megjegyezzük, hogy az igen elterjedt UNIX operációs rendszer mintegy 10%-a assembly-, a többi C nyelven íródott). A C nyelv logikájában nagyon hasonlít a Pascalra, de annál nagyobb szabadságot biztosít a programozónak. (A Pascal nyelvet a strukturált programozási szemlélet oktatására fejlesztették ki, így tulajdonképpen a Pascalt a C tanulónyelvének tekinthetjük.)

Na de lássuk a medvét!

Változók

A C nyelvben alapvetően kétféle adatábrázolás van: fixpontos és lebegőpontos.

A fixpontos gyakorlatilag egészet jelent, mérete¹ szerint lehet:

char	(1 byte),
int	(2 byte),
long	(4 byte).

A lebegőpontos (azaz tizedes tört) lehet:

float	(4 byte),
double	(8 byte),
long double	(10 byte).

Na igen, a szemfülesek mindjárt észrevették hogy valami hiányzik: a szöveg!

A szöveget a C nyelvben karakterekből álló tömbökben tároljuk. Hogyan? Minden betűnek megvan a maga száma (kódja), és hogy mindenki egyforma kódokat használjon, a betűkre szabványt dolgoztak ki (sajnos többfélét is). A legáltalánosabban az ASCII szabvány terjedt el Magyarországon.

Mint már említettük, minden betűnek megvan a maga kódja, de fordítva ez már nem igaz! Ugyanis

van néhány olyan kód, amihez nem betű tartozik, hanem a nyomtatónak szóló üzeneteket, dudálást, tabulátort... jelentenek, és van egy kód, ami az égadta világon semmit nem jelent, ez a 0 (nulla). Ezt a kódot a C nyelvben arra használjuk, hogy a szöveg végére tesszük (legtöbbször automatikusan kerül oda) és onnan tudjuk, hogy addig tart a szöveg.

Tömbök

Természetesen használhatunk tömböket is. Minden tömb indexelése a nulladik elemtől kezdődik, és n elemszámú tömb esetében n-1 a legnagyobb (legális) tömbindex. Mi az hogy legális? Ez azt jelenti, hogy senki sem tiltja meg, hogy 105. tömbelemre hivatkozzunk, pedig csak 100 elemű a tömb. Ez az egyik leggyakoribb, és legnehezebben kivadászható hiba programozás közben. Ugyanis ilyen esetekben előfordulhat, hogy másik változókat, esetleg a programkódot sikerül átírni, ami általában végzetes.

A tömb egy elemére a tömbváltozó neve, és utána szögletes zárójelek közé tett az index(ek)-el hivatkozhatunk. (pl: **tomb[100]**, **matrix[20][30]**).

Mutatók

A C nyelvben nemcsak egyszerű változókat illetve tömbváltozókat használhatunk (mint a BASIC-ban vagy a Pascal-ban), hanem mutatókat (pointereket) is. A mutató egy adott változó címét jelenti. A mutatónak van típusa is aszerint, hogy milyen típusú változóra mutat. A C megengedi az „akárhova mutató” pointert is, az ilyet **void** típusúnak nevezik.

Egy változó címét a változónév elé tett **&** jellel képezhetjük. pl: ha **c** egy **char** típusú változó, akkor **&c** a **c** változó címét jelenti, azaz a **c** változóra mutató pointer. Ennek ismeretében már értéket is tudunk adni a mutatóinknak (pl: **s=&c;**). Na jó, de hogyan kaphatjuk vissza annak a változónak az értékét, ahova a pointer mutat? Úgy, hogy a mutató elé egy csillagot teszünk (**c=*s**).

Egy tömb neve a nulladik elemére mutató pointer is egyben. Tehát ha tömb egy száz elemű int-ekből álló tömb, akkor a **tomb[0]** ugyanazt jelenti mint a ***tomb**, azaz a tömb nulladik elemét. Itt kezd érdekes lenni, hogy miért is van a mutatóknak típusa! Ugyanis a fordító figyel arra, hogy amennyiben hozzáadunk a mutatóhoz egyet, akkor ne csak egy byte-tal mutasson odébb, hanem annyival, amennyi az adott változó hossza. Így aztán az előző példánál maradva a **tomb[1]**-nek is

¹ A leírtak a Borland cég C-fordítóira (Turbo C, Turbo C++, Borland C++) igazak 100%-ig

van megfelelője: ***(tomb+1)**. Ezért fordul az elő, ha egy void típusú mutatóhoz adunk hozzá valamit, akkor a fordító „ugat”, azaz szöveges üzenettel figyelmeztet.

Változók deklarálása

Mielőtt a változókat használnánk, deklarálni kell őket, azaz meg kell mondani a változó nevét és típusát (tömb esetén a méretét is). A deklarációban rögtön kezdőértéket is adhatunk változóinknak. Ha ezt nem tesszük meg, akkor a változó értéke határozatlan lesz. Ha a kezdőérték megadásából egyértelműen kiderül a tömb mérete, akkor nem kötelező megadni. Mutatót úgy deklarálunk, hogy a változónév elé egy *-ot (csillagot) teszünk.

```
int    x;
char   betu='a';
int    y=20;
long   munka=8, pihenes=8, szorakozas=8;
char   szoveg[100];
int    primek[]={2,3,5,7,11,13,17,19,23};
int    sakktabla[8][8];
char   amoba[3][3]=
{{'x','o','x'},{'o','o','x'},{'x','x','o'}};
int    *pi;
char   *szia=„Ave Ceasar morituri te salutant!”;
int    *pprim=primek;
int    py=&y;
void   *p;
```

Változók érvényességi tartománya

Attól függően, hogy a változókat hol deklaráltuk, vannak belső (internal), és külső (external) változók.

Ha egy változót a függvényeken kívül, azok előtt deklarálunk, akkor azt minden függvény közösen használhatja (külső változó).

Ha a függvényen belül (vagy általában, ha két kapcsos zárójel között) deklaráljuk a változót, akkor annak érvényességi tartománya a nyitó zárójeltől a bezáró kapcsos zárójelig tart, más függvényekben ugyanilyen néven deklarált változókat nem éri el. Ugyanez igaz a függvény paramétereire is.

Ha ugyanolyan néven volt már egy külső változó, vagy a zárójeleken kívül egy belső változó, akkor azt a zárójeleken belül nem tudjuk elérni (mivel ezen a néven a belső változókra hivatkozunk).

Típuskonverzió

Ha egy kifejezésben két különböző típusú változó szerepel, akkor először közös típusúvá kell alakítani őket, még azelőtt, mielőtt a műveletet elvégeznénk. (Ezt a C általában elvégzi helyettünk.) Mindig az *alacsonyabb* típusú változó alakul át *magasabb* típusúvá, és az eredmény is a *magasabb* típusú lesz. Például: egy **int** és egy **long** összeadásakor az eredmény **long** lesz. Ha a konverziót a fordító nem végzi el automatikusan (pl: ha lebegőpontos számot akarunk tömbindexként használni), akkor ezt

nekünk kell elvégezni. Ezt úgy tehetjük meg, hogy a változó elé zárójelbe a kívánt típus nevét írjuk. (pl: ha **x** egy **long**, akkor **(int)x** már úgy viselkedik mint egy **int**.) A mutatókat is hasonlóképpen konvertálhatjuk (pl. ha **p** egy **long**-ra mutató pointer, akkor azt így konvertálhatjuk **int**-re mutató pointerre: **(int *)p**).

Operátorok

Aritmetikai operátorok + (összeadás), - (kivonás), * (szorzás), / (osztás), % (maradék).

A % (modulo) operátor két egész szám osztásakor keletkező maradékot adja, lebegőpontos számokra nincs értelmezve.

Relációs operátorok: < (kisebb), > (nagyobb), >= (nagyobb vagy egyenlő), <= (kisebb vagy egyenlő).

Logikai operátorok: == (egyenlő), != (nem egyenlő), && (és), || (vagy).

Inkrementáló, és dekrementáló operátorok: ++ (inkrementálás), -- (dekrementálás).

A ++ operátor egy változó értékét növeli eggyel, a -- pedig csökkenti eggyel a változó értékét. Ha a változó előtt áll, akkor még az előtt növeli meg a változó értékét, mielőtt az felhasználásra kerül, ha utána, akkor a változó felhasználása után.

Bitenkénti operátorok: & (és), | (megengedő vagy), ^ (kizáró vagy), << (bitléptetés balra), >> (bitléptetés jobbra), ~ (egyes komplement).

Értékadó operátorok: = (legyen egyenlő)

C nyelvben az **x = x op y** kifejezést rövidebben is leírhatjuk **x op= y** alakban, ahol **op** lehet +, -, *, /, %, <<, >>, &, |, ^. Így vannak +=, -=, *=, /=, %=, <<=, >>=, &=, |=, ^= értékadó operátorok.

Kiértékelés sorrendje

Operátor:	Asszociativitás:
() []	balról jobbra
! ~ ++ -	jobbról balra
* / %	balról jobbra
+ -	balról jobbra
<< >>	balról jobbra
< > <= >=	balról jobbra
&	balról jobbra
^	balról jobbra
	balról jobbra
= += -= *= /= %= <<= >>= &= = ^=	balról jobbra

Függvények

A C nyelvben nagyon kevés utasítás van, majd nem mindent függvényekkel intézhetünk el. Hogy a nyelv mégis egységes legyen, vannak standard függvények, amelyek minden gépen illetve fordítóprogramon ugyanúgy működnek. Ilyen például a scanf és a printf függvény.

A függvény tulajdonképpen egy olyan alprogram, aminek változókat adhatunk át, azokkal csinál valamit, majd egy értéket visszaad.

A függvény leírásának vannak bizonyos szabályai:

A függvénynév elé a visszatérő érték típusát írjuk, majd a függvénynév következik, utána zárójelek között a függvényparaméterek felsorolása vesszővel elválasztva (mindegyik elé oda kell írni a típusát is). Majd a nyitó kapcsos zárójel után a függvényben használt változókat deklaráljuk, ezután pedig magát az algoritmust írjuk le. Ha van visszatérő érték, akkor azt a **return** utasítással adhatjuk vissza, majd a függvény végét bezáró kapcsos zárójel jelzi.

pl:

```
float felszin(float a, float b, float c)
{
float f;                /* felszin */
float lap1,lap2;        /* lap1 és lap2
                        deklarációja */
float lap3=b*c;         /* lap3 deklarációja
                        értékadással */

lap1=a*b;
lap2=a*c;
f=2*(lap1+lap2+lap3);

return(f);              /* visszatérés f
                        értékével */
}
```

Ha a függvénynek nincs visszatérő értéke, akkor a függvénynév elé void-ot teszünk, ha paramétert sem vár, akkor a paraméter helyére is void-ot kell tenni. pl:

```
void varj(void )
{
printf(„\nNyomj meg egy billentyűt!”);
getch();
}
```

Ha a függvényt azelőtt szeretnénk használni, mielőtt leírtuk volna, akkor deklarálni kell. A függvény deklarációjában meg kell adni a függvény visszatérő értékének típusát, a függvény nevét, zárójelek között vesszővel elválasztva a paraméterek típusát (a neveit is lehet, de nem kötelező), majd pontosvesszővel le kell zárni. pl:

```
float felszin(float a, float b, float c);
void varj(void );
```

A függvényt a következő formában hívhatjuk meg:

```
x=terfogat(a,b,c);
```

A program futása mindig a main nevű függvényvel kezdődik.

Példaprogram

Már eleget tudunk ahhoz, hogy írjunk egy rövid programot!

A programszövegben a `/*` ez itt egy megjegyzés `*/` jelek közé tett szöveget a fordító megjegyzésnek tekinti (olyan, mintha ott sem volna).

```
#include <stdio.h>      /* A printf és a scanf
                        függvény az stdio.h
                        file-ban van deklarálva,
                        ezért a fordítót értesí-
                        teni kell, hogy ezt is
```

```
#include <conio.h>/    szerkessze bele */
                        * a getch a conio.h file-
                        ban van deklarálva */

float beker(char *txt)
{
float ret;              /* ret deklarálása */
printf(„\n%s ”,txt);   /* txt kiírása */
scanf(„%f”,&ret);     /* ret bekérése */
return(ret);           /* visszatérés ret-tel */
}                       /* beker függvény vé-
                        ge */
                        /* függvénydeklará-
                        ciók */

float felszin(float a, float b, float c);
float terfogat(float a, float b, float c);
void varj(void);
void main(void)        /* főfüggvény, itt indul
                        a program */

{
float a,b,c;           /* változók deklarálása
                        */

float felsz;
printf(„\nTéglatest térfogatát és felszínét szá-
                        moló program”);
varj();                /* várakozás egy bil-
                        lentyű leütéséig */
a=beker(„Írd be a téglatest első élét:”);
                        /* beker függvény hí-
                        vása, a változó beolva-
                        sása */
b=beker(„Második él:”); /* b bekérése */
c=beker(„Harmadik él:”); /* c bekérése */
felsz=felszin(a,b,c);
printf(„\n A téglatest felszíne: %.2f \n térfo-
                        gata %f”, felsz, terfogat(a,b,c));
                        /* eredmény kiírása
                        */

varj();
}                       /* main függvény vé-
                        ge*/

float felszin(float a, float b, float c)
{
float f;                /* felszin */
float lap1,lap2;        /* lap1 és lap2 deklará-
                        ciója */
float lap3=b*c;         /* lap3 deklarációja ér-
                        tékadással */

lap1=a*b;
lap2=a*c;
f=2*(lap1+lap2+lap3);
return(f);              /* visszatérés f értéké-
                        vel */
}                       /* felszín vége */

float terfogat(float a, float b, float c)
{
return(a*b*c);         /* visszatérő érték
                        a*b*c */
}                       /* terfogat vége */

void varj(void )
{
```

```
printf(„\nNyomj meg egy billentyűt!");
getch(); /* várakozás egy billentyű
        lenyomásáig */
} /* varj vége */
```

Programozás közben segítséget az F1 billentyűvel kérhetünk.

Ha a kurzorral ráálunk egy C utasításra, vagy egy standard függvényre és ekkor CTRL+F1 -et nyomunk, akkor az annak az utasításnak vagy a függvénynek a leírását olvashatjuk.

A példaprogramban előforduló függvények működésének itt a leírása:

getch

```
int getch();
```

Vár egy billentyű leütéséig, és annak kódját adja vissza.

printf

```
int printf(char*format, ...);
```

A függvény formázottan írja ki a változók értékét a standard outputra, ami alapértelmezésben a képernyő (konzol).

Visszatérő értéke az output karakterek száma. Ha valami hiba történik (pl: lezárjuk a standard outputot) a visszatérő érték un. EOF lesz (stdio.h-ban van definiálva).

A formátum mondja meg, hogy hány változó értékét, és azokat milyen formátumban írja ki. A formátum után a paraméterlistán vesszővel elválasztva magukat a változókat kell felsorolni. (Ha több, kevesebb vagy más típusú változót írunk be, mint amennyit a formátumban előre jeleztünk, akkor nem jelez hibát, hanem a formátumot tekinti szentírásnak, és az alapján próbálja teljesíteni a rábízott feladatokat, azon az áron is, hogy esetleg semmit, vagy „szemetet” ír ki.)

A formátumban előforduló % (százalék) helyére fogja behelyettesíteni a változók formázott értékét, a százalék után még lehet néhány karakter, ami megmondja, hogy az adott változót milyen formában jelenítse meg:

```
%[flag][width][.prec]type
[flag]
```

A - jel esetében balra igazítja, + jel esetében (alapértelmezés, tehát nem kell kitenni) jobbra igazítja a szöveget.

```
[width]
```

Az itt megadott szélességben fogja a változó értékét kiírni, ha a szélességérték elé egy 0-t teszünk, akkor az üres helyeket 0-val fogja feltölteni.

```
[.prec]
```

Egy tizedespont után írt szám arra ad utasítást, hogy hány tizedes pontossággal jelenítse meg a változó értékét.

type

A változó típusát mondja meg:

d előjeles decimális int
o előjel nélküli oktális int
u előjel nélküli decimális int
x előjel nélküli hexadecimális int
f float
ld előjeles decimális long
lo előjel nélküli oktális long
lu előjel nélküli decimális long
lx előjel nélküli hexadecimális long
lf double
c előjeles char
s szöveggént kiírás a szöveget lezáró 0 karakterig, vagy [.prec] db karakter kiírása.
p pointerként töbrénő kiírás

Van néhány speciális karakter, amit gyakran használunk, ezek ún. láthatatlan karakterek, ezeket nem egy, hanem több karakterrel jelöljük:

```
\n soemelés
\t tabulátor
\0 0 kódú karakter
\\ \karakter
```

scanf függvény leírása.

```
int scanf(char *format, ...);
```

A scanf függvény a standard inputról (billentyűzet) várja a bemenetet az adott formátumnak megfelelően, és a megfelelő változókba írja azokat. Visszatérő értéke a sikeresen beolvasott változók száma.

Formátuma hasonlít a printf formátumához, azzal a különbséggel, hogy maszkok (nem a formátumot definiáló szövegek) nem megengedettek. A paraméterlistáján a formátum után pedig nem a változókat, hanem azok címeit várja, mivel a C nyelv függvényparaméter-átadása olyan, hogy az átadott változó eredeti értékét a függvény nem tudja megváltoztatni. Ezért általánosan művelt gyakorlat a C-ben, hogy a változó címét adjuk át a függvény paraméterlistáján, és így az adott függvény már hozzáfér a változóhoz olyannyira, hogy az eredeti értéket is meg tudja változtatni. Gyakori hiba, hogy nem a változó címét, hanem a változót írják be a scanf paraméterlistájára (A legtöbbször ez is végzetes!).

Helyhiány miatt most nem foglalkozhattunk a különböző vezérlési szerkezetekkel, struktúrákkal, unionokkal, függvénypointerekkel, dinamikus tárolással. Amint arra alkalom lesz, nemcsak ezekkel a témakörökkel ismerkedhetünk meg, hanem szó lesz néhány speciális függvényparaméter átadásról (printf, main...), példaként pedig egy kicsi, de látványos ablakrendszer fog szolgálni.

Baros László
 MosaiC SzoftverStúdió

```

p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO a STEP 2 * 1: p1 = k1 + i
p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1))): PSET (p1, p2), 2
GOSUB 10: PSET (p3, p4), 1: NEXT
l = 1: FOR i = a TO 60 STEP 1: p1 = k1 + i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO 420 STEP 5 * 1: p1 = k1 + i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
l = -1: FOR i = 440 TO 60 STEP 5 * 1: p1 = k1 - i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO a STEP 2 * 1: p1 = k1 - i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
l = 1: FOR i = a TO 60 STEP 1: p1 = k1 - i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO 520 STEP 5 * 1: p1 = k1 - i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS
LOCATE 1, 12: PRINT "Hiperbola inverze"
o1 = 179: o2 = 100: r = 80: z1 = o1: z2 = o2: GOSUB 9: GOSUB 11
k1 = 140: k2 = 100: a = 40: b = 50
l = -1: FOR i = 440 TO 60 STEP 5 * 1: p1 = k1 + i
p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO a STEP 2 * 1: p1 = k1 + i
p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1))): PSET (p1, p2), 2
GOSUB 10: PSET (p3, p4), 1: NEXT
l = 1: FOR i = a TO 60 STEP 1: p1 = k1 + i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO 420 STEP 5 * 1: p1 = k1 + i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
l = -1: FOR i = 440 TO 60 STEP 5 * 1: p1 = k1 - i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO a STEP 2 * 1: p1 = k1 - i: p2 = k2 - SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
l = 1: FOR i = a TO 60 STEP 1: p1 = k1 - i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
PSET (p1, p2), 2: GOSUB 10: PSET (p3, p4), 1: NEXT
FOR i = 60 TO 520 STEP 5 * 1: p1 = k1 - i: p2 = k2 + SQR(b ^ 2 * (ABS((i ^ 2 / a ^ 2) - 1)))
GOSUB 10: PSET (p3, p4), 1: NEXT
GOSUB all: CLS
END
9 LINE (z1 - 3, z2) - (z1 + 3, z2): LINE (z1, z2 - 3) - (z1, z2 + 3): RETURN
10 v1 = p1 - o1: v2 = p2 - o2: v = SQR(v1 ^ 2 + v2 ^ 2)
v3 = v1 / v: v4 = v2 / v
d = r * r / v: p3 = o1 + (d * v3): p4 = o2 + (d * v4): RETURN
11 l = 12: FOR i = 0 TO 360 - 1 STEP 1
x = o1 + (r * COS(u * i)): y = o2 + (r * SIN(i * u))
x1 = o1 + (r * COS((i + 1) * u)): y1 = o2 + (r * SIN((i + 1) * u))
LINE (x, y) - (x1, y1): x = x1: y = y1: NEXT: RETURN
12
v1 = a1 - b1: v2 = a2 - b2: v = SQR(v1 ^ 2 + v2 ^ 2): v3 = v1 / v: v4 = v2 / v
FOR i = 0 TO v STEP 4: PSET (b1 + i * v3, b2 + i * v4), 1: NEXT
RETURN
all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN

```



```

GOSUB all
CLS
SCREEN 1
WINDOW (0, 0)-(639, 479)
COLOR 0, 2
pi = 3.14159: u = pi / 180
o1 = 320: o2 = 170: r = 160: GOSUB 11: o1 = 320: o2 = 120: r = 110: GOSUB 11
LOCATE 19, 25: PRINT "K1": LOCATE 15, 31: PRINT "K2"
o1 = 320: o2 = 10: r = 320
LINE (320, 7)-(320, 13): LOCATE 23, 21: PRINT "0"
FOR i = 0 TO 179 STEP 2: x = o1 + r * COS(u * i): y = o2 + r * SIN(u * i)
PSET (x, y): NEXT
o1 = 320: o2 = 10: r = 320: pi = 3.14159: u = pi / 180
o3 = 320: o4 = 170: r1 = 160
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 1: NEXT
LOCATE 8, 33: PRINT "K2": LOCATE 2, 33: PRINT "K1"
o3 = 320: o4 = 120: r1 = 110
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 1: NEXT
a = p4 - 330: r9 = a / 2: k8 = 320: k9 = 330 + r9: CIRCLE (k8, k9), r9, 1
o3 = 320: o4 = k9: r1 = r9
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 - a: o4 = k9: r1 = r9
CIRCLE (o3, o4), r1, 1
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 + a: o4 = k9: r1 = r9
CIRCLE (o3, o4), r1, 1
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 + 2 * a: o4 = k9: r1 = r9
CIRCLE (o3, o4), r1, 1
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 - 2 * a: o4 = k9: r1 = r9
CIRCLE (o3, o4), r1, 1
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 + 3 * a: o4 = k9: r1 = r9
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 - 3 * a: o4 = k9: r1 = r9
l = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 + 4 * a: o4 = k9: r1 = r9

```

```

1 = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
o3 = 320 - 4 * a: o4 = k9: r1 = r9
1 = 3: FOR i = 0 TO 359 STEP 1: p1 = o3 + r1 * COS(u * i)
p2 = o4 + r1 * SIN(u * i)
GOSUB 10: PSET (p3, p4), 2: NEXT
GOSUB all
END
all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN
9 LINE (z1 - 3, z2)-(z1 + 3, z2): LINE (z1, z2 - 3)-(z1, z2 + 3): RETURN
10 v1 = p1 - o1: v2 = p2 - o2: v = SQR(v1 ^ 2 + v2 ^ 2): v3 = v1 / v
v4 = v2 / v: d = r * r / v: p3 = o1 + (d * v3): p4 = o2 + (d * v4): RETURN
11 l = 12: FOR i = 0 TO 360 - 1 STEP 1: x = o1 + (r * COS(u * i))
y = o2 + (r * SIN(u * i)): x1 = o1 + r * COS(u * (i + 1))
y1 = o2 + r * SIN(u * (i + 1)): LINE (x, y)-(x1, y1), 2: x = x1: y = y1: NEXT
RETURN

```

Axonometrikus kép rajzolása

```

ad:
g = 1.2: o1 = 320: o2 = 250
n = 8: e = 12: pi = 3.14159: u = pi / 180
DIM x(n), y(n), z(n), x1(n), y1(n), x2(n), y2(n), p(2 * n), q(2 * n), al(60), be(60)
ad1:
SCREEN 0
COLOR 15, 1, 8
CLS
LOCATE 2, 2: INPUT "Az ax. kep cime:"; c$
LOCATE 4, 2: INPUT "Elforgatas szoge:"; al
LOCATE 6, 2: INPUT "Dontes szoge:"; be
LOCATE 8, 2: INPUT "Nagyitas-kicsinyites merteke:"; m
CLS
SCREEN 1
WINDOW (0, 0)-(639, 479)
COLOR 0, 2
LINE (5, 25)-(635, 465), 2, B
ax:
q$ = INKEY$
IF q$ = "w" THEN SCREEN 2: GOTO ad1
IF q$ = "q" THEN GOTO lem
RESTORE
FOR i = 1 TO n: READ x(i), y(i), z(i)
x(i) = -x(i): y(i) = -y(i): z(i) = -z(i)
x1(i) = x(i) * COS(al * u) - y(i) * SIN(al * u)
y1(i) = x(i) * SIN(al * u) + y(i) * COS(al * u)
x2(i) = x1(i) * m: p(i) = o1 + g * x2(i)
y2(i) = (y1(i) * SIN(be * u) - z(i) * COS(be * u)) * m
q(i) = o2 + g * y2(i): NEXT

```

Induláskor megadjuk a rajz címét, a kívánt döntési és forgatási szögeket és a rajz méretét. (1-nél nagyobb szám nagyít, ennél kisebb kicsinyít).

Az ábra kirajzolódik. Ha mérete nem megfelelő vagy az elforgatással, döntéssel nem vagyunk megelégedve, a W gombbal előlről kezdhetjük.

A sikeres rajzot a Q megnyomására az adott címmel lemezre vehetjük. (A lemezre vett rajzot az itt közölt RAJZHIVO programmal vihetjük újra a képernyőre.)

A program most egy kockát rajzol. Ha a DATA sorokat kicseréljük és egy másik alakzat csúcsainak három-három koordinátáját adjuk meg, majd megadjuk az éllel összekötendő csúcspárokat (az utolsó DATA sorban), kirajzolódik az új alakzat. A program elején át kell javítani a csúcsok (n) és az él (e) számát az új alakzatnak megfelelően.

