



**MIKROSZÁMÍTÓGÉP  
MAGAZIN**

**A NEUMANN JÁNOS SZÁMÍTÓGÉP-  
TUDOMÁNYI TÁRSASÁG LAPJA**  
ÁRA: 30, - FORINT

**1985/6**



# VT 16 személyi számítógép



16 és 8-bites feldolgozás,  
MS/DOS, CP/M 86 és CP/M 3,0 kompatibilis operációs rendszerek,  
szoftver kompatibilitás IBM személyi számítógéppel,  
kiváló minőségű grafikus ábrázolás,  
nagy operatív és háttértár kapacitás,  
távadatfeldolgozás.



Információ:  
VIDEOTON Számítástechnikai Gyára  
1021 Budapest, Vörös Hadsereg útja 54.  
Telefon: 213-187





**A kiadvány  
a Tudományos  
és Informatikai  
Intézet  
együttműködve készül**

**A szerkesztő bizottság  
vezetője:  
Kovács Győző**

**Munkatársak:  
Broczkó Péter  
(hírek)  
Buday György István  
(személyi számítógépek)**

**Jakab Ágnes  
(ember-gép kapcsolat)  
Kovács Győző  
(levelezés)**

**Lindner László  
(sakkprogramozás)  
Pataki Ernő  
(programozástechnika)**

**Petróczy Judit  
(könyvek)**

**Pogány Csaba  
(alkalmazástechnika, tantolyam)**

**Simonyi Endre  
(klub)**

**Takácsy Ildikó  
(favágás)**

**Vadkerti János  
(µprogramok)**

**Varga András  
(iskola - számítógép)**

**Vass Nándor  
(alkalmazások)**

**Votisky Zsuzsa  
(játékprogramok)**

**Zárda Sarolta  
(piac)**

**A szerkesztőség munkatársai:**

**Kardos Zsuzsa  
Nacsa Sándor**

**Felelős szerkesztő:  
Könyves Tóth Pál  
Szerkesztőség:  
1027 Budapest II., Fő utca 68.  
Telefon: 154-250**

**Kiadja a Delta Szaklapkiadó  
és Műszaki Szolgáltató  
Leányvállalat  
Felelős kiadó:  
Faklen Pál igazgató  
1442 Budapest, VII., Garay utca 5.  
Telefon: 415-583, 215-440**

**Terjeszti a Magyar Posta  
Előfizethető  
bármely postahivatalban,  
a kézbesítőknél,  
a Posta hírlapüzleteiben  
és a Posta  
Központi Hírlap Irodában  
(Budapest V., József nádor tér 1.  
Postacím: 1900 Budapest)**

**vagy postautalványon,  
valamint átutalással  
a PKH 215-96162  
pénzforgalmi jelzőszáma  
Előfizetési díj:  
egy évre 180,- Ft,  
fél évre 90,- Ft.**

**Szedte  
a Nyomdaipari Fényszedő Üzem  
(857901/9)**

**Nyomás:  
Petőfi Nyomda, Kecskemét,  
Külső Szegedi út 6.  
(85/50066)  
Telefon: 28777  
Felelős vezető:  
Ablaka István igazgató**

**INDEX: 25629  
ISSN 0236-6088**

**Címképünk:**

## COMPUT '80

**64 kb-átos operatív memóriával  
rendelkező irodai munkahelynek,  
formatervezett mikroszámítógép,  
a COMPUT<sup>R</sup> család új, a VBKM  
Elektronikai Gyára által  
továbbfejlesztett tagja  
és az IPW-1 tip. hálózat-  
kondicionáló, mely szavatolja  
a zavartalan áramellátást,  
a számítógép üzembiztonságát.  
VBKM Elektronikai Gyar  
1475 Budapest, Pf. 86  
X., Venyige u. 3.  
Tel.: 476-590**

## Tartalom

A TV-BASIC és a tömegoktatás	2
Akik csatát veszíthetnek, de háborút nem	11
A technológiák szerepe	12
Pályázati felhívás	18
Iparunk fejlődéséhez nélkülözhetetlen a mikroelektronika	18
Tudáspróba	27
Adok - veszek - cserélek	31

## ISKOLA-SZÁMÍTÓGÉP

Közelítő értékek használatával kapcsolatos hibajelenségek	4
Gépi kódú rutin BASIC sorokká alakítása	5
Két bajtot kereső program	7
Gépi kódú program beírása külső magnóról	7

## TANFOLYAM

Alapozás XIII.	8
----------------	---

## PROGRAMOZÁSTECHNIKA

Formulák ábrázolásmódjáról	16
----------------------------	----

## µPROGRAMOK

Az UNIN rutin	20
Diagramszerkesztő	21
Scroll	22
Magyar betűkészlet	22
A DOS 5.1 program	23
Karaktergenerátorok	24
Ujjgyakorlatok a képernyőfájrra	24
Gépi kódú rutin BASIC sorok átszámozásához	25
Megszakítás I.	26

## PIAC

Mire jó a C16?	28
Micsoda kiszolgálás!	31

## µKLUB

Építsünk számítógépet! XI.	34
Ki ad magyarázatot?	35

## JÁTÉKPROGRAMOK

	37
--	----

## FÓRUM

	38
--	----

## AZ OLVASÓ ÍRJA

	40
--	----

## SAKKPROGRAMOZÁS

Lépésről lépésre	42
Ismét magyar sakkprogram a nemzetközi arénában	43

## KÖNYVEK

	45
--	----

## HÍREK, ÉRDEKESSÉGEK

	46
--	----



# A TV-BASIC és a tömegoktatás

*„Akiben van tehetség, köteles azt kiművelni a legfelsőbb fokig, hogy embertársainak ennél nagyobb hasznára lehessen. Mert minden ember annyit ér, amennyit embertársainak használni, hazájának szolgáltni tud.”*

KODÁLY ZOLTÁN

Sohasem szerettem, ha valaki igen kevés tapasztalat birtokában, viszonylag kevés adatból általánosít, én mégis elkövettem ezt a hibát – ha jól számolom, legalább féltucatszor az utóbbi fél évben. Felettebb szívesen állítom ugyanis, hogy a TV-BASIC bebizonyította, hogy a távoktatásnak vagy tömegoktatásnak nálunk is jövéje van, és talán egy ilyen oktatási rendszer fokozatos bevezetésével kellene megoldani a tömegképzés felsofoktatásának problémáját.

Maradjunk egyelőre az informatikánál, és nézzük a tényeket. 1985. január 16-tól tizenkét héten át mutatta be a televízió a TV-BASIC-et, amelynek kimondott célja a főleg nálunk használt gépeken legelterjedtebb programozási nyelv megtanítása volt. Folyamatosan vitakoztam azokkal a hazai és külföldi szakemberekkel, akik vitatták a választás – a BASIC nyelv – jogosságát, mondván, hogy ennél sokkal fejlettebb nyelvek is vannak (például a PASCAL), miért nem azokat tanítjuk. A válasz: mert hittük és hisszük, hogy programozási nyelvet papíron tanítani és tanulni nem lehet, a nyelv tanítása gyakorlat nélkül fátátkát sem ér. Gyakorlatot pedig csak számítógép mellett üve lehet igazán szerezni.

Amikor a sorozatot elkezdjük, a hazai számítógéphez nem volt túl rossz, annak ellenére, hogy lényegesen kevesebb gép volt az országban, mint amennyi ma van. Akkor már befejeződött a középiskolák számítógéppel való felszerelése (kb. 2000 gépet adtak át az oktatásnak), az illetékesek úgy becsülték, hogy személyi tulajdonban is legalább 20 ezer gép volt. Arra számítottam, hogy az adásokat 100–150 ezer néző figyeli rendszeresen, és a sorozat végén kb. 20 ezren fognak vizsgázni. A hivatalos adatok szerint az előzőt alá, az utóbbit túlbecsültem, hiszen kb. 250 ezren nézték az adást (kb. annyian, mint a Tv-híradó 2. kiadását), és 2799-en jöttek el vizsgázni (több, mint hatvezren jelentkeztek).

Miután a számítógépen való gyakorlatát komolyan gondoltuk, megszerveztük a µKlub mozgalmat részben az NJSZT szervezetekben, részben a Tudománytervezési és Informatikai Intézet segítségével az iskolákban, de csatlakozott a mozgalomhoz az OPI, az FPI, a KISZ, a TIT és a Népművelési Intézet is. Nagyon sokat jelentett, hogy a Magyar Űrhadsereg illetékes vezetői is felkarolták az ügyet, és lehetőséget adtak a sorállományú katonáknak, hogy a vizsgára jól felkészülhessenek. Ez egyébként az eredményen is látszott: a katonák az országos átlagnál mintegy 50 százalékkal jobb eredményt értek el. Igazságtalan lennék, ha nem emlékeznék meg azokról az intézményekről, amelyek nemcsak a saját dolgozóiknak, de

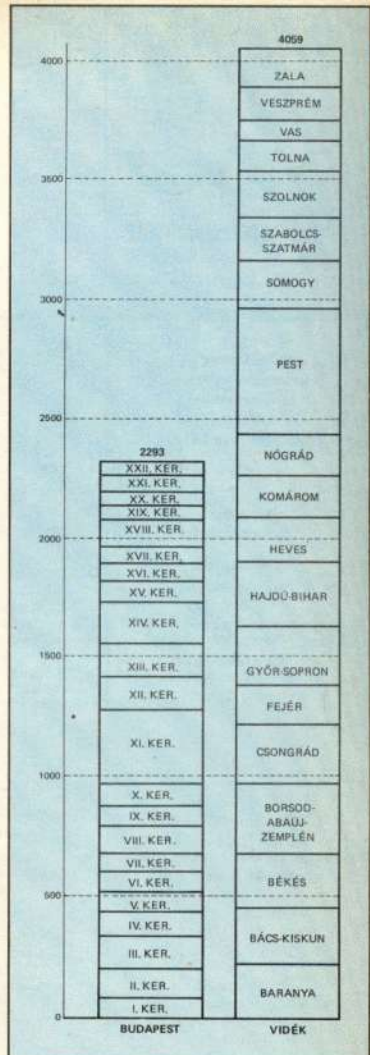
külső érdeklődőknek is megengedték, hogy munkaidő után a nem használt gépeken dolgozhassanak. Az eredmény tehát társadalmi összefogásból származott, ami önmaga is jelzi, hogy milyen sokan felismerték az informatika jelentőségét.

A tanfolyam értékmérője a vizsga. Miután nem lehetett tudni, hogy hányan fognak vizsgázni, tesztlapokat készítettünk a SZÁMALK és az ELTE szakembereivel. A mércét magásra tettük, mert a vizsgázóktól felsőfokú programozási nyelvismeretet kívántunk, ez volt a feltétele a bizonyítvány megszerzésének. A vizsgára vidékről kétszer annyian jelentkeztek, mint Budapestről, és nagyjából ez volt a valóban vizsgázók aránya is. Erdekesen alakult a vizsgázók életkori megoszlása: más volt az arány a férfiaknál és más a nőknél. A férfiak közül a tizenévesek jöttek el a legtöbben vizsgázni, a legifjabb vizsgázó 8 éves volt, a legidősebb 78. A nőknél arra számítottunk, hogy ez a tanfolyam leginkább a egyeslen lévő kismamákat fogja érdekelni, és így is történt. A legtöbb nő a 26–35 éves korosztályt képviselte, és közülük került ki az eredményesen vizsgázók jó része is.

Nem hagyhatom megjegyzés nélkül azt a tényt, hogy a főiskolások és az egyetemisták korosztálya gyakorlatilag távol maradt a vizsgától, de a tanfolyamtól is. Beszéltem néhány fiatalabbal, akik elmondták, hogy a tanfolyam azért nem érdekelte őket, mert az ott tanultakat nem tudják az egyetemen, főiskolán hasznosítani. A TV-BASIC túlságosan gyakorlati volt, az egyetemen a programozásotatás sokkal inkább elméleti. Az egyik beszélgetőpartneremnek például strukturált FORT-RAN-t kellett tanulnia, abban írt programokat, amelyeket a tanár korrigált, de az egyetemi évek alatt számítógép mellé még nem tudott leülni. Azt mondták nekem, hogy amíg az egyetemeken nem áll elegendő szabad használatú számítógép vagy terminál a hallgatók rendelkezésére, addig a TV-BASIC-et meghagyják a középiskolásoknak.

Ezzel szemben viszonylag sok 23–32 éves fiatalember jött el vizsgázni. Azt hiszem azért, hogy pótolják az elmulasztottakat. Ez a tény azért mást is mutat. Nevezetesen, hogy a kezdő dolgozók úgy érzik, a munkahelyen jobban meg fogják becsülni őket, ha az informatikában is képzik magukat. Megragadták tehát ezt az alkalmat is, hogy a szükséges tudást megszerzezzék.

A vizsga előtt elhatároztuk, hogy az a vizsgázó kap bizonyítványt, aki a 75 kérdéses tesztlap 60 kérdésére felel. Utólag a határt 57 pontban állapítottuk meg, mert 3 kérdést „diszkvalifikálnunk” kellett, ti. ezekre nem lehetett egyér-



A budapesti és a vidéki vizsgázók aránya az 1985. évi TV-BASIC tanfolyamon

telmü választ adni. A férfi vizsgázók többsége 60 pont körüli eredményt ért el, a nők többsége 45–55 közötti pontszámmal végzett.

Nyilvánvalóan mindenki felmerül a kérdés, hogy miért? A válasz nem egyszerű. A júliusban tartott norfolki (USA) konferencián (World Conference on Education in Computer Technology) egy hosszú és érdekes vitán vettem részt, ami pontosan ezt a kérdést feszegette. Ott kifejtették, hogy a hölgyek kisebb érdeklődésének okát több tényezőben kell keresni. Ezek közül megemlítem azt a rossz társadalmi beidegződést, hogy az informatika férfias pálya, továbbá azt a nevelési szisztémát, hogy a kislányokat 8–9 éves koruktól kezdve (például a skandináv államokban) főleg lányoknak való dolgokra: főzésre, mosásra, gyereknevelésre tanítják, ami elveszi az időt



gyerekektől, és így nem tudnak informaticus-  
val foglalkozni.

A vita egyik résztvevője érdekes kísérletbe  
kezdett, amelynek első eredményeit be is mutat-  
ta. 5-6 éves fiúkat és lányokat ültetett a számi-  
tógép mellé, és azt vizsgálta, hogy később ho-  
gyan fejlődik a gyerekek számítógépes tudása  
mind a lányoknál, mind a fiúknál. Egy párhuz-  
amos kísérlet keretében nagyobb, 10 év feletti  
gyerekeket, ugyancsak fiúkat és lányokat ve-  
gyeszen tanított, és ott is vizsgálta a fejlődést,  
illetve a lemorzsolódást. Azt látta, hogy ha a  
gyerekek 8 éves koruk előtt kezdik el a számító-  
géppel a munkát, akkor a lányok ugyanúgy  
érdeklődnek a gépek iránt, mint a fiúk, és leg-  
alább úgy értenek is az informatikához. Azok-  
nál a gyerekeknél, akik később, 10 éves koruk  
után kezdik el a számítógépes foglalkozásokat,  
tehát abban a korban, amikor már erősen ér-  
deklődnek a világ dolgai iránt, a helyzet ugyan-  
az, mint nálunk a TV-BASIC esetében: a számi-  
tógép környékén negyed-ötöd-tizedet annyi a  
lány, mint a fiú.

Ezért nem szabad megállni a középiskolák  
számítógépesítésénél, és ezért lesz igazán nagy  
jelentőségű az általános iskolák számítógéppel  
való felszerelése, mert akkor valóban eltűnik az  
informatikában a lányok és a fiúk között ma  
még meglévő, sajnos jelentős tudás- és érdeklő-  
désbeli különbség is.

A korábbi TV-BASIC beszámolóim óta eltelt  
néhány hónap, és ezalatt a fejlődés persze nem  
állt meg. 1985-ben a magántulajdonú gépek  
száma kb. megnégyesződött, és egyben igen  
lényeges minőségi változás is látszik. Az embe-  
rek egyre több Commodore 64-es gépet vásá-  
rolnak, a kisebb, például ZX81 gépek iránti  
érdeklődés erősen csökkent. Itt-ott már nagy  
kapacitású, professzionális számítógép (pé-  
ldául Macintosh, IBM PC) is megjelent a laká-  
sokban. Nem kell különösebb jóstehetség ah-  
hoz, hogy lássuk: a jövőben ez az irányzat  
folytatódik.

A hazai személyszámítógép-gyártók nemrit-  
kán elérik az évi néhány száz, de az 1-2 ezer  
feletti példányszámot is, és egyre megbízhatóbb  
gépeket ajánlanak. Nekem úgy tűnik, hogy a  
gyártás ütemétől az alkalmazói igény messze  
elmarad, azzal nem nő arányosan. Ebben való-  
színűleg közrejátszik az a tény is, hogy viszony-  
lag kevesen tudják, hogy mire is lehet és kell a  
számítógépet alkalmazni, hiányzanak az ily  
módon képzett emberek.

Nyilvánvaló, hogy ezt a több millióra becsül-  
hető alkalmazói tömeget hagyományos módon,  
a hagyományos esti iskolákban, illetve az egye-  
temeken elfogadhatóan rövid idő alatt nem le-  
het kiképezni. Erre csak egy jól szervezett tö-  
megoktatási rendszer képes. Ezért nem lehet  
már abbahagyni a TV-BASIC jellegű oktatást,  
sőt azt általánosan elfogadott oktatási intéz-  
ménnyé kell fejleszteni. Túl szép lenne cél-  
ként kitűzni, hogy hozzuk létre például az  
angol vagy mondjuk a holland Open Uni-  
versitynek megfelelő általános oktatási intéz-  
ményt, ehhez talán ma még nem értek meg a  
feltételek. De ahhoz igen, hogy legalább az  
informatikában megkezdjük a tömegokta-  
tást, mert enélkül a számítástechnikai ismer-  
etek társadalmi méretű fejlesztése nem lesz  
megvalósítható.

KOVÁCS GYÖZÖ

# FORTH PROGRAMOZÁSI NYELV - NAGYSZÁMÍTÓGÉPEN

A mikroszámítógépek terjedésével pá-  
rhuzamosan nő a FORTH programozási  
nyelv híveinek tábora. Ennek oka nem propa-  
gandatámpány vagy egy világgé-  
támozgatás, hanem e nyelv páratlan tulaj-  
donságai.

## HATÉKONYSÁG

A rövid átfutású fejlesztés, a kis memo-  
riaigény és a gyors programfutás általa-  
ban egymásnak ellentmondó követelmé-  
nyek. Eppen ezért nehezen hiszi el, aki  
még személyesen nem győződött meg róla,  
hogy a FORTH, amely egy gyors fejleszté-  
st lehetővé tevő interaktív programozási  
nyelv, tipikusan az ASSEMBLER kód-  
nál tömörebb, annál csupán kb. kétszer lass-  
abb, a hagyományos magas szintű pro-  
gramozási nyelveknél pedig gyorsabb  
programot eredményez.

## NYELVI STRUKTÚRA

A hagyományos programozási nyelvek  
(COBOL, PL/I, PASCAL) korlátosságokká  
probalgóját megvédeni a programozót saját  
hibáitól és biztosítani a világos, áttekinthető  
programszöveget. Az ASSEMBLY nyelv  
ezzel szemben hozzáférést enged a hard-  
ver minden lehetőségéhez, de nem támo-  
gatja jól strukturált programok írását, és a  
forrásszöveg nehezen követhető. A  
FORTH ebben a tekintetben is egyedi  
kompromisszum: jól karbantartható, jól  
strukturált, moduláris programot eredmé-  
nyez, és mégis megenged mindent, amire  
az adott gép képes.

## ÁTVIHETŐSÉG

Nehéz napoknak néz elébe az, akinek  
egy már megrit programot másik számító-  
géptípusra, eltérő operációs rendszerre  
kell átdolgoznia. A FORTRAN-tól a PAS-  
CAL-ig ezt a problémát a nyelvek szabvány-  
osításával próbálták megoldani. A gyakorlat  
bizonyította ennek a megközelítésnek a  
kudarcat: az egyes fordítóprogramok  
közötti apró eltérések és a szabványokon  
túlmenő szolgáltatások miatt egy program  
átvitelétől a fordítóprogramra általánosan  
nagy feladat. A FORTH a programozó  
kezebe teszi a megoldást: tetszőleges módon  
megváltoztathatja, kiterjesztheti magát a  
fordítóprogramot, ezzel maga állítja elő új  
„fordítóprogramot”. Mi több, a FORTH hívei  
magát a programozást is úgy tekintik, mint  
a nyelv lépésről lépésre való kiterjesztését  
addig, amíg az alkalmazás nem válik az  
adott feladat egyszerű megoldására.

Az átvihetőség kérdése azért is fontos,  
mert ennek megoldásával a közelmúlt nagy-  
sikerű mikrogépes programjai (spreadsheet  
programok, adatbáziskezelők, word pro-  
cesszorok stb.) a mini- és nagygép-felhasz-  
nálók számára is hozzáférhetővé tehető.

Ha a programok átvihetők, akkor mód  
nyílik keresztfeljesztésre is: nagygépes  
feladatokat mikrogépesen lehet megoldani,  
ezzel tehermentesítve a nagygépet és javítva  
a munkakörülményeket: a programozó  
munkáját akár haza is viheti.

## KIKÉPZÉS

Várhatóan egyre több olyan programozó  
kerül nagygépes projektekbe, aki mikrogé-  
pen tanulta a szakmát, és legszívesebben  
mikrogépes vagy ahhoz hasonló környezet-  
ben dolgozik. Ezek szakértelmét közvetlenül  
lehet hasznosítani, ha mindhárom gépkat-  
egóriába (mikro-, mini- és nagygép) hozzáfér-  
hetővé tesszük a FORTH nyelvet.

## PROTOTÍPUS-KÉSZÍTÉS

Számítógépes projektek vezetőinek régi  
problémája a munka előrehaladásának  
mérése. Előfordult már, hogy az elkészült  
rendszer nem felelt meg a felhasználók-  
nak, hogy a határidő eléréstével a fej-  
lesztők semmi működőt nem tudtak produ-  
kálni... A legjobb ellenszernek a prototí-  
pus-készítés tűnik: meg kell ragadni a  
probléma magját, és ennek megoldására  
már a projekt elején működő, korlátai kö-  
zött használható programokat írni. Ha a  
végső megoldás ennek a „pilot-rendszer-  
nek” a fokozatos fejlesztésével, a korlátok  
lebontásával készül, akkor lehetőség van  
az előrehaladás folyamatos ellenőrzésé-  
re, és a munka folyamán mindig van egy  
demonstrálható, korlátozottan használható  
prototípus, ami sokkal többet ér a meg-  
bizónak annál a kijelentésnél, hogy „a  
rendszer 90%-ig készen van”. A FORTH  
nyelv és a hozzá tartozó filozófia kifejezet-  
ten támogatja az ilyen munkamódszert.

Ha érdeklődik a FORTH programozási  
nyelv és a FORTH sajátos programfejlesztési  
filozófiájá iránt, olvassa el *Leo Brodie*  
Starting FORTH és *Thinking FORTH* című  
könyveit. *Lipovszki-Subai-Beszeda* FORTH  
programozási rendszer és nyelv (LSI ATSZ)  
című könyve magyar nyelven olvasható. SIEMENS  
BS 2000, IBM OS/TSO és OS/GUTS  
környezetben működő FORTH iránt érdeklő-  
dőknek rendelkezésére áll a SZÁMALK In-  
formatikai Főosztályon dr. Káldi Tamás a  
853-111/133-as telefonszámán.

(x)



## Közelítő értékek használatával kapcsolatos hibajelenségek

Sokan azt hiszik, hogy a számítógépek pontosan számolnak. Nem gondolnak arra, hogy sok esetben szükség szerint nem lehet pontosan számolni, csak valamilyen közelítő értéket használhatunk. Például az  $1/3$  számot csak közelítőleg adhatjuk meg véges tizedes (vagy kettes) törttel. Irracionális számot is csak közelítő pontossággal adhatunk meg.

A véges számú számjegy használata miatt a számítógép az esetek jelentős részében még a pontos adatokból kiszámított értéket is csak kerekítve adhatja meg. Figyelembe kell venni azt is, hogy a számítógép közelítő eljárásokat használ. Például a HT-1080Z iskolaszámítógépen a PRINT SQR(25)-5 parancs eredménye 9.53674E-07 = 0.000000953674. Minthogy a számítógép az 5 számot pontosan ábrázolja, ezért ez azt jelenti, hogy a gép négyzetgyökvo-nási algoritmus szerint  $\sqrt{25} \approx 5.000000953674$ . Ezt a számot hat értékes jegyre kerekítve adó-dik az 5.00000 szám. A PRINT SQR(25) pa-rancs hatására a képernyőn az 5 szám jelenik meg, mivel a gép a felesleges nullákat nem írja ki a képernyőre. Ugyanígy az A=SQR(25); PRINT A parancsok hatására a képernyőn az 5 szám látható, de az A változóban nem a kijelzett, hanem a belső ábrázolásiérték tá-roldódik. Ez a kis eltérés sok hiba forrása lehet. Például

```
10 FOR I=21 TO 30
20 SQR(I) < 5 THEN PRINT "*"
   : ELSE PRINT "": I
30 NEXT I
```

program I=25 esetén ?-et ad \* helyett.

Ha egy elfogadható pontosságú közelítő értékből kiindulva sok számítást végzünk, akkor előfordulhat, hogy a számolás során a kis hibák összegződnek, és a végeredmény jelentős hibát tartalmaz. Néhány ilyen hibá-ra adunk példát.

Jelölje  $x$  az  $x^2 = x + 1$  egyenlet negatív gyökét, azaz legyen  $x = (1 - \sqrt{5})/2$ .  $x$  irracionális, ezért mind kettes, mind tizedes tört alakban csak közelítő pontossággal adható meg. Az  $x$  segítségével két sorozatot adunk meg. Az egyik legyen

$$a_i = x^i \quad (i = 0, 1, 2, \dots)$$

Ezt a sorozatot így is megadhatjuk:

$$a_0 = 1, a_i = a_{i-1} \cdot x \quad (i = 1, 2, 3, \dots)$$

Azaz a hatványozás helyett ismételt szorzással számíthatjuk ki a sorozat tagjait.

A második sorozatot így adjuk meg:

$$b_0 = 1, b_i = x, b_{i+1} = b_i + b_{i-1} \quad (i = 1, 2, 3, \dots)$$

A megadásból nyilvánvaló, hogy  $b_0 = a_0$ ,  $b_1 = a_1$ . Továbbá

$$b_2 = b_1 + b_0 = a_1 + a_0 = x + 1 = x^2 = a_2$$

$$b_3 = b_2 + b_1 = x^2 + x = x(x+1) = x \cdot x^2 = x^3 = a_3$$

$$b_4 = b_3 + b_2 = x^3 + x^2 = x^2(x+1) = x^2 \cdot x^2 = x^4 = a_4$$

A gondolatmenet folytatásából következik, hogy  $b_5 = a_5$ ,  $b_6 = a_6$ , ... Azaz minden nem negatív egész számra  $b_n = a_n$ . Tehát a két sorozat egyenlő. (A két sorozat egyenlőségét valójában teljes indukcióval bizonyíthatjuk. Ennek mene-

te kiolvasható a fenti számolási menetből.) Írjunk programot a kétféle kiszámítási módra. Egy ilyen program a HT-1080Z iskolaszámítógépre az alábbi (a sorozatok a  $a_0, b_0$  tagjait nem írjuk ki):

```
10 A = 1 : B0 = 1 : I = 1
20 PRINT STRINGS(32, "*");
   STRINGS(32, " ");
30 X = (1 - SQR(5))/2
40 B1 = X
50 PRINT : PRINT TAB(20) "A(I)"
   TAB(36) "B(I)"
60 FOR I = 1 TO I + 9
70 A = A * X : B = B1 + B0
80 PRINT "I = "; I, A, B1
90 B0 = B1 : B1 = B
100 NEXT I
110 PRINT STRINGS(32, "*");
   STRINGS(32, " ")
120 AS = INKEY$
130 IF INKEY$ = "" THEN 130 ELSE 50
```

A 20-as, 50-es és 110-es utasítások feladta a képernyőn megjelenítendő kép megszerkesztése. A 130-as utasítás hatására a gép egy (BREAK-től és SHIFT-től különböző) billentyű leütéséig várakozik, és így a képernyőre irtakot láthatjuk. Egy billentyű leütése után megjelenik a sorozat következő 10 tagja. Futtassuk a programot. Az eredmény meglepő. Például  $a_{10} = 8.13064E-03$ ,  $b_{10} = 8.12316E-03$ .  $a_{10}$  és  $b_{10}$  harmadik számjegye eltér egymástól.

Még jobban eltérnek az eredmények a sorozatok következő 10 tagjánál. Ha  $i > 16$ , akkor  $b_i < 0$ , pedig váltakozó előjelű a sorozat.

További tagok kiszámításakor azt tapasztaljuk, hogy  $b_i < 0$  és abszolút értéke rohamosan nő.  $b_{128}$  at a számítógép már nem tudja ábrázolni. (A közölt eredményeket HT-1080Z számítógépen kaptuk. A program az esetleg szükséges módosítások után más számítógépeken is futtatható, és a lényegét tekintve hasonló „rendetlenséget” tapasztalunk.)

A tapasztaltak magyarázata a következő.

A számítógép a pontos  $x = (1 - \sqrt{5})/2$  helyett annak egy közelítő értékét (jelöljük ezt  $X$ -szel) számítja ki és tárolja. Jelölje  $\Delta$  az elkövetett hibát, azaz legyen  $\Delta = x - X$ . Ekkor  $x$  helyett  $X$ -szel számolva a

$$b_0 = 1, b_i = x, b_{i+1} = b_i + b_{i-1} \quad (i = 1, 2, 3, \dots)$$

sorozat helyett a

$$B_0 = 1, B_i = X, B_{i+1} = B_i + B_{i-1} \quad (i = 1, 2, 3, \dots)$$

közelítő sorozatot számoljuk. Erre a sorozatra

$$B_i = 1 + b_i, B_1 = X = x - \Delta = b_1 - \Delta,$$

$$B_2 = B_1 + B_0 = (b_1 - \Delta) + b_0 = b_2 - \Delta$$

$$B_3 = B_2 + B_1 = (b_2 - \Delta) + (b_1 - \Delta) = b_3 - 2\Delta$$

$$B_4 = B_3 + B_2 = (b_3 - 2\Delta) + (b_2 - \Delta) = b_4 - 3\Delta$$

Azt tapasztaljuk, hogy a hiba mindig az előző két tag hibájának összege. Általában ha

$$F_i = Q, F_1 = 1,$$

$$F_{i+1} = F_i + F_{i-1} \quad (i = 1, 2, 3, \dots),$$

akkor

$$B_i = b_i - F_i \Delta \quad (i = 0, 1, 2, \dots)$$

Minthogy a pontos  $b_i$  az  $i$  növekedésével „nullához közelít”, az  $F_i$  pedig a képzési módból következően tetszőlegesen nagy lesz, ezért nagy  $i$ -re  $B_i \approx -F_i \Delta$ .

Ez az összefüggés megmagyarázza a tapasztalatot. Olyan számítógépen, ahol  $\Delta = x - X$  negatív, nagy  $i$ -re  $B_i$  pozitív lesz.

