



mikro számítógép magazin





A METROPRINT GT., Budapest, Bajcsy-Zsilinszky út 78. szám alatti üzlete kiváló minőségű hirdás-technikai és számítástechnikai cikkek, valamint azok tartozékainak gazdag kínálatával várja egyéni és közületi vásárlóit egyaránt.

Videomagnósnak szóló ajánlatunk: közkedvelt magyar és külföldi videofilmek kölcsönzése, illetve árusítása.

Nyitva tartás: naponta 10 órától 18 óráig, csütörtökön 10 órától 20 óráig

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG LAPJA

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Bakos Tamás
(programozástechnika)
Broczkó Péter
(hírek)
Ferencz Mária
(tervezőszerkesztő)
Kovács Győző
(levelezés)
Petróczy Judit
(könyvek)
Pinke György
(NJSZT, alkalmazások)
Simonyi Endre
Szebentszki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné
Vizesi Mária

Címképünk:
Kiss Ilona munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levél cím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap-
és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkezelés
hivataloknál
és a Posta Hírlap-előfizetési
és Lapellátási Irodáján
(1900 Budapest XIII.,
Lehel u. 10/A)
vagy átutalással a 215-96 162
pénzforgalmi jelzőszámra.

Megjelenik havonta.
Egy szám ára 30,- Ft
Előfizetési díj:
egy évre 360,- Ft
fél évre 180,- Ft
Külföldön terjeszti
a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
86-0253



Szikra Lapnyomda
Budapest (88-1371)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

Tartalom

2	Tízéves az IIG
6	Feladatok — megoldások
18	Bioritmus
20	Játék a betűkkel
22	Fortuna a számítógépen
28	Rendszerfejlesztési eszközök
31	Van új a nap alatt?
34	Merre tart a világ?
39	μINFORM
40	Olvastunk . . .
44	Egy sarrakkal olcsóbb!
45	Programtermék
48	Adok — veszek — cserélek

ISKOLA-SZÁMÍTÓGÉP 3

3	Mérés és irányítás számítógéppel
4	A TVC 64 k+
5	A szem és a fül duójának „mestervizsgálja”

CSIPEGETŐ 7

7	Bájtspórolás
8	Újra Impossible! EXOS 2.1 — diákszemmel
9	Airborne Ranger
10	Képpont-pozicionált print. TOP-lista

PROGRAMOZÁSTECHNIKA 11

11	BASIC és gépi kód
12	Hatáskörök és láthatóság
14	Összegzés nagyméretű tömbön C64-re

ENTERPRISE 24

24	A programtárolás ábécéje
25	Megkérdéztük az ENTERPRISE-ről
26	Az EXOS 2.1
27	Rajzoljunk? — Igen! Mi a manó?

μKLUB 37

37	Védelem a „gyilkos” ellen
37	Adom a magyarázatot!
38	Ki ad magyarázatot? Egysorosok

SAKK 42

42	A sakkprogramozókat segítő statisztikai adatok
----	--

AZ OLVASÓ ÍRJA 43

PONTVADÁSZAT 46

HÍREK — ÉRDEKESSEGEK KÖNYVEK 48

μ mikro számítógép
magazin



NEUMANN & UTFRANC

Néhány héttel ezelőtt grazi levelet hozott a posta. A feladó Hermann Maurer professzor volt, aki meghívót küldött az intézet tízéves jubileumát megünneplendő két-napos szimpóziumra. Ez eddig magánügy, ha azonban azt is hozzáteszem, hogy az IIG volt a ma legkorszerűbbnek mondott Videotextrendszernek — amit a Magyar Posta is megvásárolt, és az év végén már talán üzembe is helyez —

felhasználó-, az összes többi pedig postaorientált. Egyes statisztikák szerint ugyanis ebben a rendszerben egyórás terminálmunkához a terminál kezelője legfeljebb 5-10 percig használja a telefonvonalat. A többi rendszerben a terminál általában nem intelligens eszköz, így a vonalat mindaddig tartani kell, amíg a terminálon dolgoznak.

Visszatérve az IIG-hez és a MUPID-hoz, az intézet egy sor más

ványának megfelelően a Control Data-val együtt hoztak létre. Ez a szerzői eszköz (AUTHORING TOOL) az AUTOOL volt. Az IIG befolyását mutatja, hogy ma mintegy húsz laboratórium működik különböző egyetemeken Ausztriában, Kanadában, az NSZK-ban és az USA-ban.

A MUPID projekten kívül az IIG számos más elméleti és gyakorlati kutatást is ellát, azok közül melyek intézményként is működik. Követve a hagyományokat, a hallgatók egy része *technikai matematikai* stúdiumokat hallgat, amely elnevezés a számítástudomány matematikai és elméleti aspektusait takarja. Az 1985/86-os tanévben indult telematika szakon az előbbihez képest sokkal gyakorlatibb tárgyakat oktatnak.

Az évfordulót ünnepelve a hivatalos szónoklatokból nem hiányoztak a szuperlatívuszok sem, de ennek a kis intézetnek az országos hatását tekintve semmi sem tűnik túlzásnak. Az intézeti beszámolókból viszont az hallatszott ki, hogy még ennél is többet tehetnének, ha több kutatói/tanári állást és a kutatásokhoz még több anyagi támogatást kapnának.

A szimpózium alkalmával röviden elmondták a következő *tízéves* tervüket is, amelynek egy részét ma még nem tudják megfogalmazni, hiszen a számítástudomány olyan gyorsan fejlődik, hogy bizonyos témákban ad hoc módon és nagyon gyorsan kell dönteni, amikor azok megjelennek. Persze ők sem prófétaik a saját hazájukban. Például a számítógéppel segített tanulásban (CAI) és annak gyakorlati alkalmazásában az egyetem többi részlege hozzájuk képest fényévreire el van maradva. Szeretnénk létrehozni egy, az egész egyetemet kiszolgáló központosított CAI rendszert központi kurzusbankkal, az egész egyetemet behálózó terminálrendszerrel. Ez a fejlődés együtt járhatna azzal, hogy a telematika integrálódna más egyetemi intézetek stúdiumába is. Nagyon fontos jövőbeni céljuk a kooperáció más kutatóintézetekkel, termelővállalatokkal, melyek részben anyagilag is támogatják fejlesztéseiket, kutatásukat.

Hosszú távú tervük között szerepel nemzetközi kapcsolataik fenntartása és bővítése is, hiszen elsősorban a *kis országok* még kisebb kutatási szervezeteinek szükségük van arra, hogy a legújabb kutatási eredményekkel *közvetlenül megismerkedjenek*. Hogy ilyen módon kövessék vagy — mint az IIG példa rá — meg is előzék (MUPID és Vix) a sokkal jobban felszerelt és anyagiakkal is jobban ellátott kutatási intézményeket és országokat.

Véleményem szerint ez a tanúsága számunkra az IIG első tíz sikeres esztendőjének.

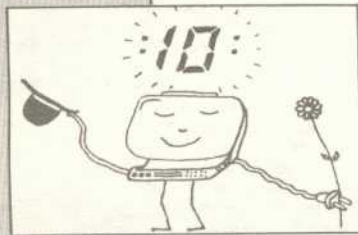
KOVÁCS GYÖZÖ

TÍZÉVES AZ IIG

Institut

für Informationsverarbeitung,

Graz



„Esti ima”
MUPID-unk a VIDEO-
TEX-ben
Amelyet
a Te szabványodban
kódoltak.
Jöjjön el
a Te UNIT-Separát-
rod,
Legyen meg a Te
SCI-d
mind C0-ás, mind a
C2-es
változatban.
A mi napi
teleszövegünket
add meg nekünk ma.
Felejtsd el
a mi HIBÁINKAT,
Miként mi is
elfelejtettük azt
a Postának.
És ne vigy minket
a kísértésbe,
De szabadíts meg
bennünket
a CONTROL-RESET-
től,
Mert tiéd a MEMO-
RIA,
A Z80 és a lemezgyé-
ség
Mindörökre*00#”
(Ismeretlen
grazi szerző művének
nyers fordítása.)

a „szülőháza”, Maurer professzor pedig a „szülőapja”, akkor az IIG tevékenysége egyáltalán nem közböcs számunkra.

1980-ban kapta az intézet a megbízást az osztrák Videotext-rendszer kifejlesztésére, és egyben az osztrák posta hivatalos tudományos tanácsadója is vált ebben a témában. Kutatási és oktatási feladatai is gyorsan nőttek.

A MUPID (Multi Purpose Universally Programmable Intelligent Decoder, vagy ahogyan tréfásan nevezik: Maurer und Posch Intelligent Decoder; Posch professzor volt a hardvertervező) projekt 1981-ben kezdődött. A feladat egy részét az *alkalmazandó rendszer kiválasztása* és meghatározása volt, másrészt a rendszerhez szükséges hardver- és szoftverelemek kifejlesztése. Legalább ilyen fontos volt az első alkalmazásoknak a rendszerben való megvalósítása is.

1982-ben készült el az első 200 MUPID-terminál, ami a finom felbontású, nagy színválasztékú grafikával igen nagy figyelmet keltett a Videotext-világban. A MUPID leginkább azzal tűnt ki a többi meglévő rendszer közül, hogy ez volt az első olyan Videotext-hálózat, amely lehetségessé vált az ún. táv-programozás vagy *teleszöveg*. Az osztrák Vix-rendszer abban különbözik ugyanis a többitől, hogy a *műveletek végrehajtása*, azaz a programok futtatása nem a központi gépen, hanem a *terminálon* történik. En azt szoktam mondani, hogy az osztrák rendszer

feladata mellett tulajdonképpen *kétszer fejlesztette ki a dekódert* is. Először egy Prestel szabványú rendszer készült (CO), majd pedig a német piacra spekulálva kifejlesztettek egy CEPT szabványú rendszert (C2) is. Ma is ez működik Ausztriában.

Az IIG egyik legnagyobb eredménye volt, hogy a MUPID-ot és a Videotext bevitte az oktatásba is, *oktatási laboratóriumként*. Graz központjában van az intézetnek és a laboratóriuma, számos tároló számítógéppel (file server) és MUPID, illetve IBM PC terminállal. Nemzetközi együttműködéssel és különféle intézmények, nemkülönben az osztrák állam támogatásával egy rendkívül impozáns projekt (COSTOC — Computer Supported Teaching of Computer Science) indult a *számítástudomány számítógépes tanulására*. A tanuló a szaktárgyak egy részét már nem előadásokon hallgatják, hanem a már említett laboratóriumban MUPID-terminálokon *kérdézik* le a központi tároló számítógéptől. A COSTOC projekt elkészült kurzusait az országos Videotext-rendszerekre is le lehet hívni, így akinek otthoni terminálja van, annak még az egyetemre se kell bemennie, hogy bizonyos tantárgyakat megtanulhasson.

Az IIG azért is tetszik nekem, mert ahhoz a felsőoktatási intézménymodellelhez közelít, amelyet én is ideálisnak tartok, tudniillik ahol nem az *oktatáson*, hanem a *tanuláson* van a hangsúly. A tanár feladata, hogy a tananyagot a *közreműködése nélkül* a tanuló megtanulhassa. Ebben az általam ideálisnak tartott *tanulási intézményben* a tanár pedagógus, tervező, konzulens, kutatásvezető, egy szóval a tanulás irányítója és nem a tananyag orális előadója.

A COSTOC projektjéhez a megfelelő programkészítést is ki kellett fejleszteni, amelyet a PLATO szab-



MÉRÉS ÉS IRÁNYÍTÁS SZÁMÍTÓGÉPPAL

II. RÉSZ

A legutóbbi egy-két évben a számítógépeket mind nagyobb számban alkalmazzák az oktatásban irányítási, mérési stb. feladatok elvégzésére is

A sorozatot indító cikkben egy a HT-1080Z-hez kapcsolható USER PORT bővítő hardver leírását ismertettük. Bizonyára vannak olyan olvasóink, akik a készüléket már megépítették és használják is. Az említett cikkben utaltunk arra, hogy az eszköz használatához két mintaalkalmazást mutatunk be.

Előjáróban hangsúlyoznunk kell, hogy nem kívánjuk az eszköz programozását „megtanítani”; ezt tekintse mindenki saját feladatának, amelyhez a HT-1080Z dokumentációjának egyik füzeté, a Használati útmutató megfelelő segítségét is ad. Célnk csak az, hogy két jellegzetes példán, egy kombinációs és egy szekvenciális hálózat működésének számítógépes elemzését bemutassuk a bővítő által nyújtott lehetőségeket.

Első példánk egy „8-ból 1” multiplexer ellenőrzése. A multiplexer nyolc adatbemenetére a USER PORT A₀...A₇ bitjei, azaz a bővítő Q₁...Q₈ kimenetei, a három kiválasztó (select) bemenetre a USER PORT B₀...B₂ bitjei, azaz a bővítő Q₉...Q₁₁ kimenetei adják az információit (lásd az 1. lista 300-as és 310-es sorát). A multiplexer kimenő jelét a bővítő D₂-es, azaz a USER

PORT B₃-as bemenete fogadja (lásd az 1. lista 360-as sorát!).

A mért vezérlő program által előállított „screen” az 1. ábrán látható. Ebben az esetben az adatbemeneteket H bemeneti szintnél vizsgáltuk, folyamatos mérés üzemmódban. Az adatbemeneteket és a kiválasztó bemeneteket vezérlő információk értékét az egyszerűség kedvéért decimálisan írtuk ki a programmal. Ha valaki úgy kívánja, a program módosításával a bitenkénti (bináris) kiírást is megvalósíthatja.

A második példa egy „szokványos” négy bites bináris aszinkron, inverz sztatikus törlésű (Clear) előreszámláló idődiagramjának felvételét mutatja be. A vezérlő program érdeklomban a számláló törlésével

DATA INPUT 'H' LEVEL OR 'L' LEVEL (H/L)? H
CONTINUOUS OR STEP BY STEP CONTROL (C/S)? C

CONTROL IS COMPLETE
REPEAT OR END (R/E)? E

MPX DATA IN (VALUE:DEC)	SELECT (VALUE:DEC)	MPX OUT
128	7	1

1. ábra

1. lista

```

10 DATA1,2,4,8,16,32,64,128
20 DEFINT A,B,D,I,S,Y:DEFSTRX,Z:CLS
30 PRINT"DATA INPUT 'H' LEVEL OR 'L' LEVEL (H/L)? "
40 X=INKEY$:IFX="H"ORX="L"THENPRINTX:GOTO50ELSE40
50 FORI=0TO7:READA(I)
60 IFX="L"THENA(I)=255-A(I)
70 NEXTI:RESTORE
80 PRINT"CONTINUOUS OR STEP BY STEP CONTROL (C/S)? "
90 Z=INKEY$:IFZ="C"ORZ="S"THENPRINTZ:GOTO100ELSE90
100 PRINT0792,"MPX DATA IN SELECT MPX OUT"
110 PRINT0856,"(VALUE:DEC) (VALUE:DEC)"
120 FORI=44TO120:SET(I,34):SET(I,46):NEXTI
130 FORI=35TO46
140 SET(44,I):SET(75,I):SET(102,I):SET(120,I)
150 NEXTI
160 FORS=0TO7:D=A(S):GOSUB200:OOSUB30
170 PRINT0924,D:PRINT0939,S:PRINT0951,Y
180 IFZ="S"THENPRINT0128,"PRESS ANY KEY!"ELSE210
190 Z=INKEY$:IFZ=" "THENI=90
200 PRINT0120," "
210 NEXTS
220 PRINT0256,"CONTROL IS COMPLETE"
230 PRINT"REPEAT OR END (R/E)? "
240 X=INKEY$:IFX="R"ORX="E"THENPRINTXZ
250 IFX="R"THENRUN
260 IFX="E"THENPRINT0448,"END OF CONTROL":END
270 GOTO240
280 REM DATA & SELECT BITS OUT
290 OUT31,7:OUT30,255:OUT31,15:OUT30,239:REM IN-N OUT-Y
300 OUT31,14:OUT30,D:REM DATA OUT
310 OUT31,15:OUT30,S:REM SELECT OUT
320 RETURN
330 REM MPX OUTPUT BIT (Y) IN
340 OUT31,7:OUT30,255:OUT31,15:OUT30,223:REM IN-Y OUT-N
350 FORI=1TO100:NEXT
360 OUT31,7:OUT30,252:OUT31,15:Y=INP(31)
370 Y=(YAND8)/8:REM MPX Y-IN
380 RETURN
    
```

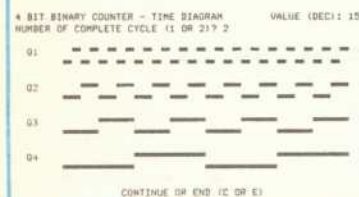
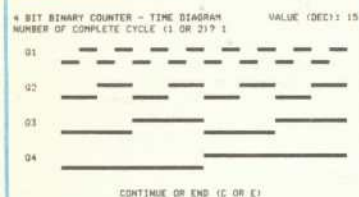
2. lista

```

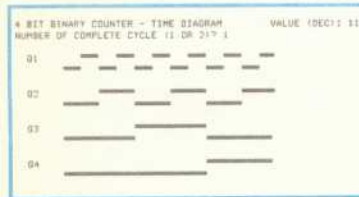
10 DEFSTRS:DEFINTB,I,X
20 CLS:PRINT " 4 BIT BINARY COUNTER - TIME DIAGRAM"
30 PRINT " NUMBER OF COMPLETE CYCLE (1 OR 2)?"
40 S=INKEY$:IFS="1"ORS="2"THENI1=VAL(S)ELSE40
50 PRINT0100,I1:IFI1=1THENI1=5ELSEI1=2
60 IFI1=2THENX1=3ELSEX1=0
70 PRINT0196,"Q1":PRINT0388,"Q2"
80 PRINT0580,"Q3":PRINT0772,"Q4"
90 OUT31,7:OUT30,255:OUT31,15:OUT30,239:REM IN-N OUT-Y
100 OUT31,14:OUT30,253:OUT31,14:OUT30,255:REM CLEAR COUNTER
110 FORX=20TO110:X1:GOSUB160:GOSUB200
120 FORI=0TO11:GOSUB250:X=X+1:NEXTI:X=X-1:NEXTX
130 PRINT0990,"CONTINUE OR END (C OR E)":
140 S=INKEY$:IFS="C"THEN20
150 IFS="E"THENCLS:PRINT"END":ENDELSE140
160 REM SUBROUTINE CLOCK-OUT
170 OUT31,7:OUT30,255:OUT31,14:OUT30,255:REM CLOCK=H
180 OUT31,14:OUT30,254:REM CLOCK=L
190 OUT31,7:OUT30,254:OUT31,15:OUT30,223:RETURN:REM IN-Y OUT-N
200 REM SUBROUTINE INPUT
210 OUT31,7:OUT30,252
220 OUT31,15:A=INP(31):B=INP(31):B=BAND15:REM 4 BIT IN
230 PRINT045,"VALUE (DEC)":":B
240 OUT31,7:OUT30,255:OUT31,15:OUT30,239:RETURN:REM IN-N OUT-Y
250 REM SUBROUTINE DRAW
260 B=(BAND1):B1=(BAND2)/2:B2=(BAND4)/4:B3=(BAND8)/8
270 SET(X,12-3*B0):SET(X,21-3*B1):SET(X,30-3*B2):SET(X,39-3*B3)
280 RETURN
    
```

kezdődik (lásd a 2. lista 100-as sorát!). A törölt jelet — ez L szint — a USER PORT A₁-es, azaz a bővítő Q₂-es kimenete adja a számlálónak. A folyamatos számlálást a program I10-es sora látja el az órajelkiadó és kimenetlekerdező szubrutinok változtatott hívásával. Az órajeleket a USER PORT A₀, azaz a bővítő Q₁ kimenetéről kapja a számláló, a négy kimenet állapotát a bővítő D₉...D₁₂, azaz a USER PORT B₀...B₃ bemenetein fogadja a számítógép.

2. ábra



3. ábra



Az idődiagram felvétele egy vagy két számlálási ciklusra lehetséges; ezt a program elején, az I1 változóba bevitt érték határozza meg.

A 2. ábrán két teljes idődiagramot láthatunk. A különbség csupán a teljes számlálási ciklusok számában van. A 3. ábra a mérés közben megszakított állapotnak megfelelő szecrent mutat be.

Második példánk — a számláló idődiagramjának felvétele — egyben arra is utal, hogy — természetesen megfelelő programmal — akár I2 sugaras oszcilloszkópot helyettesítő eszközhöz is juthatunk (sajnos ez a „műszer” egy kicsit lassú).

Két program látszólagos bonyolultsága senkit ne riasszon el a munkától. Ezek csupán azért ilyenek, hogy a képernyőn szemléltetendő eredmények áttekinthetőek, könnyen kiértékelhetőek legyenek. Emiatt a programokat a feltétlenül szükségesnél bővebb lere eresztettük.

NAGY IMRE

A TVC 64 k+

A TVC 32 k és TVC 64 k után a Videoton új TV—Computer-változat gyártását kezdte meg, melynek neve TVC 64 k+.

Az előző két TVC-hez képest a hardver által nyújtott lehetőségek két irányba bővültek:

— Használható rajta a VT—DOS operációs rendszer (a 64 k-s változatnál ehhez kisebb hardvermódosítás szükséges).

— A videomemória 64 kb-ot méretű (szemben az előző változatok 16 kb-átos videomemóriájával), így a gép négy kép egyidejű tárolását teszi lehetővé (4×16 kb-ot). A videomemória azon része, amelyet képelőállításra nem használnak, adatokat tárolhat, illetve gépi kódú programok is futtathatók benne.

A TVC 64 k+ változat új, bővített BASIC operációs rendszerrel rendelkezik, melynek verziószáma: BASIC 2.0. Az új BASIC alatt is futtathatók a korábbi 1.2 változat BASIC programjai. A BASIC programokban két ok miatt lehet szükség módosításokra:

— Az új utasításokkal azonos nevű változót használunk.

— A PLOT utasításba beépített kéréstés miatt a megrajzolt pontok, vonalak a képernyőn egy raszterponttal elcsúsznak. Néhány új parancs, utasítás:

Parancsok: AUTO (automatikus sorszámozás)

RENUMBER (átsorszámozás)

Utasítások: ON EXCEPTION GOTO (hibakezelő BASIC-szubrutin definiálása)

EXCEPTION (szándékos hiba-előidézés)

PLOT (relatív plot)

PLOT RECTANGLE (téglalap és négyzet rajzolása)

PLOT POLYGON (szabályos sokszög rajzolása)

PLOT ELLIPSE (kör és ellipszis rajzolása)

Függvény: HEX\$(x) (hexadecimális átírás)

A TVC-ben bizonyos ROM-rutinokat nyitltá tettek, és restart vektorokon keresztül biztosították ezek meghívását gépi programok számára. Így hívható a hardverkezelést végző operációs rendszer és a BASIC aritmetikai rutinccsomag. A 2.0 változatban ezek ugyanúgy működnek, sőt további rutinok is hívhatóvá válnak. A bővítések miatt azonban a ROM-címek megváltoztak. Ezért azok a programok, amelyek közvetlen ROM-hivatkozásokat tartalmaztak, futtatás előtt módosításra szorulnak. A rendszerváltozók számára fenntartott területen újabb változókat helyeztek el, ezért az itteni „üres” helyeket használó programokon is változtatni kell.

Várhatóan néhány, a régebbi típusokra íródott gyári és NOVOTRADE-program módosítások nélkül nem fog futni a 64 k+-on.

A VT—DOS operációs rendszer a Videoton TV—Computer új, fejlett operációs rendszere. A 64 k+-on futtatható, a 64 k-s gép hardvermódosítással, a 32 k-s pedig memóriabővítő használatával alkalmas a VT—DOS futtatására.

Maga az operációs rendszer egy floppycsatló kártya, egy dugaszolható programmodul és egy rendszerlemez formájában jelenik meg. A floppycsatló kártya tartalmazza a lemez fizikai kezelését és a fájlkezelést szolgáló programrészeket is, így a programmodul nélkül alkalmazva a gép a BASIC-ből használhatja a floppy perifériát.

A szabványos UPM funkciókat használó felhasználói programok futtathatók a VT—DOS-ban is. Jelentős előnye a VT—DOS-nak, hogy operátori parancsai szinte teljesen megegyeznek az IBM PC és a vele kompatibilis gépek parancsaival.

A VT—DOS is ugyanolyan lemezszervezést használ, mint az IBM PC-vel kompatibilis számítógépek. Ez azt jelenti, hogy e gépekkel a VT—DOS adatszintje kompatibilis. A lemez kompatibilitásához természetesen szükséges, hogy a TVC-n ne 720 kb-ot, hanem 360 kb-ot formáljat megformált lemezt használjunk, hiszen az előzőt a PC nem is ismeri. (Célszerű a lemezt 360 kb-ot meghajtóban formálni.)

A VT—DOS jóval hatékonyabb környezetet biztosít a felhasználói programok számára, mint az UPM (de azért programkompatibilis vele).

— Egyszerű lehetőséget ad a véletlenül törölt fájlok visszaállítására.

— Több mint 40 parancsot tartalmaz.

— Igen sok floppy meghajtó kezelésére alkalmas.

VAMOS SÁNDOR

A SZEM ÉS A FÜL DUÓJÁNAK „MESTERVIZSGÁJA”

Ha létezik olyan program, amely a kor emberének készült, akkor ez az. Annak az embernek, akinek a megfigyelőképességét szinte minden pillanat, minden helyzet próbának veti alá. Programunk (1. lista) is ezt teszi, mégpedig úgy, hogy nemcsak a zenei, hanem a figyelemmegosztási képességét is firtatja.

Használatakor 21 különböző magasságú hangot hallunk. Miközben a második hangtól kezdve arra válaszolunk, hogy a megszólaló hang magasabb vagy mélyebb lett-e az előzőnél, figyelniük kell a véletlenül felvillanó számokat is. A program méri a felhasználó időt, és a végén visszakérdezi a két számot. Értékel a válaszokat, úgy, hogy különböző súlyllyal veszi figyelembe a szám- és hangtalálások számát, valamint a felhasználó időt (300–340-es sorok).

A korábbi két (88/6-7. szám) és a most közzétett program nemcsak méri, hanem fejleszt is a zenei képességeket.

Ujjak sebessége

Az előző három programmal szemben ez a negyedik (2. lista) nem fejleszt semmilyen zenei képességet, pusztán mér. Azoknak, akik billentyűs zenét tanulnak, ajánlatos félévenként megnézni, hogy milyen fürgén és milyen egyenletesen mozognak az ujjai.

A program tág határok között nyújt sikerélményt a felhasználónak, attól függően, hogy a 10-es sorban milyen M értéket állít be. Ez és az ujjak fürgesége kölcsönösen szabják meg a program mérésáthárát. A hét ütés/mp például azt jelenti, hogy ha ennél rövidebb időn belül következik két vagy több ütés, akkor ezt érvénytelen-

```

10 PRINT "D":DIHT(21)*S=54272:FORI=0TO24
20 POKES+1,0:NEXT POKES+24,7
30 POKES+5,32:POKES+6,249
40 X=INT(RND(0)*15)+5
50 C=INT(RND(0)*15)+5:IFC>XTHEN40
60 PRINT"MELYEBB HANG?"
70 PRINT"HOSSZABB HANG?"
80 PRINT"HOSSZ LE EGY BILLENTYUT?"
90 GET# :IFR#=""THEN90
100 PRINT "D":TI#="000000":D=C:G=C
110 IF C<X THEN D=C+6
120 IT=TI:FORR=1TO21:POKES+4,33
130 Z=900*(INT(RND(0)*15)+2):T(R)=Z
140 IFT(R)=T(R-1)THEN130
150 F=INT(Z/256):A=Z-256#F
160 POKES,A:POKES+1,F:IFR=1THEN4250
170 IFR=0:OR=C THEN I=190
180 GOTO200
190 PRINT:FORI=1TO300:NEXT:PRINT "D"
200 GET# :IFT#=""THEN200
210 IFT(R)/T(R-1)>>1 AND T#="" THEN240
220 IFT(R)/T(R-1)<<1 AND T#="" THEN240
230 GOTO260
240 T=+1:GOTO260
250 FORI=1TO600:NEXT
260 FORI=1TO300:NEXT:POKES+4,32
270 POKES190,0:NEXT
280 I=INT((T1-T)/60)
290 INPUT"MI VOLT A KÉT SZÁM?"N,M
300 IFN<0 ANDM<0THEN330
310 IFN<0 ORM<0THENL=1:GOTO330
320 L=2
330 J1=#,2*INT(10*T/4+5)/10
340 J2=#L-2*INT((L-21)/10)/10:JE=J1+J2
350 PRINT:PRINTD"ES"0"VILLANT FELM"
360 PRINT "TALALATOD:"T
370 PRINT"MFELHASZNALT IDO:"I"MSDOPERC"
380 IFJE<1THENJE=1
390 IFJE>5THENJE=5
400 PRINT"MELEGJESITHENYVED ERTEKELOSE:"
410 PRINTTR(12)JE:PRINT:PRINT:END

```

1. lista

nek minősíti a program, és csak egynek veszi.

Az érdektádas után szájszóor kell a számítógép billentyűzetét leütünk. Ja, ha ezt két kézzel, felváltva a második, harmadik és negyedik ujjal csináljuk, közelről, gyengén megütve a billentyűket.

```

10 PRINT "D":INPUT"BEALLITAS(1-150)"M
20 REM###JAVASOLT ERTEK: M=30###
30 GET# :IFR#=""THEN30
40 TI#="000000":IT=TI:PRINT"R#"
50 FORI=1TO9
60 GET# :IFR#=""THENS5=S5+64:GOTO80
70 PRINT# :NEXT I:TI=GOTO180
80 FORT=1TOM:NEXT
90 PRINT"R":GOTO60
100 PRINT:POKE190,0:POKE649,0
110 FORI=104TO2023
120 IFPEEK(I)=32THENS5=S5+64:GOTO210
130 IFPEEK(I)<>79ANDPEEK(I-1)<>73THEN160
140 D=D+1:IFD=1ANDS3THENI70
150 GOTO190
160 S4=S4+1:S3=1:GOTO200
170 S3=0:S5=S5+64+1:S4=0
180 HHH=1:GOTO200
190 S3=0:S4=0:D=0
200 NEXT
210 I=INT(10*(11-T)/60+5)/10
220 AT=INT(10*(100-55*H)/10+5)/10
230 HH=INT(10*(52-100)/(10*2)+5)/10
240 PRINT100-55*H"ERVENYES ES"
250 PRINT55-H"ERVENYTELEN LEUTES VOLT"
260 PRINTID"MSDOPERC ALATT"
270 PRINT "AZ ATLPO:"AT:
280 PRINT"LEUTES MSDOPERCENKENT"
290 PRINT"MERESHTARTI:"HH:
300 PRINT"UTES/MP:"POKE649,10:END

```

2. lista

Az ütéket nem kell számolni, mert a százdik után a billentyűzet lelitása következik (100-as sor). Ha M értéke 150–30-ig változik, akkor a mérésáthár mintegy 5-17 ütés/mp-re áll be. Akkor legrealisabban ez az érték, ha M=30. Ha valaki a fejlődését akarja nyomon követni, természetesen mindig ugyanazt az M értéket kell beállítania.

Aki úgy érzi, hogy gyorsan és egyenletesen üti a billentyűket, az a 60-as sorban GOTO 80 helyett írjon GOTO 90-et, s így erős nagylításban szemlélheti a műveletét. Megjegyzem, hogy egyébként ez a mód pontatlanul tájékoztat az ujjak mozgásáról.

KALMÁR GYULA

ISKOLASZÁMÍTÓGÉP-SZERVIZ

Dudapest VII.
Baross utca 19. 1077
Telefon: 428-939

Vállalja:

**IBM PC/AT, IBM PC/XT és
Commodore típusú (C16, C Plus/4, C64,
C128) gépek javítását, átalánydíjas
szervizét,**

**egyedi programok, programcsomagok
készítését.**

A TUDOMÁNSZERVEZÉSI ÉS INFORMATIKAI INTÉZET

előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatási programokból.

Horváth Zsuzsa 685-011/2663 mellék
vagy 813-197

Budapest, Pf.: 454, 1372

FELADATOK

- MEGOLDÁSOK

- Sorozatunkat ●
- elsősorban ●
- középiskolásoknak ●
- szánjuk, ●
- de reméljük, ●
- hogy minden ●
- olvasónknak ●
- tanulási ●
- lehetőséget ●
- és szórakozást ●
- nyújt. ●

4. FELADAT: ÖTSZÖG

Egy rajzolóprogram két paramétert olvas be, majd ezek értékétől függően, különféle méretű csillagszögekkel összeállított ábrát rajzol ki. Az előző számban erre több példát is mutattunk. Emlékeztülül az 1. ábrán látható egy újabb. Ez a (2,3) paraméterekhez tartozik. Írja meg a programot!

Megoldás:

Vizsgáljuk meg először az ábrákat. Mindegyiken közös a közpén látható csillagszög. Ezt újabb csillagszögyűrről veszik körül. Furcsa módon ezek száma mindig az elsőként megadott paraméter értéke. A második paraméter jelentésére talán még ennél is könnyebb rájönni. A gyűrűk elemei láthatóan nem egyszerű csillagok, hanem több csillagból felépülő láncok. Egy gyűrűelem pontosan a második paraméterrel megadott számú csillagszögből áll.

Ez jól megfigyelhető az 1. ábrán is. Ott a központi csillagszöveget két gyűrű veszi körül, és minden gyűrű összesen ötször három darab csillagból épül fel.

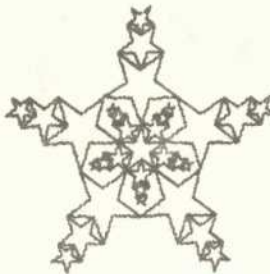
A programban a két paramétert növelnek és kiöltönek neveztem, mivel az első az ábrát újabb csillagszögyűrről rajzolásával lényegesen megnöveli, a második pedig csak ezt módosítja, kiölti.

A továbbiakban nevezük az ötszög köré írható kör középpontját és sugarát az ötszög középpontjának, illetve sugarának. Egy ötszögyűrről sugarát is értelmezhetjük: ez a gyűrű felépítő legelső ötszög középpontjának a távolsága a középső ötszög középpontjától.

Az ábra megrajzolása ilyen módon két részre tagolódik: egy növelő és egy kiöltő fázisra. A növelő fázisban meg kell határozni a következő gyűrű sugarát és legelső ötszögének méretét. Ebben a 2. ábra segíthet. A kiöltő fázisban a kiöltő ötszögek középpontját és sugarát határozzuk meg. Az ehhez szükséges arányokat a 3. ábra mutatja.

Az ábrákon látható konstansok szerepeltek már a feladat kiírásánál is: $ca = 0,381966$ és $ra = 1 - ca = 0,618034$.

Az egyes gyűrűket vizsgálva bizonyára az is feltűnik, hogy mindegyik gyűrű ötszögei azonos, az egymást követő gyűrűknél pedig ellentétes helyzetűek.



1. ábra

A programot ZX—Spectrumba Pascal nyelvben (HP47M18 fordítással) írtam meg, és bizony annak specialitásait is erősen kihasználtam. Ezt úgy próbáltam ellensúlyozni, hogy a gépfüggő részeket megjelöltem (488—1898 sorok). A megoldás ezek átolvasása nélkül is érthető. Spectrumoknak viszont ezek a rutinok más programok írásakor is sokat segíthetnek.

A Pascal nyelv választása azért célszerű, mert a strukturált programozást támogatja. Ez a példa-program jól követhető. A gépfüggő részeket (pontrajzolás, vonalrajzolás) így jól el lehet rejteni.

Az ötszögrajzolás is külön blokkba került. Megfigyelhető, hogy nincs semmilyen rajzolás az ÖTSZÖG rutinon kívül. Ennek működése jól áttekinthető, paramétereit is könnyen érthetjük: egy (X,Y) középpontú, R sugarú. H helyzetű csillagszöveget rajzol. Mivel csak kétféle helyzetű csillagszögből áll az ábra, nem meglepő, hogy H Boolean változó.

A főprogram egy ciklus belsejében van, s így lehetővé teszi a program egymás után több paraméterrel való kipróbálását. Most vizsgáljuk meg ennek működését.

A paraméterek beolvasása után a kezdőértékeket állítjuk be. Először a TABL eljárás segítségével kiszámítjuk egy egységnyi sugarú szabályos tízszög csúcspontjainak koordinátáit, mint ahogy azt már az előző két számban is közöltük. Ezután egy „csodaképlet” segítségével a legelső csillagszög méretét határozzuk meg. Ennek egyedüli célja, hogy a végső ábra a növelési lépések után is elférjen a képernyőn.

Ezután két egymásba ágyazott ciklus következik. A külső a növelést, a belső a kiöltést vezérli. Itt állítjuk meg az csillagszögek helyzetét és méretét. A számítás az elmondottak alapján követhető.

5. feladatunk már nem kapcsolódik a számítógépes grafikához. Látszólag rendkívül egyszerű, semmilyen különleges ismeretet nem igényel.