Írta: Pethes Endre

```

FOR i = 1 TO e: READ i1, i2: LINE (p(i1), q(i1))-(p(i2), q(i2)), 1: NEXT
FOR k = 1 TO 2000: NEXT
GOTO ax
END
lem:
DEF SEG = &HB800
BSAVE c$, 0, 16536
DATA -60,-60,-60,60,-60,-60,60,60,-60,-60,60,-60
DATA -60,-60,60,60,-60,60,60,60,60,-60,60,60
DATA 1,2,2,3,3,4,4,1,5,6,6,7,7,8,8,5,1,5,2,6,3,7,4,8
bev:
SCREEN 0: CLS
LOCATE 3, 12: PRINT "GRAFIKA-BEHIVO PROGRAM"
LOCATE 6, 6: INPUT "Irja be a lemezen lévő, behivandó kép címét !"; c$
SCREEN 1
COLOR 0, 2
vegr:
DEF SEG = &HB800
BLOAD c$, 0
GOSUB all
SCREEN 2
CLS : SCREEN 0
LOCATE 10, 6: PRINT "Kér másik grafikát (i-n)?"
vala:
v$ = INKEY$
IF v$ = "" THEN GOTO vala
IF v$ = "i" THEN CLS : GOTO bev
END
all: a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN

```

Cikloisok

A program a geometriában és a műszaki gyakorlatban sokszor szereplő gördülő mozgás során keletkező görbéket, a cikloisokat és az evolvenst mutatja be.

```

be:
SCREEN 0
COLOR 0, 14, 13
CLS
LOCATE 2, 12: PRINT "G Ö R D Ü L Ö M O Z G A S"
LOCATE 5, 2: PRINT "A síkbeli alakzat pontjai gördülés közben görbéket irnak le."
LOCATE 6, 2: PRINT "Ezek a görbék a 'RULETTA'-k."
LOCATE 8, 2: PRINT "Ha egyenesen kör gördül, a körhöz rögzített pontok 'KÖZÖNSEGÉS CIKLOIS'-okat"
LOCATE 9, 2: PRINT "irnak le. Ezek csúcsosak,nyújtottak és hurkoltak lehetnek."
LOCATE 11, 2: PRINT "Ha egy kör egy másik körön kívül gördül, a leirt görbék az 'EPICIKLOIS'-ok."
LOCATE 13, 2: PRINT "Ha egy kör egy másik kör belsejében gördül, a görbék a 'HYPOCIKLOIS'-ok."
LOCATE 15, 2: PRINT "Körön gördülő egyenes pontjai 'EVOLVENS'-eket irnak le."
LOCATE 20, 2: PRINT "Melyiket kívánja megnézni ?"
LOCATE 21, 2: PRINT "(Ciklois = 1, epiciklois = 2, hypociklois = 3, evolvens = 4, kilép = 0)"
LOCATE 22, 2: INPUT a
GOTO va

```

```

va:
IF a = 1 THEN GOTO ci
IF a = 2 THEN GOTO epi
IF a = 3 THEN GOTO hy
IF a = 4 THEN GOTO ev
IF a = 0 THEN END

ci:
SCREEN 1
WINDOW (0, 0)-(319, 225)
COLOR 0, 2
CLS
LOCATE 2, 12: PRINT "Kozonseges ciklois"
a = 20: b = 12: c = 40: pi = 3.14159: t = pi / 30: r = a
o1 = 35: o2 = 175
DIM x(120), y(120), x1(120), y1(120)
DIM k(120), l(120), k1(120), l1(120)
DIM m(120), n(120), m1(120), n1(120)
DIM o(120), o1(120)
FOR i = 0 TO 120
x(i) = a * i * t - a * SIN(i * t): y(i) = a - a * COS(i * t)
x1(i) = x(i) + 35: y1(i) = o2 - r + y(i)
NEXT
LINE (o1 - r, o2 - r)-(310, o2 - r), 2
CIRCLE (o1, o2), r, 1
FOR i = 0 TO 119
LINE (o1, o2)-(o1, o2 - r - 9), 2
LINE (x1(i), y1(i))-(x1(i + 1), y1(i + 1)): NEXT
o2 = 125: LINE (o1 - r, o2 - r)-(310, o2 - r), 2
CIRCLE (o1, o2), r, 1
LINE (o1, o2)-(o1, o2 - r - 9), 2
FOR i = 0 TO 120
k(i) = a * i * t - b * SIN(i * t): l(i) = a - b * COS(i * t)
k1(i) = k(i) + o1: l1(i) = o2 - r + l(i): NEXT
FOR i = 0 TO 119
LINE (k1(i), l1(i))-(k1(i + 1), l1(i + 1)): NEXT
o2 = 55: LINE (o1 - r, o2 - r)-(310, o2 - r), 2
CIRCLE (o1, o2), r, 1
LINE (o1, o2)-(o1, o2 - r - 9), 2
FOR i = 0 TO 120
m(i) = a * i * t - c * SIN(i * t): n(i) = a - c * COS(i * t)
m1(i) = m(i) + o1: n1(i) = o2 - r + n(i): NEXT
FOR i = 0 TO 119
LINE (m1(i), n1(i))-(m1(i + 1), n1(i + 1)): NEXT
GOSUB all
LOCATE 25, 2: INPUT "Ujabb valasztas :"; a
GOTO va
END

all:
a$ = INKEY$
IF a$ = "" THEN GOTO all
RETURN
epi:
SCREEN 1
WINDOW (0, 0)-(319, 225)

```



```

COLOR 0, 2
CLS
LOCATE 2, 15: PRINT " EPICIKLOIS"
p1 = 180: p2 = 100: a = 31: b = 31: h = 48: n = 10
c = a + b: d = c / b: pi = 3.14159: t = pi / 60
CIRCLE (p1, p2), a, 2
CIRCLE (p1 + a + b, p2), a, 1
FOR i = 0 TO 120
x(i) = c * COS(i * t) - b * COS(d * i * t): y(i) = c * SIN(i * t) - b * SIN(d * i * t)
x1(i) = p1 + x(i): y1(i) = p2 - y(i)
k(i) = c * COS(i * t) - h * COS(d * i * t): l(i) = c * SIN(i * t) - h * SIN(d * i * t)
k1(i) = p1 + k(i): l1(i) = p2 - l(i)
m(i) = c * COS(i * t) - n * COS(d * i * t): o(i) = c * SIN(i * t) - n * SIN(d * i * t)
m1(i) = p1 + m(i): o1(i) = p2 - o(i)
NEXT
FOR i = 0 TO 119
LINE (x1(i), y1(i))-(x1(i + 1), y1(i + 1))
LINE (k1(i), l1(i))-(k1(i + 1), l1(i + 1))
LINE (m1(i), o1(i))-(m1(i + 1), o1(i + 1))
NEXT
GOSUB all
LOCATE 25, 2: INPUT "Ujabb.valasztas:"; a
GOTO va

hy:
SCREEN 1
WINDOW (0, 0)-(319, 225)
COLOR 0, 2
CLS
LOCATE 2, 15: PRINT "HYPOCIKLOIS"
p1 = 80: p2 = 100: a = 78: b = 26: h = 40: n = 12
pi = 3.14159: t = pi / 60: c = a - b: d = c / b
CIRCLE (p1 + a, p2), a, 2
CIRCLE (p1 + a + c, p2), b, 1
FOR i = 1 TO 120
x(i) = c * COS(i * t) + b * COS(d * i * t)
y(i) = c * SIN(i * t) - b * SIN(d * i * t)
x1(i) = p1 + x(i) + a: y1(i) = p2 - y(i): NEXT
FOR i = 1 TO 119
LINE (x1(i), y1(i))-(x1(i + 1), y1(i + 1))
NEXT
FOR i = 1 TO 120
k(i) = c * COS(i * t) + h * COS(d * i * t)
l(i) = c * SIN(i * t) - h * SIN(d * i * t)
k1(i) = p1 + k(i) + a: l1(i) = p2 - l(i): NEXT
FOR i = 1 TO 119
LINE (k1(i), l1(i))-(k1(i + 1), l1(i + 1))
NEXT
FOR i = 1 TO 120
m(i) = c * COS(i * t) + n * COS(d * i * t)
o(i) = c * SIN(i * t) - n * SIN(d * i * t)
m1(i) = p1 + m(i) + a: o1(i) = p2 - o(i): NEXT
FOR i = 1 TO 119
LINE (m1(i), o1(i))-(m1(i + 1), o1(i + 1)): NEXT i
GOSUB all
LOCATE 25, 2: INPUT "Ujabb.valasztas:"; a

```

```

GOTO va
ev:
SCREEN 1
WINDOW (0, 0)-(319, 225)
COLOR 0, 2
CLS
LOCATE 2, 15: PRINT " EVOLVENS"
p1 = 120: p2 = 80: a = 14: b = 26: h = 32: n = 37
pi = 3.14159: t = pi / 60: c = a - b: d = c / b
DIM x4(170), y4(170), x5(170), y5(170)
DIM k4(170), l4(170), k5(170), l5(170)
DIM m4(170), o4(170), m5(170), o5(170)
DIM y7(170), l7(170), o7(170)
CIRCLE (p1, p2), a, 2
LINE (p1 + a, p2 - 2 * a)-(p1 + a, p2 + 3 * a), 1
FOR i = 0 TO 160
x4(i) = a * COS(i * t) + a * i * t * SIN(i * t)
y4(i) = a * SIN(i * t) - a * i * t * COS(i * t)
y7(i) = y4(i) + p2
x5(i) = p1 + x4(i): y5(i) = p2 - y4(i)
k4(i) = x4(i) - h * COS(i * t)
l4(i) = y4(i) - h * SIN(i * t)
l7(i) = l4(i) + p2
k5(i) = p1 + k4(i): l5(i) = p2 - l4(i)
m4(i) = x4(i) + n * COS(i * t)
o4(i) = y4(i) + n * SIN(i * t)
o7(i) = o4(i) + p2
m5(i) = p1 + m4(i): o5(i) = p2 - o4(i): NEXT
FOR i = 0 TO 150
LINE (x5(i), y5(i))-(x5(i + 1), y5(i + 1))
LINE (k5(i), l5(i))-(k5(i + 1), l5(i + 1))
LINE (m5(i), o5(i))-(m5(i + 1), o5(i + 1))
NEXT: FOR i = 0 TO 60
LINE (x5(i), y7(i))-(x5(i + 1), y7(i + 1))
LINE (k5(i), l7(i))-(k5(i + 1), l7(i + 1))
LINE (m5(i), o7(i))-(m5(i + 1), o7(i + 1))
NEXT
GOSUB all
LOCATE 25, 2: INPUT "Ujabb.valasztas:"; a
GOTO va

```

CSAK LEMEZEN



Betűíró

Egy alapbetűtípust variálhatunk: készíthetünk kicsiny, nagy, széles, keskeny betűket változatos méretben és arányokkal. Egy-sornyi feliratot úgy készítünk, hogy megadjuk X és Y értékét (a betű bal alsó csúcsának koordinátái). Az S a betű színe, Q a szélessége, W a magassága (a legkisebb érték mindkettőnél 1).

A kiírandó szöveget DATA sorban adjuk meg, betűnként vesszővel elválasztva. Szóközt a kötőjellel adunk. Minden sort 0-val fejezünk be.

Tetszés szerinti nagy- és kisbetűket is írhatunk. Az ékezetes betűk a következők:

Kisbetűk: á=aa, é=ee, í=ii, ó=oo, ö=oe, ő=oi, ú=uu, ü=ue, ű=ui.

Nagybetűk: Á=A2, É=E2, Ó=O2, Ö=O3, Ő=O4, Ú=U2, Ű=U3, Ū=U4.

Vessző ; kettőspont | pont .

A közölt mintapéldák sokféle változatot mutatnak be. A szövegek oldalanként printerrel ki-nyomtathatók.

Zenebeíró

Először a tempót kell megadni, majd a darab címét (maximum 8 betű). Ezután a képernyőn egy sor zongorabillentyű látható (első sor), alatta különböző hosszúságú hangjegyek, szünetjelek (második sor).

Kezdetkor az első sor C hangja alatt függőleges vonalka áll, ez a kurzormozgató gombokkal balra és jobbra irányítható. A beírandó hang alá állunk, megnyomjuk a H gombot, ezután a vonalka a második sor alá ugrik. Hasonlóan

mozgatva a kellő hosszúságú hang alá visszük, és megnyomjuk kétszer a H-t.

Ekkor a kért hang megfelelő hosszúságban megszólal. Ez jelzi, hogy a gép a hangot elfogadta. A jel újra az első sor alatt áll és új hangra vár; folytassuk a hangok bevitelét. Ha a V-t megnyomjuk, a beírt dallam lejátszódik. Ekkor a tempót módosíthatjuk, a zenét lemezre vehetjük vagy folytathatjuk. A téves beírás törölhető: a vonal jelet H-val a második sorba mozgatjuk, a T jel alá visszük és újból megnyomjuk a H-t; ekkor a hang törlődik, majd a gép kiírja, melyik hangnál tartunk. Ugyanígy vihetjük be a szüneteket is, ezeket a gép hangonként számolja. (A pont a második sorban rövid szünetet jelent, egymás utáni azonos hangok elválasztására alkalmas.)

A lemezre felvett zenét a 'ZENEHIVO' programmal hívhatjuk be és játszhatjuk le, ennek tehát lemezünkön kell lennie.

Kúpszeletek simulókörrei

A megszerkesztett kúpszeletek megrajzolása (kihúzása) nem könnyű. A munkát megkönnyíti, ha meg tudjuk rajzolni legalább a görbék csúcsában (tengelypontban) a fő görbületi köröket.

A program azt mutatja be, hogyan kell ellipszis, parabola és hiperbola esetében meghatározni ezek középpontját és sugarát.

Ellipszis-szerkesztések

A program bemutatja, hogyan szerkeszthetjük meg az ellipszis

nagy- és kistengelyét, majd ezek ismeretében az úgynevezett két-körös eljárással a görbe további pontjait, ha egy ellipszisnek ismerjük egy konjugált átmérő-párját.

A szerkesztéseket lépésenként mutatjuk be. Bármely gomb megnyomásával lépegethetünk.

Tesoszámok

A matematika többféle érdekes kapcsolatot ismer számok között. Programunk megkeresi azokat a számokat, amelyek saját négyzetükkel olyan különös kapcsolatban állnak, hogy ha a szám számjegyeit és négyzetének számjegyeit összeadjuk, a két összeg egyenlő lesz egymással. Ezeket a számokat neveztük el tesoszámoknak. Papíron, vagy fejben igencsak sokáig kereshetnénk őket, a számítógép viszont pillanatok alatt rájuk talál. A program PC-n, Quick Basicben készült.

Számkiíró program

A program kérésére a számjegyekkel beírt számot a gép betűkkel írja ki a képernyőre.

Dátumköz-számoló

Alkalmazásával megállapíthatjuk, hogy egy adott nap az évnek hányadik napja, illetve két időpont között hány nap telt el. A program a szökőéveket is figyelembe veszi.

A programokat írta:
Pethes Endre

Rajzok

A program C-64-esünk egyik hiányosságát kiküszöbölve, csodálatos grafikákat varázsol a képernyőre. Csak győzzük kivárni...

Írta: Gábryel

```

100 REM ##### <E0
110 REM # C-64 # <E8
120 REM # RAJZOK # <51
130 REM # # <D2
140 REM # # <93
150 REM # KESZITETTE : GABRYEL # <0F
160 REM # # <14
170 REM ##### <2D
180 : <FC
190 GOTO 410 <FF
200 : <0D
210 FA=ABS(X1-X0):FB=ABS(Y1-Y0) <5C
220 FC=SGN(X1-X0):FD=SGN(Y1-Y0) <5A
230 FM=FA:IF FB>FA THEN FM=FB <98
240 FP=FM/2:FQ=FM/2 <9C
250 FOR FI=1 TO FM <41
260 : <91
270 GA=8192 <F1
280 GB=INT(X0/8):GC=INT(Y0/8) <2C
290 GD=Y0 AND 7 <CE
300 GE=GA+GC*320+8*GB+GD <8C
310 GF=7-(X0 AND 7) <7B
320 POKE GE,PEEK(GE) OR (2↑GF) <4F
330 : <85
340 FP=FP+FA <B1
350 IF FP>=FM THEN FP=FP-FM:X0=X0+FC <3C
360 FQ=FQ+FB <B2
370 IF FQ>=FM THEN FQ=FQ-FM:Y0=Y0+FD <A6
380 NEXT <59
390 RETURN <BF
400 : <F9
410 POKE 53265,59:POKE 53272,27:A=1024 <27
420 FOR I=0 TO 999:POKE A+I,9:NEXT <2F
430 FOR I=49152 TO 49179 <88
440 READ A:POKE I,A:S=S+A:NEXT <C6
450 IF S<>4429 THEN PRINT "HIBA" <D7
460 DATA 169,000,133,251,169,032,133 <24
470 DATA 252,169,000,170,160,000,145 <53
480 DATA 251,200,192,000,208,249,230 <F7
490 DATA 252,232,224,064,208,240,096 <E4
500 SYS 49152 <82
510 : <90
520 FOR I=0 TO 90 STEP 5:J=90-I <5C
530 X0=160:Y0=100-J:X1=160+I:Y1=100 <E3
540 GOSUB 200 <6E
550 X0=160+I:Y0=100:X1=160:Y1=100+J <A3
560 GOSUB 200 <D1
570 X0=160:Y0=100+J:X1=160-I:Y1=100 <C7
580 GOSUB 200 <53
590 X0=160-I:Y0=100:X1=160:Y1=100-J <5C
600 GOSUB 200 <E3
610 NEXT <D7
620 : <A7
630 DATA 102,102,108,120,108,103,99, <93
640 DATA 127,49,48,60,48,49,127, <15
650 DATA 62,99,96,62,3,99,62, <A3

```

```

660 DATA 127,99,6,28,51,99,126, <63
670 FOR I=0 TO 31:READ A:POKE 8512+I,A <C4
: NEXT <C4
680 POKE 8200,24:POKE 8201,24 <7D
690 POKE 8202,24:POKE 8203,24 <78
700 POKE 198,0:WAIT 198,1:PRINT "[SH/C <0F
LR]" <0F
710 POKE 53272,21:POKE 53265,27 <1A
720 : <7E
730 : <BE
740 : <FF
750 FOR I=49152 TO 49179 <53
760 READ A:POKE I,A:S=S+A:NEXT <22
770 IF S<>4429 THEN PRINT "HIBA" <10
780 DATA 169,000,133,251,169,032,133 <C0
790 DATA 252,169,000,170,160,000,145 <1F
800 DATA 251,200,192,000,208,249,230 <A4
810 DATA 252,232,224,064,208,240,096 <51
820 SYS 49152:POKE 53280,..:POKE 53281, <18
9 <18
830 FOR I=0 TO 999:POKE 1024+I,9:NEXT <C4
840 POKE 53265,57:POKE 53272,27 <DA
850 C=COS(π/3):S=SIN(π/3) <5E
860 C1=COS(π/36):S1=SIN(π/36):SF=.95 <D6
870 X=95:Y=0:CX=140:CY=96:SC=1.16 <70
880 : <59
890 FOR J=1 TO 40 <D6
900 FOR I=0 TO 6 <01
910 SX=X*SC+CX:SY=CY+Y <2C
920 IF I=0 THEN X0=SX:Y0=SY <F9
930 X1=SX:Y1=SY:GOSUB 350 <4D
940 XN=X*C-Y*S:Y=X*S+Y*C:X=XN <4F
950 X1=SX:Y1=SY:NEXT <2B
960 XN=SF*(X*C1-Y*S1):Y=SF*(X*S1+Y*C1) <4C
:X=XN <4C
970 NEXT <5E
980 GOTO 1200 <C1
990 : <70
1000 FA=ABS(X1-X0):FB=ABS(Y1-Y0) <67
1010 FC=SGN(X1-X0):FD=SGN(Y1-Y0) <32
1020 FM=FA:IF FB>FA THEN FM=FB <8F
1030 FP=FM/2:FQ=FM/2 <0B
1040 FOR FI=1 TO FM <FD
1050 : <B2
1060 GA=8192 <69
1070 GB=INT(X0/8):GC=INT(Y0/8) <0F
1080 GD=Y0 AND 7 <53
1090 GE=GA+GC*320+8*GB+GD <13
1100 GF=7-(X0 AND 7) <B5
1110 POKE GE,PEEK(GE) OR (2↑GF) <91
1120 : <C6
1130 FP=FP+FA <CE
1140 IF FP>=FM THEN FP=FP-FM:X0=X0+FC <07
1150 FQ=FQ+FB <82
1160 IF FQ>=FM THEN FQ=FQ-FM:Y0=Y0+FD <01
1170 NEXT <8B
1180 RETURN <5A
1190 : <1C
1200 : <0B
1210 DATA 102,102,108,120,108,103,99, <1F
1220 DATA 127,49,48,60,48,49,127, <EC
1230 DATA 62,99,96,62,3,99,62, <4D
1240 DATA 127,99,6,28,51,99,126, <CB
1250 FOR I=0 TO 31:READ A:POKE 8512+I,A <E0
: NEXT <E0
1260 POKE 8200,24:POKE 8201,24 <58
1270 POKE 8202,24:POKE 8203,24 <B6
1280 POKE 198,0:WAIT 198,1 <E2
1290 POKE 53272,21:POKE 53265,27 <9B
1300 END <80

```

FLOPPY-POSTA A programok lemezen is megrendelhetők!

Rajzoló

A joystickkal és egy megjelenő nyíl segítségével rajzolhatunk a grafikus képernyőre. A rajzolás sebességét a 65-ös sorban változtathatjuk meg, a Shift+Ctrl pedig képernyőt töröl.

Írta: **Badenszky Csaba**

```
0 REM *** RAJZOLO *** <D0
5 PRINT CHR$(147) <D1
10 FOR Z=32768 TO 33200 <B8
20 READ A$ <D8
30 S$=LEFT$(A$,1):GOSUB 80:A=S*16 <FF
40 S$=RIGHT$(A$,1):GOSUB 80:A=A+S <E6
50 POKE Z,A:E=E+A:PRINT "[HOME]"432-( <B8
Z-32768);"[LEFT]"
60 NEXT :IF E<>55560 THEN PRINT "DATA <05
ERROR!!":END
65 S=20:REM SEBESSEG <C5
66 POKE 33108,S <64
70 SYS 33192:REM START <83
80 IF ASC(S$)>64 THEN S=ASC(S$)-55:RE <08
TURN
90 S=VAL(S$) <04
100 RETURN <98
101 DATA A0,00,A2,20,84,FA,86,FB,98,91 <19
,FA,C8,D0,FB,E6,FB
102 DATA CA,D0,F6,60,A9,00,85,FA,85,FB <96
,AS,FD,48,4A,4A
103 DATA F0,11,AA,18,E6,FB,A9,40,65,FA <D0
,85,FA,90,02,E6,FB
104 DATA CA,D0,F0,68,29,07,F0,09,18,65 <5C
,FA,85,FA,90,02,E6
105 DATA FB,A5,FC,48,29,F8,18,65,FA,85 <43
,FA,90,02,E6,FB,A9
106 DATA 80,85,02,68,29,07,F0,06,AA,46 <BB
,02,CA,D0,FB,A5,FE
107 DATA F0,02,E6,FB,18,A5,FB,69,20,85 <84
,FB,A0,00,B1,FA,05
108 DATA 02,91,FA,60,F0,E0,E0,90,08,04 <67
,02,A2,01,8E,15,D0
109 DATA 8E,27,D0,CA,8E,20,D0,8A,A2,A0 <14
,A0,80,8E,00,D0,8C
110 DATA 01,D0,A0,3B,A2,1B,8C,11,D0,8E <D7
,18,D0,A2,40,9D,3F
111 DATA 03,CA,D0,FA,A8,B9,74,80,9D,40 <B3
,03,E8,E8,E8,C8,C0
112 DATA 07,D0,F2,A9,0D,8D,F8,07,A2,40 <99
,CA,D0,FD,20,4E,81
113 DATA AD,00,DC,29,10,D0,1C,AD,10,D0 <4F
,85,FE,38,AD,00,D0
114 DATA E9,18,85,FC,B0,02,C6,FE,38,AD <08
,01,D0,E9,32,85,FD
115 DATA 20,14,80,A5,CB,C9,33,D0,CF,AD <0E
,8E,02,C9,01,D0,C8
116 DATA 20,00,80,4C,B8,80,AD,01,D0,C9 <31
,32,F0,03,CE,01,D0
117 DATA 60,AD,01,D0,C9,F9,F0,03,EE,01 <1A
,D0,60,AD,00,D0,C9
118 DATA 18,D0,07,A9,01,2C,10,D0,F0,0D <C1
,CE,00,D0,AD,00,D0
119 DATA C9,FF,D0,03,CE,10,D0,60,AD,00 <C2
,D0,C9,57,D0,07,A9
120 DATA 01,2C,10,D0,D0,0B,EE,00,D0,AD <39
,00,D0,D0,03,EE,10
```

```
121 DATA D0,60,4C,90,81,4C,96,81,4C,9C <73
,81,4C,A2,81,CE,C0
122 DATA 02,D0,30,A9,10,8D,C0,02,AD,00 <CF
,DC,A8,29,05,F0,E2
123 DATA 98,29,09,F0,E0,98,29,06,F0,DE <3F
,98,29,0A,F0,DC,98
124 DATA 29,01,F0,10,98,29,02,F0,0E,98 <3E
,29,04,F0,0C,98,29
125 DATA 08,F0,0A,60,4C,F6,80,4C,01,81 <3E
,4C,0C,81,4C,28,81
126 DATA 20,0C,81,4C,F6,80,20,28,81,4C <66
,F6,80,20,0C,81,4C
127 DATA 01,81,20,28,81,4C,01,81,20,44 <29
,E5,20,00,80,4C,7B,80
```

Sűrítő

A program kiveszi a Basic forrásprogramokból a felesleges szóközoeket, valamint a megjegyzéseket. A csak REM-et tartalmazó sorokat sorszám + kettőspont formában meghagyja, hiszen máshol hivatkozhatunk ezekre.

Írta: **Badenszky Csaba**

```
10 INPUT "FORRAS FILE";F$ <B8
20 INPUT "CEL FILE";C$ <A7
30 OPEN 1,8,0,F$ <17
40 OPEN 2,8,1,"0:"+C$ <A8
50 GET#1,A$,A$,A$,A$ <F4
60 PRINT#2,CHR$(1);CHR$(8); <F4
70 I=0:T=4:C=2049:S=1:DIM S$(80) <81
80 GET#1,L$,H$ <61
90 IF L$="" THEN L$=CHR$(0) <06
100 IF H$="" THEN H$=CHR$(0) <D2
110 S$(2)=L$:S$(3)=H$ <C0
200 GET#1,A$:IF A$="" THEN 500 <14
210 A=ASC(A$) <FD
220 IF A=34 THEN GOSUB 300 <CC
230 IF A=32 AND I=0 THEN 200 <F4
240 IF A=143 AND I=0 THEN 320 <DF
250 S$(T)=A$:T=T+1 <62
260 GOTO 200 <C4
300 IF I THEN I=0:RETURN <44
310 I=1:RETURN <30
320 IF T=4 THEN S$(T)=CHR$(58):T=5:GOT <59
O 340
330 T=T-1 <88
340 GET#1,A$:IF NOT A$="" THEN 340 <30
350 GOTO 500 <B5
500 PRINT S;TAB(5)ASC(S$(2))+256*ASC(S <94
$(3))
510 S=S+1 <2B
520 S$(T)=CHR$(0) <23
530 C=C+T+1 <55
540 CH=INT(C/256) <6B
550 CL=C-CH*256 <49
560 S$(0)=CHR$(CL) <16
570 S$(1)=CHR$(CH) <9A
580 FOR Z=0 TO T <C8
590 PRINT#2,S$(Z); <DC
600 NEXT <A7
610 GET#1,A1$,A2$ <F0
620 IF A1$="" AND A2$="" THEN 700 <E3
630 I=0:T=4:GOTO 80 <A4
700 A$=CHR$(0) <13
710 PRINT#2,A$;A$ <C3
720 CLOSE 2:CLOSE 1 <02
730 PRINT "[DOWN][RIGHT]KESZ" <6E
```

Music voices

A program 8 különböző hangeffektust mutat be és egy beépített zenét is tartalmaz (DIXIE). Érdeemes meghallgatni!