Hasonló jelenséget tapasztalunk, ha  $x$  az  $x^2 = 1 - x^2$  egyenlet pozitív gyökét jelöli, azaz  $x = (\sqrt{5} - 1)/2$  és az

$$a_0 = 1, a_i = a_{i-1} \cdot x \quad (i = 1, 2, 3, \dots)$$

$$b_0 = 1, b_i = x, b_{i+1} = b_i - b_{i-1} \quad (i = 1, 2, 3, \dots)$$

sorozatokot vizsgáljuk. Erről a két sorozatról is belátható, hogy egyenlők. Számítási szempontból viselkedésük hasonló „rendetlenséget” mutat, mint amit az előző esetben tapasztaltunk. Ekkor a programban a 30-as és 70-es sort kell megváltoztatni a következő módon

$$30 X = (SQR(5) - 1)/2$$

$$70 A = A * X : B = B0 - B1$$

Érdekes lehet alkalmas  $c, d$  számra az  $x^2 = cx + d$  egyenlet valamelyik gyökével kísérletezni, és a megfigyelt eredményre magyarázatot adni. A HT-1080Z iskolaszámítógépen érdekes eredményt ad az  $x^2 = 3x + 4$  egyenlet  $x = -1$  gyöke. Számítsuk ki ezt a gyököt a másodfokú egyenlet megoldóképletével. Ekkor  $x = (3 - \sqrt{25})/2 = -1$ . Mint korábban már volt róla szó, a gép által számított  $X = (3 - \sqrt{SQR(25)})/2$  számérték nem pontosan  $-1$ . Defináljuk a sorozatokat az

$$a_0 = 1, a_i = a_{i-1} \cdot X \quad (i = 1, 2, 3, \dots)$$

$$b_0 = 1, b_i = x, b_{i+1} = 3b_i + 4b_{i-1} \quad (i = 1, 2, 3, \dots)$$

előírásokkal. Könnyen belátható, hogy a két szorzat egyenlő, mégpedig

$$a_i = b_i = \begin{cases} 1, & \text{ha } i \text{ páros} \\ -1, & \text{ha } i \text{ páratlan} \end{cases}$$

Végezzük el a számítást a géppel. Használhatjuk a korábbi programot a következő módosítással:

$$30 X = (3 - \sqrt{SQR(25)})/2$$

$$70 A = A * X : B = 3 * B1 + 4 * B0$$

Az abból eredő hiba, hogy  $X$  nem pontosan  $-1$ , mindkét sorozat kiszámításánál jelentkeznek, de igen eltérő módon. Az első sorozatban  $a_1 = a_3 = a_5 = a_7 = a_9 = -1$ ,

$$a_2 = a_4 = a_6 = a_8 = a_{10} = 1,$$

A továbbiakban már jelentkeznek a pontatlan számítás hibája,  $a_{11} = -1.00001$ ,  $a_{31} = -1.00001$ ,  $a_{32} = 1.00002$ , ... Igen gyorsan változik viszont a másik sorozat. Tagjai egyre nagyobb abszolút értékű negatív számok. A magyarázatot az első példa mintájára nem nehéz megadni. A pontos  $x = -1$  helyett egy  $X = -1 - \delta$  ( $\delta > 0$ ) számmal számolunk. Könnyen belátható, hogy ha  $i$  nagy, akkor

$$B_i \approx -C_i \delta,$$

ahol

$$C_0 = 0, C_1 = 1, C_{i+1} = 3C_i + 4C_{i-1}$$

$$(i = 1, 2, 3, \dots)$$

Ez a sorozat pedig igen gyorsan növekszik.



## Gépi kódú rutin BASIC sorokká alakítása

A BASIC programok sok esetben tartalmaznak gépi kódú rutinokat a HT-1080Z esetében is. Ezeket általában a BASIC-ben DATA sorokban tároljuk, és a program elején a kívánt helyre POKE-oljuk.

Sajnos az adatokat kézzel kell beírni, ami fárasztó, időrabló tevékenység, ráadásul a gépelési hibákból adódó tévesztések, elnézések sok bosszúságot okoznak.

Gondolom, sokakban felmerült már a gon-

dolat, hogy jó lenne valamilyen segédprogrammal az assemblerrel megszerkesztett programot mindjárt BASIC DATA sorokká alakítani. Nos, ez a feladat egy kis ügyeskedéssel megoldható.

A µM 1984/2. száma ismertette a BASIC sorok tárolási módját a memóriában, de azért nem árt, ha megvizsgáljuk a DATA utasítás és az azt követő programsor megjelenési formáját a munkaterületen: például 10 DATA 5, 143, 92

Azonnal beláthatjuk, hogy az adatokat az interpreter nem egyszerűen a szám hexadecimális alakjában írja le, hanem helyértékenkénti ASCII kóddal, hexadecimális alakban. Ez érthető is, mert például a 143-at egyszerűen beírva egy bájtira, az IF utasítás tokenjét kapjuk (természetesen az adatokat elválasztó vessző is ASCII kóddal szerepel.)

Mindezek ismeretében most már írhatunk olyan programot, amely BASIC sorokat „késztí”, és a gépi kódú rutin kezdő-, illetve végcímeinek lekerdeezése után a következő algoritmus szerint működik:

- a munkaprogram END utasítása után a leendő sorpointer számára felszabadít két bájt
- sorszámot, DATA utasítást és szöközőket ír elhelyezi az adatot, vesszőt tesz (amíg a sor be nem telik)
- visszamenőleg beállítja a sorpointer
- ha a lemásolandó adatok végére ért, kiteszi az EOF kódot, és a változótable kezdetét jelző pointert az ezt követő bájtira állítja
- önmagát megsemmisíti.

A BASIC interpreter a változókat közvetlenül a program mögött tárolja. Annak biztosítására, hogy még véletlenül se fussunk bele a változótableba, két lehetőség kínálkozik: vagy a változótable mutatóját állítjuk „jó magasra” vagy „fal” sorokat írunk a munkaprogram után. Célszerűbb az utóbbi megoldást választani.

Az END után folyamatos sorszámozással kb. négyezer annyit aposztrófot kell beírunk, mint a bemásolandó adatok mennyisége (minden gépi kódú adatot biztonsági okokból 8 bit hosszúságúnak tekintve).

Ha a programot tovább akarjuk fejleszteni, hogy például fogadjon el a címeket hexadecimális alakban is, vagy ha bármit változtatunk, a 235-ös sorban levő V változó értékét állítsuk át úgy, hogy mindig az END-et követő második bájtira mutasson.

A program lefuttatása után már csak az aktuális program hozzáfűzése a dolgnak. Egyetlen program hozzáfűzése esetén nem érdemes az időt a MERGE vagy valamilyen APPEND beolvasztásával és indításával tölteni; ehelyett parancs üzemmódban is hozzáfűzhetjük a kívánt programot az adatok után az alábbiak szerint:

```
POKE 16548,(PEEK(16633)-2):
POKE 16549,(PEEK(16634))
```

```
>
> LOAD
```

```
>
> POKE 16548,233:
POKE 16549,66
```

> Végül egy jó tanács a programok egymáshoz láncolásához. Mielőtt elkezdenénk az össze-fűzést, a memóriában levő programot számozassuk át négytől tízes inkrementummal, hogy az utántöltés után kiadott RE (RENUMBER) ne találjon két egyforma sorszámot, mert ez az ugróutasításoknál zavart okozhat.

①	②	③	④	⑤	⑥	⑦
F8	42	0A	00	88	20	35
				2C	31	34
					33	2C
					39	32
						00

- ① sorpointer, ② sorszám (10), ③ a DATA utasítás tokenje, ④ szöközők, ⑤ az adatokat elválasztó vessző, ⑦ a sor végét jelző nulla

### GÉPI KÓDÚ RUTIN MÁSOLÁSA BASIC PROGRAMBA

```
205 DEFINT A-Z
215 INPUT "KEZDO CIM",K:K=K-1
225 INPUT "VEGCIM:",E
235 V=17634
245 POKEV,0:POKEY+1,0:V=V+2
255 S=0:I=0:U=0
265 S=S+10*POKEV,S-INT(S/256)*256:POKEY+1,INT(S/256):POKEY+2,136:POKEY+3,32:V=V+3
275 I=I+1:U=U+1:A$=STR$(PEEK(K+U)):FORR=2TOLEN(A$):V=V+1:POKEY,ASC(MID$(A$,R,1)):NEXT
285 IFK+U=ENDR=45THEN305
295 V=V+1:POKEY,44:GOTO275
305 I=0:J=V:FORZ=1TO3:POKEY+2,0:NEXTZ:V=V+2
315 J=J-1:IFPEEK(J)+PEEK(J-1)+PEEK(J-2)<>0THEN315
325 POKEJ,INT(V/256):POKEJ-1,V-INT(V/256)*256
335 V=V+2:IFK+U=ETHEN345ELSE265
345 POKE16722,V-INT(V/256)*256:POKE16723,INT(V/256)
355 POKE16633,PEEK(16722):POKE16634,PEEK(16723)
365 DELETE205-375
375 END
385
```

### PROBA FUTTATAS

```
>RUN
KEZDO CIM? 24576
VEGCIM? 24676
READY
>LIST
10 DATA 49,0,96,205,201,1,33,15,101,175,50,11,101,50,12,101,50,14,101,205,219,100,229,213,197,175,50,10,101,62,42,205,44,100,33,255,100,6,11,205,64,0,56,237,195
20 DATA 130,99,193,209,225,58,11,101,183,194,145,100,175,50,12,101,50,10,101,183,40,5,253,35,205,170,100,229,213,221,229,209,19,253,229,225,237,82,250,91,96,209,225,195,22
30 DATA 96,253,229,225,124,205,178,100,125,205,178
READY
>
```

## **—COMPUT-80—**

MIKROSZÁMÍTÓGÉP CSALÁDRA KÉSZÍTETT RENDSZEREINK:

- FŐKÖNYVI ÉS FOLYÓSZÁMLA-KÖNYVELÉS.
- DÍJBESZEDŐ VÁLLALATOK DÍJBESZEDÉSI ÉS FOLYÓSZÁMLÁINAK VEZETÉSE.
  - KERESKEDELMI RENDSZER.
  - PÉNZTÁRGÉPES HÁLÓZAT.

## **—COMPUT-80—**

KÖZPONTI SZÁMÍTÓGÉPPEL

- TANÁCSIGAZGATÁSI ALKALMAZÁSI SZOFTVEREK.
  - MUNKAÜGYI RENDSZER.
    - BÉRRENDSZER.
- KÉSZLET-NYILVÁNTARTÁSI RENDSZER.

ÁLTALÁNOS SZÁMÍTÁSTECHNIKAI SZOLGÁLTATÁSAINK:

- adatrögzítés,
- rendszerek üzemeltetése,
- rendszerszervezés,
- programozás,
- mikroszámítógépes rendszerek fejlesztése,
- többfelhasználós mikroszámítógépes rendszerek (hardver + szoftver) bérbeadása.

TOVÁBBI FELVILÁGOSÍTÁSSAL SZÍVESEN ÁLLUNK  
RENDELKEZÉSÜNKRE

a 131-072-es telefonszámon,  
illetve levélben:



## Két bajtot kereső program

```

10  ORG 7F00H
11  LOAD 7F00H
12  CALL 01C9H
13  LD HL,0000H      ; E cimtol Keres
14  KEZD:LD A,00C6H ; Az elso szam amit Keres
15  CP (HL)
16  JP Z,MEGVAN
17  INC HL
18  LD A,H
19  CP 0060H        ; Eddig vizsgal (6000H)
20  JP NZ,KEZD
21  PUSH HL
22  LD HL,SZOVEG
23  CALL 28A7H     ; Szoveg a Kepernyore
24  POP HL
25  CALL 0049H    ; Leutesre var
26  JP 4300H     ; Vissza az EDI-be
27  MEGVAN:INC HL
28  LD A,(HL)
29  CP 0007H     ; A masodik szam amit Keres
30  JP Z,KIIR
31  JP KEZD
32  KIIR:LD A,H  ; A-ba a felso bajt
33  CALL 3373H   ; LBYTE
34  LD A,L      ; A-ba az alsó bajt
35  CALL 3373H   ; LBYTE
36  LD A,000DH  ; Soremeles (vagy 20H szoKoz)
37  CALL 0033H
38  JP KEZD
39  SZOVEG: DB ODH      Tobbet nem talalok!
40  NOP          ; A szoveg lezarasa
41  END
    
```

A HT-1080Z gépet gyártó cég nem kényeztette el a felhasználót a ROM rutinok közkinccsé tételével: felkutatásuk a programozóra hárul. A memóriaterület átvizsgálását a gépre bízhatjuk. Az alábbi keresőprogram sok hasznos szubrutin megtalálását segítette.

Eredetileg az LBYTE-ot – egy bajt hexadecimális kiíratását – is meg kellett írnom, csak miután a „KERESŐ” megtalálta az EPROM-ban, került be szubrutinként a programba.

Célszerű a keresést a programrész legjellemzőbb utasításával kezdeni. Például egy hexadecimális számjegy kiíratásánál áruklódó négy jobbra léptető utasítás, de hát ez többféle lehet, ezért a 0FF-ek (RRCR) beírásával nem jártam eredménnyel. Az ADD A,07H (C6 07) viszont egyetlen helyen szerepelt! Így könnyű volt megtalálni a 3373H címen induló LBYTE-ot.

KASZÁS DEZSŐ

```

10  REM 1234567890123456
20  FOR I=0 TO 15
30  READ A
40  POKE 17134+I,A
50  NEXT
60  POKE 16526,238 :POKE 16527,66
70  X=USR(0)
80  POKE 16866,201
90  DATA 33,225,201,34,226,65,175,205
100 DATA 17,2,205,150,2,195,231,2
110 REM * Gépi kódu beolvasás külső magnóról *
120 END
    
```

## Gépi kódu program beírása külső magnóról

A HT-1080Z gép kényes pontja a magnó. Az automata szabályozós példányokkal gyakran csak a saját felvételüket lehet visszajátszani. Így szükségessé válik külső magnó használata, amely a gépen csak BASIC-ből érhető el az eredeti paranccsal.

A magnóválasztó szubrutin (CALL 0212H) a belső magnót indítja, ha az akkumulátorban 0 van, ha az A-ban 1 van, a külső magnó indul.

Az assembler EDITOR-ban az A nullázása a XOR A egybajtos utasítással történik. Ehelyett nem fér be az LD A, 0FFH. Szerencsére a magnóválasztó rutin előtt egy DEC A utasítás áll, így csupán eggyel előbbre kell címezni a szubrutint, és a külső magnó indul. Tehát

a meglévő gépi kódu másoló programunkban megkeressük vagy a géppel megkeresztjük a fenti szubrutint. (CD 12 02) és a 12H értéket 11H-ra írjuk át. Az EDI-nél ha a 4A23H és 4A37H címen 12H van a belső, 11H-nál a külső magnó indul. A 7F00H-nál kezdődő MCCOPY másolóprogramnak a 7F27H és a 7F6FH memóriáját kell átírnom.

Ha a belső magnó végleg felmondja a szolgálatot, az előbbi BASIC program elindításával beolvashatjuk külső magnóról a gépi kódu programokat. A 10-es sor 16 bajt memóriát foglal, amelyet a program majd felülír.

KASZÁS DEZSŐ

## Számítógép-fejlesztők Gyártók!

Nálunk azonnal kapható MOM 1800/900 típusú 5 1/4" hajlékonylemezes meghajtóegység. Tápegység és doboz nélkül ára 18 000,- Ft. Érdeklődni lehet a 252-880-as telefonszámon.

### Téli szabadság





**Digitális automaták építőelemének, pneumatikus operátorunknak matematikai modelljét keresve, láttuk, hogy a késedelem nélküli, azonnali válaszadás feltételezése megoldhatatlan nehézségeket okoz. Most a véges, nem nulla késedelmi reagálás esetét vizsgáljuk meg.**

Az „ideális”, azonnali reakciójú, késedelem nélküli dolgozó operátor nemcsak elérhetetlen műszaki vágyálom, hanem – amint láttuk – nem egyszerűsíthető, nem könnyíti a munkát, hanem új problémákkal rendkívül módon megnehezíti. Pedig azt várnánk, hogy az egyszerűsítő feltételeinkkel készült modellel könnyebben dolgozhatunk. Vágyálunk tehát csalókának bizonyult. Ha megvalósulna, lehet, hogy többet vesztenénk a vámon, mint amennyit a megvalósulás révén nyertünk. Az biztos, hogy az emlékezés, az információátvitel megvalósítása kérdésessé válna. Ne gondoljuk azonban azt, hogy az „ideális” operátorok minden esetben teljesen használhatatlanok. Nagyon is jól használhatók, sőt megfelelő körben nélkülözhetetlenek is. Szerepük hasonló az egészszámok szerepéhez. Az egészszámok – mint tudjuk – nem alkalmasak pl. minden egyes szakasz hosszúságának a jellemzésére. Az egészszámok halmazához hozzá kell venni más számokat is, hogy minden szakasz hosszúság egy számmal jellemezhető legyen.

A valós számok halmazának bővítése, a komplex számok létrehozása az előbbiekhöz hasonló, de még szemléletesebb példa. A komplex számok valós számokból és imaginárius számokból tevődnek össze. Ha egy valós számhoz hozzávesszünk egy imaginárius számot, az így kialakított számokkal több feladatot tudunk megoldani, mintha csak valós számokkal dolgoznánk.

A következőkben minden azonnal reagáló, végtelenül fürge operátorhoz hozzávesszünk egy olyan operátort, amely a lomhaságról, a tehetetlenségről gondoskodik. Ez a kiegészítés, hozzávétel természetesen többféle lehet aszerint, hogy *miel* és *aszerint* is, hogy *hogyan* végezzük a „bővítést”. A legegyszerűbb „tehetetlenítő” elem a közönséges késleltető. Lássuk tehát, mire tudjuk használni a késleltetővel „lassított”, „tehetetlenített” ideális operátorokat!

## Mi a késleltető?

Rendkívül fontos szerepük van az olyan operátoroknak, amelyek kimeneti jellemzőinek értékei bemeneti jellemzőinek értékei közül kerülnek ki. Ennek az operátorosztálynak fontos speciális részalmozását alkotják azok az operátorok, amelyek kimenetét (előbb-utóbb) minden bemeneti érték megjelenik, mégpedig pontosan „annyiszor”, „ahányiszor” az a bemeneten tapasztalható volt. További fontos speciális osztályba azok az operátorok tartoznak, amelyekre az igaz, hogy ha a bemeneten egy értéket egy másik időben megelőző, akkor ezek az érté-

# Alapozás XIII.

kek a kimeneten is ugyanilyen időbeli sorrendben jelennek meg. Ez utóbbi operátorok a sorrendtartó késleltető operátorok. Azért nevezzük így őket, mert hiszen kimenetükön a bemeneten tapasztalható értékek jelennek meg, mégpedig úgy, mintha a bemeneten tapasztalható, a késleltetőbe „belépő” értékek jelenének meg – bizonyos késéssel – a kimeneten, belépésük sorrendjében. Végül pedig a sorrendtartó késleltető operátorok speciális osztályát említjük, amelynek elemeire jellemző, hogy minden belépő érték megjelenik a kimeneten is, mégpedig minden esetben a belépéstől számított ugyanannyi idő elteltével. Ezek az állandó idővel késleltetők.

Jól szemléltethetők ezek az operátorosztályok egy kiállítóhelyiség példájával. Legyen a kiállítóhelyiségnek egy bejárata és több kijárata. (A bejáraton egyszerre csak egy ember léphet be.) Figyeljük a bejáratot és az egyik kijáratot. Nyilvánvaló, hogy a figyelt kijáraton csak a bejáraton valamikor belépett személyek (ők felelnek meg az értékeknek) távozhatnak. (Előfordulhat, hogy mindenki e kijáraton távozik.) A belépéstől a kilépésig eltelt idő személyenként különböző lehet. Az is lehet, hogy egy látogató előbb hagyja el a termet mint egy olyan másik, aki ő utána érkezik.

Ha viszont a kiállítóteremnek egyetlen bejárata és egyetlen kijárata van, minden látogató, aki a bejáraton belépett, a kijáraton is előbb-utóbb át fog haladni. Az persze most is előfordulhat, hogy lesz olyan, aki valaki után érkezik, de előtte távozik.

Ha megakadályozzuk a látogatókat egymás megelőzésében, akkor a belépés sorrendje nyilván azonos lesz a kilépésével. (Ez az „előzni tilos” intézkedés következménye.)

A kiállítás szolgáltatórendszer. Az ilyen szolgáltatói rendet „FIFO” elvűnek hívják, a „first in first out” kezdőbetűi alapján. (Az angol szakkifejezés jelentése – szabadon fordítva – az, hogy aki előbb lép be, előbb is lép ki, aki előbb kért, előbb kap stb.)

Végül pedig, ha pl. minden látogató azonos hosszúságú pályán, azonos sebességgel kényeserül haladni, akkor minden belépővel a belépéstől számított azonos idő elteltével a kijáratnál is találkozhatunk. Ez utóbbi eset felel meg az állandóidejű késleltetőnek, vagy más szóval a közönséges késleltetőnek.

Először ilyen közönséges késleltetővel fogjuk „kiegészíteni” a végtelenül fürge operátorunkat.

## Késleltetések állandóidejű késleltetővel

A közönséges, állandóidejű késleltetőnek két jellemzője van. Az egyik a késleltetési idő, a másik pedig a tartalma. A késleltetési időről tudjuk, hogy kicsoda. A tartalomról sem nehéz kitalálni, hogy mit jelenthet. A pillanatnyi tartalom a szóban forgó pillanatot megelőző, a

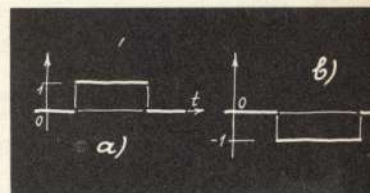
késleltetési idővel egyenlő időtartamú darabja a bemeneti folyamatnak.

Ha ebben a pillanatban „kapunk a kezünkbe” egy (közönséges) késleltetőt, kimeneti folyamatára egy darabig semmi befolyást nem tudunk gyakorolni. A kimeneti folyamat a kézhez kapást követően, a késleltetési időnyi időtartam leteltig, tőlünk függetlenül (de meghatározottan) zajlik. A késleltetőből, a késleltetési idő időtartama alatt, a kézhezkapás előtt meghatározott és kialakult tartalma fog „ürülni”. Ha a kézhezkapást követően azonnal elkezdjük „táplálni” a késleltetőt, azaz bemenetére általában kívánt folyamatot juttatunk, akkor e beavatkozások hatása legkorábban a késleltetési idő eltelté után jelentkezhet csak.

Előfordulhat tehát, hogy használat előtt a késleltető „ki kell üríteni”, illetve „fel kell tölteni” (azzal ami számunkra a jövőben kedvező lesz). Ilyen szempontból a késleltető egy csőhöz hasonlítható, amelyben egy előző tevékenység visszahagyott valamilyen folyadékot, amire nincs szükségünk. Ha más folyadékot kezdünk benyomni, a cső másik végén, egy ideig az előző folyadék fog távozni. (A keveredést egyszerűség kedvéért elhanyagoljuk.)

A késleltető „tartalma” tehát „tartalmi”, nem pedig késleltetési kérdés. Foglalkozunk most egyelőre csak a késleltető késleltetési idejével.

Világos, hogy két egymással sorbakapcsolt késleltetőből álló rendszer szintén késleltető

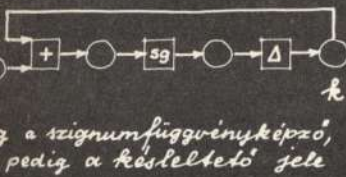


1. ábra

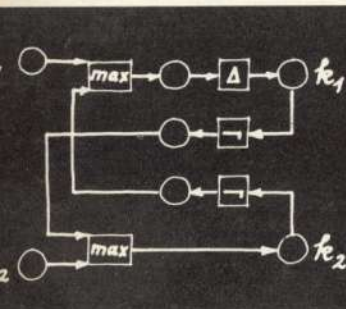
lesz, melynek késleltetési ideje a két késleltetési idő összege. (A két tartalom „egymás mellé tevése” révén pedig az eredő késleltető tartalmát állíthatjuk elő.)

Kapcsoljuk egy egyszerű késleltető kimenetét saját bemenetére! Az így kialakuló rendszer kimenetét nyilván periodikusan változó folyamatot figyelhetünk meg. Egy periódus hossza a késleltetési idő lesz. (Azért mondjuk, hogy egy periódus hossza, mert e periódushosszak minden pozitív egészszere is periódushossz lesz. Azt azonban nem állíthatjuk, hogy a késleltetési idő lesz a legkisebb periódushossz. Mint pl. olyankor, amikor a késleltető tartalma pl. pontosan két teljes periódusát tartalmazza egy periodikus folyamatnak.) Nyilvánvaló, hogy akárhány nulla késleltetési idejű késleltetőt





2. ábra



3. ábra

operátor „késlekedővé alakítási” feladatának megoldása sokféleképp elvégezhető. (A késleltetés fogalma még nem teljesen pontos. Itt definiáltunk egy késleltetés-fogalmat egy operátorral, de van más lehetőség is.)

Vizsgáljunk meg néhány egyszerű kapcsolást közönséges, állandó idővel késleltető operátorokkal! Felhívjuk a figyelmet, hogy az ilyenekkel való gyakorlatok nemcsak digitális berendezések megértéséhez, hanem sok más matematika alkalmazási feladat sikeres megoldásához is jó alapot szolgáltatnak. Nem túlzás, hogy ha az azt mondjuk, *nem lehet eleget gyakorolni* a késleltetővel való rendszerépítést és az ilyen rendszerek elemzését. Az ilyen rendszerek rendkívül sok olyan gyakorlati rendszer modellezésére is alkalmasak, amelyekre ma még egyáltalán nincsenek használható modellek. A késleltetőket tartalmazó rendszerekben mind előző értékekre támaszkodó operátorok működnek. Az előző értékek támaszkodó operátorok pedig – nézünk szerint – a tudomány mai fejlettségi szintjén a legfontosabbak közé tartoznak.

### Egyszerű kapcsolások késleltetővel

Befejezül néhány olyan egyszerű rendszer kapcsolatábráját közülük, amelyekben közönséges, állandó idejű késleltetők szerepelnek. A feladat a rendszernek minél pontosab megértése.

A rendszerekkel (operátorcsapatokkal) való foglalkozás során ne tévesszük szem elől, hogy a végtelenül gyors műveletvégzés feltételezése nehézségeket okoz. Vegyük észre azt is, hogy egy szakadás, egy „ugrás” egy folyamatban szintén műveletnek, értékmegváltoztató műveletnek tulajdonítható. Noha ez a végtelen nagy sebességű értékmegváltozás is ritka, legalább olyan ritka mint a végtelen nagy sebességű műveletvégzés, mégsem kell következetesen mindenholnán eltávolítani, mert – meglepő módon – a gyakorlatban sok, jól használható közelítő modellel élnyönsen (és problémamentesen) alkalmazható.

Még egyszer figyelmeztetünk azonban arra, hogy végtelen fürgeség nemcsak a megszokott műveletvégzésekben, hanem értékek véges, nem zérus megváltoztatásában sincs; nemcsak a megszokott műveletek eredményeinek kialakításához szükséges véges nem zérus idő, hanem értékek véges, nem zérus mértékű megváltoztatásához is, és ez utóbbi tevékenységet is természetes valamilyen operátornak tulajdonítani.

Vizsgáljuk a rendszerek viselkedését először úgy, hogy a bemenetekre az 1. ábrán látható különböző hosszúságú „impulzusokat” adunk. Mit tapasztalunk a 2. ábra esetében, ha a  $b$  pontra pl. a késleltető késleltetési idejének négyzeresége adunk az 1.a) ábrán látható folyamatot? A kimeneti pont megtanulja, megjegyzi az 1 értéket, „részokik” és addig biztosan megtartja, ameddig a  $b$  ponton – 1 vagy ennél kisebb érték nem jelentkezik. (Sőt még egy kicsit tovább is. Meddig?) Ha – 1-et adunk  $b$ -re (elégg hosszú ideig. Meddig?), akkor  $k$  erre „szokik rá”, és marad ezen az értéken, addig biztosan, ameddig a  $b$  ponton +1 vagy ennél nagyobb érték meg nem jelenik. (Sőt még egy kicsit tovább is. Pontosan meddig?)

Felfedeztünk tehát egy információáról operátorcsapatot. Ilyen tulajdonságú operátorcsapat végtelen sok van.

Még egy ilyet bemutatunk, amelynek ellentétben az előzővel, két bemeneti pontja van (3. ábra). A bemenetekre itt csak olyan folyamatokat kapcsolunk, amelyek értéke csak 0 vagy 1 lehet! (Ezek „fűj”, „nem fűj” állapotok) egymásutánját leíró (0–1) folyamatok.) E rendszerrel értékpárok megőrzését tudjuk megvalósítani. Rendszerünk közvetlenül csak a 0 és 1, ill. az 1 és 0 értékpár korlátlan ideig való megőrzésére használható.

Ha a bemenetek pl.  $b_1 = 0$  és  $b_2 = 1$  értéket tartjuk fenn egy ideig, (legalább meddig?) ez az értékpár „bizonyos idő múlva” (ez mennyi?) a kimeneteken is megjelenik és, ha a bemenetek azután  $b_1 = 0$ ,  $b_2 = 0$  állapotba kerülnek, a kimenetek állapota  $k_1 = 0$ ,  $k_2 = 1$  marad korlátlan ideig. Ezt az állapotot a  $b_1 = 1$ ,  $b_2 = 0$  állapot bizonyos ideig való fenntartásával változtatathatjuk meg. Az így kialakuló  $k_1 = 1$ ,  $k_2 = 0$  állapot korlátlan ideig fennmarad, akkor is ha a bemenetek „visszatérnek” a 0 állapotba. A  $k_1$  és  $k_2$  kimeneti pontpár tehát „részoktatható” a 0 és 1, ill. az 1 és 0 bemeneti értékekre, így a rendszer információ megőrzésére alkalmas (a „részokás” modellezésének képessége révén). A 0 és 0 értékpár és az 1 és 1 értékpár megőrzésére e rendszer nem alkalmas. Két ilyen rendszer felhasználásával azonban már szervezhetünk olyan operátorcsapatot, amellyel ez a feladat is megoldható. Ezzel majd még foglalkozunk.

Megjegyezzük, hogy a  $j$ -edik operátor 0-ból 1-et, 1-ből pedig 0 értéket képez.

Talán feltűnik, hogy nem minden végtelenül gyorsan működő operátorhoz csatlakoztathatunk késleltetőt. Végezze el e munkát az Olvasó! Utána pedig vizsgálja meg rendszerét! Ki fog derülni, hogy egy feladatkorban ugyanígy működik, mint az itteni egyszerűsített rendszer.

Pneumatikus operátorunk viselkedésének leírására használható matematikai modellt keresve, az „ideális”, végtelenül gyors matematikai műveletvégzőkkel kísérletezünk először. A nagymértékű egyszerűsítés – nem várt módon – nem egyszerű (esetleg túl egyszerű) megoldásra, hanem kezelhetetlenül nehéz próbákra vezetett. A végtelenül gyors működésű operátorokat egyszerű késleltetővel „lelassítva” azonban már – matematikailag – jól kezelhető rendszereket is sikerült kialakítaniuk. A modellek világában tehát lényeges előrelépést sikerült tennünk. Az természetes, hogy amíg a modellező eszközzel nem boldogulunk, addig nem beszélhetünk haladásról. Ez szükséges, de nem elégséges feltétel. Még kell még oldanunk azonban azt is, hogy jól működő modelleink működése a megkívánt mértékig hasonló is legyen a modellezett rendszer működéséhez.

Látni fogjuk, hogy sok esetben az egyszerű késleltetővel kiegészített „ideális” matematikai műveletvégző operátoraink elegendők lesznek pneumatikus (és más) eleven működő rendszereink elegendő pontosságú modellezésére. Mielőtt ennek mikténtje történik, késleltetővel kapcsolatos feladatokkal és „finomabb”, késleltetésre modellező rendszerekkel is foglalkoznunk kell még.





**INFORMÁCIÓTECHNIKAI  
VÁLLALAT**

Központ: Budapest V., Bécsi u. 8.  
Levél cím: 1369 Budapest, Pf. 314  
Telefon: 184-899 Telex: 22-4381 22-6841

## **ORSZÁGOS SZERVIZHÁLÓZATUNK**

korszerűen felkészített műhelyei útján végezzük a számítástechnikai berendezések műszeres vizsgálatát, bemérését, ellenőrzését.