5. FELADAT: SZORZÁS

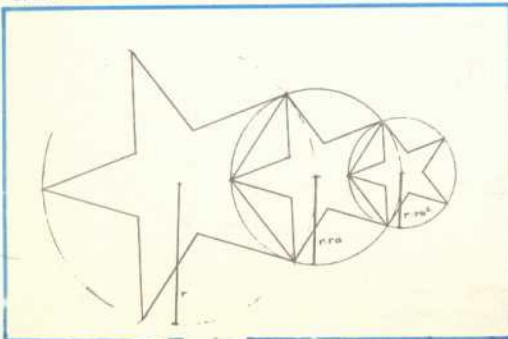
Írjon olyan programot, amelyik beolvas két negyvenjegyű számot és kiírja ezek szorzatát!

```

L 10 PROGRAM CSILLAG;
110 CONST
120 PI=3.141592653589793;
130 RA=0.618034;
140 CA=0.381966;
150 C=0.000001;
160 SGN=0;
170 SGNH=0;
180 SGNL=0;
190 SGNR=0;
200 CONST
210 G=0.000001;
220 SGNH=0;
230 SGNR=0;
240 SGNL=0;
250 SGN=0;
260 SGNH=0;
270 SGNR=0;
280 SGNL=0;
290 SGN=0;
300 SGNH=0;
310 SGNR=0;
320 SGNL=0;
330 SGN=0;
340 SGNH=0;
350 SGNR=0;
360 SGNL=0;
370 SGN=0;
380 SGNH=0;
390 SGNR=0;
400 SGNL=0;
410 SGN=0;
420 SGNH=0;
430 SGNR=0;
440 SGNL=0;
450 SGN=0;
460 SGNH=0;
470 SGNR=0;
480 SGNL=0;
490 SGN=0;
500 SGNH=0;
510 SGNR=0;
520 SGNL=0;
530 SGN=0;
540 SGNH=0;
550 SGNR=0;
560 SGNL=0;
570 SGN=0;
580 SGNH=0;
590 SGNR=0;
600 SGNL=0;
610 SGN=0;
620 SGNH=0;
630 SGNR=0;
640 SGNL=0;
650 SGN=0;
660 SGNH=0;
670 SGNR=0;
680 SGNL=0;
690 SGN=0;
700 SGNH=0;
710 SGNR=0;
720 SGNL=0;
730 SGN=0;
740 SGNH=0;
750 SGNR=0;
760 SGNL=0;
770 SGN=0;
780 SGNH=0;
790 SGNR=0;
800 SGNL=0;
810 SGN=0;
820 SGNH=0;
830 SGNR=0;
840 SGNL=0;
850 SGN=0;
860 SGNH=0;
870 SGNR=0;
880 SGNL=0;
890 SGN=0;
900 SGNH=0;
910 SGNR=0;
920 SGNL=0;
930 SGN=0;
940 SGNH=0;
950 SGNR=0;
960 SGNL=0;
970 SGN=0;
980 SGNH=0;
990 SGNR=0;
1000 SGNL=0;
1010 SGN=0;
1020 SGNH=0;
1030 SGNR=0;
1040 SGNL=0;
1050 SGN=0;
1060 SGNH=0;
1070 SGNR=0;
1080 SGNL=0;
1090 SGN=0;
1100 SGNH=0;
1110 SGNR=0;
1120 SGNL=0;
1130 SGN=0;
1140 SGNH=0;
1150 SGNR=0;
1160 SGNL=0;
1170 SGN=0;
1180 SGNH=0;
1190 SGNR=0;
1200 SGNL=0;

```

2. ábra



Az itt bemutatott BASIC hívható, amely a VCD0 memóriájából 3400 bajtot hagy szabadon, SYS 873-mal indítható és a következő utasítások pótlására szolgál: RESTORE N, RENEW, SOUND, TIRON, TROFF, MERGE, valamint billentyűnyomásokkor hanggal való visszajelzésre jó. Az egyes lehetőségek használatát az alábbi:



```

1200 VARI ? INTEGER;
1240
1245 BEGIN { TABEL }
1250 FOR I = 1 TO 10 DO
1255 BEGIN
1260 R = RUGARU;
1265 M = HE-ZET; OTSZOGET
1270 CU(1) = CGU(1+P15);
1275 END { FOR }
1280 END; { TABEL }
1300
1310
1315
1320
1325 PROCEDURE OTSZOG
1330 { (X,Y) KOZEPPOINTU;
1360 R = RUGARU;
1365 M = HE-ZET; OTSZOGET
1370 RAJZOL }
1375 END { X,Y,R,REAL };
1400 VARI I ? INTEGER;
1410
1415 BEGIN { OTSZOG }
1440 IF M
1445 THEN
1450 PLOT I;
1470 TRUNC(X);
1480 TRUNC(Y+CA+R);
1490 ELSE
1500 PLOT I;
1510 TRUNC(X);
1520 TRUNC(Y+CA);
1530 FOR I = 1 TO 10 DO
1540 IF M=000(I)
1550 THEN
1555 DRARUO I;
1560 TRUNC(X+R+SI(1));
1565 TRUNC(Y+R+CO(1));
1570 ELSE
1600 DRARUO I;
1610 TRUNC(X+CA+R+SI(1));
1620 TRUNC(Y+CA+R+CO(1));
1630 END; { OTSZOG }
1640
1650
1655 BEGIN { CSILLAG }
1660 REPEAT
1670
1710 { PARAMETERBEGLUVASAS }
1720 PAGE;
1730 URITE('NOVELO OTSZOGEK SZ
1740 READ(NOVEL);
1750 URITE('KITOLTO OTSZOGEK SZ
1760 READ(KITOLTO);
1770 PAGE;
1780
1790 { KEZDOERTEK }
1800 TABEL;
1810 RD = R+R*EXP(NOVEL+LN(ICR));
1820 CA = R+CA*NOVEL;
1830 M = FALSE;
1840
1850 { RAJZOLAS }
1860 OTSZOG(X,ZET,Y,R);
1870 FOR N = 1 TO NOVEL DO
1880 BEGIN
1890 R = RD;
1900 FOR K = 1 TO KITOL DO
1910 BEGIN
1920 R = R+R*(CA+R);
1930 R = R+R*CA;
1940 J = R-ORO(NOT(H));
1950 REPEAT
1960 UNTIL X<=SI(L);
1970 Y<=R+CO(L);
1980 R;
1990 J = J+2;
2000 UNTIL J>=10;
2010 END; { FOR }
2020 RD = RD/CA;
2030 M = NOT(M);
2040 END; { FOR }
2050 REPEAT
2060 UNTIL CH=SI(CH);
2070 UNTIL CM<>CHR(0);
2080
2090 UNTIL CM=
2100 END; { CSILLAG }

```

- Kiszámított RESTORE: RESTORE N, ahol N egy adatokat tartalmazó sor száma. Ha nincs a sorban adat, hibajelzést kapunk.
- RENEW: SYS # a ezután az előzőleg NEW-val kिरारott programunk ismét használható, ha közben nem adtunk változó értéket.
- Hang kiadása: S p1, p2, p3, p4, p5, ahol p1 a mély oszcillátor frekvenciájának megfelelő érték (129–255); p2 a közép oszcillátor frekvenciájának megfelelő érték (129–255); p3 a magas oszcillátor frekvenciájának megfelelő érték (129–255); p4 a zajeffectus értéke (129–255); p5 a hangereő (0–15).
- BASIC program nyomkövetésének bekapcsolása: SYS 7498.
- BASIC program nyomkövetésének kikapcsolása: SYS 7534.
- MERGE végrehajtása: miután a gépbe betöltöttük az alacsonyabb sorszámokat tartalmazó programot, SYS 7594-gyel betöltjük a magasabb sorszámokat tartalmazó programot, s kiadjuk a SYS 7611 parancsot. Ennek hatására a két program összefűződik, s a továbbiakban szabadon használható.
- Billentyűnyomásokor hangjelzés bekapcsolása SYS 7547; kikapcsolása: RUN/STOP+RESTORE együttes lenyomása; hangjel hangerejének állítása: POKE 7566,X, ahol X: 0–15; hangjel magasságának állítása: POKE 7571,X, ahol X: 129–255; hangjel zajeffectusának állítása: POKE 7576,X, ahol X: 129–255.

Fontos megjegyzés: a kiszámított RESTORE, hang kiadása és nyomkövetés csak a SYS 673 kiadása után működik.

A BASIC program sorainak magyarázata:

JSR 51229	RESTORE	sorszám nélküli RESTORE végrehajtása
JMP 51118		következő utasítás
JSR 52618		egy kifejezés beolvasása és
JSR 55287		20–21-be tétele
JSR 58078		a programsor kezdőcímeinek kiszámítása
BCS 5		
LDX #17		ha nem sikerült, akkor:
JMP 56231		UNDEF'D STATEMENT ERROR

BÁJTSPÓROLÁS

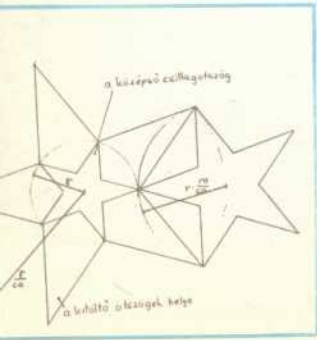
- 1 az USR vektor helyének átírása, SYS 0 helyes végrehajtása érdekében
- 2 a RAMTOP beállítás
- 3-4 a vezérlő és a RESTORE N adatainak elhelyezése
- 5-6 a RENEW-hoz szükséges adatok elhelyezése
- 7-8 a zene kiadásához szükséges adatok elhelyezése
- 9-10 a MERGE-hez szükséges adatok elhelyezése
- 11-12 a billentyű lenyomásakor a hangjelzéshez szükséges adatok elhelyezése
- 13-14 a nyomkövetéshez szükséges adatok elhelyezése
- 15 a kontrollösszeg ellenőrzése
- 20-24 a vezérlő adatai
- 30-34 a RESTORE N adatai
- 50-53 a hangkiadó utasítás adatai
- 60-62 a MERGE adatai
- 70-75 a billentyű lenyomásakor a hangjelzés adatai
- 80-85 a nyomkövetéshez szükséges adatok
- 40-42 a RENEW adatai

7622

7652

LDA 95		a következő DATA elemre mutató beállítás
LDY 96		a következő DATA elemre mutató beállítás
SBC #1		1 kivonása
BCS 1		
DEY		
STA 65		
STY 66		
JMP 51118		a következő utasítás
LDA #0		a paraméter-számláló elmentése (hang kiadása)
STA 255		a következő karakter
JSR 115		
JSR 55198		0–255-ig érték beolvasása X-be
LDY 255		
TXA		
STA 36874, Y		és a megfelelő hangregiszterbe tétele
INY		a paraméter-számláló növelése
CPY #5		összehasonlítása 5-tel
BEQ 8		
STY 255		
JSR 52989		a vessző ellenőrzése
JMP 7629		
RTS		
LDY #1		(RENEW végrehajtása)
TYA		
STA (43), Y		a BASIC program kezdete
JSR 50483		a programsorok újratáncolása
LDA 34		
LDY 35		
CLC		
ADC #2		2 hozzáadása
BCC 1		
INY		
STA 45		a BASIC program vége mutató beállítás
STY 46		
JSR 58078		a CLR végrehajtása
JMP 58292		a READY kiírása, parancs mód-ba visszatérés
7498-től: LDA 776		(nyomkövetés)
STA 7532		a következő utasítás végrehajtása mutató
LDA 777		elmentése 7532–7533-ra
STA 7533		
LDA #97		és átírása
STA 766		
LDA #29		
STA 777		
RTS		
BIT 157		program módban dolgozik a gép?

3. ábra



Az assembler program és magyarázata:

673-tól: LDA #172 : a következő utasítás végrehajtása mutató átírása

STA 776 : 684-re

LDA #2

STA 777

RTS

JSR 115 : a következő karakter a program-sorból

CMP #83 : megegyezik az S kódjával?

BNE 6 : ha nem, akkor ugrás

JSR 7622 : hang kiadása

JMP 51118 : következő utasítás vétele

CMP #140 : RESTORE kódja?

BNE 3 : ha nem, akkor ugrás

JMP 710 : RESTORE N végrehajtása

JSR 121 : aktuális karakter betöltése

JMP 51175 : a 28-as BASIC utasítás végrehajtása

710 JSR 115 : a következő karakter betöltése

CMP #58 : kettőspont?

BNE 6 : ha nem, akkor sorszámzozott

BNE 6
 JSR 52027 :a SPACE kiírása
 JSR 56777 :az aktuális sorszám kiírása
 JMP 51172 :az utasítás végrehajtása
 LDA 7532 :a nyomkövetés kikapcsolása
 STA 776 :a következő utasítás végrehajtása mutató
 LDA 7533 :visszaírása
 STA 777
 RTS
 SEI : (billentyűnyomás esetén hangjelzés bekapcsolása)
 : az IRQ vektor átírása
 LDA #135
 STA 788
 LDA #29
 STA 789
 RTS
 LDA 203 : az éppen lenyomott billentyű kódja
 CMP #64 : =64?
 BEQ 18 : ha igen, akkor nincs lenyomva billentyű, ugrás
 : hangerő
 LDA #15
 STA 36878
 LDA #199 : hangmagasság
 STA 36876
 LDA #0 : zaj
 STA 36877
 JMP 60095 : az IRQ végrehajtása
 LDA #0 : ha nincs lenyomva billentyű, ide ugrik a program
 STA 36876 : hang kikapcsolása
 STA 36877 : zaj kikapcsolása
 JMP 60095 : ugrás az IRQ-ra

LDA 45
 LDY 46
 SEC
 SBC #2 : 2 kivonása a programvég mutatóból
 BCS 1
 DEY
 STA 43 : és betöltése a programkezdő mutatóba
 STY 44
 JSR 57701 : a program betöltése szalagrol
 LDA #1 : a programkezdő mutató visszaállítása 4097-re
 STA 43
 LDA #16
 STA 44
 JMP 50483 : programsorok újralancolása

TÓTH DEZSÓ

Újra Impossible!

Bizonyára mások is tapasztalták már, hogy a különböző helyeken

FÖLDVÁRI GYÖRGY

```

100 PRINT "S":S=0
110 POKE 53280,5:POKE 53281,5
120 PRINT "IMPOSSIBLE TRAINER"
130 FOR I=828 TO 930
140 READ A:POKE I,A:S=S+A
150 NEXT
160 IF S=10740 THEN 180
170 PRINT "HIBA A DATA SORBANI":END
180 SYS 020
190 :
200 DATA 169,10,162,149,160,3,32,189
210 DATA 255,32,134,3,169,3,162,159
220 DATA 160,3,32,189,255,32,134,3
230 DATA 165,1,72,169,38,133,1,162
240 DATA 107,160,3,142,239,176,140
250 DATA 240,176,104,133,1,76,48,8
260 DATA 169,169,162,0,160,234,141,8
270 DATA 149,141,23,149,142,9,149,142
280 DATA 24,149,140,10,149,140,25,149
290 DATA 76,85,56,162,8,160,1,169
300 DATA 1,32,186,255,169,0,32,213
310 DATA 255,96,73,77,80,79,83,83
320 DATA 73,66,76,69,73,77,49,0,0
READY.
  
```

EXOS 2.1 DIÁKSZEMMEL

Néhány száz példányban megjelent az Enterprise számítógépek operációs rendszerének leírása, az EXOS 2.1 című könyv. A kiadvány nagy hibája, hogy kész, rövid programokat nem tartalmaz.

Programom a rendszer alkalmazására ad néhány lehetőséget. Több részre tagolódik:

VID — egy grafikus lap létrehozása (VIDEO MODE, COLOR, X, Y és DISPLAY gépi kódban)
PRI—PRINT — a grafikus lapra (karaktérszám, kurzorvezérlés)
CHR — a kiírandó üzenet
JOY — botkormány figyelése, a border színezése

FLS — a paletta érdekes színezése, a SEC_COUNT változó használata
STA — a STATUS sorban saját szöveg
USE — a felhasználói megszakítási rutin használata (USER ISR). Az FLS és STA rutinkra mutat.

A program megértésében segít a leírás (nagy segítség, ha a SIMON-nal fordítjuk vissza). Akiknek nincs meg az EXOS ismertető, azoknak adok némi támpontot. Hajtsunk végre az EXOS-szal mondjuk egy SET BORDER 255 utasítást.

LD B,0lh egy EXOS változó állítása (SET)

LD C,lBh lBh=27 dec, a BORD_VID rendszerváltozó száma.

LD D,FFh az irrandó érték
 EXOS l0h funkcióhívás
 RET visszatérés.

Hasonló módon működik a program többi része is.

HAJNAL CSABA

```

100 PROGRAM EXOS DEMO_HCS
110 ALLOCATE 200
120 CODE VID=HEXS
  ("01,16,01,16,01,17,10,01,
  17,01,16,02,17,10,01,18,01,
  16,20,17,10,01,19,01,16,02,
  17,10,3e,01,11,00,13,17,
  01,18,07,06,56,49,44,45,4f,
  3a,3e,01,06,01,0e,01,16,02,
  1e,1a,17,0b")
130 CODE PRI=HEXS
  ("3e,01,01,04,00,11,1f,13,
  17,08,18,04,1b,3d,21,20,3e,
  01,01,10,00,11,2f,13,17,08,
  c9")
140 CODE CHR=HEXS
  ("1b,45,50,31,32,38,
  4b,20,45,58,4f,53,20,44,45,
  4d,4f")
150 CODE JOY=HEXS
  ("3e,69,06,09,0e,00,17,0b,
  3e,00,b9,28,09,01,1b,01,3a,
  f0,b7,57,17,10,c9")
160 CODE FLS=HEXS
  ("21,08,b9,3a,f0,bf,11,
  0f,00,06,1c,77,23,3c,77,19,
  10,19")
170 CODE STA=HEXS
  ("21,74,13,01,1d,00,11,be,
  be,ed,b0,c9,45,4e,54,45,52,
  50,52,49,53,45,20,31,32,38,
  4b,20,48,4f,4d,45,20,43,4f,
  4d,50,55,54,45,52")
180 CODE USE=HEXS
  ("21,ed,bf,36,56,21,ee,bf,
  36,13,c9")
190 CALL USR(VID,0)
200 CALL USR(USE,0)
210 CALL USR(JOY,0)
220 GOTO 210
240 REM VID:4827dec
  
```

```

1 POKE1,229:POKE2,29
2 POKE 55,75:POKE56,29:0=0
3 FOR#67310754:READB:POKER,B
4 0=0+B:NEXT
5 FOR#7532107629:READB:POKER,B
6 0=0+B:NEXT
7 FOR#7622107651:READB:POKER,B
8 0=0+B:NEXT
9 FOR#7594107621:READB:POKER,B
10 0=0+B:NEXT
11 FOR#7547107593:READB:POKER,B
12 0=0+B:NEXT
13 FOR#7498107546:READB:POKER,B
14 0=0+B:NEXT
15 IF#0<27980:HEMPRINT"RDATHIB":END
20 DRTA169,172,141,8,3,169,2,141,9
21 DRTA3,96,32,115,0,201,83,208,6
22 DRTA32,195,29,76,174,199
23 DRTA201,140,208,3,76,199,2,32,121
24 DRTA8,76,32,1,190,32,115,0,201,98,208
30 DRTA6,32,29,208,76,74,199,32,138,205
32 DRTA32,247,215,19,32,198,176,5,162,17
33 DRTA76,55,196,165,95,164,96,233,1,176
34 DRTA1,136,133,63,134,66,76,174,199
40 DRTA168,1,152,145,43,32,51,197
41 DRTA165,34,164,35,24,105,2,144,1
42 DRTA200,133,45,132,46,32,96,198
50 DRTA76,116,196,169,0,133,255,32
51 DRTA115,0,32,1,169,15,164,255,198
52 DRTA153,10,144,208,192,5,240,8
53 DRTA132,255,32,253,206,76,205,29
54 DRTA96,165,45,164,46,56,233,2,176
55 DRTA1,136,133,43,132,44,32,101,225
56 DRTA169,1,133,43,169,15,133,76
60 DRTA51,197,120,165,135,141,20,3
61 DRTA169,29,141,21,3,96,165,203,201
62 DRTA64,240,19,169,15,141,14,144
70 DRTA169,199,141,12,144,165
71 DRTA8,141,13,144,76,191,234,169,0
72 DRTA141,12,144,141,13,144,76,191
73 DRTA234,173,8,3,141,108,29,173,9,3
80 DRTA141,109,29,169,97,141,8,3,169
81 DRTA29,141,9,3,96,36,157,208,6,32
82 DRTA59,203,32,201,221,76,228,199
84 DRTA173,108,29,141,9,3,173,109
85 DRTA29,141,9,3,96
  
```



AIRBORNE



RANGER

Az AIRBORNE RANGER a klasszikus kommandós játékok egyik legjobbika.

Sajnos magyar nyelvű leírása mindeddig nem létezett.

A kommandósnak és környezetének háromdimenziós ábrázolása

különösen emeli a játék színvonalát. Bár a végrehajtandó feladatok között

sok a hasonló, kezelése egyszerű és könnyen megtanulható. Így a helyzetünk mindig áttekinthető.

beállított nehézségi fokot a feladattól függően. Rangunk kiválasztásakor szerénységünkről tehetünk tanúbizonyságot.

AZ AKCIÓ VÉGREHAJTÁSA

Célunk egy „igazi” harci terep végén található. Ide kell eljutnunk élve, és a feladat végrehajtása után innen kell a helikopterrel visszatérnünk. Az akcióra mintegy 600 másodpercnyi időnk van, ennek letelte után megjön a helikopter. Célunk megvalósításának érdekében néhány ellenséges katonát és katonai objektumot meg kell semmisítenünk. Erre a következő fegyvereink vannak:

carabine — géppisztoly
law rocket — kézi rakétavető
time bomb — időzített bomba
grenade — gránát
valamint egy kés a közelharchoz.
Először felülnézetből láthatjuk a terepet, amint éppen fölötte repülünk. Ilyenkor érdemes néhány utánpótláscsomagot eldobni, egyenletesen felosztva velük a pályát (mindehhez három csomagunk van). Végül a pálya végén a csengő megszólalása után célszerű nekünk is kiugrani (lehetőleg nem aknamező közepére). Sikeres földet érésünk után a képernyő bal felső sarkában látható az éppen használatban levő fegyverünk, a hátrelévő idő másodpercben, sebességünk száma (három kis kocka) és ezektől jobbra emberünk pulzusa, amit egy növekvő vonal imitál. Ha megsebesülünk, a három kis négyzet egyike fehérré válik (a harmadik sebülés végzetes!). A RUN/STOP billentyű megnyomásával egy térképet hívhatunk elő, melyet a botkormánnyal mozgathatunk fel-le.

A TÉRKÉP

Ennek előhívásakor megáll az idő visszasmúlálása. A térkép alatt leolvasható a fegyvereink, sebességünk (wounds) és elsősegélycsomagjaink száma (first aids). A térképen különböző tereptárgyak vannak feltüntetve. A lövészárkokat hosszú, széles sáv jelöli. A kacsargörbés hosszú vonalak falak, dróterítések vagy esetleg

hőfűvások. Az aknamezőket apró halokkal jelölték. Az ellenséges állásokat kis UFO-szerű csészaljak, esetleg kis hengerek jelzik. Ledobott utánpótláscsomagjainkat kis x-ek jelölik. A helikopter érkezési helyét nagy kereszt mutatja, amely azonban csak a térképen látható. Néhány feladatnál előfordulhatnak befagyott tavak, illetve folyók, amelyeken át kell kelünk. Ezeket kívül a terep tele van fatönkökkel és bokrokkal.

A BILLENTYŰK

F1 — karabély
F3 — gránát
F5 — rakétavető
F7 — kés
RUN/STOP — térkép előhívása
INST/DEL — elsősegély használata
SPACE — lehasalás, felállítás
CRSR ↑ ↓ — futás, lassítás
← — helikopter hívása
5—6—7 + tüzgomb — időzített bomba

MOZGÁS, KÜZDELEM

Mindkettő a botkormánnyal lehetséges. Futhatunk, küszhatunk, illetve löhetünk a célkereszttel a megfelelő irányba. A haladáshoz huzamosabb ideig kell a botkormánnyal a kívánt irányba húzni.

NÉHÁNY TANÁCS

Az utánpótláscsomagok közül egyet ajánlatos nem sokkal a cél elé ledobni. Úgyeljünk arra, hogy a ledobott csomagok nyílt terepre essenek, tehát ne aknamezőre, lövészárkokba stb., mert megsemmisülnek! Takarékoskodjunk a löszerrel! Egy katonának elég egy golyó is! Kevés a rakétánk, ritkán használjuk! Ámbár a katonai objektumok ellen kitűnő szolgálatot tehet.

ÉS A VÉGE

A feladat végrehajtása vagy elhalálozásunk után megjelenik pontszámunk, és kiderül, hogy teljesítettük-e a feladatot vagy sem. Pontszámunk az elpusztított ellenség függvényében alakul.

MENÜK, FELADATOK

Az első menüben szakképzettségünkről érdeklődnek. A kezdők a Practice Rangert választják, így sem lesz könnyű dolguk. Választani mindenhol a botkormánnyal lehet a menüből (Port 2). A harmadik lehetőség (Format a new roster disk) érdekességekkel egészíti ki a feladatokat egy új játéklemez létrehozásával. A következő menüben tizenkét feladat közül választhatunk:

1. Üzemanyagraktár megsemmisítése
 2. Titkos kód kulcsának ellopása
 3. Ellenséges harci repülő felrobbantása
 4. Egy tiszt foglyul ejtése
 5. Olajvezeték elvágása
 6. Radarállomások megsemmisítése
 7. Ellenséges gépfegyveres állások elpusztítása
 8. Foglyok kiszabadítása
 9. Kísérleti repülő fényképezék ellopása
 10. Tűzcsok kiszabadítása
 11. Repülőgép-eltérítés
 12. Szabotázsakció készlettelése
- Ezután a feladat részletes ismertetése következik, majd izlés szerint növelhetjük a

KÉPPONT-POZICIONÁLT

1. lista

```

10 REM .....
20 REM - KEPPONT POZICIONALT .....
30 REM - PRINT .....
40 REM .....
50 REM - KESZITETTE: .....
60 REM - ARANYI FERENC .....
70 REM .....
80 REM - COPYRIGHT CBT 1987 .....
90 REM .....
100 REM - AZ ADATOK BEOLVASA'SA .....
110 LET SUR=0: INPUT "HOVA TOLTSEM ?": A
120 RESTORE: FOR G=A TO A+77: READ B:
IF B=>0 THEN POKE G, B: LET SUR=SUR+B:
NEXT G
130 IF SUR>7334 THEN PRINT AT 10,5:
FLASH 1: "HIBA VAN AZ ADATOKBANI":
PAUSE 0 STOP
135 REM -RELOKA' LA'S-
140 LET C=INT (A+74)/256: POKE
A+2,C: POKE A+1, A+74-C-256
150 LET C=INT ((A+75)/256): POKE
A+18,C: POKE A+17, A+75-C-256
155 REM -KII'RATA'S-160 PRINT AT
0,0: "ADDR = ":A
170 PRINT AT 2,0: "CHAR = ":A+74
180 PRINT AT 4,0: "KOOR = ":A+75
190 STOP
200 DATA 58,0,0,111,38,0,41,41,41,
237,75,54,92,9,235,237,75,0,0
210 DATA 62,191,205,172,34,6,8,197,
245,20,71,241,245,14,0,183,40
220 DATA 7,203,56,203,25,61,32,249,
120,182,119,35,121,182,119,43
230 DATA 36,124,230,7,32,10,125,
198,32,111,56,4,124,214,8,183,19
240 DATA 241,193,16,209,201,0,0,0,-1
    
```

PRINT

Ez a kis rutin (1. lista) ZX-Spectrumra készült. Lehetővé teszi, hogy a képernyő bármely képpontjára tegyünk egy karaktert bal felső sarkát, vagyis minden PLOT pozícióba, mind a 256 x 192 képpontba tehetünk egy karaktert. A rutin előnye, hogy nem törli le a hátteret, amire a kataktert teszi, hanem közte és a karakter között logikai OR műveletet hajt végre.

Közlöm a program assembler listáját (2. lista) és a BASIC-betöltőt is, amit az 1. lista tartalmaz. A BASIC program elvégzi a rutin relokalását is. Először megkérdezi, hogy hová töltsse a programot, végrehajtja a relokalást, kiírja a kezdőcímet, aztán azt a címet, ahová a karakterkódot kell tölteni, majd azt, ahová az X, illetve az Y koordináta kerül.

Az assembler program bármely fordítóba beírható, és csak az első sorban található ORG címet kell átírni a relokaláshoz. A rutin vezérlése BASIC-ből:
 POKE char, karakterkód (32-128)
 POKE koor, X-koordináta
 POKE koor+1, Y-koordináta
 RANDOMIZE USR addr

ARANYI FERENC

```

1000 ORG 6540
20 ADDR LD A, (CHAR)
30 LD L, A
40 LD H, 0
50 ADD HL, HL
60 ADD HL, HL
70 ADD HL, HL
80 LD BC, (23006)
90 ADD HL, BC
100 EX DE, HL
110 LD BC, (KOOR)
120 LD A, 191
130 CALL 8876
140 LD B, B
150 DE
160 PUSH BC
170 PUSH AF
180 LD A, (DE)
190 LD B, A
200 POP AF
210 LD C, 0
220 OR A
230 JR Z, TOV
240 VISZ
250 SRL C
260 DEC A
270 JR NZ, VISZ
280 TOV LD A, B
290 OR (HL)
300 LD (HL), A
310 INC HL
320 LD A, C
330 OR (HL)
340 LD (HL), A
350 DEC HL
360 INC H
370 LD A, H
380 AND 7
390 JR NZ, FOLY
400 LD A, L
410 ADD A, 32
420 LD L, A
430 JR C, FOLY
440 LD A, H
450 SUB 8
460 LD H, A
470 FOLY INC DE
480 POP AF
490 POP BC
500 DJNZ 1DE
510 RET
520 CHAR DEFB 0
530 KOOR DEFB 0
    
```

2. lista



KEDVES OLVASÓNK!

Bizonyára neked is van otthon számítógéped, és biztos neked is okozott problémát programkészleted hiányossága, vagy egyszerűen jó lenne valakivel – valakikkel – a gépen keresztül ismeretséget kötni. Ebben próbálunk segíteni annyiban, hogy elkezdjük a hozzád hasonló érdeklődésű, s hozzánk levelet írók címeinek összegyűjtését.

Írj te is! Írd meg, hogy kikkel, milyen típusú gépről, milyen szinten kívánsz levelezni. Küldj levelet, benne egy saját címeddel ellátott, felbélyegezett borítékok, és munkánkért cserébe egy számítógépekkel kapcsolatos írást, rövidebb programot, trükkök ismeretét, véleményét vagy valamit, ami szerinted érdekes lehet egy újság hasábjain.

A borítékokat márciusban fogjuk visszaküldeni az addig összegyűjtött címek listájával. Ha az akció sikerül, természetesen folytatjuk, s újabb leveled esetén újabb címeket küldünk.

TOP - lista

Felhasználói

1. News room E64
2. Newsmaster 03070
3. GEOS 1.32 E64
4. Printmaster E64 03070
5. Game maker E64
6. Music maker E64
7. Art Studio 1.2d E64 03070
8. HI-DOS 03070
9. 128 Starpainter E64
10. Rockmonitor III E64

Játék

1. Defender OTC. E64 Amiga
2. Aliens ZX E64 Spect.
3. The Last Ninja E64 Spect.
4. Twin kingdom E64 Spect.
5. Island caper E64 Spect.
6. Back to the future E64 Spect.
7. Titanic E64 Spect.
8. Slap flight E64 Spect.
9. Skyfox II. E64 Spect.
10. Alt.world games E64

Listánkat felhasználói, illetve játékprogramokból állítjuk össze. A legjobbakat, legérdekesebbeket a beküldött javaslatok alapján rangsoroljuk. Ehhez kérjük az olvasók közreműködését. C64-re, ZX-Spectrum-ra, Enterprise-ra, TVC-re, Atarira és IBM-re készült programokat várunk hivatonta.

Címünk: Mikroszámítógép Magazin

Szerkesztősége

1371 Budapest, Pf. 433.

Diákszerkesztőség

BASIC ÉS GÉPI KÓD

Legutóbb a decimális aritmetikáról volt szó. Részletes magyarázatot fűztem az augusztusi számban közölt bemutató listákhoz, és néhány szót ejtettem a decimális aritmetika alkalmazásáról. Most egy olyan gépi kódú rutint mutatok be, amely egy címfórmátumú egész számot BCD formájúra alakít, majd a képernyőre, illetve az aktív kimeneti fájlba írja.



Az előzőhöz hasonlóan ez a program is bemutató jellegű, de szubrutinként is alkalmazható. BASIC betetőt most sem adok, akinek monitor- vagy assembler programja van, begépelheti, kipróbálhatja.

A rutin a kezdetpufferben van elhelyezve. A lista üres sora azt jelzi, hogy az első két utasítás nem tartozik a rutinhoz, csak előkészíti azt. „Eles” alkalmazásánál az A/X regiszterpárba töltjük az átalakítandó-kiírandó számot, majd JSR \$0370 utasítással hívjuk a rutint. Az előkészítő utasítások itt a „váltóözetület kezdete” mutató tartalmát töltik a regiszterekbe.

\$0370—0373: a regiszterekben tárolt címfórmátumú számot egy kétbájtos munkaterületre tesszük le.

\$0376—038E: legalább ötjegyű BCD-szám részére kell helyet foglalni, erre a \$0380—0382 címterületet jelöljük ki. A magasabb helyértékű számjegyek \$0380-ra kerülnek. Ezek az utasítások nullázzák a szám helyét. Nagyobb terület törölésekor célszerű lehet ciklust szervezni, de most nem érdemes.

\$0381—039C: az átalakító rutin, amely a \$0383—0384 címen található bináris számot BCD alakra konvertálja és elhelyezi a kijelölt helyen. Az eljárás lényege, hogy a bináris szám jégeit egyenként a C állapotbitbe toljuk, majd decimális üzeműdoban a BCD-szám minden számjegyének értékét önmagához adva megkétszerezük. A C bit aktuális értéke minden számjegynél a kétszeres értékhez hozzáadódik. A bináris szám 16 bitjének megfelelően, a fenti művelet tizenhat-szoros játszódik le, ezt az X regiszterrel számoljuk. Az ASL utasítás helyén is állhatna ROL, ennek most nincs jelentősége, de ha netán sokbájtos bináris számot BCD-re alakítunk, akkor ez utóbbi egyszerűbbé tenné a ciklusszervezést. Figyeljék meg, hogy az I és D feltételbitket csak az összeadás idejére kapcsoljuk be.

A NOP választja 'el egymástól az átalakító és a kiírató részt. Szerepe most nem az elválasztás, hanem az, hogy a program könnyen módosítható. Ha csak BCD-re alakítani akarjuk a számot, de kiíratni nem, a NOP helyére RTS-t írunk, s máris teljesül a kívánságunk, de a kiírató rutin is épségben megmarad, és JSR \$039F utasítással hívható.

\$039F—03BA: a kiírató rész, amely nem sokban különbözik a legutóbbi számban elemzett példaprogram kiírató részétől. Itt az LDA utasításokban abszolút indexelt címzési módot alkalmazunk, és ciklusban végzünk el a kiírást. Az X indexregiszter 0 kezdőértékét az átalakító részből hoztuk tovább azzal, hogy a NOP-nak az előző bekezdésben javasolt módosításánál a hívás előtt gondoskodni kell az X nullázásáról.

Nagy hátránya a rutinnak, hogy minden alkalommal hatjegyű számokat ír ki, elől értéktelen nullákkal. Ennek elkerülésére a közeljövőben ismertetek egy módszert. A másik hátránya az általánosság hiánya: csak 16 bites számokkal dolgozhat, az eredményt csak a meghatározott három bajton tudja tárolni. Ezen is lehet segíteni.

Eldőnye a módszernek, hogy eredményként garantáltan BCD számot ad, bár ezt nem mindenki értékeli. Fontosabb ennél a gépüggetlenség. Ennek lényege, hogy csak olyan beépített rutint használunk, amely a KERNAL ugrótáblán keresztül érhető el. Ez a rutin most épenséggel a BSOUT, ami mindhárom általunk vizsgált géptípuson JSR \$FFD2-vel hívható.

A BASIC interpreterben található egy olyan aritmetikai rutin, amely az A/X regiszterpárban átadott kétbájtos pozitív egész számot ASCII alakban kiírja. Ennek beépítési pontja a C64-en \$BDCD, a VC20-on \$DDCD, a C16-en pedig \$A45F, tehát minden géptípuson más. Ha hívti akarjuk, típusától függően más-más címet kell a JSR operandusaként beírni.

Eldőnye még a most bemutatott eljárásnak a nagyobb sebesség. Ha belendzünk az interpreter ROM-listájába láthatjuk, hogy milyen nyolcvonalat működik az imént említett ROM-rutin: az egész számot először lebegőpontosossá alakítja, majd lebegőpontosról ASCII-ra, végül kiírja az ASCII karakterláncot.



A júliusi szám 12. oldalán jelent meg kedvenc FNL függvényeimnek USR függvényekkel való helyettesítéséről szóló írásom. Ott a kétbájtos, elől nélküli egész szám alacsonyabb helyértékű bajtjának értékét kiszámító FNL definíciója a következő formában szerepelt: DEF FNL(D)—0-256*FNLH(D), ahol FNLH a magasabb helyértékű bajt értékét kiszámító függvény. Különös véletlen folytán ugyanannak a számnak a 10. oldalán jelent meg a Memóriaterület mentése C64-en című írás a hozzá tartozó hétérsoros programmal. Ennek második sorában van egy hasonló szöveg, ezúttal — indokoltan — függvény nélkül. Nézem az alacsonyabb helyértékű bajt számítását: LD = KEZ AND 255.

Úgy látszik, megint túlbonyolítottam valamit. Nem a függvényesítéssel, hiszen FNL függvényét csak akkor használom, ha azt több helyről is hívom. A szükségesnél komplikáltabb képlet használata nem tetszik. Hogy tettem így? Ki ad magyarázatot? Kérem, segítsenek!

BARNA LÁSZLÓ

036c	a6	2d	ldx	\$2d	
036e	a5	2e	lda	\$2e	
0370	0e	63	03	stx	\$0363
0373	0d	64	03	sta	\$0364
0376	a9	00		lda	#\$00
0378	0d	60	03	sta	\$0360
037b	0d	61	03	sta	\$0361
038e	0d	62	03	sta	\$0362
0381	a2	10		ldx	#\$10
0383	0e	63	03	asl	\$0363
0386	2e	64	03	rol	\$0364
0389	a0	02		ldy	#\$02
038b	79	60	03	lda	\$0360, y
038e	b8			sei	
038f	f8			sed	
0390	79	60	03	adc	\$0360, y
0393	d8			cid	
0394	5b			cli	
0395	99	60	03	sta	\$0360, y
0398	88			dey	
0399	10	f0		bpl	\$038b
039b	ca			dex	
039c	d0	e5		bne	\$0383
039e	ea			nop	
039f	bd	60	03	lda	\$0360, x
03a2	4a			lfr	
03a3	4a			lfr	
03a4	4a			lfr	
03a5	4a			lfr	
03a6	09	30		ora	#\$30
03a8	20	d2	ff	jsr	\$\$\$d2
03ab	bd	60	03	lda	\$0360, x
03ae	29	0f		and	#\$0f
03b0	09	30		ora	#\$30
03b2	20	d2	ff	jsr	\$\$\$d2
03b5	e8			inx	
03b6	e0	03		cpx	#\$03
03b8	d0	e5		bne	\$039f
03ba	60			rts	

HATÁSKÖRÖK ÉS LÁTHATÓSÁG

A PROGRAMOZÁS

Öt részes sorozatunkban minderről szót ejtünk, amit olvasóinknak érdemes átgondolniuk, ha programozásra adják a

fejüket, vagy ha valamely programról véleményt kívánnak alkotni. Ezek az írások a szerzőnek lapunkban közölt

ALAPKÉRDÉSEI II.

korábbi fejtegetéseivel együtt (utalunk „Az adattípus fejlődése” és a „Függvények és utasítások” című sorozatokra, lásd 1987. 8. és 9., valamint 1987. 12., 1988. 1. és 2. számunkat) teljes körűen elemzük az e sorozat címében jelzett (problematicát) témát.