Írta: **Olasz Andre**

```

100 REM ***** <7B
110 REM * MUSIC VOICES C-64* <DB
120 REM * , * <9C
130 REM * KESZITETTE: BRUCE MAESTRO * <14
140 REM ***** <46
150 : <BA
160 PRINT CHR$(147):POKE 53280,0:POKE <DB
53281,0:POKE 646,5
170 PRINT "[UP][CTRL/9][14SPC]MUSIC VO <E0
ICES[13SPC][CTRL/0]"
180 PRINT "[CTRL/4][CTRL/9][19SPC]BY[1 <81
8SPC][CTRL/0]"
190 PRINT "[CTRL/2][CTRL/9][14SPC]BRUC <CA
E MAESTRO[12SPC][CTRL/0]"
200 PRINT TAB(5)"[DOWN]HELP[DOWN][4LEF <C5
T][4C=/Y]"
210 PRINT "[CTRL/3][DOWN][RIGHT]1 .... <CC
GONG
220 PRINT "[DOWN][RIGHT]2 ..... BOMBA <F2
230 PRINT "[DOWN][RIGHT]3 ..... LOVES <C4
240 PRINT "[DOWN][RIGHT]4 ..... SOUND <AB
250 PRINT "[DOWN][RIGHT]5 ..... AUTO I <9B
NDITO
260 PRINT "[DOWN][RIGHT]6 ..... BEEPIT <03
ETT ZENE
270 PRINT "[DOWN][RIGHT]7 ..... FELSZA <EA
LLAS
280 PRINT "[DOWN][RIGHT]8 ..... LESZAL <0D
LAS
290 PRINT "[DOWN][RIGHT]9 ..... SZIREN <55
A
300 PRINT TAB(20)"[18UP][DOWN][RIGHT]E <11
..... END
310 GET A$:E=0 <95
320 IF A$="1" THEN 430 <FF
330 IF A$="2" THEN 520 <AC
340 IF A$="3" THEN 670 <04
350 IF A$="4" THEN 780 <7B
360 IF A$="5" THEN 930 <8A
370 IF A$="6" THEN RUN 1040 <57
380 IF A$="7" THEN 1710 <E3
390 IF A$="8" THEN 1780 <A8
400 IF A$="9" THEN 1850 <8A
410 IF A$="E" THEN POKE 54273,0:END <AD
420 GOTO 310 <BC
430 FOR I=1 TO 5 <62
440 S=54272 <37
450 FOR L=0 TO 24:POKE S+L,0:NEXT <BB
460 POKE S+1,130:POKE S+5,9 <7F
470 POKE S+15,30:POKE S+24,15 <90
480 POKE S+4,21 <67
490 FOR T=1 TO 1000:NEXT <C0
500 POKE S+4,20 <E4
510 GOTO 310 <11
520 REM => BOMBA <= <31
530 : <12
540 S=54272 <EE
550 POKE S+24,7 <DF
560 POKE S+5,12*16+0 <B5
570 POKE S+6,15*16+10 <47
580 FOR X=255 TO 100 STEP -1 <4D
590 POKE S+1,X <BE
600 POKE S+4,17 <B1

```

```

610 FOR D=1 TO 10:NEXT D <CE
620 NEXT X <E4
630 POKE S+24,15 <9C
640 POKE S+1,15 <31
650 POKE S+4,128 <D0
660 GOTO 310 <0B
670 REM => LOVES <= <DB
680 : <2C
690 : <6D
700 S=54272 <37
710 POKE S+24,15 <70
720 POKE S+5,9 <D3
730 POKE S+1,20 <F3
740 POKE S+4,128 <A6
750 FOR D=1 TO 10:NEXT D <85
760 POKE S+4,129:FOR I=0 TO 50:NEXT :P <A3
OKE S+1,0
770 GOTO 310 <02
780 REM => SOUND <= <FC
790 : <23
800 : <53
810 S=54272 <0E
820 POKE S+24,15 <49
830 POKE S+6,240 <98
840 POKE S+4,17 <E1
850 FOR A=1 TO 10 <DF
860 FOR X=1 TO 255 STEP 25 <9C
870 POKE S+1,X <20
880 NEXT X <B5
890 NEXT A <4A
900 POKE S+4,32 <A0
910 GOTO 310 <2A
920 REM => AUTO INDITO <= <7A
930 : <CB
940 S=54272 <87
950 POKE S+24,15 <C0
960 POKE S+6,240:POKE S+4,5 <7C
970 POKE S+4,129 <77
980 FOR I=20 TO 25:POKE S+1,I:NEXT :FO <10
R I=0 TO 50:NEXT :E=E+1
990 FOR I=25 TO 20 STEP -1:POKE S+1,I: <42
NEXT
1000 IF E=10 THEN 1020 <BB
1010 GOTO 980 <C0
1020 POKE S+1,0 <EA
1030 GOTO 310 <5A
1040 REM => BEEPITETT ZENE <= <D2
1050 : <B2
1060 O1=54272:O2=54279:O3=54286 <02
1070 H1=O1+1:H2=O2+1:H3=O3+1 <27
1080 V1=O1+4:V2=O2+4:V3=O3+4 <F5
1090 POKE 54296,15 <8C
1100 POKE V1+1,9:POKE V2+2,0 <94
1110 POKE V2+1,36:POKE V2+2,36 <BE
1120 POKE V3+1,18:POKE V3+2,170 <76
1130 T=TI <1F
1140 POKE V1,16:POKE V2,32:POKE V3,16 <AA
1150 READ S:IF S=0 GOTO 1230 <3D
1160 READ X1,Y1,X2,Y2,X3,Y3 <34
1170 IF X1 THEN POKE H1,X1:POKE O1,Y1:P <CD
OKE V1,17
1180 IF X2 THEN POKE H2,X2:POKE O2,Y2:P <76
OKE V2,33
1190 IF X3 THEN POKE H3,X3:POKE O3,Y3:P <59
OKE V3,17
1200 T=T+S <62
1210 IF T>TI GOTO 1210 <B9
1220 GOTO 1140 <9D
1230 FOR J=01 TO 54296:POKE J,0:NEXT J <FF
1240 DATA 010,051,097,000,000,000,000 <1B
1250 DATA 010,043,052,000,000,000,000 <F3
1260 DATA 020,034,075,021,154,008,147 <81
1270 DATA 020,034,075,025,177,000,000 <E8
1280 DATA 010,034,075,017,037,006,108 <DF
1290 DATA 010,038,126,000,000,000,000 <7A
1300 DATA 010,043,052,025,177,000,000 <DE
1310 DATA 010,045,198,000,000,000,000 <32
1320 DATA 020,051,097,021,154,008,147 <28
1330 DATA 020,051,097,025,177,000,000 <42
1340 DATA 020,051,097,017,037,009,159 <E9
1350 DATA 020,043,052,025,177,010,205 <29
1360 DATA 020,057,172,022,227,011,114 <7B

```

```

1370 DATA 020,057,172,034,075,000,000 <85
1380 DATA 020,057,172,028,214,008,147 <A0
1390 DATA 010,000,000,034,075,000,000 <EA
1400 DATA 010,051,097,000,000,000,000 <81
1410 DATA 020,057,172,022,227,011,114 <29
1420 DATA 010,000,000,034,075,000,000 <3A
1430 DATA 010,051,097,000,000,000,000 <1F
1440 DATA 010,057,172,028,214,010,205 <CA
1450 DATA 010,064,188,000,000,000,000 <E8
1460 DATA 010,068,149,034,075,009,159 <D7
1470 DATA 010,076,252,000,000,000,000 <FD
1480 DATA 020,086,105,021,154,008,147 <4F
1490 DATA 020,000,000,025,177,000,000 <DE
1500 DATA 020,000,000,017,037,006,108 <E8
1510 DATA 010,068,149,025,177,000,000 <7D
1520 DATA 010,051,097,000,000,000,000 <89
1530 DATA 020,068,149,021,154,008,147 <2B
1540 DATA 020,000,000,025,177,000,000 <6C
1550 DATA 020,000,000,017,037,008,023 <18
1560 DATA 010,051,097,025,177,007,053 <2B
1570 DATA 010,043,052,000,000,000,000 <2F
1580 DATA 020,051,097,022,227,006,108 <C2
1590 DATA 020,000,000,025,177,000,000 <78
1600 DATA 020,000,000,019,063,009,159 <F2
1610 DATA 010,038,126,025,177,000,000 <6B
1620 DATA 010,043,100,000,000,000,000 <83
1630 DATA 020,034,075,021,154,008,147 <B8
1640 DATA 020,000,000,025,177,006,108 <09
1650 DATA 020,000,000,021,154,004,073 <D9
1660 DATA 020,000,000,000,000,000,000 <07
1670 DATA 000 <34
1680 GOTO 310 <D4
1690 REM => FELSZLLAS <= <D4
1700 : <54
1710 S=54272 <54
1720 POKE S+24,15 <CB
1730 POKE S+6,240:POKE S+4,5 <AD
1740 POKE S+4,129 <1A
1750 FOR I=20 TO 60:POKE S+1,I:FOR K=0 <1A
    TO 99:NEXT :NEXT :POKE S+1,0 <02
1760 GOTO 310 <AF
1770 : <4D
1780 REM => LESZLLAS <= <1B
1790 S=54272 <A1
1800 POKE S+24,15 <C8
1810 POKE S+6,240:POKE S+4,5 <32
1820 POKE S+4,129 <DE
1830 FOR I=60 TO 20 STEP -1:POKE S+1,I: <CE
    FOR K=0 TO 99:NEXT :NEXT :POKE S+1 <50
    ,0 <E1
1840 GOTO 310 <5E
1850 REM => SZIRENA <= <53
1860 : <36
1870 : <C7
1880 S=54272 <A9
1890 POKE S+24,15 <36
1900 POKE S+6,240 <90
1910 POKE S+4,17 <07
1920 FOR A=1 TO 10 <25
1930 FOR X=10 TO 200 STEP 5 <E3
1940 POKE S+1,X <07
1950 NEXT X <0F
1960 NEXT A <63
1970 POKE S+4,32 <E8
1980 GOTO 310 <A9

```

FLOPPY-POSTA
A programok
lemezen is
megrendelhetők!

Color

A pársoros rutin „kidomborítja” a képernyő ötödik sorát egy raszter rutin segítségével, majd egy sor inverz szöveget visz rá. A végső hatás egy elvileg megoldhatatlan forma: a szöveg színe rasztersoranként változik. A program 40-es sorában az első átszínezett képernyősor számát módosíthatjuk, a 45-ösben pedig a sorok számát. POKE 32826,32 kiadása után a csíkok a keretre is kinyúlnak.

Írta: **Badenszky Csaba**

```

5 FOR Z=32768 TO 32875 <73
10 READ A$ <16
15 A=ASC(LEFT$(A$,1))-48:IF A>9 THEN <70
    A=A-7 <70
20 B=ASC(RIGHT$(A$,1))-48:IF B>9 THEN <D4
    B=B-7 <D4
25 A=A*16+B:S=S+A <54
30 POKE Z,A:NEXT <6E
35 IF S<>12015 THEN PRINT "[SH/CLR]DA <D1
    TA ERROR":LIST 100- <D1
40 K=4:REM KEZDOSOR (0-24) <01
45 S=1:REM SOROK SZ.(1-25) <7D
50 POKE 32780,49+K*8 <9C
55 POKE 32846,S <0B
60 SYS 32768:PRINT "[SH/CLR][CTRL/1]" <B6
65 PRINT "[3DOWN][CTRL/9][14SPC]COMMO <26
    DORE 64[14SPC][CTRL/2]" <26
100 DATA 78,A9,81,8D,1A,D0,A9,1B <5D
101 DATA 8D,11,D0,A9,51,8D,12,D0 <56
102 DATA A0,20,A2,80,8C,14,03,8E <46
103 DATA 15,03,A9,01,85,FB,58,60 <C7
104 DATA AD,19,D0,8D,19,D0,30,07 <2A
105 DATA AD,0D,DC,58,4C,31,EA,A2 <44
106 DATA 0D,CA,D0,FD,A2,07,8D,5C <3D
107 DATA 80,8D,21,D0,8D,21,D0,BC <D0
108 DATA 64,80,88,D0,FD,CA,10,EE <17
109 DATA C6,FB,EA,D0,E7,A9,01,85 <AB
110 DATA FB,A9,00,8D,20,D0,8D,21 <34
111 DATA D0,4C,81,EA,01,03,0E,0E <43
112 DATA 04,04,06,06,06,09,09,08 <78
113 DATA 09,08,08,01 <30

```

Csillagászati jelek

A rutin néhány csillagászati jelet definiál, ezek a Shift- és Commodore-gombokkal érhetők el a programban leírt módon.

Írta: **Várady Tamás**

Sprite Basic

A program 12 új Basic-utasítást szolgáltat, ezekkel egyszerűbbé válik a sprite-ok kezelése.

Írta: **Olasz Endre**

```

100 REM CS. JELEK (VARADY TAMAS) <4C
105 PRINT CHR$(142);CHR$(8);"[SH/CLR][ <5B
DOWN]"
110 FOR I=49152 TO 49195:READ X:POKE I <DB
,X:S=S+X:NEXT :IF S=7219 THEN SYS
49152:GOTO 170
120 PRINT "[DOWN] ADATHIBA ! (130-150) <B4
":END
130 DATA 169,0,133,251,133,253,169,208 <6B
,133,252,169,48,133,254,120,169,51
,133,1
140 DATA 160,0,177,251,145,253,200,208 <B9
,249,230,252,230,254,165,252,201,2
24,208
150 DATA 239,169,55,133,1,88,96 <26
160 : <7B
170 D=0:FOR A=51456 TO 51534:READ X:PO <BE
KE A,X:D=D+X:ZX=ZX+1:NEXT
190 SYS 51456 <F0
200 DATA 169,11,141,24,3,169,201,141,2 <73
5,3,96,72,138,72,152,72,165,203,20
1,63,240
210 DATA 3,76,188,254,32,160,229,169,2 <E4
9,141,24,208,169,14,141,134,2,32,6
8,229
220 DATA 169,0,133,207,76,64,201,141,3 <13
2,32,32,32,32,9,142,46,89,68,65,69
,82
230 DATA 146,141,234,160,16,185,47,201 <93
,32,210,255,136,208,247,76,188,254
235 GOSUB 9000 <C5
239 POKE 53272,29:POKE 53280,0:POKE 53 <E7
281,0
240 POKE 53280,0:POKE 53281,0:PRINT "[ <74
SH/CLR][C=/8]";
250 PRINT "[HOME] *** A CSILLAGASZATI <11
JELOLESEK ***"
260 PRINT " ***[4SPC]RENDELKEZSRE ALLN <50
AK[4SPC]***"
270 PRINT "[2DOWN] COMMODORE/F = [C=/F <77
] (FOK)"
280 PRINT "[DOWN] COMMODORE/P = [C=/P] <8F
(PERC)"
290 PRINT "[DOWN] COMMODORE/M = [C=/M] <07
(MASODPERC)"
300 PRINT "[DOWN] SHIFT/H[5SPC]= [SH/H <BF
] (ORA)"
310 PRINT "[DOWN] SHIFT/M[5SPC]= [SH/M <A1
] (PERC)"
320 PRINT "[DOWN] SHIFT/S[5SPC]= [SH/S <74
] (MASODPERC)"
330 END <3A
9000 READ A:FOR I=1 TO 8:READ A(I) <E0
9010 POKE 12287+(A*8)+I,A(I):NEXT <FE
9020 READ A$:IF A$="+" THEN 9000 <36
9030 RETURN <16
10000 DATA 123,48,72,72,48,0,0,0,0,"+ <9B
10010 DATA 103,108,108,108,0,0,0,0,"+ <43
10020 REMDATA 103,0,108,108,108,0, <19
0,0,0,"+"
10030 DATA 111,96,96,96,0,0,0,0,"+ <21
10040 DATA 83,224,128,224,32,224,0,0,0, <77
+"
10050 DATA 77,252,148,148,214,0,0,0,0,"+ <F2
"
10060 DATA 72,192,128,224,176,144,144,0, <4E
0,"*"

```

```

100 REM ***** <7B
105 REM * C-64* <A3
110 REM * SPRITE BASIC * <DA
115 REM * ===== * <D2
120 REM * 1991 * <91
125 REM * , * <DD
130 REM * KÉSZÍTETTE: BRUCE MAESTRO * <14
135 REM ***** <C7
140 PRINT CHR$(147) <07
145 : <3B
150 ::FOR I=49152 TO 49552 <44
155 ::PRINT "[HOME]";I <BB
160 :::::READ A:POKE I,A:S=S+A <2A
165 ::NEXT <EF
170 ::IF S<>49881 THEN PRINT "HIBA" <E7
175 DATA 169,011,141,008,003,169,192 <0A
180 DATA 141,009,003,096,032,115,000 <9B
185 DATA 201,033,240,006,032,121,000 <9D
190 DATA 076,231,167,032,115,000,201 <AE
195 DATA 084,208,003,032,133,192,201 <E0
200 DATA 066,208,003,032,152,192,201 <D0
205 DATA 088,208,003,032,171,192,201 <FB
210 DATA 089,208,003,032,190,192,201 <19
215 DATA 078,208,003,032,209,192,201 <27
220 DATA 065,208,003,032,239,192,234 <99
225 DATA 234,234,234,234,201,069,208 <98
230 DATA 003,032,002,193,201,082,208 <4A
235 DATA 003,032,032,193,201,071,208 <8A
240 DATA 003,032,050,193,201,080,208 <5A
245 DATA 003,032,080,193,201,085,208 <BD
250 DATA 003,032,098,193,201,086,208 <31
255 DATA 003,032,117,193,076,008,175 <3D
260 DATA 255,255,032,115,000,032,253 <C0
265 DATA 174,032,158,183,180,142,007 <48
270 DATA 032,115,000,032,253,174,032 <0D
275 DATA 158,183,180,000,142,248,007 <6C
280 DATA 076,174,167,255,255,032,115 <60
285 DATA 000,032,253,174,032,158,183 <71
290 DATA 180,000,142,021,208,076,174 <DA
295 DATA 167,255,255,032,115,000,032 <9F
300 DATA 253,174,032,158,183,180,000 <E1
305 DATA 142,000,208,076,174,167,255 <74
310 DATA 255,032,115,000,032,253,174 <05
315 DATA 032,158,183,180,000,142,001 <B4
320 DATA 208,076,174,167,255,255,032 <56
325 DATA 115,000,032,253,174,032,158 <E9
330 DATA 183,180,000,142,023,208,032 <0D
335 DATA 253,174,032,158,183,180,000 <6D
340 DATA 142,029,208,076,174,167,255 <39
345 DATA 255,032,115,000,032,253,174 <82
350 DATA 032,158,183,180,000,142,249 <95
355 DATA 007,076,174,167,255,255,032 <74
360 DATA 115,000,032,253,174,032,158 <17
365 DATA 183,180,000,142,002,208,032 <4E
370 DATA 253,174,032,158,183,180,000 <AC
375 DATA 142,003,208,076,174,167,255 <9D
380 DATA 255,032,115,032,253,174,032 <1C
385 DATA 158,183,180,000,142,250,007 <42
390 DATA 076,174,167,255,255,032,115 <79
395 DATA 000,032,253,174,032,158,183 <4A
400 DATA 180,000,142,004,208,032,253 <B6
405 DATA 174,032,158,183,180,000,142 <4D

```

```

410 DATA 005,208,076,174,167,255,255 <26
415 DATA 032,115,000,032,253,174,032 <F4
420 DATA 158,183,180,000,142,027,208 <A8
425 DATA 076,174,167,255,032,115,000 <2D
430 DATA 032,253,174,032,158,183,180 <C6
435 DATA 000,142,251,007,076,174,167 <42
440 DATA 255,255,032,115,000,032,253 <F5
445 DATA 174,032,158,183,180,000,142 <4A
450 DATA 006,208,032,253,174,032,158 <79
455 DATA 183,180,000,142,007,208,076 <10
460 DATA 174,167 <76
465 SYS 49152 <12
470 PRINT CHR$(147):POKE 53280,0:POKE <75
      53281,0:POKE 53269,0:POKE 646,5
475 PRINT TAB(6)"****[2SPC]SPRITE BASI <52
      C V1.0[2SPC]****"
480 PRINT TAB(18)"BY[DOWN][7LEFT]BRUCE <76
      MAESTRO"
485 PRINT "[CTRL/4][RIGHT][DOWN]UTMUTA <83
      TAST [I/N][CTRL/6]
490 GET A$:IF A$="N" THEN END <44
495 IF A$="I" THEN GOTO 505 <9E
500 GOTO 490 <D6
505 PRINT CHR$(147) <FE
510 PRINT "UTMUTATO:[DOWN][9LEFT][9C=/ <60
      Y]"
515 PRINT TAB(10)"UJ UTASITASOK,PARANC <44
      SOK:"
520 PRINT TAB(10)"[24C=/Y]" <FC
525 PRINT "!T,X ..... 0.SPRITE MUTAT <E9
      0
530 PRINT "!A,X ..... 1.SPRITE MUTAT <9D
      0
535 PRINT "!R,X ..... 2.SPRITE MUTAT <2E
      0
540 PRINT "!U,X ..... 3.SPRITE MUTAT <23
      0
545 PRINT "!B,X ..... SPRITE-OK BEKA <B3
      PCSOLOASA
550 PRINT "!X,X ..... 0.SPRITE X KOR <89
      DINATAJA
555 PRINT "!Y,Y ..... 0.SPRITE Y KOR <C4
      DINATAJA
560 PRINT "!E,X,Y ..... 1.SPRITE X/Y K <5B
      ORDINATJA
565 PRINT "!G,X,Y ..... 2.SPRITE X/Y K <4F
      ORDINATJA
570 PRINT "!V,X,Y ..... 3.SPRITE X/Y K <F8
      ORDINATJA
575 PRINT "!N,X,Y ..... A SPRITE-OK X/ <24
      Y IRANYU NA- GYITASA
580 PRINT "!P,X ..... SPRITE-OK ELSO <E7
      BSEGI SZINTJE
585 PRINT "[CTRL/2][3DOWN]" <19
590 PRINT TAB(32)"<SPACE>":FOR I=0 TO <80
      444:NEXT
595 PRINT TAB(32)"[UP][7SPC][UP]":FOR <4A
      I=0 TO 444:NEXT :GET A$:IF A$<>" "
      THEN GOTO 590
600 POKE 646,5:GOTO 470 <A6
605 :::::END ::::: <DB

```

FLOPPY-POSTA
A programok
lemezen is
megrendelhetők!

Betöltő

A következő rutin Basic-programok elé használható töltőként. Mivel a Basic-munkaterületet 1 kilobájttal feljebb tolja, a tényleges program tulajdonképpen a töltő fölött fut. Ha bármikor vissza szeretnénk jutni a töltőhöz, elég egy POKE 44,8 parancs kiadása.

Írta: **Olasz Endre**

```

100 REM ***** <7B
110 REM * BETELOTO C-64* <CF
120 REM * * <9C
130 REM * KESZITETTE: BRUCE MAESTRO * <14
140 REM ***** <46
150 : <BA
160 PRINT CHR$(147):POKE 53280,0:POKE <DB
      53281,0:POKE 646,5
170 INPUT "KEREM A FILE NEVET";A$ <1A
180 PRINT "[CTRL/1][SH/CLR]" <BE
190 PRINT "[2DOWN]LOAD"CHR$(34);A$;CHR <07
      $(34)",8"
200 PRINT TAB(15)"[CTRL/2][5DOWN]SEARC <4B
      HING...[CTRL/1]"
210 POKE 198,10:PRINT "[2UP]POKE646,5" <40
220 DATA 19,13,13,13,13,13,82,85,78,13 <6D
230 FOR I=1 TO 10:READ X:POKE 630+I,X: <4F
      NEXT
240 POKE 44,24:POKE 24*256,0:NEW <B6

```

Invertáló

A rutinnal pillanatok alatt invertálhatjuk a teljes képernyőt. Indítása: SYS 49152. Jól használható például szöveghkiemelésre, villogtatásra is.

Írta: **Olasz Endre**

```

100 REM ***** <7B
110 REM * INVERTALO C-64* <49
120 REM * INIC:SYS49152 * <95
122 REM * WRITTEN BY BRUCE MAESTRO * <E1
123 REM ***** <86
130 : <39
140 FOR I=49152 TO 49205 <4E
150 ::READ A:POKE I,A:S=S+A <BA
160 NEXT <2B
170 IF S<>8880 THEN PRINT "HIBA" <B4
180 DATA 162,255,189,255,003,073,128 <61
190 DATA 157,255,003,202,208,245,162 <81
200 DATA 255,189,255,004,073,128,157 <2A
210 DATA 255,004,202,208,245,162,255 <5A
220 DATA 189,255,005,073,128,157,255 <97
230 DATA 005,202,208,245,162,232,189 <7F
240 DATA 255,006,073,128,157,255,006 <C2
250 DATA 202,208,245,096,255 <3D

```


Kurzor help

A program futtatása után egyetlen utasítással pozicionálhatjuk, illetve ki- vagy bekapcsolhatjuk (\$CC memóriacím) kurzorunkat.