### **VÁLLALJUK:**

adatrögzítő és feldolgozó berendezések,	garanciális
személyi számítógépek,	és garanciaidőn túli
kisszámítógépek,	javítását
számítógép rendszerek	szerviz kiszolgálását

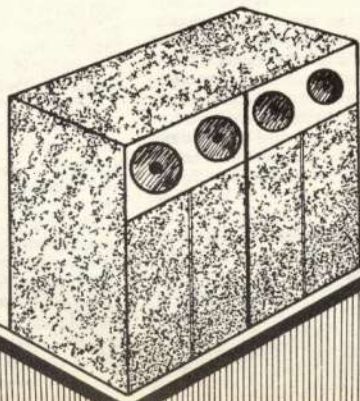
### **KERESKEDELMI FŐOSZTÁLYUNK**

Budapest III., Kerék u. 6. útján

### **FORGALMAZUNK:**

- robotron A 6402 típusú számítógépeket
- TZ 80 típ. professzionális számítógéppel vezérelt munkaidő rögzítő és nyilvántartó rendszereket,
- különféle személyi számítógépeket,
- különböző típusú számítógép printereket,
- számítógépek mágneses perifériáihoz szükséges adathordozókat: floppy diszkeket, mágnesszalagokat, mágneslemezeket,
- számítástechnikai berendezésekhez és eszközökhöz szükséges alkatrészeket.

**SZOLGÁLTATÁSAINKKAL ÉS ÁRUKÍNÁLATUNKKAL  
TISZTELT VEVŐINK RENDELKEZÉSÉRE ÁLLUNK!**



**ORSZÁGOS**

**SZERVIZHÁLÓZAT**



# Akik csatát veszíthetnek, de háborút nem

Clive Sinclair és Jack Tramiel szinte egy személyben meghatározták azt, hogy az egyszerű házi használatra tervezett személyi számítógépek néhány év alatt milliók veheték birtokukba. Rendkívüli vállalkozói és technikai felkészültségük révén rendkívüli üzleti sikereket értek el (lásd előző számunkat). A korszerű technológiák bölcsőjében, a kaliforniai Szilícium völgyben is akadt azonban egy fiatalember, aki a sikerek tekintetében bátran felveheti velük a versenyt.

## Steve Jobs

Ha a maga 62 évével Jack Tramiel a nagy vállalkozói egyéniség doyenjének tekinthető, Steve Jobs alighanem a társaság benjáminja. A jelenleg 30 éves fiatalember az Apple egyik alapítója. Újszerű termékfilozófiájával és kiváló piaci érzékével legalább annyira hozzájárult az Apple II sikeréhez, mint a gép hacker beállítottságú főkonstruktöre, Stephen Wozniak.

A cég image-ének kialakításában és belső munkahelyi stílusának formálásában is döntő szerepe volt. Az új technikai és üzleti kor sikerifjúságának, az ún. yuppie-knak egyik vezető egyénisége Jobs. „Csak mi érthetjük igazán ezt a mai világot, és csak mi tudjuk igazán olyanná formálni, hogy az mindannyiunknak jó legyen” – ez az elképzelt kijelentés sugárzik szinte minden tettéből és lépéséből.

Egy újságrónak így magyarázta az új korszak generációs valóságát: „Körülöttünk mindig is ott voltak a számítógépek. Ez különböztet meg bennünket tőletek, hapsikám. Ti B. C. – Before Computers – születettetek.” (Szójták a Before Christ – Krisztus előtt kifejezésre célozva).

Az új konstrukciójú mikroszámítógépek az ifjúság széles tömegeit hódították meg. Jobs pedig egyfajta példaképe lett a számítógép-bűvölő fiataloknak. „Mindéz a hely miatt van így.” A kaliforniai

Szilícium Völgyről van szó, ahol évekkel, sőt majdnem egy évtizeddel előbb jelentkezhetett a számítógépes korszak. „En itt születtem. Amikor 14 éves voltam, hírneves számítógépes mérnököket faggattam. Az Apple a mikroprocesszor szülője, a mikroprocesszort pedig mindössze öt mérföldnyire innen, ugyanebben a völgyben alkották meg.”

A ma kb. másfél milliárd dolláros éves forgalommal rendelkező cég vezetésében mindig is meghatározó szerepet töltött be. Legutóbb az igazgató tanács elnöke és a Macintosh részleg irányítója volt.

A Macintosh-t szinte saját gyermekként kezelte. Vesszőparipája volt egy olyan személyi számítógép konstruálása, amely mentes a felesleges technikai „cicomáktól” és az elsieltett fejlesztő munka sallangjaitól. Az igazi, a végső személyszámítógép-konstruktórt akarta. Olyat, ami mindent tud, mégis kecses és kompakt darab. Olyan gépre gondolt, amibe sem kártyát nem kell dugdosni, sem pedig más egységekkel nem kell összemadzagolni. Olyan szoftvertámogatást álmodott a géphez, melynek segítségével órák alatt a legbo-

nyolultabb dolgokat is bárki elcsajáthatja.

A sikerifjúság legragygóbb hackerjeit gyűjtötte fejlesztő csapatába, akik nem ismertek lehetetlen. Eszevesztett tempót diktált, a heti 70 órás munkahét volt a minimum. A legapróbb technikai és ergonomiai kérdésben sem volt hajlandó kompromisszumot kötni. Szinte anekdotaszzerű történet, hogy addig nem hagyott békét a fejlesztőknek, amíg azok pontosan abba a szögbe nem állították a Mac homlokfelületét, amilyenbe azt ő megálmodta.

Az alkalmazási szoftverrel való ellátás érdekében mozgósította a sikerifjúság független szoftvergyártó cégeit. A Bill Gates vezette Microsoft csapat, a Mitch Kapor irányította Lotus team és Fred Gibbons Software Publishing csoportja csak a legjelentősebbek a csatasorba állók közül. A tét nem kisebb dolog volt, mint elkerülni az IBM PC minden mást elnyomó, abszolút konstrukciós monopóliumát. „Azt szeretnénk, hogy a Macintosh az ipar harmadik mérföldköve legyen” – nyilatkozta a gép piacra hozatalakor Jobs.

A szuperalacsony árú gyártás egy technikai bővözmatványának is beillő, mindössze 21 millió dolláros költséggel megépített automatizált gyárban indult be. A két kártyára leegyszerűsített elektronikai konstrukció miatt a gyártási költségeknek csupán 8 százalékát teszik ki a munkabérek-költségek. Ilyen arányok mellett nem túlságosan törekedtek az emberi közreműködést kizáró teljes automatizálásra, hanem inkább a nagy termelékenységű munkaszervezés és a kimagasló színvonalú gépészet integrálására. Itt is újat akartak alkotni. Még Japánban is elutasított, hogy tanuljanak mások tapasztalataiból és hibáiból.

Jobs álma valóra vált. A cég Apple III és Lisa kudarcai után megszületett az az Apple gép, amit ismét vezet, mint a cukrot. Még az 1985 eleje óta tartó drasztikus piaci visszaesés sem érte végzetesen a Mac-et. Amíg az IBM-nek saját személyi számítógépeiből hatszáz ezer és egymillió darab közötti raktárkészlete volt, addig a Macintosh-ból még mindig 20-35 ezer darabot tudott havonta értékesíteni az Apple. 1984-ben pedig 280 ezret adtak el belőle.

Pedig nem is tudta minden elképzelését keresztülvinni Jobs. Kétezer dollárnál semmiképpen sem drágábban akarta piacra hozni az új csodát. Az igazgatói tanácsban azonban leszavazták, és 500 dollárral magasabb kezdő árat állapítottak meg. Ezzel megkezdődött Jobs életének is egy új szakasza. Sikerei csúcán fosztották meg őt a még nagyobb sikerek lehetőségétől.

Ugyanazok az erők kezdték korlátozni az ambíciózus fiatalember mozgási szabadságát, mint amelyek már korábban is számtalan akadályt gördítettek elé a cég dolgoiba való „én mindenkinél jobban tudom, hogy mi a jó” érzetet keltő, agresszív beleszólásai miatt.

Jobsot annak idején nem engedték beleszólni a Lisa fejlesztésébe. Meghagyták neki a Macintosh projektet, „Jobs bolondozását”, ahogy maguk között nevezték az ott folyó dolgokat. A Lisa kudarcra és a Macintosh sikere azonban gyökeresen megváltoztatta a hely-



**Give me an Apple and a mouse!**  
Hevényezett fordításban: Kérek egy almát és egy egeret!  
(By the courtesy of  $\mu$  Mikroszámítógép Magazin)



zetet. 1984 februárjában Jobs elérte az Apple átszervezését. Megőrizve az igazgató tanács elnöki posztját, az Apple II-vel egyenrangú, új Macintosh részleg vezetője lett. A Lisa-ügyek is ehhez a részleghez kerültek. Saját marketing szervezete és saját pénzügyi keretei voltak. Jobs megalapította saját cégét az Apple-n belül!

De nem volt szerencséje. Az 1984 decemberében először a házi-számítógép-piacon bekövetkezett drasztikus visszaesés hamarosan átterjedt a személyi számítógépek területére is. Az Apple-bevételek növekedtek ugyan, de nem a kívánt mértékben. Jobsnak ráadásul kimeríthetetlen étvágya volt. A végtelenen leegyszerűsített Macintosh mellé egy igazi, minden korlát nélküli és szuperteljesítményű, új Mac-et kívánt felsorakoztatni.

Kihasználva azt, hogy mint az igazgató tanács elnökének joga, sőt kötelessége volt beszámolotni a cég Pepsi Colá-tól átszábitott ügyvezető elnökét, John Sculleyt, nyomást gyakorolt rá, hogy az Apple II részlegnek szánt pénzeszközök jelentős ré-

szét is a Macintosh részleg kárpótja meg. Ezt az arcátlanúságot már az Apple II-esek technikai és marketing krémje sem volt hajlandó elviselni. Steve Wozniakkal az élen, 1985 márciusában csoportosan léptek ki a cégtől. Wozniak még óriási részvényvagyonát is kiárusította, nem kis problémát okozva az Apple részvények tőzsdei helyzetének.

Az ügyvezető elnöknek lépnie kellett. Titokban megdolgozta az igazgató tanács megnyerhető tagjait, majd május végén egy pénteki napon megvárta, amíg Jobs eltávozott, és azután rendkívüli igazgató tanácsot hivatott össze. Az eredmény nem maradt el. Jobst elmozdították valamennyi tisztségéből, és az akár gúnyolódásnak is tekinthető „terméklátnok”-i posztra helyezték, majd közölték vele, hogy senkinek sem köteles beszámolni munkájáról. A részvények 11 százalékát birtokló Jobs nem titkolta visszavágási szándékát. „A csata elvesztett, de a háború még nem fejeződött be” – közölte az érdekeltekkel.

NINO  
(Folytatjuk)

## ELEMZÉS

# A technológiák szerepe

**Sorozatunk első részében (µM, 1985. 3. szám) a mikroszámítógépes lavina első hullámának, a mikroprocesszoros technika tömeges megjelenésével, a mikrogépes hobbikorszak szerepével és a szabványos alapszoftverekkel fémjelzett korszakát tekintettük át. A második részben (µM, 1985. 4. szám) az első igazi személyi számítógép, az Apple II megjelenésével kapcsolatos technológiai tényezőkkel foglalkoztunk. Bemutattuk a képernyős szövefeldolgozást és számtáblázat-kezelést, mint a második hullám alkalmazási szempontú lavinaerősítő faktora. A sorozat harmadik részében (µM, 1985. 5. szám) az Apple II-vel egy időben jelentkező és nagysorozatú gyártóbázissal induló Tandy/Radio-Shack, valamint Commodore gépeinek robbanásszerű elterjedését elemeztük. Ennek kapcsán részletesen bemutattuk azokat a kereskedelmi technológiákat, amelyek a tömeges értékesítést elősegítették. Röviden elemeztük a CP/M alapú személyi számítógép-konstruktó mindössze néhány éves sikersorozatát a 80-as évek elején. Ezzel minden vonatkozásban megértett a helyzetet a marketinghez mindenkinél jobban értő IBM piacra lépéséhez.**

### Az IBM PC születése

John Opel 1980 márciusában vette át az IBM ügyvezetői posztját. Elődje betöltötte a 60. életévét, és a bevezetett gyakorlatnak megfelelően át kellett adnia a kormányzást. A cég jó pozícióban kezdte az új évtizedet. A 70-es években sikeresen visszaverte a miniszámítógép-gyártók támadásait, jelentősen növelte az eladásait a számítógéppontokba kerülő „nagyszámítógépek” (mainframe) piacán, és pénzügyi helyzete is kiváló volt. Stratégiai elemzői mégis gyors változtatás szükségességéről számoltak be az új felső vezetésknek.

Előrejelzéssel foglalkozó munkatársai az IBM-nek addig legnagyobb vállalkozói kihívást jelentő Amdahl cég 1981-re kivetített növekedését összevetették az újabb vállalkozók hasonló adataival, az IBM által uralt vállalati, gazdálkodó szervezeti számítógépesítést figyelembe véve. Úgy találták, hogy az ebben az irányban növekvő Apple 1981 negyedik negyedévi forgalma és nyeresége meghaladhatja az Amdahlét. Egy zakirár-

nú elemző testület azt is jelentette, hogy középtávon, az intelligens, azaz számítógéppel ellátott munkaadások fel fogják váltani az addigi, ún. buta terminálokat a nagyszámítógépes interaktív rendszerekben. Ráadásul ezek nem hogy csökkentenék a nagyszámítógépek iránti igényt, hanem még inkább növelni fogják azt.

A felső vezetők utasították őt főbb részlegüket, hogy álljanak elő javaslatokkal. Kaptak két Z80 alapú javaslatot, amelyek már élő IBM termékconcepcióra épültek, egy ilyen szempontból hasonló, az IBM Displawriter nevű szövefeldolgozó rendszerére építkező ajánlatot, egy egyetemi know-how-ra támaszkodó tervet és egy ötödiket, amely a lehető legszokatlanabb volt az összes közül. Ez minden addigi IBM termékconcepcióval szakítva azt ajánlotta, hogy a vállalkozóknál is vállalkozóibb módon, tisztán külső beszállítókra és kooperációs partnerekre támaszkodva, egy igazi személyi számítógéppel jelenjenek meg a piacon.

Ha a személyi számítógépekben kikristályosodott mikroszámítógépes lavina az IBM-től független

## „MAX-FORTH 85”

### KOMPLETT FEJLESZTŐ RENDSZER

# C 64 számítógépre

**Ára: 29 000,— Ft**

Megrendelhető:



## SZÁMÍTÁSTECHNIKAI MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET

1141 Budapest XIV., Kőszeg u. 4. Telefon: 831-805



technológiai tényezőkre támaszkodva egyre növekszik, akkor az IBM jobb, ha nem harcol ellene szokások, „belső” termékével, hanem inkább álljon a lavina élére – állították az utóbbi javaslattevők. Igaz, hogy az irodaautomatizálásba átrándulni szándékozó Exxon olajmopólium nehezen adná egy ilyen stratégiához a birtokában levő Zilog processzort, de itt van az Intel architektúrájában 16 bites, hardver buszrendszerében 8 bites új processzora, a 8088. Bárna lehet építkezni arra, a lavina ügyis abba az irányba veszi útját.

Az operációs rendszer sem gond. A Microsoft rövidesen készre igéri és potom áron ajánlja a horribilis kidolgozási árérték arányú CP/M 8086 változathoz igen hasonló MS-DOS rendszert. Lehet több rendszert is ajánlani a megfelelően kalkulált áron. Alkalmazási szoftverek tekintetében szükség van a VisiCalc-ra, egész olcsón kinnálnak egy könnyen adaptálható és egészen jó EasyScript nevű word processor programot; a Peachtree is vállalja szoftverei adaptálását, fordítóprogramokat pedig ugyancsak a Microsoft szállítana igen kedvező áron. Valahogy így szólhatott az érvelés, amikor kidolgozó lették azt a felső vezetés asztalára.

A gazdasági kockázat ebben a javaslatban volt a legkisebb, hiszen az IBM frissen elkészült, Boca Raton-i, ultramodern gyárában lehetett beindítani a szerelést. A floppyt a Tandontól, a nyomtatott az Epsontól beszerezve, a perifériális egységek gyártásába sem kellett házon belül beruházni. Az IBM félvezető gyárait sem kellett átállítani egy új területre. Más hardverelemekről is hasonló módon lehetett gondoskodni, ha az IBM-nek éppen nem volt. Végül az operációs rendszert és a moduláris hardverbővítési rendszert (á la Apple) nyíltta téve, sőt kifejezetten publikálva, a hardver/szoftver kiégésztők egyre növekvő mennyiségében lehetett reménykedni – ismét külső forrásból.

A látszólag igazi kockázat abban jelentkezett, hogy az IBM fennállása óta először igen nagy mértékben kiszolgáltatotta magát egy igen kiterjedt és növekedésével egyre nagyobb felügyeleti feladatot jelentő külső kapcsolati rendszernek. Vonatkozott ez még az értékesítési csatornákra is, hiszen a javaslat az IBM-től független kiskereskedelmi hálózatra kívánta helyezni az első, döntő időszakban a hangsúlyt. Gondoskodnia kellett arról, hogy megbízható, minőségi

szállított hírvérvén ne essen cserbe a számára ennyire új körülmények között sem.

Az IBM felső vezetésének mindenkori, legújabban híres intelligenciájára vall, hogy az új vezető is a lehető legjobb döntésre jutottak az értékelésnél, és zöld utat adtak a Philip Estridge vezette, utóbbi javaslattevőknek.

## Információs csúcstechnológia

Az IBM PC fogadtatását jól fejezi ki a Byte magazin 1981. októberi cikkének zárógondolata (írja Phil Lemmons): „Számunkra, akik nem szeretjük az óriásokat, az IBM Personal Computer megjeleneése olyan, mint a sokk. Úgy véltém, hogy az óriás bakot fog lőni akár úgy, hogy túl-, akár úgy, hogy alábecsüli a közönség által elvárt igényeket, és konokul ragaszkodni fog a mikroszámítógépes világ másik részével való inkompatibilitáshoz. Az IBM azonban egyáltalán nem lőtt bakot, hanem egyszerűen mérőfeldekkel a versenytársak elé ugrott.”

A követéző alapkonfigurációjú PC került a piacra: 16 kb-ot RAM kapacitással központi egység, billentyűzettel 1265 dollár, 48 kb-ot RAM kapacitású központi egység, billentyűzettel, egy floppy-meghajtóval és meghajtó adapterrel 2235 dollár.

Egy nyomtatóval, színes grafikai adapterrel és RAM-mal 64 kb-ot kapacitással kiegészített rendszer 3930 dollárba került. Egy további, akkor még csak 160 kb-ot meghajtó ára 570 dollár volt. A rendszer 40 kb-ot nyújt (csak olvasható) ROM tárában egy komplett BASIC interpreter is helyet kapott. A 80 szószög szövegmegjelenítés mellett 640 × 200 képpont felbontású színes grafikával is rendelkezett. Őt bővítési kártyahelet terveztek be a rendszerbe. Végül az IBM-től addig szokatlan stílusban Philip Estridge bejelentette: „Az IBM minden információt megad a meglevő háziiparnak ahhoz, hogy kártyákat tudjanak tervezni. Nyitottak vagyunk minden szoftverajánlat tekintetében.”

Összehasonlítással érdemes felsorolni az Apple akkori árait (ebben viszont már benne van a grafika): 16 kb-ot Apple II, beépített billentyűzettel 1029 dollár, 48 k-s változat 1079 dollár, 48 k-s változat floppy-meghajtóval és az operációs rendszerrel 1748 dollár.

Az IBM PC tehát igencsak versenyképesen került forgalomba. Különösen igaz ez, ha jobb para-

méterekkel rendelkező grafikáját és nagyobb tárkapacitással kiépíthetőségét vesszük figyelembe. Az alkalmazott 16 kb-ites csipek miatt akkoriban ez utóbbi még csak 256 kb-ot volt, de nem kellett sokáig várni, hogy az elvi maximum 640 kb-otja emelkedjék. A gép esztétikusságban, irodai környezetbe jobban illő volt, a floppy-meghajtókat a központi egységbe építve tartalmazta, és billentyűzet szabad, mozgatható módon csatlakozott a géphez. Az IBM konstrukció ugyan nem volt villamosmérnöki csúcsteljesítmény, de igen alaposan megtervezett és igen gondosan kivitelezett darabnak bizonyult. Talán még fontosabb, hogy nagy tömegben is olcsón lehetett gyártani. A többi, ahogy mondani szokás, már történelem.

Hol is van itt az újabb technológiai tényező, kérdezhetné bárki. Magának az IBM teljesítménynek az egészében. Az IBM ugyanis, mint az információs technológiák nem kevésbé szállítója, hanem felhasználója, azon belül is élenjáró felhasználója, tudott a mikroszámítógépes lavina élvonalába kerülni, és sajátosn tovább növelni a jelenség lavinaszerű voltát.

„Azok a vállalatok viszik a leg-többre a 80-as évekből, amelyek a legjobban képesek gazdálkodni a rendelkezésükre álló információval” – nyilatkozta 1981 közepén Opel. Két év múlva a részvényesek előtt tartott beszédében arról számolt be, hogy a 365 ezres IBM csapatból minden harmadik munkatársnak képernyős megjelenítő segíti napi munkáját. Az utóbbi persze csak felszínes mutatószám. A tartalom itt a döntő, aminek szempontjából a számítástechnika csak elengedhetetlen segédeszköz.

A tartalom az IBM PC születésével kapcsolatos események is kiválóan illusztrálják. A stratégiai elemzésekhez és előrejelzésekhez nagy mennyiségű és a lehető leg-gondosabban ellenőrzött adata volt szükség. Meg kellett határozni az ebből előállítandó elemzési információit, és ennek alapján elvégezni a feldolgozásokat.

Az öt részlegnek külön-külön, célirányos, de sokkal inkább részletekbe menő, számtalan belső és külső körülményt figyelembe vevő saját elemzést kellett végeznie. Részletesen modellezniük kellett a javasolt termékkel kapcsolatos teljes gyártási, értékesítési és támogatási szituációt, amellet, hogy ki kellett fejleszteniük a prototípust. Végül piaci prognózist kellett készíteniük, amelyhez az ellenfelek piaci helyzetének várható alakulá-

sát is fel kellett mérni. Mindez rendkívül gyorsan kellett elvégezni.

Az IBM-nél alkalmazott információcsúcstechnológia szorosan beépült a cég vállalatgazdálkodási és piaci pozíciónyerési technológiájába, ha szabad egyáltalán ezt az utóbbi a technológia szóval illetnünk. A gazdálkodási technológia a vállalati szervezet gazdasági eredményességét meghatározó eljárások és módszerek összessége. Az alapvető vállalati erőforrások mint a termelés, a pénzügyek, a kutatás-fejlesztés, a kiharcolt piaci pozíció (ide sorolandó a kereskedelem, a marketing stb.) és az alkalmazott munkaerő legjobb kihasználására irányulnak a mindenkori eljárások és módszerek.

A napi operatív ügyvitel rendszere, valamint a számítógépes feldolgozások rutinszerű, ún. adatfeldolgozási rendszere csak alap, amire az egész épület felépül. Az információs csúcstechnológia kulcsszerepe abban van, hogy öriási mértékben megnöveli a gazdálkodás hatékonyságát és átlátóképességét, és szükség esetén még az IBM méretű szervezetet is alkalmazható tesz valódi vállalkozói magatartásra.

## Winchester-lavina

Az IBM személyi számítógépes vállalkozása a mikroszámítógépes lavina tudatos kiaknázására épült. A sors sajátos fordulata, hogy egy olyan technológiai tényező juttatta további növekedéshez az IBM PC lavinát, ami többszörösen is az IBM-nél gyökeredzik. Az első mágneslemezt az IBM fejlesztette ki 1956-ban, RAMAC néven. Nem kis részben az IBM alapolakri indulva, a mágneslemezek gyártására időközben külön iparág alakult ki. A hajlékony mágneslemezes tárolókat is először az IBM fejlesztette ki, majd új vállalkozások egymáshoz kapcsolódó láncolata juttatta ahhoz a szerephez őket, amit a személyi számítógépes technika piaci sikerében már az első időszakban betöltöttek.

Az ún. Winchester-lemezek alaptechnológiájának kifejlesztésében is az IBM volt az első. 1973-ban hozta piacra az első ilyen meghajtót, ami még 14 hüvelykes tárcsát használt. Ezt az alaptechnológiát csípték fel az olcsó és gyors lemezek gyártásában fantáziát látó új vállalkozók, a tárcsaméretet egyre csökkentek, a viszonylagos kapacitást egyre nött, az árák folyamatosan estek, és ezzel együtt a sorozatnagyságok ugrásszerűen emelkedtek.



A Winchester-meghajtó, a hagyományos lemezekhez hasonlóan, merev tárcsát használ, és mozgólvasó/író feje van. Sajátossága viszont az, hogy mindez lég- és portmentesen zárt házban helyezkedik el. Ennek következtében a fejek nagyságrendekkel alacsonyabb távolságban repülhetnek a forgó lemezfelület felett, így nagyobb írási sebőséget és elvben gyorsabb hozzáférési időket lehet elérni a meghajtóknál. A forgási sebesség például tízszerese egy floppy forgási sebességének. Hátránya viszont, hogy a lemezt nem lehet cserélni.

Az eredeti IBM Winchester elnevezés még abból adódott, hogy a Vadnyugat híres fegyveréhez hasonlóan, 30+30 (adott esetben Mbájtos) tárkapacitásra (kettős kivétel lemezezésre) tervezték az első ilyen tárolót.

1979-ben jelent meg a 8 hüvelykes tárcsát alkalmazó meghajtó, majd egy évre rá az 5 1/4 hüvelykes egység. A Seagate Technology (egyik alapítója Alan Shugart) formátumát még vagy 45 más gyártó is átvette rövid időn belül, mivel jól illeszkedett a mikroszámítógépek kialakulóban levő készülékmeretéhez. A kezdeti 5 Mbájtos tárkapacitású egységeket igen gyorsan kóvtették a 10 Mbájtos, sőt még ennél is nagyobb kapacitású meghajtók. 1983-ban ádáz piaci versenyt kezdeményezett a Tandem, és a személyi számítógépekhez árban és teljesítményben egyaránt igen jól illeszkedő, 10 Mbájtos tárolók nagyban OEM árat 400 dollár alá szorította. A nagy teljesítményű mikroszámítógépek vállalati-üzleti alkalmazása szinte korlátlan felvevőpiacot jelentett. Így már 1983-ban nem kevesebb, mint 1,2 millió Winchester-meghajtót tudtak értékesíteni a gyártók.

Az IBM 1983 márciusában jelentette be a PC XT (eXTended) változatot, ami gyakorlatilag csak a 10 Mbájtos Winchester-lemez bevezetésével különbözött az előzőtől. Alapára 4995 dollár. Ezzel élt véget egyidejűleg a piaci próbálkozás időszaka – addig „mindössze” 200 ezer darabot értékesítettek – és a személyi számítógép az IBM központi vállalati terméké lett. Az igen magas színvonalon automatizált (összeszerelés 10 perc alatt) Boca Raton gyár hozzákezdett a gépek milliós sorozatnagyságú előállításához. Az így beindult IBM gözhenget akadálytalanul haladhatott előre, mivel a gyors és nagy memória-kiéppíthetőségű központi egységhez hasonló háttértár-lehetőséget biztosított az

útszen erre az időpontra beérett Winchester-laviná.

## Az alkalmazásfejlesztés BASIC-je

A nagy teljesítményű kihasználásához olyan szoftver kell, amely egyszerű alkalmas a nagyméretű adat- és információfeldolgozó alkalmazások fejlesztésére, másrészt pedig jóval hatékonyabb az addig alkalmazott alkalmazásfejlesztési technológiáknál, mind a kezdeti kidolgozás, mind a folyamatosan változó igények szerinti karbantartás szempontjából. A mikroszámítógépes lava belső öntörvényűségét semmi sem mutatja jobban, mint az, hogy mire az igazán nagy teljesítményű személyi számítógépek áradata megindult, addigra már kapható volt ilyen szoftver a piacon.

A később dBase II-nek nevezett terméket egy Wayne Ratliff nevű úr álmodta meg, még 1979 nyarán. A szerző által Vulcannak keresztelt adatbázis-kezelő programra először a Byte magazin egyik állandó rovatának írója, Jerry Pournelle figyelte fel. Az Ashton-Tate szoftverkiadó vállalat egyik alapítója, George Tate a Byte-ból szerzett tudomást a figyelemreméltó új termékéről, és hamarosan forgalmazási megállapodást kötött a szerzővel.

A CP/M környezetre készült, eredeti változat 1981 januárjában került piacra, a mikro- és személyi számítógépes szoftverek között igen magasnak számító, 700 dolláros áron. A CP/M-re jellemző 8080 assembly kód és az MS (PC) DOS-ra jellemző 8086 assembly kód közötti hasonlóság lehetővé tette, hogy az IBM változat már 1982 második felében megjelenjen.

A termék tökéletesedésének fontos állomását jelentette az 1984 júniusában piacra dobott dBase III. Ez az eredeti dBase filozófiát teljesen újragondolta. Tulajdonképpen egy felülről kompatibilis, de szolgáltatásaiban még teljesebb, új adatbázis-kezelő rendszert vezetett be. Erről már bátran elmondható, hogy igazi, igen magas szintű programozási nyelvnek is tekinthető, mivel az adat- és információfeldolgozás általános fogalmain alapuló, BASIC-szerű nyelvték mutatkozik az alkalmazásfejlesztők számára.

1985 márciusának végéig a dBase II-t 328 ezer példányban adták el, ennek kb. 30 százaléka CP/M változat. A becslések szerint ugyanakkor legalább 1 millió példányban használják szerte a világon. A dBase III-ből 185 ezer pél-

dány kelt el az első 9 hónap alatt. Az illegális másolatokról nincs becslés adat, de nyilvánvalóan ennek többszöröse. Az 1980-ban még mindössze 465 ezer dolláros éves forgalmat lebonyolító Ashton-Tate-ből időközben a mikrogépes szoftverpiac harmadik legnagyobb cége lett. Idei várható forgalma megközelíti a 130 millió dollárt.

A dBase nyelvben 3-4-szer olyan gyorsan lehet kifejlesztetni egy adat- vagy információfeldolgozási alkalmazást, mint bármelyik hagyományos programozási nyelvben. Ennek oka, hogy a nyelvi köntös igazi alkalmazásgenerátort takar. Bármilyen adat- vagy információfeldolgozási feladat közös lényegének a dBase filozófia egy ún. relációs adatbázist tekint.

Az ilyen adatbázisban véletlenül egyszerű adattáblázatok alkotják az alapot, amire minden ráépül. Az adattáblázatok minden sora azonos struktúrátú, „egyszerű” sor. A táblázat oszlopaiban minden sorban ugyanaz a fajta információ található. Bármelyik oszlop hozzáférési (lekérdezési, módosítási, törlési és beszúrás) kulcsként használható, vagyis a táblázatokhoz nem sorszám szerint, hanem tartalom szerint (asszociatív) férünk hozzá. Különböző táblázatokat a sorai azonos értelmű oszlopaiban előforduló értékek szerint, egy közös táblázatba lehet egyesíteni, illetve egy táblázatból adott szempont szerinti kivétellel résztáblázatot lehet képezni.

Az ily módon strukturálható adatbázis szerinti alkalmazáshoz nemcsak adatbázis-definíciók utasítások vannak a dBase nyelvben, hanem egy igazi alkalmazásgenerátorhoz méltóan az adatbevitel, adatkarbantartás, lekérdezés és nyomtatott tabló előállítás funkcióihoz is teljes körű nyelvi készlet áll rendelkezésre. Így gyakorlatilag bármilyen adat- és információfeldolgozási alkalmazást ki lehet fejleszteni dBase-ben, amit nyelvi hallatlan dőzszerűsége is igazol.

A dBase alkalmazások három kategóriájáról beszélhetünk. Az elsőbe olyan, munkakör-specifikus és egyéni izlés szerinti nyelvtárak sorolhatók, amelyeknél tulajdonképpen csak egy-két egyszerű listáról (táblázatról) van szó. A következő csoporthoz tartozó, ún. vertikális alkalmazások esetében az alkalmazási rendszer felhasználója nem dBase-t vásárol, hanem egy olyan, célirányos alkalmazási igényeket kielégítő szoftverterméket, amelyet dBase-ben írtak. Az Ashton-Tate megjelentette

a dBase vertikális alkalmazásainak katalógusát, amelyben több mint 170 termék található, egyenként 300 és 1500 dollár közötti árban. Mindegyik igen értékes adat- és információfeldolgozási tudást tesz megvásárolhatóvá az adott területen.