Ebben a részben azzal foglalkozunk, hogy egy programban a felhasználó által létrehozott változók és eljárások mikor születnek és meddig élnek, honnan érhetők el, el lehet-e egy adatot rejteni stb. Alapfeltevésként az, hogy ha a program szövegében egymás mellett két egyforma deklaráció áll, akkor azok egyformán működnek, azonos lesz a hatáskörük. Ha viszont más környezetben vagy más formában deklarálunk két változót, viselkedésük különböző lehet. Bizonyos nyelvekben, például a BASIC-ben erre nincs lehetőség, mégpedig azért, mert a program nem bomlik szét kisebb zárt egységekre. Ezeket az egységeket egyébként, mármint ha ilyenek az adott nyelven lehetőségek vagy megengedettek, a legváltozatosabb néven nevezzük majd: hol alprogramnak, hol rutinnak, hol szubrutinnak, hol eljárásnak, hol függvénynek. Az olvasó, aki ismer olyan magas szintű nyelvet, amelyben van paraméterlistával ellátott, felhasználó által deklarált függvény vagy eljárás, a fenti szavak bármelyikén ezt értheti, mert valami ilyesmiről beszélnek. Ha valaki még nem találkozt volna ilyennel, akkor a továbbiak megértéséhez ajánlom, hogy előbb olvassa el a Függvények és utasítások című, korábbi írásmat (lásd 1987/12. és 1988/1. és 2. számunkat — A szerk.).

A PROGRAMOK VÁLTOZÓI

A BASIC nyelvben minden változó globális, azaz a program bármelyik sorában írható és olvasható. Ez nem túl kellemes tulajdonság, mert noha a programot feloszthatjuk sorszámként szubrutinokra, ezeknek a részeknek mégsem lehetnek saját, kizárólagosan használt változói. Ezenkívül, ha a szubrutin valamilyen mennyiséggel közötti számítást hajt végre, a bemenő adato-

kat mindig bizonyos változóba kell előtte belerakni, és az eredmény adott helyen keletkezik. Ez a szubrutinok felhasználhatóságát nagymértékben zavarja, mert elég komplikált megszervezni, hogy a rutinok bármilyen sorrendben hívassák egymást. BASIC-ben gyakori programozói hiba, hogy egy meghívott szubrutin elrontja a hívó program valamilyen segédváltozójának értékét, mert a két programrész ugyanazt a változót használja valamilyen részeredmény tárolására.

Ezért azután mielőtt a szubrutin, függvény, eljárás fogalma beépült a nyelvekbe, azonnal bekerült két újabb fogalom is: a paraméterváltozó és a lokális változó fogalma. A lokális változó olyan, amelyik csak egyetlen alprogramon belül érhető el és máshonnan elérhetetlen. A paraméter olyan változó, amely a függvény meghívásakor kap értéket, vagy egy olyan lokális változó, amely a benne keletkezett értékét a futás végén átadja annak a változónak, amelyik a hívóskor a paraméter helyén állt.

STATIKUS ÉS DINAMIKUS LOKÁLIS VÁLTOZÓK

A lokális változókat tartalmazó alprogramok bevezetése rögtön problémákhoz vezet: hogyan is kell ezeket a változókat értelmezni? A láthatóság szempontjából világos ugyan, hogy amikor nem vagyunk benne a szubrutinban, akkor a lokális változók olyanok, mintha nem is lennének. De mi van akkor, amikor egy korábbi hívás után újra belépünk a szubrutinba? Teljesen friss, új változókat találunk ott, vagy a korábbi futásból ott maradt értékeket? Mindezt esetnek lehet, sőt megvan a maga értelme, a maga felhasználási köre.

Statikusnak (vagy „own” típusú) nevezzük azt a lokális változót, amelyik megőrzi értékét a futás során. Ilyenek kellene akkor, ha például egy beolvasó rutint írunk. Ez a rutin számon tartja, hogy hol tartunk a beolvasásban, és a következő hívóskor a következő egységet (karaktert, szót vagy sort) adja vissza, holott lehet, hogy fizikailag nem is ilyen egységekben olvasott be a perifériáról. A szükséges adatok nyilvántartása a rutin belügye, és annál is inkább tilos ezeket az adatokat felhasználni, mert ez mindenféle áttekinthetetlen machinációkra adna lehetőséget.

Egészen más a helyzet, ha valamilyen függvényt, például egy szögfüggvényt akarunk kiszámítani. Ilyenkor az előző számi-

tás maradványaira semmi szükség nincs, sőt esetleg ki is kell takarítani a változókat. Ilyenkor természetesen az a jobb, ha a függvénybe való belépésnél teljesen új példányokat kapunk a lokális változókból, feltöltve a deklarációban adott kezdőértékekkel.

Különösen indokolt ez a megoldás akkor, ha lehetőségünk van dinamikus méretű tömböt deklarálni, azaz csinálhatunk olyan tömböket, amelyeknek felső határait egy változó értékével adjuk meg. Ilyenkor megtehetjük, hogy például egy univerzális mátrixszorzó rutint írunk, ami bármilyen méretű mátrixokat képes összeszorozni, tehát minden meghíváskor éppen akkor a lokális tömböket hoz létre, amekkora szükséges.

A dinamikus lokális változóknak memóriagazdálkodási szempontból is nagy a jelentőségük. Ezek a változók ugyanis csak addig foglalják a memóriát, amíg a függvény működik; aminet véget ér, felszabadul a helyük. Mivel a legtöbb programnak olyan a vezérlési struktúrája, hogy először a főprogram indul el, azután ez meghív egy szubrutint, majd az egy következőt stb. — természetesen először a legutoljára hívott ér véget, majd az azt megelőző —, a memóriagazdálkodást verezszerűen lehet szervezni. Ez azt jelenti, hogy először a főprogram adatainak foglalunk helyet, ezután az először meghívott szubrutinnak, majd az ebből hivattnak és így tovább. A felszabadítás pedig éppen fordított sorrendben megy végbe, és így mindig egy változó méretű, de összefüggő memóriadarab lesz le foglalva. Vannak persze olyan nyelvek, elsősorban a párhuzamos folyamatok leírására szolgálók, amelyeknek vezérlési fonala nem ilyen egyszerű. Ezeknek a memóriagazdálkodása is sokkal bonyolultabb (például a Simula, az Ada, a Modula).

A dinamikus lokális változók használata nem csak előnyökkel jár. Általában egy icipicit csökkentik a nyelv hatékonyságát, mivel növelik a futásidőt, a fordítási időt, a lefordított program hosszát; a lefordított program nem hivatkozhat konkrét gépi címekre az adatoknál, hanem csak az úgynevezett relatív címekre a szubrutin adatterületének kezdetéhez képest, hiszen ez az adatterület csak futás közben osztódik ki.

Szerencsére a modern gépek sok és elég körmönfönt címzésesódot kezelnek automatikusan. Mindenesetre a modern nyelvekben majdnem egyöntetűvé vált, hogy a függvények és eljárások lokális változóit dinamikus módon kezelik. Csak a C nyelvben van lehetőség arra, hogy egyes változókat statikusan deklaráljunk.

GLOBÁLIS VÁLTOZÓK

Felmerül a kérdés: ha az egyes alprogramoknak vannak lokális változói, szükség van-e egyáltalán globális változókra is? Elméleti szempontból a válasz egyértelmű: nincs, sőt a program világosságát, áttekinthetőségét nézve jobb, ha globális változó nem is szerepel benne. Mint arról a függvények és utasítások kapcsán már részletesen beszéltem, egy alprogram használata akkor világos és áttekinthető a programban, ha a meghívások minden felhasznált adat „in” paraméterként, minden visszaadott érték „out” paraméterként és minden módosított változó „inout” paraméterként fel van tüntetve.

Egy lehetséges programfelépítés olyan, hogy a program gyökere egy eljárás, ez az eljárás részfeladatokat megoldó eljárásokat hív, azok újabbakat és így tovább. Ha az eljárásoknak csak lokális változói vannak, akkor minden rutinnak az összes külső adatot paraméterként kell átadni és átvenni, ami a programnak nagyon áttekinthető formát ad. Ugyanakkor némi kényelmetlenséget is okoz. Statikus változók hiányában a megőrzendő értéket vissza kell venni a hívott alprogramból, és újabb hívás esetén vissza kell neki küldeni. Ez azt jelentheti, hogy a rutinok paraméterlistája rendkívül hosszúra nyúlhat vagy nagyon boykotált (sok részadatot tartalmazó) paramétereket kell átadni és átvenni.

A másik ellenérv az, hogy a fenti felépítéssel minden rutin olyan önmagában zárt egység, amelynek semmilyen kapcsolata sincs a program többi rutinjával. A gyakorlat azonban nem ezt mutatja. Egy programon belül vannak szorosabban és kevésbé összetartozó dolgok. (Egy következő rész arról az elvről beszél majd, hogy egy eljárás egyetlen célra kell felhasználni. Nem célszerű egy olyan eljárást írni, amely minden többé-kevésbé összetartozó feladatot egyszerre kezelhet, hanem a különböző részfeladatokra különböző rutinokat kell írni. Például ha van egy beolvasó rutinom, ami egy szöveget szavanként olvas be, akkor ez a rutin számon tarthatja, hogy hányadik sor hányadik szavánál tartunk. Elképzeltető olyan alkalmazás, amelynek minden egyes szövegből szükségesége van ezekre az adatokra, de olyan is előfordulhat, hogy csak kivételes esetben kíváncsi rá. Az utóbbi feladatnál kár lenne ezeket az adatokat minden esetben visszaadni; elegendő, ha van egy másik függvényünk is, ami akkor adja vissza a szövegbeli pozíciót, amikor szükséges. Nyilvánvaló, hogy a két függvénynek vannak közös statikus adatai, amelyek viszont jó lenne, ha rejte maradvánának mások számára.)

Itt lép be a programstruktúráls újabb eleve, mely szerint a programot nagyobb egységekre (modulokra, osztályokra, csomagokra) kell bontani. Egy ilyen egységbe az egymással szorosan összetartozó változókat, függvényeket és eljárásokat gyűjtjük egybe. A modulnak van egy látható és egy privát része. A látható részbe azok az adatok és eljárások kerülnek, melyeket másoknak is hozzáférhetővé teszünk, míg a privát részben elrejtjük a megvalósítás technikai részleteit, és itt védve tartjuk azokat az ada-

Összefoglalás

A modern programozási nyelvek fő törekvése, hogy a programot kisebb, jól áttekinthető részekre bontsuk, ezek között a lehetséges kapcsolatokat minimálisan szorítsa — és ezekre is jól láthatóan felhívja a figyelmet. A részek legyenek olyan kicsik, amelyeket bárki könnyen áttekinthet. Egyes programozási módszertanok még méretet is előírnak (1 lap, 100 sor stb.); mintha egyetlen utasítást nem lehetne olyan boykotált tenni, hogy senki se értse...!

Mindezek alapján tehát: a program függvényekből és eljárásokból épül fel, amelyek egy-egy feladatot oldanak meg. Az egymással szorosan kapcsolatban levő alprogramok a hozzájuk tartozó közös adatokkal együtt egy-egy önállóan lefordítható és önállóan elolvasható és megérthető modul alkotnak. A modulok között vannak alacsonyabb és magasabb szintűek. Az alacsonyabb szintűek általánosabbak, a magasabb szintűek az intelligensebbek és feladathoz kötöttebbek. A magasabb szintűek az importáltjukon adják meg, hogy mit vesznek át az általánosabb rutinokból, és látható részükben adják meg, mire is tartalmazhatják őket a külvilág. A privát rész tartalmazza a feladat technikai részleteit, ezek azonban az egyszerű felhasználót általában nem érdeklik.

Ebben a programfelépítésben a dolgoknak pontosan három szintjét különböztetjük meg hatásuk alapján:

— Vannak az egész programra érvényes, bárhol felhasználható dolgok, amelyek a modulok látható részében szerepelnek.

— Vannak egy modulon belül felhasználható dolgok.

— Végül vannak lokális adatok, amelyek csak egyetlen alprogramon belül élnek. Rendszerint a rutin hívása után jönnek létre és a munka befejeztével meg is szűnnek.

FARKAS ERNŐ

tokat és kezelési eljárásokat is, amelyeket nem engedünk át bárkinek. Azért, hogy a modulokat önállóan is lehessen fordítani, de annak érdekében is, hogy a modulok szövege önmagában elolvasható és megérthető legyen, minden modult egy importálttal célszerű kezdeni, amiben felsoroljuk, hogy az milyen más modulokban szereplő mennyiségeket használ fel.

A program végül is nem más, mint egy modul, mely az importáltján összegyűjtíti azokat a modulokat, amelyek a feladat megoldásához szükségesek, és ezek segítségével megoldja a feladatot. Csak azokat a modulokat kell importálnia, melyek a neki megfelelő szinten oldják meg a megfogalmazott feladatot, de ezek a modulok automatikusan magukkal hozzák az alacsonyabb szintű, általánosabb modulokat is, amiket viszont ők importálnak. Így a modulok között hierarchia alakul ki az aktuális megoldástól az általános eszközök felé.

A programok ilyen felépítésénél a változókat három csoportba sorolhatjuk: vannak az egész programra nézve globális változók, amelyek a modulok látható részében szerepelnek, vannak az egy modulra érvényes változók, ezek az előbbiekhöz hasonlóan statikus helyfoglalásúak, és végül vannak az egy eljárásról vagy függvényről belülről dinamikus helyfoglalású változók. Az iméntihez hasonlóan az eljárásokat két csoportba sorolhatjuk: az egész programra érvényes, bárhol felhasználható eljárásokra és az egy modulon belül érvényes segéd-eljárásokra.

A BLOKKSTRUKTÚRÁRÓL

Azoknak, akik jártasak a modern nyelvekben, a fenti csoportosítás és javasolt programfelépítés nagyon „primitívnek” tűnhet, ha ezeket összehasonlítjuk a lehetőségekkel. Hiszen mindaddig semmit sem beszéltem a blokkstruktúráról, ami már mintegy húsz évtizede szinte minden programozási nyelvnek egyik építési elve, és összekombinálva a program modulonkénti építésével, elképesztően komplikált programstruktúrák létrehozását teszi lehetővé.

Mi is hát az úgynevezett blokkstruktúra? Az elgondolás a következő. A program egy deklarációs részből és egy végrehajtható részből áll. A deklarációs rész adatokat és eljárásokat ír elő, a végrehajtható rész pedig ezeket használhatja fel. Az eljárások szintén két részből állnak: egy lokális deklarációs részből és egy végrehajtható részből. A lokális deklarációk ugyanolyanok, mint a globálisak, de csak az eljárásokon belül használhatók. Ezek között is lehet eljárásdeklaráció, de ennek is lehetnek még lokálisabban deklarációi és így tovább. Ily módon tetszőlegesen mély hierarchia alakulhat ki a függvények között. Egy függvény belsejében látható az összes változó és függvény, amely azokban a függvényekben szerepel, amelyek a hierarchiában felette állnak, azaz amelyeknek a szövegébe ő is beletartozik. Nem lát bele viszont a függvény azokba a függvényekbe, amelyek a hierarchia másik ágán vagy nála lejjebb állnak.

Bár a magas szintűnek tekintett és elterjedt nyelvek (Pascal, C, Ada, Modula) mind tartalmaznak a blokkstruktúrát, a programozáseléttel foglalkozók ma már kifejezetten kerülendőknek tartják az alkalmazását. Hogy miért? A fő ellenérv:

— Alapelveinél fogva arra ingerli a programozót, hogy mellékhatásokat csináljon, azaz felhasználjon és módosítson olyan változókat, amelyek nem az adott (legbelső) függvényhez tartoznak, így a függvények azonos paraméterekkel is különböző eredményeket adhatnak.

— A blokkstruktúra hierarchiája és a modulok importálásán alapuló hierarchiája keresztbe tesz egymásnak. Tulajdonképpen csak a programot, mint a legkülső blokkot lehet jól felbontani modulokra. Egy beagyazott alprogramból implicite az egész addigi deklarációs hierarchiára hivatkozni lehet, nem csak az importáltban felsorolt néhány modul néhány elemére.

ÖSSZEGZÉS NAGYMÉRETŰ TÖMBÖN C64-re

A nagyméretű tömbökénél még az egyszerű átlagszámítási feladatok is igen időigényesek. Az ilyen jellegű számításokat teszi egyszerűbbé és főleg gyorsabbá az alább ismertetendő gépi kódú program. Mégpedig úgy, hogy egy tetszőleges méretű, egydimenziós, valós tömbön összegzést hajt végre két megadott értékhatár között, és az eredményt egy valós változóba teszi le. A gépi kódú program működését a SYS(50688), AQ(0), X, Y, SZUMMA utasítással indítjuk el, ahol az 50688 a gépi kódú program belépési címe, az AQ(0) az a tömb, ahol összegezni kívánunk. A tömbnek az utasítás kiadása előtt már léteznie kell, különben hibajelzést kapunk (ugyanaz vonatkozik az X, Y és SZUMMA valós változóira is); indexként itt a tömbnek mindig a 0-dik elemét kell feltüntetni. Az X változó a tömbnek azt az elemét jelöli ki, ahonnan az összegzést el akarjuk kezdeni, míg az Y a tömbnek az az indexe, ameddig az összegzést folytatni kívánjuk; a SZUMMA változóban kapjuk vissza az összegzés végeredményét:

$$SZUMMA = \sum_{i=X}^Y AQ(i)$$

1. lista

```

1000 REM A GEPI KODU PROGRAM HIVASA:
1010 REM SYS(50688),AQ(0),X,Y,SUMMA
1015 :
1020 K=50688:V=50934:S=0
1030 FORI=K TO V
1040 READ X:POKEI,X:S=S+X:NEXT
1045 :
1050 DATA 32,253,174,32,139,176,133,251
1060 DATA 132,252,160,6,177,95,141,175
1070 DATA 2,136,177,95,141,176,2,32
1080 DATA 227,198,32,253,174,32,139,176
1090 DATA 141,173,2,140,174,2,58,173
1100 DATA 175,2,237,169,2,173,176,2
1110 DATA 237,170,2,176,5,162,14,76
1120 DATA 58,164,58,173,169,2,237,167
1130 DATA 2,141,175,2,173,170,2,237
1140 DATA 168,2,141,176,2,144,230,32
1150 DATA 155,198,32,148,198,173,176,2
1160 DATA 240,25,162,0,138,72,32,129
1170 DATA 198,32,141,198,32,106,184,104
1180 DATA 170,202,208,240,206,176,2,48
1190 DATA 7,208,233,174,175,2,208,228
1200 DATA 174,173,2,172,174,2,76,212
1210 DATA 187,24,165,251,105,5,133,251
1220 DATA 144,2,230,252,96,165,251,164
1230 DATA 252,76,140,186,165,251,164,252
1240 DATA 76,162,187,173,167,2,141,171
1250 DATA 2,173,168,2,141,172,2,14
1260 DATA 171,2,46,172,2,14,171,2
1270 DATA 46,172,2,24,173,171,2,109
1280 DATA 167,2,141,171,2,173,172,2
1290 DATA 109,168,2,141,172,2,24,165
1300 DATA 251,109,171,2,133,251,165,252
1310 DATA 109,172,2,133,252,96,32,253
1320 DATA 174,32,138,173,32,155,188,32
1330 DATA 173,177,96,32,214,198,141,168
1340 DATA 2,140,167,2,32,214,198,141
1350 DATA 170,2,140,169,2,96,0
1355 :
1360 IFS=30666THENPRINT"RENDEN!" :END
1370 PRINT"HI BAS ADATOK!" :END
    
```

2. lista

```

2
110: C600
120: C600 CHCOM * #AEFD ; VESSZO ELL.
130: C600 MENFAC = #BBA2 ; A(LO), Y(HI) 5 BAJT
140: C600 MENFC2 = #BABC ; A(LO), Y(HI) 5 BAJT
150: C600 FACHEM = #BBD4 ; X(LO), Y(HI) FAC-BOL
160: C600 ADDRESS = #BBD8 ; A ES -Y A SZAMON ALL
170: C600 FRMNUM = #BDBA ; KIFEJEZES ATVETELE
180: C600 FACINT = #BC0B ; FAC-BOL 2 BAJTOS EGESZ
190: C600 HIBAJL = #AA3A ; ILLEGAL QUANTITY
200: C600 FFLUS = #B06A ; FAC-FAC2+PAC
210: C600 TMBNEV = #9F ; TOMBNEV MUTATO
220: C600 MUT = #9FB ; TOMBEM MUTATO

240: 02A7 ; ** #02A7
;
260: 02AB XLO ** #+1
270: 02A9 XHI ** #+1
280: 02AA YLO ** #+1
290: 02AB YHI ** #+1
300: 02AC WLO ** #+1
310: 02AD WHI ** #+1
320: 02AE CIMLOW ** #+1
330: 02AF CIMHIG ** #+1
340: 02B1 CINTER ** #+2
;
360: C600 ** #C600
;
; SYS(50688),AQ(0),X,Y,SUMMA

400: C600 20 FD AE TOMBSU JSR CHCOM
410: C603 20 B8 B0 JSR ADDRESS
420: C606 05 FB STA MUT
430: C608 04 FC STY MUT+1
440: C60A 00 00 LDY #6
450: C60C 21 5F LDA (TMBNEV),Y
460: C60E 0D AF 02 STA CINTER
470: C611 00 DEY
480: C612 01 5F LDA (TMBNEV),Y
490: C614 0D B0 02 STA CINTER+1
500: C617 20 E3 C6 JSR XYINT
510: C61A 20 FD AE JSR CHCOM
520: C61D 20 B8 B0 JSR ADDRESS
530: C620 0D AF 02 STA CIMLOW
540: C623 0C AE 02 STY CIMHIG
550: C626 38 SEC
560: C627 AD AF 02 LDA CINTER
570: C62A ED A9 02 SBC YLO
580: C62D AD B0 02 LDA CINTER+1
590: C630 ED AA 02 SBC YHI
    
```


Amennyiben az X vagy az Y nagyobb, mint a több elemének a száma (DM), akkor_ILLEGAL QUANTITY ERROR_ hibajelzést kapunk. Ugyanez a helyzet all elő akkor is, ha az X nagyobb az Y-nál. Így X-re, Y-ra és DM-re a következő egyenlőtlenségnek kell teljesünie ahhoz, hogy ne kapjunk hibajelzést:

$$X < Y < DM$$

Ez a gépi kódú program még a nagyméretű tömböknél is elég gyorsan hajtódik végre, mintegy 10-20-szor gyorsabban, mint ha ugyanezt BASIC-ben csinálnánk.

Az átlag kiszámításához természetesen még egy osztást is el kell végeznünk:

$$ATLAV = SZUMMA / (Y - X + 1)$$

A nagyméretű tömbök dimenzionálásánál tekintettel kell lennünk azok helyigényére is. Amennyiben például a DIM W(1511) utasítással egy lebegőpontos számokat tartalmazó, 1512 elemű, egydimenziós tömböt hozunk létre, akkor az 1512 x 5 + 7 bajtot foglal le a C64 amúgy is szűkös memóriájából.

Az 1. listán a gépi kódú program BASIC-be-töltője, a 2. listán a gépi kódú program látható.

SZABÓ PÉTER PÁL

```

600: C633 B0 05          BCS OKE
610: C635 A2 0E          HIBA LDX #14
620: C637 4C 3A A4      JMP HIBAJL
630: C63A 38            SEC
640: C63B AD A9 02      LDA YLO
650: C63E ED A7 02      SBC XLO
660: C641 8D AF 02      STA CNTER
670: C644 AD AA 02      LDA YHI
680: C647 ED A8 02      SBC XHI
690: C64A 8D B0 02      STA CNTER+1
700: C64D 90 E6          BCC HIBA
710: C64F 20 9B C6      JSR MUTSX
720: C652 20 94 C6      JSR MRYFAC
730: C655 AD B0 02      LDA CNTER+1
740: C658 F0 19          BEQ ALSOEL
750: C65A A2 00          LDX #0
760: C65C 8A            TXA
770: C65D 4B            PHA
780: C65E 20 81 C6      JSR MUTPL5
790: C661 20 8D C6      JSR MRYFAC
800: C664 20 6A B8      JSR FPL5
810: C667 6B            PLA
820: C668 AA            TAX
830: C669 CA            DEX
840: C66A D0 F0          BNE SUMTMB
850: C66C CE B0 02      DEC CNTER+1
860: C66F 30 07          BMI VEGRE
870: C671 D0 E9          BNE SUMTMB
880: C673 AE AF 02      ALSOEL LDX CNTER
890: C676 D0 E4          BNE SUMTMB
900: C678 AE AD 02      VEGRE  LDX CIMLOW
910: C67B AC AE 02      LDY CIMHIG
920: C67E 4C D4 B8      JMP FACHEM
;
; SZUBRUTINOK
;
990: C681 18            MUTPL5 CLC
1000: C682 A5 FB          LDA MUT
1010: C684 69 05          ADC #5
1020: C686 85 F8          STA MUT
1030: C688 90 02          BCC VG
1040: C68A E6 FC          INC MUT+1
1050: C68C 60            RTS
;
;
1070: C68D A5 FB          MRYFAC2 LDA MUT
1080: C68F A4 FC          LDY MUT+1
1090: C691 4C 8C BA      JMP MEMFC2
    
```

```

1110: C694 A5 FB          MRYFAC LDA MUT
1120: C696 A4 FC          LDY MUT+1
1130: C698 4C A2 B8      JMP MEMFAC
;
1150: C69B AD A7 02      MUTSX  LDA XLO
1160: C69E 8D AB 02      STA WLO
1170: C6A1 AD A8 02      LDA XHI
1180: C6A4 8D AC 02      STA WHI
1190: C6A7 0E AB 02      ASL WLO
1200: C6AA 2E AC 02      ROL WHI
1210: C6AD 0E AB 02      ASL WLO
1220: C6B0 2E AC 02      ROL WHI
1230: C6B3 18            CLC
1240: C6B4 AD AB 02      LDA WLO
1250: C6B7 6D A7 02      ADC XLO
1260: C6BA 8D AB 02      STA WLO
1270: C6BD AD AC 02      LDA WHI
1280: C6C0 6D A8 02      ADC XHI
1290: C6C3 8D AC 02      STA WHI
1300: C6C6 18            CLC
1310: C6C7 A5 FB          LDA MUT
1320: C6C9 6D AB 02      ADC WLO
1330: C6CC 85 FB          STA MUT
1340: C6CE A5 FC          LDA MUT+1
1350: C6D0 6D AC 02      ADC WHI
1360: C6D3 85 FC          STA MUT+1
1370: C6D5 60            RTS
;
1390: C6D6 20 FD AE      BASINT JSR CHCOM
1400: C6D9 20 8A AD      JSR FRMNUM
1410: C6DC 20 9B BC      JSR FACINT
1420: C6DF 20 AD B1      JSR #B1AD
1430: C6E2 60            RTS
;
1450: C6E3 20 D6 C6      XYINT JSR BASINT
1460: C6E6 8D AB 02      STA XHI
1470: C6E9 8C A7 02      STY XLO
1480: C6EC 20 D6 C6      JSR BASINT
1490: C6EF 8D AA 02      STA YHI
1500: C6F2 8C A9 02      STY YLO
1510: C6F5 60            RTS
    
```

A Periféria Kiszövetkezet egyhetes szállítással ajánlja:

- P—XT 160 E Ft-tól + ÁFA
- P—AT 220 E Ft-tól + ÁFA
- FX—1000 nyomtató: 104 E Ft, lapadagoló 50 E Ft + ÁFA

— Igény szerinti konfigurációk összeállítása.

Cím: Budapest VII. Petyerdy u. 30.

Tel.: 213-588.

enterprise
['ent əpraiz]

1. vállalkozás,
vállalat
2. merészség,
vállalkozó
szellem

„SZINKRONBAN AZ IDŐVEL...”

ENTERPRISE
128 K

Hardver:		Fogyasztói ár:
EX-DOS	lemezvezérlő kártya műszaki leírás	
	IS-DOS rendszerlemez:	10 550,— Ft
ZX EMULATOR S-B-B	Spectrum emulátorkártya: System Bus Bridge (összekötő elem):	6 300,— Ft
SPEAK EASY	beszédsszintetizátor	1 470,— Ft
EP MOUSE	„egér” grafikus vezérlő:	3 750,— Ft
		4 000,— Ft

Kiegészítők, tartozékok:

RCA, DIN, EU-	monitorkábelek, szervizben történő át-	
RO — SCART, DEM	alakítással együtt:	1 248,— Ft
EPSON, DATACOD	Centronics printerkábel:	4 730,— Ft
OP, OEM		
DUAL, OEM	RS — 232 networkkábel: joystickadapter:	998,— Ft
	Enterprise-jelvények:	458,— Ft
	Enterprise-plakátok:	38,— Ft
		120,— Ft

Szoftver:

Játékprogram-ajánlások:	BATMAN, AIRWOLF, SUPER PIPE-LINE II., JAMMIN, DOT BREAKER-DOT COLLECTOR, „HIT SOFT” — ENTER 5 MAGIC BALL
Felhasználói programok:	LISP, CBM-MFT, ENTER-VIDEO, SIMON ASS, IS-FORTH, HI-SOFT DEV-PAC, WINDOWS — II., Személyi Jövedelem Adó Program — SZJA '88.

Szakkönyvek:

EXOS, EX-DOS, IS-DOS

Ismertkedés az Enterprise számítógéppel

Hetedhét Enterprise I.

CHIP SPECIAL (programpéldatár)

128 program az ENTERPRISE 128 k-n I—II. (megjelenik IX. hó)

ENTERPRISE BASIC Oktató I—II. (megjelenik XI. hó)

ENTERPRISE termékkatalógus (megjelenik XI. hó)

Újsi ajándékságaink:

RF hangmodulátor (szervizben történő átalakítással): VIII. hó
Enterprise számítógép — adatmagnó-örvédőhuzat: VIII. hó
Enterprise/Spectrum tápfeszültség ki-be kapcsoló: IX. hó
BNC-RCA monitorkábel (szervizben történő átalakítással): IX. hó
„SOFTCART” különféle felhasználói programok cartridge-ban: IX. hó
Enterprise dual RF mikrokapcsoló (TV — Computer): IX. hó
Spectrum Emulátor Kempston interface: X. hó
Active stereo booster (hangszóró doboz mini méretben): X. hó

Keresse fel ön is, kedves ENTERPRISE-vásárlónk a budapesti
ENTERPRISE MÁRKA OSZTÁLYAINKAT

az alábbi Centrum Áruházakban:

CENTRUM KISPESTI ÁRUHÁZ

Budapest XIX. ker.,

Kossuth tér 4—5.

CENTRUM OTTHON ÁRUHÁZ

Budapest VII. ker.,

Rákóczi út 74—78.

CENTRUM ÜTTÖRŐ ÁRUHÁZ

Budapest V. ker.,

Kossuth Lajos u. 9—11.

ENTERPRISE SOFT TOP LIST '88

1. MAGIC BALL
2. SORCERY
3. BAT MAN
4. AIR WOLF
5. ENTER BALL
6. WRIGGLER
7. NODES OF YESOD
8. NAUTILUS
9. SUPER PIPELINE II.
10. CHESS CYRUS II.



ENTERPRISE
COMPUTERS

Ügyintéző: Pásztor Tamás
Ügyiratszám: 48/1988

Mikroszámítógép Magazin Szerkesztősége
1027 Budapest, Fő u. 68.

Tárgy: Commodore 600/700 vásár

Tisztelt Felelős Szerkesztő Elvtárs!

Engedje meg, hogy az alábbiakban tájékoztassam akciónkról.

CBM-610 számítógép (128 k)	40 200,- Ft
CBM-8028 margarétafejes nyomtató	45 200,- Ft
SFD-1001 floppy (1 Mbyte)	56 000,- Ft
MD-1255H SAMSUNG adatmonitor	18 700,- Ft

(kiegészítők és tartozékok nélkül!)

Közel 20 féle felhasználói programcsomag ~~20~~-~~35~~%-os engedménnyel!

BASIC 4.0 COMPILER	UTILITY I.
ASSEMBLER COMPILER	UTILITY II.
EXBASIC	UTILITY III.
EXTROM BASIC+ASS	MULTI FILE
	TRANS' C64
HELP TEST COPY	SZM-1420 EMULATOR
MATH-LIB	HONEYWELL-BULL EMU- LATOR
MEMOBASIC	VT-51 EMULATOR
MS-"SZÖSZI" STANDARD	MULTI FILE TRANS' CENTRONICS
MS-"SZÖSZI" HUN.	PRINT CHAIN
MASTER-600	
MASTER-700	

Szakkönyvvásár (magyar, angol és német nyelvű) műszaki dokumentáció, könyv ~~50~~%-os árengedménnyel!

Ameddig a készlet tart!

CENTRUM KISPESTI ÁRUHÁZ

Budapest XIX. ker., Kossuth tér 4-5. Tel.: 275-066

CENTRUM OTTHON ÁRUHÁZ

Budapest-VII. ker., Rákóczi út 74/76. Tel.: 425-741

Budapest, 1988.



Szívélyes üdvözlettel:
CENTRUM ÁRUHÁZAK VÁLLALAT
SZÁMÍTÁSTECHNIKAI SZOLGÁLAT

Lapunk rendszeres olvasói tudják, hogy a nagy érdeklődés ellenére ritkán sikerül játék- vagy játékos jellegű programokat közlőnk.

Ennek az az oka, hogy az ilyen írásokat az oktató, informatív és elemző szakcikknek kiszorítják.

Ezúttal mégis közreadunk három különböző géptípusra írt programot, mert úgy véljük, hogy egyrészt

szórakoztatóak, másrészt programozástechnikai fogásokat lehetünk el belőlük.

Következzenek hát a programok!

KIVÉTEL ERŐ

A program beküldője — Károly János — arról ír, hogy az Expressz újságban 130 forintért kínálta valaki. Ezúttal közkinccsé tesszük. S hogy mit kezdhetünk vele? Hangeffektusok kíséretében megkérdezi születésnapj adatainkat, majd megmondja, hogy hány napot éltünk eddig. Kérésre kirajzolja szellemi, fizikai és érzelmi fázisainkat. A programot RUN-nal indíthatjuk. Ha újra kívánjuk futtatni, csak az I gombot szabad megnyomni, mert más billentyű lenyomása BASIC hidegindítást eredményez.

```
10 GRAPHIC3,1:SCNCLR:VOL6
20 COLOR4,2,7:COLOR0,2,7:COLOR1,1
30 COLOR2,3,3:COLOR3,4,5
40 SOUND1,7,20:SOUND2,1000,30
50 SOUNDS,1016,10
60 FORX=10TO150
70 DRAW,X,100TOX,101:NEXTX
80 FORY=10TO190
90 DRAW,10,YTO10,Y:NEXT
100 FORX=10TO150:I=23 :A=11.5
110 Y=100-(I*SIN(X/A))
120 DRAW3,X,Y
130 NEXT
140 DRAW1,90,50TO90,150
150 CHAR2,11,0,"B I O R I T M U S"
160 CHAR3,10,1,"_____ "
170 CHAR3,5,24,"BARMELY BILLENTYUVEL"
180 CHAR3,26,24,"INDITHATO"
190 GETKEYA#
200 SCNCLR
210 COLOR4,1:COLOR0,1,1
220 COLOR1,8,7:COLOR2,5,7
230 SOUND1,1000,30:SOUND2,7,20
240 SOUNDS,1016,10
250 CHAR,8,6,"A BEKERT ADATOK ALAPJAN"
260 CHAR,5,8,"KISZAMITOM,"
270 CHAR,17,8,"MAJD LERAJZOLOM A"
280 CHAR,8,10,"HAROM CIKLUST"
290 CHAR,22,10,"KULON-KULON."
300 CHAR,4,12,"AZ FUGGOLEGES VONALAK"
310 CHAR,25,12,"AZ AKTUALIS"
320 CHAR,14,14,"NAPOT JELZIK."
330 CHAR,11,20,"KEZDHEJUK ? (I/N)"
340 GETKEYA#
350 IFA#="I"THEN360:ELSE250
360 GRAPHIC0:SCNCLR:VOL7
370 COLOR4,4,4:COLOR0,4,4:COLOR1,9,2
380 COLOR3,6,2:SOUND1,345,10
390 SOUND1,685,15:SOUND1,854,20
400 SOUND1,881,30:SOUND2,911,30
410 SOUNDS,7,5
420 DIMH(12):FORI=1TO12:READH(I):NEXT
430 CLR:N=0:PRINT"AZON MAR";
440 PRINT"AKTUALIS DATUM(EV,HO,NAP)";
```

```
450 INPUT M,L,Z:PRINT:PRINT:PRINT
460 PRINT"SZULETESI DATUM(EV,HO,NAP)";
470 INPUT A,B,C:PRINT:PRINT:PRINT
480 PRINT"_____" EGY KIS TURELMET ";
490 PRINT"KEREK, SZAMOLOK."
500 A2=A:B2=B:C2=C
510 IF MCA THEN 430
520 IF M=A THEN IF L<B THEN 430
530 IFM=A AND L=B AND Z<C THEN 430
540 IFM=A ANDL=B THENN=Z-C:GOTO650
550 IFM>ATHEN590
560 K=C:GOSUB680:K=0
570 B=B+1:IFB<L THEN GOSUB680:GOTO570
580 N=N+Z:GOTO650
590 K=C:GOSUB680:K=0
600 B=B+1:IFB<12 THEN GOSUB680:GOTO600
610 A=A+1:IFACM THEN GOSUB740:GOTO610
620 B=0
630 B=B+1:IFB<L THEN GOSUB680:GOTO630
640 N=N+Z
650 GOTO760
660 I=A/4:IFI#4=A THENI=0:ELSEI=1
670 RETURN
680 DATA 31,28,31,30,31,30
690 DATA 31,31,30,31,30,31
700 IFB=2 THENJ=A:GOSUB660
710 IFI=0 THENN=N+29-K:RETURN
720 N=N+28-K:RETURN
730 N=N+H(B)-K:RETURN
740 GOSUB660:IF I=0 THEN N=N+366:RETURN
750 N=N+365:RETURN
760 N=N+1
770 P=23:H=INT(N/P):F=N-P#H
780 U=28:H=INT(N/U):E=N-U#H
790 R=33:H=INT(N/R):S=N-R#H
800 PRINT"J":COLOR4,1:COLOR0,1
810 COLOR1,2,7
820 PRINT"AZON BIORITMUSA"
830 PRINT"_____ ON MAR :N";
840 PRINT"-IK NAPJA EL."
850 PRINT"_____ FIZIKAI CIKLUS : ";
860 PRINT INT(N/P)+1;"-IK :F;"-IK NAP"
870 PRINT"_____ ERZELMI CIKLUS : ";
880 PRINT INT(N/U)+1;"-IK :E;"-IK NAP"
890 PRINT"_____ SZELLEMI CIKLUS : ";
```

SÍTI A SZABÁLYT!