Írta: **Olasz Endre**

```

100 REM ***** <7B
110 REM * * CURSOR HELP * C-64* <1C
120 REM * ***** <DC
130 REM * INIC:SYS 49152 * <55
140 REM * AKT :CX,CURS.X KORDINATAJA* <CE
150 REM * CY,CURS.Y KORDINATAJA* <EA
160 REM * CK,CURS. KAPCSOLOJA * <88
170 REM * , , * <5F
180 REM * KESZITETTE: BRUCE MAESTRO * <68
190 REM ***** <02
200 : <0D
210 FOR I=49152 TO 49255 <54
220 :READ A:POKE I,A:S=S+A <96
230 NEXT <3F
240 IF S<>11293 THEN PRINT "HIBA" <2D
250 DATA 169,011,141,008,003,169,192 <87
260 DATA 141,009,003,096,032,115,000 <36
270 DATA 201,067,240,006,032,121,000 <C3
280 DATA 076,231,167,032,115,000,201 <18
290 DATA 088,208,003,032,051,192,201 <DA
300 DATA 089,208,003,032,068,192,201 <ED
310 DATA 075,208,003,032,085,192,076 <DC
320 DATA 008,175,032,115,000,032,253 <8E
330 DATA 174,032,158,183,180,000,142 <93
340 DATA 211,000,076,174,167,032,115 <7A
350 DATA 000,032,253,174,032,158,183 <8F
360 DATA 180,000,142,214,000,076,174 <B1
370 DATA 167,032,115,000,032,253,174 <7F
380 DATA 032,158,183,180,000,142,204 <05
390 DATA 000,076,174,167,255,255 <52

```

Monitor

Az alábbi program kódolást tesz lehetővé mnemonikok segítségével. Programunk kezdőcímét a 100-as sorban kell megadnunk X változóban, ez eredetileg 49152. Indítás után – elég szokatlan módon – azonnal írhatjuk az utasításokat a következő módszerrel. Először a mnemonikot kell beírni az ezt követő címzés megjelölésével, majd külön utasításként magát a címet. Például:

```

LDA#
FO
STA$$
DO20
RTS

```

Amint a mellékelt példa is mutatja, az abszolút

címzésre két \$ jel utal, indirekt vagy indexelt címzésekre pedig nincs lehetőség.

A monitor – kalandjátékokhoz hasonló – sorbeviteli rutinnal és néhány új funkcióval rendelkezik.

< billentyű: kilépés a Basicbe;
? : újraindítja a monitort;
<-: törli a bevitt sort.

Írta: **Horváth Csaba**

```

100 :::::POKE 53280,0:POKE 53281,0:P <52
PRINT "[SH/CLR][CTRL/2]":CLR :X=491 <52
52
101 PRINT "- MONITOR V1.2-" <9F
102 PRINT "- IRTA: HORVATH CSABA (MR. <69
CSABA)[6SPC]"
103 PRINT "[DOWN][RIGHT]CIM: ";X:PRINT <4E
"[UP][RIGHT]#"
104 PRINT " MNE:":GOSUB 105:GOTO 120 <A3
105 PRINT "[DOWN][RIGHT]>*" :A$="" :GET <56
A$
106 GET A$:IF A$="" THEN 106 <54
107 IF A$=CHR$(13) THEN RETURN <97
108 IF A$="←" THEN B$="" :PRINT "[UP][R <96
IGHT]";B$;"*[27SPC]":GOTO 106
109 IF A$="$" OR A$="#" THEN GOTO 113 <CA
110 IF A$<CHR$(48) OR A$>CHR$(90) THEN <2A
GOTO 106
111 IF A$="?" THEN RUN <DF
112 IF A$="<" THEN PRINT "[DOWN][RIGHT <B1
]VEGE.":SYS 42115
113 B$=B$+A$:PRINT "[UP][RIGHT]";B$;" <A3
*":GOTO 106
120 IF LEN(B$)>5 THEN B$="" :GOTO 104 <33
121 FOR A=1 TO 54:READ C$,KOD <45
122 IF B$=C$ THEN POKE X,KOD:X=X+1:GOT <3B
O 139
123 NEXT :PRINT "???" :RESTORE :B$="" :G <D5
OTO 104
125 DATA "ASL",10,"CLC",24,"DEX",202," <C3
INX",232,"LSR",74,"RTS",96
126 DATA "SEC",56,"TAX",170,"TXA",138 <D5
127 DATA "ADC#",105,"ADC$",101,"AND#", <CE
41,"AND$",37
128 DATA "BCC$",144,"BCS$",176,"BEQ$", <05
240
129 DATA "BMI$",48,"BNE$",208,"BPL$",1 <A9
6
130 DATA "DEC$",198,"EOR#",73,"EOR$",6 <2E
9
131 DATA "INC$",230,"JMP$",76,"JSR$",3 <9A
2
132 DATA "LDA#",169,"LDA$",165,"LDX#", <9C
162,"LDX$",166
133 DATA "ORA#",9,"ORA$",5,"SBC#",233, <31
"SBC$",229
134 DATA "STA$",133,"STX$",134,"LDA$$" <B1
,173,"STX$$",142,"STA$$",141,"JMP$
$",76
135 DATA "DEC$$",206,"SEI",120,"BCC$$" <50
,144,"BCS$$",176,"JSR$$",32,"BMI$$
",48
136 DATA "AND$$",45,"BMI$$",48,"NOP",2 <C9
34,"CPY#",192,"CPY$$",204,"LSR$$",
78
137 DATA "BEQ$$",240,"BVC$$",80,"PLA", <BD
104
139 IF B$="PLA" OR B$="NOP" THEN B$="" <C7
:RESTORE :GOTO 103

```

```

140 IF B$="ASL" OR B$="CLC" OR B$="DEX" <6C
  " OR B$="INX" OR B$="LSR" THEN B$=
  "":RESTORE :GOTO 103
141 IF B$="RTS" OR B$="SEC" OR B$="TAX" <82
  " OR B$="TXA" OR B$="SEI" THEN B$=
  "":RESTORE :GOTO 103
142 PRINT " ERTEK,CIM:" :B$="" :GOSUB 10 <E5
  5
143 IF LEN(B$)>4 THEN B$="" :GOTO 142 <3E
144 IF LEN(B$)=2 THEN :: <8F
145 POKE 63,160:POKE 64,0:POKE 65,16:P <0C
  OKE 66,15
146 IF LEN(B$)=4 THEN GOTO 155 <5A
150 FOR I=0 TO 255:READ H$ <9E
151 IF B$=H$ THEN GOTO 154 <59
152 NEXT :PRINT "???" :B$="" :POKE 63,16 <23
  0:POKE 64,0:POKE 65,16:POKE 66,15:
  GOTO 142
154 POKE X,I:B$="" :X=X+1:RESTORE :GOTO <B6
  103
155 D$=RIGHT$(B$,2):FOR I=0 TO 255:REA <8E
  D H$:IF D$=H$ THEN GOTO 157
156 NEXT :PRINT "???" :B$="" :POKE 63,16 <22
  0:POKE 64,0:POKE 65,16:POKE 66,15:
  GOTO 142
157 POKE X,I:X=X+1:POKE 63,160:POKE 64 <96
  ,0:POKE 65,16:POKE 66,15:FOR I=0 T
  O 255:READ H$
158 D$=LEFT$(B$,2):IF D$=H$ THEN RESTO <41
  RE :B$="" :D$="" :POKE X,I:X=X+1:RES
  TORE :GOTO 103
159 NEXT :PRINT "???" :POKE 63,160:POKE <AB
  64,0:POKE 65,16:POKE 66,15:X=X-1:
  GOTO 142
160 DATA "00","01","02","03","04","05" <1A
  ,"06","07","08","09","0A","0B","0C
  ,"0D"
161 DATA "0E","0F" <A6
162 DATA "10","11","12","13","14","15" <E5
  ,"16","17","18","19","1A","1B","1C
  ,"1D"
163 DATA "1E","1F" <B8
164 DATA "20","21","22","23","24","25" <F0
  ,"26","27","28","29","2A","2B","2C
  ,"2D"
165 DATA "2E","2F" <4B
166 DATA "30","31","32","33","34","35" <F3
  ,"36","37","38","39","3A","3B","3C
  ,"3D"
167 DATA "3E","3F" <5D
168 DATA "40","41","42","43","44","45" <E8
  ,"46","47","48","49","4A","4B","4C
  ,"4D"
169 DATA "4E","4F" <6F
170 DATA "50","51","52","53","54","55" <9B
  ,"56","57","58","59","5A","5B","5C
  ,"5D"
171 DATA "5E","5F" <BE
172 DATA "60","61","62","63","64","65" <D0
  ,"66","67","68","69","6A","6B","6C
  ,"6D"
173 DATA "6E","6F" <31
174 DATA "70","71","72","73","74","75" <E5
  ,"76","77","78","79","7A","7B","7C
  ,"7D"
175 DATA "7E","7F" <23
176 DATA "80","81","82","83","84","85" <FA
  ,"86","87","88","89","8A","8B","8C
  ,"8D"
177 DATA "8E","8F" <15
178 DATA "90","91","92","93","94","95" <CF
  ,"96","97","98","99","9A","9B","9C
  ,"9D"
179 DATA "9E","9F" <07
180 DATA "A0","A1","A2","A3","A4","A5" <52
  ,"A6","A7","A8","A9","AA","AB","AC
  ,"AD"
181 DATA "AE","AF" <D0
182 DATA "B0","B1","B2","B3","B4","B5" <07
  ,"B6","B7","B8","B9","BA","BB","BC
  ,"BD"
183 DATA "BE","BF" <23
184 DATA "C0","C1","C2","C3","C4","C5" <BB
  ,"C6","C7","C8","C9","CA","CB","CC
  ,"CD"
185 DATA "CE","CF" <35
186 DATA "D0","D1","D2","D3","D4","D5" <AC
  ,"D6","D7","D8","D9","DA","DB","DC
  ,"DD"
187 DATA "DE","DF" <07
188 DATA "E0","E1","E2","E3","E4","E5" <01
  ,"E6","E7","E8","E9","EA","EB","EC
  ,"ED"
189 DATA "EE","EF" <19
190 DATA "F0","F1","F2","F3","F4","F5" <14
  ,"F6","F7","F8","F9","FA","FB","FC
  ,"FD"
191 DATA "FE","FF" <29

```

Örökélet

A következő programmal lövöldözős játékokat tehetünk örökéletessé. Beállíthatjuk, hogy a sprite–sprite vagy a sprite–háttér ütközések közül melyiket tiltsa le. Az előbbi esetet választva, az ellenség lövedékei egyszerűen áthaladnak rajtunk, így sérthetetlenül válunk.

Írta: **Lantos Zoltán**

```

1 REM OROKELET KESZITO - LANTOS ZOLT <85
  AN ANONYMOUS SOFTWARE - 1991
2 : <8A
10 DATA 32,10,192,32,10,192,76,48,192 <12
  ,234,32,18,192,32,33,192,96,234,16
  2,1,32
20 DATA 198,255,32,207,255,72,32,204, <75
  255,104,96,234,72,162,2,32,201,255
  ,104,32
30 DATA 210,255,32,204,255,96,234,32, <A8
  18,192,166,144,224,64,240,19,201,1
  73,240
40 DATA 20,201,174,240,25,201,172,240 <9C
  ,30,32,33,192,76,48,192,234,32,33,
  192,96
50 DATA 234,162,169,142,142,192,76,10 <F7
  5,192,234,162,162,142,142,192,76,1
  05,192
60 DATA 234,162,160,142,142,192,234,1 <7D
  41,255,2,32,18,192,141,254,2,166,1
  44,224
70 DATA 64,240,40,201,30,208,68,32,18 <F8
  ,192,141,253,2,166,144,224,64,240,
  88,201
80 DATA 208,208,65,234,169,169,32,33, <71
  192,169,0,32,33,192,169,234,32,33,
  192,76
90 DATA 48,192,234,173,255,2,32,33,19 <A2
  2,173,254,2,76,76,192,234,173,255,
  2,32,33

```

```

110 DATA 192,173,254,2,32,33,192,173,2 <DA
    53,2,76,76,192,234,173,255,2,32,33
    ,192
120 DATA 173,254,2,76,69,192,234,173,2 <40
    55,2,32,33,192,173,254,2,32,33,192
    ,173
130 DATA 253,2,76,69,192,234,201,208,2 <9F
    08,201,173,142,192,32,33,192,169,0
    ,32,33
140 DATA 192,169,234,76,76,192 <BC
145 : <3B
146 FOR X=49152 TO 49395:READ T:POKE X <19
    ,T:Q=Q+T:NEXT
147 IF Q<>32415 THEN PRINT "HIBA A DAT <90
    ASORBAN !":END
148 : <FB
150 PRINT "OROKLET KESZITO" <DB
160 INPUT "KEREM A JATEK NEVET";A$ <67
170 INPUT "MI LEGYEN AZ UJ FILE NEVE"; <FB
    B$:K$=B$
180 PRINT "[DOWN]1-SPRITE-SPRITE UTKOZ <5E
    ES KI"
190 PRINT "2-SPRITE-HATTER UTKOZES KI" <AE
220 GET X$ <EF
221 IF X$="1" THEN POKE 49273,30 <9B
222 IF X$="2" THEN POKE 49273,31 <F4
223 IF X$<>"1" AND X$<>"2" GOTO 220 <7B
224 : <8F
240 OPEN 1,8,0,A$+",P,R":OPEN 2,8,2,"@ <C4
    :"+B$+",P,W":SYS 49152:CLOSE 2:CLO
    SE 1
250 : <D0

```

Return után pedig nulla. Így a dBase READ-jére sincs szükség, még ha több változót viszünk be, akkor sem.

A mellékelt Demo jól szemlélteti a program használatát.

Ha saját programunkba szeretnénk beültetni, nem kell feltétlenül DATA sorokba rendezni. Vegyük fel egyszerűen DYINPEXE néven a programmal egy lemezre, majd programunk első sorába írjuk be a következőket:

```
10 IN=49152: IF PEEK(IN)<>76 OR PEEK 0
(IN+1)<>37 THEN LOAD „DYINPEXE” ,8,1
```

Ilyen formában csak akkor töltődik be a file, ha szükséges, azaz ha megsérült a rutin vagy most futtattuk le először a programot.

Írta: Zahorán György

Dyinp

A következő programot begépelve, egy látványos input rutint kapunk, amit akár programjainkban is felhasználhatunk. Akik dolgoztak IBM-en és ismerik a dBase-t, azoknak biztosan ismerős lesz az ablakban való adatbevitel.

A rutin két funkciót foglal magában. A kurzorpozicionáló hívása: SYS 49155, X, Y (0-39,0-24). Az input rutin szintaxisa: SYS 49152, mezőszélesség, X, Y, [+][%][#]A[\$][%].

Az egyes funkciók jelentése:

+ - nem lehet negatív;

% - csak egész lehet (nem a típus, hanem a bevitt érték);

- a bevitt érték nem lehet nulla.

Adatbevitel során az egyes funkcióbillentyűk jelentése:

F1/F2, F3/F4, F5/F6 - keret-, háttér-, írásszín növelése és csökkentése

F8 - hardcopy, majd 11 soremelés.

Az input rutin használatához fontos, hogy programunk elsőnek felvett változója egész típusú legyen. Ez lesz az output paraméter. Adatbevitel során ugyanis nemcsak a Return billentyűvel léphetünk ki, hanem kurzor fel/le gombokkal is. Ha a fel billentyűvel léptünk ki, akkor ennek az output paraméternek az értéke 1 lesz, kurzor le esetében 2,

```

10 FOR X=49152 TO 49984 <3C
20 READ A:POKE X,A:E=E+A <74
30 NEXT X <FC
40 IF E<>110990 THEN PRINT "HIBA":END <F7
50 DATA 76,37,192,32,13,192,166 <42
60 DATA 176,164,177,76,12,229,32 <30
70 DATA 155,183,224,25,176,9,134 <D3
80 DATA 176,32,155,183,224,40,144 <BF
90 DATA 5,169,14,108,0,3,134 <74
100 DATA 177,96,165,122,133,170,165 <D4
110 DATA 123,133,171,165,170,133,122 <11
120 DATA 165,171,133,123,32,155,183 <DA
130 DATA 134,179,32,13,192,32,47 <9D
140 DATA 193,32,6,192,160,0,132 <2D
150 DATA 199,177,166,41,127,145,166 <32
160 DATA 200,196,179,208,245,164,179 <7C
170 DATA 169,29,32,210,255,136,208 <08
180 DATA 248,132,181,132,182,132,180 <D4
190 DATA 32,97,194,160,4,32,115 <72
200 DATA 0,136,240,31,32,121,0 <24
210 DATA 201,91,176,4,201,65,176 <6B
220 DATA 20,201,170,208,2,133,181 <9F
230 DATA 201,37,208,2,133,182,201 <EE
240 DATA 35,208,223,133,180,240,219 <B6
250 DATA 32,139,176,133,168,132,169 <BD
260 DATA 165,2,208,80,36,13,48 <5C
270 DATA 93,162,0,165,14,16,2 <07
280 DATA 133,182,189,0,2,201,32 <8B
290 DATA 240,13,201,58,176,29,201 <0D
300 DATA 48,144,101,232,228,179,208 <63
310 DATA 236,169,0,133,34,169,2 <26
320 DATA 133,35,138,32,176,183,165 <C5
330 DATA 97,36,14,16,4,201,144 <95
340 DATA 176,84,201,0,208,4,197 <AD
350 DATA 180,208,76,36,14,16,21 <CA
360 DATA 32,170,177,132,158,160,0 <FF
370 DATA 145,168,200,165,158,145,168 <FA
380 DATA 165,2,160,3,145,45,96 <13
390 DATA 166,168,164,169,32,212,187 <44
400 DATA 76,231,192,165,179,32,125 <15
410 DATA 180,160,0,145,71,200,138 <0E
420 DATA 145,71,200,165,99,145,71 <B5
430 DATA 164,179,136,185,0,2,145 <72
440 DATA 98,136,16,248,48,208,201 <C8
450 DATA 45,208,9,164,181,133,181 <CB
460 DATA 240,145,76,45,192,201,46 <8A
470 DATA 208,249,164,182,133,182,76 <A0

```

```

480 DATA 31,193,169,0,133,2,133 <44
490 DATA 178,32,6,192,32,36,234 <34
500 DATA 165,210,133,167,152,24,101 <77
510 DATA 209,144,2,230,167,133,166 <57
520 DATA 165,244,133,165,152,24,101 <C0
530 DATA 243,144,2,230,165,133,164 <6C
540 DATA 32,131,194,32,6,192,165 <A1
550 DATA 198,133,204,141,146,2,240 <44
560 DATA 247,120,165,207,240,12,165 <16
570 DATA 206,174,135,2,160,0,132 <4C
580 DATA 207,32,19,234,32,180,229 <32
590 DATA 201,140,208,6,32,148,194 <71
600 DATA 76,93,193,201,13,240,118 <BC
610 DATA 162,1,201,145,240,55,232 <22
620 DATA 201,17,240,50,201,133,144 <58
630 DATA 49,201,141,176,45,201,136 <46
640 DATA 240,99,72,160,1,41,4 <3F
650 DATA 208,2,160,255,132,158,104 <54
660 DATA 41,3,201,3,240,14,170 <9D
670 DATA 202,189,32,208,24,101,158 <31
680 DATA 157,32,208,76,93,193,32 <F3
690 DATA 252,194,76,93,193,134,2 <24
700 DATA 96,201,20,240,57,201,157 <26
710 DATA 240,53,201,29,240,31,201 <BB
720 DATA 148,240,83,174,24,208,224 <39
730 DATA 21,240,8,201,219,176,31 <21
740 DATA 201,193,176,8,201,96,176 <9B
750 DATA 23,201,32,144,19,201,34 <8F
760 DATA 240,15,32,210,255,230,178 <9A
770 DATA 166,178,228,179,208,4,96 <58
780 DATA 32,210,255,76,93,193,164 <56
790 DATA 178,240,249,198,178,201,157 <CB
800 DATA 240,240,177,166,170,177,164 <44
810 DATA 136,145,164,138,145,166,200 <A5
820 DATA 200,240,6,196,179,144,237 <72
830 DATA 240,235,32,131,194,169,157 <AC
840 DATA 208,212,164,179,198,179,136 <7C
850 DATA 192,255,240,40,196,178,144 <9B
860 DATA 36,177,166,201,160,240,241 <88
870 DATA 196,179,240,26,177,166,170 <56
880 DATA 177,164,200,145,164,138,145 <88
890 DATA 166,136,136,192,255,240,4 <3F
900 DATA 196,178,176,235,200,169,160 <24
910 DATA 145,166,230,179,76,93,193 <AE
920 DATA 164,179,136,177,166,174,24 <A0
930 DATA 208,224,21,240,8,201,65 <9E
940 DATA 144,4,9,128,208,6,201 <6F
950 DATA 32,176,2,9,64,153,0 <74
960 DATA 2,192,0,208,225,96,160 <2B
970 DATA 1,132,199,136,177,166,9 <02
980 DATA 128,145,166,200,196,179,208 <73
990 DATA 245,96,169,4,133,186,169 <3C
1000 DATA 126,133,184,169,0,160,4 <BE
1010 DATA 133,113,132,114,133,183,133 <49
1020 DATA 185,32,192,255,166,184,32 <FF
1030 DATA 201,255,169,14,32,210,255 <C4
1040 DATA 162,25,169,13,32,210,255 <F8
1050 DATA 160,0,177,113,133,158,41 <A6
1060 DATA 63,6,158,36,158,16,2 <F7
1070 DATA 9,128,112,2,9,64,32 <36
1080 DATA 210,255,200,192,40,208,230 <03
1090 DATA 152,24,101,113,133,113,144 <79
1100 DATA 2,230,114,202,208,210,169 <E4
1110 DATA 15,32,210,255,162,11,169 <00
1120 DATA 13,32,210,255,202,208,248 <FB
1130 DATA 32,204,255,169,126,76,195 <69
1140 DATA 255,169,0,141,13,195,141 <A1
1150 DATA 19,195,169,216,141,14,195 <35
1160 DATA 141,20,195,173,0,16,24 <CB
1170 DATA 101,158,141,0,16,238,13 <FB
1180 DATA 195,238,19,195,173,13,195 <99
1190 DATA 208,8,238,14,195,238,20 <85
1200 DATA 195,208,228,201,232,208,224 <8D
1210 DATA 173,14,195,201,219,208,217 <E3
1220 DATA 173,134,2,24,101,158,141 <17
1230 DATA 134,2,141,135,2,96,0 <2F
100 REM ---- DYINPDEMO 1992.04.18. --- <44
- <34
110 : <B7
120 OU%=0:REM AZ OUTPUT PARAMETER <EE
130 REM A NEVE KOZOMBOS, CSAK AZ A <83
140 REM FONTOS, HOGY EGESZ TIPUSU <79
150 REM VALTOZO LEGYEN AZ ELSO. <FD
160 : <7B
170 IN=12*4096:REM AZ INPUT RUTIN CIME <02
180 PO=IN+3:REM POZICIONALO RUTIN CIME <43
190 : <BD
200 A$="[10SPC]":A=0:B%=0:C=0:D=0 <32
210 PRINT "[SH/CLR]":SYS PO,10,0:REM A <7D
10. SORBA
220 PRINT "STRING BEVETELE: ":PRINT <EF
230 PRINT "POZITIV SZAM[3SPC]":PRINT <55
240 PRINT "NEM NULLA SZAM: ":PRINT <AB
250 PRINT ">0 ES EGESZ[4SPC]":PRINT <75
260 : <91
270 REM STRING <BE
280 SYS PO,10,17:PRINT A$ <38
290 SYS IN,10,10,17,A$ <55
300 IF OU%=1 GOTO 490 <F7
310 : <04
320 REM POZITIV <4F
330 SYS PO,12,17:PRINT "[8SPC]" <38
340 SYS PO,12,16:PRINT A <5D
350 SYS IN,8,12,17,+#A <DE
360 IF OU%=1 GOTO 280 <55
370 : <88
380 REM NEM NULLA <5D
390 SYS PO,14,17:PRINT "[8SPC]" <06
400 SYS PO,14,16:PRINT B% <73
410 SYS IN,5,14,17#B% <35
420 IF OU%=1 GOTO 330 <14
430 : <3C
440 REM >0 ES EGESZ <FE
450 SYS PO,16,17:PRINT "[8SPC]" <DD
460 SYS PO,16,16:PRINT D <EE
470 SYS IN,7,16,17,+#D <CF
480 IF OU%=1 GOTO 390 <62
490 END <55

```

Villogás

A pársoros rutin a keret színét villogtatja, miközben a teljes képernyő felerősödik, majd elhalványul.

Írta: **Fogarasi András**

```

10 FOR T=49152 TO 49269:READ A:POKE T <3F
,A:S=S+A:NEXT <3F
30 SYS 49152:END <F2
40 DATA 120,169,21,141,20,3,169,192,1 <53
41,21,3,160,0,140,84,192,140 <3C
50 DATA 86,192,88,96,172,84,192,200,1 <44
40,84,192,192,3,208,49,160,0 <44
60 DATA 140,84,192,172,86,192,185,87, <3F
192,162,0,157,0,216,157,0,217 <3F
70 DATA 157,0,218,157,0,219,141,32,20 <F1
8,232,208,238,169,0,141,32,208 <DB
80 DATA 141,33,208,200,192,30,208,2,1 <80
60,0,140,86,192,76,49,234,0 <80
90 DATA 0,0,6,9,11,2,8,4,14,12,10,5,3 <80
,15,7,13,1 <80
100 DATA 1,13,7,15,3,5,10,12,14,4,8,2, <80
11,9,6,0 <80

```

Animator

Különböző sprite-ok felhasználásával látványos animációkat tervezhetünk. A képernyő tetején egy 4-szer 4-es sprite-tábla látható, ezen dolgozunk. Az egyes billentyűk funkciói a következők:

Space és Shift+Space – az aktuális (villogó) sprite blokkjának növelése vagy csökkentése

N – közvetlenül tetszőleges blokkot adhatunk meg aktuális sprite-nak

Shift és Ctrl – bekér egy blokkszámot, majd az összes látható sprite ennek a formáját veszi fel

E – megadott sorszámú fázis szerkesztése

D – adott sorszámú fázis kirajzolása

Return – az „editing” után látható fázis elmentése a memóriába

F1 és F3 – a „draw”-nál látható fázist követő/megelőző állapot megjelenítése

K – sprite-ok betöltése adott blokktól kezdve

L – animációs fázisok betöltése a megadott sorszámtól kezdve

S – animációs fázisok kimentése. A kezdő és végső fázisok sorszámát valamint a nevet kell megadni,

A – animáció. A kezdő és végső fázisokat, valamint a sebességet kell megadnunk (1 a leggyorsabb)

Q – kilépés a programból.

A sprite-ok alatt látható AKT szám a villogó sprite mutatója. Mivel a program multicolor sprite-okat jelenít meg, elképzelhető, hogy az aktuális sprite nem tartalmazza a villogás színeit. Ilyenkor nem tudjuk megkülönböztetni.

Figyelem! A program begépelése vagy betöltése előtt mindig ki kell adnunk egy POKE 44,12: POKE 3072,0: NEW parancsot.