A dBase alkalmazások harmadik kategóriájához tartoznak a teljes értékű, integrált adatbázisrendszerek. Olyan egyedi fejlesztésű rendszerekről van szó, amelyek az adott alkalmazási környezetet valamilyen alkalmazási célfunkcióját egyetlen rendszerbe foglalják. Az egyedi feldolgozási logika az adott szervezet szempontjából mindig optimálisan alakított, és olyan szolgáltatások is biztosíthatók, amelyek az előző esetekben nem. Így például egy komplett szervezeti működés bármelyik információjára vonatkozóan, bármelyik időpillanatban megfogalmazhatók ad hoc igények szerinti lekérdezések. A dBase nyelv itt az alkalmazási rendszer elválaszthatatlan része.

A mikroszámítógépen használt relációs adatbázis-alkalmazások 60-70 százaléka támaszkodik ma a dBase nyelvre, így de facto szabvány lett. A nyelvi kialakítású alkalmazásgenerátorok piaca ugyan időközben bővült, ez azonban csak a dBase III tényleges kiskereskedelmi árának 400 dollár kölélébese esetében jelentkezett. A vállalati-professionális szférába kerülő szoftverek piacán a dBase III egész nyáron az első helyen állt, megelőzve nemcsak az adatbázis-konkurrenciát, hanem olyan múltbeli sikertermékeket is, mint a Lotus 1-2-3.

## Integrált döntéshozatal-támogatás

Az adat- és információfeldolgozás mellett természetesen a döntéshozatali munka megfelelő támogatására is szükség van. A VisiCalc által megnyitott, forradalmian új út egy teljes körű döntéshozatal-támogatáshoz azonban még nem volt elégséges. A döntéshozó ugyanis nemcsak strukturált számtáblázat formájában kívánja manipulálni a döntési adatokat, hanem a számokot és együttes grafikus megjelenítésére is szüksége van (kőr- és oszlopdiagramok stb.). Ezenkívül a döntési adatstruktúrák alapját képező kiindulási adatokat szeretné egyfajta döntési adatbázison tárolni, illetve a döntési számtáblázatokat egyszerű szövegfeldolgozási támogatással szövegesen kommentálni.

Ezt ismerte fel 1981-ben Mit-



chell Kapor, amikor a VisiCalc segítségével kapott eredményeket egy napon csak igen körülményesen és lassan tudta grafikusán megjeleníteni az általa fejlesztett Visi-Trend nevű program segítségével. Ezt és egy VisiPlot nevű másik terméket a Personal Software-nek fejlesztette ki. Mindkét termék az ún. DIF (Data Interchange Format) típusú speciális fájlok segítségével volt képes adatokat cserélni a Visi-Calccal, és ez az adatsere természetesen igen lassú volt. „Olyan integrált termékre van szükség, amely közös memória-adatstruktúrákat használva, bármelyik generikus alkalmazási funkció irányában pillanatok alatti lekérdezést biztosít” – vélekedett Kapor.

Korábbi szoftvertermékeit eladta a Personal Software-nek 1,5 millió dollárért, új elképzelésével pe-

A forgalmazás 1983 januárjában kezdődött meg, azelőtt sohasem látott méretű reklámkampány mellett. Ennek kezdeti költségeit már korábban igénybe vett vállalkozói tőkéből fedezték. Már az első negyedévében 90 ezer darabot adtak el, a másodikban pedig 160 ezret. 1983 októberében részvényeket bocsátottak ki; ezzel 34 millió dollár tőkéhez jutottak. 1983. évi forgalmuk 53 millió dollár volt, és ebből is jelentős nyereség képződött. Így már igazán tőkeerősen jelenthették be Symphony-nak nevezett új terméküket 1984 januárjában. A Symphony 1984 júliusában került piacra, és az értékesítés első 9 hónapja 40 millió dollár bevételt hozott. A Lotus 1-2-3-ból 1985 nyaráig 1 millió darabot adtak el, nem számítva az illegális másolatokat. A cég ez évi forgalma

nyarán Jem Manz, a Lotus Development ügyvezető elnöke. Hozzátehetjük, hogy a számítástechnika történetében is emlékezetes eseményt jelent a Lotus jelentkezése.

Az 1-2-3 elnevezés azt takarja, hogy három generikus funkciót sikerült egy interaktív nyelvbe egyesíteni: a VisiCalcnál lényegesen nagyobb méretű (256 x 2,048 cella), gyorsabb és funkcionálisan fejlettebb képernyős számtáblázatfeldolgozást, egy döntési típusú adatbázis-kezelést (az adattáblázat mindig is közel áll a számtáblázathoz), és üzleti grafikat. Ezzel olyan magas szintű alkalmazási generátort fejlesztettek ki, amellyel az interaktív, számítógépes döntési rendszerek gyakorlatilag bármelyikét VisiCalc-szerű egyszerűséggel ki lehetett fejleszteni, annak valamennyi aspektusában.

az elsőg előnyével induló Visi-Calccal csak közvetlenül utána következik a maga 15 százalékával, a többiek pedig messze lemaradva. Ismét egy de facto szabványról van szó, amit az is jelez, hogy 1985 második felében 100 dollár alatti dömping áron, egy sor számszázalékosan kompatibilis 1-2-3 utána jelent meg a piacon.

Az egyes generikus funkciókat még további lehetőségekkel kiterjesztő, valamint teljes körű szövegfeldolgozást és adatkommunikációt bevezető Symphony jelentőségét majd a jövő mutatja meg.

Az 1-2-3 a dBase és a Winchester-lemezek mellett azonban már bizonyította lavinárosító hatását, amint azt az 1983-84. évi helyzetéről készült elemzések megmutatták (lásd lapunk 1985/2. számának 22-25. oldalát).

The collage contains several distinct data visualization components:

- Table 1:**

	Jan	Feb	March	Q1 Tots
Sales	£2,500	£2,700	£2,916	£8,116
Cogs	£1,375	£1,445	£1,521	
- Table 2:**

	Apr	May	June	Q2
Sales	£3,149	£3,401	£3,673	£10,223
Cogs	£1,602	£1,690	£1,786	
Profit	£1,547	£1,711	£1,888	
- Text Block:**

4/4/83 P147  
ABREAST OF THE M  
GOOD 1ST-QUARTER PRO  
MAY FUEL FURTHER GAI  
AS THE STOCK MARKE  
MONTH ADVANCE SHOWS  
EXHAUSTION, SOME THIR  
QUARTER EARNINGS REPORTS W  
GIVE IT NEW SUPPORT.  
ANTICIPATION OF A PROFITS  
UPTURN HELPED BOOST THE DO
- Text Block:**

Last Name **Norton**  
First Name **Joseph**  
Address **7 Goldsmith Ave.**  
City **Brockton**  
County **Berks**  
Sex **M** DOB **15-MAY-39**  
Coverage **£1,500** Wait **30**  
Cola **N** Premium **£0.00**
- Text Block:**

Joseph Norton  
7 Goldsmith Ave.  
Brockton
- Text Block:**

Dear Joseph:

I am pleased to inform you of some very good disability policy. The Workingman's Insurance announced some rate changes that may affect you.

Your current disability insurance policy states:

Monthly Benefit **£1,500.00**  
Wait **30**
- Figure:** A pie chart and a bar chart. The pie chart is divided into four segments with different patterns. The bar chart has four bars of varying heights.

Ezzel véget ért a technológiák szerepéről szóló elemző sorozat. Céltudatossá az volt, hogy egyrészt történelmi elemzést adja azoknak a technológiai tényezőknek, amelyek a mikroszámítógépes lavina kibontakozását eredményes okozták, másrészt hogy a területi kezetek között lehetséges módon bemutassa az eszközök, jelenségek és folyamatok lényegi vonásait. Mindehhez kb. 500 oldalnyi közvetlen forrásanyag került felhasználásra, és ehhez járult még a vonatkozó termékek részletes megismerése.

Az elmúlt öt év során végzett folyamatos elemző munka eredménye, úgy vélem, egész másként mutatja be akár az elektronikai forradalmat, akár a korszerű informatikát, mint ahogyan azt az eddigi hazai és külföldi munkák tették. Az a szándék, hogy a dolgok mögött a vállalkozói személyiséget vagy szervezetet is megmutassam, remélhetőleg nem fedi el ezt a tényt az olyan olvasó előtt sem, aki esetleg az egészet meglehetősen publicisztikai ízűnek találja. A Mikroszámítógépes Magazinban való megjelenítés ezt nyilvánvalóan szélességgé tette. Az eredmények ilyen fórumon történő publikálásában az a cél vezetett, hogy minél szélesebb körben váljék ismertté a mikroszámítógépes lavina lényege. Hazánkban ugyanis is olyan helyzet állt elő, amikor már egy percig sem halasztható tovább, hogy az eddig hirdetett igékek kapcsolatban kimondjuk azt, amit a Császár új ruhája című, jól ismert mesében az a bizonyos kisfiú kimondott.

**Számtáblázatok, adatkommunikáció, üzleti grafika, szövegfeldolgozás és adatbázis – minden együtt van, mint a Symphony(á)-ban**

dig a szoftver témában igazi haternek mondható barátjához, Jonathan Sachshoz fordult, 1982 áprilisában a két vállalkozó megalapította a Lotus Development Corporation nevű céget. 1982 októberében jelentették be az IBM PC-re kidolgozott Lotus 1-2-3-at. Ez volt tulajdonképpen az első olyan termék, amely ténylegesen kiaknáza a 16 bites PC igazi architektúráis lehetőségeit.

elérheti a 250 millió dollárt. Ezzel a független szoftvergyártók között – beleértve a nem mikrogépes szoftvereket is – az első helyre kerül. „A termék időzítése tökéletes volt, amilyen tökéletes maga a termék. Az IBM éppen hogy beindult, és a gazdaság is hihetetlen fellendülési szakasz előtt állt. A világ története során még sohasem volt ilyen nagy pillanat egy termék elindításához” – nyilatkozta 1985

Jellemző, hogy még egyszerű adat- és információfeldolgozó rendszereket is előnyösebb 1-2-3-ban fejleszteni, mint például dBase-ben. A közös tárolási adatstruktúra azonban a képernyős számtáblázathoz közel álló szerkezet, ezért az 1-2-3 alapvető szolgáltatása a képernyős számtáblázat-feldolgozás (spreadsheet calculation). Az ilyen alkalmazások kerekén 65 százalékát uralja,



## Formulák ábrázolásmódjáról

**Ezévi 5. számunkban „Képletkezelő programok” címen megjelent cikkben tárgyaltuk, a teljesség igényét és a gyakorlati alkalmazások céljait figyelembevéve fontosnak tartom – röviden – kiegészíteni a következőkkel.**

1. Differenciálhányados-képzésre és határozatlan integrálásra (primitív-függvény meghatározásra) vagy határozott integrálásra képes program létezését állítani *túlzás*. Még függvények egy-egy jól meghatározott (nem túl széles!) körére vonatkozóan, csupán differenciálhányados-képzést végző eljárás pusztá létének bizonyítása sem mindig egyszerű, az ilyen programok megszerkesztése pedig általában nehéz feladat. Az integrálás feladata pedig legalább olyan nehéz mint a differenciálhányados-képzése. Volna olyan eljárás (gépi algoritmus) amellyel megállapítható lenne, hogy mikor fejezhető ki egy integrál zárt alakban? Nem tudunk róla. Ha tévednénk, kérjük a szerzőt és az olvasókat, hogy igazítsák ki tájékozatlanságunkat. Ilyen eljárás nélkül pedig „formulálás” integráló eljárás – véleményünk szerint – nem létezhet.

2. Infix (közbeirt operátoros) formáról (további magyarázat nélkül) csak *binér* (kétooperandusos) művelet esetében értelmes beszélni. Vannak azonban *unér* (egyoperandusos) műveletek is, pl. a gyökvonás, ahol a műveleti jel nyilvánvalóan nem állhat az egyetlen operandus „között”. *Ternér* (három operandusos) műveletek és háromnál több operandusos műveletek is vannak, amelyek nem minden esetben állíthatók elő unér vagy binér műveletekből. Ilyen esetekben a műveleti jelnek az operandusok jele közé (hova?) helyezéséhez külön megállapodásra is szükség van. Egy használható megoldás lehet a következő. Az operátor jeléből megállapítható, hogy az operátor hány operandusos (pl. A,B,C – megállapodás alapján – egyoperandusos, D,E,F kétoperandusos, s,i,t) a műveleti jel pedig mindig közvetlenül az első operandus után következik. Így pl. ha G háromoperandusú, akkor

Gxyz „prefix”,  
xGyz „infix”,  
xyzG „postfix”  
felírása ugyanannak a formulának.

3. A prefix formát balrairt operátoros vagy *baloperátoros*, a postfix formát jobbrairt operátoros vagy *jobboperátoros* formának is nevezik. Ezeket a neveket *kifejezőbbnek* tartjuk az előbbieknél.

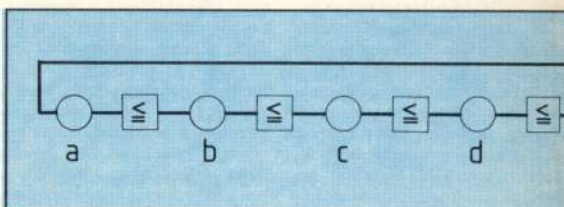
4. *Nem minden formula ábrázolható fával.* Az implicit formulának pl. egyike sem. Az explicitiek közül csak azok amelyekben minden, nem unér művelet kommutatív. (A fával ábrázolható formulákban ezért nem szerepelhet pl. az egyébként gyakori és nélkülözhetetlen hányadosképzés és hatványozás.) A fa megadásánál ugyanis *csak* azt kell megadni, hogy mely élnek van közös csúcsa, vagy azt, hogy mely csúcsok vannak közvetlenül összekötve. Ha egy ágból „kinővő” további ágak *sorrendje is fontos*, akkor ez a struktúra *több mint fa*. Lehet pl. „értékelt élő” fa, amelynek éleihez (adott feltételeknek elegendő) számok vannak rendelve.

A cikkben használt „fa” más szempontból sem csak fa, hanem *több annál*: van neki (egy) kitéüntetett csúcsa is.

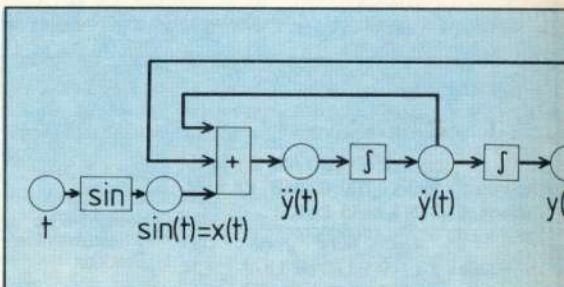
A fa tehát *több szempontból sem elegendő eszköz* (matematikai) formulák leírására.

5. Formulákban műveleti jelek mellett relációjeleket is használ a matematika és a számítástechnika is. A következő ábrázolási rendszer mindkettőre jól alkalmazható. A műveletnél az operandusokat és operátorokat kör vagy lekerekített csúcsú téglalap, az operátort pedig téglalap jelöli. A reláció esetében a relációban levők, az ún. *reláták* jele megegyezik az operandusok és operátumok jelével, a viszonyjelölő pedig az operátóréval (operációjelölővel). Példa az 1., 2. és 3. ábra. Az így kialakított (általában irányított vagy részben irányított) gráf csúcsai és élei minden esetben egyenesen értelmezhetők. Nem fordul elő az, hogy pl. él egyszer operátumnak, ill. operandusnak, máskor meg (ha pl. kiinduló operandushoz kapcsolódik) egyiknek sem felel meg.

6. Természetesen relációk ábrázolása sem valósítható meg mindig fával. Jó példa erre egyenlőségek gyakori bizonyítása amikor pl. a, b, c, d, egyenlőségét az



1. ábra. Az  $a \leq b \leq c \leq d \leq a$  reláció kapcsolatábrája



2. ábra. Az  $\dot{y}(t) = y(t) + y(t) + \sin(t)$  differenciálegyenlet kapcsolatábrája.  $\int$  az integrátor jele

$a \leq b \leq c \leq d \leq a$  egyenlőségét bizonyításával igazoljuk. Példánk esetében a formula gráfja nem fa, hanem *kör* (1. ábra).

7. Célszerű a terület fontos fogalmait (pl. kifejezés, képlet, formula, összefüggés) használat előtt pontosan definiálni, különösen akkor, ha azok értelmezése eltér a megszokottól.

8. Az „aktuális paraméter” a cikkben részfaként van definiálva ez – véleményem szerint – tévedés.

9. A kettőnél több operandus (ill. relátá) használatát – érdekes módon – nem a matematikai általánosságajhászás, hanem a gyakorlat kényszeríti ki, kényelmi és áttekinthetőségi okokból, még a kommutatív és asszociatív műveletek, pl. összeadás esetében is. Jó példát adnak erre a kapcsolatábrát használó rendszermodellezési nyelvek, ill. rendszerek, amelyekben nem ritka pl. a három bemenetű összegképző (2. ábra). Az ábrán az  $\dot{y}(t) = y(t) + y(t) + \sin(t)$  differenciálegyenlettel megadott kapcsolat differenciálhányadosképző nélküli kapcsolatábrája szerepel. (A differenciálhányadosképzők kerülését numerikus megfontolások indokolják.)

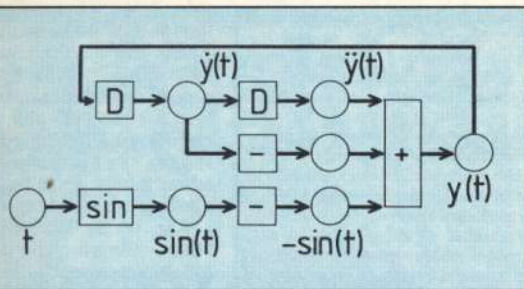
Nem kívánunk részletekbe bocsátkozni, de meg kell jegyeznünk, hogy a differenciálhányadosképzők felhasználásával megrajzolt kapcsolatára sem lesz fa. (3. ábra)

Nem látszik nehéz feladatnak fával nem leírható formulákra vonatkozó „helyettesítési szabályok” megalkotása sem. Ezt természetesen már nem tekinthetjük feladatunknak.

10. Formulák, kapcsolatok leírásának egy módja a formula kapcsolatábrájának elkészítése és leírása. E kapcsolatára természetesen nemcsak gráf, hiszen csúcsairól, éleiről más is kell tudniuk, mint azok illeszkedési viszonyait. Ezt egy példával szemléltetjük. A 3. ábrán szereplő, „formulához” tartozó kapcsolatára leírása a 4. ábra táblázatával történik. A táblázatban az érthetőség kedvéért használtuk a 3. ábra jelöléseit. A gyakorlatban a táblázat elemei másképp, pl. számokkal kódoltan vesznek részt a feldolgozási folyamatokban.

A táblázat legfeljebb 3 be-és legfeljebb 3 kimenetű operátorok leírására alkalmas. Esetünkben minden operátor egykimenetű volt. Ez azonban, noha kényelmes, de nem mindig teljesül. Általános





3. ábra. Az  $y(t) = \ddot{y}(t) - \dot{y}(t) - \sin(t)$  differenciálegyenlet kapcsolatábrája. D a differenciálhányadosképző jele

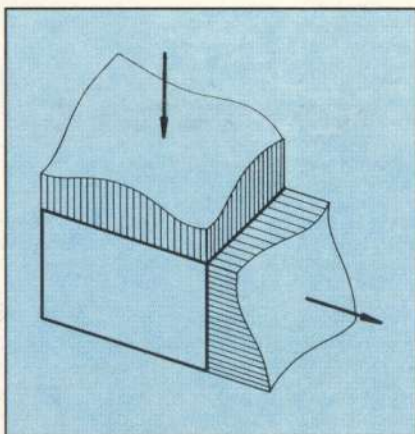
Operátor-típus	Első bemenet	Második bemenet	Harmadik bemenet	Első kimenet	Második kimenet	Harmadik kimenet
sin	t			x(t)		
+	y(t)	y(t)	x(t)	y-dot(t)		
∫	y-dot(t)			y(t)		
∫	y-dot(t)			y(t)		

4. ábra. A 2. ábra kapcsolatábrájának táblázatos leírása

esetben minden operátorra vonatkozóan be- és kimeneteinek (operandusainak és operátumainak) számát is meg kell adni valahogyan, és fel kell sorolni azokat helyes sorrendben. (Természetesen relációk esetében hasonló a helyzet.) Az operandusok és operátumok illetve a reláták számának közvetlen megadása mellőzhető, ha a helyes sorrendben felírva azokat, az operandusok sorozatát, az operátumok sorozatát és a reláták sorozatát speciális elválasztójelek különítik el a többi információtól.

11. Fontos felhívni a figyelmet arra, hogy a helyettesítési művelettel kialakított új formula nem lesz minden szempontból egyenértékű az előzővel. Nem mindig egyezik meg pl. a változók száma, a műveletek száma és maguk a műveletek sem stb. Sőt szó szerint maga az „egyenértékűség” is kérdéses lehet. Azaz nem mindig lesz egyenlő a helyettesítés előtti és utáni érték, ha a műveletek számológépi megvalósításait vesszük. Így az a furcsa helyzet áll elő, hogy a gépi formulakezelés *alkalmas egyes formulák gépi kezelésére, de nem alkalmas gépi műveletekre, relációkra vonatkozó formulák gépi kezelésére, mert ezek a formulák egyelőre nincsenek.* (A gépi összeadás és szorzás pl. nem asszociatív, és nem disztributív. Hogy milyen tulajdonságai vannak az egyes gépi műveleteknek és milyen kapcsolatok vannak az egyes gépi műveletek között, az olyan téma, amiről *jónnevelt számítástechnikus* nem szokott kérdezősködni.)

12. Vannak a gyakorlatban végtelen sok változós (operandusos, relátás) formulák is, amelyek – alkalmas módszerekkel, alkalmas jelöléssel – közelítőleg (esetleg pontosan is) kezelhetők formulamanipulációs programokkal. Pusztán érdekességként említünk ilyen operátorokat. Az egyik egy téglatest,

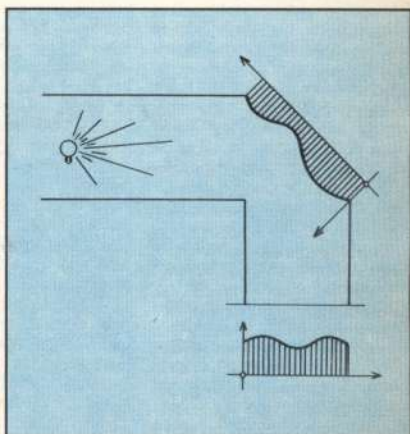


5. ábra. Egymástól függő hőmérsékleteloszlások (Végtelen sok változós rendszer)

amelynek egyik lapja időben változó, előírt hőmérsékleteloszlását ismerjük (bemeneti eloszlás). Az operátum-hőmérsékleteloszlás egy másik lapon tapasztalható hőmérsékleteloszlás lesz. A többi lapot tökéletesen szigeteltnek tekintjük (5. ábra).

A másik operátor operátuma egy megvilágítás-eloszlás, amely egy operandus felületről való visszaverődés következtében alakul ki (6. ábra).

13. Fontos tudatosítanunk, hogy egy formula grafja nemcsak akkor lehet végtelen, ha a változók (amiket a körök, lekerekített csúcsú téglalapok jelölnek) száma végtelen. Előfordulhat, hogy ezek száma véges, de közöttük végtelen sok reláció, művelet létesít kapcsolatot. Nyilvánvaló, hogy van végtelen sok változós formula amelyben a relációk és operátorok összszáma is végtelen. Érdekes, hogy sok esetben ezek kezelése – alkal-



6. ábra. Egymástól függő kétváltozós függvények. (Végtelen sok változós rendszer.) A felületek egy-egy metszetgörbéjét bevonalkáztuk

masan választott formulák közvetítésével –, véges formulamanipulációs eszközökkel is megoldható. Nem nehéz példát adni végtelen sok változós, végtelen sok relációs, végtelen sok operátoros rendszer véges formulákkal való leírására, ez azonban meghaladja hozzászólásunk terjedelmi korlátait.

14. A cikkben is és ebben a hozzászólásban is formulamanipuláció végig matematikai formulákkal való manipuláció értődik. Ez a lecsúszítás azonban *nem jogos*. Vannak más formulák is mint matematikaiak. A kémiai, fizikai, biológiai formulák, alakzatok, de a zenei, képzőművészeti, koreográfiai alakzatok, formulák is fontosak; még ha nincsenek is mind *matematikai* alakra hozva. Hogy ez a mai matematikai alaklétszlet milyen hosszú életű lesz, nem tudjuk. Azt sem tudjuk, hogy célszerű-e mindent a mai matematikai formaliz-

mus skémái szerint alakítani. Az azonban biztosra vehető, hogy a sok, ma még nem matematikai képlet, formula, alakzat manipulálásról tudomást nem venni a szakmai közvélemény rossz irányú manipulálása volna, amit mindenképpen el kell kerülnünk.

15. Meg kell különböztetni a (bal-, ill.) jobboperátoros írásmódot a zárójel nélküli írásmódtól. Ezek összekapcsolódása ugyanis nem szükségeszerű. Megfelelő értelmező szabályokkal jól használható pl. jobboperátoros és zárójeles, jobboperátoros és zárójel nélküli felírási módok definiálhatók. De infix zárójeles és infix zárójel nélküli felírási módokat is használhatunk megfelelő értelmezési szabályok mellett. (Nyilvánvaló, hogy a „balról jobbra” és a „jobból balra” szabályok mellett jól használható pl. a „szélekről befelé” szabály is.) Egy zárójel nélküli írásmód



Számítástechnikával 1969 óta foglalkozik. Kiemelt érdeklődési területe a számítógépek architektúrájának formális leírása és a személyi számítógépek oktatásának alkalmazása. Részt vett a hazai számítástechnikai képzés irányelveinek kidolgozásában, és tagja volt a személyi számítógépek iskolai alkalmazását javasoló munkabizottságnak. 1979-től elnöke a Számítástechnikai Terminológiai Bizottságnak, 1982-től társelnöke a Számítástechnikai Oktatási Bizottságnak.

Lukasiewicz lengyel matematikus munkái révén vált ismertté. E rendszer, ill. jobboперátoros változata pedig főleg Hewlett-Packard gépek révén került a számítástechnikai közudatba.

16. Egy matematikai formulának pl. egy aritmetikai „kifejezésnek” több különböző műveletvégzési sorrendű kiszámítása lehetséges. Egyes felírásai szabályos azonban, előfordulhat, hogy nem képesek minden (egyébként helyes eredményre vezető) kiszámítási sorrendet eredményező megvalósításra előállítására.

17. Nem tartom feleslegesnek megjegyezni, hogy a matematikában jelenleg a már említettek kivül az  $f(x)$  típusú baloperátoros  $(x)f$  jobboперátoros függvényjelölés, és az összes lehetséges indexes jelölés is használatban van. (A bal felső, a bal alsó, a jobb felső és a jobb alsó index mindegyikét használják.) Ezek által egy betű környezetében levő 8 hely közül 6 ki van használva. A maradék két hely sem marad azonban szabadon, a fölhúzás, a  $\wedge$ , a  $\rightarrow$ , az  $\sim$ , a  $\sim$  és a pont (idő szerinti differenciálhányados jelölésére) pl. a betű fölé szokott kerülni. A betű alatt legtöbbször csak az aláhúzás jelével találkozunk.

A hatványkitevő a klasszikus formuláirásnál jobb felső indexbe kerül. Ezért – tévesen – az operátor jobb felső indexbe írását kiteső írások is nevezik. (Ez az írásmód is elég gyakori.)

A formuláirásban tehát kevés a hely a tudomány számára. De nemcsak a hely kevés hanem a jel

is, a betű is. A többszörös indexek és az indexezett indexek is szükségesek és használjuk is őket. A szakember erre csak legyint, könnyű ezeket is egy sorba írni. Ez igaz, de ezáltal éppen az áttekinthetőség veszik el, ami pedig a formuláirás egyik fő célja. Várható tehát, hogy a formulamanipuláció a mainál több szempont figyelembevételével fog tovább fejlődni.

18. Végül pedig egy tipikus formulamanipulációs észrevétel. A cikk ellenálláshálózatának kapcsolási rajza (gráfja), amely szintén formulálható adható meg) és az eredő ellenállás formulaja érdekes gondolatokat sugall. Az eredő ellenállás formulaja „lánc törtszerű”. Igaz-e, hogy minden – egyszerűség kedvéért racionális értékű – ellenállás előállítható egységnyi ellenállásokból alkalmas kapcsolással? A válasz igen. Legyen az előállítandó ellenállás értéke  $m/n$  ( $m < n$ )! Ha  $n$  darab egységnyi ellenállást párhuzamosan kapcsolunk,  $1/n$  ellenállású rendszert kapunk. Ezt a rendszert  $m$  példányban előállítva és sorbakapcsolva  $m(1/n)$  azaz  $m/n$  lesz a kialakult rendszer eredő ellenállása. Ez a rendszer  $m$ -szer  $n$  darab egységnyi ellenállást tartalmaz. Megoldható-e a feladat kevesebb ellenállással is? (Eljutottunk az „optimalizálás” problémaköréhez, formulák különböző szempontból „optimális” alakú előállításának kérdéséhez.) A válasz most is igen! Fejtjük lánc törtsébe a racionális számot! Így pl. egy  $5/7$  értékű ellenálláshoz nem 35 darab,

hanem 5 darab egységnyi ellenállás elég. Nem tudjuk, hogy ez-e az ellenállások legkisebb száma.

Nem ellenállások előállításához, de más célra fontos az a kérdés, hogy milyen „építőelemeket” (példánkban ellenállásokat) célszerű rendszeresen amelyekből a legtöbb különböző érték előállítható, adott műveletek köztől vagy kötetlen alkalmazásaival.

Vessünk végül egy pillantást az eredő ellenállás egysorba írt törtjére, és hasonlítsuk össze ezt a hagyományos emeletet tört alakkal. Az első egyértelműen áttekinthetetlen, a második pedig arra is alkalmas, hogy egy-egy ellenállás változásának az eredőre gyakorolt hatását megbecsljük. Világos tehát, hogy az utóbbi alkalmasabb a kutatás és a gyakorlati munka számára mint az előbbi. Várjuk tehát azokat a formulamanipulációs eljárásokat is, amelyek formulák legáttekinthetőbb megjelenítéséről is gondoskodni tudnak.

POGÁNY CSABA

## FELHASZNÁLT IRODALOM

Könyves Tóth Kálmán: *Struktúra és rekurzió. Egyetemi Számítógépközpont Tájékoztató*, 15. szám, Budapest, 1975. június, 62. oldal.

Pogány Csaba: *A rendszermodellezésről. Függelék* Yaohan Chu *Folytonos rendszerek digitális szimulációja* című könyvéhez. *Műszaki Könyvkiadó, Budapest, 1977.*

Reader, Georg: *A Survey of Current Graphical Programming Techniques. Computer 18. kötet 8. szám, 1985. augusztus, 11. oldal.*

## PÁLYÁZATI FELHÍVÁS

Az NJSZT, a szekszárdi Garay János Gimnázium és a Mikroszámítógép Magazin szerkesztősége ismét pályázatot hirdet az alábbi kategóriákban:

1. Új, önálló játékprogram készítése HT-1080Z, ABC-80, Primo, Aircomp, ZX81, ZX-Spectrum, Commodore 64 gépre.

2. Valamely tantárgyhoz kapcsolódó, a tanítási órát segítő, illetve a tanulók önálló tanulását támogató oktatóprogram készítése a fent említett gépekhez, de itt előnyben részesülnek a HT-1080Z gépre írt programok.

A pályázaton részt vehet minden általános és középiskolás tanuló.

**A pályázat beadási határideje 1986. január 15.**

A programot BASIC nyelven kell készíteni (gépi kódú szubrutint tartalmazhat), és mágnesszalag-kazettán (a kazettára többször felvéve), rövid leírás kíséretében (mit tud, hogyan működik a játék, hanyadik osztály mely tantárgyának melyik anyagrészéhez kapcsolódik stb.), jelgelével ellátva (külön zárt borítékban a név, lakcím vagy

iskola) kérjük beküldeni a Mikroszámítógép Magazin szerkesztőségébe (Budapest, Fő u. 68. 1027).

A szerkesztőségnek joga van a pályázaton részt vevő programok közlésére, amiért a szokásos honoráriumot fizeti. A döntő 10-10 résztvevőjét az NJSZT tagjaiból, a Garay Gimnázium tanáraiból és a szerkesztőség munkatársaiból álló előzsűri választja ki.