Commodore Plus/4

BIORITMUS

```
900 PRINT INT(N/R)+1;"-IK ";S;"-IK NAP"
910 PRINT"XXXXXXXXXXXXXXXXXXXX";
920 PRINT"MEGRAJZOLJAM ? (I/N)"
930 SOUND1,7,5
940 GETKEYA$
950 IFA$="I"THEN960:ELSE800
960 SCNCLR
970 GRAPHIC3,1
980 SOUND1,F*30,50:COLOR4,7,2
990 COLOR0,7,2:COLOR1,2,7:VOL4
1000 CHAR,5,3,"FIZIKAI CIKLUS"
1010 FORX=0TO39
1020 CHAR,X,11,"_"
1030 CHAR,X,12,"L":NEXT
1040 FORX=0TO156
1050 DRAW,X,100-29*SIN(X/14.5)
1060 SOUND1,X*2,1:NEXT
1070 FORY=50TO160
1080 F=F:Q=F*4
1090 DRAW,Q,YTOQ,Y:NEXT
1100 CHAR,0,14,"1":CHAR,9,14,"10"
1110 CHAR,19,14,"20":CHAR,29,14,"30"
1120 CHAR,4,10,"5":CHAR,14,10,"15"
1130 CHAR,24,10,"25":CHAR,34,10,"35"
1140 CHAR,F,21,"MAI NAP"
1150 CHAR,6,23,"FOLYTATAS BARMELYIK"
1160 CHAR,26,23,"GOMBBAL"
1170 GETKEYA$
1180 IFA$=""THEN1190
1190 SCNCLR:VOLS
1200 SOUND1,E*30,50:COLOR4,3,5
1210 COLOR0,3,5:COLOR1,5,2
1220 CHAR,5,3,"ERZELMI CIKLUS"
1230 FORX=0TO39
1240 CHAR,X,11,"_"
1250 CHAR,X,12,"L":NEXT
1260 FORX=0TO156
1270 DRAW,X,100-35*SIN(X/17.5)
1280 SOUND1,X*3,1:NEXT
1290 FORY=50TO160
1300 E=E:W=E*4
1310 DRAW,W,YTOW,Y:NEXT
1320 CHAR,0,14,"1":CHAR,9,14,"10"
1330 CHAR,19,14,"20":CHAR,29,14,"30"
1340 CHAR,4,10,"5":CHAR,14,10,"15"
1350 CHAR,24,10,"25":CHAR,34,10,"35"
1360 CHAR,E,21,"MAI NAP"
1370 CHAR,6,23,"FOLYTATAS BARMELYIK"
1380 CHAR,26,23,"GOMBBAL"
1390 GETKEYA$
1400 IFA$=""THEN1410
1410 SCNCLR:VOL6
1420 SOUND1,S*30,50:COLOR4,4,5
1430 COLOR0,4,5:COLOR1,4,2
1440 CHAR,5,3,"SZELLEMI CIKLUS"
1450 FORX=0TO39
1460 CHAR,X,11,"_"
1470 CHAR,X,12,"L":NEXT
1480 FORX=0TO156
1490 DRAW,X,100-41.5*SIN(X/20.75)
1500 SOUND1,X*4,1:NEXT
1510 FORY=50TO160
1520 S=S:Z=S*4
1530 DRAW,Z,YTOZ,Y:NEXT
1540 IFS=F OR S=E THEN
1550 CHAR,0,14,"1":CHAR,9,14,"10"
1560 CHAR,19,14,"20":CHAR,29,14,"30"
1570 CHAR,4,10,"5":CHAR,14,10,"15"
1580 CHAR,24,10,"25":CHAR,34,10,"35"
1590 CHAR,S,21,"MAI NAP"
1600 CHAR,6,23,"FOLYTATAS ..... "
1610 GETKEYA$
1620 IFA$=""THEN1630
1630 GRAPHIC0:SCNCLR:SOUND1,169,10
1640 SOUND1,596,10:SOUND1,810,10
1650 SOUND2,917,10:SOUND1,939,15
1660 SOUND2,953,20:COLOR1,1
1670 COLOR4,4,6:COLOR0,2,7:COLOR1,1
1680 CHAR,10,5,"REMELEM SEGITETTEM."
1690 CHAR,9,10,"ALLOK RENDELKEZESERE"
1700 CHAR,14,13,"MASKOR IS"
1710 CHAR,7,20,"AKAR MEG VALAKI"
1720 CHAR,23,20,"JATSZANI ?"
1730 CHAR,18,22,"I/N"
1740 GETKEYA$
1750 IFA$="I"THEN360
1760 IFA$<"N"THEN1740
1770 SYS32768
```

JÁTÉK

A program és a játék lényege: a játékosoknak a képernyőn megjelenő 10 betűből adott idő (90 másodperc) alatt értelmes szavakat kell kirakniuk úgy, hogy minél több pontot szerezzenek.

A betűk között megjelenő csillag helyére tetszőleges betűt lehet írni. A játékkal — ha csak alapgépünk van —, egyszerre legtöbbször hatan szórakozhatnak.

A program bemenő adatai

- A 110—120-as sorban XS és X változó a játékosok számát tárolja. X min.=1, X max.=6.
- A 140—170-es sorban NES(i) az i-edik játékos nevét tartalmazza. A játékosok neve a képernyőformátum miatt hat karakterben van maximálva. Ha nem adunk meg nevet, akkor ezt a gép teszi meg (160—165).
- Az 550—570-es sorban SZS(i) az i-edik játékos által beadott szót tartalmazza. A szó hossza 11 karakterre lehatárolt, azért, hogy a tárterület jobban kihasználható legyen.

Egyéb változók

- E változó a képernyőszínek címe.
- M(i,j) tömbváltozó minden elemét 0 értékre állítjuk az 520-as sorban. Ha az i-edik játékos a képernyőn megjelenő j-edik karaktert felhasználta a kirakott szóban, akkor az ennek megfelelő indexű tömbváltozót 1 értékre állítjuk be a 800-as sorban. Ily módon ellenőrizhetjük, hogy mely karaktereket használtak fel, illetve nem használtak-e olyan karaktert, ami nem szerepelt a képernyőn megadottak között.
- SU(i) az i-edik játékos által szerzett összpontszám értéke.
- EGS, TR\$, KE\$ gyakrabban alkalmazott szövegtipusú konstansok.
- V változó tartalmazza a ritkábban használt betűk számát (Q, X, Y, W) a 270-es sorban.
- W változó tartalmazza a magánhangzók számát.
- Q változóban vesszük figyelembe, hogy ne legyen túl sok azonos karakter (a 360-as sorban).
Megjegyzés: a V, W, Q változók értéke a képernyőn megjelenő 10 karakterre vonatkozik.
- BE(i) az i-edik megjelenő karakter ASCII kódja, a 240-es sorban.
- PO(i) az i-edik karakter pontértéke a 250-es sorban.
- K változó a BE(i) értéket tartalmazza a karakterek ellenőrzésekor a 250-es sorban.
- TY változó a játék szempontjából a nullidőt tárolja TI változó alapján a 410-es sorban.
- K változó tartalmazza a gondolkodási idő alatt lenyomott billentyű kódját a 450-es sorban. Ezzel figyeljük az F3 funkcióbillentyűt, amit lenyomva, a játékidő letelte előtt beadhatjuk a szavakat.
- ER(i) változó az utolsó menüben alkotott szóért kapott pontot tartalmazza a 800-as sorban.
- SX változó 1 értéket vesz fel, ha a kirakott szó hosszával van probléma (rövid szó, hosszú szó, nem írunk be semmit, a 615, 625, 635-ös sorokban).
- SS, ZS, DFS a gép által küldött üzeneteket tartalmazza a 600-as és az 1000-es sorok között.

```

10 E=36879:DIMM(6,10)
20 FORI=1TO6:SU(I)=0:NEXTI=POKEE,26
30 EGS="IIII *****"
40 TR$="IIIIIIIIIIIIIIIIIIII"
50 KE$="IIIIIIIIIIIIIIIIIIII"
60 PRINT"J"TAB(47)"JATEK"
70 PRINTTR$"KAKTUSZ SOFTWARE"87"
80 PRINT"BY:LAISZ LASZLO"
100 GOSUB2000:POKE198,0:POKEE,29
110 PRINTKE$"JATEKOSOK SZAMA":INPUTX$
120 X=VAL(X$):IFX<1ORX>6THEN110
130 PRINTKE$"JATEKOSOK NEVE:"
140 FORI=1TOX:PRINT"J"CHR$(157)".JATEKOS:
150 INPUTNE$(I):NE$(I)=LEFT$(NE$(I),6)
155 IFLen(NE$(I))>6THEN170
160 NE$(I)=RIGHT$(STR$(I),1)
165 NE$(I)=NE$(I)+"MAN"
170 NEXTI
180 PRINTKE$TR$
190 PRINT"SZAVAK BEADASA"
210 POKEE,28:V=RNDC(-TI)
220 V=0:Q=0:N=0
230 POKE781,4:POKE782,65:SYS60047
240 FORJ=1TO10:BE(J)=INT(RND(1)*28)+63
250 K=BE(J):PO(J)=INT(RND(1)*9)+1
260 IFK=65ORK=69ORK=73ORK=79ORK=85THENW=W+1
270 IFK=81ORK=87ORK=88ORK=89THENV=V+1
280 IFK<65THENBE(J)=42
290 NEXTJ
300 FORJ=1TO10
310 PRINT"TAB(J*2+87)CHR$(BE(J))"
320 PRINTTAB(J*2+20)PO(J)
330 NEXTJ
340 FORI=1TO9:J=I+1
350 FORK=JTO10
360 IFBE(I)=BE(K)THENQ=Q+1
370 NEXTK,1
380 IFK<2ORW>6THEN210
390 IFV>2THEN210
400 IFQ>2THEN210
410 POKEE,26:TY=TI
420 PRINTTR$"JATEKIDO:"
430 PRINTINT(91-(TI-TY)/60)CHR$(157)" ";
440 PRINTTAB(16)"SEC"
450 K=PEEK(197)
460 IFK=47THEN500
480 IF(TI-TY)<5400THEN420
500 PRINTTR$"A JATEKIDO LETELT!"
510 FORN=1TOX:SZ$(N)=""
520 FORM=1TO10:MN(M)=0
530 NEXTM,N:GOSUB2000:POKEE,29
540 PRINTKE$"KEREM A SZAVAKAT":POKE198,0
550 FORN=1TOX:PRINT"J"SZ$(N)TAB(8);
560 INPUTSZ$(N):SZ$(N)=LEFT$(SZ$(N),11)
570 NEXTN:GOSUB2000
580 FORN=1TOX:A=1:SX=0:ER(N)=0
590 S$=""Z$=""DF$=""

```

a B E Ü K e L

- LS változó tartalmazza a vizsgálandó szó hosszát a 600-as sorban.
- B\$, C\$: a kirakott és a megjelenő betűk összehasonlításánál használjuk a 660–680-as sorokban.
- AS a kirakott szó következő betűjének ASCII kódja a 700-as sorban.

A program leírása

- 10–20 a tárfoglalás, összpontszám vektor nullázása, a képernyő színbeállítása.
- 30–100 a szövegkonstansok megadása, a bejelentkező szöveg kiírása. Hangeffektus. A billentyűzetpuffer nullázása.
- 110–170 a játékosok számának és nevének bekérése.
- 180–210 a szövegkonstansok kiírása, véletlenszám-generátor beállítása.
- 230 a képernyő negyedik sorától 65 karakter törlése ROM-rutin segítségével.
- 240–250 a betűk és pontok generálása, vektorokba töltése.
- 260–280 a magánhangzók és ritkán használt betűk számlálása, csillagkarakter beállítása.
- 300–330 a betűk és pontok kiírása.
- 340–370 a többször előforduló karakterek számlálása a Q változóban.
- 380–400 a kiválasztott 10 betű ellenőrzése Q, V, W szerint.
- 410–480 a játékidő kiírása és vizsgálata. Az F3 billentyű lenyomásának ellenőrzése.
- 510–570 az SZS(i) vektor és M(i,j) tömb nullázása. A kidöntött szavak bekérése és méretének maximálása.
- 580–1100 a beadott szavak karakterenkénti kiemlézése.
- 605 ha nem lett beírva szó, SX = 1
- 620 ha egykarakteres szó lett beírva, SX = 1
- 630 ha a szó hosszabb 10 karakternél, SX = 1
- 650–690 karakterenkénti összehasonlítás
- 700–710 nem betűkarakter szerepel-e a szóban?
- 720–740 csillagkarakter felhasználása
- 800 betűfelhasználás beállítása, pontösszegzés
- 1000–1100 a beadott szavak elemzése a képernyőn
- 1020 a hibás karakter inverzben jelenik meg
- 1110 a kapott pontok hozzáadása az előző fordulókban kapott pontösszeghez.
- 1200–1310 tájékoztató táblázat a játék állásáról. Új játék lehetőségének megteremtése.
- 2000 a játék során hallható hangeffektus.

LÁISZ LÁSZLÓ

```

600 LS=LEN(SZ$(N))
605 IFLS>=1THEN620
610 S$=" NEM OLVASHATO!"
615 SZ$(N)=CHR$(0):SX=1:GOTO640
620 IFLS=2THEN630
625 SX=1:S$=" TUL ROVID!"
630 IFLS<=10THEN640
635 SX=1:S$=" TUL HOSSZU!"
640 IFSX<1THEN650
645 Z$=" NEM FOGADOM EL!":GOTO1000
650 FORA=1TOLS:FORB=1TO10
660 B$=MID$(SZ$(N),A,1)
670 C$=CHR$(BE(B))
680 IFB$=C$ANDM(N,B)=0THEN800
690 NEXTB
700 AS=ASC(MID$(SZ$(N),A,1))
710 IFAS<65ORAS>90THEN750
720 FORB=1TO10
730 IFBE(B)=42ANDM(N,B)=0THEN800
740 NEXTB
750 DF$=" BUTASAGOT IRTAL BE!"
760 Z$=" NEM FOGADOM EL!"
780 SX=1:ER(N)=0
790 GOTO1000
800 M(N,B)=1:ER(N)=ER(N)+PO(B)
810 NEXTA
1000 PRINTC$EG$" JATEKOS NEVE: "NE$(N)
1010 PRINTEG$" A SZO: "SZ$(N)
1020 PRINTTAB(A+7)"MID$(SZ$(N),A,1)"
1030 IFSX<1THEN1040
1035 PRINT"DF$"DF$:PRINT"Z$":PRINT"Z$"Z$
1040 PRINTEG$" PONTERTEK: "ER(N)
1050 PRINTEG$" GOSUB2000
1100 NEXTN
1110 FORN=1TOX:SU(N)=SU(N)+ER(N):NEXTN
1200 PRINTKE$"M * * * A JATEK ALLASA * * * "
1220 FORN=1TOX:PRINT" "NE$(N);
1230 FORI=1TO15-LEN(NE$(N)):PRINT". ";NEXTI
1240 PRINTMID$(STR$(SU(N)),2):NEXTN
1250 PRINTEG$
1260 PRINT"MEHET TOVABB ? (I/N)"
1270 GOSUB2000:POKE198,0
1280 GETA$:IFA$=" "THEN1280
1290 IFA$="I" THEN180
1300 IFA$="N" THENEND
1310 GOTO1280
2000 POKEE-1,15:FORL=1TO10:POKEE-4,200
2010 FORM=1TO10:NEXT:POKEE-4,0
2020 FORM=1TO300:NEXT:POKEE-5,200
2030 FORM=1TO10:NEXT:POKEE-5,0
2040 FORM=1TO300:NEXT:NEXT:RETURN

```

FORTUNA A SZÁMÍTÓGÉPEN

Ez a program a lottózóknak segítség, ha rendszeresen ugyanazzal a variációval, de változó számokkal játszanak. Mentésit a kódszámok szerinti behelyettesítéstől. 14 szám 14 szelvényes variációjára készült, de természetesen átírható bármilyen másikra. Az eredeti variáció 14 szelvényes, de a program lehetőséget ad kétszer annyi (28) szelvény kitöltésére is, mivel a variációt megdupláztuk (480-530-as, 560-as sorok). Így persze a nyerési esély is kétszeres.

A variációt a VAR tömbbe kell betölteni, majd a gép elvégzi a DATA sorokból. A megjátszani kívánt számok a SZAM tömbbe kerülnek. Az utolsó szám beadása után következik a behelyettesítés, majd hangjelzést kapunk (620-as sor). A szelvényre írandó számokat a SZELS tömbbe tölti a gép, majd szelvényenként kiírja.

Szerencsés lottózást!

KATONA LÁSZLÓ



MD
Vállalkozási és Kereskedelmi Kft.
1112 Budapest, Budaörsi út 42.
Telefon: 666-617, 666-618, 666-056
Telex: 22-7676 md h

Az MD Kft. legújabb ajánlata!

Működéképtelen Winchesterét átvesszük,
reklámsáron újat adunk!

SEGATE ST 225	25,6 Mb	65 ms	35.000,- Ft
ST 238	38,4 Mb	65 ms	39.000,- Ft
ST 251	51,2 Mb	40 ms	69.000,- Ft
ST 251-1	51,2 Mb	30 ms	79.000,- Ft
ST 4096	85 Mb	30 ms	129.000,- Ft

Áraink ÁFA nélkül értendők!

MD-GUTOR szünetmentes áramforrás
csúcstechnológia képviselője
Teljesítmény: 0,5-10 kVA
Szállítás: raktárról
Pld. DP 15 1,5 kVA 590.000,- Ft

```

100 PROGRAM LOTTO 14/14
101 !
102 ! ENTERPRISE 128
103 !
110 DIM VAR(14,5),SZAM(14)
120 STRING SZEL$(1 TO 28)*15
130 FOR V1=1 TO 14
140   FOR V2=1 TO 5
150     READ VAR(V1,V2)
160   NEXT V2
170 NEXT V1
200 DATA 1,2,3,5,8,1,2,4,7,9
210 DATA 1,3,6,9,10,1,4,10,11,12
220 DATA 1,8,11,13,14,2,3,4,6,14
230 DATA 2,5,9,10,11,2,10,12,13,14
240 DATA 3,4,5,7,13,3,9,11,12,13
250 DATA 4,5,6,8,12,5,6,7,11,14
260 DATA 6,7,8,10,13,7,8,9,12,14
300 CLEAR TEXT
310 PRINT AT 3,1:"Kérem a ";
313 PRINT "megjátszani kívánt ";
316 PRINT "számokat kodszaám ";
320 PRINT "szerinti sorrendben."
330 FOR S=1 TO 14
340   PRINT AT 10,3:"Kodszaám: ";S
350   INPUT AT 10,17:SZAM(S)
360   PRINT AT 10,19:" "
370 NEXT S
380 CLEAR TEXT
390 PRINT AT 12,7:"Folyik a ";
391 PRINT "behelyettesítés!"
400 FOR V1=1 TO 14
410   LET AA$=""
420   LET BB$=""
430   FOR S=1 TO 5
440     LET A=SZAM(VAR(V1,S))
450     LET A$=STR$(A)&" "
460     IF A<10 THEN LET A$=" "&A$
470     LET AA$=AA$&A$
480     LET V2=VAR(V1,S)+7
490     IF V2>14 THEN LET V2=V2-14
500     LET A=SZAM(V2)
510     LET B$=STR$(A)&" "
520     IF A<10 THEN LET B$=" "&B$
530     LET BB$=BB$&B$
540   NEXT S
550   LET SZEL$(V1)=AA$
560   LET SZEL$(V1+14)=BB$
570 NEXT V1
600 CLEAR TEXT
610 PRINT AT 5,17:"Kész."
620 PING
630 FOR V1=1 TO 28
640   PRINT AT 22,20:"Tovább --> t"
650   IF INKEY$("<")="t" THEN 650
660   CLEAR TEXT
670   PRINT AT 10,1:V1;". Szelvény: ";
680   PRINT SZEL$(V1)
690 NEXT V1
700 END
9999 REM KL SU/SUFNI

```




OKTATÁSI IRODÁJA az 1988/89-es tanévben a következő mikrogépes tanfolyamokat indítja:

MIKROGÉPES PROGRAMOZÁSI NYELVEK

A tanfolyam(ok) megnevezése	Tartama (nap)	Ára (Ft)	Időpont(ok) 1988/1989
BASIC	5	7450	szeptember 19—23. december 12—16. február 20—24.
TURBO BASIC (kezdőknek)	5	7450	október 17—21. november 21—25. január 16—20.
TURBO PASCAL I. (kezdő, V3.0.)	5	7450	szeptember 19—23. november 21—25. január 16—20.
TURBO PASCAL II. (V4.0.) ÚJ!	5	7450	nov. 28.—dec. 02. január 23—27.
Balatonkenesén	5	9600	szeptember 26—30. október 17—21.
PROLOG	5	7450	október 17—21. november 21—25. december 19—23. január 16—20.
Professional COBOL	5	7450	nov.28.—dec. 02. január 23—27.
Balatonkenesén	5	9600	október 10—14.
FORTRAN 77	5	7450	október 03—07. január 16—20.
Programtervezés JACKSON-módszerrel	5	7450	október 17—21. nov. 28.—dec. 02. február 20—24.
Programozási módszertan PC—kre	5	7450	október 17—21. november 21—25. február 20—24.
Tanfolyamszervező: Berencsy Ildikó, tel.: 853-111/229, 220. Tanfolyamfelelős: Hont László, tel.: 853-111/234, 233.			

MIKROGÉPES RENDSZERSOFTVEREK

A tanfolyam(ok) megnevezése	Tartama (nap)	Ára (Ft)	Időpont(ok) 1988/1989
MS-DOS operációs rendszer	5	7450	október 10—14. november 21—26. december 05—09. január 02—06. szeptember 12—16.
Balatonkenesén	5	9600	szeptember 12—16.
MS-DOS rendszerprogramozóknak ÚJ!	5	7450	december 12—16.
Balatonkenesén	5	9600	szeptember 19—23.
UNIX—XENIX	5	7450	október 03—07. február 06—10.
UNIX rendszerprogramozóknak ÚJ!	5	7450	február 13—17.
Balatonkenesén	5	9600	okt. 31.—nov. 04.
IBM OS/2 operációs rendszer ÚJ!	5	7450	nov. 28.—dec. 02. január 16—20.
LAN—TAP	5	7450	november 14—18.
NOVELL lokális hálózat	5	7450	november 14—18. január 09—13. február 06—10. október 03—07. október 03—07. január 02—06. október 17—21. január 16—20. október 10—14. január 09—13. október 24—28. január 23—27.
Balatonkenesén	5	9600	október 03—07. október 03—07.
ASSEMBLER I.	5	7450	október 03—07. október 03—07. január 02—06. október 17—21. január 16—20.
ASSEMBLER II.	5	7450	október 10—14. január 09—13. október 24—28. január 23—27.
C programozási nyelv I.	5	7450	október 10—14. január 09—13. október 24—28. január 23—27.
C programozási nyelv II.	5	7450	október 10—14. január 09—13. október 24—28. január 23—27.
Programfejlesztés „C”-ben ÚJ!	5	7450	február 06—10. okt. 31.—nov. 04.
Balatonkenesén	5	9600	február 06—10. okt. 31.—nov. 04.
Tanfolyamszervező: Nagy Erzsébet, tel.: 853-111/220, 229 Tanfolyamfelelős: Antoni Alfonz, tel.: 853-111/231 Zöld Sándor (Balatonkenesén), tel.: 853-111/231			

MIKROGÉPES PROGRAMCSOMAGOK

A tanfolyam(ok) megnevezése: (SZKÓD)	Tartama (nap)	Ára (Ft)	Időpont(ok) 1988/1989
dBASE III. PLUS (kezdőknek) (D. E.)	5	7450	szeptember 26—30. november 21—25. január 23—27.
dBASE III. PLUS programozás (B. I.)	5	7450	szeptember 19—23. nov. 28.—dec. 02. február 20—27.
Balatonkenesén	5	9600	október 24—28.
FOXBASE (kezdőknek) (D. E.) ÚJ	5	7450	október 17—21. december 19—23. február 19—27.
FOXBASE dBASE III. Plus Új előléte (D. E.)	3	4450	szeptember 21—23. november 09—11. január 04—06. november 08—10. jelentkezéstől függ
dACCESS III. (D. E.)	3	4450	szeptember 26—28.
PROGRESS (B. I.) ÚJ	3	7450	december 05—07. jan. 30.—febr. 01.
FREETEXT szöveges Újadatbáziskezelő (B. I.)	3	4900	november 08—10. február 20—24. december 12—16.
DATAFLX (N. E.) ÚJ	10	14900	szeptember 12—14. február 01—03. november 08—10. szeptember 26—30.
BECKER—BASE (D. E.) ÚJ	5	7450	január 23—27. november 08—09. december 12—16. szeptember 19—23.
Szövegfeldolgozó rendszerek (N. E.)	5	7450	november 14—18. január 02—06.
XV—writer (N. E.) ÚJ	3	4450	február 14—18.
MICROSOFT WORD (D. E.) ÚJ	3	4450	február 20—24. december 12—16.
WORDSTAR (D. E.)	3	4450	szeptember 12—14. február 01—03.
MULTIPLAN (D. E.)	3	7450	november 08—10. szeptember 26—30.
PLUSCALC (D. E.)	2	2980	január 23—27.
QUATRO (D. E.) ÚJ	5	7450	december 12—16. szeptember 19—23.
LOTUS 1—2—3 (D. E.)	5	7450	november 14—18. január 16—20.
SYMPHONY (kezdő) (D. E.)	5	7450	október 10—14. december 12—16. október 17—21.
SYMPHONY programozás ÚJ (D. E.)	5	7450	december 19—23. október 17—21.
FRAMEWORK II. (D. E.)	5	7450	december 19—23. október 03—07. január 02—06. december 05—09.
OPEN ACCESS (D. E.)	5	7450	október 03—07. január 02—06. december 05—09.
Számítógépes grafika ÚJ (N. E.)	5	7450	szeptember 26—30. december 12—16.
AUTOCAD grafikai programcsomag használata (B. I.)	5	7450	február 20—24. szeptember 26—30. december 12—16.
SMART—WORK használata ÚJ (B. I.)	5	7450	február 20—24. október 10—14. október 17—21. december 19—23.
Szakértői rendszerek Új fejlesztése GENIEYS-szel száloda nincs! (B. I.)	5	7450	október 03—07. február 20—24. október 24—26. december 05—07.
Statikai alkalmazások (B. I.)	3	4900	október 24—26. október 24—26. szeptember 12—16. január 21—25. január 09—13.
Venture Desktop Publisher ÚJ (D. E.)	5	7450	október 24—26. október 24—26. szeptember 12—16. január 21—25. január 09—13.

Ebben a táblázatban a szervezőket kódolva adtuk meg
SZKÓD = (B. I.) esetén:
 Tanfolyamszervező: Berencsy Ildikó, tel.: 853-111/229, 220
 Tanfolyamfelelős: Hont László, tel.: 853-111/234, 233
SZKÓD = (D. E.) esetén:
 Tanfolyamszervező: Dembrocki Erzsébet, tel.: 853-111/154, 237
 Tanfolyamfelelős: Gerő Judit, tel.: 853-111/238
SZKÓD = (N. E.) esetén:
 Tanfolyamszervező: Nagy Erzsébet, tel.: 853-111/229, 220
 Tanfolyamfelelős: Antoni Alfonz, tel.: 853-111/231
 Megjegyzés: A Balatonkenesén tanfolyamokat időközben tartjuk.



Cím:
 Számítástechnika-alkalmazási Vállalat
 OKTATÁSI IRODA
 Budapest XI., Szakasits Árpád út 68.
 Levélcím: Budapest 112. Pf. 146. 1502
 Telex: 22-4498

Az elmúlt években többféle számítógéppel ismerkedtünk meg. Ezek mindegyikének van valamilyen sajátossága a programtárolásban is. A HT és a Commodore ilyenkor a sorszám előtt két bájtól tárolja a következő program sor kezdetének címét (láncolás). A Sinclair a sorszám után két bájtól tárolja az aktuális program sor hosszát.

Az Enterprise-nál is hasonló elvek alapján lehet a programot tárolni, de BASIC-je, operációs rendszere egy sor eltérő megoldást tartalmaz a korábbi házi számítógépekhez képest. Természetesen az ezeknél felhalmozódott módszereket is felhasználhatjuk a memória kiismeréséhez.

Az „átszállóknak” könnyű. Akik most kezdik, azoknak segítünk. A legegyszerűbb, alapértelmezésű Enterprise használatánál a BASIC nyelvű program tárolása a 4827 Dec címen kezdődik. A program sorok a memóriában kódolt formában helyezkednek el.

Az 1. lista kiírja a képernyőre a saját memóriában való elhelyezkedését. A futás eredményeinek egy részét mutatja a 2. lista. Az ered-

sor beírva vizsgálhatjuk annak memóriában való elhelyezkedését. Lehet, hogy a 42-es sorba is kell írni valamit, mert az interpreter nemcsak a sor beírásakor végez bizonyos ellenőrzéseket, hanem a futtatáskor ellenőrzi például a FOR-NEXT szerkezeteket is.

SZ. LUKÁCS JÁNOS

```

10 REM proba 1
20 FOR T=4827 TO 4911
30 LET P=PEEK(T)
40 PRINT T;P,CHR$(P)
50 NEXT T
    
```

1. lista

```

100 REM kulcsszo tabla
110 OUT 177,5
120 LET N=0
130 FOR J=30124 TO 31125
140 LET L=PEEK(J)
150 PRINT N,
160 LET N=N+1
170 FOR K=J+1 TO J+L
180 PRINT CHR$(PEEK(K));
190 NEXT
200 PRINT
210 LET J=J+L+5
220 NEXT
    
```

3. lista

```

10 REM ....RAM. tapogato
20 GOTD 100
30 !
40 REM ide a vizsgalando basic sor
50 !
90 LET H=PEEK(4869)-1
100 LET H=PEEK(4869)-1
110 FOR L=4869 TO 4869+H
120 LET J=PEEK(L)
130 PRINT L;J,
140 IF J>32 AND J<160 THEN PRINT C
HR$(J);
150 PRINT
160 NEXT
    
```

4. lista

A PROGRAMTÁROLÁS

á b é c é j e

mény alapján látható, hogy a program sor hossza egy bájtól tárolódik, tehát egy program sor maximum 255 bájt lehet, a legrövidebb program sor pedig 7 bájt hosszú. A program sor száma a következő két bájt. Az itt tárolható maximális érték 9999 lehet.

Minden BASIC sor kulcsszóval kezdődik. Az interpreter program listázásánál (a program szerkezetének jobb áttekinthetőségéért) a FOR-NEXT, IF-THEN-ELSE, DO-LOOP szerkezeteknél tagolt kiírást tesz lehetővé. A skatulyázás mélységét – egy bizonyos határig az egyre beljebb kezdett kiírást – a sorszám utáni bájt jelzi. A kiírások egyre beljebb kezdésének mértéke a TEXT üzemmódtól függ. FOR-NEXT ciklusból 128 ágyazható egymásba.

A következő két bájt jelzi, hogy kulcsszó következik, majd az a sorszám, ahol a ROM-ban levő táblában az adott kulcsszó elhelyezkedik.

A kulcsszótáblát a 3. listával írhatjuk ki, és láthatjuk, hogy azok abc-sorrendben következnek egymás után. A program sorban levő számok ábrázolása igen különböző lehet. Erről később lesz szó. A program sor végét 0, a program végét a sorvégjelző θ után még egy θ bájt jelzi.

Azt, hogy a programokat hogyan tárolhatjuk, a 4. lista alkalmazásával tovább vizsgálhatjuk. A 40-es sorban mindig más-más program-

2. lista

4842	19	:	a sor hossza
4843	20	:	a sorszám
4844	0	:	ket byte-on
4845	0	:	skatulyazas jelzo
4846	96	:	kulcsszo bevezeto
4847	31	:	kulcsszo token
4848	33	!	egybetus valtozonev kov.
4849	84	T	a valtozo neve
4850	19	:	= token
4851	194	:	egesz szam kov.
4852	219	:	a kezdoertek
4853	18	:	18*256+219
4854	34	"	ket betus szoveg kov.
4855	84	T	:
4856	79	O	:
4857	194	:	egesz szam kov.
4858	47	/	a vegertek
4859	19	:	:
4860	0	:	sor vege
4861	19	:	a sor hossza
4862	30	:	a sorszám
4863	0	:	ket byte-on
4864	1	:	skatulyazas jelzo
4865	96	:	kulcsszo bevezeto
4866	40	(kulcsszo token
4867	33	!	egybetus valtozonev kov.

Romvári Gábor a HCC Enterprise Klub vezetője. A klub 1987 szeptembere óta működik a VSZM Közösségi Házban, Nagy István létesítményvezető támogatásával. 1988 májusában már háromszázötven tagja volt.

M. M. Miért éppen az Enterprise Klub létrehozásának gondolata fogant meg a VSZM Közösségi Ház vezetőiben?

R. G. Mert az Enterprise új géptípus. Abban az időben egyetlen ilyen klubról sem volt tudomásunk. Már akkor elkezdtük a szervezését, amikor a gépek értékesítése megindult. Alapvetően új igényt akartunk kielégíteni.

M. M. A VSZM Közösségi Ház régebben szervezett más géppel klubot?

R. G. Igen, C64-re. Ez igen népszerű volt, de a gépeket az LSI-ATSZ elvitte, amikor az ott tartott tanfolyamot befejezte. Ezután a tagok szétszéledtek.

M. M. A Centrum segítette a klub megalakulását?

R. G. Nem. Kértünk támogatást, amit meg is ígértek. Végül is nem adtak semmit, mert érdektelennék tartották a másolások miatt a klub működését. Mi a segítség fejében megígértük, hogy megszűntetjük a programok másolását, és javaslatot is tettünk, aminek a lényege az volt, hogy kapjuk meg a forgalomba kerülő programokat, és azokat egymás között cseréljük. A szellemi termékért járó díjat megfizetjük volna.

M. M. Ez számomra egy kicsit bizarr ötletnek tűnik, de mit szolt az érdekelt?

R. G. Úgy érzem, ők úgy gondolták, hogy azonnal tiltsuk be a másolást. Ezt én egy szóbeli ígéretre nem tettem meg. Ezután megszakadt velük a kapcsolat.

M. M. Bizonyára tudja, hogy a programok másolását hazánkban is törvény tiltja. Hogyan egyeztetik ezt össze a klub tevékenységével?

R. G. Természetesen tudom. De kérdem én, a Bitlet rendezhetett hivatalos másolást a Csokonai Művelődési Házban?

M. M. Ezen a példán felbuzdulva ön bosszánatos bűnnek érzi a másolást?

R. G. Nem teljesen, de ha egy hivatalos

rendezvényen megengedhetik maguknak, akkor miért nem csinálhatja egy kisebb klub? A világon minden klub elsősorban a másolások alapján jön létre.

M. M. Ez valószínűleg igaz, de a tagok által készített programok cseréjére!

R. G. Szerintem ez mindenütt csak álca!

sem vennék elejét a másolásoknak, mert új helyet keresnének. Nem látom értelmét ilyen jellegű beavatkozásnak.

M. M. Tehát kénytelen elfogadni, annak ellenére, hogy tudja: törvénytelen.

R. G. Így van. A másolások számát csökkenteni a hatékony programvédelemmel lehetne, és olyan áron kellene a programokat forgalomba hozni, hogy senkinek se legyen érdeke a feltérés.

MEGKÉRDEZTÜK AZ ENTERPRISE -RÓL

Minden klubban gyári programokat is cserélnék. Ezenkívül vannak olyan programok, amelyek másképpen be sem jutnának az országra.

M. M. Az Enterprise Klub milyen céllal alakult meg? Feltételezem, hogy nem a programok másolására.

R. G. Természetesen nem. A klubot azért hoztuk létre, mert szerettünk volna ott-hont adni egy új géptípusnak, ahol alkalom nyílik az információcsere és a tanulásra. Tisztában voltunk azzal, hogy a másolás nem kerülhet el! Úgy érzem, ezért nem a klubokat kell megszüntetni, hanem az emberek szemléletét kell megváltoztatni.

M. M. Hogyan?

R. G. Erre nem tudok pontosan válaszolni, de ezt a kérdést nemcsak számítástechnikai oldalról kell közelíteni. Az ilyen magatartás általános.

M. M. Ön mint klubvezető meg tudná-e tiltani a másolást?

R. G. A klubban igen, ha mindig figyelünk a másikat és beavatkozunk. De ha az illetőt ki is tiltanánk a klubból, ezzel

M. M. Mi a véleménye arról, hogy a fejlesztők olyan védelmet terveznek, amely feltérési kísérletnél tönkretesz a számítógépet?

R. G. Amennyiben ez a programhoz mellékelt tájékoztató szerepel, semmi kivételmentet nem találunk benne. Ezt a tulajdonságot azonban nagyon figyelemfelhívóan kell közölni.

M. M. A kereskedők és a fejlesztők azon a véleményen vannak, hogy a kalózok miatt senki sem fog programot készíteni. Így a másolók lesznek a felelősök a gyenge ellátásért.