Írta: Kovács Gergely

```

10 FOR X=53176 TO 53247 <C1
20 READ A:POKE X,A:E=E+A <74
30 NEXT X <FC
40 IF E<>8728 THEN PRINT "HIBA 50-150 <16
   ":END
50 DATA 32,87,226,162,8,134,186 <09
60 DATA 32,121,0,240,44,32,253 <27
70 DATA 174,32,138,173,32,247,183 <C1
80 DATA 72,32,121,0,240,21,104 <32
90 DATA 132,193,133,194,32,253,174 <41
100 DATA 32,138,173,32,247,183,132 <2E

```

```

110 DATA 174,133,175,76,237,245,104 <03
120 DATA 132,195,133,196,160,0,44 <E8
130 DATA 160,1,132,185,169,0,76 <4B
140 DATA 165,244,60,54,52,39,69 <D9
150 DATA 82,62 <8A
160 : <7B
170 FOR X=49152 TO 49486 <30
180 READ A:POKE X,A:E=E+A <4F
190 NEXT X <59
200 IF E<>49441 THEN PRINT "HIBA 210-6 <02
   80":END
210 DATA 120,169,36,141,20,3,169 <A2
220 DATA 192,141,21,3,169,48,141 <5F
230 DATA 18,208,173,17,208,41,127 <0B
240 DATA 141,17,208,169,129,141,26 <A0
250 DATA 208,169,127,141,13,220,88 <05
260 DATA 96,173,0,207,141,0,208 <D7
270 DATA 173,1,207,141,1,208,173 <41
280 DATA 2,207,141,2,208,173,3 <59
290 DATA 207,141,3,208,173,4,207 <FC
300 DATA 141,4,208,173,5,207,141 <63
310 DATA 5,208,173,6,207,141,6 <F1
320 DATA 208,173,7,207,141,7,208 <94
330 DATA 173,8,207,141,8,208,173 <63
340 DATA 9,207,141,9,208,173,10 <EB
350 DATA 207,141,10,208,173,11,207 <23
360 DATA 141,11,208,173,12,207,141 <33
370 DATA 12,208,173,13,207,141,13 <F4
380 DATA 208,173,14,207,141,14,208 <C9
390 DATA 173,15,207,141,15,208,169 <76
400 DATA 36,141,24,208,169,88,162 <7C
410 DATA 168,160,192,32,47,193,238 <3A
420 DATA 25,208,173,32,207,41,8 <A3
430 DATA 208,9,174,32,207,173,33 <EC
440 DATA 207,32,57,193,76,188,254 <00
450 DATA 173,16,207,141,0,208,173 <17
460 DATA 17,207,141,1,208,173,18 <F2
470 DATA 207,141,2,208,173,19,207 <CC
480 DATA 141,3,208,173,20,207,141 <AD
490 DATA 4,208,173,21,207,141,5 <56
500 DATA 208,173,22,207,141,6,208 <E3
510 DATA 173,23,207,141,7,208,173 <EA
520 DATA 24,207,141,8,208,173,25 <18
530 DATA 207,141,9,208,173,26,207 <97
540 DATA 141,10,208,173,27,207,141 <6E
550 DATA 11,208,173,28,207,141,12 <D4
560 DATA 208,173,29,207,141,13,208 <68
570 DATA 173,30,207,141,14,208,173 <4B
580 DATA 31,207,141,15,208,169,20 <1A
590 DATA 141,24,208,169,48,162,36 <47
600 DATA 160,192,32,47,193,238,25 <08
610 DATA 208,173,32,207,41,8,240 <FF
620 DATA 12,173,32,207,41,7,170 <38
630 DATA 173,33,207,32,57,193,76 <10
640 DATA 49,234,141,18,208,142,20 <5E
650 DATA 3,140,21,3,96,157,39 <95
660 DATA 208,173,18,208,24,105,30 <CF
670 DATA 205,18,208,208,251,173,34 <4D
680 DATA 207,157,39,208,96,0 <85
690 : <6D
700 FOR X=49500 TO 49623 <4B
710 READ A:POKE X,A:E=E+A <CF
720 NEXT X <5A
730 IF E<>69270 THEN PRINT "HIBA 740-9 <5B
   10":END
740 DATA 173,82,193,141,203,193,173 <4D
750 DATA 83,193,141,204,193,173,85 <58
760 DATA 193,141,91,193,160,0,169 <F7
770 DATA 200,205,18,208,208,251,162 <C6
780 DATA 100,202,208,253,206,91,193 <8E
790 DATA 208,239,173,85,193,141,91 <F5
800 DATA 193,162,0,169,11,141,207 <92
810 DATA 193,169,248,141,206,193,169 <99
820 DATA 8,141,90,193,32,202,193 <4B
830 DATA 169,7,141,207,193,169,240 <F6
840 DATA 141,206,193,169,16,141,90 <9C
850 DATA 193,32,202,193,169,16,24 <31
860 DATA 109,203,193,141,203,193,144 <A5
870 DATA 3,238,204,193,200,204,84 <28
880 DATA 193,208,175,96,255,255,255 <8B

```

```

890 DATA 255,255,255,255,255,189,64 <5E
900 DATA 65,157,240,7,232,236,90 <3D
910 DATA 193,208,244,96,0 <52
920 : <8A
925 PRINT "PLEASE WAIT . . .":FOR T=20 <77
    48 TO 3048:POKE T,32:NEXT :POKE 53
    030,123
930 FOR T=0 TO 15 <4F
940 READ A:POKE 52992+T*2,A+100 <96
950 READ A:POKE 52993+T*2,A <E9
960 NEXT <9E
970 FOR T=0 TO 7:POKE 53287+T,12:NEXT <6D
980 POKE 53276,255:POKE 53285,11:POKE <D6
    53286,0:POKE 53281,15:POKE 53280,0
990 POKE 53024,0:REM-----AKT SPRITE- <BC
    --
1000 POKE 53025,12:REM----AKT SPRITE CO <5D
    LOR
1010 POKE 53026,12:REM-----SPRITE COLO <65
    R--
1020 POKE 53269,255 <E3
1030 PRINT "[CTRL/1][SH/CLR] <82
1040 SYS 49152 <73
1050 DATA 24,50 <5C
1060 DATA 48,50 <10
1070 DATA 72,50 <D1
1080 DATA 96,50 <45
1090 DATA 24,71 <33
1100 DATA 48,71 <E4
1110 DATA 72,71 <A5
1120 DATA 96,71 <19
1130 DATA 24,92 <07
1140 DATA 48,92 <7B
1150 DATA 72,92 <38
1160 DATA 96,92 <AF
1170 DATA 24,113 <3E
1180 DATA 48,113 <B5
1190 DATA 72,113 <73
1200 DATA 96,113 <C4
1210 P=0:REM-----AKT SPRITE <CD
1220 E=0:REM-----EDITING FAZIS <4F
1230 X=0:REM-----KKIRAJZOLT FAZIS <2A
1240 M=0:REM-----MAX FAZISSZAM <B3
1250 D=16384:REM--PTR-EK CIME <69
1260 GOTO 1680 <67
1270 B=2032+P:IF P<7 THEN B=3064+P <86
1280 PRINT "[HOME][11DOWN]AKT[10SPC][5L <65
    EFT]:PEEK(B)
1290 PRINT "EDITING[3SPC][2LEFT]:E/16 <A7
1300 PRINT "DRAWN[5SPC][2LEFT]:X/16 <46
1310 POKE 53025,2:POKE 53025,8:POKE 530 <EB
    25,10:POKE 53025,12
1320 GET A$:IF A$="" THEN 1310 <AF
1330 IF A$="[DOWN]" THEN IF P<12 THEN P <3B
    =P+4
1340 IF A$="[UP]" THEN IF P>3 THEN P=P- <41
    4
1350 IF A$="[RIGHT]" THEN IF P<15 THEN <43
    P=P+1
1360 IF A$="[LEFT]" THEN IF P>0 THEN P= <91
    P-1
1370 POKE 53024,P <4D
1380 REM-----FUNCTIONS----- <AE
1390 IF A$=" " OR A$=CHR$(160) THEN 153 <85
    0
1400 IF A$="[F1]" THEN X=X+16:GOTO 1680 <9B
1410 IF A$="[F3]" THEN X=X-16:GOTO 1680 <9F
1420 IF A$="N" THEN 1580 <5D
1430 IF A$="[SH/CLR]" THEN 1640 <29
1440 IF A$="E" THEN 1740 <A5
1450 IF A$=CHR$(13) THEN GOSUB 1710:E=E <71
    +16
1460 IF A$="D" THEN 1770 <28
1470 IF A$="S" THEN 1800 <3C
1480 IF A$="L" THEN 1890 <75
1490 IF A$="Q" THEN 1960 <84
1500 IF A$="A" THEN 2000 <88
1510 IF A$="K" THEN 2150 <D9
1520 GOTO 1270 <43
1530 REM-----(+ OR -)----- <EB
    --
1540 B=2032+P <FF
1550 IF P<8 THEN B=3064+P <BE
1560 IF A$=" " THEN POKE B,PEEK(B)+1:GO <95
    TO 1270
1570 POKE B,PEEK(B)-1:GOTO 1270 <47
1580 REM----- (NUMBER)----- <A3
    --
1590 PRINT "[SH/CLR][16DOWN]SZAM";:INPU <1B
    T A:PRINT "[SH/CLR]
1600 B=2032+P <C0
1610 IF P<8 THEN B=3064+P <63
1620 POKE B,A <DA
1630 GOTO 1270 <78
1640 REM----- (CLEAR)----- <7A
    --
1650 PRINT "[SH/CLR][17DOWN]TORLO MINTA <F3
    ";:INPUT A:PRINT "[SH/CLR]
1660 FOR T=0 TO 7:POKE 2040+T,A:POKE 30 <45
    64+T,A:NEXT
1670 GOTO 1270 <BC
1680 REM----- (DRAW ANOTHER)----- <5C
1690 FOR T=0 TO 7:POKE 2040+T,PEEK(D+X+ <BF
    T+8):POKE 3064+T,PEEK(D+X+T):NEXT
1700 GOTO 1270 <6C
1710 REM----- ($SAVE IT)----- <94
    --
1720 FOR T=0 TO 7:POKE D+E+T+8,PEEK(204 <D5
    0+T):POKE D+E+T,PEEK(3064+T):NEXT
1730 RETURN <2C
1740 REM----- (EDIT ONE)----- <68
1750 PRINT "[SH/CLR][14DOWN]EDIT";:INPU <29
    T A
1760 PRINT "[SH/CLR]:E=A*16:GOTO 1270 <3F
1770 REM----- (DRAW ONE)----- <9C
1780 PRINT "[SH/CLR][15DOWN]DRAW";:INPU <EF
    T A
1790 X=A*16:GOTO 1680 <A7
1800 REM----- (SAVE)----- <62
1810 PRINT "[HOME][17DOWN]SAVE" <A5
1820 PRINT "FROM";:INPUT A <64
1830 PRINT "TO[2SPC]";:INPUT B <89
1840 PRINT "NAME";:INPUT NM$ <64
1850 POKE 53269,0 <E5
1860 SYS 53176NM$,D+A*16,D+B*16 <F9
1870 POKE 53269,255:PRINT "[SH/CLR] <E9
    1880 GOTO 1270 <5B
1890 REM----- (LOAD)----- <3F
    -
1900 PRINT "[HOME][14DOWN]LOAD ANIMATIO <A0
    N"
1910 PRINT "FROM";:INPUT A <D3
1920 PRINT "NAME";:INPUT NM$ <A3
1930 POKE 53269,0 <AE
1940 SYS 53176NM$,D+A*16 <A1
1950 POKE 53269,255:PRINT "[SH/CLR]:GO <37
    TO 1680
1960 REM----- (QUIT)----- <25
    -
1970 PRINT "[SH/CLR][14DOWN]ARE YOU SUR <7D
    E YOU WANT TO QUIT (Y/N) ";:INPUT A
    $
1980 IF A$="Y" THEN SYS 64738 <9E
1990 PRINT "[SH/CLR]:GOTO 1270 <B0
2000 REM-----ANIMACIO----- <F0
    -
2010 PRINT "[SH/CLR][14DOWN]ANIMACIO" <9F
2020 PRINT "BEGIN";:INPUT A <93
2030 PRINT "END[2SPC]";:INPUT B <FA
2040 POKE 49491,64+INT(A/16) <51
2050 POKE 49490,(A-INT(A/16)*16)*16 <E3
2060 REM↑↑↑↑ ANIMDAT PTR ↑↑↑↑↑ <98
2070 POKE 49492,B-A:REM-----FAZISOK SZ <EC
    AMA--
2080 PRINT "SPEED";:INPUT A <F5
2090 POKE 49493,A:REM---SEBESSEG 1/60SE <3A
    C-BEN
2100 POKE 53025,PEEK(53026) <6A

```

```

2110 PRINT "[SH/CLR][11DOWN]PRESS A KEY <6D
! [2SPC]<=>TO[SH/SPC]EXIT":POKE 198
,0:WAIT 198,1:GET A$
2120 IF A$="←" THEN PRINT "[SH/CLR]":GO <4F
TO 1270
2130 SYS 49500 <62
2140 GOTO 2110 <E4
2150 REM------(LOAD SPRITESET)----- <2E
-
2160 PRINT "[SH/CLR][12DOWN]LOAD SPRITE <D0
S":PRINT "FILENAME";:INPUT NM$
2170 PRINT "FROM[4SPC]";:INPUT A:POKE 5 <01
3269,0
2180 SYS 53176NM$,8192+A*64:PRINT "[SH/ <4B
CLR]
2190 POKE 53269,255:GOTO 1270 <41

```

Diagram

A program egy oszlopdiaagram-rajzoló rutint tartalmaz, amely látványosan, a képernyő tetszőleges helyére képes akár negatív magasságú oszlopot is megjeleníteni.

Bemutatóként egy szinuszgörbét rajzol, ez akár egy bioritmus-program magja is lehetne.

Írta: **Lantos Zoltán**

```

10 REM OSZLOPDIAGRAM RAJZOLO - 1992 L <9B
ANTOS ZOLTAN - ANONYMOUS SOFTWARE
20 : <55
30 SYS 58692 <51
40 : <B6
50 REM ===== <60
60 REM == FORMA == <77
70 REM ===== <C1
80 : <79
90 DATA 32,100,111,121,98,248,247,227 <3A
,170
100 DATA 32,99,119,120,226,249,239,228 <EA
,160
110 FOR I=1 TO 2:FOR D=0 TO 8 <6C
120 READ T(I,D) <EB
130 NEXT D,I <31
140 : <F9
150 REM ===== <1B
160 REM == SZINUSZ FGV == <5A
170 REM ===== <84
180 : <FC
190 DEF FN X(A)=SIN(A/4/π)*50 <7E
200 : <0D
210 Y=12 <80
220 FOR C=2 TO 78 STEP 2 <DE
230 X=C/2:M=FN X(C):REM AZ OSZLOPDIAGR <7F
AM MAGASSAGA
240 : <10
250 REM ===== <DF
260 REM == OSZLOP RAJZOLAS == <AC
270 REM ===== <5E
280 : <13
290 KP=1024+X+40*Y <95
300 L=SGN(M):P=L*40:D=1:N=0:IF L=-1 TH <BC
EN D=2:N=-1:KP=KP-P
310 FOR I=1 TO M STEP L <49

```

Monitor plus

A program egy rövid és egyszerű monitor, amellyel a jól kódolók pillanatok alatt megírhatnak rövidebb rutinokat. A kezdőcím beírása után a bájtokat kéri (decimálisan). Gépelés közben segítségként láthatjuk a legfontosabb bájtok memóriaikonjait, valamint az aktuális memóriacímet is.

Írta: **Horváth Csaba**

```

90 POKE 53280,0:POKE 53281,12:PRINT " <5E
[SH/CLR][CTRL/2]":PRINT CHR$(8);:P
RINT CHR$(142);
91 PRINT "[3SPC]* MONITOR+:[2SPC]MR. <BA
HORV[U]P].[DOWN][LEFT]ATH CSABA *"
92 PRINT "CIM?":GOSUB 104 <BD
93 IF A>65535 OR A<256 THEN GOTO 92 <E6
94 X=A <44
95 PRINT "[SH/CLR]LDA$ZZZZ=173;LDX$ZZ <A7
ZZ=174;LDY$ZZZZ=172"
96 PRINT "STA$ZZZZ=141;STX$ZZZZ=142;S <73
TY$ZZZZ=141"
97 PRINT "-[DOWN][LEFT]NOP=234;SEI=12 <7A
0;RTS=96;RTI=64"
98 PRINT "DEC$ZZZZ=206;LDA$ZZZZ,Y=188 <D8
;LD#ZZ=169"
99 PRINT "-[DOWN][LEFT]JMP$ZZZZ=76;CL <BE
I=88"
100 GOTO 130 <D8
104 B$="" <BB
105 INPUT "[DOWN][2RIGHT]*[3LEFT]";A <BB
116 RETURN <EE
130 PRINT "CIM:";X <E6
131 PRINT "DATA: ";PEEK(X);"[LEFT][2SP <28
C]":INPUT "UJ DATA:[6SPC][6LEFT]";
D
132 IF D>255 OR D<0 THEN GOTO 131 <D0
133 POKE X,D:X=X+1 <61
134 PRINT "[3UP]";:GOTO 130 <79

```

FLOPPY-POSTA
A programok
lemezen is
megrendelhetők!

Grafika

A következő program tulajdonképpen egy grafikus szubrutinyűjtemény. Segítségével tetszőleges grafikákat rajzolhatunk. Az egyes funkciókat menübe foglalták, itt önállóan is kipróbálhatjuk őket.

Írta: **Lantos Zoltán**

```

10 REM GRAFIKAI RUTINOK - 1992 LANTOS <72
    ZOLTAN - ANONYMOUS SOFTWARE
20 : <55
30 SYS 58692:POKE 55,0:POKE 56,64:CLR <CA
40 POKE 211,0:POKE 214,5:SYS 58640 <85
50 PRINT SPC(13);"GRAFIKA DEMO" <42
60 PRINT SPC(12);"-----" <DA
70 PRINT TAB(9)"[DOWN]A .. GRAFIKUS K <6B
    EPRE VALTAS"
80 PRINT TAB(9)"B .. KEP MENTESE" <84
90 PRINT TAB(9)"C .. KEP BETOLTESE" <2A
100 PRINT TAB(9)"D .. VONAL HUZASA" <91
110 PRINT TAB(9)"E .. KOR RAJZOLASA" <AC
120 PRINT TAB(9)"F .. PONT RAJZOLASA" <FC
130 PRINT TAB(9)"G .. KEP TORLESE" <02
140 PRINT TAB(9)"H .. KILEPES" <74
150 : <BA
160 GET A$:X=ASC(A$+CHR$(0))-64:IF X<1 <72
    OR X>8 THEN 160
170 ON X GOTO 230,310,510,690,840,980, <50
    1100,1200
180 : <FC
190 REM ===== <B8
200 REM == GRAFIKA == <40
210 REM ===== <48
220 : <8E
230 GOSUB 1290 <97
240 GET A$:IF A$="" THEN 240 <52
250 GOSUB 1400:GOTO 160 <F6
260 : <91
270 REM ===== <33
280 REM == KEP MENTESE == <D6
290 REM ===== <BC
300 : <43
310 A$="":INPUT "[DOWN]FILE NEV ";A$:I <B4
    F A$="" THEN 30
320 GOSUB 1290 <0D
330 OPEN 15,8,15,"S:"+A$:CLOSE 15 <AF
340 OPEN 1,8,2,A$+",P,W" <FD
350 : <07
360 REM KEPMEMORIA <E5
370 FOR X=24576 TO 32768 <2D
380 PRINT#1,CHR$(PEEK(X)); <B7
390 NEXT X <46
400 REM SZINMEMORIA: 17408-18407 <60
410 : <BA
420 CLOSE 1 <95
430 GET A$:IF A$="" THEN 440 <C0
440 GOSUB 1400 <2E
450 GOTO 30 <D0
460 : <7E
470 REM ===== <8A
480 REM == KEP BETOLTESE == <FE
490 REM ===== <0B
500 : <50
510 A$="":INPUT "[DOWN]FILE NEV ";A$:I <08
    F A$="" THEN 30
520 GOSUB 1290 <99
530 OPEN 1,8,0,A$+",P,R" <11
540 : <53
550 REM KEPMEMORIA <D1
560 FOR X=24576 TO 32768 <DA
570 GET#1,A$ <6F
580 POKE X,ASC(A$+CHR$(0)) <D6
590 NEXT X <F3
600 CLOSE 1 <80
610 GET A$:IF A$="" THEN 620 <CB
620 GOSUB 1390 <8E
630 GOTO 30 <FB
640 : <29
650 REM ===== <EE
660 REM == VONAL HUZASA == <33
670 REM ===== <6F
680 : <2C
690 INPUT "[DOWN]X1 ";X1 <C2
700 INPUT "Y1 ";Y1 <0C
710 INPUT "X2 ";X2 <2D
720 INPUT "Y2 ";Y2 <63
730 : <BE
740 GOSUB 1290 <A7
750 GOSUB 1540 <77
760 GET A$:IF A$="" THEN 760 <84
770 GOSUB 1400 <15
780 GOTO 30 <D5
790 : <23
800 REM ===== <67
810 REM == KOR RAJZOLASA == <C5
820 REM ===== <E6
830 : <15
840 INPUT "X ";X <34
850 INPUT "Y ";Y <F9
860 INPUT "R ";R <AF
870 : <18
880 GOSUB 1290 <21
890 GOSUB 1700 <AC
900 GET A$:IF A$="" THEN 900 <07
910 GOSUB 1400 <3D
920 GOTO 30 <BD
930 : <CB
940 REM ===== <9B
950 REM == PONT RAJZOLASA == <CE
960 REM ===== <E5
970 : <CE
980 INPUT "[DOWN]X ";X <5D
990 INPUT "Y ";Y <90
1000 GOSUB 1290 <86
1010 GOSUB 1480 <19
1020 GET A$:IF A$="" THEN 1020 <81
1030 GOSUB 1400 <B5
1040 GOTO 30 <8B
1050 : <B2
1060 REM ===== <67
1070 REM == KEP TORLESE == <8D
1080 REM ===== <81
1090 : <76
1100 PRINT "BIZTOSAN (I/N) ?" <5B
1110 GET A$:IF A$<>"I" AND A$<>"N" THEN <2A
    1110
1120 IF A$="N" THEN 30 <2A
1130 POKE 56,160:CLR :DIM A$(10000):CLR <92
1140 GOTO 30 <D1
1150 : <59
1160 REM ===== <0B
1170 REM == VEGE == <8D
1180 REM ===== <A9
1190 : <1C
1200 PRINT "[SH/CLR][DOWN]VISZLAT !":PO <2B
    KE 55,0:POKE 56,160:CLR
1210 END <0A
1220 : <6D
1230 REM ===== <A5
1240 REM == RUTINOK == <1A
1250 REM ===== <0B
1260 : <D0
1270 REM GRAPHICS <5D
1280 : <71
1290 BQ=PEEK(53280):PQ=PEEK(53281):MQ=P <85
    EEK(53270):SQ=PEEK(53265):JQ=PEEK(
    53272)

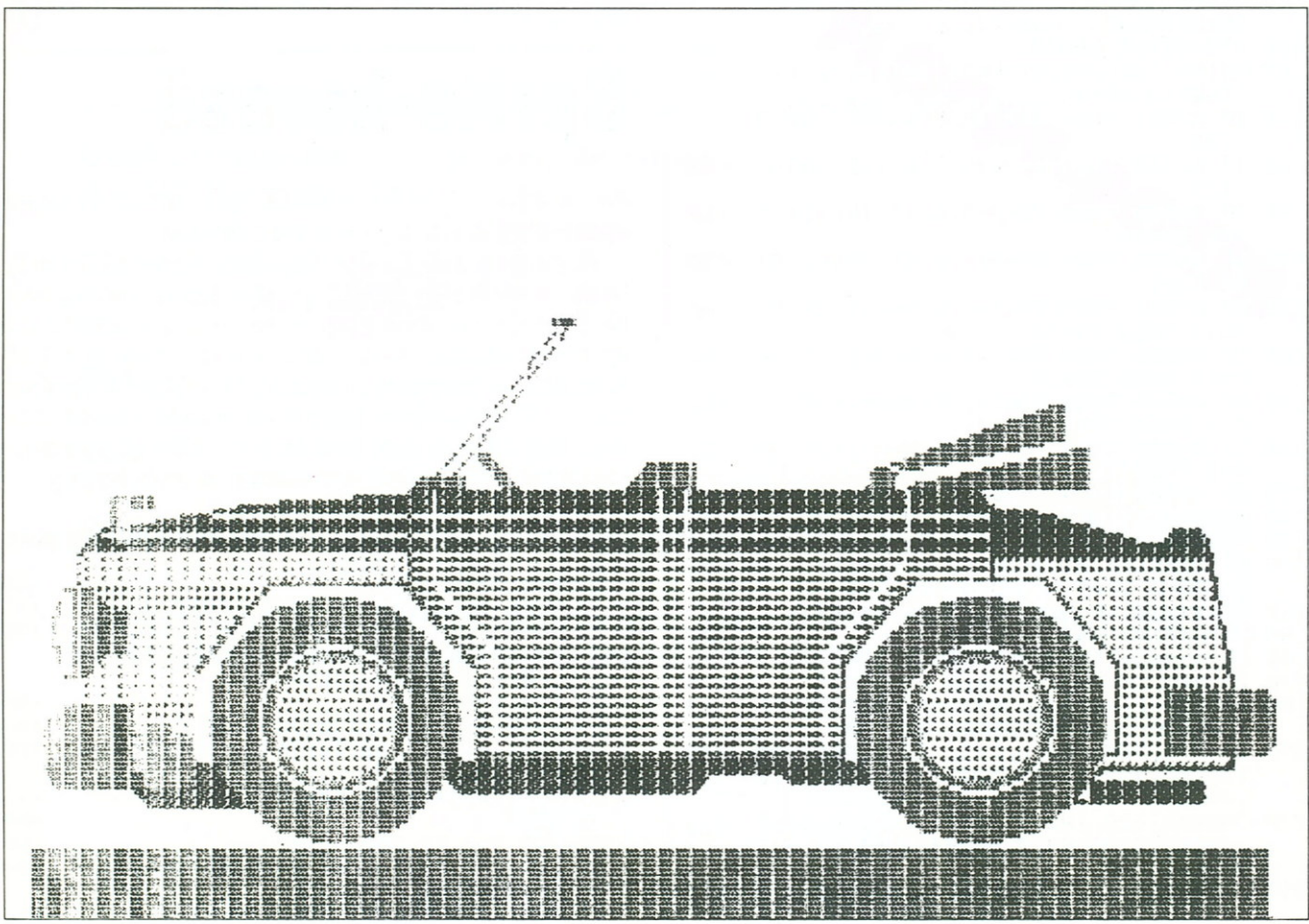
```



```

1300 XQ=PEEK(211):YQ=PEEK(214):CQ=PEEK( <A5 1500 RETURN <4F
56576):DQ=PEEK(648):POKE 56576,198 1510 : <EF
1310 POKE 648,68:POKE 53265,0 <6D 1520 REM LINE <03
1320 GQ$="":FOR IQ=216 TO 240:GQ$=GQ$+C <D3 1530 : <91
HR$(PEEK(IQ)):NEXT :SYS 58651:POKE 1540 IF X1<>X2 THEN 1600 <91
199,1 1550 IF Y1=Y2 THEN X=X1:Y=Y1:GOSUB 1480 <55
1330 KQ$=CHR$(166):FOR KQ=1 TO 7:KQ$=KQ <83 1560 QV=0 <7D
$+KQ$ 1570 FOR Y=Y1 TO Y2 STEP SGN(Y2-Y1) <B4
1340 FOR LQ=7-KQ TO 3:PRINT KQ$;:NEXT L <72 1580 X=X1+(Y-Y1)*QV+0.5:GOSUB 1480 <15
Q,KQ:PRINT KQ$;LEFT$(KQ$,73); 1590 NEXT :RETURN <5C
1350 PRINT CHR$(19):POKE PEEK(648)*256+ <C2 1600 IF Y1<>Y2 THEN 1650 <AF
999,230:POKE 53272,29:POKE 53265,5 1610 QF=0 <09
9 1620 FOR X=X1 TO X2 STEP SGN(X2-X1) <D7
1360 RETURN <67 1630 Y=Y1+(X-X1)*QF+0.5:GOSUB 1480 <94
1370 : <07 1640 NEXT :RETURN <B3
1380 REM TEXT <7D 1650 QF=(Y2-Y1)/(X2-X1):IF ABS(QF)>1 TH <84
1390 : <A9 EN QV=1/QF:GOTO 1570
1400 SYS 58648:POKE 53280,BQ:POKE 53281 <26 1660 GOTO 1620 <FA
,PQ:POKE 53270,MQ:POKE 53265,0 1670 : <3A
1410 FOR IQ=216 TO 240:POKE IQ,ASC(MID$ <59 1680 REM CIRCLE <8A
(GQ$,IQ-215)):NEXT 1690 : <DC
1420 POKE 53272,JQ:POKE 56576,CQ:POKE 6 <D5 1700 C=R:P=0:N=0:H=X:G=Y <80
48,DQ:POKE 211,XQ:POKE 214,YQ:SYS 1710 N=N+P:IF N>=C THEN R=R-1:C=C+R <A8
58640 <09 1720 X=H+R:Y=G+P:GOSUB 1480:Y=G-P:GOSUB <D3
1430 POKE 53265,SQ <AB 1480
1440 RETURN <CC 1730 X=H-R:GOSUB 1480:Y=G+P:GOSUB 1480 <3C
1450 : <82 1740 X=H+P:Y=G+R:GOSUB 1480:Y=G-R:GOSUB <0A
1460 REM PLOT /0-319,0-199/ <6D 1480
1470 : <B7 1750 X=H-P:GOSUB 1480:Y=G+R:GOSUB 1480 <50
1480 OY=320*INT(Y/8)+(Y AND 7):OX=8*INT <B7 1760 P=P+1 <13
(X/8):MA=2*((7-X) AND 7):AV=24576+ <0A 1770 IF R<P THEN RETURN <64
OY+OX 1780 GOTO 1710 <F2
1490 POKE AV,PEEK(AV) OR MA <2E

```



Autóverseny

A Commodore 64-re írt játékban a feladat a többi autó kikerülése. A játék Joystick Port 1-ről irányítható (tűz-gyorsítás, hátra-lassítás).