A döntő – melyen az előzsűri által kiválasztott 10-10 program versenyez – Szekszárdon, a Garay János Gimnáziumban rendezendő Garay Napok alkalmából, 1986 márciusában lesz.

A döntőbe jutott tanulókat a Garay János Gimnázium vendégül látja.

Mindkét kategóriában az első három helyezett programot díjazuk, és mindkét kategóriában kiadjuk a közönség díját.

NJSZT Vámos Tibor elnök	Garay János Ált. Gimn. Zentai András igazgató	µMagazin Szerk. Kovács Győző a szerk. biz. vezetője
-------------------------------	--	--

– Pályafutása során a számítástechnika igen szétszertező területével foglalkozott: elméleti és gyakorlati kérdésekkel, előadásokat tartott, publikált. Különböző szerzett tapasztalatai bizonyára tovább szélesítették szakmai horizontját. Mindezek alapján miként ítéli meg a hazai mikro-számítástechnika helyzetét? A szakmában nem kevesen borulátóan nyilatkoznak erről. Egyértelmű, hogy inkább a „részasztin szemüvegek” táborához tartozik?

– Sajnos azt kell mondanom, hogy komoly léphátrányban vagyunk. Az elmaradás egyik alapvető oka, hogy más országokhoz képest sokkal később ismertük fel a téma fontosságát. Ha nem sikerül igen rövid időn belül legalább közepes szintre jutnunk, akkor iparunk fejlődésének, versenyképességének esélyei oly vékony mértékben csökkennek, hogy hamarosan az európai mezőny hátsó régiójába esünk vissza. Ma már a fejlődő világhoz tartozó országok is – mint Brazília vagy Dél-Korea – komoly vetélytársnak számítanak termékeink számára.

– Arra gondol, hogy egyre kevesebb árucikkfajta adható el mikroelektronikai tartalom nélkül?

– Arra is, hiszen ma már a legegyszerűbb háztartási gépek és készülékek sem képzelhetők el beágyazott mikroprocesszorok nélkül. Sőt a fejlesztésben a mérnöki tervezők munkáját



# Iparunk fejlődéséhez nélkülözhetetlen a mikroelektronika

## BESZÉLGETÉS SIMA DEZSŐVEL



kezdve, a termelésen át az értékesítésig egyre nélkülözhetetlenebb a számítógép. A számítástechnika ipari alkalmazásában nem lehet eléggé hangsúlyozni azt az alaptételt, hogy a számítástechnika, illetve a rend, a szervezethez rokonfogalmak: feltételezik, elősegítik, erősítik egymást. Rendkívül fontos tény, hogy a gazdasági infrastruktúrában a számítógépes információrendszerek elterjedése abszolút szükségszerű.

– Mit ért konkrétan ezen? Kérem, mondjon példát rá!

– Ha nálunk egy termelő vállalathoz rendelés fut be egy bizonyos árra, beindul egy folyamat: feladatok szerint lebontják a rendelést, felméri az anyagigényt, saját készletet stb.

Ha mindez manuálisan történik, akkor a folyamat túrheterogén, lassú, ráadásul nem is megbízható. Így nem lehet időben reagálni a piaci igényekre, nem lehet konkrétan és megbízhatóan árakat, szállítási határidőket meghatározni. Így nem léphetünk ki a siker reményével a piacra, mivel piaci versenytársaink már rendelkeznek modern számítógépes információrendszerrel, amellyel könnyen áttekinthetővé és irányíthatóvá teszik az egész folyamatot, a gyártáselőkészítéstől a piaci elhelyezésig.

A mikro-számítástechnika néhány év alatt fantasztikus teljesítménynövekedést mutatva hatolt az élet szinte valamennyi területére, tehát nagymérték-

ben társadalmiasult. Aki ezt nem ismeri fel idejében, és nem alkalmazkodik hozzá, az reménytelenül lemarad, elsüllyed. Ha tehát nem akarunk erre a sorsra jutni, „át kell térnünk” – mint annak idején István királynak népével együtt a kereszténységre.

– Úgy legyen. De mit és hogyan kellene tennünk, hogy a jámbor óhaj ne csupán elsőhajtott vágy maradjon?

– Az egyszerűsítés veszélye nélkül erre nehéz válaszolnom. Az már az eddig elmondottakból is kiderülhetett, hogy iparunk fejlődéséhez, egyáltalán szinten tartásához maradéktalanul ki kell aknázni a mikroelektronika lehetőségeit. És mivel az ipar gazdaságunk kulcsa, sokkal kiemeltebben kellene kezelni a mikroelektronikát is, kihasználva egyúttal az ipar központi irányításában rejlő lehetőségeket.

Így a rendelkezésre álló szűkös anyagi forrásokat hatékonyabban, operatívabban tudnánk felhasználni, és erőfeszítéseinket a legégetőbb feladatokra koncentrálnánk. Mivel az ipar egyik motorja hosszú távon a műszaki oktatás, erre nálunk is nagyobb hangsúlyt kellene helyezni és az eddiginél lényegesen nagyobb anyagi támogatásban kellene részesíteni. Kőkorszaki berendezésekkel csak kőkorszaki szakikat lehet képezni, a holnap állandó továbbfejlesztésre képes szakemberegárdáját aligha!

A költségvetésben nálunk az ipari és az oktatási tárca a múltban sajnálatos élességgel különült el egymástól. Az oktatási költségvetésből a műszaki felsőoktatásra nyomasztóan alacsony hányad jutott. Az ipar vezetői részéről az utóbbi időben egyre nagyobb készséget tapasztalunk arra, hogy e tartatlan állapotot változtassunk. Emögött már az a felismerés rejlik, hogy modern ipar és annak megfelelő szintű oktatás egymás nélkül nem létezik.

– A laikus közvéleményben mintha túlságosan kedvező kép formálódott volna ki a dolgok állásáról. Rengeteg személyi számítógép működik házakban, beindult az iskolaszámítógép-program, örvendetesen nagy érdeklődés fogadta a TV-BASIC-et, szakembereink sikerral állják meg helyüket külföldön, gombamódra szaporodnak és virágznak a számítógép-alkalmazási vállalkozások, és még folytathatnám a sort.

– Nem áll szándékomban a kedvező jelenségeket alábecsülni. Néhány Ön által említett tényezőről azonban más a véleményem. A rengeteg fajta mikrogep például gátolja a kompatibilis, egységes géppark kialakulását, következképpen az egymásra építhető szoftverállományok kialakulását. A meglévő jó szoftvereket pedig a rendkívül gyenge marketing-tevékenység miatt alig lehet külföldön eladni. Akármilyen kiváló egy szoftvertermék, semmit sem ér, ha nincs rá vevő. Azonkívül sok kiváló termék egyszerű fellobbanás. Úgy kell szoftvert csinálnunk, hogy már készülünk a következőre, sőt gondolkozunk az azután következőre is. A szoftver előállításának iparszerű folyamatnak kell lennie, iparszerű szervezethez, eszközökkel. Tudnunk kell „karbantartani”, állandóan továbbfejlesztetni, a piacra bevezetni és a piacon maradni. Csak olyan dologba szabad belekezdeni, amit hosszabb távon is képesek vagyunk csinálni, amit alaposan átgondoltunk, és erőt, kapacitásunkat számba véve meggyőződünk arról, hogy végig tudjuk vinni. Jó néhány csödbe jutott kisebb-nagyobb vállalkozás példája is erre int.

– Végül hadd kérdezsek egy, a sok közül számomra érdekes társadalmi funkciójára. Ön a Számítástechnikai Terminológiai Bizottság elnöke. Mit csinál tulajdonképpen ez a bizottság?

– Azon dolgozunk, hogy ennek az új tudományoknak olyan hazai nyelve alakuljon ki, amely mentes a szélsőségektől. Ne legyen tehát angol keveréknyelv, de ne is legyen mindenen túlmagyarosított. Szaknyelv is legyen, és illeszthető is legyen nyelvünk egészébe. Ezért dolgozunk ki a Számítástechnikai terminológiai szabványokat, amit a közeljövőben megjelenő Értelmező szótár egészít majd ki.

LACZKA MIKLÓS







## Diagram-szerkesztő

Az alábbi programot a Practical Computing ötlete alapján írtam át Spectrumra. Újdonsága az, hogy önműködően kiszámítja a bemenő adatok szélső értékeit, és a hisztogramot ezek között az értékek között ábrázolja.

```
455 FORI=1TOAV-1:IFMID$(A#,I,1)=" "THENAV=AV-1:NEXT:GOTO465
460 A4#=#A4#MID$(A#,I,1):NEXT
465 GOSUB535:J#=#A4#:GOSUB540:A4#=""#A1#=""#GOTO485
470 GOSUB545:FORI=1TO08:IFMID$(A#,I,1)=" "THENAV=AV-1:NEXT:GOTO480
475 T=08:NEXT
480 GOSUB535:J#=#RIGHT$(A#,AV-1):GOSUB540
485 IFAV>8THENA495
490 GOSUB535:FORI=AVTO8:J#=#A4#A4#A2#:GOSUB540:NEXT
495 IFB0<7AND((B1=B0ANDLEFT$(A#,1)=" ")OR(B1<>0ANDRIGHT$(A#,1)=" "))THENA1:=AV
+1:GOTO240
500 IFB6=0THENFORI=1TOVSTEP-1:POKEI,ASC(MID$(A#,I-X+B,1))AND63OR128:NEXT:GOTO19
505 FORI=1TOVSTEP-1:POKEI,ASC(MID$(A#,I-X+B,1))AND63:NEXT:GOTO195
510 B2=0:B3=0:IFB1<>7THENRETURN
515 FORI=1TO8:IFMID$(A#,I,1)="*"ORMID$(A#,I,1)="-"THENB2=7:NEXT:RETURN
520 IFMID$(A#,I,1)=","THENB3=7
525 NEXT:RETURN
526
527 REM:CURSOR POZICIOHALLO RUTIN
528
530 POKE214,L:POKE211,C:SYS58732:RETURN
531
535 T3=PEEK(51):T4=PEEK(52):RETURN
540 L$=L$+"":POKE820,PEEK(71):POKE821,PEEK(72):POKE(PEEK(820)+PEEK(821)*256),41
LEN(L$)
545 POKE51,T1:POKE52,T2:A#=""#J#:#L$=L$+"":POKE51,T3:POKE52,T4:J#=""#":RETURN
```

## A bevitt definiálására használt paraméterek

A program elején a rutin egyszerű meghívására van szükség, a paraméterek beállítására miatt.

- L - a bevitt mező elejének (képernyő) sorszáma  
 C - a bevitt mező elejének (képernyő) oszlopszáma  
 B - a bevitt mező hossza  
 B0 - üres mező engedélyezése  
 B1 - a bevitt mező típusa (alfanumerikus, numerikus stb.)  
 B6 - a bevitt mező megjelenítése: negatív, pozitív  
 B7 - üres mező kitöltése, vagy a kitöltött mezőn javítás  
 B8 - megszakítás engedélyezése  
 CO - szinkód

## A szerkesztő parancsok

- CRSR→ - egyvel jobbra pozicionál (a bevitt mező kitöltött részét van hatása)  
 CRSR← - egyvel balra pozicionál (a bevitt mező elejét van hatása)  
 HOME - a mező elejére pozicionál  
 CLR - törli a mezőt és az elejére áll  
 INST - egy karakter beszurására a kurzor helyére (csak akkor működik, ha a mező végén van még üres hely)  
 DEL - egy karakter törlése a kurzor helyén

## Kilépés a rutinból

- RE- - bármely pozíción leütve a bevitt végét jelenti, AS tartalmazza a beadott sztringet, A pedig AS numerikus értékét (a BASIC konvenció szerinti)  
 CRSR↑ - a bevitt mező első karaktereként leütve a rutin megszakad, az előző tartalomtól függetlenül AS=CHRS(14) és A=0 keletkezik. A programozónak különös gondot kell fordítania a megszakítási funkció leállítására, ha már nincs rá szükség. Ennek elmulasztására a képernyőn kisbetűre váltással figyelmet a rutin. A parancs engedélyezése: B8=7.

## A rutin kezelése

Üres mező kitöltése. A megadott pozíción a kért hosszúságú üres negatív mező elején várakozik. A beadást szintaktikailag ellenőrzi, és csak helyesen beírt adatot fogad el.

A bevitt karaktertorsor végén a CRSR←, HOME, CLR, RETURN billentyűket érti (itt nincs értelmezve a DEL, a CRSR→ és az INST parancs). A kitöltött szövegrészen belül az összes parancs el.

Már létező mező módosítása. Ez a mód csak szintaktikailag helyes mezőre alkalmazható (például arra, amit ezzel a rutinnal készítettünk), ui. az így kapott karaktertorsor helyesnek értelmezi. A kapott mező tartalmát negatíva váltja, majd a

mező elején várakozik, lehetővé téve a javítást (a "B" paraméter megadása nem szükséges).

Dátum megadása. Csak helyes dátumot fogad el, a szökőévet, hónapot és napot is figyeli. Hiba esetén azt ellenőrzi, hogy az a hónapban vagy a napban volt-e, és ennek megfelelően a 3. vagy 5. karakterre áll, újabb bevittet várva.

## A paraméterek értelmezése

- L - a bevitt mező kezdetének képernyő-sorszáma 0-24-ig  
 C - a bevitt mező kezdetének képernyő-oszlopszáma 0-39-ig  
 B - a bevitt mező hossza (max. 40)  
 B=0 esetén kurzorvillogással RETURN-t vár  
 B1 - B=0 - alfanumerikus bevittet  
 B1=7 - numerikus bevittet mező (előjel és tizedespon lehetőség)  
 B1=9 - numerikus bevittet mező (előjel és tizedespon nem lehet)  
 B1=48 - dátumbevittet mező (6 karakteres, EEHHNN formátumú)  
 B6 - B6=0 - CR után a mező negatív marad  
 B6=7 - CR után a mező pozitíva vált  
 B7 - B7=0 - üres mező kitöltése  
 B7=7 - kitöltött mezőben módosítás (ekkor a rutin hívása előtt már AS-nek tartalmaznia kell a módosítandó karaktert, "B" megadása viszont szükségesletlen. "A"-val pozicionálhatjuk a kurzort AS-on belül)  
 B8 - B8=0 - megszakítási funkció nincs engedélyezve  
 B8=7 - megszakítási lehetséges  
 B0 - B0=0 - üres mező nem zárható le CR-rel  
 B0=7 - üres mezőt is elfogad  
 CO - A CO szinkód értékei: fekete (0), fehér (1), piros (2), cian (3), lila (4), zöld (5), kék (6), sárga (7), narancs (8), barna (9), halv. piros (10), szürke (11), szürke2 (12), vil. zöld (13), vil. kék (14), szürke3 (15).

## A felhasznált változók

A,A2,AV,AS,A15,A25,A45,T5  
 B,B1,B2,B3,B4,B5,B6,B7,B8  
 C,CO  
 EE  
 HH  
 I  
 L  
 LN,N(12)  
 S1,S2,S3,S4  
 T1,T2,T3,T4  
 X  
 Y  
 Z

histogram

XXXXXXXXXXXXXXXXXXXX

```
14 16.4 16.8 21.2 20.6 26
+.....+.....+.....+.....
```

```
jan.
feb.
=====
márc.
=====
apr.
=====
máj.
=====
jún.
=====
```

```
9502 REM *****HISTOGRAM*****
(5-C14)Z=OSZLOPOK SZÁMA
MX=MAXIMUM MN=MINIMUM
P$=OSZLOP ANUAAA
X$=HÓNAJ NEVE
Y =A HÓNAP ADATA
IC=LEPESKÖZ AN=MX KÖZÖTT
9503 REM *****HISTOGRAM*****
9508 INPUT "HISTOGRAM CIME: ";C$
ZLOPOK SZÁMA:
9510 LET MX=0: LET MN=1000000: L
ET I$(#)
+.....+.....+.....+.....+.....
+.....+.....+.....+.....+.....
===== DIM X(Z,10): DIM Y(Z)
9520 REM *****
9522 FOR N=1 TO Z
9524 INPUT "TARGYHÓNAJ: ";X$(N)
MENNISÉG: ";Y(N)
9526 IF Y(N)=0X THEN LET MX=Y(N)
9528 IF Y(N)>MN THEN LET MN=Y(N)
9530 NEXT N
9540 REM *****KILÉPÉS ÉS PRIN
9541 CLS: PRINT TAB 16-(LEN T
/2):#
9542 LET IC=(MX-MN)/5
9544 PRINT TAB 3;MN,TAB 6;MN+IC,
TAB 13;MN+2*IC,TAB 18;MN+3*IC,TAB
23;MN+4*IC,TAB 28;MX,TAB 3;L$
9545 FOR N=1 TO Z
9550 PRINT X$(N) "====",TAB 4;p
(1 TO Y(N))*25/(MX-MN))-MN+125/
(MX-MN)))
9552 STOP
9560 REM *****
9561 CLS: LPRINT TAB 16-(LEN T
/2):#
9562 LET IC=(MX-MN)/5
9564 LPRINT TAB 3;MN,TAB 6;MN+IC,
TAB 13;MN+2*IC,TAB 18;MN+3*IC,TAB
23;MN+4*IC,TAB 28;MX,TAB 3;L$
9566 FOR N=1 TO Z
9570 LPRINT X$(N) "====",TAB 4;p
(1 TO Y(N))*25/(MX-MN))-MN+125/
(MX-MN)))
9570 NEXT N
9572 STOP
9570 LIST
9580 SAVE "HISTOGRAM" LINE 9570
```

A program sok magyarázatot nem kíván. Szubrutinként is jól használható. Ebben az esetben az X\$ és az Y változók értéküket a főprogramtól kaphatják. Ez esetben a 9554 és a 9572-es sorokban a STOP helyére RETURN kerül.

Természetesen az oszlopok bármilyen mennyiségű aktuális értékét ábrázolhatják. Az ábrázoláshoz használt "=" jel a nyomtatott kimélet, helyette a képernyőn bármilyen grafikus jel, például csillag, inverz space írható.







# A DOS 5.1 program

A legtöbb VC 1541 lemez meghajtó tulajdonosa nem tudja, hogy milyen hatékony eszköz van a birtokában: a DOS 5.1 nevű program, amely a floppy DEMO lemezén található. A következőkben ezt a programot ismertetjük részben a nyugatnémet Commodore újság 1984. évi 5. számában olvasottak alapján.

A program három funkciója:

- a floppy hibacsatorna gyorsan és kényelmesen kiolvasható;
- a lemez directory megjeleníthető a képernyőn anélkül, hogy a gépben levő program megsemmisülne;
- a lemezparancsok e rendszer alatt rövidített formában adhatók meg.

A programot a következőképpen lehet betölteni a gépbe: LOAD "C-64 WEDGE", 8.

Ezután RUN hatására a gép betölti a DOS 5.1-et az 52224-53080 (hex: CC00-CF58) címre. Ezt a területet a NEW utasítás nem éri el, azaz NEW hatására, illetve más BASIC program beolvasása után is bent marad a memóriában. A DOS 5.1 parancsokat direkt utasításként kell megadni, azaz nem építhetők be a programba.

A legtöbb parancs @ jellel kezdődik, ami helyettesíthető a >-val, hatása ugyanaz. A program használja még a ←, ↑, / és a % jelet.

A ← jel helyettesíti a SAVE parancsot: ← programnév megegyezik a SAVE "programnév", 8-cal.

A ↑ jellel a LOAD parancsot és a program indítását helyettesíthetjük: ↑ programnév ugyanaz, mint LOAD "programnév", 8:RUN.

A / jellel a LOAD parancsot helyettesíthetjük, de ez a jel nem indítja el a programot.

A % jellel gépi kódú programot tudunk betölteni a tárba. % programnév hatása ugyanaz, mint LOAD "programnév", 8.1.

Ha a lemezegységén villog a piros lámpa, tehát hibajelzés van, akkor a @ és RETURN lenyomása után megjelenik a képernyőn a hibaszám, a hiba, a sáv és a szektor, ahol bekövetkezett a hiba. Így a floppy-hibacsatorna kiolvasásához nem kell külön programot írunk. Ha nincs hiba, akkor 00, OK, 00, 00 jelenik meg a képernyőn.

A @\$ parancs segítségével kihartható a directory anélkül, hogy a gépben levő program felülíródna. A további parancsok:

DOS 5.1 parancs	Parancs DOS 5.1 nélkül
@N: lemeznév, ID (a directoryt törli)	OPEN 15,8,15 PRINT #15,"N:diszknev ID" CLOSE 15
@S: programnév (a programot, ill. állományt törli a lemezen)	OPEN 15,8,15 PRINT #15,"S:programnév" CLOSE 15
@R: új név = régi név (átnevezés)	OPEN 15,8,15 PRINT #15,"R:új név = régi név" CLOSE 15
@C: másolatnév = eredeti program (duplikál egy programot)	OPEN 15,8,15 PRINT #15,"C:másolatnév = eredeti program" CLOSE 15

A DOS 5.1-et időlegesen kikapcsolni a @Q parancsral, újraindítani pedig a SYS 52224 parancsral lehet.

RICHTER JÖRG



```

1 POKE 52,28:POKE 56,28:Y=7168:X=34816:FOR B=#TO 511:PO-
  KEX=X+B:PEEKY+Y:BNEXT
2 READ:IFH=-:ITHEN NEW
3 F=X+H:FOR L=FTOF+7:READH:POKE
  L,H:NEXT:GOTO 2
4 DATA 23,16,124,146,130,130,254,130,0:REM A' (W)
5 DATA 36,8,254,136,128,224,128,254,0:REM E' (S)
6 DATA 59,130,56,198,130,130,130,124,0:REM O,(.)
7 DATA 0,42,130,130,130,130,124,0:REM U,(@)
8 DATA 27,16,124,146,146,130,130,124,0:REM O' (j)
9 DATA 60, 40, 124, 170,170,130,130,124,0:REM O' (<)
10 DATA 39,146,146,146,130,130,134,122,0:REM U' (f)
11 DATA 30,42,170,170,130,130,134,122,0:REM U' (t)
12 DATA 17,16,56,4,60,68,58,0:REM KIS A' (Q)
13 DATA 35,3,16,60,66,126,64,60,0:REM KIS E' (W)
14 DATA 61,36,0,60,66,66,66,60,0:REM KIS O' (.)
15 DATA 43,36,0,66,66,66,70,58,0:REM KIS U' (+)
16 DATA 29,8,8,60,66,66,66,60,0:REM KIS O' (j)
17 DATA 62,30,20,60,82,66,60,60,0:REM KIS O' (>)
18 DATA 38,16,16,74,66,66,70,58,0:REM KIS U' (R)
19 DATA 28,40,40,10,66,66,70,58,0:REM KIS U' (E)
20 DATA -1

```

## A program használata

A rutint csak új program írásakor vagy meglévő program átalakításakor tudjuk használni. A programot betöltjük, az önmagát törli, és ezután írjuk rá az új főprogramot. Ilyen esetekben a komplett program két részből áll: ha a rutin 2. sorában NEW helyett LOAD-ot írunk, a két részprogram folyamatosan betölthető.

A főprogramban az új betűkészlet bekapcsolása POKE 36869, 255-tel történik (eredeti tárterület 240); ez a video-tárterület (VIC) egyik RAM tárcíme, amelynek tartalma a video-(képernyő-) mátrix és a jelgenerátor kezdőcímenek részét képezi (külön-külön az alsó és felső félbájt).

Az eredeti karaktereket – ez a „nem változtatott” betűknél kisetű helyett nagybetűt jelent – RVS/ON üzemmódban tudjuk beírni (például Q, W, + stb.), majd RVS/OFF-fal folytatjuk az új karakterek írását. RUS/ON hatására a képernyőn a ROM-ból beolvasott nagybetűkészlet jelenik meg.

Ha csak nagybetűvel akarunk írni, legyen az 1. sorban Y = 32768, ez a jelgenerátor nagybetűkészletének kezdőcíme. A rutint csak a 11. sor végéig írjuk be, nem feledve a "–1"-et a sor végéről.

A program közben is válthatjuk a 36869 cím tartalmát. Ha ez lapozással esik egybe, a különböző karakterkészletek a program során keverhetők.

Visszanyerhető az eredeti tárterület, ha a főprogram szövegét nem tartalmaz. Ekkor a program szöveges leírása után adjunk LOAD utasítást; a töltés az előző programot automatikusan törli. A BASIC és sztringterület határát a LOAD-ot megelőzően állítjuk vissza: POKE 52,30:POKE 56,30.

Úgy tűnhet, hogy meglehetősen körülményes a magyar karakterek használata ezen a géptípuson. Kis gyakorlással, kísérletezéssel azonban könnyen meg lehet szokni ezeket az egyszerű programozási fogásokat, amelyek lehetővé teszik, hogy gépünk helyesen írjon magyarul. És ez megéri a fáradságot.

HEIDEKKER GYÖRGY

olvassa be a ROM jelgenerátorból a kisetűbetűkészletet: 64 karakter, karakterenként 8 bájt, azaz  $64 \times 8 = 512$  bájt.

A 2-es és 3-as sor az új ékezetes betűket tölti be ugyanennek a területnek a ritkán használt karaktereibe. Bizonyos fogásokkal azonban ezek is használhatók!

A 4-19-es sor adatait az egyes új betűket határozzák meg. Egy-egy új karakter 8 bájt ( $8 \times 8$ -as mátrix), az adatok egy-egy bájt decimális megfelelő.

Az egyes DATA sorokban az első szám annak a karakternek a Commodore képernyőkódja, amelynek az új, ékezetes betű megfelel, illetve amelynek a billentyűjét lenyomva, kiíratjuk a magyar betűt.

A betűkészletből szándékosan hiányzik a hosszú í, kísérletező kedvűk bizonyára könnyen elkészítik például a 20-as sorba, a % jel helyére, végére helyezve a programot leállító "–1"-et.

## A hátrányokat se titkoljuk

A program legényesebb hátrányait a következők:

- az amúgy is szűkös BASIC tárterület 3069 bájtja csökken;
- eltűnik a kurzor;
- az átalakított jelek eredeti karakterei (Q, W, + stb.) körülményesebben írhatók;
- a kész magyar betűkészletet tartalmazó programok általában csak több részprogramból tölthetők be;
- nem tudunk negatív karaktereket írni.



## ZX-SPECTRUM

### Karaktergenerátorok

A karaktergenerátor feladata, hogy a képernyő megadott helyére (pozíció) előre megadott jelet (karaktert) helyezzen. A ZX-Spectrum által használt generátor egy sorba 32 db 8 bit széles karaktert írhat. Ha több információ akarunk megjeleníteni egy sorban, keskenyebb karaktereket kell alkalmaznunk.

Az itt közölt karaktergenerátorok egyike 4, a másik 6 bit szélességű karaktereket ír. A jelek magassága itt is 8 bit. Mindkét program relokálható.

#### Négybites generátor

Könnyen kiszámítható, hogy egy sorban 64 karakter jeleníthető meg. A pozíciók sorszámozása soros elrendezésű, azaz a bal felső sarok a 0., a jobb alsó a (64 × 24 - 1) 1535. pozíció. Ezt a pozíciószámot a 23728-es és 23729-es címen elhelyezkedő, Spectrum által nem használt rendszerváltozóban kell elhelyeznünk a következő módon: POKE 23728, p-256\*INT(p/256)  
POKE 23729, INT(p/256)

A megjelenítendő karakter ASCII kódját a 23681-es címen levő, ugyancsak használatlan rendszerváltozóban helyezhetjük el. Az eredeti karakterkészletet nem használhatjuk, hiszen most fele akkora betűkre van szükségünk. Így a 23606-23607-es címen elhelyezkedő rendszerváltozóban kell elhelyeznünk a saját karakterkészletünk kezdőcímét.

Karakterkészlet-generálásakor két egymást követő karaktert a következőképpen helyezzünk el az operatív tárban (1. ábra):

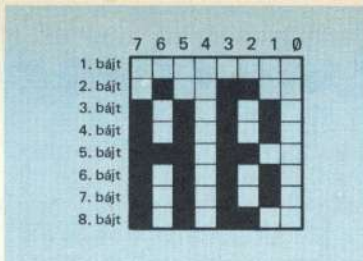
CÍM	TARTALOM
a+0	0
a+1	76
a+2	170
a+3	170
a+4	236
a+5	170
a+6	170
a+7	172

Komolyabb feladatok megoldására ez a program nem ajánlott, mert az abc-nek egyes betűi nem generálhatók egyértelműen. Számok viszont kitűnően megjeleníthetők feleakkora helyen, mint a hagyományos módszerrel. A karakterkészlet kiterjedése bármekkora lehet. Például az ASCII karakterkészlet vezérlőkarakterekkel 512 bajton elhelyezhető.

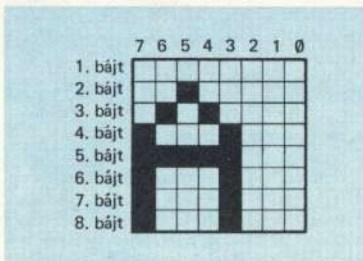
#### Hatbites generátor

A pozíció megadása:  
23728: oszlopszám (0-41)  
23729: sorszám (0-23)

A 42 db karakter nem tölti ki teljesen a képernyőt.



1. ábra



2. ábra

ernyőt:  $256 - 42 \cdot 6 = 4$  bitet nem használ ki a program. A karakterkód betöltése az előzőhöz hasonlóan történik, a karakterkódot a 23681-es címre helyezzük. Ha egyszerűség kedvéért gépi kódból hívjuk meg a karaktergenerátort, a közölt program első 7 bajtjára nincs szükség, így az A regiszterben az ASCII kódot, a D regiszterben a sorszámot, az E-ben pedig az oszlopszámot adhatjuk meg.

Karakterkészlet készítésénél az előbb említett módon kell eljárunk, de itt 8 bajton egy karaktert helyezzünk el 2 helyett. Vigyáznunk kell továbbá arra, hogy minden bajt 0 és 1. bitje 0 legyen (lásd a 2. ábrát).

CÍM	TARTALOM
a+0	0
a+1	32
a+2	80
a+3	136
a+4	248
a+5	136
a+6	136
a+7	136

Ez a generátor már a teljes angol betűkészlet képes megjeleníteni.

Példa a "COMPUTER" szó kiírására:

```

10 REM 4 BIT SZELESSEG
20 REM 2.SOR 10. OSZLOP
30 LET A$ = "COMPUTER"
40 FOR N=1 TO LEN A$
50 POKE 23728, N+137 : POKE 23729, 0
60 POKE 23681, CODE A$(N)
70 RANDOMIZE USR BELEPES
80 NEXT N

10 REM 6 BIT SZELESSEG
20 REM 2. SOR 10. OSZLOP
30 LET A$ = "COMPUTER"
40 FOR N=1 TO LEN A$
50 POKE 23728, N+9 : POKE 23729,2
60 POKE 23681, CODE A$(N)
70 RANDOMIZE USR BELEPES
80 NEXT N
    
```

A példákban a BELEPES nevű változó a karaktergenerátorok kezdőcímét tárolja. Itt eltekintettünk a karakterkészlet kezdőcímének átállításától, de ezt a gyakorlatban természetesen el kell végezni.

### Ujjgyakorlatok a képernyőfájra

Mint ismeretes, a ZX81-nél úgy szoktunk visszatöltés után automatikusan induló programot írni, hogy a programba írjuk bele a SAVE utasítást. Ezt aztán a billentyűzetről vezérlési üzemmódban GOTO parancssal hajtjuk végre. Ekkor a gép elvégzi az ott található SAVE utasítást, majd folytatja a program futtatását.

A leírtak miatt a magnószalagra kerülő programba a NXLIN rendszerváltozóba a SAVE utasítást követő utasítás abszolút címe kerül (tehát nem a BASIC-sorszáma). Visszatöltés után a gép a program futtatását az ezen a címen kezdődő BASIC sor végrehajtásával folytatja.

Láttam már olyan játékpogramot, amely a játék végén ahelyett, hogy új játékot adott volna, magnótöltésbe kezdett. Ez nem annyira programozói, mint inkább kezelői hiba. A programozó ti. olyan sorszámokra helyezi el a SAVE utasítást, amelyre sohasem adódhat át a vezérlés; általában a program végére. Néha - egyszerű programozói trükk - olyan helyre bújtatják el a SAVE-et, ahol nehéz észrevenni, például szubrutinok közé. Ilyenkor a SAVE előtt RETURN áll. Ha a kezelő véletlenül bebillentyűzi ennek sorszámát, anélkül, hogy tudná, törli ezt a sort. Az így magnóra töltött program visszatöltés után beletéved a SAVE utasításba.