R. G. Tízennyolcezer gép van forgalomban, feltehetően ennyi tulajdonossal. A klubnak háromszázötven tagja van. Ha csak a nem klubtagok vásárolnak, akkor sem fognak a fejlesztők és a kereskedők éhenhalni. Nem lehet ezt a problémát egy klubra ráfogni! Az Enterprise egyébként is speciális helyzetben van, mert ez az első olyan nagy számban elterjedt gép, amelynek esetében a fejlesztők nem számíthatnak tág külföldi piacra és felhasználók külföldi programokra.

PINKE GYÖRGY

A különböző operációs rendszerek általában két célt szolgálnak: szabványosítják és megkönnyítik az I/O (In/Out = Be/Ki) műveleteket. Az EXOS 2.1 (Extendable Operating System — bővíthető operációs rendszer) készítői egy csatorna alapú I/O rendszert fejlesztettek ki, sok meglepően jó ötlettel. A csatornák (channel) keresztül különböző egységeket (device) érhetünk el. Ezek tulajdonképpen az ún. „perifériák”, de nem véletlen az eltérő elnevezés! Egy device-nak nem kell feltétlenül külső eszköznek (perifériának) lennie, akár egy program is „állíthatja azt magáról”, hogy „ő” egy eszköz; így lehet például RAM-diszket létrehozni.

Mielőtt elkezdénének az EXOS-szal való ismerkedést, szót kell ejtenünk az EXOS néhány sajátosságáról. Az általános gyakorlatól eltérően az EXOS-ban a sztringek (karakterláncok) végét nem speciális ka-

Egy EXOS hívás a következőképpen néz ki:

```
RST 30h
DEFB fn code
```

A 30h cím az EXOS helépesi pontja, és az utána álló bájtt tartalmazza a két funkcióját (ez tulajdonképpen egy „kibővített”, kétbájtos RST utasítás). A legelterjedtebb assemberben, az ASMON-ban ehhez beépített makro van definiálva. A két sor helyett elég ezt írni: EXOS fn code

Az EXOS hívásokra általában igazak a következők: A hívás paramétereit az A, B, C, D, E regiszterek tartalmazzák és az eredmény is ezekben a regiszterekben adódik vissza. A többi regiszter (a HL, IX, IY és a vesszős regiszterek) értéke megőrződik. Ha a paraméterek nem férnek el a regiszterekben, a DE

lő egységeknél az írásra nyitottat, a többi egységnél ugyanaz mint 3).
Funkciókód: 5 GET
Be: A csatornaszám
Ki: B kiolvasott bájtt

Az adott csatornáról beolvass egy bájtot; ha nincs adat, vár.
Funkciókód: 6 READ

Be: A csatornaszám
BC hossz
DE puffer címe
Ki: BC hossz — beolvasott hossz
DE új puffer cím

Az adott csatornáról megpróbált beolvasni BC bájtot ($0 \leftarrow BC, \leftarrow 65535$). Ha nem sikerül, hibakóddal tér vissza, de a BC érvényes (a maradék bájtok száma), nem érvényes viszont a BC, ha nem volt nyitva a csatorna.

Funkciókód: 7 PUT

Be: A csatornaszám
B beírandó bájtt

Ki: —

3 ADDR kép címének lekérdezése

4 FONT karakterkészlet visszaállítása

KEYBOARD

8 FKEY funkcióbillentyű beállítása

9 RJoy botkormány leolvasása

NET

16 FLSH puffer ürítése

17 CLR puffer törlése

EDITOR

24 MARG margó beállítás

25 LDOC szöveg betöltése

26 SDOC szöveg kimentése

Funkciókód: 16 EXOS VARI-

ABLE

Be: B alfunkciókód

C EXOS változó száma

(D új érték — csak íráskor)

Ki: (D kiolvasott érték — csak ol-

vasáskor és átbillentéskor)

Ezzel a funkcióval lehet az EXOS változókat (leírásuk a BASIC kézikönyvben) írni, olvasni, átkapcsolni. B értéke íráskor 1, olvasáskor 2.

Funkciókód: 24 ALLOCATE SEGMENT

Be: —

Ki: C szegmensszám

DE EXOS határ

Ezzel a funkcióval lehet a rendszerítő szegmenseket igényelni. Ha nincs szabad szegmens, a rendszer hibát jelez, és DE tartalmazza, hogy milyen címig használhatjuk a C-ben megadott számú szegmenst (a többi részét a rendszer használja).

Funkciókód: 25 FREE SEGMENT

Be: C felszabadítani kívánt szegmens

Ki: —

Ezzel a funkcióval lehet egy már szükségletlen szegmenst a rendszernek visszaadni.

Funkciókód: 28 ERROR

Be: A hibakód

DE puffercím

Ki: DE nem változik

Ezzel a funkcióval a hibakódokat lehet átalakítani — a pufferek 40h méretűnek kell lennie, bár a jelenlegi EXOS-üzenetek ennél rövidebbek.

Átneztük a legalapvetőbb EXOS hívásokat. A következő számban egy példaprogramot közlünk, amely az ASMON-nal lefordítva bemutatja az EXOS hibakódjait és egyúttal példát is ad a csatornakezelésre is.

ERDEI ANDRÁS

AZ EXOS 2.1

rakter jelzi, hanem szerves részük a hossz — az első bájtt. A beépített egységek nevei:

EDITOR:	szövegszerkesztő
VIDEO:	képernyő
TAPE:	magzó
SOUND:	hanggenerátor
KEYBOARD:	billentyűzet
NET:	hálózat
SERIAL:	RS232 csatlakozó
PRINTER:	nyomató
DISK:	lemezegység

Amikor azonban ezekre az egységekre hivatkozunk (például egy assembly programban), a következő formát kell használnunk:

```
VIDNAM DEFB 6          ; a név
                        ; a név
DEFB "VIDEO"          ; az egységség neve
```

Ehhez a konvencióhoz igazodik a fájlnev is. Szintaxis: egység neve — egység száma: fájl neve.
Történetesen tehát:

```
PRGNAM DEFB 11
DEFB "TAPE—
1:PROG"
```

A továbbiakban először megismerkedünk a különböző EXOS funkciókkal, majd a segítségükkel kezelhető beépített eszközökkel.

giszterpár mutatja a paraméterek helyeit a memóriában. Visszatérés-kor az A regiszter a hiba kódját tartalmazza (0 ha nem volt hiba).

A legfontosabb EXOS hívások a következők:
Funkciókód: 1 OPEN CHANNEL
Be: A csatornaszám
DE fájl (ill. egység) neve

Ki: —
Megnyitja az adott csatornát (fájlkezelő egységeknél — például a magzó — olvasásra).

Funkciókód: 2 CREATE CHANNEL
Be: A csatornaszám
DE fájl (ill. egység) neve

Ki: —
Megnyitja az adott csatornát (fájlkezelő egységeknél írásra, a többi egységnél — például képernyő — ugyanaz mint 1).

Funkciókód: 3 CLOSE CHANNEL
Be: A csatornaszám
Ki: —

Lezárja az adott csatornát (fájlkezelő egységeknél az olvasásra nyitottat).

Funkciókód: 4 DESTROY CHANNEL
Be: A csatornaszám
Ki: —

Lezárja az adott csatornát (fájlkeze-

lő Beírja az adott csatornára a bájtot.

Funkciókód: 8 WRITE

Be: A csatornaszám

BC hossz

DE puffer címe

Ki: BC hossz — beírt hossz

DE új puffercím

Az adott csatornára megpróbál beírni BC bájtot — minden mint a 6-os funkcionál.

Funkciókód: 9 ASK CHANNEL STATUS

Be: A csatornaszám

Ki: C csatornaállapot

Rendezi a csatorna állapotát: C=00h ha van olvasandó bájtt, C=FFh ha vége, C=01h egyébként. Ez a lehetőség a billentyűzet lekérdezésénél hasznos, ha nem akarunk addig várni, amíg le nem nyomnak egy billentyűt.

Funkciókód: 11 SPECIAL FUNCTION

Be: A csatornaszám

B alfunkció kódja

CDE paraméterek

Ki: CDE paraméterek

Az egyes egységek speciális funkcióit lehet elérni ezzel a hívással. A beépített egységek a következő alfunkciókódokat ismerik fel:

VIDEO

1 DISP kép megjelenítése

2 SIZE kép méret/üzemmód lekérdezése

RAJZOLJUNK? — IGEN!

A beépített botkormánnyal mozgathatjuk a kurzort. Teherbíró képessége azonban a játék hevében hamar felmondja a szolgálatot. Rajzolásra ennek ellenére mégis nagyszerűen felhasználhatjuk a belső 0 sor-számú botkormánnyt (1. lista). Akik tehát eltökölték, hogy nem csak játszani akarnak, azoknak nem árt a program listájáról egy-két megjegyzés.

A 170-es és 180-as sorokban a BAND kulcsszó a bináris és függvénykapcsolatot végzi el, hasonlóan, mint a HT, a Commodore és a Sinclair gépeknél az AND. Itt külön kulcsszó van a bináris, valamint a nyelvtani ÉS kapcsolatot értékek meghatározására.

Próbáljuk meg kiírtni például az AND 4 és a 3 BAND 4 értékét. Figyeljük meg a különbséget és kísérleljük meg a törvényszerűségeit felismerni. Ha túl rémisztőnek tűnik, akkor beírhatjuk a 170-180-as sorok helyett a 2. listát.

A botkormány kimelésére a 3. listával gyorsítsuk meg a billentyűismétlést. (Csak legyen, aki ezek után programot tud írni a gépen.) Az eredeti állapottól a programból a 205-ös sorral állíthatjuk vissza.

Az így összeállított program a képernyő közepére helyez egy pontot. A botkormánnyal nyolc irányban mozgathatjuk a pontot, amit adott esetben a SPACE billentyűvel és a botkormánnyal radrozhatunk ki. A W billentyű lenyomásával tiszta lappal újra kezdetjük a rajzolást, a Q billentyűvel pedig befejezhetjük a rajzolást és visszaállíthatjuk a billentyűismétlés sebességét.

SZ. L. J.

1. lista

```
100 REM botkormánnyal rajzol
110 GRAPHICS
120 SET PALETTE BLACK,WHITE
130 LET X=640
140 LET Y=360
150 PLOT X,Y;
160 LET A=JOY(0)
170 LET X=X+4*(A BAND 2)*(X>0)-8*(A BAND 1)*(X<1200)
180 LET Y=Y+2*(A BAND 4)*(Y>0)-1*(A BAND 8)*(Y<710)
190 SET INK(-1*(A<15))
200 IF INKEY$="w" THEN RUN
210 GOTO 150
```

2. lista

```
170 IF (A=2 OR A=6 OR A=10 OR A=18 OR A=22 OR A=26) AND X>0 THEN LET X=X-1
175 IF (A=1 OR A=5 OR A=9 OR A=17 OR A=21 OR A=25) AND X<1200 THEN LET X=X+1
180 IF (A=3 OR A=4 OR A=5 OR A=19 OR A=20 OR A=21) AND Y>0 THEN LET Y=Y-1
185 IF (A=8 OR A=9 OR A=10 OR A=24 OR A=25 OR A=26) AND Y<710 THEN LET Y=Y+1
```

3. lista

```
115 SET 11,1
116 !
204 !
205 IF INKEY$="q" THEN GOTO 220
220 SET 11,30
```

Mi a manó?

Miért nem ad az Enterprise színes képet a kétnormás színes televízió, ha monitor-csatlakozó videojelet kötik rá? Mert a videokimeneten nincs a PAL-jel egyik legfontosabb összetevője, a BURST-jel. Ezen úgy segíthetünk, hogy az UHF-modulátor bemenetére adott videojelet kivezetjük a monitorcsatlakozóra, amitől az már valóban a teljes PAL-videojelet tartalmazza. A modulátor egy kis fémdoboz, amelynek bemenete felülről nézve a bal oldalon van, kimenete pedig az UHF-csatlakozó, ami a doboz és egyben a számítógép hátsó oldalán található. Természetesen praktikusabb, ha a szerelést a szerviz szakembereivel végeztetjük el.

Lemez meghajtó. A Videoton által a TVC-hez gyártott meghajtók kitűnően használhatók az Enterprise-hoz is, csak egy kis átalakítás szükségeseltetik. Az interfész bemenete pontosan fordítva van az Enterprise kontrolléréhez képest. Nem tehetünk mást, mint hogy a csatlakozást gátoló illesztőt lereszeljük. Ezt egyébként nem tanácsos megtenni, de ezúttal ez az egyetlen és biztos megoldás.

Főlős controller. A Videoton által gyártott lemez meghajtókhoz a vásárlónak meg kell vennie a TVC-kontrollert is, akár akarja, akár nem. Ez

az Enterprise-tulajdonosoknak felesleges kiadás, a TVC-tulajdonosoknál pedig előfordulhat, hogy csak kontrollerre lenne szükségük. A Centrum megígérte, hogy tárgyal a szállítóval a controller és a lemez meghajtó szétválasztásáról.

Kiss Árpád olvasónk fényűjséggel küldi üdvözlését az Enterprise-osoknak. A program a képernyőn futó feliratot állít el. Kis átalakítással bármilyen szöveg „futtatható” a segítségével.

```
100 TEXT
110 ALLOCATE 15
120 CODE BALRA=HEXS("3E,
20,01,01,00,54,5D,
09,01,27,00,ED,80,
12,C9")i
130 READ A
140 FOR I=1 TO A
150 READ AS
160 POKE 42720,ORD(AS)
170 CALL VAR
180 NEXT
190 FOR I=1 TO 40
200 CALL VAR
210 NEXT
220 RESTORE
230 GOTO 130
240 DEFF VAR
250 FOR Q=1 TO 50
260 NEXT
270 CALL USR(BALRA,42681)
280 END DEF
290 DATA
38,U,D,V,-,M,I,N,D,
E,N,,E,N,T,E,R,P,R,
I,S,E,,G,E,P,T,U,
L,A,J,D,O,N,O,S,N,A,
K,
```



A sorozat alapelrendelése — azon a régi felismerésen túl, hogy az elektronika és a számítástechnika elválaszthatatlan egymástól — a következő tapasztalatot summázza.

A szoftver — a programok — jelentősége egyre nő, de az is tény, hogy az igazán jó (az adott számítógép nyújtotta lehetőségeket maximálisan kihasználó) programok megírásához a programozóknak rendelkeznie kell alapfokú áramkörti hardverismerettel is. Megelőzti ezt, hogy szaporodnak az olyan berendezések, mikroprocesszort alkalmazó rendszerek száma, amelyek programvezéreltten működnek. Az ilyen rendszerek tervezőinek és fejlesztőinek is szükségük van integrált hardver-és szoftverismeretekre.

A mikroévek nélkülözhetetlen elemei a tárák vagy más néven memóriák, amelyekkel az információ hosszabb-rövidebb ideig tárolható. A digitális információ egysége, a bitet olyan elem tárolhatja, amelynek két stabil állapota van. Ilyen elektronikus elem a már bemutatott tároló, a flip-flop. Ebből egy áramkörti lapkán akár több százézer is elhelyezhető.

Ismerkedjünk meg a tárákhoz kapcsolódó néhány fogalommal. A mikroévekben használt tárákban az információ (a program utasításait, adatokat) bitsoporként vagy más néven szavaként helyezik el. A szóhossz a mikroévek fontos jellemzője, általában 4...64 bit hosszúságú. Mi a továbbiakban a nyolcbites szóhosszúságú (bájt szervezésű) tárákat — mint a leggyakrabban használtakat — mutatjuk be.

A tárbán elhelyezett szó helyét a *cim* határozza meg. A tár *rekesz*re van osztva. Minden rekesznek címe van és minden rekesz cellában helyezkedik el a szó bitjei. Lényeges jellemző a tárkapacitás, amelynek nagyságát a maximálisan tárolható bitek vagy szavak számával adják meg. A tárkapacitás és a címvezetők száma között egyszerűen kapcsolatot van: N címvezetéknel a tárkapacitás 2^N, amí például 10 címvonalnál 2¹⁰ = 1024 szó, azaz 1 kilószó. A tárák másik, nagy jelentőségű jellemzője az olvasáshoz szükséges idő, azaz a hozzáférési idő. Ez a művelet

elindítása és a kimeneten az információ megjelenése között eltelt idő.

Mikroévekben alapvetően két típusú tárat, a mikroépg programját tároló, csak olvasható memóriát (a ROM-ot) és az adatok tárolására szolgáló írható-olvasható memóriát (RAM-ot) alkalmaznak. Mi először a leggyakrabban használt, a felhasználó által programozható és ultraibolya fényvel törölhető, csak olvasható memóriát, az EPROM-ot mutatjuk be.

EPROM

A mikroszámítógépes rendszerek fejlesztése során eljutunk ahhoz a lépéshez, amikor a megírt, hibátlanul tartott programot a fejlesztés alatt álló készülékben lévő mikroprocesszor rendszerben akarjuk futtatni. Ehhez a kifejlesztett programnak megfelelő bájtorsorozatot az EPROM-memóriába kell írni, s a mikroprocesszor ezt olvassa, az adott programot végrehajtja.

Ezek a felhasználó által többször programozható és törölhető memóriák általában nyolcbites adatszélességgel készülnek. Az első, már jól használható memóriák 1024 × 8 bit = 8 kbit kapacitásúak voltak. A legnagyobb gyártó, az Intel cég, a 2708 jelölést adta ennek a típusnak. Rövidesen megjelentek a 2716 (2 k × 8 bit = 16 kbit), 2732 (32 kbit) típusok is. Jelenleg már alkalmazkák a 2764, 27128, 27256, 27512 és nagyobb típusokat is, melyeknek kapacitása rendre 8, 16, 32, 64 kb-ot. Ezek az egységes kezelhetőségért és a kompatibilitásért szabványos lábkiosztással készülnek, azaz a különböző kapacitású típusoknál a legtöbb kivezetés azonos helyen van. A 24, illetve 28 kivezetéssel készülő típusok lábkiosztását az 1. ábrán foglaltuk össze.

Az EPROM-ok kivezetései funkcionálisan négy csoportba oszthatók:

Címvezetések (jelölés: A0...A_N)
A címvezetésekre adott bináris kód jelöli ki az elérni kívánt tárolóhelyet. A címvezetések száma a memória kapacitásától (mértéktől) függ. Ha a memória 1 kb-ot, akkor ezeknek a száma 10 (2¹⁰ = 1024 = 1 k). A 2764-es típusnál ez a szám 13.

Adatkivezetések (jelölés: D0...D7)
Vezérlőkivezetések

CS — (chip select) memóriaengedélyező vezeték. Az L szintje engedélyez, hogy az adat a belső adatbemenetre kerüljön.

DE — adatkimenet-engedélyező vezeték. Az L szintje engedélyez, hogy az adat a belső adatkimenetekről az adatkivezetésre kerüljön.
VPP — programozó kivezetés. Erre a lábra kapcsolatt, adott nagyságú és idejű feszültségimpulzussal lehet a tokot programozni.

Tápfeszültség-kivezetések
Ucc: +5 V
GND: 0 V

Az EPROM-ok programozása

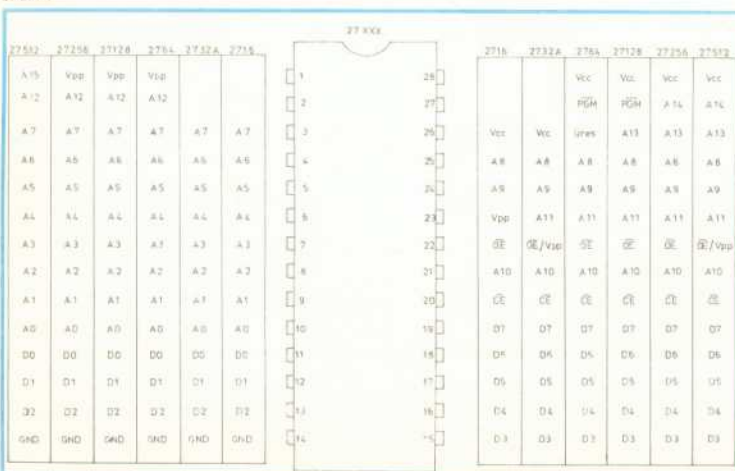
Az EPROM-memóriába a bájtok bevitelét és tárolását megvalósítást az EPROM programozásának (zsargonban: EPROM-égetésnek) nevezzük. A tokokat bájtontként programozzuk, mégpedig úgy, hogy a címvezetékekkel kiválasztjuk a tárolóhelyet, az adatonalakra ráadjuk a beírandó bájtakat megfelelő bitkombináció, majd a programozó kivezetésre adott megfelelő feszültségimpulzus hatására a bájt az adott memóriacímre beíródik.

A feszültségimpulzus a programozó kivezetésre adott, rögzített nagyságú és időtartamú egyenfeszültség. A nagysága általában 25 V, 21 V és 12,5 V. Időtartama általában 50 ms. Ez azt jelenti, hogy például a 2716-os típus programozása 2048 ± 50 ms = 102,4 másodpercig tart, de ez az idő a 27256 típusnál már közel fél óra! Ezért a nagyobb kapacitású EPROM-ok programozására a gyártók ún. gyors programozási algoritmusokat fejlesztettek ki, amelyeknek a segítségével jelentősen lerövidíthető a programozási idő.

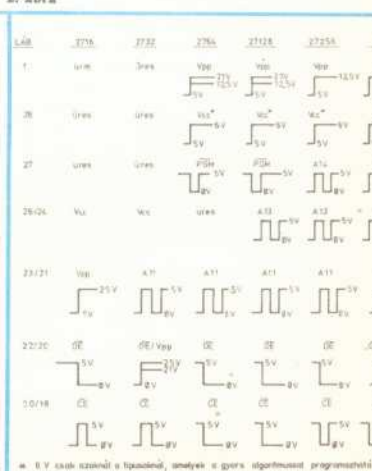
Ennél a gyors programozási módnál a programozó impulzus szélessége nem 50 ms, hanem egy adott algoritmus szerint változó szélességű, éspedig 4–60 ms-ig terjedően. Ilyen programozásnál a tok tápfeszültségét is megemelik a programozás idejére 5 V-től 6 V-ra. A különféle típusok programozását a 2. ábrán foglaltuk össze.

Az eredeti, programozatlan állapotban az EPROM minden bitje 1 állapotban van. A programozás során csupán az adatonalvonl levő 0 szint

1. ábra



2. ábra



* 1 V csak számláló tipusoknál, amelyek a gyors algoritmust programozhatják.

móriák

íródik be. Nagyszámú adat programozásokra a programozás folyamatát célszerű automatikussá tenni. Ezt a célt szolgálják a mikroprozessoros rendszerekben alkalmazott EPROM programozók. A programozót kezelő program leggyakrabban használt parancsai:

- az EPROM törölt (üres) állapotának ellenőrzése
- az EPROM tartalmának memóriába olvasása
- az EPROM programozása a memóriában lévő adatokkal (az egyes bajtok helyes beírását ellenőrzi)
- az EPROM tartalmának és a tárban lévő tartalomnak az összehasonlítása

Az EPROM-ok törlése

Az EPROM törlésén az értjük, hogy a már beprogramozott és így az egyes tárolóhelyeken lévő logikai nulla állapotú tárolócellákat ismét 1 állapotba hozzuk, a benne lévő tartalmat „kivádoszuk”. Törölési a tok tetején lévő átlátszó kvarcablakon keresztül, ultraibolya fényvel lehet. Ilyen fényt bocsát ki például a kvarclámpa és a germicidosz. A törlés ideje (a „napoztatás”) 15–20 perc, az EPROM gyártója által specifikált időnek megfelelően. A forgalomban lévő EPROM-törölő készülékek egy állítható kapcsolóórát és az ultraibolya fényt szolgáltató fényforrást tartalmaznak. A törölendő EPROM(ok) behelyezése után a készüléket és a fényforrást bekapcsoljuk, majd az, a kapcsolóórán beállított idő elteltével kikapcsol. A gyakorlatban, ha a gyártó ismeretlen, a biztos törlés időtartama például úgy határozható meg, hogy először öt percre állítjuk be a törlés időtartamát, majd az EPROM ürességét ellenőrzük, majd ismét tíz percre törlés, utána ellenőrzés, egészen addig, amíg a tokot üresnek nem találjuk. A törlési idő ezeknek az ötperces időtartamoknak az összege. Egyszerű kvarclámpa használatkor ügyelni kell arra, hogy a törölendő tokokat a fényhez közel helyezzük el, de gondoskodni kell a tokok hűtéséről (ventilátoros hűtés oldalról), valamint a fény szemét károsító hatásának megakadályozásáról, azaz árnyékolásáról.

Primo EPROM-programozó

A cikksorozat célja többek között az, hogy a tudást megalapozó ismeretekkel az olvasó is képes legyen kis, önálló mikrogépes fejlesztések megvalósítására. Ehhez egy alapvető munkaeszköz egy EPROM-programozó készülék. Mivel kisebb fejlesztésekhez elegendő a 2 kbájtos 2716-os, illetve a 4 kbájtos 2732-es EPROM-ok használata, ezért megoldásként egy, a Primo számítógéphez illesztett EPROM-programozó leírását közöljük.

Az EPROM-programozó tervezésénél a fő szempontok a könnyű kezelhetőség, az egyszerű és olcsó áramkörti kialakítás voltak. A programozó kapcsolási rajza a 3. ábrán látható.

— Az áramkör fő részei:

1. Az EPROM adatkivezetéseire adatkivittelt végző háromállapotú kimenettel ellátott tároló (V1)
2. Az EPROM-ból adatbeolvasásra szolgáló háromállapotú kapuáramkörök (V3–V4)
3. A vezérlőjeleket tároló áramkör (V2)
4. A programozófeszültséget előállító áramkör (DC-DC konverter)

5. Az EPROM címvezetékehez kapcsolódó 12 bites számláló áramkör (V5).

A be/kimeneti egységekhez szükséges címdekódoló áramkört nem kellett kialakítani, mert a dekódolt jelek a Primo hátsó csatlakozóján megtalálhatók. A kapcsolási rajz bal oldalán látható, zárójelbe tett kivételazonosítók a Primo 2x25 pólusú hátsó csatlakozójára vonatkoznak. Az EPROM címait, mivel címzésre számlálót alkalmazunk, csak egyenként, sorban egymás után lehet léptetni. Ennek az az előnye, hogy csupán két vezetéket (számlálóórás és léptetés) szükséges a kívánt cím beállítására. Egy adott címre állást egy törölt- és adott számú léptetőimpulzussal lehet elérni.

Ez a megoldás kissé kényelmetlennek látszik, de a gyakorlat azt mutatja, hogy ritkán dolgozunk az EPROM egy-egy bajtjával; általában összefüggő tartományokat írunk és olvasunk, ami azt jelenti, hogy a léptetéses megoldás jól használható.

A háromállapotú kimenettel ellátott tárolóra azért volt szükség, mert amikor az EPROM-ból az adatokat kiolvassuk, a vonalakra kapcsolódó tároló kimenetnek nem szabad terhelniük a szintén kimenetként működő EPROM adatkimeneteket (a két kimenet „összeveszne”).

A programozó 2716-os és 2732-es típusú EPROM-ok programozását teszi lehetővé. Három funkciója van: ellenőrzés (törölt-e az EPROM, azaz minden bit 1 állapotban van); írás (memóriatartalom beírása az EPROM-ba); olvasás (EPROM-ban lévő tartalom beolvasása a memóriába).

Az EPROM és a memória tartományainak kijelölésére a következő paramétereket használhatjuk:

MKCIM: a memóriatartomány kezdőcíme
 MYCIM: a memóriatartomány végcíme
 EKCIM: az EPROM-beli tartomány kezdőcíme

EVCIM: az EPROM-beli tartomány végcíme.
 Egy program indításakor célszerűen a képernyőre egy menü íródik ki, ahol a kiírt kérdésre adott választként meg kell adni a kívánt funkciót (E, 1 vagy 0 betű); a tipust (2716 => 0, 2732 => 1), a funkciót által megkivánt paramétereket, visszelve elválasztva. Ezután egy megerősítő kérdés íródik ki: MEHET (I/N)?

I-t lenyomva a funkció végrehajtódik, N-et lenyomva új értékek adhatók meg.

A programozás, olvasás vezérlését a 74LS273 típusú tárolóba írt bajt végzi el. A vezérlő bajt egyes bitjeinek kiosztása:

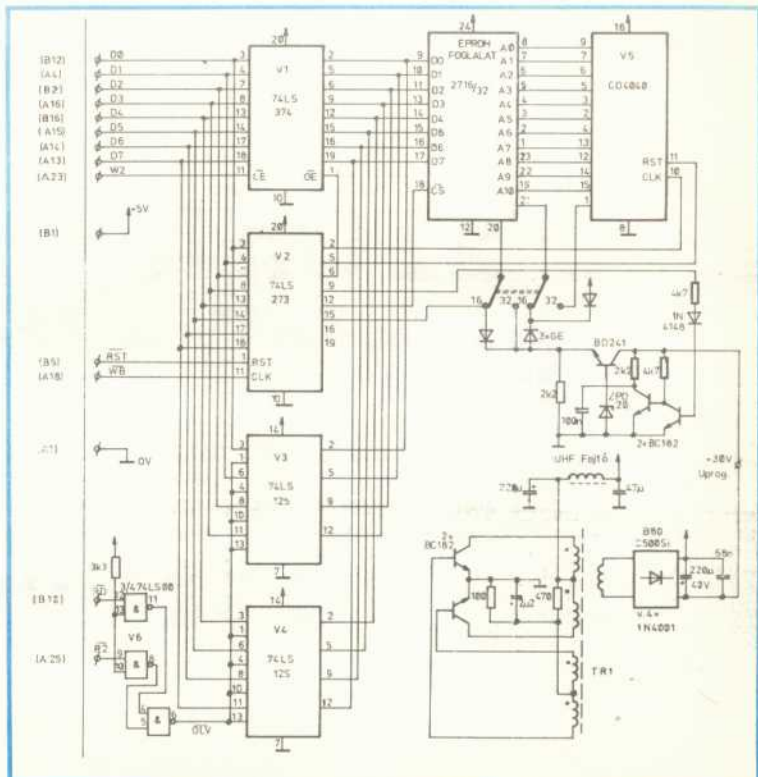
- Bit 0 CD 4040 számláló órajele (lefutó éle billen).
- Bit 1 CD 4040 számláló töröljele (1-nél több)
- Bit 2 74LS374 számláló (ADAT port) magas impedancia vezérlő EPROM olvasáskor (1 — magas impedancia (tri state))
- Bit 3 programozófeszültség (25 V vagy 21 V) rákapcsolása az EPROM-ra (1 — bekapcsolás)
- Bit 4 CS — EPROM kiválasztó jele (18-as kivétel) 0 — kiválasztva
- Bit 5 OE — vezérlés 2716-os típusnál
- Bit 6 üres (A13)
- Bit 7 üres (A14)

Megjegyzések:

- A 6. és 7. bitek a programozó továbbfejlesztésekor alkalmasak nagyobb kapacitású (2764, 27128) EPROM-ok legfelső címvezetékeinek vezérlésére, ahogy ezt a zárójelben jeleztük.
- A Primo bekapcsolásakor a fenti összes bit alacsony szinten van!
- A programozáshoz szükséges +30 V-os egyenfeszültséget 24 V ~ feszültség egyenirányításával is előállíthatjuk.
- A fentiek alapján a program akár gépi kódban, akár BASIC-ben könnyen elkészíthető.

DR. KÓNYA LÁSZLÓ

3. ábra



ÁTALÁNYDÍJAS JAVÍTÁSI ÉS KARBANTARTÁSI = ÖRÖK SZERZŐDÉS GARANCIA



**COMMODORE, ATARI, TVC stb. személyi számítógépek,
valamint IBM PC/XT/AT professzionális PC számítógépek
és perifériák (floppy, printer)
garanciális és fizető JAVÍTÁSA, KARBANTARTÁSA.**

SZERVIZEINK:

1053 Budapest V., Magyar u. 12-14.

1083 Budapest VIII., Szilgyony u. 9.

1191 Budapest XIX., Gábor A. sétány 3.

3100 Salgótarján, Arany János u. 3.

3525 Miskolc, Fazekas u. 1-3.

T.: 173-551 Tx: 7621

T.: 343-153

T.: 32-14-007

T.: 46-17-011

4034 Debrecen, Holló László u. 14.

5600 Békéscsaba, Bartók Béla u. 37. T.: 66-27-195

6726 Szeged, Székelysor 13. T.: 62-13-377

7400 Kaposvár, Füredi u. 24. T.: 82-16-307

7624 Pécs, Jurisics M. u. 17. T.: 72-11-812

9700 Szombathely, Szalonok u. 31. T.: 94-14-519

1053 Budapest, Henszlmann I. u. 9. Telefon: 174-144 Telex: 22-7621

VEVŐSZOLGÁLAT — FOTOELEKTRONIK — NOVOTRADE — ALFA G.T.

VÉTEL — ELADÁS 1077 Budapest, Dohány u. 16. Tel.: 428-936

Az Ön feje nem adattárolásra való, hanem fontos döntések hozatalára.
Joggal várhatja el, hogy kezében legyen az eszköz, ami munkáját minőségivé teszi,
döntését megalapozza.

A számítástechnikában viszont a széles választékból, nem könnyű
a legjobb mellett dönteni.

az Ön fejével gondolkodtunk,

amikor létrehoztuk az első, Közép-Európában egyedülálló számítástechnikai szalont.
Meghívtuk a legfontosabb gyártókat és forgalmazókat, hogy a választék együtt legyen
áttekinthető, kipróbálható, tanulmányozható, összehasonlítható.

Felkészült szakembereink várják az érdeklődőket, a leendő vásárlókat.
Reméljük, döntésünk új korszakot nyit az Ön mindennapi munkájában.

PC szalon

Budapest XIII., Sallai Imre u. 6.

☎ 315-136, 310-776

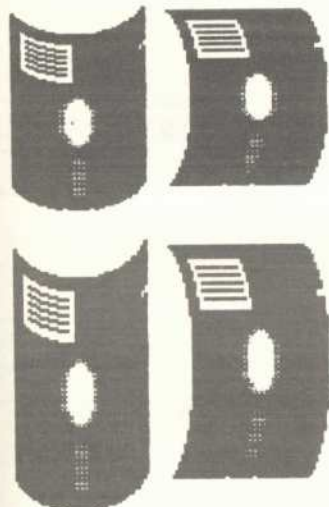
Lépjen új korszakba velünk.

NOVOTRADE



VAN ÚJ A NAP ALATT?

EGA, Amica Paint és Giga Paint

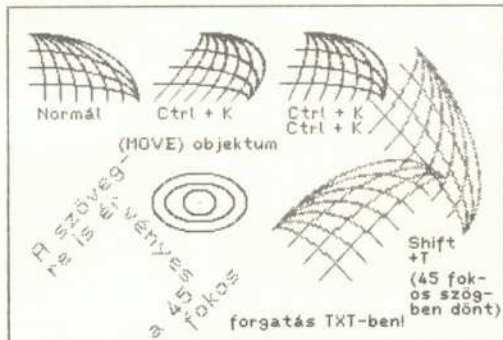


Mostanában kevés új rajzoló- és festőprogram készült a C64 számítógéphez. Sokaknak úgy tűnt, hogy a lehetőségek kimerültek. A kételkedőkre cáfolt rá az NSZK-beli Markt & Technik kiadó, mely néhány hónapon belül három grafikai programmal rukkolt elő.

Azt, hogy a rajzoló- és a festőprogramoknak milyen alapvető szolgáltatásai vannak, a legtöbb programokat gyűjtő és használó jól tudja. Mégis lehetnek olyan olvasóink, akik most ismerkednek a géppel és az ilyen szoftverekkel is. Melyek tehát azok a funkciók, amelyeket többé-kevésbé minden „elektronikus rajzművész” programnak tudnia kell? Először is kell kört, ellipszist, négyzetet, különböző vastagságú vonalakat rajzolni, vagyis alapvető geometriai formákat és ezek kitértített változatait. Természetesen lehetőséget nyújt arra is, hogy „szabad kézzel”, azaz egérrel, botkormányval rajzoljunk, mintha egy papírlap lenne előtünk. A jobb programok a pontosabb munkát koordinátakijelzőkkel segítik, a rajzrészletek kicsinyíthetők, nagyíthatók, kisebb feliratok készítéséhez szöveget is írhatunk a rajzokra, és mindezt tetszőlegesen lemezre rögzíthetjük, illetve onnan visszakereshetjük a képernyőre, vagy mátrixnyomatón kinyomtathatjuk.

Az említett nyugatnémet újdonságok mindegyikének tulajdonságai az alapkövetelményekben meghatározottak, ezért most csak azokra az új rajzforgalmazó eszközökre hívom fel a figyelmet, amelyek a megszokottól eltérnek.

Néhány lehetőség az EGA-val



EGA V3.2

Néhány napos munkámba került, amíg beirtam a „64'er” magazinból az EGA V3.2 listáját (EGA = Electronic Graphite Abundance). A fáradságot egy kiválóan működő rajzóprogram volt a jutalom, amely meglepő újdonságokkal szolgált. A jobb német festő- és rajzóprogramok jellemzőjét a szoftver írója, S. Meyer most is érvényesítette. Ez azt jelenti, hogy az ebben a műfajban általános menürendszert elkerüli. Így nem kell „végigenni” az étlap fogásait, mert a billentyűzettel gyorsabb kapcsolatot tarthatunk a számítógéppel.

Amint betöltöttük a programot, első rátekintésre is újdonsággal találkozunk. Az x,y koordinátakijelzés, az aktuális rajzfunkció és más fontos információk állandó visszajelzése nem csökkenti a „rajzlap” területét, mert ezeket egy ügyes trükkrel a képernyő szegélyére varázsolták. Egyszerre egy fél A/4-es lapon dolgozhatunk, ami 640 x 400 pontból áll, s ezt négy részre osztva, képernyőként használhatjuk. A szomszédos „lapokra” automatikus az átmenet, ami még a rajzófunkciókat sem zavarja. Ezek között is találhatunk olyat, ami nem általános. Ilyen például a tetszőleges új rajzolása, ami tulajdonképpen egy negyed ellipszis.

Amica Painttel készült rajz, a Giga Paint univerzális nyomtatórutinájával különbözőképpen kinyomtatható

A HÍRADÁSTECHNIKAI TUDOMÁNYOS EGYESÜLET

Mikroszámítógépes programnyelvek és operációs rendszerek szakosztálya előadást tart az alábbi témában:

Vita a gazdaságpolitikai koncepciók és a műszaki haladás viszonyáról:

- a műszaki fejlesztés esélyei
- mit tesz lehetővé a gazdaságpolitikai feltételrendszer
- és mit kellene lehetővé tennie.