Írta: **Horváth Csaba**

```

1 REM *** AUTOVERSENY *** <D9
    *** C-64-RE ***
2 REM ***KESZITETTE: *** <10
    ***HORVATH CSABA***
3 POKE 775,200:POKE 808,254 <D6
4 GOTO 1000 <E3
5 CLR :TI$="000000" <E0
15 PRINT "[SH/CLR][CTRL/2]":POKE 5328 <FA
    0,0:POKE 53281,0:V=53248
25 FOR I=832 TO 894:READ A:POKE I,A:N <AB
    EXT
35 FOR I=896 TO 958:READ A:POKE I,A:N <43
    EXT
45 POKE 2040,13:POKE 2041,13:POKE V+3 <6C
    9,14:POKE V+40,1:POKE V+23,7:POKE
    V+29,7
50 FOR I=0 TO 24:POKE 1036+I*40,160:P <64
    OKE 55308+I*40,5
65 POKE 1051+I*40,160:POKE 55323+I*40 <40
    ,5:NEXT
75 POKE V,168:POKE V+1,170:POKE V+21, <19
    3:X=168
85 POKE V+2,168:POKE V+3,0:HX=168:HY= <61
    0
90 POKE V+30,0:POKE V+31,0: <6F
91 POKE V+23,4:POKE V+29,4 <A3
95 TI$="000000":TR=2 <FD
96 POKE V+23,7:POKE V+29,7 <2B
99 POKE 56320,127 <AD
100 A=PEEK(56321):POKE 54296,1 <76
105 SE=TR*10+10:L=TI+SE:PRINT "[HOME][ <35
    RIGHT]PONT:";L:PRINT "[HOME][3DOWN
    ]SEBESS[UP].[DOWN][LEFT]EG : "
106 ATS=SE+SE:ER=ER+1 <02
107 PRINT "[DOWN][RIGHT]";INT(SE/1.5); <68
    "[LEFT] MP/H ";
110 IF A=251 THEN X=X-INT(.05*SE):GOTO <9D
    127
120 IF A=247 THEN X=X+INT(.05*SE):GOTO <C3
    127
121 IF A=239 THEN TR=TR+.5:IF TR>=14 T <EF
    HEN TR=14
122 IF A=253 THEN TR=TR-.5:IF TR<=1 TH <28
    EN TR=1
123 IF A=231 THEN TR=TR+.5:X=X+1:IF TR <CD
    >=14 THEN TR=14
124 IF A=235 THEN TR=TR+.5:X=X-1:IF TR <D3
    >=14 THEN TR=14
125 IF A=245 THEN TR=TR-.5:X=X+1:IF TR <5C
    <=1 THEN TR=1
126 IF A=249 THEN TR=TR-.5:X=X-1:IF TR <22
    <=1 THEN TR=1
127 IF X>206 THEN X=206:GOTO 130 <8B
128 IF X<114 THEN X=114:GOTO 130 <AF
130 POKE V,X:POKE 54296,INT(SE/16*1.7) <9C
140 IF PEEK(V+30)=3 OR PEEK(V+31)=1 TH <51
    EN POKE 2040,14:GOTO 400
150 HY=HY+TR:IF HY>240 THEN HY=30 <8A
160 HX=HX+INT(RND(TI)*5)-2 <8B
165 IF HX<120 THEN HX=120 <F1
170 IF HX>216 THEN HX=216 <1D
180 POKE V+2,HX:POKE V+3,HY:POKE 54296 <6B
    ,INT(SE/16*1.7):GOTO 100
200 DATA 3,255,192,3,255,192,3,255,192 <B2
    ,3,255,192,3,255,192,3,255,192,3,0
    ,192
210 DATA 3,129,192,2,195,64,2,126,64,2 <11
    ,60,64,2,36,64,2,60,64,2,60,64,2,6
    0,64,2
213 DATA 126,64,2,195,64,3,129,192,3,0 <9F
    ,192,3,255,192,3,129,192
300 DATA 0,3,24,1,34,168,49,36,83,73,3 <08
    6,23,53,40,39,20,202,46,8,3,14,0,3
    0,158
310 DATA 101,8,92,23,196,28,63,242,24, <BD
    31,252,0,63,255,48,31,255,176,31,2
    55,128
320 DATA 63,255,192,15,223,224,7,79,22 <E8
    4,2,71,192,0,2,64,0,0,0,0
400 PRINT "[SH/CLR]" <87
401 PRINT "[7DOWN]":PRINT "[2RIGHT]PON <9F
    T :";TI:
403 FOR Y=0 TO 1700:NEXT :RUN 5 <16
1000 PRINT CHR$(8):PRINT CHR$(142):PRIN <C4
    T "[SH/CLR][CTRL/1]":POKE 53280,1:
    POKE 53281,1
1001 PRINT "[DOWN][5SPC]*****" <F3
    *****"
1002 FOR WE=0 TO 2:PRINT "[5SPC]***[22S <5A
    PC]***":NEXT
1003 PRINT "[5SPC]***[5SPC]AUT[UP].[DOW <20
    N][LEFT]OVERSENY[6SPC]***"
1004 FOR QE=0 TO 2:PRINT "[5SPC]***[22S <FF
    PC]***":NEXT
1005 PRINT "[5SPC]*****" <63
    *****"
1010 PRINT "[4DOWN]" <E4
1011 PRINT "[DOWN][5RIGHT]NYOMJ[UP].[DD <CD
    WN][LEFT]AL LE EGY BILLENTY[UP][SH
    /R][DOWN][LEFT]UT"
1012 GET A$:A$="" <73
1013 GET A$:IF A$="" THEN 1013 <7C
1014 GOTO 15 <7A

```

Sprite-kereső

Az alábbi, C-64-re készült pár soros rutinnal sprite-okat kereshetünk a memóriában.

A program elején a P változóban kell megadni, hogy melyik 16 kilobájtos területen mozogjunk (0-3), a O-ban pedig az egymás mellé rajzolható sprite-ok számát (így akár 2 x 4-es, vagy 8 x 1-es alakzatot is megjeleníthetünk). Rus/Stop + Restore után a T változóban kapjuk az aktuális blokk számát. Ezt 64-gyel szorozva és P x 16384-et hozzáadva, a sprite első bajtjának memóriacímét kapjuk.

Írta: **Kovács Gergely**

```

1 P=1:O=2 <C8
2 POKE 56576,151-P:POKE 53272,20 <1F
3 POKE 53276,255:POKE 53269,255:POKE <AE
    53285,9:POKE 53286,0:POKE 53281,5
    :PRINT "[CTRL/6][SH/CLR]"
4 FOR T=0 TO 7 <6E
5 POKE 53287+T,10 <A3
6 POKE 53248+T*2,24+T*24-(INT(T/O)* <DE
    (O*24))
7 POKE 53249+T*2,50+INT(T/O)*21 <11
8 NEXT <BA
9 FOR T=0 TO 255 <09
10 FOR Y=0 TO 7:POKE P*16384+2040+Y,T <AF
    +Y:NEXT Y
11 POKE 198,0:WAIT 198,1:NEXT T <A2

```

ELŐFIZETŐI MEGRENDELŐLAP

Egyéni/Közületi előfizetőként megrendelem a negyedévenként 96 oldalon, megjelenő Mikrovilág programfüzetet.

A kiadvány lapszámonkénti ára: 146 Ft, négy lapszám egyéves előfizetési díja: 500 Ft / év (az előfizetői kedvezmény összesen 84 Ft.). Az előfizetési díjat a címemre megküldött postai csekken (számlán) fizetem be (utalom át) az IDG Magyarország Lapkiadó Kft. számlájára.

Név:

Cím:

Cégszerű aláírás:

A Mikrovilág Magazin mellékletében szereplő programokat lemezen is megrendelem (alkalmanként 200 Ft + postaköltség).

igen

nem

Utánvétellel megrendelem a **Mikrovilág Magazin** programmellékletében

közölt programokat példányban lemezen.

Ára: 200 Ft + postaköltség.

Cím: Mikrovilág Programfüzet
1536 Budapest, Pf. 386

Név:

Cím:

Géptípus: C-64

Plus/4

IBM PC

Előfizetőink kedvezményes áron (kéthavi előfizetési díj elszámolásáért) kapják meg a programlemezt, ha ezt a megrendelőlapot kitöltve visszaküldik címünkre.

A Mikrovilág Magazinnak 19. óta vagyok előfizetője.

.....
(aláírás)

100 gramm program – lemezen

Éppen egy évvel ezelőtt jelentettük meg a 100 gramm program című szoftvergyűjteményt, amely most, az új tanév kezdetén ismét aktuálissá vált. Olyan, az iskolai oktatásban is jól használható programokat gyűjtöttünk össze, amelyek szavatossága

még közel sem járt le. A füzet a következő, C-64 és/vagy Plus/4-es gépeken futtatható szoftvereket tartalmazza, amelyeket 5,25 inches floppyra másolva 200 forintért (+ postaköltség) minden megrendelőnknek utánvétellel elküldünk.

	C-64	Plus/4		C-64	Plus/4
GRAFIKUS PROGRAMOK			Emberke		●
Basic-ellenőrök	●	●	Master raster		●
Grafikus és karakteres	●		Trükkös színek	●	
Extragyors rajzoló	●		JÁTÉKPROGRAMOK		
Rajzoló + demo	●		Torpedo	●	
Mandelzoom		●	Mars	●	
Hatalmas betűk		●	Cowboy	●	
Grafikus segítség		●	Ping	●	
FELHASZNÁLÓI PROGRAMOK			Ninja	●	
Grafikus képernyőtörítés	●		OKTATÓ PROGRAMOK		
Képernyőtörő	●		Százalékszámító	●	
Formázott nyomtatás	●		Pithagoraszi számhármások	●	
Renumber	●		Kétismeretlenes egyenlet	●	
Mempeeker	●		Sík egyenlete	●	
Irkáló	●		Axonometrikus torus		●
Átváltó	●		Szögátszámító		●
Gráfrajzoló		●	Kis szorzótábla		●
Megainverz		●	Számosztó		●
List plus		●	Parabola definíciója		●
Pozíciós írás		●	A jó öreg Simon's Basic	●	
Kivilágosító		●	Hengerkiterítés	●	
Mini-DOS	●		Szabályos ötszög	●	
Sprite editor	●		Törtműveletek	●	
Lemez katalógus	●		Területszámoló	●	
TRÜKKÖK			A kör kerülete és területe	●	
Colorscroll	●		Körsor	●	
Sejtautomata		●	Területmérés		●
Super clear		●	Távolságmérés		●
Pattogó raster		●	Ábrázoló 1, 2, 3, 4		●

FLOPPY-POSTA

Utánvétellel megrendelem a Mikrovilág 100 gramm program című kiadványának programlemezét példányban.

Ára: 200 Ft + postaköltség.

Cím: Mikrovilág Programfüzet
1536 Budapest, Pf. 386

Név:

Cím:

Géptípus: C-64

Plus/4

Munkaállomások

Grafika a Silicon-völgyből

Előző számunkban a Silicon Graphics munkaállomásairól írtunk, ahol említettük, hogy a köztudatban elsősorban grafikus alkalmazásokról ismertek. Most bemutatunk egy „csupasz” Iris Indigo-t és megemlítünk néhány grafikus szoftvert is.

Nehogy valaki azt higgye, hogy a lebontott Indigo valami monstrum: alig 38 centi magas, 24 centi széles és 26 centi mély. Persze a profi gépnek szép ára van. Az alábbi szolidabb konfiguráció ára például egymillió forint körül van.

- Alapgép, amely hasonlít a képen szereplőre,
- 16 megabájtos RAM,
- 230 megabájtos winchester,
- 768x1024 felbontású Virtual24 grafikus kártya,
- 16 inches monitor,
- billentyűzet, egér,
- operációs rendszer.

Jópofa a Silicon Graphics egyik szlogenje: az Iris Indigo egyaránt megállja a helyét otthonában és a munkahelyén. És persze a lista korántsem teljes, mert ugye mit ér a remek gép szoftverek nélkül? Márpedig arra aligha számíthat a kedves géptulaj-

donos, hogy majd megveszi vagy lemásolja a kalózmásolatot. A grafikus szoftvereknek ugyanis csillagászati áruk van, és éppen ezért kevés a valószínűsége, hogy kéz alatt terjednek.

A grafikus és animációs szoftverek valóban rengetegbe kerülnek, de természetesen többféle kategóriával találkozhatunk. Az egyik legszínvonalasabb háromdimenziós tervező és animációs programcsomag, a SoftImage milliókba kerül (a szó szoros értelmében!), míg a hasonló, ám jóval kisebb tudású 3D-GO „csak” százazrekbe. A munkaállomásokon is megtaláljuk a híres tervezőprogramot, az AutoCAD-et, amely hasonló árkategóriában szerepel, mint a PC-s változat.

A munkaállomásokat elsősorban grafikai teljesítményük alapján részesítik előnyben, de a nagy sebesség jól jön a tudomány sok-sok területén, mint például a molekula-modellezésben, a térinformatikában, de természetesen csodásan használható akár kiadványszerkesztésre is.

Az adatokat és az illusztrációt a Creative Engineering Kft.-től kaptuk.

Bognár Ákos

A Motorola 56001 DSP chip – része az audiorendszernek, amely analóg és digitális jelet egyaránt fogad, így további digitális hangfeldolgozás is lehetséges

Bővíthető SIMM modulokban található memória – 8–96 megabájtig bővíthető

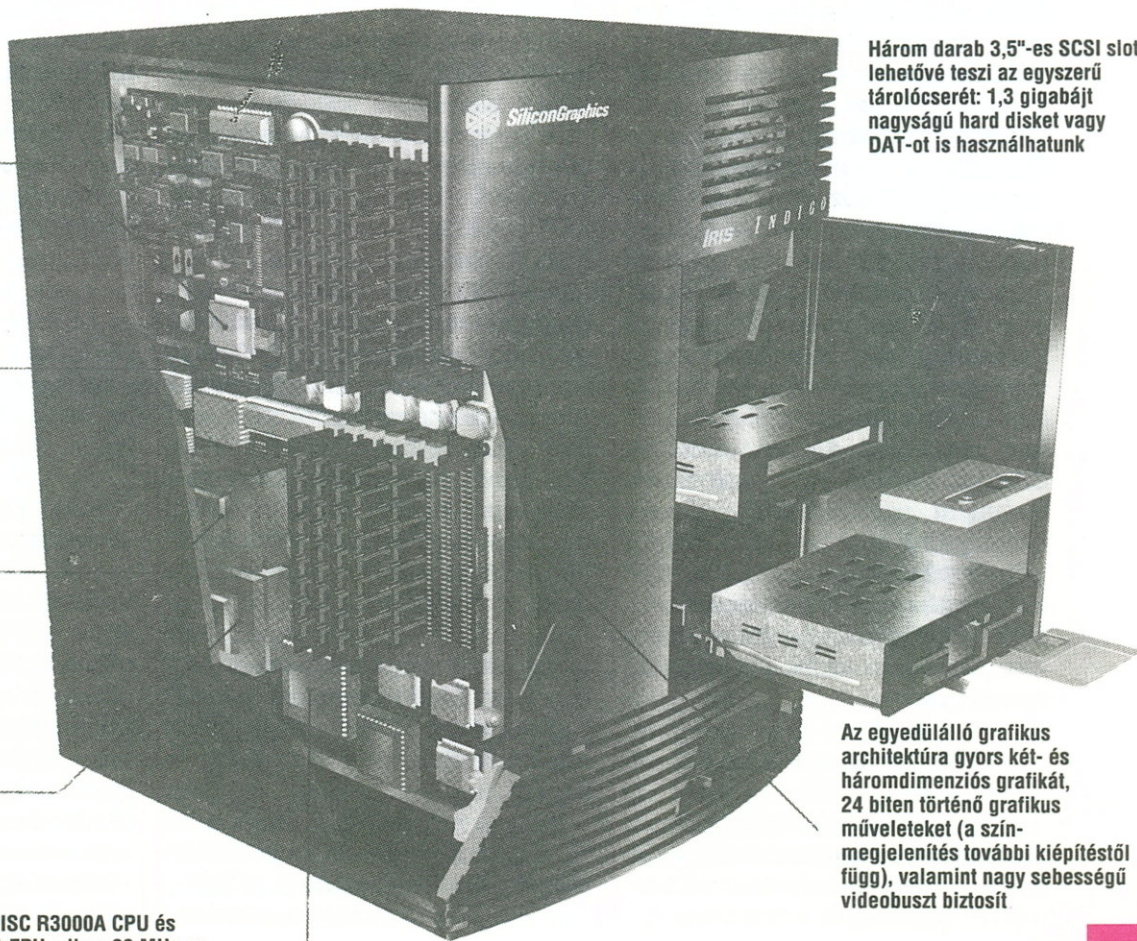
A GIO rendszerbusszal 133 megabájt/s belső adatátvitelt érhetünk el (ez az adat időközben megváltozott, hiszen az újabb Indigókba RISC 4000-es processzort tesznek, amely már 64 bites és 50 MHz sebességű)

Az Iris Indigo kontroller-chipje 2,3 megabájt/s lemezek közötti adatátvitelre képes

MIPS RISC R3000A CPU és R3010A FPU adja a 33 MHz-es sebességet (ez más „mértékegységben” másodpercenként 30 millió műveletet jelent)

Három darab 3,5"-es SCSI slot lehetővé teszi az egyszerű tárolócserét: 1,3 gigabájt nagyságú hard disket vagy DAT-ot is használhatunk

Az egyedülálló grafikus architektúra gyors két- és háromdimenziós grafikát, 24 biten történő grafikus műveleteket (a szín-megjelenítés további kiépítéstől függ), valamint nagy sebességű videobuszt biztosít



MINISZTERIUMI SZOLGÁLTATÁSOK

Adatbázis – megnyírbált információkkal

Az Ipari és Kereskedelmi Minisztériumban egy egész főosztály dolga az információgyűjtés és elemzés, valamint az ezekhez kapcsolódó szolgáltatások nyújtása. Az ipari informatikai főosztály vezetőjét, *Törökné Szenté Ágnes*t felkeresve arra a kérdésre kerestünk választ, vajon kik és hogyan hasznosíthatják az itt kiépített információs bázist? Azt az adathalmazt, amelyet nem is olyan könnyű aktualizálni.

– Az ipar, a belkereskedelem és az építőipar valamennyi statisztikai, mérleg- és külkereskedelmi adata itt van főosztályunkon, mi tagadás, elég nehezen értük el, hogy megkapjuk valamennyit. Először csak az ipar adataira készítettünk egy számítógépes rendszert, még a nyolcvanas évek közepén. Jelenleg száz számítógép üzemel a minisztérium épületében, Novell hálózatba kötve. Sajnos ezek mind PC-s rendszerek, s anyagi források hiányában csak koncepcionális elképzeléseink vannak a továbblépésről. Főosztályunk három osztályból áll: az információs, az elemző és a számítástechnikai osztályokból. Tevékenységünk motorja az információs osztály. Az összes adatgyűjtés, adatforrás-felkutatás ott történik. Eddig elég jó helyzetben voltunk, idén azonban minden megváltozott.

– *Mitől más 1992, mint az előző év volt?*

– A statisztikai törvény, és a személyes adatok védelméről szóló törvény új helyzetet teremtett. Eddig az volt a munkameg-

osztás, hogy a Központi Statisztikai Hivatal különböző tárcákon keresztül gyűjtötte az adatokat. Az iparvállalatoktól – ami ezer nagyvállalatot jelentett – minden hónapban megkaptuk a havi statisztikákat, feldolgoztuk és átadtuk a KSH-nak. Ez január elsejétől megszűnt. A KSH kijelentette, hogy technikailag alkalmas az adatok gyűjtésére és feldolgozására, ezért ezentúl közvetlenül hozzá érkeznek majd be az adatok. Emiatt az ipari adatok például csak a tárgyhónapot követő negyvenedik napon jelennek meg nálunk.

– *Tehát megfordult az ügymenet, s a KSH-n keresztül kerültek ide az adatok?*

– Így igaz, de még nem kerültek ide. Egyrészt van egy tájékoztatói rendszer, amelyben a KSH publikálja, hogy mekkora az ipari termelés, milyenek a kereskedelmi folyamatok, az idegenforgalom, de ezek az adatok globálisak, csak nemzetgazdasági szinten jól használható elemzések, ágazati szempontból azonban használhatatlanok. 1992-ben tehát úgy állunk, hogy tavaly év végéig minden adat megvan korrektül, az idei első negyedéves adatokat viszont még meg sem kaptuk. Mire a KSH az éves iparstatisztikát feldolgozza, augusztus-szeptember lesz. Olyan nagy csúszással tudja átadni a KSH, hogy csak távolról tudjuk követni az eseményeket.

– *Mindennek az oka az új statisztikai törvény, meg a KSH?*

– Ne értsen félre, én minden

fórumon elmondtam, hogy az új statisztikai törvény egy korszerű, szellemében nagyon jó jogszabály. Az Európai Közösséghez való csatlakozásunk feltételét megteremti. De nálunk jelen pillanatban nincs ilyen helyzet, a statisztikai rendszer szétesett, ami idáig jól működött, az az utóbbi időben nehézkessé vált, mert „deregulálni” kellett. Azt mondtuk, nem kell annyi adat, ne terheljük az adatszolgáltatókat. Ez odáig ment, hogy ma már gyakorlatilag alig lehet adatot kérni a vállalatoktól. A teljes körű adatgyűjtés 50 foglalkoztatott feletti létszámú vállalatokra, vállalkozásokra terjed ki, de ezt a KSH is nagyon nehezen tudja elkérni. Legtöbbször összeírónak kell kimenni a céghez és begyűjteni az adatot, mert arra hiába várnak, hogy a kitöltött kérdőív postán megérkezzen. A vállalatok azt csinálnak, amit akarnak. Pedig, ha tudnák, hogy az EK-ban például nagyon szigorúan kell adatokat szolgáltatni, sokszor sokkal mélyebben, részletesebben, mint a mi korábbi hazai gyakorlatunkban.

– *Mit mond a törvény az adatszolgáltatási kötelezettségről?*

– A KSH a minisztériumoktól minden adatot átvesz, de nekünk nem ad vissza, csak aggregált szinten. A statisztikai szolgálatnak mi is tagjai vagyunk, a minisztériumok, a versenyhivatal, legfelsőbb bíróság, az ügyészség. Úgy fog összeállni a statisztika, hogy május 31-ig le kell adnunk, mi mit szeretnénk 1993-ra. Az alap a KSH adatgyűjtése, s párhuzamos adatgyűjtés

nem lehet. Az IKM például nem kérheti be a termelési értéket a vállalatoktól, mert azt a KSH gyűjti. Így mi nem tudunk hozzájutni, pedig a munkánkhoz kellene. Erre azt szokták mondani: átmeneti időszak. Kérdés, hogy meddig tart ez? Amikor összeáll az országos adatgyűjtési kör, akkor a KSH elnöke mellett működő független testület, az Országos Statisztikai Tanács véleményezi. Ezután a KSH a kormány elé viszi és kormányrendelettel hagyatja jóvá. Ez a '93-as évre vonatkozólag októberben készül el, s akkortól kötelező lesz.