A fentiek illusztrálására bemutatok egy példát.

```

10 LET I=INT(2*RND)
30 IF I <= 1 THEN GOSUB 300
...
50 IF I > 1 THEN GOSUB 400
...
70 GOTO 10
    
```

```

300 ... utasítások
320 RETURN
330 SAVE "PELDA"
340 RUN
...
400 ... utasítások
420 RETURN
    
```

Ebben a programban, ha valaki nem nyomja meg eléggé a LIST gombot, és listáztatni akarja a programot, LIST 320 helyett a 320-as sor törlését idézi elő. A hibát nem kell feltétlenül észrevenni, hiszen törlésnél is listázást végez a gép. Ha ezt a programot SAVE vagy GOTO 330 utasítással magnóra töltjük, a program lát-szólag hibátlan, még ízben is lefutathat, míg egyszer a 300-as szubrutinba téved.

A fentiekhez hasonlóan érdemes megnéznünk, lehet-e megöltött képernyőfájlt magnóra vinni. Ez talán kevésbé ismert téma. Csupán azt kell tudnunk, hogy mikor törli a képet a gép, és mikor nem. Nos, a ZX81 mindig törli a képet, ha a billentyűzetről kiadott parancssal indítjuk (a beviteli műveletek nem tartoznak ide). Ezért értelmetlen például billentyűzetről PLOT utasításokkal rajzolni.

Ezek után világos, hogy akár SAVE, akár GOTO utasítással töltjük magnóra programunkat, azonnal képernyőtörölés jön létre, és a szalagon főlöseges helyet foglal el a teljesen üres képernyőfáj (kb. 20 másodperc). A megoldás az, hogy megírt programunkkal kirajzoljuk a képet, és utána a programból végrehajtatjuk a SAVE utasítást, valahogy így:

```

10 első utasítás
8990 GOTO 10
    
```



9000 PRINT AT 6.6;"\*\*\*\*\*"  
 9010 PRINT AT 7.6;"\*"  
 9020 PRINT AT 8.6;"\* PROBA PROGRAM \*"  
 9030 PRINT AT 9.6;"\*"  
 9040 PRINT AT 10.6;"\*\*\*\*\*"  
 9050 PRINT AT 21.30;"ZJ"  
 9060 SAVE "PROBA PROGRAM"  
 9070 RUN

Futás közben a vezérlésnek nem szabad átadódnia a feliratozó programrésztelre. Magnóra töltést így indítjuk: GOTO 9000. Magnóról való visszatöltése után a tervezett címfelirattal jelenik meg a program, és mindjárt fut is.

Most térjünk vissza a fentiekben elmondottak alkalmazására. Próbáljuk meg például átírni a NXTLIN változót. Írjunk rövid programot, majd billentyűzzük be: POKE 16425,125. Ennek hatására a program futása azonnal megindul (feltételezve, hogy a program valóban rövid). Persze így nem lehet magnóra kitölthető öninduló programot írni, mert a program lefut, és mire kitölnénk, a NXTLIN újra a program végére, a DFILE-ra mutat.

Tulajdonképpen elvégezhetjük a fenti műveleteket gépi kódú utasításokkal is, a következőképpen: 21 7D 40 LD HL, FIRSTLINE  
 22 29 40 LD (NXTLIN), HL  
 C3 07 03 JP SAVE ROUTINE

(T. Baker után)

Azok számára, akik nem jártasak ebben, van más javaslatom is: a LOADER használata. Mióta megvan, nagy haszonnal alkalmazom a Raucher Attila-féle LOADER-t. Az alábbiakban szereplő címek erre vonatkoznak; más LOADER-eknél utána kell nézni, hová kerülnek a szükséges címek és programrészteltek.

Raucher a beolvasott program nevét a 4B00=19200 cím után helyezi el, a programot magát pedig a 4C00=19456 címtől. Tehát a betöltött program valamennyi címadata a 0C00<sub>H</sub>=3072<sub>D</sub> értékkel toldódik el:  
 D FILE 400C<sub>H</sub> → 4C0C<sub>H</sub> = 19468<sub>D</sub>  
 NXLIN 4029<sub>H</sub> → 4C29<sub>H</sub> = 19497<sub>D</sub>  
 ELSŐ SOR407D<sub>H</sub> → 4C7D<sub>H</sub> = 19581<sub>D</sub>

Ezek alapján az alábbiak szerint hajtjuk végre a varázslást. Betöltjük a LOADER-t, elindítjuk. Beolvasatjuk vele a programot, amellyel manipulálni akarunk. A LOADER megáll, kírja a program nevét és hosszát, és megkérdezi, hogy SAVE-t vagy LOAD-ot akarunk-e végezni. Ne kérjünk LOAD-ot! Ez elrontja a beolvasott programot. Írjunk be S betűt. Erre megindul a magnóra töltés. Állítsuk meg a gépet BREAK utasítással, majd nyomjuk meg a SLOW billentyűt.

A dolog most már igen egyszerű. Írjunk be, hogy  
 POKE 19497,125  
 POKE 19498,64  
 RUN

A LOADER ekkor rutinszerűen megkérdezi tőlünk, hogy SAVE vagy LOAD? Írjunk S betűt, és így töltjük ki magnóra a tárolt és módosított programot. Ha ezt bármikor visszaolvasatjuk, automatikusan indulni fog, mintha GOTO 1 utasítást adtunk volna ki. Pedig hiányzik belőle a SAVE is és a GOTO 1 is! Ha valakinek türelme van, megkeresheti a betöltött (tárolt) program bármelyik sorát, és arra is átadhatja a vezérlést.

A fentiek után nyilvánvaló, hogy akár a képernyőfajlt is megkereshetjük:

LET DFILE=PEEK 19468+256\*PEEK 19469+3072

Ide be is írhatunk bármit. Például a nevem kezdőbetűit a következőképpen írhatom fel a bal felső sarokba:

POKE DFILE+1,63  
 POKE DFILE+2,47

Ezután a fenti módon magnóra töltöm a LOADER-rel tárolt programot. LOADER nélküli visszaolvasás után alul megjelenik a 0/0, a bal felső sarokban pedig: ZJ.

DR. ZANA JÁNOS

## ZX-SPECTRUM

# Gépi kódú rutin BASIC sorok átszámozásához

BASIC programjaink bővítése, tesztelése során gyakran kerülünk megoldhatatlannak látszó helyzetbe, ha két olyan program sor közé kell egy harmadikat illesztenünk, amelyek sorszámainak értéke csak 1-gyel tér el.

Gyakran tanácsalanok vagyunk akkor is, ha több, már előre elkészített rutint akarunk egyetlen programba foglalni, de nem tudjuk ezeket egymás után fűzni, mivel sorszámaik átfedik egymást, tehát még a MERGE utasítást sem használhatjuk.

A következő gépi kódú segédprogram ezt a feladatot oldja meg. A program – minimális módosításokkal – futtatható minden olyan Z80 alapú mikroszámítógépen, amelynek BASIC sora – a sorkezdést és sorvég kitüntetést – azonos felépítésű a Spectruméval. A program a 7F 34H (32 564D) címen kezdődik, de a szabad tárterület bármely részén elhelyezhető, mivel nem tartalmaz direkt ugrási utasításokat.

A program kezeli a rendszerváltozók egy részét, amelyek géptípusonként más és más helyen vannak, tehát ezeket mindig az adott géptípus határozza meg.

A segédprogramot egy bevívó programmal helyzetjük el a tár kívánt részén. A program-bevitelt, majd NEW után a rendszer készen áll BASIC programjaink újrászámozására. (A rutin kimentésével és újrátöltésével a kézikönyv 20. fejezete foglalkozik.) A rutin segítségével eredeti BASIC programunkból tetszőleges kezdő sorszámú és lépésközi programot transzformálhatunk, természetesen 1-9999 között elhelyezkedő határokkal.

Figyelem! A program a GO TO, GO SUB és az egyéb, soron belüli sorszámhivatkozásokat nem módosítja, ezt manuálisan kell elvégeznünk. Erdemes a hivatkozott sor elé rövid REM utasítást írni, ezzel mintegy megelőzve a célsorokat.

A rutin a következő paranccsal aktivizálható:

LIST kezdő sorszám  
 RANDOMIZE lépésköz  
 RANDOMIZE USR kezdő cím (CIM)

A "LIST kezdő sorszám" helyett a sorszámkezdést kijelölése a következőképpen is végrehajtható:

kezdő sorszám ENTER

de ezt elővigyázatosan alkalmazzuk, mert már meglévő sorszámat törölhetünk.

### A program

```
10 REM BEVIVO PROGRAM
20 CLEAR 32563:REM UJ RAMTOP
30 LET CIM=32564:REM KEZDOCIM
40 LET AS="" :ide kerül a programlista 2. oszlopában álló karaktersorozat"
50 FOR N=0 TO LEN AS-2 STEP 2
60 LET A=CODE AS(N+1)-48-/7
  AND AS(N+1)>="A"
70 LET B=CODE AS(N+2)-48-/7 AND
  AS(N+2)>="A"
80 POKE CIM+INT(N/2),16*A+B
90 NEXT N
```

### A program listája

CIM (HEX)	KÓD	Z80 ASSEMBLER
7F 34	2A 53 5C	LD HL,(PROG)
7F 37	DD 2A 49 5C	LD IX,(E PPC)
7F 3B	DD E5	QL PUSH IX
7F 3D	D1	POP DE
7F 3E	72	LD (HL),D
7F 3F	23	INC HL
7F 40	73	LD (HL),E
7F 41	23	INC HL
7F 42	4E	LD C,(HL)
7F 43	23	INC HL
7F 44	46	LD B,(HL)
7F 45	09	ADD HL,BC
7F 46	23	INC HL
7F 47	ED 4B 48 5C	LD BC,(VARS)
7F 4B	E5	PUSH HL
7F 4C	ED 42	SBC HL,BC
7F 4E	E1	POP HL
7F 4F	F0	RET P
7F 50	ED 5B 76 5C	LD DE,(SEED)
7F 54	DD 19	ADD IX,DE
7F 56	18 E3	JR QL

MAGYAR ZOLTÁN

## Parafrazis





# M PROGRAMOK

## COMMODORE 64

### Megszakítás I.

A Commodore 64 megszakításrendszere sokféle feladat – elsősorban játékok – megoldásához nyújt hasznos támogatást. Megpróbáljuk a legfontosabb ismereteket közreadni ezzel kapcsolatban, mivel sehol sem találtunk teljes utalást. A felhasználó kétfajta maszkolható megszakításrendszert használhat. Az egyik a Video Interface Chip (a továbbiakban VIC) által generált, a másik az, amit a Complex Interface Adapterek (a továbbiakban CIA) kezelnek. Most az utóbbról lesz szó.

A gépen normál körülmények között 1 másodperc alatt kb. 60 megszakítási kérelem jelentkezik. Ilyenkor megszüntet minden munkát, az állapotregisztert és az utasításszámlálót a verembe rakja (RTI-nel ezeket az értékeket visszatölti a megfelelő helyre). Végrehajt egy feltétel nélküli ugrást a \$FFFE(65534)-en található címre, alapállapotban ez \$FF48(65352). Az itt levő rutin először elmenti a regisztereket a verembe, mégpedig a következők sorrendben: akkumulátor, X-regiszter, Y-regiszter. Utána aszerint, hogy BREAK történt vagy nem, kerül a vezérlés a \$0316(790)-en vagy \$0314(788)-on levő címre. (A BREAK is megszakítást generál, amelyet az különböztet meg a többitől, hogy az állapotregiszternek nem a 2., hanem a 4. bitje lesz 1-es.) Mi az előbbi esetek közül csak a másodiknál foglalkozunk. Ennek a címnek a tartalma eleinte \$EA31(59953). Ez a rutin végzi a 0. lapon elhelyezkedő óra aktualizálását, a kurzorvillogtatást és a billentyűzet ellenőrzését.

Ha saját célunkra szeretnénk felhasználni a megszakítást, akkor a \$0314(788) tartalmat kell átírni úgy, hogy a saját rutinunk címét írjuk be. A KERNAL kikapcsolásakor a \$FFFE(65534)-et használjuk. Vigyázzunk arra, hogy a regisztereket ilyenkor mindjárt az elején tegyük a verembe, és csak a rutinunk végén mentjük vissza. Ekkor természetesen az eredeti rutin nem kerül végrehajtásra.

A példaprogram a képernyő jobb felső sarkában megjeleníti a pontos időt, majd egy megadott időpontban kirja: ébresztő. Időzítések alkalmazásához remek segítséget nyújt.

Mindkét CIA-chipnek van egy 4 regiszteres, decimális üzemmódu, írható, olvasható, tizenkét órás, délután-délután jelzővel ellátott, ébresztős belső órája, amely a chipke 8. (tizedmásodperc), 9. (másodperc), A. (perc), B. (óra) regiszterein helyezkedik el. A B regiszter 7. bitje jelzi

#### BASIC segédlet az ébresztő beállításához

```
150 CIM=7*4096
200 PRINT "KEREM AZ IDOT ES AZ EBRESZTEST"
210 INPUT " (OOPP OOPP) ";A#
250 IDO=VAL(LEFT$(A#,1))*16+VAL(MID$(A#,2,1))
270 IF IDO>16+2 THEN 500
280 POKE CIM,IDO
300 IDO=VAL(MID$(A#,3,1))*16+VAL(MID$(A#,4,1))
310 IF IDO=>6*16 THEN 500
320 POKE CIM+1,IDO1
350 ALM=VAL(MID$(A#,6,1))*16+VAL(MID$(A#,7,1))
370 IF ALM>9*16+2 OR (ALM<9*16 AND ALM>16+2) THEN 500
380 POKE CIM+2,ALM
400 ALM=VAL(MID$(A#,8,1))*16+VAL(MID$(A#,9,1))
420 IF ALM=>6*16 THEN 500
430 POKE CIM+3,ALM
450 SYS CIM+13
460 END
500 PRINT "HIBAS ADAT"
510 GOTO 210
```

a délélt-délután. Az óra íráskor addig nem indul el, amíg a tizedmásodperceket be nem írjuk, ezért ezt ajánlatos utólagra hagyni. Ugyanez a módszer az óra olvasásakor is. A regiszterek nem módosulnak, míg a tizedmásodperceket ki nem olvastuk. Az óra persze fut tovább.

Ezeket a regisztereket használhatjuk az ébresztő programozásához is. Ha a CIA-chip F regiszterének 7. bitje 1-es, akkor az ébresztőt, ellenkező esetben az órát írjuk. Olvasáskor mindig csak az órát kapjuk. Az érvényes ébresztési időpont az utólagra beírt érték, az előzőek elvesznek. Az ébresztő működtetéséhez engedélyezni kell az ébresztőre való megszakítást is. Ezt a \$DC0D(56333) módosításával érhetjük el. Ha ennek a regiszternek a 7. bitje magas, akkor ha a regiszter többi bitjébe 1-et írunk, a bit magas lesz, ha pedig 0-t írunk, akkor a bitek értéke nem változik. Olyan, mintha VAGY műveletet végeznénk. Mikor a 7. bit 0, akkor éppen fordítva működik. A bitek 1-et írva, azok törölődnek, 0-t írva nincs változás. Olyan, mintha ÉS műveletet hajtanánk végre. Megszakítást akkor engedélyezünk, ha a megfelelő bit magas.

#### A program gépi kódú kiegészítője

```
10 #=17000
20 HOUR .BYTE #00
30 PERC .BYTE #00
40 ALHOUR .BYTE #00
50 ALPERC .BYTE #00
60 ENGED .BYTE #01
70 UZENET .DISP 'EBRESZTO'
80 SET
90 LDA #0314 ;IRO-CIM LO-BYTE ELMENESE
100 STA VISSZA+1
110 LIR #0315 ;IRO-CIM HI-BYTE ELMENESE
120 STA VISSZA+2
130 LDA #076 ;UJ CIM BEIRASA
140 STA #0314
150 LDA #070
160 STA #0315
170 LDA #000 ;FREKVENCIA ALLITASA 50 HZ-RE
175 ORA #D0E
180 STA #D0E
190 LDA HOUR ;IDO BEIRASA
200 STA #D0B
210 LDA PERC
220 STA #D0A
230 LDA #000
240 STA #D09
250 STA #D08
260 LDA #000 ;EBRESZTORE ALLITAS
270 ORA #D0F
280 STA #D0F
290 LDA ALHOUR ;EBRESZTESI IDO BEIRASA
300 STA #D0E
310 LDA ALPERC
320 STA #D0A
330 LDA #000
340 STA #D09
350 STA #D08
360 LDA #07F ;VISSZALLITAS ORARA
370 AND #D0F
380 STA #D0F
390 LDA #084 ;EBRESZTO MEGSZAKITAS ENGEDELYEZESE
400 STA #DC0D
403 LDA #D0D ;CONTROL REGISZTER TORLESE
406 STA #D0D
410 CLI
420 RTS
430 LDA #C5
440 CMP #04 ;F1 LE VAN-E NYOMVA
450 BNE IDE
460 LDA #01 ;ORA KIIRAS MEHET
470 STA ENGED
480 JMP TOVR3
490 IDE CMP #05 ;F3 LE VAN-E NYOMVA
500 BNE TOVR2
510 LDA #00 ;ORA KIIRAS LEALLITASA
520 STA ENGED
530 TOVR2 LDA ENGED
540 BNE TOVR2
550 JMP VISSZA
560 TOVR3 LDA #D0D ;TORTENT-E EBRESZTO MEGSZAKITAS
570 AND #01
580 BEQ TOVRS
590 JMP EBRESZ
600 TOVRS LDA #D0B ;ORA-BYTE OLVASASA
610 AND #10
620 BEQ TOVRS4
```



Ezek közül számunkra most a megszakítás-regiszter a lényeges. A következőképpen néz ki:

```

630 LDA #F31 ;HA VAN TIZES
640 STA #0420 ;0.SOR 32.KARAKTER
650 JMP TOVA1
660 TOVA1 LDA #F20 ;HA NINCS TIZES->SZOKOZ
670 STA #0420 ;MINT 640
680 TOVA1 LDA #DD0B ;MINT 600
690 AND #F0F ;CSAK AZ EGYESEK
700 CLC
710 ADC #F30 ;KEPERNYO KODJA ALAKITAS
720 STA #0421
730 LDA #DD0A ;PERC OLVASASA
740 AND #F0 ;CSAK A TIZESEK
750 LSR A ;ELTOLAS EGYESEKKE
760 LSR A
770 LSR A
780 LSR A
790 ADC #F30 ;MINT 710
800 STA #0423
810 LDA #DD0A ;MINT 730
820 AND #F0F ;MINT 690
830 CLC
840 ADC #F30 ;MINT 710
850 STA #0424
860 LDA #DD09 ;U.A. A MODSZER MINT A PERCEL
870 AND #F0 ;CSAK MOST MASODPERCEL
880 LSR A
890 LSR A
900 LSR A
910 LSR A
920 ADC #F30
930 STA #0426
940 LDA #DD09
950 AND #F0F
960 CLC
970 ADC #F30
980 STA #0427
990 LDA #DD09
1000 LDA #F3A ;" " KEPERNYO KODJA
1010 STA #0422
1020 STA #0425
1030 LDY #F0F
1040 LDA #F01 ;FEHER SZIN BEALLITASA
1050 CIKL1 STA #D81E,X ;SZIN MEMORIA 0.SOR 30.KARAKTER
1060 STA #D84E,X ;SZIN MEMORIA 1.SOR 30.KARAKTER
1070 DEX
1080 BNE CIKL1
1090 LDA #F5D ;" " KEPERNYO KODJA
1100 STA #041F
1110 LDA #F6D ;" " KEPERNYO KODJA
1120 STA #0447 ;1.SOR 31.KARAKTER
1130 LDA #F49 ;" " KEPERNYO KODJA
1140 LDY #F08
1150 CIKL1 STA #0447,X
1160 DEX
1170 BNE CIKL1
1180 JMP VISSZA
1190 EBRESZ LDY #F00 ;EBRESZTIS
1200 CIKL2 LDA UZENET,X ;SZOVEG KIIRASA
1210 STA #0470,X ;2.SOR 32.KARAKTER
1220 LDA #F71 ;FEHER SZIN
1230 STA #D870,X ;SZIN MEMORIA 2.SOR 32.KARAKTER
1240 INX
1250 CPY #F08
1260 BNE CIKL2
1270 LDA #F84 ;MINT 403
1280 STA #DD0D
1290 VISSZA JMP #FFFF ;VISSZA AZ EREDETI RUTINBA
1300 .END

```

Nézzük meg egy kicsit részletesebben a CIA-chip regiszter-kiosztását.

CIA #1:(megszakító regiszter olvasás:\$DC0D(56333))

7 6 5 4 3 2 1 0  
IR 0 0 FLG SP ALRM TB TA

CIA #2:(megszakító regiszter írás:\$DD0D(56589))

7 6 5 4 3 2 1 0  
S/C X X FLG SP ALRM TB TA

Ha megszakítás történt, akkor a megfelelő bit magas lesz. Látható, hogy ébresztés esetén ez a 2. bit. A regiszter úgy törölhető, hogy az aktuális tartalmat visszairjuk. Vigyázzunk, hogy minden megszakítás kezelése után tároljuk a megfelelő bitet, mert addig a gép nem generál újabbat, amíg a bit magas.

Például ha esetünkben kétszer szeretnénk használni egymás után az ébresztőt, akkor a másodikat nem kapnánk meg, ha nem töröltük előbb a megszakító olvasás-regisztert. Más megszakítás természetesen történik.

A program 20-as, 30-as sora az óra, a 40-es, 50-es sor az ébresztő adatait tartalmazza. Ezek beállítása történhet akár BASIC-ből, akár gépi kódból. Az előbbre ad példát a bemutatott BASIC segédlet. Az ébresztő állításánál ügyelni kell arra, hogy 12 óra után már délután van. Ilyenkor az eredeti értékhez adjunk hozzá 80-at. Például ha 12 óra 32 perc szeretnénk állítani az ébresztő értéket, akkor 92 óra 32 perc a helyes beállítás.

A 80-as sorban a cím átirásakor mindig le kell tiltani a megszakítókat, mert különben kiakad a gép.

Az enged bájttal az szerint 1-es vagy 0, hogy a F1-gyel engedélyeztük az óra kiírását, vagy F3-mal nem.

Ha valaki erre szeretne ébredni, akkor tanulmányozza a Sound Interface Device (SID) chip működését, és írja át az ébresztő részt. A gép ilyenkor nem szabad kikapcsolni.

MÜLLER GÁBOR-LAKY ZSOLT

## TUDÁSPRÓBA

1. A plusz 2 alapú számábrázolási rendszerben az egész számok előállításánál használt súlyok – mint tudjuk –  $2^0, 2^1, 2^2, \dots$ . A számjegyek (a súlyok együtthatói) 0 és 1 lehetnek. A mínusz 2 alapú számábrázolási rendszerben a súlyok  $(-2)^0, (-2)^1, (-2)^2, \dots$ , váltakozó előjelűek. A felhasználható számjegyek viszont szintén csak 0 és 1 lehetnek. Mi a legnagyobb abszolút értékű szám a legfeljebb négyjegyű – 2 alapú számábrázolási rendszerbeli számok között? (A – 2 alapú számábrázolási rendszerben nem használunk előjelet.)

2. Van-e olyan számábrázolási rendszer, amelyben egy  $n$  jegyű egész szám ellentettje (mínusz egyszerese) több mint  $n$  jegyű?

3. A – 2 alapú számábrázolási rendszerben felírt  $n$  jegyű egészek között pozitív vagy negatív szám van több?

4. Van-e olyan számábrázolási rendszer, amelyben azt, hogy egy egész szám pozitív vagy negatív, az ábrázolt szám számjegyeinek száma határozza meg?

5. Lehet-e két paritás hibás szám összege helyes?

6. Lehet-e két paritás hibás szám különbsége helyes?

7. Lehet-e két paritás hibás szám szorzata helyes?

8. Mi a nagysági viszonya a következő, – 2 alapú számábrázolási rendszerben felírt számoknak? 100, 111, 1010.

9. Igaz-e, hogy egy  $a_n$  ( $n = 1, 2, \dots$ ) számsorozat akkor és csak akkor tart egy véges  $a$  értékhez, ha van olyan  $a$ -hoz tartó, szigorúan monoton növekvő  $b_n$  ( $n = 1, 2, \dots$ ), és  $a$ -hoz tartó szigorúan monoton csökkenő  $c_n$  ( $n = 1, 2, \dots$ ) sorozat, hogy  $b_n \leq a_n \leq c_n$  igaz minden  $n$ -re?

10. Mi az ANDISIM?

11. Igaz-e, hogy „két 8 bites mikroprocesszort párhuzamosan kapcsolva 16 bites mikroprocesszort kapunk?”

12. Igaz-e, hogy a kereskedelmi forgalomban levő lézernyomatók a papírra égetik az információt?

13. Igaz-e, hogy a „tintasugaras” nyomtató tintasugárral ír?

14. Lehet-e „tintasugaras” nyomtatóval többszínű ábrát készíteni?

15. Igaz-e, hogy a „tintasugaras” nyomtatók piezoelektromos elven működő berendezéssel oldják meg a tintának a papírra továbbítását?

16. Van-e kristályos folyadékkal dolgozó nyomtató a piacon?

17. Hasznosították-e már a „magnetofon elvet” nyomtatókban?

18. Igaz-e, hogy a „mágnés dobos” nyomtatók hővel rögzítik a festékanyagot a papírra?

19. A piacon levő különböző típusú „tintasugaras” (vagy inkább tintacséppes) nyomtatók száma a) 10 alatti, b) 10 és 20 közötti, c) 20 feletti?

20. Igaz-e, hogy a lézernyomatókkal műanyagra is lehet írni?

Összeállította TAKÁCSY ILDIKÓ

### REGISZTER

### Funkció

0	A kapu adatregiszter
1	B kapu adatregiszter
2	A kapu adatirány-regiszter
3	B kapu adatirány-regiszter
4	A időzítő alsó bájtt
5	A időzítő felső bájtt
6	B időzítő alsó bájtt
7	B időzítő felső bájtt
8	tizedmásodperc
9	másodperc
A(10)	perc
B(11)	óra (de./du. jelző)
C(12)	soros kapu
D(13)	megszakításregiszter
E(14)	A kontroll-regiszter
F(15)	B kontroll-regiszter



## MELYIKET VEGYEM?

# MIRE JÓ A C16?

Sok olvasó kérésére – és más fórumokon elhangzott, elfogult tájékoztatások hatására – időről időre összehasonlításon alapuló tanácsokat adunk. Elsőként a legolcsóbb gépekről, mivel ezeket kéri leginkább.

Legolcsóbbnak az amerikai irodalomban használt kategorizálás szerint a 200 USA dollárnál olcsóbb gépeket értjük. E határnak kb. 600 nyugatnémet márka felel meg. Az USA és NSZK árakat közöljük, megjegyezve, hogy egyrészt a folyóiratokban közölt árak is szóródnak, másrészt a nem hirdető cégek árai ezektől jócskán eltérhetnek mindkét irányban.

Az MC (NSZK) 1985. szeptemberi száma alapján ide tartozó gépek néhány hardverjel-

lemzőjét tartalmazza az 1. táblázat. Az ár értelmezéséhez tartozik, hogy azonos teljesítményű és minőségű rendszerek összehasonlítása adna csak reális értékelést, de ilyet nem lehet készíteni. Például a fóliamembrános billentyűzetű ZX81 nem alakítható át készen vásárolható megoldással egy szokványos minőségű és élet tartamú billentyűzetes géppé; a különböző memóriakiépítések és bővítési lehetőségek eltérnek; az egyes egységek minősége nagyon különböző stb. Az árak összehasonlításakor figyelembe kell venni azt is, hogy a ZX81-nek nagyjából megfelelő Timex–Sinclair gép, a C16, az Atari 600XL július óta eltűnt az amerikai hirdetések közül, a Dragon 32-t, az MTX500-at pedig az Egyesült Államokban nem forgalmazták.

Ugyancsak figyelembe veendő, hogy egyszer (egyik előbb, másik utóbb, de végül is mind) elromlik, és akkor lehet-e javíttatni? Van-e mivel? Van-e kivel? Vannak gépek – például a Commodore és a Sinclair –, amelyeknek legfontosabb alkatrészei nem kerültek kereskedelmi forgalomba, vagy nem javíthatók, vagy ha mégis, csak rendkívül drágán. Gondoljunk csak a Sinclair gépek ULA-problémáira vagy a C64 alkatrészellátás máig is megoldatlan voltára.

Figyelembe kell venni azt is, hogy bármilyen vagy csak különleges perifériák csatlakoztathat-

tók-e a géphez? Lehet-e kapni egyáltalán olyan csatlakozókat, amelyek a perifériák illesztéséhez szükségesek? A C16 csekély üzleti sikerében és abban, hogy rendkívül rövid idő alatt eltűnt az amerikai piacokról, valószínűleg jelentős szerepet játszott az igen kis memória és a korábbi programok használhatatlansága mellett az, hogy sehol sem kapható csatlakozókat használ.

Az ugyancsak alapvető fontosságú alapszoftvert, amit a géppel együtt adnak, a 2. táblázat ismerteti.

Olyan gép, amely mindenféle célra a legalkalmasabb, vagy amelyik teljesen alkalmatlan, nem létezik. Egy konkrét cél és lehetőségeink felmérése esetén viszont már találhatunk egy legjobb vagy néhány közel egyformán jó gép-rendszert.

Kiindulási alap lehetne a két táblázat együttes használata, de az 1. táblázatnál említett problémákon kívül a szolgáltatott szoftver minőségére is vonatkozik az, hogy nagyon nagy az eltérés. Például a C64 és a C16 BASIC-fordító mérete azonos, de az első nagyon kis teljesítő-képességű, mert a gyártó intelligens játéknak tervezte, és a játékokat általában a fordító használata nélkül működtetik, a második pedig e hiányosságok jó részét megszüntette.

1. táblázat

GYÁRTÓ	ATARI CORP.		COMMODORE BUSINESS MACHINE		DRAGON DATA LTD.		MEMOTECH LTD.	SINCLAIR RESEARCH LTD.		
	ATARI 600XL	ATARI 800XL	C16	C64	32	64	MTX 500	ZX81	SPECTRUM	SPECTRUM+
Memória (k)	24 + 16	24 + 64	16 + 16	16 + 64	16 + 32	32 + 64	24 + 32	8 + 1	16 + 16	16 + 48
Használható memória (k)*	2	32	2	31	15	41	?	1	8	40
Processzor	6502C	6502C	7510	6510	6809E	6809E	Z80A	Z80A	Z80A	Z80A
Speciális alkatrész	3	3	2	5	–	–	?	1	2	2
Billentyűzet	62, normál tv. monitor, színes, 40 × 24, 320 × 192	62, normál tv. monitor, színes, 40 × 24, 320 × 192	66, normál tv. monitor, színes, 40 × 25, 320 × 200	66, normál tv. monitor, színes, 40 × 25, 320 × 200	53, normál tv. monitor, színes, 32 × 16, 256 × 192	53, normál tv. monitor, színes, 32 × 16, 256 × 192	79, normál tv. 40 × 24, 256 × 192	40, fólia tv. 32 × 24, 64 × 48	40, gumi tv. színes 32 × 24, 256 × 176	58, normál mint Spectrum
Kijelző										
Hang	4 szólam, minden szabályozható, 3,5 oktáv	4 szólam, minden szabályozható, 3,5 oktáv	2 szólam, minden szabályozható, 4 oktáv	2 szólam, minden szabályozható, 5 oktáv	1 szólam, minden szabályozható, 5 oktáv, külső	mint 32	3 szólam, minden szabályozható, külső	–	1 szólam, részben szabályozható 10 oktáv	mint Spectrum
Be/ki	2 botkor-mány, spec. magnó, diszk, nyomtató	2 botkor-mány, spec. magnó, nyomtató	2 botkor-mány, spec. magnó, egyéb spec. perifériák	2 botkor-mány, spec. magnó, egyéb spec. perifériák, 2 analóg	2 botkor-mány, spec. magnó, nyomtató, magnó portok	mint 32 és soros nyomtató	nyomtató, 2 hálózati	speciális illesztőn keresztül minden	speciális illesztőn keresztül minden	mint Spectrum
Bővítések	memória, egyéb perifériák	egyéb perifériák	egyéb perifériák	egyéb perifériák	színes monitor	mint 32	mint az előző	mint az előző	mint az előző	
Egyéb	Atari kompatibilis	mint 600	nagyon kis memória	a legelterjedtebb	TRS80 Color kompatibilis	kiváló professzionális	hálózatba építhető	a legolcsóbb	–	–
Ár USA	55 + 30 (spec. magnó)	83 + 30 (spec. magnó)	75 + 30 (spec. magnó + illesztő)	140 + 25 (spec. magnó)	–	100	–	–	100 (CMOS változat)	–
Ár NSZK	183 + 99	298 + 99	185 + 100	548 + 88	250	–	575	99 + 189 (64 k)	298 (48 k) 398	–

\* Nagy felbontású grafikával, ahol az egyáltalán van.