Előadó: Dr. Vértess András (a Gazdaságkutató Intézet igazgatója).

Helye: MTEESZ Székháza, Budapest V. ker., Kosuth tér 6-8. IV/437.

Idője: 1988. október 11. 15 óra

Minden érdeklődőt és hozzászólót várunk.

Az NSZK programfejlesztői már eddig is jó hírnevet szereztek maguknak a C64-re írt rajzoló- és festőprogramokkal. A jól bevált „régieket”, a Hi-Eddi+, a Profi Painter, a Star Painter, a Cheese, a Giga Cad után a „64'er” című nyugatnémet szakfolyóirat kiadójának sikerült meglepnie a felhasználókat. Az EGA V3.2, a Giga Paint és az Amica Paint programok teljesen új színeket hoztak a grafikusok és a számítógépes grafikát kedvelők palettájára. Mindhárom program a Markt & Technik kiadó ösztönzésére és bábkösdásával látott napvilágot, illetve monitorképernyőre.

Kész kitöltőmintákat nem kapunk, de kis gyakorlattal korlátlan számú állíthatunk elő. A program másik nyencsége az, hogy a mozgásra kiválasztott területet és az ábrára írt szöveget két fozokozatban dönthetjük. Egy alap-karakterkészlet listáját kínálja a magazin, ezenkívül a beépített karakterkészítővel viszonylag egyszerűen bármilyen betűkészletet tervezhetünk és tárolhatunk lemezen. Itt az a különlegesség, hogy a betűk nagyságának csak a gép memóriája szab határt. A szöveget akár negatívban is írhatjuk és — akárcsak a mozgásra kiválasztott területet — 45 fokos (!) lépésekben forgathatjuk. Kevésbé általános az a lehetőség is, mely módot ad arra, hogy a képet az x,y tengelyen megnyújtsuk.

Hasznos tulajdonsága a programnak, hogy a képeit elfogadják más rajzolók és a Printfox oldaltervező is, de az EGA maga sem finnyás, és cserébe szintén betölti azok képeit. Ezt a tulajdonságát kihasználhatjuk az új karakterek tervezésénél is.

Az elkészült alkotásunkat Epson vagy azzal kompatibilis nyomtatóval (például Citizen 120D) nyomtathatjuk ki. (A nyomtatórutinra vonatkozó programlista beírása után még nem működöképes, azt külön le kell futtatni, és ekkor generálja a végleges rutint.) Időközben megjelent az MPS—801-re vonatkozó lista is.

Az előnyök mellett eltörpülnek a hátrányok. Ilyen például, hogy nincs „undo”, vagyis visszalépési lehetőség egy elrontott rajzfázis után. A hiányzó radirt kisebb kecsélgelés után a negatív blokkal való letakarában találtam meg. Vonalvastagság, pontosabban -vékonyaság csak egyféle van, és csak a „szabadkézi” rajznál működöképes a kijelölt terület mozgatásával nyert, tetszés szerinti ecsetméret, illetve forma.

Végeredményben viszonylag csekély befektetéssel — néhány oldalnyi program begépelésével, vagy a magazin lemez mellékletek megvásárlásával — egy jó és más programokat egyéni eszközökkel kiegészítő, nagy felbontású rajzolóprogramhoz juthatunk az EGA által.

A multikolor zsonglőr

A Markt & Technik kiadó multikolor módban működő „csodagyereke”, a „64er” hivatalpályázatának díjnyertes programja az Amica Paint. A beharangozott extra lehetőségek tömege felkeltette szerkesztőségünk érdeklődését, ezért megkértük, küldjenek egy példányt tesztelésre. Nem csalódtunk. A program írja, a fiatal Oliver Stiller valóban megérdemelte az első díjat. Noha az Amica Paint elnevezést másból vezeti le (Advanced Multicolor Computer Aided PAINT), mégis nagyon sok olyan hasonló funkciót ismerhetünk fel, amely az Amiga DeLux Paint II rajzoló, festőprogramjának tulajdonságaira emlékeztet.

Az Amica Painttel olyan képeket, számítógépes grafikákat készíthetünk és olyan

gyorsan, amilyenekről a C64-tulajdonosok nem gondolták volna sohasem, hogy gépük erre is képes. A programra jellemző a különböző funkciók tömege, és ezek között jócskán akad olyan is, melyet eddig csak nagyobb gépek tettek lehetővé. Tekintettel a választék dzsungelére, az Amica Paint használata nem túl egyszerű. Az eligazodást viszont segíti a sok, jól áttekinthető menü és almenü, a munkát pedig gyorsítja a nyomógombos parancskiválasztási rendszer. A program két üzemmódban dolgozik: vagy Commodore 1351-es típusú analóg egérrel vagy botkormánnyal. Az első tendőnk tehát a beviteli eszköz kiválasztása, amit külön program választ meg.

Nem az első rajzoló ez a kezében, de mégis több napra volt szükségem, amíg a rajzoló-, valamint egyéb funkciók halmazán átrágtam magam, és kiismertem magam közöttük. Az EGA-hoz hasonlóan az Amica is az aktuális visszajelzéseket a képernyő felső és alsó szegélyére írja sprite-ok segítségével. Így zavartalan a 200x160 pontból birtokunkban levő elektronikus rajztábla.

Megpróbálom a sok új lehetőségből egy kis izelítőt adni. Szabályos négyzet, például két, tetszőleges paralelogramma, három kattintással rajzolható meg. A körív és ellipszisív, mint újdonság itt már külön van választva. A kitöltött felületek kialakításához külön menüből választhatók ki az utasítások. Ebben találjuk meg a kitöltött, tetszőleges nagyságú ellipsziscikket és körcíkkeit is. Nagy erőssége az Amica Paintnek a kitöltőminták, az ecsetek, a „szórópisztoly” szórásának a szinte korlátlan szerkeszthetősége. Például a képernyő kijelölt részéből „kivágot” rajzdarabkával festhetünk tovább. Lényeges tulajdonsága, hogy ezek a tetszőleges méretű, színösszeállítású minták, vonalak (!) és ecsetformák a rajzolófunkcióban is működöképes. A képernyőn bármekkora ablak kijelölhető, és akkor csak ott fog a „ceruza” és az „ecset”. Újdonság az is, hogy a megrajzolt sikkforma vetett árnyékát a kívánt szögben és távolságban gombnyomásra automatikusan kirajzolhatjuk. Érdekes plasztikus hatás érhető el a szabályozható méretű és erősségű elmosási utasítással és a kijelölt területen a színek összekeverésével. Térhatású alakzatok is előállíthatók egy olyan parancsral, amelynek hatására a kívánt irányban megvastagítható a rajzolt forma körvonalának alsó széle.

Figyelemre méltóak a kijelölt blokkokra vonatkozó lehetőségek is. Hogy ezeket lemezre írni, betölteni, másolni, kicsinyíteni, nagyítani, tükrözni tudjuk, az már nem újdonság. Viszont az, hogy a kiválasztott területet bármilyen szögben elforgassuk, a térbe döntjük, sőt, ha kedvünk tartja, akár szalagként meghajlítjuk — egy C64-en fűró programtól teljesen szokatlan teljesítmény.

Azt mondják, ahol fény van, ott árnyék is van. Ez a megállapítás igaz az Amica Paintre is. Azt, hogy a kiszolgálása kicsit nehézkes (például ha egy aktuális szint akarunk módosítani, ehhez három billen-



Egy Pagefókszal tervezett obléma a Giga Painttel tovább alakítva és a Giga Printtel kinyomtatva

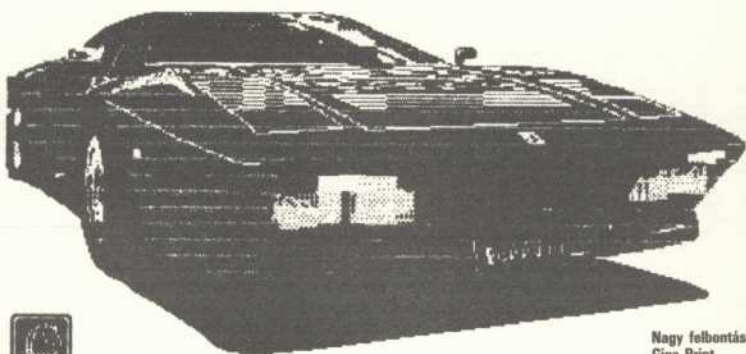
tyü megnyomására van szükség), azt nem vethetjük a szemére, mert ez a gigantikus funkció tömeg következménye. Nagyobb hiba viszont, hogy hiányzik a nyomtatórutin és egy karakterszerkesztő, mert a meglévő egy betűtípus csak ideiglenes alap lehet egy ilyen, szinte professzionális programnál. A kiadó ígérete szerint ezeket a hiányosságokat pótolni fogják, és lehet, hogy mire ezeket a sorokat olvasóink kézhez kapják, a hiányzó programrészek rutinjának listája már megjelenik a „64'er”-ben. (A karakterszerkesztő programlistája lapzártá után meg is jelent, bár kezelése elég nehézkes. Ugyancsak megjelent két karakterszerkesztő, a „Mini” és a „Profi” és egy korszerűbb szövegkezelést nyújtó lista is.) Ezek után – ha kifizetjük a programlemezért a harminc-egynéhány márkát – bátran állíthatjuk, hogy a legtöbbet tudó multikolor rajzoló-festő program birtokosai letünk.

Giga Paint a „Bookware”

A Markt & Technik kiadó új sorozatot indított Bookware néven. Ez az új szó lényegében az angol book, azaz könyv szót tolja a „software” szó második felével. Olvasóink ebből kitalálhatják, hogy mit takar ez az új fogalom. Igen, olyan könyvről van szó, amely egy teljes értékű szoftvert is tartalmaz mellékletként, méghozzá a nyugati könyv- és programárhoz képest viszonylag nem túl drágán, 59 márkás áron.

A sorozat első tagjai között van a frissen elkészült Giga Paint, amit későbbre a kiadó szintén elküldött lapunknak kipróbálására.

A szuperlatívuszokkal beharangozott programcsomagban nem csalódtam. Miután átrágtam magam a program 260 oldalas „book” részén, megállapítottam, hogy végre ez az az univerzális grafikai program, amit már régóta vártak a C64-es grafikai célokra is használók. Ez az első olyan program, amely választhatóan alkalmas mind a multikolor, mind a HiRes, azaz nagy felbontású üzemmódban való használatra. Ideális arra is, hogy a különböző rajzolóprogramok segítségével készített műveinket betölthessük, majd a továbbfejlesztett változatot akár egy újabb rajzolóprogramban felhasználható formátumban ismét lemezzre írassuk. Mintegy harminc különböző programot „házasíthatunk” így össze. (Saj-



Nagy felbontású Giga Print nyomat

nos az 1988/7. számunkban beharangozott GEOS kompatibilitás csak a GeoPaint képek betöltésére érvényes.)

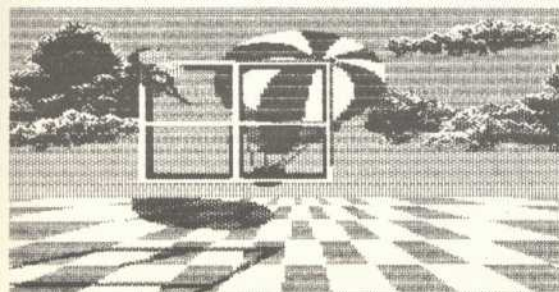
A Giga Paintnek az a tulajdonsága, hogy a képernyő külső szegélyén helyezi el a menükorsort, szintén az új NSZK irányvonalat képviseli. Az újdonság a tízre bővíthető piktogramos vízszintes mozgatási lehetősége. A főmenük egész oldalt foglalnak el, és ha eltűnnek, a teljes képernyőn rajzolhatunk. Erre négy (!) bittérképünk van, amelyet scrollozhatunk is. A két állandó betűkészleten kívül választhatóan másik kettő tölthető be a RAM-ba. Egyidejűleg 64 esetformából és 120 mintából válogathatunk.

A program a rajzfunkciókon kívül alkalmas maximálisan 40 x 24 pontméretű betűkészlet előállítására, sprite-ok tervezésére, és még saját beépített monitorja is van.

Most pedig röviden az új csemegékről, melyekről a program írói, Wolfgang és Klaus Oppacher, valamint Markus Wenzel gondoskodtak.

A fokozatmentes kicsinyítést és nagyítást 16-féle módon lehet újra a képernyőre tenni. A Giga Paint az első olyan C64-es program, amely alkalmas arra, hogy több tetszőleges ponton keresztül szabályos görbét rajzoljon. Érdekes hatások érhetők el a párhuzamos, az X, illetve Y tengelyen tükrözött rajzokkal. A munkát segíti a három programozható tabulátor és az ugyancsak tetszőlegesen előre szabályozható kurzorléptetés. A koordinátakijelzés tized milliméteres finomságig állítható.

Az Amica Paint multikolor képeit a saját nyomtatóprogramja tetszőleges árnyalatokkal jeleníti meg



Különleges hatások érhetők el a matematikai úton megvalósítható torzításokkal és véletlen pontszóródással. Az ezekkel való bánás kicsit ugyan nehézkes, de a végeredmény megéri a fáradozást, sőt ezek még a hagyományos rajzolófunkciókban is működnek. A matematikában kevésbé járatosakat a könyv példafüggvényekkel segíti. Nemcsak a sok minta valamelyikével rajzolhatunk az állítható méretű esetekkel, de ezeket tetszés szerint szaggathatjuk is.

Természetesen ez a sok minden nem fér be egyszerre gépünk memóriájába, ezért néha valamiről le kell mondanunk (például az egyik „rajztábláról” a négyből), illetve a programcsomag bővítései választhatóan tölthetők be. Kicsit nehézkes a színek használata, és lapzártáig nem sikerült kibogoznom egy kielégítő undo-funkciót sem (lehet, hogy nincs is?). Ezekről eltekintve a program és a leírás a legmagasabb pontszámot érdemli a C64-es professzionális rajzoló kategóriában.

A programcsomag szerves része néhány kiegészítő program. Így a maga nemében kiemelkedő a Giga Print nyomtatórutin, amely a szabadon meghatározható árnyalatokkal több mint 120 különböző típusú mátrixnyomatóval teszi lehetővé a választható terület állítható pontméretű és felbontású kinyomtatását. A pontméret 1–99-ig állítható, így több részletben akár egy hatalmas plakát is készíthető. A Giga Font lehetővé teszi, hogy a Giga Painttel előállított betűkészletet a nyomtatóba tölthessük. A Giga Grabber segítséget nyújt ahhoz, hogy grafikákat, betűket és sprite-okat átvessünk más programokból. A Giga Basic sok hasznos BASIC-kiegészítéssel is szolgál, amelynek az előnyeit két demo-program, a Diashow és a Puzzle mutatja be.

A Markt & Technik kiadó újdonságai azt bizonyítják, hogy a „jó öreg” 64-est még nem kell elfelejteni. Jó lenne, ha ezek a kiváló programok mielőbb törvényesen minden érdeklődőnek hozzáférhetővé válnának hazánkban is.

HUPPÁN BÉLA

(Szerk. megj.: a színes képösszeállításunk a következő számunkban közöljük.)

Sorozatunkban azokat az új hardver- és szoftvertermékeket ismertetjük, amelyek várhatóan általánosan elterjednek, és meghatározó szerepük lesz a fejlődés irányainak kialakításában.

Plotterek II.

Advance Bryans Instruments

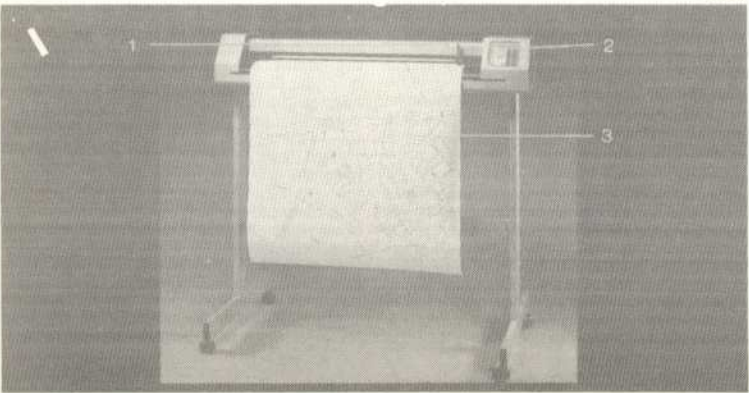
A 6600 Series (1. kép, 1. táblázat) egyes tulajdonságai, például a karakterkészletek száma, jobb, mint az előzőké. Különlegessége, hogy nagyítani is tud. A cég az előnyös tulajdonságokat a saját, tudniillik a plotter 68000 sorozatú 16 bites processzor számítási teljesítményének javára írja. A 6700, 6800 sorozatúak (2. kép, 2. táblázat) kezelő billentyűzete és számos azon keresztül elérhető funkciója kedvezőbb. A megjegyzések a képhez: 1 — akár húsz toll is lehet (!); 2 — a billentyűzet skálázást, öntesztet, forgatást, rajzterület-letakarást is lehetővé tesz; 3 — a papír 952 x 2057 mm is lehet.



1. táblázat

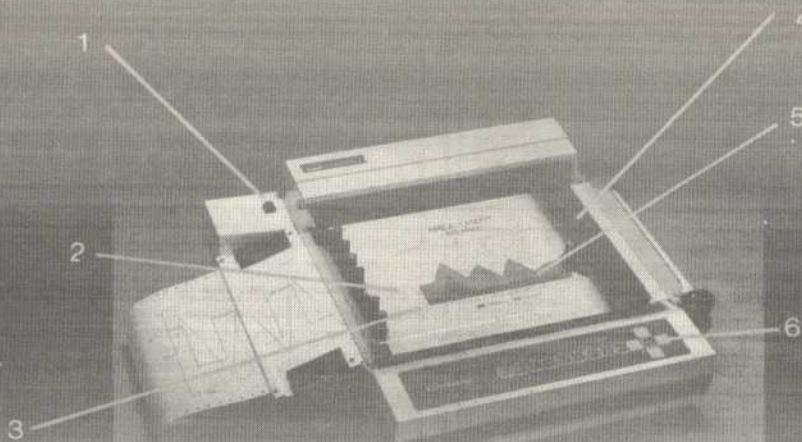
Rajzterület	ISO A4—A1
Tollszám	8
Tollcserehiba	0,15 mm
Felbontás	0,025 mm
Ismételhetőség	0,05 mm
Pontosság	0,1%
Gyorsulás	1 g felett
Sebesség	40 cm/s
Tollpozicionálási sebesség	55 cm/s
illesztő	RS232C vagy IEEE 488
Beépített intelligencia	Különböző vonaltípusok, ívek, körök, oszlop/kör diagramok, ablak, skálázás
Beépített program	HPGL
Karakterkészlet	ötféle
Karakterforgatási pontosság	jobb, mint 1°

2. kép



A 6100 sorozat A4, A3 méretű rajzokat készíthet. Ennek megfelelően szerényebb célú és árú készülék (3. kép). Mivel Centronics csatlakozójú, ezért párhuzamos csatornán is kezelhető. Kompatibilis például a Lotus 1—2—3 és a Symphony programmal. A megjegyzések a képhez: 1 — kétotollas; 2 — beépített tollváltó/tároló rendszer; 3 — 0,05 mm-es felbontás, 20 cm/s írási sebesség; 4 — a porvédő egyúttal zajcsökkentő is; 5 — igen kisméretű; 6 — minden lényeges funkció billentyűvezérelt.

A 6300 sorozat lefutó regisztrátumok készítésére is alkalmas (4. kép); vagyis rollnit is használhat. Mivel ezzel ipari regisztrátum készítésére jó, ezért ellátták a mérőműszerek ipari szabványos csatlakozójával, az IEEE 488 szabványúval. Szoftverkompatibilis az 1—2—3-mal és az AutoCaddel. Se-



4. kép

bessége eléri a 6600 sorozatét. A megjegyzések: 1 — 50 egymást követő rajzpéldányt képes rajzolni; 2 — maximum tíz toll lehet; 3 — 0,025 mm felbontás, folytonos vonal rajzolója; 4 — HPGL (Hewlett-Packard Graphic Language) kompatibilis; 5 — A3 méretig; 6 — számos beépített funkció (például rajzforgatás, hibakeresés).

A Unique 2000 fotoplotter egy A3 méretű eszköz, amivel fényérzékeny filmekre lehet fénytollal rajzolni. Csak a filmet kell behelyezni sötétszobában, a rajzoltatás már világosan történhet. A cég szerint ez az első olcsó fotoplotter.

2. táblázat

Rajzméret	ISO A4—A0+ (962 × 2057 mm)
Tollszám	20
Felbontás	0,025 mm
Mechanikus felbontás	0,025 mm
Ismételhetőség	0,063 mm
Sebesség	71 cm/s
Illesztő	RS232C
Átviteli sebesség	300—9600 baud
Beégetett program	DM/PL
Puffertároló	14 k (opcionális 1 M)
Egyéb	Beállítható rajzforgatás, rajzterület, origó, önteszt, pontosság-teszt

3. táblázat

Rajzterület	ANSI C—D ISO A2—A1
Tollszám	8
Tollváltás	0,1 s
Felbontás	0,025 mm
Mechanikai felbontás	0,013 mm
Ismételhetőség	0,1 mm
Pontosság	0,2%
Sebesség (toll fel)	40 cm/s
Sebesség (toll le)	50 cm/s
Gyorsulás	2 g
Illesztő	RS232C, IEEE 488
Puffertároló	7448 bájt

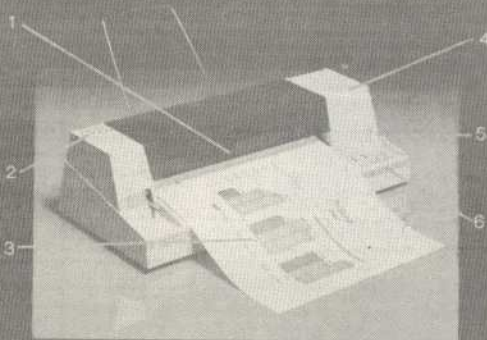
Hewlett-Packard

A mérés- és számítástechnika talán leginkább a minőségre, a teljes rendszerek kialakítására törekvő cége a névadó két alapító oszcillátorával indult ötven éve. A folytatás a pH-mérő volt, aztán mindenfajta kémiai laboratóriumi műszer következett, majd ezek kiegészítése számítástechnikai háttérrel: először számológépekkel, végül számítógépekkel. Később a kiegészítőből főtermék lett. Jelenleg a számítógéprendszerek — a perifériákat is beleértve — uralkodnak a cég termékkatalógusában. (Egyébként ez a cég adta Bill Hewlett személyében az első milliárdost is a számítástechnikának.) Ők dolgozták ki az IEEE 488 mérővonal szabványt, a plotterek gyakorlatilag szabvánnyá vált nyelvét, a HPGL-t, a legismertebb lézernyomtatót stb.

A DraftPro típust (3. táblázat) a különlegesen nagy (20) karakterkészlet és a jó átlagos paraméterértékek jellemzik.

SIMONYI ENDRE

3. kép





OKTATÁSI IRODÁJA

**az 1988/89-es tanévben a következő
mikrogépes tanfolyamokat indítja:**

MIKROGÉPES MŰSZAKI TANFOLYAMOK

A tanfolyam(ok) megnevezése	Tartama (nap)	Ára (Ft)	Időpontok 1988/1989
IBM PC/XT áramköri elemek	5	5500	szeptember 12—16. december 12—16.
XT/AT elemkészletének előírásai	2	2300	szeptember 19—20. december 19—20.
IBM XT felépítése és karbantartása	5	7500	szeptember 26—30. november 14—18. január 02—06.
IBM AT felépítése és karbantartása	5	7500	október 10—14. november 21—25. január 16—20.
IBM XT hibakeresési és javítási módszerek	5	9000	október 03—07. nov. 28.—dec. 02. január 09—13.
IBM AT hibakeresési és javítási módszerek ÚJ	3	6500	november 08—10. január 23—25.
16 és 32 bites szuper-mikroprocesszorok	5	5500	október 17—21.
Az INTEL mikroprocesszor-család új elemei	5	5500	november 14—18.

Tanfolyamszervező: Gombos Péter, tel.: 853-111/154, 237.
Tanfolyamfelelős: Gerő Judit, tel.: 851-294, 853-111/238.

MIKROGÉPES GÉPKEZELÉSI ISMERETEK

A tanfolyam(ok) megnevezése	Tartama (nap)	Ára (Ft)	Időpont(ok) 1988/1989
IBM XT és kompatibilis gépek (MIKROSZTÁR 16, CONTROLL MCS6, P-16, COMMODORE PC10/20, stb.)	5	7450	szeptember 12—16. november 14—18. január 09—13.
IBM AT és kompatibilis gépek (MIKROSZTÁR AT, CONTROLL MCS7, P-16M, VICTOR V260 stb.)	5	7450	október 10—14. december 05—09. február 06—10.

Tanfolyamszervező: Dombroski Erzsébet, tel.: 853-111/154, 237.
Tanfolyamfelelős: Gerő Judit, tel.: 851-294, 853-111/238.



**Cím: Számítástechnika-alkalmazási Vállalat
OKTATÁSI IRODA
Budapest XI., Szakasits Árpád út 68.
Levél cím: Budapest 112. Pf. 146. 1502
Telex: 22-4498**

A Mikroszámítógép Magazin

1983 decemberében jelent meg először. Ha Ön csak a közelmúltban csatlakozott olvasóink táborához és szereti a lapot, biztosan szívesen beszerezne régi számait. Erre már kizárólag a lapfolyóirat Neumann Társaságnál van lehetőség.

Sajnos, hiánytalan sorozat összeállítására már nincs mód, de jó néhány szám példányai korlátozott számban még megvásárolhatók nálunk.

Kaphatók: 1984. 4. szám
1985. 2., 4., 5. szám
1986. febr., márc., máj., júl., aug., okt., nov., dec.
1987. 1., 2., 5., 7., 8., 9., 10., 11. szám
1988. 1., 3., 4., 5., 6. szám

Címünk: Budapest V., Báthori u. 16. 1054

Vidéki olvasóinkat kérjük, hogy levélben rendeljék meg a kért számokat. Csekket küldünk részükre, amelynek beérkezése után postázzuk a lapot.

NJSZT Titkárság

BEHARANGOZÓ

Napjaink igen kedvelt témája a számítástechnika, nagyon sok rendezvény foglalkozik vele. Ezekhez csatlakozik a II. Nyiregyházi Szervezési és Számítástechnikai szimpózium is, garantáltan izgalmas témákkal. Az eszmecsere mintegy hatvan előadót hívtak meg.

Három szekció lesz:

- Az adattárolás és kommunikációs technikák szekció dr. Majtényi György vezetésével dolgozik;
- A termelési informatikai rendszerek csoportot dr. Tóth Tibor irányítja;
- A számítógépes műszaki technológiák szekciót dr. Horváth Imre fogja össze.

Az előkészítő bizottság úgy tervezi, hogy évente egy alkalommal ezekben a témákban fórumot teremtsen az érdeklődőknek az SZVT és NJSZT támogatásával. **1988-ban november 9—12. között várják a részvevőket.**

Részletes információt, névre szóló jelentkezési lapot Koltai Magdolnától lehet kérni a 06 42 13 474 telefonon. Bemutató kiállítás tartásához a rendezők területet adnának.

A KLUBLAPOK ÍRÁJK

A program segítséget nyújt a „Trojai faló” (más néven vírus) típusú programok ellen, amelyek a merevlemezre lévő programokat törlik, felülírják vagy akár formattálják is. A HD - (hard disk - merevlemez) védelem program — futtatása után — csak fájlok futtatását és betöltését engedélyezi, de minden merevlemezre vonatkozó író, formattáló utasítást megtagadja. Hibáuzenetet jelenít meg a képernyőn, illetve az aktuális kimeneti eszközön, és mintha a parancsot tökéletesen végrehajtották volna, a parancs visszatér. Így láthatóan vélik a „gyilkoló program” üzenete, miután a program azt hiszi, hogy a formattálás vagy írás kész.

Indítás után a program átírja a DOS megszakításvektorát, ami ezek után magára mutat, az eredeti mutatókat pedig elmenti. Ha olvasási utasítás volt, ezekre ugrik, ha nem, hibáuzenet után visszatér a felhasználói programra. Eképpen lehetővé teszi azon programok elleni védelmet, amelyek egyenesen a megszakításvektor által mutatott helyre ugranak. Nemcsak a DOS normál megszakításvektorát, de a közvetlen szektoriró rutinjának megszakításvektorát is felülírja, ami által a bonyolultabb speciális programok is hatástalanok.

A program indításától kezdve a DOS következő betöltéseiig működik. Ajánlatos egy külön lemezre ezt a programot

```
DUMMY PROC FAR
SUB AX, AX :0 legyen AX-ben
RET
DUMMY ENDP
```

;; Ez a rutin állítja át a megszakításvektorokat, és menti el a régi

```
INIT PROC
SHOW COPR: MOV AH, 9
LEA DX, COPYRIGHT
INT 21H
GET INT 13: MOV AX, 3513H
INT 21H
MOV WORD PTR OLD 13, BX
MOV WORD PTR OLD 13+2, 0
SET INT 13: MOV AX, 2513H
LEA DX, SENTRY
INT 21H
SET INT 26: MOV AX, 2526H
LEA DX, SENTRY
INT 21H
INIT RET ENDP
```

```
CSEG ENDS
END 60
TITLE TITLE — HARD DISK VÉDELEM
```

```
CSEG SEGMENT BYTE PUBLIC
PUBLIC MAINCODE, GO, COPYRIGHT, ALERT MSG, OLD 13
PUBLIC OLD 26, INSTALL, TSR, SENTRY, WICH DISK, OK
```

VÉDELEM A „GYILKOS” ELLEN

MEREVLEMEZ-VÉDELEM PC-RE

felvanni, és a lemez AUTOEXEC.BAT fájljában erre a programra ugrani. Így új, ismeretlen programok indítása előtt helyre ugranak. Nemcsak a DOS normál megszakításvektorát, de az ismeretlen program már nem teheti tönkre a merevlemez.

A program assemblerben íródott, betöltését egy assembler editor segítségével végeztethjük el. (Megjelent a Valley Computer Club Newsletter 1987. 8. számában.)

KARUCZKA GÁBOR

```
PUBLIC ABORT, DUMMY, INIT, SHOW COPR, GET INT 13
PUBLIC SET INT 13, SET INT 26
MAINCODE PROC
ASSUME CS:CSEG
ORG 100H
JMP INSTALL
INT 13, 10
DB 0B : HARD DISK SENTRY
DB 0B : (C) Andrew M. Fried
DB 0B : Működés alatt a HD-t
DB 0B : károsítani nem lehet
DB 0B : $$$ FIGYELEM $$$ A lemezre
DB 0B : írás megakadályozva! $$$
DB 0B : Az eredeti interrupt 13H
DB 0B : Az eredeti interrupt 26H
ALBERT MSG DB 13, 10, 07,
OLD 13 DD 0
OLD 26 DD 0
```

```
INSTALL: CALL INIT ; Program inicializálás
TSR: LEA DX, INIT
MOV CX, 4
SHR DX, CL
MOV AX, 3100H
MAINCODE: ENDP
```

;; Ez az új INT 13H rutin . Ha nem a HD-re vonatkozik az utasítás . vagy pedig megengedett utasítás érkezik . akkor az eredeti kezelő rutint indítja . Ha a HD-re vonatkozik . író vagy formattáló utasítást talál . kírja a hibáuzenetet . majd visszatér az eredeti programra a hibakód törlése után .

```
SENTRY PROC
WHICH DISK: CMP DL, 80H ; HD-re vonatkozik?
JB OK ; ha nem vissza
CMP AH, 3 ; ez egy írás kód?
JE ABORT ; ha igen megállítani
CMP AH, 5 ; ez egy format kód?
JE ABORT ; ha igen megállítani
CMP AH, 0BH ; ez egy FD kód?
JE ABORT ; ha igen megállítani
OK: JMP DWORD PTR OLD 13 ; eredeti kezelő rutinra ugrás
ABORT: PUSH AX ; regiszterek elment.
PUSH DS
MOV AH, 9 ; DOS print kód
PUSH CS ; DS reg = CS reg
POP DS
LEA DX, ALERT MSG
INT 21H ; hibáuzenet címe
POP DX ; print rutin hívás
POP DS ; regiszterek vissza
SUB AX, AH
IRET ; visszatérés
SENTRY ENDP
```

;; Ez az új INT 26H. Hívása esetén törli az AX regisztert, ami a hibakódot jelentené és visszatér

ADOM A MAGYARÁZATOT!

Az előző havinál kissé részletesebb, példáival is alátámasztott magyarázatot küldött egy olvasónk:

A b változó dupla pontosságú definiálása még nem jelenti azt, hogy a funkcio is — ebben az esetben exp(1) — dupla pontossággal lesz kiszámítva. Joggal merül fel a kérdés, vajon miért? A probléma gyökere a számbárázolásban keresendő. A GW—BASIC kétféle módon dolgozza fel a számokat, mégpedig aszerint, hogy miként definiáltuk. Az alábbi ábra segítséget nyújt ennek megértéséhez:

```
Egyszeres pontosságnál
Bájt 0 Bájt 1 Bájt 2 Bájt 3
loman S:himan exp
```

Kétszeres pontosságnál

```
Bájt 0 Bájt 1 Bájt 2 Bájt 3 Bájt 4 Bájt 5 Bájt 6 Bájt 7
loman S:himan exp
```

ahol: loman a mantissza első bájtját, S az előjelet, h a mantissza felső bájtját, exp a kitevőt jelenti.

Rögtön világos, hogy eredményünk miért hibás. A szám nem dupla pontossággal volt megadva, és így a dupla széles változó csak egy egyszerű számbárázolás számot kapott. A megoldás: B = EXP(1 #).

Erdemes még kipróbálni a következő példákat, ami igazolni fogja a fenti állításokat:

```
10# = 6# /7.1# :b# = 6# /7.1#:print a#, b#
```

KOVÁCS ANDOR GYÓZÓ NSZK

KI AD MAGYARÁZATOT?

ISMERETLEN 6500 KÓDOK

Melyik utasítás mit csinál, és milyen regisztereket érint, valamint hogyan állítja a jelzőbitekét?

Az itt felsorolt nevek (mnemonikok) az F—MON 1541, az operandusok a lap 1986. augusztuszi számának 24—25. oldalán közölt táblázat formátumában lettek lejegyezve. A zárójelben szereplő decimális számot jelentenek!

1, \$93 (147) AX Y (zp), y
2, \$0B (11) ANN zp (E két kód azonos funkciójú az F—MON 1541 szerint.)

3, \$2B (43) ANN zp (Ez az AND + op LSR A programmal egyenértékű, ez tehát nem feladat, csak tájékoztatás.)

4, \$4B (75) ANL#
5, \$6B (107) DAR#
6, \$8B (139) TAN#
7, \$9B (155) AXS absz, y
8, \$9C (156) YXA absz, x
9, \$9E (158) YXA absz, y

Az utóbbi két utasításról a RAT—MON 64 azt állítja, hogy egyenértékűek az STY absz, y utasításokkal, tapasztalat szerint azonban ez nem igaz.

10, \$9F (159) AYY absz, y
11, \$AB (171) ??? (*) (Ez az egyetlen kód, amelyre az F—MON is kérdőjellel válaszol, azonban a disassemblerében kutatva — a címzés-mód táblázatban — az F—MON mégsem az egybájtos

utasításokhoz sorolja, hanem a közvetlen címzésűekhez. Ez igaz lehet, lásd később.)

12, \$8B (187) TSA absz, y (Környezetét tekintve nagyon valószínű, hogy a veremmutatóval — is — operál.)

13, \$CB (203) XAS#

Ugyancsak megjegyzésként említem a

14, \$EB (235) SBC #+-. (Ez az utasítás teljesen egyenértékű a \$E9 kódú utasítással.)

Azt, hogy a címzés-mód helyesen állapította meg az F—MON 1541 két szerzője, onnan lehet — többé-kevésbé — biztosan megállapítani, hogy az S—MON (nem az S—MON +) lépésenkénti üzemmódban az általa definiálatlanul itelt utasításhoz érve végrehajtja azt, a szükséges értékkel növelve a programszámlálót, megfelelő értékre állítva a jelzőbitekét. X-szel, ill. Y-nal létrehozott ciklusokkal kísérletezve az F—MON címzés-módjai jók. Így például az \$AB kódú utasítás elérésekor is kettőt lépett a programszámláló.

Kérdés továbbá, hogy a 8502-es processzoron nem mások-e az új utasítások? Az a tapasztalat, hogy például egy alkalommal az Y-nal képzett ciklust a gép összehasonlíthatatlanul lassabban hajtotta végre definiálatlan utasítással, mint a „hagyományos” utasítással. Miért? Az új utasítások nevei esetleg támpontot adhatnak megértésükhöz.

Pápai Ákos

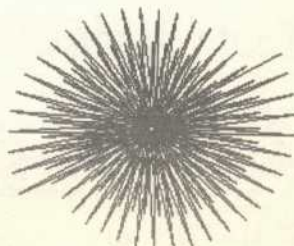
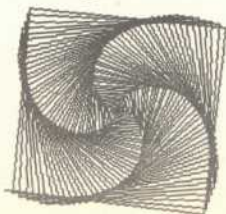
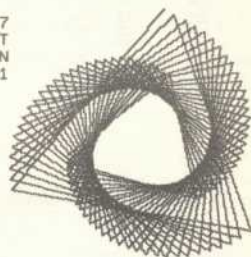
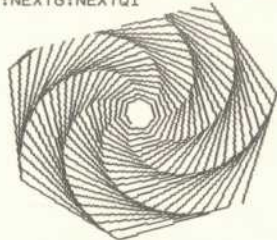
A RAINBOW® nevű amerikai szaklap minden számában közli az olvasók által beküldött legerdekesebb egy- és kétsoros programokat. Ezek a programozók valódi ügyességpróbái, hiszen legfeljebb 255 karakternyi helyet kell a lehető legjobban kihasználniuk (a lap a Tandy cég CoCo típusú számítógépeivel foglalkozik, és az ezekre írt Microsoft BASIC változat ennyi karaktert fogad el egy sorban). Az már szinte ráadás, hogy ezek a programok gyakran még jól használhatók is.