– *Mi történik, ha az ipari miniszternek valamilyen kérdésben döntenie kell, s nincsenek hozzá adatai, információi?*

– Rendkívüli esetben a KSH elnöke kérhet adatgyűjtést. Ha elmegyünk hozzá és rávesszük, meggyőzzük, hogy ez fontos, akkor kérhet be adatokat soron kívül is. Erre nem hiszem, hogy fogadókészség lenne. Ráadásul minden adatgyűjtésnek van anyagi vonzata is. A mi minisztériumunknál még nem olyan színvonalú az egész informatikai kultúra, szóval ha faragni kell a költségvetésből, azt mindig tőlünk veszik el. Időnként emlegetem, hogy ha nem lesznek adatok, az egész ház működés-képtelen lesz!

– *Úgy tűnik, itt mindenkinek van információja, csak a minisztériumnak nincs. Van az ÁVÜ-nek, az APEH-nek, a TB-nek, a bankoknak, a NGKM-nek és a KSH-nak.*

– Így van. Ráadásul az információs rendszerek az új törvény szerint nem kapcsolhatók össze. Egy helyen találkozhatnak az adatok, a KSH-ban. A KSH pedig technikailag pillanatnyilag alkalmatlan erre. Lehet, hogy négy év múlva hatalmas számítógépparkkal győzi majd. Jogászok sem tudtak elfogadható választ adni a kérdésemre, hogy a személyes adatok védelme miért nem csak a természetes személyek adatainak védelmét jelenti? Az 56

ezer jogi személy adatai miért tartoznak a törvény hatálya alá? A több mint négyszázezer egyéni és társas vállalkozás adataihoz miért nem lehet hozzáférni? Miért ne lehetne megtudni, mennyi a termelési értéke, ennyi a GDP-je?

– *Akkor a céginformáció-adásra szakosodó cégek honnan szerzik az adatokat?*

– Van egy kibúvó. A vállalatok önkéntesen szolgáltathatnak adatot. Mi is elrendelhetünk önkéntes adatszolgáltatást...

– *Hogyan lehet önkéntességet elrendelni?*

– Úgy, hogy elrendeljük az adatgyűjtést, kiküldjük a kérdőívet, amelyre rá van írva, hogy az adatszolgáltatás önkéntes. Vagy teljesíti, vagy nem. Miután saját vállalatunk sem kényeztetnek el adatokkal, ilyenfajta adatgyűjtéssel még nem próbálkoztunk. Mi a KSH-tól átvett adatokat megpróbáljuk karban tartani, és ennyiben a mienk naprakész. Tájékoztatási Ügynökségünk ezekből az adatokból kiadványokat állít össze, amelyek önköltségi áron megvásárolhatók. 1992-ben egyébként a tevékenységek egységes osztályozása is megváltozott, az EK rendszeréhez igazítottuk a magunkét.

– *Vannak más kiadványaik is?*

– Igen. Készítünk műszaki-gazdasági adattárat, melyben összefoglaló áttekintést adunk a termelés, az értékesítés, a beruházások, a technológiai színvonal, a fejlesztések, a gazdálkodók számának és létszámának alakulásáról. Egy másik, magyarul, angolul és németül megjelenő ismertetőnk a minisztériumunk tevékenységi köréhez tartozó területeket mutatja be, kiemelt közgazdasági mutatók segítségével. Ezek az adatok jól használhatók piackutatáshoz, partnerkereséshez, a privatizációval kapcsolatos érdeklődésekhez. Számítógépes adatállományunkból – térítés ellenében – lehívhatók a gazdálkodó szervezetek statisztikai számjelei, cégnevük és cí-

mük. Plusz szolgáltatásként ezeket etikettcímekére is nyomtatjuk igény esetén. Névjegyzék-adatállományunk a cégekről, részletesebb adatokat is tartalmaz: postacím, telefon, telex, telefax, vezetők neve és telefonszáma. A Ki mit gyárt, forgalmaz, exportál, importál? adatállományunk az ipari gazdálkodók termékeit tartalmazza, továbbá a kereskedelmi cégek által forgalmazott árufélék listáját. Online kapcsolatunk van külföldi adatbankokkal, ahonnan a nemzetközi üzleti élet információi hívhatók el, forintfizetés ellenében. Operatív adattárunkból lehetőség van szakágazati, területi elemzések, prognózisok készítésére.

– *Összegzésként elmondhatjuk, hogy kialakult egy olyan helyzet, melyben az Ipari és Kereskedelmi Minisztérium ipari informatikai főosztálya csak nagy nehézségekkel és csak megközelítően teljes információs kötetet tud produkálni. Az iparvállalatoktól ezt az információcsokrot más simán képes kiadni?*

– Nem, más sem tudja. Legalábbis én ezzel vigasztalom magam. Kérdés, hogy mit jelent a megbízhatóság és a teljeskörűség? Mindenesetre a kormányzati információs struktúra kiépítése elindult. A fejlesztésben benne van az összes kormány szerv. Mindig elakadunk a tartalmi oldalnál, illetve állandóan a technikai oldal felé kalandoznak el többen. Pedig először minden kormány szerv képviselőjének le kellene ülni egy asztalhoz és elmondani: nála milyen adatok vannak és abból mit hajlandó átadni a többieknek? Idáig mindig azon bukkott el a dolog, hogy az átadásra szánt adatokért milliókat kértek a felkínálók. Eből persze mi is tanultunk, mi sem adunk minden adatot, s többnyire amit kiadunk, azért is pénzt kérünk. Azért bízom benne, hogy a tárcák közötti kapcsolat normalizálódik és kialakulhat az együttműködés, jó adatcsere.

Bárány Tibor

ON-LINE ÉS CD-ROM FEJLESZTÉSEK

Új technológiák békés harca

Az Országos Műszaki Információs Központ és Könyvtár (OMIKK) közel egy évszázada szolgálja a szakértelmiségieket. Korán kezdték a hagyományos és az on-line irodalomkutatást és -fejlesztést, amely a magyar műszaki és tudományos életben úttörő szerepet töltött be. A legismertebb talán az az on-line jellegű nemzetközi szakirodalmi kutatás, amely a világ legnagyobb adatbázisainak legfrissebb információit kínálja. Természetes ellentmondás, hogy előfizetői díja magas, éppen ezért egyelőre nem terjedhet el széles körben.

A gazdasági kényszer

„Az utóbbi években sajnos nálunk is lecsapódott a cégek nehezé anyagi helyzete – mondja Horváth Péter főigazgató – amikor a nadrágszíjat megszorítják Magyarországon, gyakran éppen a szakinformációról mondanak le.”

Az OMIKK a nagykönyvtárhoz hasonlóan az állami költségvetésből dotált intézmény, azonban másoknál korábban kényszerült piaci jellegű vállalkozásokra az információszolgáltatásban, már csak azért is, mert sohasem kaptak támogatást az on-line szolgáltatások felhasználásához, amelyek a hazai bérshinthez képest drágák. Az egyetemi kutatóhelyek, kezdő tehetőségek számára például aligha elérhetőek.

Kapcsolódásunk Európához

Idén tavasszal az OMIKK európai szemináriumot rendezett, ahol az Európai Közösség fő információs szolgáltatásaival is-

merkedtek a résztvevők. Az EK-központ munkatársai erősen hangsúlyozták, hogy ők elsősorban nem pénzt, hanem információcserét várnak a magyar partnerektől, bár tudják, hogy a nagy hazai adatbázisok kiépítése évekkal elmaradt a nemzetközi élvonaltól. „Ahhoz, hogy behozzuk az információt, információt kell adnunk” – fogalmazta meg Horváth Péter az elkövetkezendő évek fő feladatát, s ennek jegyében angol és magyar nyelven, számítógépes és nyomtatott formában ismertetik meg a külföldi partnereket a magyar műszaki és tudományos eredményekkel. A nemzetközi piaci verseny is erre készíti az OMIKK-ot, hiszen a szakinformáció, például a könyvek, ára sokszorosan megnövekedett, így minden hazai nagykönyvtárnak relative egyre kevesebb lehetősége van beszerzésre.

Osztott katalógus

Az OMIKK a nyolcvanas évek elejétől számítógépre viszi katalógusait, s különösen biztató az OSCAR elnevezésű osztott katalógusrendszer, amelyben fokozatosan számítógépre viszik valamennyi hazai műszaki nagykönyvtár katalógusát. Olyan számítógépes programot adaptáltak a hazai viszonyokra, amely bármely résztvevő számára, bármely hazai nagykönyvtárban biztosítja minden releváns könyv azonnali adatszolgáltatását on-line rendszeren.

On-line versus CD-ROM

Az on-line szolgáltatás magas ára idáig sok érdeklődőt kizárt.

Az 1992-ben beindult CD-ROM fejlesztési program azonban bárki számára, nagyjából ingyenesen teszi elérhetővé a külföldi információt. „Tulajdonképpen a CD-ROM sem olcsó” – mondja Válas György főmunkatárs, a szakirodalmi keresés egyik első magyar szakembere –, „csupán arról van szó, hogy itt az egyszeri nagyberuházást a könyvtár, akár folyóiratvásárlási keretből is megfizeti, s így az olvasó számára nem jelentkezik a nagyobb költség.”

Ha egy átlagos olvasó bejön az OMIKK-ban, akkor egy óráig ingyen kereshet a CD-ROM adatbázisokból, sőt bizonyos segítséget is ingyenesen kap. Aki szakszerűen keres, ennyi idő alatt igen sok „találatot” érhet el. Csak a sornyomatatóval elkészített anyagért, vagy a floppy-lemezért fizet. Ugyanarról a terminálról lehetséges a kombinált on-line és CD-ROM keresése is, ami sokszorosan csökkentheti az on-line szolgáltatás árát.

A legújabb és a legrégebb

Általában nem jelent gondot az, hogy a CD-ROM-ok legfrissebb információkkal való feltöltése csak negyedévenként történik meg. Ha valakinek mégis a super-friss információra van szüksége (ami főleg az üzleti életben, a politikában és egyes fejlesztési munkáknál fontos), akkor lényegesen többet fizet, hiszen külön kell foglalkozni vele. A nagyon régi információkat még nem vették CD-ROM-ra, ha valakinek évekkal ezelőtti adata van szüksége, a kedvéért még a modern elektronika korában is előszedik a régi könyveket. Egyáltalán

nem meglepő, hogy éppen a tálmányok, szabadalmak esetén kerül erre leggyakrabban sor, hiszen annyi mindent feltaláltak már évtizedekkel ezelőtt is.

Megszállott gyerekek és csúcstechnológusok

Tíz nagy CD-ROM adatbázis teljesen szabadon és ingyenesen használhat bármely olvasó, és további tizenegyet kérhet külön. Kik jöttek az első időszakban? – kérdeztem Bayer Juditot, a CD-ROM szolgáltatás vezetőjét. „Kétféle tipikus olvasónk van – válaszolja – az egyik csoport a gyerekekből és serdülőkorúakból áll, akik boldogan vették birtokba a CD-ROM keresőket. Ők nagyon kíváncsiak, nagyon sokat tudnak, nagyon sokat akarnak tudni és ezzel – szerencsére – sok munkát adnak nekünk. A másik csoportra az jellemző, hogy minderről sokkal többet tudnak nálunk; ők a számítástechnika élvonalbeli szakemberei; leggyakrabban a fejlesztők.”

Beszélgető kapcsolatok

„Egyre többen keresik a CD-ROM szolgáltatásokat nálunk is és más könyvtárakban is – mondja Bayer Judit –, s ezért minél előbb szükség lenne egy olyan központi CD-ROM adatbázis-nyilvántartásra, amilyen a könyvek központi nyilvántartója, nehogy lemaradjunk a sok helyen történő beszerzésekről, vagy pedig pénzt pocskékoljunk párhuzamos beszerzésekkel; ne ismételjük meg, ami az egyre drágább szakkönyvekkel is történt. A CD-ROM első beszerzése nem olcsó mulatság. Jó lenne tudni, hogy egyáltalán hol, mi történt az elmúlt években...”

Technikák a mérlegen

Válas György éppen ezen dolgozik: a legfrissebb információt gyűjti arról, hogy milyen nagy adatbázisok léteznek hazánkban; honnan melyek érhe-

tőek el, s reméli, hogy közeledünk az országos összekapcsolt hálózatokhoz.

Mégis, a mai körülmények közepette ő is nagy híve a CD-ROM-nak, mivel az on-line keresés még akkor is jelentős költséget tétel fel, ha a külföldi partner esetleg kegyesen ingyen kínálná az információt. Ilyen eset persze csak ritkán fordul elő: utoljára egy éve kínáltak ilyen ajándékot a franciák. Ezzel szemben a CD-ROM, ha már valahol behozták az országba, akkor igen olcsón használható; ráadásul a felhasználó lemezre másolhatja az őt érdeklő részleteket.

Kérdés az, hogy lévén Magyarország kis ország, a nyelv elszigetelt, mit jelent a „kis példányszámú hanglezem” és hogyan léphetnének a szervezés terén, hogy minél több intézmény és felhasználó között oszthassák meg a hazai CD-ROM lemezek kibocsátásának a költségeit.

Szekeres Péter

FLOPPY-POSTA

Utánvétellel megrendelem a Mikrovilág 100 gramm program című kiadványának programlemezét példányban.
Ára: 200 Ft + postaköltség.

Cím: Mikrovilág Programfüzet
1536 Budapest, Pf. 386

Név:

Cím:

Géptípus: C-64

Plus/4

„Fontos” és forintos árak

Ez alkalommal körülnézünk az angol piacon, lehetőség adva az olvasónak arra, hogy összevesse az ottani árakat az itthoniakkal. Segítségül: 1 font jelenleg valamivel több mint 143 forintot ér. A táblázatban az árakat fontban adtuk meg, utána következik a budapesti Atari bolt kínálata – forintban, majd összehasonlításul fontban (ahol nem szerepel csak az angol ár, az a termék idehaza nem kapható). Érdeemes odafigyelni a magyar árakra, és ne felejtsek el, hogy az angliai árakat még nem terheli a vámilleték és a forgalmi adó!

Konfigurációk

- 520 ST Discovery Xtra Pack (512K STE négy játékprogrammal és tanítóprogrammal) – £299.99
- 520 Turbo Pack (512K STE joystick-kal és játékprogrammal) – £359.99 (33 445 Ft = £233,88)
- 1040 Family Curriculum Pack (1MB STE öt felhasználói szoftverrel) – £399.99 (52 480 Ft = £366.99)
- Mega STE 1 plus SM124 (1MB STE 16/8 MHz kapcsolható CPU, külső csatlakozású billentyűzet, mono monitor.) – £816.62
- Mega STE 2 plus SM124 (2MB STE 16/8 MHz kapcsolható CPU, külső csatlakozású billentyűzet, mono monitor) – £1169.12 (183 990 Ft = £1286.64)
- TT030/2 (68030 processzor, 32 MHz, 2 MB RAM, 48 MB merevlemez, nagy felbontású színes monitor) – £2344.12 (296 980 Ft = £2076.78)
- TT030/4 (68030 processzor, 32 MHz, 4 MB RAM, 48 MB merevlemez, nagy felbontású színes monitor) – £2579.12 (336 980 Ft = £2356.50)
- TT030/8 (68030 processzor, 32 MHz, 8 MB RAM, 48 MB merevlemez, nagy felbontású színes monitor) – £2814.12 (409 980 Ft = £2866.99)
- Stacy 2/20 (Hordozható ST 2 MB RAM memória, LCD képernyő, 20 MB merevlemez, trackball) – £1644.99 (234 990 Ft = £1643.28)
- Stacy 2/40 (Hordozható ST 2 MB RAM memória, LCD képernyővel, 40 MB-os merevlemezzel) – £1762.49
- Stacy 4/40 (Hordozható ST 4 MB RAM memória, LCD képernyővel, 40 MB-os merevlemezzel) – £2114.99 (339 990 Ft = £2377.55)
- Atari egér (mouse) – £24.99 (2490 Ft = £17,41)
- SC1224 (Alacsony és közepes felbontású színes monitor) – £306.51 (29 990 Ft = £209.72)
- SM124 (Nagy felbontású 12"-es papírféher monokróm monitor) – £149.99
- SM194 (Nagy felbontású 19"-es (A3-as) nagyságú monokróm monitor) – £1879.95 (171 990 Ft = £1202.72)
- SLM605 Laser printer (300 dpi, 6 lap/perc. Minimum 2MB-os ST-vel használható, mert 1 MB-ot



Atari Mega STE



Atari Stacy

lefoglal a rendszer-RAM-ból) – £1057.49 (134 990 Ft = £943.98)

Egyre több perifériát SCSI interfésszel látnak el, ezért szükségessé vált, hogy az Atarikhoz is elkészítsék ezt az illesztési felületet. A legújabb Atari típusokba (például a TT sorozat tagjaiba) ezt az interfészt már gyárilag beszerelték, de több gyártó cég készít olyan interfészt a régebbi típusú Atarikhoz, amely a számítógép DMA portjára csatlakozik és szabványos SCSI kimenete van. Ezzel lehetőség nyílt nem Atari típusú (Megafile 44), cserélhető 44 vagy 88 megabájtos merevlemez, SCSI csatlakozású nyomtató, lézerlevilágító és számtalan más egység csatlakoztatására a géphez.

SCSI adapterek

BBDP (BBDP Electronics) – £49
 GE-Soft IV (Gasteiner Technologies; 8 periféria csatlakoztatása 256 partíciót kezel a merevlemezben) – £69
 ICD Ad SCSI Plus és Micro (Silica System) – £99.95
 Supra SCSI interface (Frontier Software) – £100.00

Külső floppy-meghajtók

Power Drive 3.5" (Power Computing) – £79.95
 CSA354 ST (Cumana) – £78.14
 Datadisk drive (Pandaal Marketing) – £79.95
 Multidrive (Power Computing; 3.5" és 5.25"-es meghajtó egy egységben) – £199.00
 PC720e Power Drive (Power Computing; 3.5"-es meghajtó, a tápfeszültséget az ST joystick portjáról kapja) – £59.95
 Zydec 1 MB drive (Evesham Micros) – £59.95



Atari TT

CSA1000S (Cumana; 5.25"-es 40/80 sáv kapcsolható külső meghajtó) – £116.32

Gasteiner external drive (Gasteiner; külső meghajtó külső tápegységgel) – £54.95

Kovács P. Attila

Atari 1040 STE



Jó a szöveg?

Az amerikai statisztikák szerint a felhasználók több mint fele szövegszerkesztésre (is) használja a számítógépet. Hasonló az arány nálunk is, hiszen a szövegszerkesztő, és a táblázatkezelő szoftverek a legelterjedtebbek. Csakhogy amíg az angol nyelvterületeken az alapvető gépismereten túl a programok használatához elég az anyanyelv ismerete, addig nálunk egy középfokú nyelvvizsga után is némi bátorság kell ahhoz, hogy egyedül kezdjen megtanulni „szövegszerkesztőül” is az alkalmazó.

Ezért is történhetett, hogy a magyar nyelvű szövegszerkesztők között a közelmúltig egyeduralmú ÉkSzer határozottan népszerűvé vált a felhasználók körében. Az MT-Computer forgalmazásában megjelent azonban a világ eladási listáját vezető WordPerfect 5.1-es verziójának magyar változata, s ezzel egy csapásra megoszlott a piac is. Sokan ugyanis hajlandók áldozni a magyarul beszélő WordPerfectre, mások viszont arra esküsznek, hogy legjobb az a szövegszerkesztő – amit már ismernek.

Mielőtt a két szoftver legtöbbet használt funkcióinak összeméréséhez kezdenénk, két dolgot le kell szögezni:

- az összehasonlítás csak az alapvető szolgáltatásokra terjedhet ki (tekintettel arra, hogy mindkét szövegszerkesztő program kézikönyve több mint száz oldalban foglalja össze a részletes tudnivalókat),
- a szoftverek kifejlesztésének feltételei közel sem voltak azonosak.

A WordPerfect ugyanis a világ egyik legfejlettebb fejlesztési hátterét biztosíthatja, az ÉkSoft

Számítástechnikai és Kereskedelmi Kft. fejlesztőinek pedig szinte nulláról kellett felépítenie néhány évvel ezelőtt egy komplett szövegszerkesztő-algoritmust – bár a rossz nyelvek szerint ez sokban hasonlít egy korábban tiszavirág-életű angol nyelvű szoftverhez.

INSTALLÁLÁS

Mind a két szoftvercsomagot igen egyszerűen tárolhatjuk már egy IBM PC/XT kategóriájú gépen is. Rendszerfeltételeik annyiban hasonlítanak, hogy mindkét szerkesztő bármely „mezei” XT-n is tökéletesen futtatható, az viszont jelentős különbség, hogy az ÉkSzer csak a hivatalosan megvásárolt egységcsomagban található kulcslemez segítségével indítható, míg a WordPerfect a winchesterből közvetlenül is (általában wp beírásával) startol.

Az alkalmazás kezdeti lépéseinél komoly segítséget adnak azok a matricák, amelyeket mindkét gyártó mellékel a szoftvercsomaghoz. Ezeket a klaviatúra fölé (vagy a billentyűkre) ragaszthatjuk, ezzel a betanulás során felére csökkenthetjük a funkciók memorizálásának (esetleg könyvből való újrakeresésének) idejét.

Fontos azonosság, hogy mindkét szoftver használható hálózatra kötött gépeken is.

A MENÜ

Akár tetszik, akár nem: a szövegszerkesztő szoftverek mindegyikét legtöbbször egyszerű levelek, dokumentumok egyszeri megírására, és lemezen (vagy winchesteren) való tárolására használjuk. Így van ez akkor is, ha közben minden felhasználó

sejti, hogy a szoftvernek csak a legkézenfekvőbb szolgáltatásait alkalmazza ilyenkor, s éppen a hosszabb ismerkedés utáni finomságok jelentenek a valódi használati értéket.

Nézzük, mit kínál az ÉkSzer:

A monitor felső sorában a menü elemei láthatók, alul pedig egy információs ablak van, amelyben a három, egy időben készíthető dokumentumcím, és a nyomtató fajtája látható.

Mi tagadás, munka közben a WordPerfect sokkal kevésbé zavaró, és rögtön működőképes, tiszta képernyőt kínál.

Egy időben két dokumentumon dolgozhatunk (ezt „Szöv1” illetve „Szöv2” jelöli a jobb alsó sarokban, ahol még a lapszám és a kurzorpozíció található), itt a funkciógombok leütésével akkor juthatunk hasonló választási helyzetbe, amikor már valamit kezdeni akarunk az elkészült anyaggal. (Bár kívánságra az **Alt** és az = egyidejű leütésével úgynevezett „redőnyemenüt” varázsolhatunk a képernyő tetejére.)

Ha már az egyéni ízlésnél tartunk: mindkét szoftverben – igaz alapos tanulmányozás után – személyre szólóan is kialakíthatjuk a szokásainknak legjobban megfelelő billentyűkiosztást.

HELP!

A tanulás időszakában minden szoftver leggyakrabban használt szolgáltatása a Help: nos itt egál van, mindkét vizsgált szoftver klasszikusan segít: az ÉkSzer az aktuális menüben az F1 után kezdi magyarázni, mit és hogyan, a WordPerfecttől pedig az F3 után kaphatunk segítséget a kívánt funkció első betűjének leütése után. Mindez – elvileg –

lehetővé teszi, hogy a kézikönyvek salátává olvasása mellett a lagapróbb szolgáltatásokat is teljesen egyedül sajátítsuk el, csak hogy ez kísértetiesen hasonlít Karinthy egyik klasszikus dilemmájához, miszerint: lehet-e igaz barátság férfi és nő között? Ha igen, miért nem? Ami szövegszerkesztőkre értelmezve így szól: megtanulható-e a használat egyedül, ha igen, miért nem? És mindez annyira valós, hogy létezik egy amerikai telefonszám, ahol egy igazi WordPerfect-guru várja az alkalmazók kérdéseit, csörög is a telefon – ugyanúgy, mint az Ékszer „forró drótján”, ahol szintén minden segítséget megadnak a fejlesztők. No de vissza a szolgáltatásokhoz:

FORMÁTUM

A legáltalánosabban használt „intelligens írógép” funkció legfontosabbja, hogy mennyire tudjuk szépre varázsolni a majdan kinyomtatandó szöveget. Itt pedig a hasonlóság mellett van lényeges különbség is: nincs az a megjelenítési forma, amire ne lenne képes mind a két szoftver. Csakhogy amíg az ÉkSzer WYSIWYG (azaz: amit látsz, azt kapod nyomtatásban is) rendszerű, addig az elkészült anyagot WP-ben egy komináció után tekinthetjük meg (**Shift+F7**, utána **6**) ráadásul itt már a csatlakoztatott nyomtatókról is beszélnünk kell. Ugyanis mindkét szoftver ismerői azt tanácsolják, hogy a dokumentumot még a nyomtatás előtt mentsük ki, merthogy a printer-meghajtók inkompatibilitása esetén kiszámíthatatlan helyzetbe kerülhet az alkalmazó. Nemritkán „melegindítás” válik szükségessé, s olyankor valóban nem árt, ha a lemezen (winchesteren) már tároltuk a korábbi munkát.

Mindkét szövegszerkesztőn egyformán egyszerű a munkahelyi „szabványnak” megfelelő levél-, nyomtatványformátum rögzítése, ezért mindkettő segítségével gyorsan készíthetünk azonos tartalmú, de változó címzésű körleveleket is – ilyenkor a cím-

zetteket külön adatbázisból választhatjuk. Ugyanígy alkalmassak a borítékra ragasztható etikettcímkek kinyomtatására is, ami nagyban egyszerűsíti a hivatali adminisztrációt.

HELYESÍRÁS-ELLENŐRZÉS

Korszerű szövegszerkesztők-nél elengedhetetlen szolgáltatás, hogy kérésre a számítógép nyelvtani szempontok alapján is ellenőrizze a szöveget, sőt „beépített lektorként” helyesbítési javaslatot is tegyen a felfedett hibákra. Ismét egál, hiszen az ÉkSzer korábban a NyelvÉsz-t, az 5.0 verziótól kezdve pedig a Lektor elnevezésű helyesírás-ellenőrző segédprogramot kínálja fel, a WordPerfect pedig az idei ifabón debütált „Helyes-e?” ellenőrzőszoftvert használja – bár ez az első körben nyomtatott kézikönyvben még nem is szerepelt.

TÁBLÁZATKEZELÉS VAGY -KÉSZÍTÉS?

A klasszikus értelemben vett táblázatkezelésre egyik szoftver sem alkalmas – alaphelyzetben. Az ÉkSzer azonban képes a dBase-ben készült táblázatok közvetlen megjelenítésére, amit a WP-vel is megvalósíthatunk a megfelelő segédprogramok betöltésével. A táblázatformátum kirajzolására és a zsebszámológép funkcióira azonban mindkét program használható. Csakhogy a megjelenő táblázat sorainak és oszlopainak összegzésére ilyenkor nem árt egy valódi kalkulátort is a komputer mellé készíteni.