GYÁRTÓ	ATARI CORP.	COMMODORE BUSINESS M.		DRAGON DATA LTD.		MEMOTECH LTD.	SINCLAIR RESEARCH LTD.		
TÍPUS	600XL, 800XL	C16	C64	32	64	MTX500	ZX81	SPECTRUM	SPECTRUM +
BASIC méret (k) Mi nincs?	B	8	8	16	2 × 16	16	8	8	8
Többlet	Sztring tömb, ELSE	–	grafikus, kijelzősformáló, programmodosító utasítások nyomkövető parancs, ELSE	–	–	periféria-kezelő, kijelzés formáló utasítások	speciális BASIC	speciális BASIC	speciális BASIC
Megjegyzés	Képernyőszerkesztés hang és grafikus utasítások hosszú változónevek több változós USR	Képernyőszerkesztés, diszkekezelő, szerkesztőt, programozást segítő utasítások, ON ERROR	Képernyőszerkesztés	Korlátlan számú és méretű mozgatható képterület, LOGO-szerű grafikus utasítások	mint 32 és nagy sebességű adatátvitel	Részleges képernyőszerkesztés, 32 mozgatható képterület, ablaktechnika			
Egyéb	jó, jól használható	jó, jól használható	gyenge	nagyon teljesítőképesség, jól használható	mint 32	teljesítőképesség, nagyon jól használható			
		ROM monitort, asszemblert is tartalmaz		a memória működés közben megjeleníthető	mint 32	ROM monitort, asszemblert is tartalmaz (címké is)			

## 2. táblázat

1. Játéknak – elsősorban a kiemelkedően nagy számban rendelkezésre álló játéckomponensek miatt – leginkább a Spectrum (Spectrum +), a C64, az Atari 800XL (Atari 600XL) gépek valamelyikét ajánlom.

2. Műszaki számítások végzésére, ha azokhoz kevés adatot használunk, és az eredmény is kevés adat, akkor bármelyiket választhatjuk, bár hosszabb élettartamot és a normál billentyűs gépektől várhatunk. Az előbbi megkötések nélkül a C16 igen kis belső memóriája miatt kiesik. A többi használható.

3. Olcsó adatfeldolgozáshoz (ezen lemezegység nélküli feldolgozást értek) csak a megfelelő magnetofon-illesztő, billentyűzetű, memória-méretű gépek használhatók. Így a 2. pontban említettek kivételével kiesik a C64 is, mert a c't című lap mérése szerint átlagos átviteli sebessége 280 bit/s (a Dragoné 1500, illetve 3000), és az átviteli megbízhatósága se megfelelő (a Dragon esetében – különféle közönséges magnetofonnal – csak olyan esetben adódtak átviteli problémák, ha a magnetofon telepe már erősen kimerült). Lényeges előnye a Dragonnak a többihez képest igen gyors BASIC (az ismert teszt szerint). Az MTX500 ilyen adatait nem ismerem.

4. Egyéb adatfeldolgozásra (nagy mennyiségű adat nagy sebességű feldolgozására ez a gépkategória alkalmatlan) csak a Dragon és az MTX500 rendelkezik megfelelő átviteli sebességű (250 kbit/s), áru, tartós üzemre tervezett lemezegységgel.

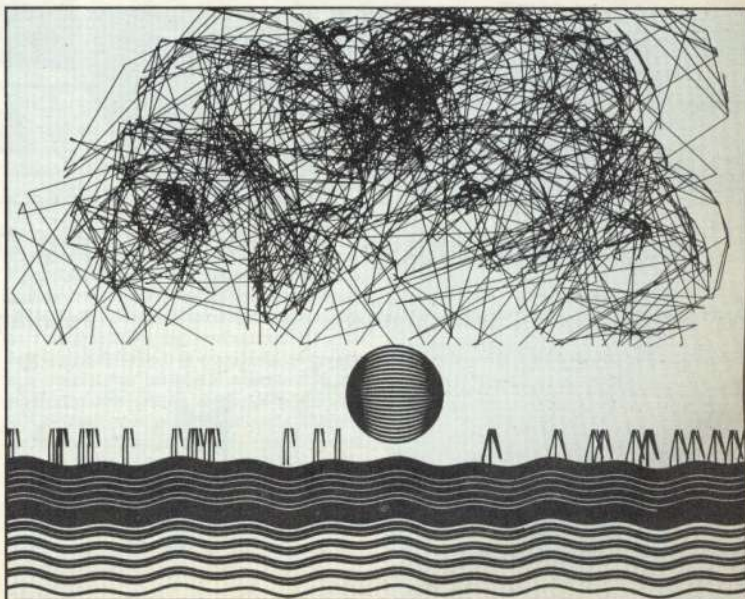
5. Többgépes rendszerek kiépítéséhez egyedül a Dragonnak van megfelelő, UNIX-szerű, OS9 operációs rendszere.

6. Mozgóábrás alkalmazásokhoz bármelyik gép megfelel, de a C16 a rendkívül kis belső memóriája miatt nem tud képeket tárolni. A Dragon viszont különlegesen jó grafikus utasításai segítségével ezt kiemelkedően nagy sebességgel és kis memóriaigénnyel valósítja meg.

7. Oktatási alkalmazás esetén a 6. pontban elmondottakon kívül figyelembe veendő előnye a Dragonnak, hogy BASIC fordítójával felülről csaknem teljesen kompatibilis az IBM PC-fordítója. Így azok, akik ezen a gépen tanulják a BASIC nyelvet, zökkenőmentesen, minimális ráfordítással sajátíthatják el a legelterjedtebb és

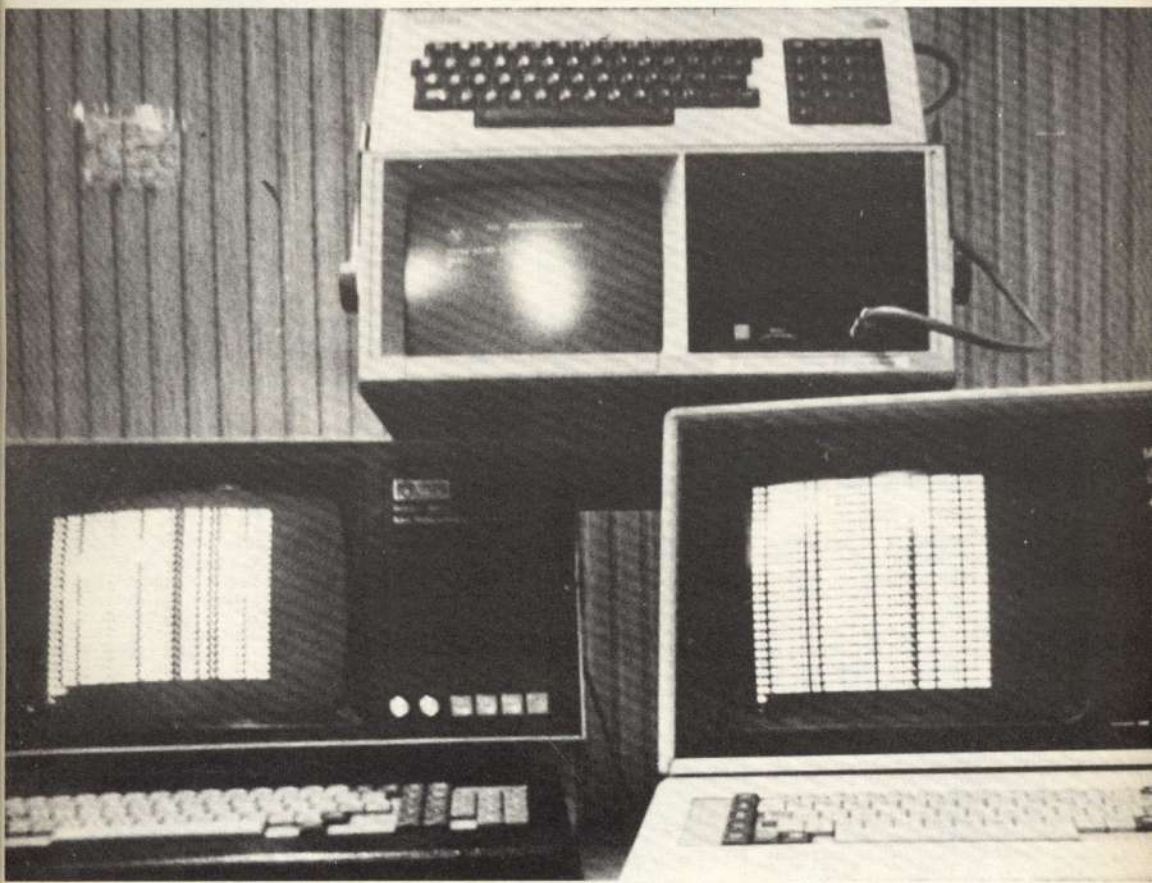
együttal az egyik legteljesítőképesebb BASIC változatot. Valószínűleg a felsorolt előnyei miatt írta a Dragon 32 típusról a világ legnagyobb példányszámú szaklapja, a Byte, hogy megjelenésekor fél évig a legjobb teljesítmény/ár viszonyú gép volt.

DR. SIMONYI ENDRE





# Megvenni drága, bérelni olcsó!



**Először próbálja ki, melyik felel meg Önnek a legjobban!**

**Az alábbi személyi számítógépek  
és GP-IB rendszervezélők  
és nyomtatók azonnal bérelhetők:**

MO8X  
Rolitron, ROSY 80 B  
EMG 666 B  
Hewlett-Packard 9815  
Rohde-Schwarz PUC  
TERTA TMT 120 (mátrix nyomtató)



**Felvilágosítás, előjegyzés, ügyintézés  
420-967, 420-126 telefonon,  
vagy személyesen**

**Címünk:**

**MTA Műszerügyi  
és Méréstechnikai Szolgálat  
Műszerkölcshözvételi Főosztály**

**Budapest, VI., Lenin krt. 67. II. em. 207.**



# Micsoda kiszolgálás!

Úgy adódott, hogy az elemi iskoláit éppen most befejező és rendre jeles bizonyítványt produkáló kisdidiákat is reménytelenül megfertőzte a számítógép-betegség. S mert egy tanfolyam elvégzése után elemi fokon a BASIC nyelvet is „beszélte”, meg kötegszámra gyártotta papíron a különböző programokat, turistákat Münchenben járva, az ottani kínálatot – no meg az árakat – látva, elcsábult a család, s megszületett a döntés: jó, legyen számítógép. De milyen, melyik a sok tucatnyi kínálatból, és persze a pénztárcához is keményen igazodva.

Ottani barátaink tanácsára a Pro Markt discont áruházban kötöttünk ki, feltételezve – és nem alaptalanul –, hogy a discont eleve érdeklőzőket jelent, és ha egyszer számítógépekre szakosodott cégről van szó, nyilván kapunk némi szakmai eligazítást is.

A szóban forgó áruház München egyik csúnyácska negyedének legszélén található. Pontos címét csak azért nem adom közre, mert végtére is a sorok írására nem reklámozónökként békelt fel, ám a majd harmegyedre óraig tartó vásárlás részletei talán nem érdekeltek.

Kezdődött az ügylet egy telefonbeszélgetéssel, érdeklődően, hogy ilyen és ilyen – hadd tegyem hozzá: meglehetősen alacsony – árhatár között kapható-e náluk valami viszonylag jól használható készülék. A rendkívül udvarias igenlő válasz után máris szívélyes érdeklődés: ha elárulnám, hogy kinek s mi célból kellene a készülék, akkor még pontosabb felvilágosítással szolgálhatnának, esetleg azon nyomban konkrét javaslatok is lennének, sőt – döntésemet megkönnyítendő – még aznap postáznák címre a cég teljes árukínálatát tartalmazó katalógust, természetesen ingyen.

Már ez is lefegyverző volt, de maradtunk a személyes találkozásban, minek kapcsán finoman megjegyezték: láthatóan számítógép ügyben még járatosokkal állnak szemben, ezért a vásárlási tanácsadás kissé elhúzódhat. Ha javasolhatnák, inkább a reggeli vagy a késő délutáni órákban keressük fel őket, akkor ugyanis nyugodtabbak a körülmények.

Hallgatva a tanácsra, a javasolt időpontok egyikében kerestük fel a céget. A vásárlási procedúra azzal kezdődött, hogy bemutatták az általunk megjelölt árfejkévszű gépek választékát. Csak úgy, szemre. Nem feledeztek meg arról, hogy például a formatervezési különbségekre és az ebből adódó praktikumokra is felhívják a figyelmünket, sőt arra is, hogy melyik típus milyen színválasztékban kapható. Majd következett egy skicc-rajzos magyarázat arról, hogy melyik berendezéshez milyen tartozékok, kiegészítő és kapacitásbővítő elemek vásárolhatók, és miért vásárlandók, aztán a különböző gépek

előnyeiről és esetleges hátrányairól szóló rövid értekezés zárta a kiselőadást.

Eközben az is kiderült, hogy a ki tudja melyik gép igazi tulajdonosa és használója egy kisdidi lesz, akivel a cég egy másik alkalmazottja külön is elbeszélgetett – no persze alkalmi tolmács segítségével –, szabályos kis vizsgáztatást rendezve. Szóba került, hogy mit tud a kisdidi a BASIC nyelvből, milyen a programírási gyakorlata, és milyenek az ezzel kapcsolatos elképzelései. A reményteljes tulajdonos különböző gépeken maga is kipróbálhatta a tesztkérdéseket, villámgyorsan megtanították egyszerű programok szerkesztésére, kíváncsian arra is, hogy a korábbi beszélgetésből tanultakat vajon hasznosítja-e az éppen szóban forgó program lehetséges továbbfejlesztésénél. S miközben e sajátosságot vizsgáztatták vagy hűs perccel is eltartott, szinte már kényszeredetten kérdeztem az eladót, hogy hát végül is a sok ajánlat közül akkor most melyik mellett döntünk? Mire a válasz: „Uram, pontosan ebben szeretnénk Önöknek segíteni. Szakértőnk azonnal közli az ajánlati listát.”

Közölte. Nem az eladó, hanem a cég szakértője. És indoklással hozzátette, hogy megfigyelése szerint a gyerekek jelenleg és néhány évig ez és ez a típus felel meg leginkább, amely egyébként idővel ilyen és ilyen módon bővíthető, gazdagítható. Természetesen a jelenleginél olcsóbban, mert az általa ajánlott berendezés ára tavaly még annyi volt, és az idén már csak ennyi, amúgy pedig ajánlatánál azt is figyelembe vette, hogy vadonatúj típusról van szó, tehát nem fenyeget a veszély, hogy néhány év múlva esetleg ismét hozzájuk látogatva és a kiegészítő berendezések után érdeklődve, sajnálkozva tárják szét a karjukat. Viszont már most felhívta a fiatalember figyelmét néhány érdekes és sokoldalúan használható játépprogramra, s amennyiben érdekel bennünket, máris bemutatja ezeket.

Érdekl, bemutatta, hármát kiválasztottunk, és az egyike rögvest 15 százalékos árengedményt adott az eladó. Következett a számlázás, majd megkaptuk a cég árukatalógusát, a velünk foglalkozó eladót és a szakértő névjegyet, és a tanácsalban belépéstől számított jó háromnegyed óra múltán, a lehetőségekhez képest kiokosítva és némi sajnálkozás kíséretében távoztunk az áruházból. Sajnálkoztak ugyanis, hogy csak német nyelvű kézikönyvvel szolgálhatnak, bár nekünk angola vagy franciára lett volna szükségünk, és sajnálkoztak, hogy nem tudják külön is csomagolni a vásárolt holmikat, de hát ők csak egy discont áruház...

## ADOK – VESZÉK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikro-számítógépekkel kapcsolatos hírdetéseket közlünk. A díjszabás: közületeknek gépelé sor 20,- Ft. Az NJSZT tagjainak az első három sor ingyenes. Hírdetéseiket a szerkesztőség címére várjuk.

- ZX-SPECTRUM 48 K-S, új, olcsón eladó. Páll József, Újszász, Mátyas út 9. 5052
- ZX-SPECTRUMHOZ irt játépprogramot cserélnek. Kálmán Imre, Agárd, Sándor u. 1.
- CSEREAJANLATI ZX-SPECTRUM és C64 programok bő választékban. Berdálóvics Géza, 54327 Bilje, Jugoszlávia
- ZX-SPECTRUM EXTRÁKKAL szerelve (video-kimenet, hardver reset), Kempston interfésszel, továbbá 400 darabból álló felhasználói és játépprogram-gyűjteménnyel eladó. Honfi József, Csákvár, Május 1. u. 11. 8083. Telefon: 06-24-44-307 (7-16 óra között)
- ATARI 800 XL géppem van. Szeretnék Atari-tulajdonosokkal kapcsolatba lépni. Klampéckzi Gábor, Elek, Belsőlatiba u. 26. 5742
- COMMODORE 64-RE készült programokat, programleírásokat cserélek. Szabó Zoltán, Pp. Szőlő u. 35. 1034 Telefon: 883-761 este.
- VC-20 MAGNÓVÉL, gépkönyvvel, programokkal 20 000 forintért eladó. Kovács Gábor, Sellye, Ifjússág u. 41. 7960
- SINCLAIR QJ számítógépre professzionális és saját készítésű programokat adok, veszek, cserélek. A géppel kapcsolatos egyéb információk is érdekelnek. Szoftverfejlesztés és memóriabővítésben is tudok segíteni. Várhegyi Tamás, Debrecen, Tessedik Sámuel u. 118-120.
- A DOKIBER orvosi és biológiai szoftverfejlesztést vállal Commodore, Sinclair és IBM kompatibilis gépeken. Jelenleg több mint húsz programot tudunk ajánlani. Debreceni Orvosgyógyászati Kórház, DOTE KISZ Bizottság, Debrecen, Móricz Zsigmond krt. 22.
- VC-20 tartozékokkal eladó. Irányár: 55 000,- Ft. Zsigori Csaba, Bp. Varga Gyula András park 4/b. I. 9. 1149.
- AJÁNDÉKOZZON AZ INTERBIT GMK-TOL hobbi-, játék- és segédprogramokkázzattak! Új, saját fejlesztésű programok a karácsonyi vásárra. Olcsó áruk! Géptípusok: ZX81, ZX-Spectrum, C64. Cím: 1631 Bp., Pf. 11. Telefon: 856-028.
- COMMODORE C64 hardver-leírás, könyvet vásárolnék. Kozma Miklós, Oroszlányi Banki D. út 10. 11/2.
- AMSTRAD (SCHNEIDER) számítógéptulajdonos partnert keres. 1033 Budapest, Bogdani út 12/A. 11/2. Lengyel Tamás.
- RÁDIÓAMATŐR PROGRAMOK C64-re: morzeotatók, gyorsítávrész, automata adó, Q50 nyílvántartó, RH 11/2, versenyző, jegyzőkönyvíró, adó-vevő interféce. Herczeg József, 6723 Szeged, József A. sgt. 69/A.
- TA ALPHETRONIC PC professzionális, CP/M lemez és teljes dokumentációi bérbeadó, vagy eladó, 9023 Győr, Zombor u. 11. Tel.: 14-155 Réz Gyula.
- COMMODORE 64 típusú személyi számítógéphez megőzadásigis üzemi igényeket kielégítő programokat vásárolunk. Éjeér megyei TKSZOV Gazdaságpolitikai titkársága (8000 Székesfehérvár, József A. u. 55. III. em. Telefon: 22-14-055).

-tes





## **BÉRSZÁMFEJTÉS-ADATGYŰJTÉS PROGRAMCSOMAG CBM 64-re**

**A programcsomag alkalmas a bérszámfejtési munka gépiesen, havonta ismétlődő számítási, gyűjtési feladatainak megkönnyítésére. Tartalmazza és kezeli a jutalmakat, a visszamenőleges béremeléseket, elkészíti a kifizetendő összegek címletjegyzékét és a különféle bérnövelő vagy -csökkentő tételek gyűjtését munkaviszony-kódonként és munkaszámonként.**

***A rendszer programjai:***

- főmenü***
- törzskarbantartó***
- bérszámfejtés***
- fizetésijegyzék-listázó***
- címletező***
- gyűjtő***
- jutalomszámfejtő***
- jutalomlistázó***
- visszamenőleges fizetésemelés-elszámoló program.***

**A programcsomag legfeljebb 850 dolgozó adatainak kezelésére alkalmas.**



**Alkotó Ifjúság Egyesülés**  
**Számítástechnikai Iroda**  
**Budapest V., Garibaldi u. 2.**  
**Levélcím: 1519 Budapest, Pf. 330.**  
**Telefon: 112-666, 113-608**  
**Telex: 22-7272**

## **KÉSEDELMIKAMAT-SZÁMÍTÓ PROGRAM CBM 64-re**

A program alkalmas a késedelmes pénzügyi átutalások kamatának kiszámítására és egyben a kamatfizetésre felszólító levél elkészítésére és kinyomtatására.

### **Hardverigény:**

- 1 db CBM 64 alapgép**
- 1 db floppy-egység vagy kazettás egység**
- 1 db nyomtató**

## **GÉPJÁRMŰNYILVÁNTARTÓ ÉS JÁRULÉKSZÁMITÓ PROGRAM CBM 64-re**

A program alkalmas az 50/1981. (X. 31.) és a 85/1982. (XII. 4.) PM sz. rendelet alapján 20 termelőegységre 300 gépjármű adatainak tárolására, a járulék negyedévi és év végi kiszámítására és kinyomtatására.

### **Hardverigény:**

- 1 db CBM 64 alapgép**
- 1 db floppy-egység**
- 1 db nyomtató**



## Építsünk számítógépet! XI.

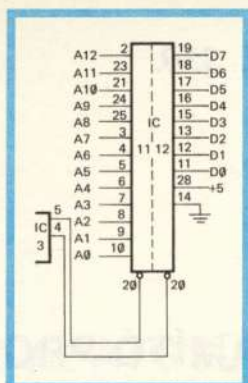
A kártyán két 28 lábú ROM tok részére van hely. Jelenlegi kialakítása esetén ezek 2764 típusúak (IC11, 12). A ROM-ok az adatforgalomban a mikroprocesszorral állnak közvetlen kapcsolatban. Megcímzésük a már említett IC3 4., illetve 5. lába segítségével történik. A ROM-okban tárolt adatok címének kiválasztása az alsó 13 címvonallal (A0–A12) történik. Az 5 V tápfeszültséget a  $V_{CC}$  (28. lábra) vonalra, a földet a 14. lábra vezetjük, amint az 1. ábrán látható.

A belső RAM-ok részére 8 db 24 lábú tokot képeztünk ki. A jelenlegi megoldásnál ezek 6116 típusúak. Az IC13–20 adataira ugyanaz vonatkozik, amit a ROM-oknál elmondtam. A kiválasztás itt sokkal bonyolultabb. Az IC3 6., illetve 7. lába és az A11 és A12 címvonalak egy-egy kettőről négyre dekódoló (74LS139) IC (IC21–22) segítségével dolgoznak a RAM IC-k (IC1–12) lábaira. A belső címzésnél itt a két felső cím vonal a kiválasztást szolgálja, így itt márnincs szerepe. Az írás-olvasás kiválasztása az R/W vezetékéről jövő jellel (21. lábra) és az IC1 által invertált jellel (20. lábra) történik. Mindez a 2. ábrán látható.

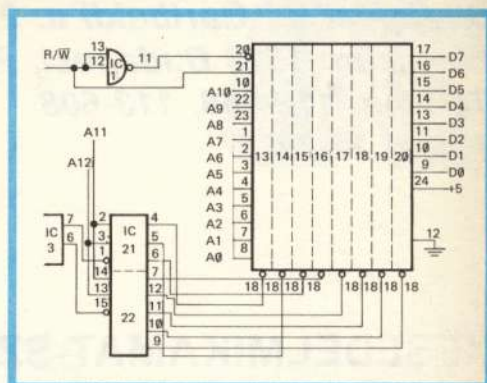
A párhuzamos csatorna 8 jelvezetékét a mikroprocesszorral két meghajtó (7407) IC (IC6,9) köti össze. A két alkatrész bekötésétől függően be- vagy kimenetként használható. Jelenleg ez utóbbit használjuk. A csatorna 10. lábára +5 V, 1. lábára a föld kerül. A leiratakat a 3. ábrán mutatom be.

Az általán megépített változatnál a billentyűzet külön kártyán keresztül csatlakozik a központi egységhez. Amennyiben a billentyűzet közvetlenül kapcsolódik a mikroprocesszorhoz, úgy a 3. ábrán látható két IC oldalait felszerelve kell bekötni.

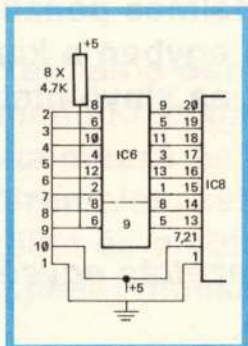
A kártyán levő utolsó funkcionális egység a magnetofon-illesztő. Ez két önálló részegekből, az ír- és az olvasóegységből áll. Az írőegység a már említett soros vonalat kiszolgáló jel-sorozatot egy meghajtóval leválasztja (az IC9



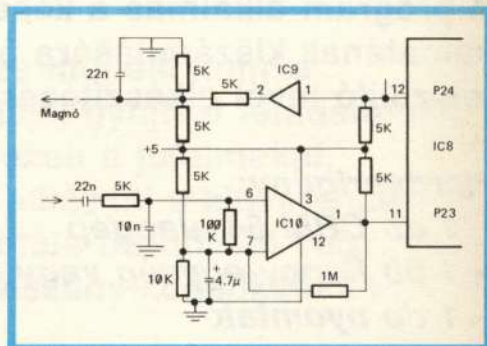
1. ábra



2. ábra



3. ábra



4. ábra

egy további része). A négyzögjeleket lekerítjük, és így vezetjük a magnetofonhoz. Az olvasóegység első része a torzult és lekerekített jeleket négyzögjelekké alakítja. Ezeket a jeleket egy összehasonlítóval (IC10, LM339) 0 és 1 jelekké alakítjuk, amelyeket a mikroprocesszor P23 soros bemenő vonalára vezetjük. Az egység rajzát a 4. ábra mutatja, a működés pedig a  $\mu M$  1985/3. számának 30. oldalán, a Kézírtaszabvány című cikkben látható.

E kártya ismertetésével véget ért a gép alapkiépítésének bemutatása. A tápegység és a doboz nem speciális, tehát a követelményeknek megfelelően elkészíthető vagy beszerezhető.

A cikksorozat ezzel befejeződött. A további hardver és a szoftver ismertetését csak akkor tartom célszerűnek, ha már jelentős számú gép elkészült. Remélem, hogy ez hamarosan bekövetkezik.

DR. SIMONYI ENDRE

## ADATMÁSOLÁST ÉS ADATARCHIVÁLÁST VÁLLALUNK

mágnesszalagról Commodore 64-es floppy discettre vagy floppy discettről mágnesszalagra.

A discetten létre tudunk hozni • soros, • relatív és • random fájlokat

Vállalunk továbbá számítógépes rendszerekre adatrögzítést.

A vidékiek részére a postán elküldött adathordozókról is elvégezzük a szolgáltatásokat.

**RODATA**  
Leányvállalat

1028 Budapest, Pincester u. 14–16.

Telefon: 164–680

Ügyintéző: Sipos Géza



## Ki ad magyarázatot?

Lapunk 1985/3. számában egy *Primo*-problémára kértünk választ. A beérkezettek közül *Kutassy Jenő* választát ábrák és a szerző címének hiányában nem tudjuk közölni. Közreadjuk viszont *Jakó János* magyarázatát:

A fényképen jól látható, hogy a karakterek torzán jelennek meg a képernyőn, mert utolsó öt bájtkuk hiányzik. Használó hatást érünk el, ha CLS után CTR+P-t nyomunk le, majd a RETURN billentyűvel „lejövünk” a képernyő aljára, ahol a kurzor eltűnik. Ekkor LIST után is csonkák lesznek a karakterek.

A fényképen látható program futása utáni hiba a képernyőre írás pontatlansága miatt történt. Ugyanis e programot futtatva a gép a CHR\$(15) hatására átvált, és függőlegesen ír. Ha ekkor állítjuk meg a programot, a betűkből még kevesebb, csak az első 3 bájtk látunk. Ezt a hatást mérsékli a CHR\$(23) – a 15. kód inverze – úgy, hogy még 3 bájtk látunk a karakterekből. Ugyanis ezeknél az írásmódváltásoknál a kurzor nem pontosan illeszkedik, nem az „elméleti” helyén van, hanem egy kicsit arrébb, de ez a táv elég ahhoz, hogy a hiba bekövetkezzen. A hatást tehát nem valamilyen belső hiba, és nem a CHR\$(16)–CHR\$(17) – alsó index, felső index – okozza, hanem csak egy apró pontatlanság.

Az 1984/6. számban A PRIMO című írásban megjelent l. képhez annyit lehet hozzáfűzni, hogy az valóban egyedi hiba: a sorszám hiányzik, mert SHIFT-tel volt nyomva az 1 és a 0 billentyű. A 4. képpel kapcsolatban az a megjegyzésem, hogy az elől elhelyezkedő 16 k ROM után található 60–70 bájtk igen fontos a jó működéshez, ezeket csak indokolt esetben szabad átírni, mert a gép „kikészülhet” tőle – mint az ábrán látható program esetében –, és ki kell kapcsolni, mert „elszáll”.

Azt hiszem, ennyi magyarázat elég. Ezek után leirom néhány, a Primóval kapcsolatos felfedezésemet.

Az én gépem A-48 típusú Primo. Elég sok bosszúságot okozott, hogy nem tudtam, hol kezdődik a képernyő-memória, hiszen az A-32 gépeknél 6800H ez a kezdőcím, és az arra irt gépi scroll nem akart futni az én masinámon. Nemrég azonban sikerült rájövniem, hogy A800H a kezdőcím, ami azért furcsa, mert a PEEK-POKE utasításpár nem használható a képernyő kapcsán.

A billentyűzet érzékenységét a 16454-es címen lehet beállítani. Ha ide 1-et „pökölnék”, akkor nagyon gyors lesz – ez jó a játékprogramoknál –, ha pedig például 0-át, akkor borzasztóan lelassul. Rövid kísérletezés után rájöttem, hogy kb. 5–6 érzékenységi fokozat van.

A Primo képes önmagát ki-csapcsolni, alapállapotba hozni. Ezt akkor tapasztaltam, amikor a következő parancsot írtam be: A=CALL(0). Ugyanez a hatása annak is, ha gépi kódban valamit elrontok, vagy ha a fentebb említett fontos címeiket sorban átírogatom.

A 16456–16457 címenek a kurzor helye van, de be is állíthatunk egy másik értéket. Az első címen a vízszintes, a másodikon a függőleges helyzethez tartozó érték található. A 0, 0 pozíció a képernyő bal felső sarkában van. Próbáljuk ki ezt a programot:

```
5 CLS
10 POKE 16456,200,100:
PRINT"OK"
```

Segítségével grafikus felbontásban lehet írni a képernyőre.

A cikkben az 1. képre adott „magyarázat” téves, én ugyanis nem SHIFT 1 és SHIFT 0 karaktereket billentyűztem be, hanem 1 és 0 karaktereket, az átváltást a gép csinálta.