A bemutatott rajzokat nézve az olvasók bizonyára úgy gondolják, hogy ilyeneket bármikor elő tudnak állítani. Mégis mi akkor ebben az érdekes? Két dolog. Az egyik az, hogy ezeket — és összesen százféle hasonlót — egyetlen program állította elő. A másik, hogy a program a *listában* közölt nyúlarknyi volt.

Azoknak, akik a programot saját gépükön kívánják megvalósítani, közöljük az átiráshoz szükséges tudnivalókat. A LINE(X1,Y1)—(X2,Y2), PSET utasítás jelentése: húzz egyenest az (X1,Y1) koordinátájú ponttól az (X2,Y2) koordinátájú pontig. A PMODE4 utasítás a legnagyobb felbontású grafika bekapcsolója. A PCLS letörli a bekapcsolt grafikának megfelelő területet. A SCREEN1,1 a képernyőre viszi a grafikus képet. A LINE—(X2,Y2),PSET utasítás annyiban különbözik az imént ismertetett egyenest rajzolótól, hogy a korábbi egyenesek utolsóinak felrajzolt végpontját veszi kiindulási pontnak. S.E.

EGYSOROSOK

```
0 V=2.6:FORQ1=1TO100:P=90:LINE(79,15)-(79,15),PSET:PMODE4:PCLS:SCREEN1,1:FORX=1TO881:STEPV=P-.5:K=128-COS(X)*P:L=92-SIN(X)*P:LINE-(K,L),PSET:NEXT:V=V+.1:FORG=1TO2000:NEXTG:NEXTQ1
```



Rovatunkban az Apple, Atari, Commodore és Sinclair mikrook tulajdonosait feltehetően érdeklő, angol és német nyelvű cikkekről informáljuk olvasóinkat egy tartalomleíró szöveg segítségével.

A forráshely karaktersorozatát nyílv vezető be, ezt a / jelleg a folyóirat kódja követi (lásd táblázat). A két / jel között a megjelenési adatokat (év, hó), illetve a cikk kezdő oldalaszámát szerepeltetjük. A második / jel után pedig - az esetleges másolatkerést megkönnyítendő - a cikk teljes oldalterjedelmét közöljük.

A folyóiratok megtekinthetők a SZÁMALK (Bp. XI. Szakasits Á. u. 68.), illetve - a x-gal jelzettek - az OMIKK

(Bp. VIII. Múzeum u. 17.) szakkönyvtárában. (A másolás díja oldalanként 8 Ft.)

A Magazin helyi nyári miatt mindössze egyetlen címszű, a programlista közzétételére vállalkozhatott a folyamatosan bővített adatbázisból. A kezdő visszhang alapján az OMIKK háromhavi bontásban kiadja a tartalomleíró szövegek permutálásával és alfabetikus rendezésével szerkesztett teljes anyagot. Az "APACS Mikroindex" első füzeté már megjelent, ára 54,- Ft.

Több példány vásárlása/rendelése esetén 10 darabonként 2 tiszteletpéldányt térítésmentesen ajándékozik a terjesztőknek az OMIKK levélszolgálat (Bp. Pf.: 12. 1428).

PROGRAMLISTA
amiga!antasioc||lepredefinialo fajok k|
onvertisasa data sorokkal demo
->cube/'87.07-74/2

PROGRAMLISTA
amiga!atari st||apple ii||atari xl||xel
c64||meteorologia||idojaras elorejelze
s beirando valtozok alapjan
->cube/'87.07-36/3

PROGRAMLISTA
amiga!jatek||helvetia-game>
->run2/'87.07-78/7

PROGRAMLISTA
amicioc||atari xl||xel||jatekprogram ke
szites||2.resz||karaktermozgatas
->anti/'87.07-28/7

PROGRAMLISTA
apple ii||grafika||keprajzolas||hires f
ajok lepcsozotes megjelenitese
->cube/'87.07-100/4

PROGRAMLISTA
apple ii||grafika||print shop||konverta
las apple hires modushoz
->cube/'87.07-93/5

PROGRAMLISTA
atari st||jatek||quadromania>
->happ/'87.07-65/2

PROGRAMLISTA
atari xl||xel||'lathatatlan' uzenetek f
elvillantasa||ghost writer>
->anti/'87.07-36/6

PROGRAMLISTA
atari xl||xel||jatek||chicken xl>
->anti/'87.07-35/3

PROGRAMLISTA
atari xl||xel||jatek||jump>
->happ/'87.07-101/4

PROGRAMLISTA
atari xl||xel||jatek||red white and blu
e||a hónap legjobb jateka
->anti/'87.07-31/4

PROGRAMLISTA
atari xl||xel||jatekprogram keszites||sh
ape vezeres basicbol
->cube/'87.07-68/7

PROGRAMLISTA
atari xl||xel||nyomatasi||hires keprajo
las karakterekkel||ascii art convert
er> ->anti/'87.07-20/5

PROGRAMLISTA
c128||filekzeses||lemezegység||1571||r
elativ fajok beszelese
->cube/'87.07-109/4

PROGRAMLISTA
c128||grafika||jatekprogram keszites||
paintmaster> ->run2/'87.07-108/3

PROGRAMLISTA
c128||programfajok osszekapcsolasa||a
hianzo 'append' parancs potlasi
->run/'87.07-72/3

PROGRAMLISTA
c16||intelligens irogepkerit valo muko
dtes> ->chip/'87.07-134/1

PROGRAMLISTA
c64||eproma egyetho pr.promorito
->happ/'87.07-52/5

PROGRAMLISTA
c64||hibasuro 'input' rutin||jelzes a
kiakadas helyett||alkalmazasi utmutat
o> ->run/'87.07-76/4

PROGRAMLISTA
c64||jatek||krieg der sterne>
->run2/'87.07-127/6

PROGRAMLISTA
c64||jatek||motocrash>
->happ/'87.07-51/1

PROGRAMLISTA
c64||jatek||tri-solitair>
->run/'87.07-56/3

PROGRAMLISTA
c64||lemezarkatalogusnyomatasi leme
zboritokra> ->run2/'87.07-119/4

PROGRAMLISTA
c64||katas||matematika||alapsuvelete
k gyakorlati> ->run/'87.07-81/3

PROGRAMLISTA
c64||programiras||basic 7.0 funkciok b
epitese> ->cube/'87.07-67/7

PROGRAMLISTA
c64||programiras||basic-sor atszamozas
->64ez/'87.07-61/1

PROGRAMLISTA
c64||programiras||forralista formatal
as> ->run2/'87.07-101/2

A folyóirat neve Kódja

x 64'er Magazin
Antics
x Chip Magazin
x Compute!
x Dr. Dobb's Journal
Elektor Electronics
Happy Computer
x mc - Zeitschrift
Run (USA)
Run (NSZK)
x Your Computer
x ZX Computing Monthly

64er
anti
chip
cute
dobb
etor
happ
mc
run
run2
your
ZCOM

PROGRAMLISTA
c64||programiras||szovegsor-gorgetes
->run2/'87.07-107/1

PROGRAMLISTA
c64||szines nagybetus szovegsor gorse
beselutmutato a szovegbeirashoz
->cube/'87.07-105/3

PROGRAMLISTA
spectrum||felhasznaloi karakterkeszite
t generalisa> ->chip/'87.07-120/5

PROGRAMLISTA
sprite||c64||jatekprogram keszites||saj
at sprite/karakter beszites idegen
programokba> ->run2/'87.07-111/7

PROGRAMLISTA
sprite||c64||jatekprogram keszites||ter
vezes kepnyon billentyukkel||sprite
e designer> ->cube/'87.07-76/6

PROGRAMLISTA
statiztika||c128||grafikonkeszites||ko
rszeletkeszito rutin
->run/'87.07-92/2

PROGRAMLISTA
statiztika||c64||oszlopdiaagram keszit
es> ->64er/'87.07-69/1

PROGRAMLISTA
szovegfeldolgozas||adatszer||c64||hete
rogen halozati filekonverzio||c64||lba
||lba||magyar elektronika||88/1 szam
(teljes forditas)> ->64er/'87.07-47/7

PROGRAMLISTA
szovegfeldolgozas||atari xl||xel||basico
s illeszttho szubrutin 80 oszopec
modushoz> ->anti/'87.07-24/5

PROGRAMLISTA
szovegkezes||adatvitell||atari xl||xl
e||test file splitter>
->cube/'87.07-91/2

PROGRAMLISTA
szovegkezes||c64||uzenetkeszites lem
ezre||sya 49152-vel indithato zenemul
ok hasznalata> ->64er/'87.07-62/4

PROGRAMLISTA
visszamenetes||c64||renew generator>
->happ/'87.07-57/1

PROGRAMLISTA
zve||c64||jatekprogram keszites||soun
dmaker> ->run2/'87.07-103/4

A Fővárosi IV. és XV. kerületi Ingatlankezelő Vállalat

felvesz

VI—160 PC IBM kompatibilis mikroszámítógépeikhez

programozókat.

Jelentkezni lehet: részletes önéletrajzzal, személyesen a vállalat Személyzeti és Oktatási Osztályán.

Cím: 1042 Budapest Munkásotthon u. 66—68. II. 212.

Minden kedden 17-től 20 óráig
HCC ENTERPRISE klub

a VSZM

Közösségi Házban

(Bp. XI., Fehérvári út 120.)

Klubvezető: Romvári Gábor

Telefon: 810-950/473

„Jóirodalom” a Mikroszámítógép Magazínban

Az ötlet jó egy éve merült fel egy szerkesztőségi beszélgetésen. Valamennyien egyetértettünk abban, hogy sem az olvasók jelentős részétől, sem a lapotól nem lenne idegen, ha a lap — mértékkel — nemcsak (vagy nem teljesen) műszaki jellegű, jó írásokat is közölne. Bár e rovat írója nem ragaszkodott volna mereven ahhoz, hogy minden ilyen cikkben legyen valami *műszaki* jelleg is, végül (majdnem) beadta a derekát. Első lépésben elkészült egy válogatás J. Updike magyarul még, tudomásunk szerint, nem publikált, rövid írásaiból. A közlés jogát megszereztük. Ennek eredményeképpen nyújtunk át most olvasóinknak egy újabb novellát. Fogadják szeretettel. Előbb azonban egy „szakmaibb” kiterőt teszünk.

Ki vagyok én?

A számítástechnikával behatóan foglalkozók (is) előbb-utóbb szembe találják magukat ma (még?) megválaszolatlan kérdésekkel: mi az, hogy „gondolkodni”, mi az, hogy „tanulni”, mi a „személyiség” stb.? A számítógépek kapcsán azért merülnek fel ezek a kérdések gyakrabban, mint más gépek — például a mosógépek, a lemezjátszó „gépek” stb. — kapcsán, mert a számítógépekben szinte korlátozás nélkül *modellezni* tudjuk a világot (a tárgyakat, a fogalmakat, a világban lezajló folyamatokat) — úgy is mondhatjuk, a gépek belsejében *tákróztani* tudjuk a világot a maga szinte teljes komplexitásában. Mondhatnánk-e a korlátlan lehetőségek képzetével a fejünkben, hogy gépeink akár önálló, saját *személyiséggel* is rendelkezhetnek. A válaszhoz persze először át kellene tudni, mi az, hogy *személyiség*. Douglas R. Hofstadter számítógép-kutató és Daniel C. Dennett filozófus 1980-ban össztalálkozott Palo Altóban. Beszélgetésükből egy szép könyv született: „The Mind’s I” (Bantam Books 1981., 501 oldal), amely a címe szerint „Fantáziázások és tündöklések a személyiségről és a létről”. A könyv részben nevelés iránt, tudósok (Borges, Turing, Lem, Dawkins és mások) témába vágó, válogatott írásainak gyűjteménye. A két szerző ezeket fűzi saját gondolatait az emberi (a természetes) és mesterséges intelligenciáról.

E rovat gyakorlatához híven a könyvre egy gondolat felvillantásával kívánjuk felhívni olvasóink figyelmét. Ezt úgy választottuk meg, hogy valamennyire rimelyen a cikkünk második részében teljes egészében közreadott Updike-írára.

Szóval következék kedvesinálónak egy gondolat kísérlet ismertetése és a hozzá kapcsolódó tündős Hofstadter és Dennett könyvének előszavából: Úrhajónk szerencsésen landolt a Mars gyilkos klímájú felszínén. Közben megrongálódott, hazatérésre alkalmatlanná vált. Kész, vége: „Isten veled, család, Isten veletek, barátok”. De mégis, van valami remény. Az űrhajón van egy *Teleclone Mark IV* típusú *árvívó szerkezet*, meg egy hozzá való használati utasítás is. Ha bekapcsolod ezt a gépet és ráirányzod a földi Teleclone-vevőre, majd belépsz a Teleclone-adó kamrájába, akkor ez gyorsan és fájdalommentesen molekuládra szed széd téged, és a molekuláid részletes és pontos „térképét” a földre sugározza. A földi Teleclone-vevőnek hatalmas atom- és molekulakészlete van. Ezekből a földre sugárzott „térkép” alapján azonnal (mondjuk stílusosan: real time) rekonstruál téged. Ha fénysebességgel hazérkeztél, kilépsz a Teleclone-vevő kamrájából — egyetlen szerzetted karjai közé, akik nemskára ámulva fogják halgatni újratúsod fantasztikus történetét.

Szóval egyszerűen kilépsz a földi Teleclone-vevő kamrájából a kellemes földi atmoszférába. Megállapítod, hogy az eltelt három év alatt mindenki megváltozott egy kicsit. Lányod időközben kamasz lett. Lehetséges-e, hogy ő az, aki még nem is olyan rég az ódében üldögélt? De tényleg, azóta, hogy utoljára láttad a lányt, szinte minden egyes sejtje, atomja, molekulája kicsérélődött, elmozdult eredeti helyéről, de az is kétségtelen, hogy a drámai kicsérélődések ellenére ő mégis ugyanaz a személy, akitől annak idején elbúcsúztál. Ekkor eszedbe ötlék: és te, vajon te is ugyanaz a személy vagy-e, aki ettől a kislánnyal elbúcsúztál? Talán, bár *valaki* meghalt ott a Marson, a Teleclone-adó „szétszerező” kamrájában. Te vagy-e az, aki a Teleclone segítségével túlélted a Mars-beli kalandot, vagy valaki más? És a

lányod mit gondol majd, amikor megtudja, mi is történt valójában? Elfogad-e téged szülőjének továbbra is a „telecloneozás” után?

Kicsit általánosabban: mitől marad meg valaki az összes alkotórészének kicsérélődése után is ugyanannak a *személyiségnek*? Hol „székel” a személyiség? Az agyban? És ha egyszer az orvostudomány képes lesz *agydátálerezre*? Más tessetel te ugyanaz a személyiség maradnál-e, aki voltál? Az alábbi két mondat közül vajon melyik az „igazibb”?

Nekem agyam van.

En tulajdonképpen az agyam vagyok. (Az énem az agyam.)

Mellesleg: az emberek nem mindig tudták, mi célú szolgálat az agy. Azt mondják, Arisztotelész még azt hitte, az agy arra való, hogy lehűtse a vért. Ígáz-e, hogy minden, ami bennünket körülvesz, teljesen azonos az alkotórészek összességével? Tétélezzük fel, hogy tökéletesítjük a Teleclone-t és a Teleclone V — mondjuk — már úgy el tudja készíteni a „molekulatérképünket” (összeszerelési utasításunkat), hogy nem semmisít meg közben bennünket: aki belép az adó kamrájába, azt csak „átvilágítja”, „letapogatja”. Ekkor mi lenne? Előáll talán ugyanaz a személyiség két példányban. És meddig lenne *azonosnak* tekinthető a kettő? Nyilván nem sokáig. Memóriájuk hamarosan más-más információkkal tölődne fel.

Mi jön ki abból, a két tudós (egy számítógépes és egy filozófus) — segítséggel hívja mások munkáit — nekik áll a személyiségről és a lélekről gondolkodni, beszélgetni, tündönni? Egy szép és érdekes könyv.

—KE—

J. Updike Az ártatlanság bolygója

Valamikor, úgy 1999-ben a kutatók felfedezték, hogy a Jupiter félig cseppfolyós, óriás, meleg, gomolygó ködében egy minden szempontból kellemes kis bolygó sodródik argon égbolygaitól és csillagok, olvadó berilliumtengeresekével. Az új világ partjaira érkező első földlakók megdöbbenve azt ott lakók minden szegény nélküli mezőtelensége. Nemcsak hogy mezeitelenek voltak — testük hosszirányban barázdált, enyhén domborodó, gyöngyszínű, eloxált konzervdobozszerű henger volt, melyből, helyváltoztatás céljából, hat fogpiszkáló vékonyaszerű vétag állt ki, meg még egy folyhoszomponos hetedik is (az idegi funkciók ebben koncentráálódtak) —, de úgy látszott, nincs különbség a nemek között sem. Valójában nem is volt. Valamiféle szórakozott, futólagos eljárással szaporodtak, melyet „bimbózás”-nak neveztek és a Minerva lakói a Kinai — Amerikai Űrjárási Hivatal — a Sino — American Space Agencyből képzett betűszóval: a SASA, klasszikuson nevelkedett hivatalnokai ezt a nevet adták a bolygónak) nem sokat törődtek az egészsel. Nyilvánvaló volt, hogy valahányszor a lábnyomok (vagy inkább a lyukak, mert a helyváltoztatás olyan nyomokat hagyott, mint a sítókak a ropogós hóban) matematikailag elégséges mértékben átlapolódva mélyedtek be a nikkelt és azbeszt keverékből álló talajba, egy új kis konzervdobozoska sarjadzott vagy „bimbózott” ki. Ez az új lény — hijával minden szülői vagy fajfenntartói gondoskodásnak — mintha össze három Minerva-év (ez öt földi hétnek felel meg) alatt érte el végleges méreteit (körülbelül tizenötszöves földi húvelyeknek megfelelő magasságot), akkor türelmetlenül lerázta gyökereiről a nikkelt, és elfoglalta helyét a mezőgazdaság, az ipar, a kereskedelem, a kormányzás valamelyikének gyümölcsöző rutinjában, mivel — akárcsak a földünkön — nagyjából ott is ezek uralták a létet.

A felfedezők erotikus érdeklődése — de ahogy az argonlevegő-készülékek tökéletesedtek, a nagykövetelek, a mindenféle vizsgálatokat végző kutatók, a kalmárlelkű gyarmatosítók is — elképédést váltott ki és félreértésekre adott alkalmat a Minerva-béliek között. A neki erőszakra irányuló kezdeti kísérletek állt voltak sikeresebbnek mondhatók, mint az új világ anyagiul peremhelyzetben levő egyes bennszülöttjeinek későbbi kísérletei, hogy prostitútaikká magukat. A kielégítő kontaktusok hiánya azonban nem akadályozta meg a szülőházuiktól elszakadt földlakókat abban, hogy a Minerva-béliekkel szerelembe ne essenek és ennek következtében ne produkálják a szonettek, az álmatlan éjszakák, a kimerítő levelek, a féltékeny lelkiallapotok és a fülledt álmok szokásos romhalmazát. Ezek a kis kon-

Dap Hartman, holland egyetemi hallgató megpróbálta statisztikai adatok feldolgozásával a sakkjáték törvényszerűségeit feltárni. Ehhez 832 játszma adatait dolgozta fel, amelyeket a következő versenyeken játszottak:

— Hoogovens	nagymesterverseny, 13 forduló
1984	
— Hoogovens	mesterverseny, 9 forduló
1984	
— Hoogovens	nagymesterverseny, 13 forduló
1985	
— Hoogovens	mesterverseny, 11 forduló
1985	

— Hoogovens	nagymesterverseny, 13 forduló
1986	
— Hoogovens	mesterverseny, 11 forduló
1986	
— OHRA	1984nagymesterverseny, 9 forduló
	svájci rendszerben,
— OHRA	1985mesterverseny, 9 forduló
	svájci rendszerben,
— OHRA	1985külön csoport, 5 forduló
	visszavágóval,
— OHRA	1985nagymesterverseny, 9 forduló
	svájci rendszerben,
— Timman—Juszupov	páros mérkőzés 1986

A 832 játszma közül 450-et vittek döntésre. Ebből 278-at világos, 172-t sötét nyert meg, 382 partit pedig döntetlenül végződött.

A sakkprogramozókat segítő statisztikai adatok

Sötét gyalog

Megnyitás	Középjáték	Végjáték
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00		
95 95 75 40 35 10085 95 : 65 65 45 00 05 75 45 45 : 35 35 10 00 05 35 35 25		
05 05 10 35 45 00 10 05 : 25 20 15 55 55 10 50 45 : 35 35 10 20 35 20 35 25		
00 00 15 25 20 00 00 00 : 10 10 15 35 25 15 05 10 : 20 20 30 20 35 25 25 25		
00 00 00 00 00 00 00 00 : 00 05 10 10 10 00 00 00 : 05 05 15 05 05 05 05 05		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 05 05 05 00 00 00 00 05		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00		

Világos gyalog

Megnyitás	Középjáték	Végjáték
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00		
00 00 00 00 00 00 00 00 : 05 05 05 20 20 05 05 05 : 05 05 05 15 15 05 05 10		
00 05 25 75 45 20 00 00 : 20 20 20 35 45 25 05 15 : 35 25 35 20 20 25 20 25		
05 05 05 10 15 00 25 10 : 30 20 20 05 20 45 40 : 30 35 20 05 15 25 35 35		
90 95 70 10 25 75 75 90 : 30 45 35 00 05 45 45 40 : 35 30 20 00 00 35 35 30		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00		

Sötét huszár

Megnyitás	Középjáték	Végjáték
00 45 00 00 00 00 20 00 : 00 05 05 05 05 15 05 00 : 00 00 00 00 10 20 05 00		
00 00 00 15 10 00 00 00 : 00 00 05 35 20 05 05 05 : 05 05 10 25 25 10 05 05		
05 00 35 00 00 05 05 05 : 10 10 35 05 10 35 10 05 : 00 10 15 15 25 25 15 00		
00 00 00 05 00 00 00 00 : 05 05 10 10 10 10 05 05 10 : 10 10 25 25 25 10 10 05		
00 00 00 00 00 00 00 00 : 05 10 05 05 10 05 05 00 : 05 10 10 10 10 10 05 05		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 05 05 10 05 05 05 00		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 05 05 05 05 05 00		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00		

Világos huszár

Megnyitás	Középjáték	Végjáték
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 00 00 05 00 00 00 00 00		
00 00 00 00 00 00 00 00 : 00 00 00 00 00 00 00 00 : 05 00 05 05 05 05 00 00		
00 00 00 00 00 00 00 00 : 00 00 00 05 00 00 00 00 : 00 05 10 10 10 10 10 00		
00 00 00 00 00 00 00 00 : 05 05 05 05 10 05 05 00 : 10 10 10 10 20 10 10 00		
00 00 00 05 00 00 00 00 : 05 00 10 10 10 10 00 05 : 00 10 20 20 20 15 10 10		
00 00 40 00 00 00 40 00 : 00 10 20 05 10 20 10 00 : 10 10 20 20 15 20 10 00		
00 00 00 10 05 00 00 00 : 00 00 00 15 10 05 05 05 : 00 00 10 20 15 10 10 00		
00 40 00 00 00 00 40 00 : 00 05 00 05 05 00 00 00 : 00 00 00 00 10 00 00 00		

A játszmákban 62 965 különböző pozíció jött létre, és ezek közül 2788-ban állt valamelyik fél királya sakkban.

Az ANALYZE nevű program a játszmák betáplálása után különböző szempontok szerint elemezte azokat, és létrehozta a statisztikai adatbankot. Az adatbank többek között tartalmazta minden játszma minden lépése után — azaz 69 965 pozícióban — az illető állás mozgékonyságának, centrumértékének, anyagi értékének nagyságát, a legális lépések számát, a királybiztonságot, királytámadást és számos más tényező pontértékét.

Vizsgálva az értékeléseket, nagyon érdekes és tanulságos az, ahogyan az egyes figurák elhelyezkednek a játék különböző fázisaiban. Kiderül például, hogy a sötét g3 huszár szinte kivétel nélkül az f6 mezőt foglalja el, és azt a legtöbbszor csak a középjáték vége felé hagyja el. Ugyanez mondható el a világos g3 huszárjáról is, ami az f3 mezőn van. A sáncolásban rész nem vevő bástyák pedig csak ritkán kerülnek el az alapvonalról.

A következőkben azt elemezzük, hogy a különböző figurák hány százalékos valószínűséggel találhatók meg az egyes mezőkön a játszma három szakaszában: a megnyitásban, a középjátékban és a végjátékban. Az egyszerűség kedvéért a program a harmincadik félépésig tekinti az állást megnyitásban, középjátékban pedig azokat a pozíciókat számítja, amelyekben a tiszték száma, a királyokat nem tekintve, nagyobb hatnál. Ellenkező esetben az állást mint végjátékot dolgozza fel az értékelő algoritmus.

Mindezek figyelembevételével a szerző a gyalogok és a huszárak elhelyezkedési valószínűségére a bemutatott táblázatokat adja világos és sötét színnel a megnyitásban, a középjátékban és a végjátékban. A táblázatokban ötszázalékos pontossággal, kerekítve tüntették fel az adatokat.

A táblázatokban, ha a vonalak mentén összeadjuk az elhelyezkedési valószínűség százalékában kifejezett értékeit, megfigyelhetjük, hogy nem mindenkor kapjuk meg a várt 100 százalékot. Vannak olyan esetek, amikor már bizonyos gyalogok lecserélődtek, ekkor a 100 százaléknál kevesebbet kapunk, és vannak olyanok is, amikor ütés következtében kettős gyalog keletkezik egy vonalon, ekkor pedig 100 százaléknál nagyobb értéket kapunk.

A táblázatok jól mutatják, hogy melyek a népszerű megnyitások, hogyan alakul a megnyitásban és a középjátékban a világos és a sötét gyalogszerkeztúra, mennyire szeretnek a versenyzők nyílt, illetve zárt állásban játszani. Azt is leolvashatjuk, hogy milyen mértékben aktivizálják a játékosok a huszár a középjátékban. Szembetűnő, hogy a megnyitásban a középjátékkal és a végjátékkal ellentétben sokkal kevesebb mezőnek van zérusnál nagyobb pontértéke, ami azt mutatja, hogy a jó sakkozók a megnyitásban ritkán lépnek ugyanazzal a figurával többször, és ezáltal több mezőt érintenek. A középjátékban és még inkább a végjátékban megfigyelhetjük, hogy az egyes mezőknek nincs olyan nagy értékük, viszont sokkal több mezőn megfordul ugyanaz a figura, azaz sokkal több mezőnek van nullától eltérő értéke.

KOVÁCS P. ATTILA

Az Olvasó írja

Az 1988/7. számban, Az olvasó írja rovatban Négyesi Károly és Négyesi Pál panasza, hogy a nyomtatókba friss festékszalagot kell tenni. Tény és való, hogy a programlisták nehezen olvashatók az újságban. Az is igaz, hogy friss festékszalaggal ez javítható. Ez viszont drága. S mi van akkor, ha program van, de nincs nyomtató?

Erre szeretnék egy javaslatot tenni. Az író-gép manapság eléggé elterjedt dolog. Ha nem is minden lakásban van, azért hozzá lehet jutni. Akad a szomszédnak vagy a munkahelyen, az irodákban. A programokat úgy, ahogy a képernyőn látjuk, le lehet gépelni. Tovább tart, ezt belátom, viszont olcsóbb és a nyomtatás minősége is jobb.

A megoldás nem lenne rossz, ha az írógépen lennének olyan karakterek, amelyeket a forrásprogram leírására használni kell. De hát sajnos nincsenek, az pedig nagyon felemás megoldás lenne, ha a szerzők a betűket és a számokat gépelik, a speciális karaktereket pedig kézzel írják a szövegbe. Vagyis változatlanul az egyetlen elfogadható megoldás, hogy a programot jó festékszalaggal nyomtatjuk ki.

Frank Parkanyi, Hillcrest, NY 11432 USA

Örömmel és meglepetve kaptam kézhez a Mikromagazin 1988/6. számát, amit kéresemre mintapéldányként a Kultúra küldött meg. Tudtam, hogy Magyarországon a számítástechnika és a tudomány igen fejlett, de hogy ilyen színvonalú magazin is létezik, az meglepett.

A fiam munkahelyén számítógéppel dolgozik — nagyon szereti —, és már több „nyelven beszél”. Azt mondta: „Dady, tanulj meg számítógépen dolgozni, nagyon fogod szeretni, és nem fogsz utatkozni a nyugdíjas idődben.” Igen ám, de én hiába élek Amerikában, csak magyarul értek mindent jól, tehát csak magyar leckéket vehetek a tanuláshoz. Így történt, hogy kértem a Kultúra Külkereskedelmi Vállalat levelezőjét, küldjön magzint és könyvet, amiből a bázist megismerem. Kérem, segítsenek, milyen könyvből tanulhatom meg az alapismereteket. Persze csak nagyon alacsony fokú anyag jó, mert a 67 éves fejem már lassabban fog. Ismételtlen gratulálok a magazinhoz, és várom szíves válaszukat.

Nem is tudom, hogyan köszönjem meg az elismerő szavakat szerkesztőségünk valamennyi tagja nevében, és persze a Kultúra munkatársának, Fekete Ilonának, hogy a Mikromagazint elküldte Párkányi úrnak. Kérésnek megpróbálnak eleget tenni, és a kért könyvet, illetve anyagokat el fogjuk küldeni.

Szabó Krisztián, Siklós

Kérem, hogy közölje le ezt a levelet. Négy éve kaptam egy ZX—Spectrumot. Ezen BASIC-ben és gépi kódban készítettem kisebb-nagyobb programokat. Az újságból hiányolom a Spectrummal foglalkozó cikkeket (nem a programokat), és ezzel a kis hibával is a legjobb hazai számítástechnikai lapnak tartom. Néhány Spectrum klub címét és a foglalkozások idejét szeretném megtudni Pécsen, esetleg Budapesten.

Mi részben a Magazinhoz küldött cikkekből tudunk válogatni (ezt már sokszor megírtam), részben saját szerzőinket kérjük fel egy-egy aktuális téma feldolgozására.

A Spectrumról szóló cikkek az előbbiektől és nem az utóbbiak közé tartoznak. Ha kapunk ilyen színvonalas írásokat, akkor azokat szívesen közöljük.

A budapesti Spectrum klub a HCC keretében működik, a vezetője Hivessy Ferenc, a foglalkozásokat minden hétfőn 18 órától a Szellőző Műveknél (XI., Építész u. 8/12.) tartják és minden hónap második szerdáján 18 órától a Ganz Villamossági Művek MHSZ Rádióklubjában (II., Fény u. 11.). Ami a pécsi klubot illeti, azt tanácsolom, hogy forduljon az NJSZT Baranya megyei szervezetéhez (Gyöngyösi László elnökhöz vagy Gyimesi László titkárhoz: SZÜV, Pécs, Szántó Kovács János u. 3.).

Siklósi Tamás, Székesfehérvár

Lapjuk rendszeres olvasója vagyok. Mindig elolvastam ezt a rovatot. Most is itt találtam egy levélrészletet, amely felkeltette az érdeklődésemet. Ezzel kapcsolatban szeretnék néhány információhoz jutni. A Mészáros Gyula levelében említett nyomtató érdekelne, mivel már régóta szeretnék egyet. Ezzel kapcsolatban az a kérés, ha egy mód van rá, szeretném megkapni Mészáros Gyula vagy az Angström Gmk címét. Örömmel vettem továbbá a lapjamban indult új cikksorozat, amely hardverrel foglalkozik. A legnagyobb öröm számomra, hogy úgy látszik, a cikksorozat az alapoknál kezd. Remélem, a folytatás is hasonlóan érdekes lesz.

Az Angström Gmk címe: Budapest, Zsiroghegy u. 110. 1020. Telefon: 699-499. Reméljük, hogy nemcsak a hardversorozat, de a többi írásunkat is örömmel olvassa.

Nagy József, Kunszentmárton

Közel két éve vagyok boldog (?) TV-Computer tulajdonos. A gépet, így ahogy van, hibával, erőnyeivel együtt szeretem. Megbízható, jó segítőtárs tanulásban, munkában egyaránt. A tulajdonképpeni gondom: lakóhelyemen és környékén erre a géptípusra szinte semmi sem kapható! Hosszas utánjárással sikerült megszereznem az O. S. kézikönyvet, aminek tanulmányozása nagyon segítette fejlődésemet. Kérem az következő: amennyiben mód van rá, küldje el levelemet egy TVC klubnak (ha létezik ilyen). Szívesen lennék tagja egy ilyen közösségnek. Minden érdekel, ami a géppel kapcsolatos. Kicsit szűkítve: Pascal, LOGO, FORTH-grafika, felhasználói programok, bővítési lehetőségek.

A HCC tervezti TVC klub megalakítását. A TVC-hez eddig két könyv jelent meg, mindkettő a Novotrade kiadásában, és meg is vásárolható a Novotrade Áruházban. Garai Géza: TVC operációs rendszer. Ára 190.— Ft. Bazics Bernadett—Balogh László: TVC Basic. Ára 99.— Ft. Köszönöm a leveleket.

KOVÁCS GYÖZŐ

A

Helyközi Távbeszélő Igazgatóság

Számítógéppontja felvételre keres:

- TPA 1148-as számítógépre gyakorlott rendszerprogramozót, orosz és angol nyelvtudással előnyben,
- érettségizett, gépelni tudó, ügyintézésben jártas adminisztrátort,
- középiskolai végzettséggel rendelkező, TPA 1148 nagygépes, illetve IBM kompatibilis PC-s ismeretekkel bíró futtatási felelőst.

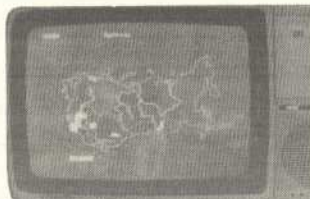
Egyéves munkaviszony után 33%-os vasúti utazási igazolvány, sokféle üdülési kedvezmény, postalai kedvezmények. Bérezés kollektív szerződés alapján.

Jelentkezni lehet:

Helyközi Távbeszélő Igazgatóság Számítógéppontja

Budapest VIII., Horváth Mihály tér 17-19.
Telefon: 340-797

Egy sarokkal olcsóbb!



„Városnézés” a Szovjetunió programmal

vétel mellett, amire 6 hónap garanciát vállalunk.

Az e havi kedvezmény

Matematikához a Számelmélet, Egyenlet, Kombinatorika, Függvényábrázolás és -transzformáció, Gyökkeresés, Galton Prim programokat, fizikához a Soros RC körben lejátszódó folyamatok modellezése, Mágneses mező, kémiahoz a Mend, biológiához a Növényhatározás, idegen nyelv-tanuláshoz az Úrcsata, Magyar—orosz szótár, Interaktív magyar—angol és magyar—német szótárprogramokat, földrajz tanuláshoz a Magyarország-, Szovjetunió- és USA-, s mindezeket kivül a Ki mit tud? és az Unifel programot ajánljuk.

Kínálatunkból, ha választotok, programként 10 százalékos, tiszlél több program vásárlásakor 20 százalékos árengedményt adunk.

Egy program ára 313 forint.

A Kombinatorika segít megoldani az ismétlés nélküli vagy ismétléses permutációkat; kombinációkat, variációkat állít elő, lottószámokat generál, s többhasábos totószelvény kitöltésére is alkalmas, fix mérközéséknél.

A Függvényábrázolás és transzformáció a gimnáziumi alaptanterv tíz, leglényegesebb függvényének ábrázolására, transzformáltjainak bemutatására alkalmas (lineáris, másodfokú tört, gyök, exponenciális, logaritmus és trigonometrikus). A program egyszerre több görbét is ábrázolhat a koordináta-rendszerben.

A Soros RC körben lejátszódó folyamatok modellezése program igazából a harmadikos gimnazistáknak szól. Célja, hogy a sorosan kapcsolt kondenzátort és ellenállást tartalmazó áramkörökben lejátszódó folyamatokat bemutassa, és szemléltesse, hogyan befolyásolják az áramköri paraméterek az áramerősséget, illetve a kondenzátor feszültségének időbeli változását.

A Mend az első gimnazistáknak íródott. A Mengyelejev-féle periódusos rendszerben uralkodó törvényszerűségek felismerését, alkalmazását mutatja be az adatok becsülése alapján.

Az Úrcsata program játékos formában segíti az orosz nyelvű főnevek nemének gyakorlását. Círil betűkkel írja ki a szöveget. Gimnáziumban és szakközépiskolában tanuló elsősorúknak ajánljuk a nyelvtanulást könnyítésére. Egyébként a program folyamatosan regisztrálja a helyes és helytelen válaszokat, értékeli a nekirugaszkodás teljesítményét, és az eredményt a végén százalékosan összesíti.

A Szovjetunió program a hatalmas ország 15 szövetségi köztársaságának fővárosait, jelentősebb városait ismerteti, gyakorlatja, s befejezésül elárulja, mi volt jó és mi nem. Végül ajánljuk az Unifel programot, mellyel bármilyen tantárgyból maximum 40-50 kérdést és választ lehet faggatódikténi géphez vinni.

Lányok, fiúk! Még kedvenc tantárgyaitok is olykor-olykor kifognak rajtatok. De ne csüggedjétek, ne búsuljatok. A Tudományszervezési és Informatikai Intézet Videoton TV Computeren futtatható, kedvezményesen megvásárolható oktató-programjaival megkönnyíti az otthoni fejlettségét. A programok jobban „súgnak”, mint a legügyesebb padtárs. Tábla híján a monitoron mutatják érzékletesen a tananyagot, és addig nyúzóhatjátok magatok példákkal, amíg ki nem okosodtatok. Ráadásul a bonyolult számokkal, szerkesztésekkel sem kell papíron bajlódnotok. Ennyi segítséggel a szigorú szülői kikerdezés már igazán csak gyerekjáték!

Aki tehát (szülő is lehet) a lapunkból kivágott sarokszelvényt a TII-ben (Budapest XI., Egy József u. 1—9. BME „E” épület XI. em. 111. Postacím: 1372 Bp. Pf. 454) átadja, vagy megrendelésével együtt elküldi oda, havonta húsz oktató-program közül válogathat, olcsóbban juthat hozzájuk. A kedvezmény a szelvényen feltüntetett hónapban érvényes.

Postai megrendeléskor (a megrendelőlapra legyen rajta a név, pontos cím, személyi szám, dátum) programjainkat után-



SZÁMÍTÁSTECHNIKAI KUTATÓ INTÉZET ÉS INNOVÁCIÓS KÖZPONT 1015 Budapest, Donáti u. 35—45.