GRAFIKUS LEHETŐSÉGEK

Nos itt az első lényeges eltérés. Bár az ÉkSzer is képes bizonyos grafikai beolvasásokra és azok megjelenítésére, sőt, hasábtördelésre is alkalmas – igényesebb kiadványoldalak elkészítése igazán csak a WordPerfecttel lehetséges. Az igazi különbség persze abból adódik, hogy a nyomdai előkészítéssel foglalkozó cégek sokkal inkább elfogad-

ják a WP-vel készült anyagokat, mint az ÉkSzerrel szerkesztetteket. Ráadásul a WordPerfect egy olyan piktogram-készletet is felkínál, amely néhány órás gyakorlat eredményeként képessé teszi a felhasználót arra, hogy (például a szórólapokról ismert formában) „majdnem nyomdai” jellegű szabványoldalt készítsen.

DOKUMENTÁCIÓ

Sajnos kevéske. Az ÉkSzer keményfedeles borítóján ott a (joggal viselt) Kiváló Áruk Fóruma embléma, de érezni, hogy az összeállítás decimális bontásakor nem az ismerkedő logikáját, hanem a befutott szoftverek receptjét követték. A WordPerfect magyar nyelvű kézikönyve pedig olyan vázlat szerűre sikeredett, hogy nincs igazi alkalmazó, aki az ötször vastagabb angol nyelvű „kiegészítés” nélkül az utolsó elágazásokig képes lenne leásni magát – nem is szólva a magyar helyesírási hibák kilobájtjairól.

ÖSSZEFOGLALVA

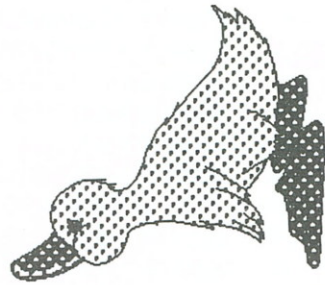
Vitán felüli, hogy a WordPerfect egy sokkal fejlettebb, a felhasználó minden igényét szinte hiánytalanul kiszolgáló szoftver – de dokumentációját tekintve (egyelőre) sokkal kevésbé „magyar” szövegszerkesztő, mint az ÉkSzer. Más fogalmazásban: a WordPerfect jobb, de kevésbé érthető a csak magyarul értő felhasználónak (túlságosan gyakran fordul elő a „keresse az angol kézikönyvben” utalás), mint az ÉkSzer, amelynek mindennapos alkalmazása nehézkesebb, de használatát sokkal hamarabb elsajátíthatjuk.

Igen jellemző, amit a WP kézikönyvéből másolok ide: „A WordPerfect rendkívül megbocsátó program...”. Az ÉkSzer is: minden szövegszerkesztési hibánk korigálható, és ezért mind a kettőbe könnyű beleszeretni.

A baj csak az, hogy önmagától egyelőre egyik sem képes még a legegyszerűbb levelet sem megírni. Várjuk az újabb verziókat.

Guttray László

WordPerfect AKCIÓ!



DrawPerfect 1.1
PlanPerfect 5.1

18.000,-
18.000,-

DrawPerfect 1.1 + WP 5.1 for Windows
PlanPerfect 1.1 + WP 5.1 for Windows

39.999,-
39.999,-

Amíg a készlet tart!
Vásároljon viszonteladóinknál!

Hivatalos disztribútor : MT-Computer Rt. H-1075 Budapest, Király u. 1/d. Tel./Fax: 122-1623/122-5099

A számítástechnika története 7.

A SZILÍCIUM VÖLGY

Egyszer volt, hol nem volt... a San Francisco-öböl déli végénél, úgy nyolcvan kilométerre a város központjától, egy termékeny síkság dombok közé zárva, gyümölcsös ültetvényekkel. Ez volt a helyzet az ötvenes évek elején.

Volt egy egyetem is a környéken: a Stanford University, 1891-ben alapították és már a század eleje óta hatással volt a környék ipari életére. Részben az egyetem támogatta *Lee de Forest* kutatásait is, amelyek eredményeként 1912-ben megalkotják a vákuumcsöves erősítőt. Itt, Palo Altóban hozta létre két volt stanfordi egyetemi hallgató, *W. R. Hewlett* és *D. Packard* a róluk elnevezett céget, amelyben 1950-ben mindössze 200 ember dolgozott. Audio- és mérőberendezéseket gyártottak, illetve fejlesztettek.

Hewlett és Packard egyetemi támogatója *Frederick Terman* rádiómérnök egyetemi tanár volt, aki 1951-ben kigondolta az egyetem környezetében létrehozandó Stanford Kutató Parkot. Ez volt az első tudományos park az Egyesült Államokban, amelyet sok száz követett azóta szerte a világon.

A Szilícium Völgy azóta híressé vált története (maga az elnevezés csak 1971-ből származik) akkor veszi kezdetét, amikor *William Shockley*, a tranzisztor egyik felfedezője 1955-ben visszaköltözik Palo Altóba és megalapítja a Shockley Semiconductor Laboratories-t, saját vállalatát, amely az első félvezetőcég ezen a környéken. Shockley cége gyorsan fejlődik, és 1957-ben már az ő általa összegyűjtött ragyogó tehetségű villamosmérnökök közül hagyják ott néhányan a céget, például *R. Noyce* és *G.*

Moore. Ismert nevek, az utóbbi alapítja az Intelt. Az Egyesült Államok nyugati partvidéke ez idő tájt válik az ország gazdasági súlypontjává. A Szilícium Völgy a fejlett technikájú társadalom jelképe lett.

1954–55-ben már világossá válik, hogy a Bell Laboratórium igen előrehaladt a félvezető és ezen belül is szilícium kutatásaiban. Elkészülnek az első szilícium napelemek és rövidesen megépítik az első tranzisztoros számítógépeket is. Ezeket már a számítógépek második generációjának tekinthetjük – legalábbis a hardvert tekintve. De ekkor készül el az első tranzisztoros rádió, az első elektronikus szintetizátor és a híres angol elektronikai cég, az EMI pedig megépíti az elektronikus képletapogatót.

1954-ben azonban más szempontból is generációváltásra került sor a számítástechnikában. A számítógépek első nemzedékét még csakis gépi kódban lehetett programozni. Azonban, ami a gép számára természetes nyelv volt, azt meg kellett tanulni a programozónak is, a felhasználókról nem is beszélve. Így 1954–55-ben *J. W. Backus* és munkatársai az IBM-nél kidolgozták a FORTRAN (formula translator – mondjuk így, képletfordító) számítógépes nyelvet.

Mi volt ebben az előrelépés? A FORTRAN néhány egyszerű angol szó és különböző matematikai jelek variációjából tevődik össze, egyszerűen elsajátítható a legtöbb ember számára.

Természetesen a számítógépnek is meg kellett tanulnia FORTRAN-ul. Ez úgy történt – és azóta is úgy történik –, hogy megírják azt a programot, amely a

nyelv utasításait értelmezi vagy lefordítja gépi kódra, a gép anyanyelvére.

A fordító vagy az értelmező program betöltése után a FORTRAN-t ismerő felhasználó szabadon tud „társalogni” számítógépével.

Mint említettük, a FORTRAN-t a tudományos kutatóknak írták. Azonban hamarosan megszületett több más nyelv is, így az ALGOL (ALGOritmic Language – algoritmikus nyelv), amely a műszaki tudományos alkalmazásokra irányul, és alfanumerikus szimbólumokat, illetve ma is ismert programszavakat használ (GOTO, READ, PRINT stb.).

1957–58-ban a nagy űrhajózási események közben viszonylag kisebb figyelmet kaptak más eredmények. 1958-ban megépítették az első lézert, az indukált emisszióval működő fényerősítőt, amelynek segítségével nagy energiájú és koherens fényimpulzusokat lehetett előállítani. Ennek a fizikai felfedezésnek számos következménye származott, most csak a számítástechnika területén leglátványosabbakat említsük meg: a lézeryomtatókat és a CD-ROM-ot, az optikai lemezt. Ez év októberében *Jack S. Kilby*, a Texas Instruments munkatársa elkészíti az első integrált áramkört: germániumlapkára ún. mezatranzisztorokat, ellenállásokat és kondenzátorokat helyez el. Elgondolása az volt, hogy minden elemet félvezetőből kell készíteni ugyanazon a hordozón előállítva. Ebben igaza volt, eljárása mégsem lett győztes, mert alig valamivel később *Robert Noyce* egy másik eljárást dolgozott ki, az ún. pla-

náris diffúzió technikáját, amelyvel sokkal könnyebbé vált a különböző – félvezető, vezető és szigetelő elemek előállítása és összekötése a lapkán, azaz a chipen. Ezzel az ötlettel válik sorozatgyártásra alkalmassá az új technológia, és lényegében a mai napig ez a tömeges IC-gyártás alapelve. Két év múlva a Bell Laboratórium kidolgozza az epitaxiát, azt a módszert, amellyel lassú hűtés mellett úgy lehet réteget növeszteni egy kristályfelületre folyadék vagy gázfázisból, hogy közben az anyag fizikai tulajdonságai széles határok között változtathatók. Most ez is az integráltáramkör-gyártás alaptermékéhez tartozik.

Miközben a fizikai kutatásban és a technológiában évtizedeket meghatározó eredmények érlelődtek, még egy folyamatot érdemes megemlítenünk. A számítógépeket eddig – némi ipari támogatással – elsősorban egyetemi és kutatóhelyek építették a kormányzat tevékeny részvételével. Az ötvenes évek első felében toldott át a súlypont az ipari gyártásra. Már említettük, hogy *Eckert* és *Mauchly* saját vállalatot alapítottak, amelyet azonban

1950-ben felvásárolt a Remington Rand cég. Itt fejezték be a UNIVAC-ot, amely már ismert típusjelzés az idősebbek számára. 1955-ig mintegy 20 darab körül szállítottak különböző helyekre. Hosszas habozás után az IBM is munkába kezdett. Az 1948-ban épített SSEC után –, amely még 23000 jelfogót és 13000 elektroncsövet tartalmazó hibrid óriásgép volt, 1953-ra elkészült a híressé vált 701-es modell, egy tudományos célú, elektronikus számítógép. Ebből 19 rendszert építettek meg.

Goldstine könyvében olvashatjuk, hogy a nagy világcég felfogásának megváltozásában kulcsszerepe ifjabb *Thomas Watsonnak* volt, aki tisztában volt az elektronika jelentőségével. Ő hívta az IBM-hez *C. C. Hurd* matematikust, aki viszont *Neumann*t szerezte meg tanácsadónak 1951-től. Így a 701-esben, majd a már köznapi munka végzésére készített 702-ben már nyilván az ő hatásuk is érződött. A 701-est a 704-es modell követte ferritmemóriával, míg a 702-est a 705-ös, amelyet százas szériában gyártottak. A 704-es már értett FORTRAN-ul, bár az egész rendszer

csak 1957-ben jelent meg. Hasonló sikert ért el a 650 típusjelű mágnesdobos rendszer. Az évtized végére 1000 fölé emelkedett az eladott készülékek száma. Ez már olyan programozói és alkalmazói közösséget jelent, amely jelzi egy sok százezer, napjainkra sokmillió embert foglalkoztató szakma kialakulását.

Az ötvenes években már a világ sok államában építettek elektronikus számítógépeket. 1955-ben elkészül a Szovjetunióban az első URAL nevű gép. A japán fejlesztők 1957-ben építik meg az első tranzistoros számítógépet, tehát nem sok idővel követik a Bell Laboratórium eredményeit. Hamarosan feltűnnek azok a vállalatnevek is, amelyek később az Egyesült Államokon kívüli számítástechnikát jelzik, így a francia Bull, amelyik 1956-ban hoz piacra egy kisméretű elektronikus számítógépet. Magyarországon 1959 januárjában jelentette az Esti Hírlap, hogy hároméves munka után az MTA Kibernetikai Kutatócsoportja szovjet tervek alapján megépítette az első elektronikus számítógépet.

Horváth Péter

Utánvétellel megrendelem a **Mikrovilág Magazin** programmellékletében

közölt programokat példányban lemezen.

Ára: 200 Ft + postaköltség.

Cím: Mikrovilág Programfüzet
1536 Budapest, Pf. 386

Név:

Cím:

Géptípus: C-64

Plus/4

IBM PC

Előfizetőink kedvezményes áron (kéthavi előfizetési díj elszámolásáért) kapják meg a programlemez, ha ezt a megrendelőlapot kitöltve visszaküldik címünkre.

A Mikrovilág Magazinak 19. óta vagyok előfizetője.

.....
(aláírás)

FLOPPY-POSTA

A PC-akció reakciója

Az előző számunkban megjelent „PC-akció”-val kapcsolatban számos érdeklődő telefont kaptunk, és jó néhány észrevétel is érkezett. Akik telefonon vagy személyesen megkeresték a hirdetésben megadott címet, bizony olyan részleteket tudhattak meg, amelyek némileg csökkentették a kezdeti lelkesedést. Idézzünk egy levélből:

Tisztelt Szerkesztőség!

Az előző havi Mikrovilágban figyelmesen elolvastam a „PC-AKCIÓ” című hirdetést, és úgy döntöttem, hogy megveszem a régóta áhított 386-os gépet. Először azt hittem, hogy a szokásos módon kell majd intézni a dolgokat (kereseti kimutatás, kezesek stb.), azután megkapom a csekkeket, leszállítják a gépet és 10 hónapon keresztül fizetem a kamatmentes részleteket.

Mivel telefonon nem sikerült információt kérni, ezért személyesen kellett felkeresni a „névtelen” céget.

Mint kiderült, a Mikrovilágban megjelent hirdetésből fontos információk maradtak ki! Ez egy „Unigroup”-módszer, ahol a vásárló annak függvényében jut hozzá a gépéhez, hogy hány további tagot tud beszerezni. Ezenkívül egy sorsolást is ki kell várnia, ami csak akkor esedékes, ha megfelelő számú jelentkező akad. Ha senkit nem tud beszerezni a leendő vásárló, akkor elképzelhető, hogy csak pár hónap elteltével kapja meg a gépét. Természetesen addig fizeti a kamatokat. A szerződés ezenkívül 90%-ban olyan feltételeket szab, amelyek nem valami előnyösek a vásárló számára. Érdekes az is, hogy a sorsolás után a cég határozza meg a szállítási határidőt.

Tulajdonképpen a módszer ellen – mivel most hallottam először róla – semmi kifogásom nincs. Annál inkább baj van a hirdetéssel, amit ez a cég feladott! Félrevezetőnek, hiányosnak tartom. Maga az akció pedig szerintem nem volt megfelelően előkészítve. Így maradt számomra – és sok csalódott felhasználó

számára – az egyetlen lehetőség: kuporgatom tovább a pénzemet. Lehet, hogy így jobban járok!

Üdvözlettel:

Matusa István

Mint kiderült, nem a hagyományos részletfizetési akcióról van szó, hanem egy olyan értékesítési szisztémáról, amelyet az autvásárlás világából kölcsönöztek. A módszer lényege röviden az, hogy megfelelő számú jelentkező esetén tízes csoportokat alakítanak ki, akik 10 hónapon át befizetik a részletet – ám a csoport tagjai közül minden hónapban csak egy valaki kap számítógépet, mégpedig az, akit a csoportból kisorsolnak. Az esély növelhető, ha a tag újabb jelentkezőket toboroz, illetve akkor, ha egyszerre több havi részletet befizet.

Így minden csoportnak lesz olyan szerencsés tagja, aki már az első hónap után bekapcsolhatja a hón áhított gépet, és lesz olyan is, aki kilenc hónapig csak a részleteket fizeti. A módszerről egészen biztosan más véleménye lesz az első és az utolsó géptulajdonosnak. De ne felejtjük el, hogy a kamatmentes részletfizetési lehetőségnek lehet árnyoldala is, s végül tíz hónap után mindenki viszonylag kedvező áron juthat AT kategóriás konfigurációhoz. A tíz hónapon belül pedig érdemes Fortuna istenszorszonyhoz fordulni.

Abban viszont igaz van levélíróinknak, hogy a hirdetésből kimaradtak ezek az igazán nem lényegtelen információk. Amelyek ismeretében is biztosan lesznek jelentkezők, ugyanakkor biztosan kevesebb lesz a csalódott érdeklődő is.

Úgy illik, hogy a másik félnek, a PC-akció szervezőinek is lehetőséget adjunk a válaszra:

Most úgy tűnik, mintha magyarázkodnunk kellene, pedig minden bizonnyal egy több megabájtos tévedésről van szó. A „figyelmes” olvasónak természetesen igaza van abban, hogy az előző számban megjelent, ingyenesen leközölt hirdetésből kimaradtak információk. Célunk a figyelem felkeltése volt és nem a PC-VÁSÁRLÓI KLUB Működési Szabályzatának tételes felsorolása – bár a kamatmentességet akkor is hangsúlyoztuk.

Akik vették a fáradságot, hogy személyesen, levélben vagy telefonon megkeresve bővebb információt szerezzenek, azoknak ezer örömmel (ingyen és bérmentve) adtunk, küldtünk teljes értékű felvilágosítást.

Célunk az, hogy az egy összegben gépet venni nem tudó – de hön óhajtó – számítógép-barátoknak lehetőséget kínáljunk egy AT kategóriás konfiguráció részletfizetéssel történő beszerzésére. Mindezt *bolti áron*, a lízingnél szokásos kamatterhektől mentesen. Ezt csak csoportos szervezéssel tudjuk elérni, vagyis összegyűjtünk annyi – egy géptípust vásárolni szándékozó – vásárlót, hogy a havi részletekből beszerezhető legyen az áhított gép. Ezt a gépet a csoport tagjai között kisorsoljuk – így az első befizetés után már géphez jutott a csoport egy tagja. A következő hónapokban a befizetésekből újabb gépek beszerzése válik lehetővé, ami szintén sorsolások útján lel gazdákra.

Ezt az „UNIGROUP” módszert egészítettük ki egy olyan pontrendszerrel, ami mindkét félnek kölcsönös előnyöket kínál. A csoportlétszámot 20 főre emelve havonta két gép beszerzése válik lehetővé. Az egyik gépet a csoport még gépre váró tagjai között sorsoljuk ki, a másikat a legma-

gasabb pontszámot gyűjtött csoporttag veheti át. Egy tag pontjainak száma megegyezik az általa toborzott PC-KLUB-tagok számával. Ráadásul minden egyes pont 100 forinttal csökkenti a még hátralévő havi részletek összegét! (A pontgyűjtés a futamidő végéig aktuális, növekedéséről és ezáltal a további befizetések változásáról azonnal kap értesítést.) Nem nehéz kiszámolni, hogy egy kis szervezői munka mekkora anyagi előnyt eredményezhet.

Fontos megjegyezni, hogy a gépek átvételénél lehetőség van az egyéni igények figyelembevételére is. A kiegészítések felárát egy összegben kell kifizetni a gép átvételekor. Ekkor kezd el ketyegni a 12 hónap garancia is.

Hisszük, hogy a fenti konstrukció működőképes, sőt talán nem vakmerőség kijelenteni, hogy sikerre számíthat, mivel a vásárlók érdekeit igyekszik a maximális mértékben figyelembe venni. (A

cég nyeresége a nagy tételben való beszerzésben rejlik és így nem a vásárló zsebébe terhel.)

A kedves levélíró figyelmét valószínűleg elkerülték a fenti tények, mindazonáltal hálásak vagyunk neki, hogy lehetőséget adott az esetleges további félreértések megelőzésére. Továbbra is ezer örömmel adunk és küldünk tájékoztatást az érdeklődőknek, akiket előre is üdvözlöl a PC-VÁSÁRLÓI KLUB nevében:

Nagy Zoltán.

Egy gépelt sor 36 karakter, ára 100 forint
A szöveget és a befizetést igazoló nyugtát (rózsaszín postautalványon) az alábbi címre küldjék:
IDG Magyarországi Lapkiadó Kft.
1536 Budapest, Postafiók 386
Bankszámlaszámunk: 203-28016

IBM XT/AT programcsere. DS/HD originál lemezek eladók.
 Szőnyi László,
 Budapest, Tavirózsa u. 5.
 Tel.: 184-8471

IBM programcsere a legnagyobb választékból! Érdeklődni lehet 9-18 óráig, szombaton 9-12 óráig.
 Tel.: 183-1817, 1142
 Budapest,
 Álmos vezér útja 17.

Amiga-programok lemezzel együtt 80 Ft, lemez nélkül 30 Ft. Kiegészítők, + 512k 3700 Ft, Action Replay III. 14 500 Ft, képdigitalizáló 900 Ft (HAM) külső drive stb.
 Amiga Box,
 1399 Budapest, Pf. 783

TVC-programok 7-9 Ft/db áron, TVC, C-64 monitorok, nyomtatók 4500 Ft-tól eladók.
 Dobrovics Zsolt,
 8790 Zalaszentgrót, Pf. 62

Amiga-programok nagy választékban! Lemezen listát küldök!
 Dikó István,
 1053 Budapest,
 Veres Pálné u. 9.
 Tel.: 137-3193

Enterprise! Válaszd a legmegbízhatóbbat! Fantasztikus választék. Napról napra új programok első kézből. Ezt nem szabad kihagyni! Válaszborítékért lista.
 Csomós Tibor,
 7261 Taszár, Pf. 18

C-64-re a legújabb '92 játékprogramok eladók lemezen/kazettán (15 Ft/db).
 Shich Ádám,
 1035 Budapest,
 Miklós u. 3. VII/35.
 Tel.: 188-4665

Amiga-programok nagy választékban eladók. 3,5"-es lemezen csak 75 Ft, 5,25"-es lemezre is!
 PPK, 7632 Pécs,
 Bókai u. 32.

Amigára több mint 5000 lemeznyi program!
 Keresztes Gábor,
 1142 Budapest,
 Laky köz 11.
 Tel.: 251-2523

C-64-es programok eladók kazettára! 3-6 Ft/db. Válaszbélyegért 6700 db programról listát küldök!
 Bohács Tibor,
 4320 Nagykálló, Petőfi u. 8.
 Tel.: 42-63-389

Eladó Atari 520 STEM számítógép fekete-fehér kis TV-vel és sok tartozékkal (32 lemez, egér, joy, 40 db-os 3,5"-es lemeztároló, porvédő stb.).
 Ára: 30 000 Ft.
 Balya Levente,
 2870 Kisbér, Komáromi u. 1.

C-128/64 gépre programcsere - vétel, eladás. 9100 program közül lehet választani. Kérésre listát küldök.
 Járóka László,
 1148 Budapest,
 Adria sétány 6. L/I. 2.

Enterprise-hoz spectrum emulátor eladó.
 Steidl Csaba,
 2600 Vác, Gerle u. 22.

Magyar nyelvű Amiga-szakkönyvek kaphatók! Amiga Basic, DOS, felhasználási kézikönyv, gépi kódú programozás, hardver-reference Manual, C nyelv. A könyvek több száz oldalas, szakszerű fordítások! Kérésre részletes tájékoztatást küldök!
 Haár László,
 1133 Budapest, Dráva út 11.
 Tel.: 173-2008 (10-18-ig)

A legújabb Amiga-programok nagy választéka. Listát küldök!
 INF Soft,
 3201 Gyöngyös, Pf. 253

TVC-re megjelent a GATO Bt 02-es DEMO-ja: 37 db fájl, 500 kB méret, digitalizált képekkel és demókkal!
 Megrendelhető lemezen 250 Ft-ért, vagy kazettán 350 Ft-ért!
 Gutbrod András,
 2023 Dunabogdány,
 Kossuth L. 190.

Új IBM XT TURBO 640 kB CGA színes grafikus kártyával 2 db 360 kB-os drive-val eladó.
 Tel.: 277-2751

C-16 +/4-es színvonalas programok olcsón eladók. 91-92-es játékok, felhasználói programok, demók. Lemezen és kazettán. Nagyobb programvásárlás esetén kedvezmények. Válaszborítékért listát küldök.
 Tisóczki Tamás,
 6100 Kiskunfélegyháza,
 Szent Imre herceg u. 35.

Eladó C-64 + új floppy + monitor + magnó + sok egyéb. Ár: 32 000 Ft.
 Bódogh Attila,
 2800 Tatabánya,
 Dózsa Gy. u. 59. 2. lph. 4/3.

Enterprise-programok eladók. Válaszborítékért listát küldök. 2000 program, kedvezmények, ajándékok.
 Zemen László,
 1164 Budapest, Olló u. 16.

Hi Enterprise-os barátom! Tőlem megszerezheted e gépre a legjobb játékokat, legszebb demókat, legszuperebb zenéket.
 Válaszborítékért katalógus.
 Nagy Zita,
 2143 Kerepestarcsa, Pf. 21

C-64-re színvonalas lemezes játék- és felhasználói programok kaphatók.
 Ár: 10-15 Ft/prg.
 Válaszboríték nem kell!
 Lőrincz Endre,
 6800 Hódmezővásárhely,
 Pf. 88

A NÉV a nagy nevek mögött



KAO - EZ A NÉV ÖNNEK LEHET, HOGY NEM ISMERŐS, DE ISMERŐS OLYAN CÉGEKNEK MINT PL. MICROSOFT, LOTUS ÉS SOK SOFTWARE - GYÁRTÓNAK, AKIKNEK A SAJÁT NEVÜK SZEREPEL KAO LEMEZEKEN. A MÁGNESLEMEZEK JÓ MINŐSÉGE ÉS MEG-

BÍZHATÓSÁGA A KAO-T AZ ÉSZAK-AMERIKAI PIAC EGYIK LEGNAGYOBB SZÁLLÍTÓJÁVÁ TETTE. TEHÁT, HA A LEMEZEK, STREAMER KAZETTÁK ÉS DAT-OK MINŐSÉGE ÉS MEGBÍZHATÓSÁGA ÖNNEK FONTOS, AKKOR CSAK EGY NEVET TARTHAT SZEM ELŐTT : KAO.

KAO®

MAKROTREND - 1143 Budapest, Hungária krt. 65 - 67. Tel: 183-4356 Fax: 163-7888

hi fi áruház

Játék és szórakozás az egész családnak

Nintendo®

Ez olyan játék, melytől bizsereg az ember uja.
Mindez asztali és hordozható kivitelben.



A GAME BOY játék és szórakozás, melyet mindenhová magaddal vihetsz.
Az izgalmas játékok minden képességedet próbára teszik.
Szórakozás az n-edik hatványon.
Akár egyedül, akár társaságban játszol vele, fantasztikus világot nyit meg
előtted a

Nintendo®

Az eredeti japán játékkazetták szinte teljes választéka mindkét típushoz
kapható az áruházunkban

Címünk: 1052 Budapest, Belgrád rkp. 22.
Nyitva tartás: hétköznap 10-18 óráig, pénteken 9-17 óráig

RAMOVILL

Egyel a szelvényvel való vásárláskor
áruházunkban 5% árengedményt adunk
minden NINTENDO termékünkre.
RAMOVILL