... HA KINÖTTE
az eddigi PC-k nyújtotta
lehetőségeket

AKKOR



PROPER—132 számítógép

IBM PC/AT-kompatibilitás
Feldolgozó teljesítmény: PC-nél 1B-szor, PC/AT-nál
5-ször nagyobb

Szolgáltatások kiterjesztése

- EGA-szintű, nagy leibontású grafika
- gyors RAM-BIOS

Rendszerbővítő opciók

- 2 megabájt EXRAM
- matematikai társprozessor stb.

Típusú alkalmazások

- hálózati SERVER állomás
- nagy teljesítményű adatfeldolgozás
- grafikus és CAD/CAE alkalmazás
- szakértői rendszerek

A 32 bites architektúrát kihasználó szoftvertermékek: TOOLBOX—386, PRO—X

VÁLLALKOZUNK:

- Grafikus alkalmazási rendszer kialakítására (1024 × 1024 felbontással, VDI és GKS szoftverrel az AutoCAD V. 9 bázisán)
- Grafikus és szöveges adatbázisokat együttesen kezelő rendszerek kialakítására (videokamerás képbővítést, dBASE III Plus adatbázis-kezelést)
- PRONET-bázisú komplex hálózati alkalmazások létrehozására, telepítésére

PRONET 3.10 hálózat

IBM PC—PCXT—PCAT-kompatibilis számítógépekkel kialakítható hálózati rendszer:

- adatátviteli sebesség: 1 Mbit/s
- adatátviteli közeg: egyszerű csavart érpár

Nemzetközi LAN-szabványokkal, szoftvermegoldásokkal kompatibilis

- MS—DOS 3.x
- IBM PC NETWORK, IBM PC LAN
- NETBIOS

Egyenrangú vagy dedikált szolgáltatások

- dedikált FILE SERVER
- (2—8 megabájt EXRAM, gyorsított állomány-rendszer, nem dedikált használat is lehetséges)
- dBASE—NETPLUS
- (dBASE III LAN-PACK-kal kompatibilis)
- GATEWAY család
- (X.25 TPA/DEC, BSC 3270)
- FAX-csatlakozás

VEGYE FEL...

a kapcsolatot szakembereinkkel!

KÉSZTERMÉKEINKKEL,
VÁLLALKOZÁSAINKKAL
az Ön rendelkezésére állunk!

További információk:



Hardware Rendszertervezési Laboratórium
Telefon: 150-269

PROGRAMTERMÉK

GENERÁCIÓK

Már megint egy újabb cikk a fiatalok és az idősebbek viszonyáról? Vagy a számítógép-generációk leragadt csontjain próbál még valaki rágódni? Nem. Igérem, hogy aktuálisnak tűnő témát feszegetek: az oktatóprogramok generációit próbálom rendszerezni, hátha ezzel sikerül körvonalazni a közeljövő ilyen jellegű teendőit. Aztán helyet szorítok még egy „második generációs” földrajzi témájú program elemzésének is.

Kezembe kerültek régi oktatóprogram-katalógusok. Kicsit bajba kerültem, amikor időrendben sorba akartam rakni őket. Szinte alig találtam bennük születési időpontra utaló adatokat. Ahhoz azért mindenesetre elegendőnek bizonyultak, hogy megfigyelem az oktatási szoftver generációváltásait.

A „nulladik generációs” programok azok, amelyek a piaci szervezetségnek ellenére bukkantak fel. Az első százas széria az ABC 80-ra született. A programokat a szerzők saját indíttatásból ajánlották fel annak, akitől a gépet kapták, vagyis a Tudomány-szervezési és Informatikai Intézetnek, hogy próbálja meg forgalmazni azokat. Erre a korszakra mégis inkább a cserekereskedelem volt a jellemző.

Az „első generáció” a TII iskolaszámítógépprogrampályázata kapcsán keletkezett. Ezek a programok HT—1080Z iskolaszámítógépre készültek. A programokat a TII, majd a Novotrade és a TII közösen forgalmazta. A Novotrade hovatovább úgy döntött, hogy önállóan is beszáll ebbe az üzletbe. Két írója is elkezdett oktatási programok írásának menedzselésével foglalkozni: a Deltasoft iroda és a Játékstúdióiból szerveződött Octasoft iroda. A piacon mások is hozzáláttak oktatóprogramok írásához, akik közül kiemelkedik az International House.

Közben az iskolákban hirtelen újabb számítógépek jelentek meg: a Videotonok és a Commodore-ok. Ezekre persze sürgősen programok kellettek. Kézenfekvő volt, hogy a HT-re készült programokat gyorsan át kell írni az új gépekre. Az így születő programok többnyire az első generáció egyszerű másolatai voltak. Mindenesetre az adaptálásoknak nevezhetjük „második generáció”-nak.

A „harmadik generáció” alapjellemzője, hogy nem a „népi kohó” mozgalom keretében keletkezik, hanem tervszerű, tudatos tevékenységként jön létre. A programokat nem amatőrök, hanem profik tervezik, bár a megírás ráhagyatják a témában szakértő tanárookra is. Ezzel azonban még mindig nem mondhatjuk el, hogy „itt van már a Kánaán”, mert a harmadik generáció csak a meglévő tananyagokat segíti, és célja is csak ez lehet. Ez nem ugyanaz, mintha egy tananyagot eleve úgy tervezünk meg, hogy a számítógépet az alkalmazott oktatási eszközök egyikeként eleve feltételezzük. Ha majd ideig jutunk, akkor feltétlenül egy újabb, egy „negyedik generáció”-ról beszélhetünk. (Valószínűleg még sokáig kell várunk erre.)

A TII új programjai is várhatóan csak a harmadik generáció szintjét ütik majd meg, hiszen a tananyagok készítéséért, fejlesztéséért hivatalból nem az intézet a felelős.

Amivel a TII előbbre viheti a dolgot, az az esetleg a szerzőknek nyújtandó, egységes szabványokon alapuló szoftvertechnológia.

Az eddig született, ezernél valamivel kevesebb oktatóprogram egy-két nagyságrenddel elmarad a szükséges mennyiségtől. Sajnos, az iskolaszámítógép-program beruházásait nálunk ma is a gépár uralja. Ha ez hosszabb távon így marad, akkor a számítógépeket nem integrálhatjuk az oktatásba, megmaradnak a hobbitársak privilégiumainak.

ÖSSZEFOGLALÓ ADATOK

Forgalmazó:	Tudomány-szervezési és Informatikai Intézet
Terméknév:	A SZU és az USA városai
Szerző:	Modász Attila
Géptípus:	HT, C Plus/4, TVC
Hordozó:	kazetta vagy lemez
Dokumentáció:	rövid, a célnak megfelelő
Ár:	600 Ft körüli (gépfűgő)

MINŐSÍTŐ ADATOK

Kezelhetőség:	kiváló
Teljesesség:	jó
Dokumentáltság:	közepes (jelenleg)
Használhatóság:	jó
Ár/teljesítmény:	jó
Összbenyomás:	jó

Attól félek, hogy az itt mondottak pénz-igénye sokkal nagyobb, mint amit az oktatásért felelősek szoftverberuházásra bármikor is terveztek.

Ilyen bevezető után nehéz átváltani valamilyen program elemzésére. Bár a piacon vannak már harmadik generációs programcsomagok is, egy-két átírából származó, második generációs oktatóprogram is megérdemli, hogy szót ejtsünk róla.

A TII katalógusaiban szerepelnek olyan programok, amelyek láthatóan több gépen is használhatók, de az újabb géptípusokra már úgy kerültek át, hogy a szerzők figyelembe vették az új gép adta lehetőségeket, és az időközben megnövekedett számítástechnikai tudást is kamatoztatták az átírások. A most elemzendő földrajzi program ilyen (lásd az összefoglaló adatokat).

Kezdjük az árral. Ezt mi nem azért közöljük, hogy pontos piaci információkat szolgáltatassunk, hanem az ár/teljesítmény viszony értékeléséhez keressünk alapot. Folytathatjuk a dokumentációval. A prog-

ramleírás elég vékonyka, de figyelembe vehető, hogy utal az ide tartozó tananyagokra, és az új TVC-s változathoz új, szöveg-szerkesztővel szerkesztett dokumentációt is ígér.

A program TVC-re írt, modernizált változatait láttam, két programra bontva a témát, és alaposan kihasználva a géptípus adta grafikai lehetőségeket. Az USA program az Egyesült Államok egybefüggő elhelyezkedésű, 45 tagállamának fővárosait gyakoroltatja vagy kérdezi ki. A SZU program a Szovjetunió 15 köztársaságának fővárosát és 16 nagyvárosát gyakoroltatja és kérdezi ki. A két program teljesen azonos elven működik. A városok helyét elég nehéz pontosan eltalálni, a programok tehát elég szigorúak. Az ellenőrző menüben válaszolni lehet, hogy hány kérdést adjon fel a program. Külön menüben kell kérni az értékelést. Van olyan kombinált menü, amikor egy-egy állam vagy köztársaság nevét és helyét is megkérdezi a program, két kérdésenként értékelve — külön-külön a pozíció és a városnév eltalálását — az eredményt.

A program rosszul beállított tévén nem jól érvényesül! A grafikus lehetőségeket alaposan kihasználó programoknál ez törvényszerű. Vigyázni kell tehát az állomás — fekete-fehér tévén a megfelelő kontraszt és fényerő —, a színes tévén pedig meg a kontrasztbeállításra is. Ez olykor-olykor elég aprólékos munkát igényel, de megéri. Rossz szerszámmal nem szabad dolgozni! Demoralizáló.

A programokat egy apró hibán csíptem rajta. A SZU program ugyanazt a köztársaságot egy menüben kétszer is feladta kikérdézők, ráadásul a véletlen ördögének köszönhetően pontosan egymás után. Így kaptam egy potyakerést. Köszönöm. Az elért százalékokkal így sem szívesen dicsekszem. A programot a táblázatban összefoglaltak szerint minősítettem.

Bár az első-második generációs pályázati programok — szerteágazó stílusuk folytán — nem oldják meg az „oktatásiszoftver-krizist”, figyelembe kell venni a pedagógusok számítógépes tudását is. Merthogy valamilyen módon mihamarabb kifizetődő lehet bevonní őket nagyobb mennyiségű, harmadik generációs programcsomagok írásába.

A pályázati szoftver korszaka legalább segített feltérképezni a felhasználható szakemberállományt. Ehhez kell még megtalálni a helyes célokat és nem utolsósorban „pézt, paripát, fegyvert”.

ZSADÁNYI PÁL

Minden számunkban két feladatot közlünk: az első logikai, matematikai tudást, a második számítástechnikai alapismereteket is igényel.

A feladatok után közöljük az elérhető maximális pontszámot. A rész megoldásokat is pontozzuk.

A pontgyűjtést, vagyis a pontvadászatot az esztendő végén zárjuk. A legjobb tíz versenyző nevét magazinunkban közzé is tesszük, ők lesznek azok, akik könyvtulajánt is kapnak.

A helyes megoldások a feladatok közreadása után két lapszámmal később jelennek meg, így a pontvadászoknak jut idejük a gondolkodásra.

Beküldési határidő: 1988. november 15.

Címünk: Mikroszámítógép Magazin Szerkesztősége, 1371 Budapest, Pf. 433.

Az 1988/4. szám 1. feladatára 11, a 2. feladatára 10 megjelölt érkezett be. Ezek közül az 1. feladat megjelölési százaléka — a maximális 5 pontot véve 100 százaléknak — 84,6 százalék, a 2. feladatnál szintén a maximális 5 pontot véve 100 százaléknak, 84 százalék.

Az 1988/4. szám 1. feladatára sok olyan megjelölt érkezett be, amely részletes taglalással a bizonyítandó oszthatóságot meg nem lévőnek találta, ezt ellenpéldákkal is bizonyította. Megfelelő indoklásokkal ezeket is elfogadtuk, minthogy a feladat pontatlan megfogalmazása hozzájárult ehhez. Itt ugyanis az $n(n+1)$ -gyel való oszthatóság helyett azt kellett volna — mint ahogy az egyik megjelölt is javasolta — írni, hogy azt állítjuk: az illető szummának $n(n+1)$ tényezője. Így ugyanis kihangsúlyoztuk volna, hogy nem az egész számú oszthatóságot kívánjuk vizsgálni, amibe a kifejezés nevezője is belejár, hanem az $n(n+1)$ tényező jelenlétét.

Jó vadászatot kíván a feladatok összeállítója:

Dr. Hoffman Tibor

1. feladat

Mint ismeretes, az idén bevezetett személyi jövedelemadó rendszerében, ha az adóalap meghatározott, az adó százalékát sávosan határozzák meg, mégpedig a következőképpen.

0 és 48 000 Ft közötti	0%
jövedelemrészt után	
48 001 és 70 000 Ft közötti	0%
jövedelemrészt után	
70 001 és 90 000 Ft közötti	20%
jövedelemrészt után	
90 001 és 120 000 Ft közötti	25%
jövedelemrészt után	
120 001 és 150 000 Ft közötti	30%
jövedelemrészt után	
150 001 és 180 000 Ft közötti	35%
jövedelemrészt után	
180 001 és 240 000 Ft közötti	39%
jövedelemrészt után	
240 001 és 360 000 Ft közötti	44%
jövedelemrészt után	
360 001 és 600 000 Ft közötti	48%
jövedelemrészt után	
600 001 és 800 000 Ft közötti	52%
jövedelemrészt után	
800 001 Ft feletti jövedelemrészt után	56%
	60%

Adjuk meg zárt formulában az a adóalap függvényében a fizetendő adó p százalékát. A zárt formulában az abszolútérték-jelölés is használható. (5 pont)

2. feladat

Egy digitális jeleket továbbító vagy tároló rendszerben előfordulhatnak hibák. A legegyszerűbb hiba az, ha egy bitben lép fel hiba, tehát 0 helyett 1 áll vagy fordítva. Az ilyen egyetlen bitben fellépő hibát ki lehet védeni hibajelzéssel vagy hibakorrekcióval. Az egyetlen hiba észlelését azzal tehetjük a legegyszerűbbé, hogy egy n bites rendszerhez még egy $n+1$ -edik ellenőrző bitet adunk hozzá, és előírjuk például azt az ellenőrzést, hogy az 1-es értékű bitek száma mindig legyen páros. Egy páratlan számú 1-es bittel leírható bináris számhoz ekkor az ellenőrző bit 1 lesz, míg a páros számú 1-es bittel leírható bináris számnál ez az el-

lenőrző bit 0 lesz. Ha most ebben a rendszerben a továbbításnál vagy tárolásnál egyetlen bit hibádzik, a páros bitszámosság páratlanra változik és ez hibajelzést ad.

Hogyan lehet egyszerűen egy hibajavító kódot létrehozni, amely nemcsak jelzi, hanem ki is javítja a hibát? (8 pont)

Az 1988/8. szám feladatainak megoldása 1. feladat

Ha a háromszoros gyököt y -nak jelöljük, akkor az egyenletnek

$$ax^3 - 3ayx^2 + 3ay^2x - ay^3 = 0$$

alakúnak kell lennie, vagyis

$$-3ay = -(a^2 + 3a - 1),$$

$$3ay^2 = b + 3a^2 - a,$$

$$-ay^3 = -ab.$$

Ebből, mivel $a \neq 0$,

$$b = y^3 \text{ és } y = \frac{a^2 + 3a - 1}{3a}$$

adódik. Ezeket a második egyenletbe visszahelyettesítve

$$\frac{(a^2 + 3a - 1)^2}{3a} = \frac{(a^2 + 3a - 1)^2}{27a^3} + 3a^2 - a,$$

vagy polinom alakra hozva

$$8a^4 - 30a^3 + 66a^2 - 63a + 33a^2 - 9a + 1 = 0$$

A váltakozó előjelek miatt ennek a valós gyökei mind pozitívak. Racionális gyökei így 1 , $\frac{1}{2}$, $\frac{1}{4}$ vagy $\frac{1}{8}$ lehetnek. Behelyettesítéssel látható, hogy

$$a = 1 \text{ és } a = \frac{1}{2}$$

kielégíti ezt a hatodfokú egyenletet, és ezek mindegyike háromszoros gyök. Így nincs más gyök. Ezekhez

$$b = 1, \text{ illetve } b = \frac{1}{8}$$

értékek tartoznak. Az egyenlet gyökei ekkor

$$x = 1, \text{ illetve } x = \frac{1}{2} \quad (3 \text{ pont})$$

2. feladat

Az első lépésként mindenképpen a tetszőleges módon kiválasztott ötjegyű próbaszámra kérdezzük meg a gép kiértékelő választát. Ha ez a válasz 35, akkor szerencsénk volt, rögtön eltaláltuk a tárolt számot. Ha a válasz 44-nél nagyobb, akkor a számok már megvannak, csak az elhelyezésükre kell *permutációs* programot készíteni.

Ha a válasz nem nagyobb 44-nél, akkor a következőképpen járhatunk el. A válasz értékének a tárolása mellett most a kiválasztott ötjegyű szám *egyes jegyeit* változtatjuk meg *rendre egymás után* és kérdezzük meg a gép választát. Ha egy ilyen változtatással a gép válasza *kisebb* a tárolt első válasznál, akkor a megváltoztatott számjegy változtatás előtti értéke már azonosítva van, mint a keresett szám számjegye, és azt is tudjuk, hogy a változtatásra felhasznált számjegy nincs a tárolt szám számjegyei között.

Ha a gép válaszában *első számjegye megegyezik* a tárolt első válasz *első számjegyével*, akkor tudjuk, hogy az eredeti és a cserélt számjegy is szerepel a tárolt szám számjegyei között. Viszont ha a gép válasza *nagyobb*, mint a tárolt első válasz, akkor az újonnan beadott csereszámjegy a tárolt szám számjegyei között van, míg a megváltoztatott számjegy nincs. Így módon lépésenként eljuthatunk egy olyan kétjegyű számszámhoz, melynek az első jegye 5, tehát így ismerjük az 5 kívánt számjegyet. Most már csak *permutációs lépések* szükségesek e számjegyek helyes elhelyezéséhez.

Ezt a következőképpen végezhetjük el. Az $50+k$ alakú válaszban k azt jelenti, hogy hány számjegy van a helyén. $k \neq 5$ esetén ($k = 5$ a megoldás jelenti), e válasz tárolása mellett sorban felcserélünk 2-2 számjegyet a számból és megkérdezzük a gép választát. Ha az új válasz *kisebb* a tárolt előző válasznál, akkor a felcserélt két számjegy közül a felcserélés előtt legalább egyik már a helyén volt. Ha az új válasz 2-vel *kisebb* az előzőnél, akkor mindkét számjegy a helyén volt. Ha 1 -gyel *kisebb*, akkor további számcserével állapítjuk meg, hogy melyik volt a helyén. Ha az új válasz *megegyezik* a tárolt előző válasszal, akkor a felcserélt két számjegy közül egyik sem volt a helyén és a cseré után sincsenek a he-

lyükön. Ha az új válasz nagyobb a tárolt előző válasznál: ha 2-vel nagyobb, akkor a felcserélés után mindkét számjegy a helyére került. Ha csak 1-xyel nagyobb, akkor az egyik a helyére került, a másik pedig biztosan nincs a helyén. Azt, hogy melyik van a helyén, egy további szám-cserével állapíthatjuk meg.

Ezzel a stratégiával viszonylag kevés lépéssel eljuthatunk a megoldáshoz. (8 pont)

Az 1988/6. szám 2. feladatának az 1988/8. számban megadott megoldásában lévő szubrutinok közül még célszerű két szubrutint szerkeszteni. Az egyik egy olyan, amely két, már kitalált számjegy egymás közötti cseréjénél határozza meg a gép választát, rögzítve a cseré előtti helyzetet. A másik szubrutin olyan, amely egy új szám bevezetésekor határozza meg a gép választát, rögzítve az előző helyzetet is.

Nevezük az első szubrutint ICSRE-nek és ez legyen egy olyan függvény, amely az I-edik és J-edik számjegy felcserélésével, rögzítve a régi helyzetet, az új választ adja. A másik legyen IUSZM, mely valamely régi helyett egy új számjegy bevezetésekor, rögzítve a régi helyzetet, az új választ adja meg. Legyen az első:

```
FUNCTION ICSRE (I,J,IC)
C AZ I-EDIK ES J-EDIK ELEM CSEREJE UTAN IC-BEN VAN
C A CSERE ELOTTI JELZOSZAM
C ES ICSRE-BEN A CSERE UTANI
IC = Kerd(I,A,IB)
IH = IB(I)
IB(I) = IB(J)
ICSRE = Kerd(I,A,IB)
RETURN
END
```

A másik szubrutin, az IUSZM, mely az I-edik helyen lévő számjegyet helyettesíti egy még fel nem használt számjeggyel. Hatásosabb a keresés, ha még az IA és IB tömbökön kívül felhasználunk egy szintén 5 elemű IE(5) tömböt, mely azokat a számjegyeket tartalmazza az 1-től 5-ig megnevezett elemeiben, melyek biztosan nem szerepelnek a keresett tárolt számban, egy szintén 5 elemű IV(5) tömböt, mely a már biztosan megtalált számjegyeket tartalmazza 1-től 5-ig, és egy szintén 5 elemű IJV(5) tömböt, mely azoknak a számjegyeknek az IA tömbben való helyét adja, melyek a próbaszámban már biztosan a helyükön vannak. Ezeket a tömböket a főszubrutinban definiáljuk.

```
Az IUSZM szubrutin tehát:
C FUNCTION IUSZM(I,J,II,IH)
C A MEG FEL NEM HASZNALT I SZAMMAL HELYETTESITI A
C PROBASZAM JJ-EDIK ELEMET ES IUSZM A FELCSERELES
C UTANI JELZOSZAM. A FELCSERELES ELOTTI JELZOSZAM
C IC-BEN VAN
C IH TARTALMAZZA A JJ HELYEN LEVO REGI SZAMOT
IC = Kerd(I,A,IB)
IH = IB(JJ)
DO 6 J = 1,9
DO 7 K = 1,5
IF(JJ.EQ.K)GOTO 7
IF(J.EQ.IB(K))GOTO 6
7 CONTINUE
DO 8 II = 1,L
IF(J.EQ.IE(II))GOTO 6
8 CONTINUE
IB(JJ) = J
IUSZM = Kerd(I,A,IB)
I = J
RETURN
6 CONTINUE
END
```

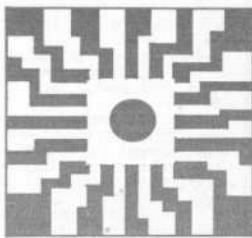
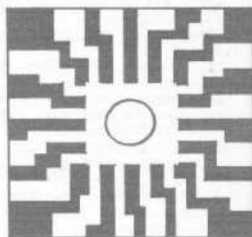
A főszubrutint e szubrutinok felhasználásával az alábbi módon írhatjuk fel. Itt a keresett számot M-mel, a próbaszámot N-nel jelöljük.

```
FUNCTION KRSTT(M,N)
DIMENSION IA(5),IB(5),IE(5),IV(5),IVV(5)
C AZ M SZAM SZAMJEGYEIT AZ IA TOMB BEN, AZ
C N-EIT AZ IB TOMB BEN HELYEZZUK EL.
C AZ IE TOMB ELSO L ELEME AZ M-BEN BIZTOSAN
C NEM SZEREPLO SZAMJEGYEKET ADJA.
C AZ IV TOMB ELSO LVV ELEME AZ M-BEN BIZTOSAN
C SZEREPLO SZAMJEGYEKET ADJA.
C AZ IVV TOMB ELSO LVV ELEME ADJA AZOKNAK A
C SZAMJEGYEKNEK AZ IA TOMB BEN VALO HELYET,
C MELYEK A PROBASZAM BAN BIZTOSAN A HELYUKON
C VANNAK.
CALL BONT(M,IA)
```

```
CALL BONT(N,IB)
L = 0
LV = 0
LVV = 0
27 IC = Kerd(IA,IB)
IF(IC.LT.50)GOTO 11
IF(IC.NE.55)GOTO 12
17 KRSTT = IPSZM(IB)
RETURN
12 IF(IC.LE.52)GOTO 13
C INNEN 14-IG IC = 53
DO 14 I = 1,5
DO 15 J = 1,LVV
IF(IVV(J).EQ.I)GOTO 14
15 CONTINUE
DO 14 J = 1 + 1,5
DO 16 II = 1,LVV
IF(IVV(II).EQ.J)GOTO 14
16 CONTINUE
ID = ICSRE(I,J,IC)
C ID CSAK 51,52, VAGY 55 LEHET
IF(ID.EQ.55)GOTO 17
IF(ID.EQ.51)GOTO 18
C INNEN 19-IG ID = 52
DO 19 JJ = J + 1,5
DO 20 II = 1,LVV
IF(IVV(II).EQ.JJ)GOTO 19
20 CONTINUE
IC = ICSRE(I,J,ID)
IG = ICSRE(I,JJ,IC)
C IG CSAK 51,52, VAGY 55 LEHET
IF(IG.EQ.55)GOTO 17
IF(IG.EQ.51)GOTO 21
IC = ICSRE(I,JJ,IG)
IM = ICSRE(J,JJ,IC)
IF = (IM.EQ.55)GOTO 17
IC = ICSRE(J,JJ,IM)
IVV(LVV + 1) = J
IVV(LVV + 2) = JJ
LVV = LVV + 2
GOTO 14
21 IVV(LVV + 1) = 1
IVV(LVV + 2) = JJ
LVV = LVV + 2
GOTO 14
19 CONTINUE
GOTO 14
18 IVV(LVV + 1) = 1
IVV(LVV + 2) = J
LVV = LVV + 2
14 CONTINUE
13 IF(IC.EQ.50)GOTO 22
C INNEN IC = 51, VAGY 52
DO 23 I = 1,5
DO 24 J = 1,LVV
IF(IVV(J).EQ.I)GOTO 23
24 CONTINUE
DO 25 J = 1 + 1,5
DO 26 II = 1,LVV
IF(IVV(II).EQ.J)GOTO 25
26 CONTINUE
ID = ICSRE(I,J,IC)
IF = (ID.EQ.53)GOTO 27
IC = ICSRE(I,J,ID)
25 CONTINUE
23 CONTINUE
DO 28 I = 1,5
DO 29 J = 1,LVV
IF(IVV(J).EQ.I)GOTO 28
29 CONTINUE
DO 30 J = 1 + 1,5
DO 31 II = 1,LVV
IF(IVV(II).EQ.J)GOTO 30
31 CONTINUE
ID = ICSRE(I,J,IC)
IF(ID.GE.52)GOTO 27
```

```
IC = ICSRE(I,J,ID)
30 CONTINUE
28 CONTINUE
11 DO 32 I = 0,9
DO 32 JJ = 1,5
ID = IUSZM(I,JJ,IM)
IDD = INT(ID/10)
ICC = INT(ICC/10)
IF(IDD.LE.ICC)GOTO 33
IV(LV + 1) = I
LV = LV + 1
IE(L + 1) = IH
L = L + 1
GOTO 27
33 IF(IDD.EQ.ICC)GOTO 32
IE(L + 1) = I
L = L + 1
IV = (LV + 1) = IH
LV = LV + 1
IC = IUSZM(IH,JJ,I)
GOTO 27
32 CONTINUE
RETURN
END
```

(10 pont)



AKNA-
BIZTONSÁG

Elektronikus vezérlésű bányabiztosító berendezést szereltek fel a Veszprémi Szénbányák akjai üzemében. Ennek a bányabiztosítónak nagy előnye, hogy nem kell lebontani és áttelepíteni, mert hidraulikus szerkezetekkel tetszés szerint mozgatható a vágatban. Legújabbán pedig számítógép vette át a gépkezelőktől a vágatokat védő önjáró acélszerkezetek mozgásának irányítását, ellenőrzését. Hangos jelzéssel figyelmeztet, ha rendelkezésséget észlel a berendezés működésében vagy az acélszerkezet anyagában. Az Arminaknában üzembe helyezett elektronikus vezérlésű pajzsbiztosítókhoz csak feleannyi kezelőre van szükség, mint korábban, ugyanakkor növekedett a biztonság és a termelékenység. A korszerűsítés egyben a leendő Ajka II. bánya műszaki főpróbjája is, mert azt majdan ilyen eszközökkel kívánják felszerelni.

10 Gigabájt

Az IBM kaliforniai kutatóközpontjában kifejlesztettek egy kísérleti magnelemezt, amelyen a tervek szerint 10 milliárd bitet lehet tárolni. A rögzítési sűrűség ötvenszer nagyobb, mint eddig. A barázdákat hasonló fotolitografikus módszerrel állítják elő, mint amilyet a félvezető morzsák előállításánál alkalmaznak. A lemez felületére először kobalttövezetet, majd filmet visznek fel, ezután elektronsugárral barázdákat húznak rá. Előhívással és maratással távolítják el a felesleges anyagot, majd végül a lemezen csak a kobalttöveg maradnak.

KÖNYVEK

1001/2 játék
C64/128 GEOS, NEWSROOM,
GAME MAKER, PRINT SHOP,
MUSIC MAKER
(Budapest, 1988. LSI ATSZ, 133 oldal.
Ára: 99,— Ft.)

Az 1001 játék és a Graphic BASIC 64-en című könyvek folytatása ez a kötet. Utalva a korábban szerzett tapasztalatokra, az olvasó kevesebb, de részletesebb leírást tartalmazó játépprogramokra és játékok POKE-okat próbálhat ki. A szövegben előforduló szak kifejezéseket a kötet végén található mintakönyv magyarázza meg. A gyűjteményben 16 játék leírása olvasható.

1001/1 játék
C64/128 Graphics BASIC
(Budapest, 1988. LSI ATSZ, 115 oldal.
Ára: 100,— Ft.)

A kötetben található számítógépes játékok fő elemei: a grafikus ábrázolás, az animáció, a reagálási sebesség, a hangeffekus, a zene, a beszéd szintetizálására bizonyos esetekben a szöveg. Ezeket az elemeket a különböző játépprogramok különböző fokozaton valósítják meg. Az ügyesség játékok fontos szerepet játszanak a kezűesség és a helyzetfelismerő képesség fejlesztésében, a stratégiai játékok a logikus gondolkodásra nevelnek.

ADOK—VESZEK—CSERÉLEK

tíhben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hírdetéseket közlünk. A díjazás: közlőteknek gépet soroként (60 karakter) 100,— Ft., magán személyeknek az első sor 50,— Ft., minden további sor 20,— Ft. Az NDSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

ATARI 130 XE számítógép (128 k), Disk3 drávj, Interfész 850, Assembler cartridge és egyéb tartozékok eladó. Tel.: 653-582

Floppy disk kétoldalas, dupla sűrűségű, 40 tpi, MD2D eladó. Tel.: 425-242

Commodore 64-hoz GYORS-HÁTTERTÁR cartridge. Kapacitása 2-31 kb-ot. A GYORS-HÁTTERTÁR-ba maximum 7 db, Gyakran használt program vihető be. A gép bekapcsolása után művelet jelentkezik belőle, és gombnyomásra a kiválasztott program azonnal fut. Javított programcsomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft.

Hozott programok elhelyezése a GYORS-HÁTTERTÁR-ban. Irompler László, Budapest, Atilla u. 22. 1201. Tel.: 287-493 este.

C64-re "Super" játépprogramok adok-veszek és cserélek - kazettán, Filetőth Atilla, Budapest, Sáró u. 13. V/120. 1156

C64-re programok adok és cserélek. Választ listával kérek! Krajinics Atilla, Szécs, Rákóczi u. 71. 2699

C64 floppy, printer, monitor eladó külön is + tartozékok, Szabó József, Karcag, Kossuth tér 7-9. 5300

C Plus/4 számítógép magnóval, programokkal eladó. Tel.: 143-031, 330-345 15-17 óra között

C Plus/4 magnóval, 150 db programmal és 10 db szakkönyvvel eladó. Árajánlatok

kat a következő címre kérem: Lantos András, Székesfehérvár, Palotai u. 71. IV/96. 8000

C128, C64, C Plus/4, C16-os programokat eladok, veszek, cserélek nagy választékból - elsősorban lemezen, de lehet kazettán is. Listát kérek és küldök. Keresztfalvi János, Budapest, Dobórdó u. 4. 1034

Schneider 64 k Color Personal Computer CPC 464 személyi számítógépmagnóval, monitorral, joystickkal, játépprogramokkal sűrűsben eladó. Bodó Péter, Herta, Arany J. u. 115. 6326

Szekosna GP 50-es, 127 mm-es szalaggal üzemelő printer eladó. Sinclair géphez közvetlenül csatlakoztatható. Bézi Ferenc, Szeged, Gáspár 2. u. 9/A. 6723

Spectrumhoz eladó: Beta 128/5-03 VER. floppy illesztő, MOM MF: 900/1800 floppy (a lemez megfordítható), Commodore 64-ros nyomtatóhoz varázspompos illesztő, Multiface 1 interfész. Steiner Erzsébet, Veszprém, Cserháti 4/C. IV/19. 8200

Spectrum 48-ra program eladó: 15 Ft/db. Dövecz László, Lenti, Átkötő u. 5/d. 8960

Spectrum (48/128 k-s) programkasszáikkal a legújabb angol slágerlistákról eladók. Közvetlenül választóberényben kaphók. Horváth Péter, Siklósi, Pf.: 129. 7800

XZ-Spectrum (48 k + II. interfész, joystick, 2 magyar nyelvű könyvvel), programokkal eladó. Varga Szabolcs Viktor, Budapest, Tóalmás u. 97. 1172 (17 órától)

C64-es programok cserélek kazettán, lemezen. Tökéletes, színvonalas csere-alap! Az egyes programkategóriák klasszikus és újdonságai egyaránt érdekelnek. Porosz Péter, Budapest, Rácz Aladár u. 30-32. 1121

C64-es felhasználói- és játépprogramok cserélek lemezen és kazettán. Több

mint kétféle program van(!). Szabó Csaba, Ócsa, Óllói u. 2B. 2364

C64-es programokat, programleírásokat, hardverleírásokat cserélek. Minden program érdekel, de csak lemezen. Cseréalap kb. 800 program. Keresem a Deskkap 1 programot. A választást listával kérem: Filip Róbert, Budapest, Szakassits Á. u. 65. IV/33. 1119

C Plus/4, C16 tulajdonosok figyelem! Elcserélném teljesen új 1551-es floppyt 1541/II-esre vagy 1541-esre, esetleg elcserélném. Több mint háromszoros sebesség, párhuzamos adatátvitel! 1541-essel teljesen szoftverkompatibilis. TVC 64 k jó állapotban, olcsón eladó. Kiss Zoltán, Lányok, Petőfi u. 28. 7759

C16, Plus/4 programokat cserélek. Kovács Arthur, Szeged, Engels u. 11. 6771

C128, CCP/M, C64 üzemmódban működő játékok és felhasználói programok cserélek, veszek. Leírások is érdekelnek. Mészáros Károly, Pécs, Sommelweis u. 36. 7623

Enterprise programok cserélek (kazettán). Varga Zoltán, Székesfehérvár, Lovóde u. 9/C. IV/2. 8000

Enterprise programok cserélek. Horváth Gábor, Pécs, Egri Gy. u. 100. 7632

Enterprise magnóval, 22 db gyári programmal vagy C Plus/4 számítógéppel magnóval elcserélném Spectrum 48 k + interfész + joystick + magnóra, vagy eladnám. Horváth András, Budacsany, Park u. 35. 8261

Spectrumhoz játépprogramok cserélek. Listát kérek! Sándor János, Kaposvár, Hérafa u. 4. 7400

Spectra 48-as felhasználói és színvonalas játépprogramok cserélek. Listát kérek! Horváth Miklós, Pécs, Csokonai u. 14. 8500

Ez a rovatunk KODEX 2000 szövegszerkesztővel készült.

EPSON®



Az EPSON RX—80-as típusú mátrixnyomtatót nem azért ajánljuk Önöknek, mert világbajnok volt a tartós nyomtatásban (2904 órát megszakítás nélkül nyomtatott), még csak azért sem, mert csupán három gomb is elegendő a kezeléséhez, vagy csak hozzá kell illesztenie személyi számítógépéhez, és egy megfelelő programmal máris üzemképes lesz, hanem azért:

- mert **megbízható**,
- mert **nagy teljesítményű** (100 karakter/s),
- mert **könnyű** (5 kilogramm),
- mert a **nyomtatófej és festékszalag-kazetta hosszú élettartamú**,
- mert egykártyás felépítésű és **könnyen javítható**,
- mert karakterkészlete tartalmazza a **magyar ábécét** is,
- mert **53 vezérlőparanccsal** rendelkezik,
- mert **96 ASCII karaktert, 32 grafikus karaktert és 11 nemzeti karaktert tartalmaz**,
- mert a cég a **nyomtatópiac 35,5 százalékát uralja**,
- mert **12 hónapig garanciális**,
- mert az EPSON nyomtató **más, mint a többi...**

Folyamatosan kapható valamennyi Centrum Áruház Műszaki Osztályán!

	Fogy. ár:
EPSON RX—80 mátrixnyomtató	56 400,— Ft
Ékezetes magyar ABC E—PROM BŐVÍTŐ	2 030,— Ft
Centronics PC kábelcsomag	4 730,— Ft
Enterprise—128 kábelcsomag	4 730,— Ft
Commodore soros interface kábel- és softwarecsomag	18 000,— Ft
Commodore 600/700 PIC printer interface chain, kábel- és softwarecsomag	20 800,— Ft
TV computer nyomtatókábelcsomag	1 630,— Ft

Figyelem! Valamennyi kábelcsomag mellé díjtalanul adjuk a Data Becker sorozat Az EPSON nyomtatók című könyvét.

**KERESSE ÁRUHÁZAINKBAN
AZ EPSON RX—80-as típusú
mátrixnyomtató részletes
műszaki tájékoztatóját!**

BUDAPEST KONGRESSZUSI KÖZPONT
1123 Budapest XII., Jagelló út 1-3.



Computing and Management
Organization Services

Rechentechnisches und
Leitungsorganisatorisches
Unternehmen des Zentralen
Amtes für Statistik

Предприятие по
вычислительной
технике и
организации
делопроизводства

KSH Számítástechnikai és
Ügvyitelszervező Vállalat

1145 Budapest, Szugló u. 9-15.

1140 Budapest 70, Pf.: 4

Telefon: 642-000 Telex: 22-6216

Újdonságainkkal várjuk Önt

az Orgtechnik-Compfair '88 kiállításon