Újabb feladat egy Commodore 64 furcsaság. Amikor az 1985/1. szám 32. oldalán lévő programot lemezről képernyőre vittem, ott END, kinyomtatva RETURN lett a 1130, 1140 sorokban. A két kulcszókód 80H, ill. 8EH. A használt nyomtatás MPS 801.

S.E.

- Ha rendelkezik VT 20/A-val,
- ha van szabad géptideje,
- ha használnak Önöknél COMMODORE 64-et,

akkor

## KÖSSZE ÖSSZE őket!

A VT 20/A számítógép fejlett adatfeldolgozó rendszerré alakítható COMMODORE 64-ek rákapcsolásával.

A COMCON programcsomag alkalmazásával a VT 20/A max. 8 önálló terminált nyer.

## A COMMODORE 64

rendelkezésére áll:

- egy BASIC-ből elérhető
- hálós-hierarchikus adatbáziskezelő rendszer
- nagy háttérkapacitással (a C 64 10 MB-on is dolgozhat)
- nagy megbízhatósággal

A munkahelyek egyidőben több adatbázist is használhatnak.



**SZÁMÍTÁSTECHNIKAI  
MŰSZAKI FEJLESZTŐ  
KISSZÖVETKEZET**

Budapest XIV., Köszeg u. 4.

Telefon: 831-805

Ügyintéző: Fogarassy Károly





# FOTOELEKTRONIK ISZ

Tisztelt Leendő Ügyfelünk!

Örömmel értesítjük, hogy szövetkezetünk  
Budapest XIII., Rajk László u. 46/b. szám alatt  
műszaki cikkek adásvételével foglalkozó

## SZÁMÍTÁSTECHNIKAI SZAKÜZLETET

nyitott.

Újonnan megnyílt szaküzletünkben széles körű  
hardver- és szoftver-választékkal várjuk  
ügyfeleinket. Raktárról, vagy igen rövid átfutási  
idővel tudjuk biztosítani az alábbi  
számítástechnikai eszközöket:

Commodore 64 konfiguráció  
Commodore 610 konfiguráció  
Commodore 720 konfiguráció  
Commodore PC 10 konfiguráció  
Commodore PC 20 konfiguráció  
IBM PC AT konfiguráció  
IBM PC XT konfiguráció  
Sinclair Spectrum konfiguráció  
Sinclair QL konfiguráció  
Apple II. konfiguráció

Epson, C. Itoh, Seikosha, Hewlett Packard  
printerek, plotterek BASF, Mitsubishi, AMPEX  
Winchesterek, floppyk.

Fentiekén kívül egyedi hardver, szoftver, műszer,  
mérés- és automatizálás-technikai  
berendezésekre vonatkozó igényét is rövid  
határidőn belül kielégítjük.

Önnek elegendő hozzánk elfáradnia, számítás-  
technikai, illetve mérés- és automatizálás-techni-  
kai problémáját felvázolnia. Biztosíthatjuk, hogy  
számítástechnikai problémái mind hardver, mind  
szoftver tekintetében a legmegfelelőbb kezekbe  
kerülnek, és anyagi lehetőségéhez mérten azok  
maximálisan megoldásra kerülnek.  
Hardver és szoftver leasinggel is foglalkozunk.

Maradunk tisztelettel:



Számítástechnikai Szaküzlet

1136 Budapest  
Rajk László u. 46/b.



## ZX81-re

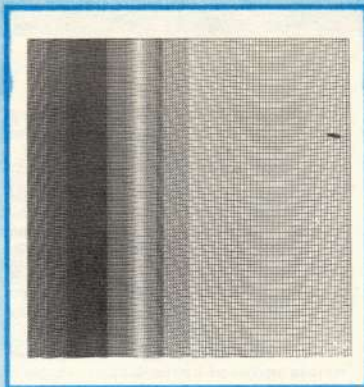
### A RÓKA

### ÉS A LIBÁK

A játék sakktablán folyik. Sem a róka, sem a libák nem ugorhatnak. Ebben a játékban nem veszik le a figurákat a tábláról. A róka egy lépésben egy mezőt haladhat előre vagy hátra, a libák csak előre mehetnek, ferden.

A játék célja a róka számára: szétszakítani a libák vonalát. Ha ez sikerül neki, megnyeri a játékot. A libák akkor győznek, ha a rókát úgy be tudják szorítani valamelyik mezőbe, hogy nem tud lépni.

NYITRAI LÁSZLÓ



## Commodore 64

### JÁTÉK-

### TRÜKKÖK

A számítógép remek játékpártner is lehet. Hosszú órákat tölthetünk játékprogramok előtt ülve, de sokszor előfordul, hogy bosszankodunk, és átkozzuk a program íróját, aki csak néhány „életet” adott ahhoz, hogy az egyre nehezebb feladatokat megoldjuk. Ilyenkor rendszerint újabb és újabb erőfeszítésekkel próbálunk apránként túljutni az akadályokon, de próbálkozásaink sokszor sikertelenek.

Sokat kell gyakorolni, míg kiismerjük alkalmi elcsépségeink mozgását, megtaláljuk a helyes megoldást. Néha órákig kísérletezünk különböző eredmény nélkül.

Létezik azonban a gyakorlásán kívül egy másfajta, célravezetőbb megoldás is. Mivel a játékok csak egy program, saját igényeink és akarunk szerint változtathatunk rajta. Csak a megfelelő POKE utasítás címét kell ismernünk, és azt, hogy milyen értéket kell beadnunk ahhoz, hogy a játék számunkra könnyebb legyen.

A játék neve	Beadandó	Hatás
Tír Na Nog	POKE 34202,200	könnyebb teszi a játékot
Manic Miner	POKE 16571,234	örökélet
Falcon Patrol	POKE 16754,234	örökélet
Attack of the Mutant Camels	POKE 11639,255 majd SYS 4096	örökélet
Skramble (Ani-rog)	POKE 8609,173	örökélet
Hover Bovver	POKE 35680,96	örökélet
Gridrunner	POKE 35869,173	örökélet

(Kopexy)

## VIC-20-ra

### SZÓPÓKER

A közismert játékprogram adaptációja. A 280-as sor cseréjével új feladatokat tűzhetünk ki. Ha ezek száma nem 7, akkor a 10-es sort is módosítani kell.

MAROSVÁRI SÁNDOR

```

10 PRINT "VIC-20: N=0: IF Z>7
    THEN 300
15 PRINT "M SZOPOKER"
20 READ P
25 ILEN(A$) <> 5 THEN 10
30 PRINT "KEZDHETTED!"
40 INPUT B$
42 IF LEN(B$) = 5 THEN 50
45 PRINT "*****" : "NEM TUDSZ 5-10"
46 PRINTSPC(8) : "SZAMOLNI?"
   : GOSUB 190 : GOT0 40
50 FOR I=1 TO 5
60 H$=MID$(A$,I,1)
70 K$=MID$(B$,I,1)
80 IF H$=K$ THEN N=N+1
90 NEXT I
100 PRINT "*****" : N
120 N=0 : GOT0 40
130 POKE 36879,15
140 FORA=170 TO 255 STEP 2
150 POKE 36875,A
160 NEXTA : POKE 36878,15
170 POKE 36875,0
180 RETURN
190 POKE 36877,220
200 FOR L=15 TO 0 STEP -1
210 POKE 36878,L
220 FORM=L TO 100
230 NEXT M
240 NEXT L
250 POKE 36877,0
260 POKE 36878,0
270 RETURN
280 DATA KOMOR,ABLAK, MISKA, IKLAD,
    VÁRTA,PETTY,CSENG
300 PRINT "V: TAB(11)
    "A KESZLET KIMERULT."
310 PRINT TAB(47) : "VIZONTLATASRA!"
320 GET I$: IF I$="" THEN 320
330 LIST 280
    
```



### Program(ön)kritika

A ZX81 RAM-ja kezdődik a 16509-es címen.

Az 1985/2. számunkban a FOLYOSÓ című program 70-es sorában a ; lemaradt.

Az EGY UTASÍTÁSSAL programnak szintén a 70-es sora hibás, mert ; helyett + kell hogy álljon.

Ugyanebben a számban az autóverseny nem hibás ugyan, de nem fér el 1 k-ban.

Visszatérve egy régebbi számunkra: az 1984/5. szám 6. oldalán a 110-es sorból kimaradt egy OR. A sor helyesen: 110 IF I <> 1 OR I > 8 OR J < 1 OR J > 8 THEN GOT0 70

```

5 REM **NYITRAI LASZLO 1985**
10 CLS
20 PRINT AT 1,5;"ROKA ES M
30
40 REM A ROKA -SZAMITOGEP-
40 PRINT AT 14,10;"
: AT 5,0;
50 PRINT TAB(10);"
60 PRINT TAB(10);"
AB 10
65 NEXT I
80 LET P=PEEK 16396+256*PEEK 1
6397+256*PEEK 16398
90 LET H=P/3
92 POKE H,15
94 PRINT AT 5,21;"LEPESIRANY"
: AT 5,24;"V: TAB 25;" : AT 11,2
: "A: TAB 25;" : VALTAB;" : AT 14,21;"
: "K: KIMENES"
110 REM -A LIBAK LEPESEI-
120 CLS
130 FOR I=1 TO 4
140 LET A(I)=P/7*33+2*I-2
150 POKE A(I),CODE "L"
160 NEXT I
165 LET L=H
180 FOR I=1 TO 4
190 IF PEEK (A(I)-3)=128 AND PE
ER (A(I)+3)=128 AND PEEK (A(I)-
34)=128 THEN LET L=A(I)-1
200 IF PEEK (A(I)+2)=128 AND PE
EK (A(I)-34)=128 AND PEEK (A(I)-
32)=128 THEN LET L=A(I)+1
210 NEXT I
220 LET R=33+SGN (L-H-33*(INT (
(L-P)/33)-INT ((H-P)/33)))
230 IF R=33 THEN LET R=32+2*(PE
EK (H+34)=128)
240 IF PEEK (H+R) <> 128 THEN LET
R=R-56
250 IF PEEK (H+R) <> 128 THEN LET
R=R
260 IF PEEK (H+R) <> 128 THEN LET
R=R-56
270 IF PEEK (H+R) <> 128 THEN GOT
0 410
280 POKE H,128
290 LET H=H+R
300 POKE H,CODE "R"
310 IF H>P+230 THEN GOT0 406
320 FOR I=1 TO 1
340 POKE A(I),364-PEEK A(I)
350 LET IS=INKEY$
360 IF I$="5" OR I$="8" THEN GO
TO 400
365 IF I$="K" THEN GOT0 406
370 IF I$="V" THEN GOT0 340
380 POKE A(I),CODE "L"
390 NEXT I
395 GOT0 330
400 LET R=34+2*(I$="8")
405 IF PEEK (A(I)+R) <> 128 THEN
GOTO 340
402 POKE A(I),128
403 LET A(I)=A(I)+R
404 POKE A(I),CODE "R"
405 GOT0 185
406 PRINT AT 16,5;"EN GYZOZTEH..
407 PAUSE 4E4
408 GOT0 2
410 PRINT AT 16,0;"MEGFDGTUK A
ROKAT"
420 PAUSE 4E4
421 GOT0 10
430 SAVE "ROKA ES A LIBAK"
    
```



LEPESIRANY  
V: VÁLTÁS  
K: KIMENES

MEGFDGTUK A ROKAT..



Fórum rovatunkban olyan rövid olvasói észrevételeket, közléseket és állásfoglalásokat adunk közre, amelyek érdekes egyéni véleményeket, tapasztalatokat tükröznek, vagy pedig egy viszonylag szűkebb körben használt géptípusra és ezzel kapcsolatos technikai eszközökre kívánják felhívni a figyelmet. Továbbra is várjuk ilyen tárgyú leveleiket.

## Schneider CPC 464

A hifi-berendezéseiről ismert Schneider cég 1984 szeptemberétől számítógépet is kínál. Az új gépek minden adottsága megvan ahhoz, hogy konkurenciát komolyan sarokba szorítsa. Két változatban kerül piacra: színes és zöld monitorral. A monitor azért lényeges, mert szerves részét képezi a gépnek. Ebben található ugyanis a számítógép áramellátását szolgáló elektronika is.

Maga a számítógép a billentyűzet alá van építve, amely két rövid kábellel csatlakozik a monitorhoz. A központi egység három részből áll: ASCII, illetve DIN billentyűzet (kivánság szerint), miniatűr blokk és kurzor-billentyűk, valamint egy beépített kazettás magnó. A billentyűzet minősége kitűnő, hasonló a C64-éhez. A magnó is említésre méltó. A motor ki-be kapcsolását a számítógép vezérli. A magnónak két felvételi sebessége van: 1000 és 2000 Baud. Lejátszáskor automatikusan vált a kettő között. Ezzel a megoldással sok bosszúság okát sikerült megszüntetni.

A készülék lelke egy Z80A mikroprocesszor, amely 4 MHz-cel fut. Az alappéphe 32 kbájt ROM és 64 kbájt RAM be van építve. A 64 kbájtól 16 a képernyő részére foglalt, a BASIC programoknak 42 kbájt marad szabadon, ami nagyon jónak mondható (a C64-nél 39 kbájt).

A képet három különböző felbontásban lehet megjeleníteni: 160 x 200 16 színben, 320 x 200 4 színben és 640 x 200 2 színben, ami megfelel az IBM PC grafikájának. A zöld monitoron színek helyett szürke fokozatok lehet látni. A színeket 27 színt tartalmazó palettáról választhatja a felhasználó.

Szöveget is három különböző módon kaphatunk: a 25 sorban 20, 40 és 80 betűs megjelenítés lehetséges. A 25 x 80-as módus már professzionális szövegfeldolgozásra, illetve programozásra is alkalmas.

Újdonság a CPC 464 zenei képessége is. A gépnek 4 hanggenerátora van; az egyik háttérzajt ad, a másik három szabadon programozható. 7 oktávot „tud”, és fejjelhangot vagy erősítőn keresztül a zene sztereóban is élvezhető.

A gép BASIC-je ugyancsak jól sikerült. Egy

angol cég, a Locomotive Software írta, és a megszokott Microsoft BASIC-hez képest jó néhány érdekes és hasznos tulajdonsággal rendelkezik.

A megszokott FOR-NEXT konstrukció mellett használható a más programnyelvekből ismert WHILE-WHEND is. WAIT utasítással megállítható a program futása, míg valamelyik bemenetiről adatot kap. AFTER x,y GOSUB 100 utasítással időköz kötiött programmegszakítás érhető el. Az időt 10 ms pontossággal lehet megadni. Három független óra áll rendelkezésre. Az EVERY utasítással a megszakítás után az óra nullázható. Ezekhez tartoznak a DI (disable interrupt) és EI (enable interrupt) utasítások is, amelyekkel a megszakításokat lehet ki-be kapcsolni. A többi BASIC utasítás, mint a TRON, TROFF, Print USING, LOWERS, INSTR, ON ERROR stb. már ismertebbek.

Újdonság viszont, hogy a képernyőt „ablakokra” lehet osztani. Minden egyes ablak úgy viselkedik, mint egy önálló kis display. Maximum 8 részre osztható a képernyő, ehhez 7 szöveges és egy grafikus lehet. A betűk 8 x 8-as mátrixban jelennek meg.

256 betű és jel között lehet válogatni, de saját betűkészletet is generálhat a programozó. Így például használhatóak a magyar ékezetes betűk vagy a görög abc is.

A sok jó ötlet közé két gyengébb is becsúszott: az editor csak soronként működik, ami manapság már elavultnak számít, továbbá hiányzik a C64-ről ismert sprite-ok lehetősége.

Ennek ellenére a BASIC nem rosszabb, sőt több tekintetben jobb, mint a C64 BASIC, a Simon's Basicet is beleértve. Mindezek mellett még érezhetően gyorsabb is.

A CPC 464 nagyon jól kiépíthető. Az alappéphez van egy botkormány-csatlakozó, egy interfész a Centronics nyomtatóhoz és egy USER Port. Ez utóbbihoz lehet a későbbiekben V24 adaptert, ROM-modulokat, RAM-modult (9 Mbájtig!) és floppykat csatlakoztatni.

A Schneider floppy 3" méretű és CP/M operációs rendszerrel fut. Az árban a CP/M és egy LOGO interpreter benne van. A CP/M rendszerrel hozzáférhetővé válik egy olcsó és hatalmas terjedelmű szoftvervilág.

A Schneider cég 1984-ben 40 ezer CPC 464-et akart eladni, 1985-re pedig 100-200 ezer darabot tervezett. Ezek a számok realitások is, ha az árakat nézzük. A CPC 464 ára zöld monitorral 900 nyugatnémet márka, színes monitorral 1400 márka. A hajlékonylemez CP/M-mel és LOGO-val 850 márka kerül.

Összehasonlítva az árat és a teljesítményt az Atari-Commodore súlycsoportba tartozó számítógépekével, elmondhatjuk, hogy meglepően olcsó, viszonylag nagy teljesítményű PC került a piacra, amely a 8 bites kategóriában, a hajlékonylemez változatban még az olyan professzionális/félp professzionális gépeknek is veszélyes konkurenciát jelent, mint a Kaypro vagy az Apple IIE.

## Szerkesztőségünk kiegészítése

A Schneider gépek időközben egyre bővülő piaci részesedést harcoltak ki maguknak, és a haza géptulajdonosok köre is egyre bővül. Közük tartozik Forrai István, az Economix Kiszövőtkezet tagja is. Tőle kaptunk részletes felvilágosítást a legújabb kínálatról.

A Schneider gépek tulajdonképpen az angol Amstrad cég gyártmányai, és a német nyelvtérlet kivételével mindenütt az Amstrad forgalmazza őket. Már az éve eleje óta árulják a 664-es típust, amely a kazettás egység helyett a 3 hüvelykes, 180 kbájt kapacitású floppy tartalmazza beépítve. Az NSZK-ban a zöld alapszínű képernyős változat 1500 márka, a színes képernyős 2000 márka kerül. A zöld alapszínű monitorral felszerelt gép ára Angliában 250 font, az értékötletadó is beleértve. Az NSZK-beli árak az értékötletadó nélkül értendőek.

Minőségű többletet hozott a 6128-as típus megjelenése. Ennek RAM kapacitása 128 kbájt, és a jóval hatékonyabb CP/M 3.0 változatot használja. A beépített floppy tartalmazó gép ára 1600, illetve 2100 márka, képernyőtől függően. Az év második felében került a Dixon kiskereskedelmi hálózatába a legújabb, 8256-os típusszámú gép. Ez már teljesen professzionális berendezés, és nagyon ígéretes marketing koncepció van mögötte.

A 458 fontért árusított gép teljes személyisámítógép-képzést tartalmaz. A konfigurációhoz ugyanis nyomtató is tartozik, és professzionális minőségű monitorra van, amely 90 oszlopban és 30 sorban jeleníti meg az információit. Mivel a bevégzés hatékony szövegfeldolgozó is kap, a berendezés azonnal használható elektronikus szövegfeldolgozási célokra, ami egyben a marketing koncepció lényege is. A 8256-os ugyanis, bár jóval többet tud még szövegfeldolgozási szempontból is, versenyképes az elektronikus írógépekkel.

A technikai bravúrt úgy érte el az Amstrad, hogy a teljes elektronikát, még a nyomtató hardvervezérlését is a monitor készülékhezárazta be, így a központi elektronikával old meg periferiális feladatokat is. 80 karakter széles nyomtatója két üzemmódban működik: az ún. draft módban 90 karakter/mp sebességgel, a majdnem levelezési minőségű módban pedig 20 karakter/mp sebességgel. Nagy előnye, hogy közönséges, vágott papírral is jól használható, mivel ehhez külön, automatikus adagolója van. Igen nagy teljesítményű személyi számítógépként is minden további nélkül használható, ugyanis RAM tárhajlékban 256 kbájt, és két floppy meghajtót képes befogadni – a 8256-osba csak egy hajlékonylemez van beépítve. Hamarosan megjelenik 720 kbájt kapacitású floppy meghajtója is.

A magunk részéről nagyon ígéretesnek tartjuk az Amstrad megközelítést, mivel a rendelkezésünkre álló információból az is kiderül, hogy nagy sorozastra van kilátás: 1986-ban havi 40 ezer 8256-os gépet kíván piacra hozni a cég. Szerkesztőségünk az is tudomására jutott, hogy a tavaszi Hungarokex-Qtelle katalógusban már forintért is kínálni fogják a Schneider-Amstrad gépeket.

ATTILA LADÁNYI



## A Commodore 64 felhasználásáról

43 éves üzemmérnök vagyok, az államigazgatásban dolgozom. 1984 végén került munkahelyemre néhány C64 rendszer. A gépet igen hamar sikerült rendeltetésének megfelelően, személyi számítógépként munkám egyes, gyakran ismétlődő műveleteinek elvégzésére alkalmaznom. Felhasználói, saját és külső vállalkozók által készített programokat egyaránt futtatunk.

A C64 nagyszerűségét abban látom, hogy igen üzembiztos, kiválóan alkalmas a számítástechnikai kultúra területén a kezdő lépések megtételére. Mint munkaeszköz sem okoz csalódást, ha lehámozzuk róla az illúziókat.

Lapjuk hangot adott egyes C64-ellenes érzelmeinknek; ezekkel nem értek egyet. Meg vagyok győződve arról, hogy ezt a gépet az utókor egyszer még korszakalkotó gyártmányként fogja emleget-

ni, és olyan legendás lesz, mint az autók világában a Ford „T” modellje volt.

A gép annyira hoztam nőtt, hogy ma már nem helyezkednék el olyan munkahelyen, ahol nincs személyi számítógép. Élvezetet találok ebben a munkában, és ezt a társadalmi munkamegosztás mai, objektíve túlféjtett szintjén figyelemreméltó szempontnak tartom. A műszaki értelmiségnek a személyi számítógép új intellektuális élmény forrása lehet. Nem éreztem fáradtságosnak az erre a célra fordított rengeteg szabad időt.

A mellékelt kis programot illusztrációnak számom; egy feldolgozó programomból emeltem ki. Az a változóban levő értéket a magyar helyesírás szabályainak megfelelően szöveggé alakítja át. Nevezetesen nálam egy államigazgatási dokumentumról volt szó, amelyben egy legfeljebb 9 jegyű tetszőleges egész számot betűkkel is le kell írni. A programlista alatt az eredmény képernyőfelvétele olvasható. A megoldás tökéletlenségem nem kérdés előtt; szívesen venném rövidebb, frappánsabb változatát, tanulságos lenne. Egyébként csak idő és anyagik kérdése, hogy a lapjuk útján megismert lehetőséget, a magyar ékezes ábécét gépünkön megvalósítsuk.

Jelenleg ezt még csak szándékként említhetem. Levelem megírására is ez okból nem alkalmaztam az Easy Script szövegszerkesztőt.

A jövőt illetően – bár kacérkodtam a gépi kódolás megismerésével – erősebb az a meggyőződéssem, hogy a magamfajta felhasználónak legfeljebb a magas szintű nyelv elsajátítása célszerű. A gépi nyelv maradjon a specializáltasá kényere; az ő feladatuk – és a tömeges számítógép-felhasználás feltétele is – a minél könnyebben elsajátítható rendszerek kifejlesztése. Nekem nem érdekes a „fekete doboz” mélyére túlságosan látni.

Gondolataimról azzal a céllal számoltam be, hogy mint egy olvasótípus képviselőjének érdeklődését a szerkesztőség megismerje, és lapjuk arculata egybekel mellett ezt is, súlyának megfelelően tükrözhesse.

KÁTAI LÁSZLÓ

## Ékezeset szövegszerkesztő

Érdeklődéssel olvastam, még nagytóval is megnéztem az ékezeteket „ékezetesen” bizonyító leveleket. En is készítenem ékezes szövegszerkesztőt, de más megoldással. Egy szövegsor két képsor foglал el az ékezeset betűk a szövegsorok leptomásával érhetőek el. A kis- és nagybetűk váltását CAPS LOCK vezérlő, a számokat az OR lenyomása után lehet írni.

Az ékezeset betűket 12 UDG jelből állítottam össze, és egy 6 x 10-es tömbben tárolom.

```
b$(1)
b$(2) 123456789
b$(3)
b$(4) áéíóóóóóóó
b$(5)
b$(6) ÁÉÍÓÓÓÓÓÓ
```

A szövegszerkesztőm 26 különböző funkciót valósít meg (aláhúz, tabulál, üres sorokat szűr be szövegbe, lapoz bárhova stb.). A szövegmező 500 szövegsor, de ez növelhető. A program TIMEX 2068 Sinclair gépre készült, ami a Spectrum amerikai testvére. A különbség az elegánsabb külső, valamint néhány többszolgáltatás, de ezeket nem használja a szövegszerkesztő programom. A nyomtató Alphacom 32 mátrixnyomtató.

GELENCSÉRNÉ LAZAROVITS KLÁRA

## Primo észrevételek

Nekem nagyon tetszenek a Primón bevezetett újítások (egyszerű editálás; gépi kódú programok másolása segédprogram nélkül, a grafika 192 x 256, az origó a bal alsó sarokban van, magyar ékezeset betűk, hibajelzés + hangjelzés stb.), és szeretném, ha még jobb tulajdonságokkal rendelkeznének ezek a gépek. Így hát jönnek a javaslataim:

1. A Primo számos grafikus karaktert tartalmaz, például 7 szegmensre kijelző képeket és egyes figurákat játékoknál jól lehetne használni. Karakterdefiniálással ezeket meg lehet oldani, de egyszerűbb volna, ha a gép eleve tartalmazná ezeket az önmagukban is használható karaktereket.

2. Igen hasznos volna, ha a BASIC tartalmazná a két pont közé vonalat húzó és a körrajzoló rutinokat is.

BEDNÁRIK ATTILA

```
100 dataegy,tizen,egyszaz,kettő,huszon,kettőszaz
110 dataharom,harminc,haromszaz,negy,negyen,negyszaz
120 dataöt,ötven,ötsszaz,hat,hatvan,hatszaz,het,hetszaz
130 datanyolc,nyolcvan,nyolcsszaz,kilenc,kilencven,kilencszaz
140 x#(1)="tíz":x#(2)="husz"
150 printchr$(14)
160 for i=1to9:readq$(i),w$(i),y$(i):next
170 input "szam: ";a:h=len(str$(a))-1:a#=#right$(str$(a),h)
180 for i=1toh:s(i)=val(mid$(a#,h+1-i,1)):next
190 i=0:k=0
200 i=i+3:i#i>h+2 then200
210 z$(i-2)=y$(s(i))
220 z$(i-1)=w$(s(i-1))
230 i#s(i-2)=0ands(i-1)<3thenz$(i-1)=x$(s(i-1))
240 z$(i)=q$(s(i-2))
250 k=k+1
260 for j=i-2to1:u$(k)=u$(i)+z$(j):next
270 goto200
280 onk goto300,310,290
290 u$(3)=u$(3)+"millio"
300 i#u$(2)<"*":oruf(1)<"*":thenu$(3)=u$(3)+"*"
310 i#u$(2)="*":then340
320 u$(2)=u$(2)+"ezer"
330 i#u$(1)<"*":thenu$(2)=u$(2)+"*"
340 for j=kto1step-1
350 os#os#+u$(j)
360 next
370 goto390
380 os#u$(1)
390 printos#
400 print "Kersz uj szamot? i/n"
410 gett:i#ft#="*":then410
420 i#t#="i":thenclr:goto100
```

ready.

run

```
szam? 123456789
egyszazhuszonharommillio-negyszazötvenhatezer-hatszaznyolcvankilenc
```

kersz uj szamot? i/n

```
szam? 987654
kilencszaznyolcvanhatezer-hatszazötvennegy
```

kersz uj szamot? i/n

ready.

hrdcpu



## KEDVES OLVASÓINK!

**Azt ígértük, hogy ebben a rovatban az egyes rovatvezetők is válszolnak a nekik küldött levelekre. Így ma a levelek sorát dr. Simonyi Endre bejelentésével, illetve néhány levélre adott válaszával kezdjük.**

Sok olvasó keresett meg, hogy számítógépéhez különböző dolgokat kérjen. A kérések egy része olyan gépre vonatkozik, amelyből itthon még kevés van, és például azt kérik, hogy hozzuk össze valakivel őket, akinek hasonló gépe van. Ezért új szolgáltatást indítottunk, amelynek keretében kapcsolatot hozunk létre olyan gépek tulajdonosai között, amelyeknek még nincs működő klubjuk; továbbítjuk a kérését, ha közli gépének adatait (típus, perifériák, szoftver stb.), és bényeg-gep-ellátott választásokról is mellékel; értesítjük, ha hasonló kérésrel más is megkeres bennünket.

Természetesen telefonon is adunk felvilágosítást. Levélben vagy telefonon továbbra is megkereshetitek szerkesztőségünket mindenfajta számítástechnikai problémával. Üvegeket minden hétfőn 8-14 óra között a szerkesztőségben (154-250/676-os mellék), 17-20 óra között dr. Simonyi Endre telefonján (369-183) tartunk.

**Csányi Csaba, Sátoralfajhely,**

**Kazinczy u. 59. 390**

Az Épsütnak számítógépet! sorozatban eddig leközlötték megépítettem az alap NYAK-ot. A főláncot az Önök által közölt fényképről másoltam! A problémám ott kezdődött, amikor a video NYAK-ra került a sor. A fénykép némely része elmosódott, és valószínű, hogy én tévedek, de a NYAK-terv és a kapcsolási rajz néhol eltérő.

Más problémám is vannak. 1. Az alkatrészyegyzékben nem szerepelnek olyan alkatrészek, amelyek a későbbiekben igen. 2. A jegyzékben szerepel a 8T26 jelelő IC, én 8T26ANA típusút vásároltam. Nem tudom megállapítani, mennyiben különbözik az előbbtől. A katalógusban a 8T26 megegyezik az SN74125-tel, ami 14 lábú. A 8T26ANA viszont 16. Kérem, hogy tájékoztasson! 3. A billentyűzetről nem tudom, hogy a számítógép mely pontjaira csatlakozik.

A cikkek azt írtam, hogy a gépnek több változata van. Ezek közül az egyik látszik a fényképen, és ettől a rajz kismértékben különbözik. Ez azonban nem okoz problémát, mert a fénykép szerinti változat néhány átkötéssel a rajznak megfelelő alakú. A többi változat:

1. A DM, illetve DS 8835 ilyen. Ez a 8T26A funkcionális megfelelője, de más a lábkiosztása. 2. Igaz van az a helyes a 8T26A, ami 16 lábú. 3. A billentyűzet-illlesztő a sorozat utolsó, XI. részében szerepel csatlakozási egység párhuzamos csatornához csatlakoztatható.

**Buzi Gábor, Boldog**

**Rákóczi út 49. 3016**

Olvastam a  $\mu M$  1985/3. számában, hogy van néhány dolog, ami megoldásra vár a Primóval kapcsolatban. Mivel nekem van Primóm (A-32), és már eléggé kismertem, tollal ragadtam, hogy megírjam észrevételeimet, főleg az 1984/6. számban megjelenettel kapcsolatban. Az idézőjelek között az eredeti szövegéből idézek.

„A kétértelmű pontosság a valóságban nem a számítási, hanem a kiírás pontossága.” De ha a gép nem számolja ki kétértelmű pontossággal, akkor hogyan tudná kiírni így, kétértelműen pontosan? Egyébként a két megjelent programsor hibás. Egyrészt mert ki-

hagyták belőle a # -et, ami nélkül a gép valós számokkal számol, másrészt a két sor teljesen megegyezik, így nem csoda, hogy ugyanaz az eredmény.

„A logikai igen értéke a szokásostól eltérően — 1... Az ON N GOTO M utatitástánál N=0 és N>M esetén továbbmegy.” A logikai igen értéke a legtöbb gépnél — 1, és valamennyi ON GOTO utatitást használó gép a fenti módon használja ezt az utatitást. Legálábbis azok, amelyeket én ismerlek. Hol van hát itt a „szokatlan megoldás”, ami „kedvezőtlen”?

Végeztül azt kérném, hogy közöljék a Primó-programokat, vagy legalább gépi kódú részleteket. Ez utóbbi még hasznosabb lenne. Nem igaz, hogy nehéz a Primóval gépi kódban dolgozni, hiszen a DEMOKAZETTán lévő programokban sok gépi kódú rész található, például a „Vitorlás”-ban gépi kódú DRAW és részleges balra SCROLL. Sajnos elég nehéz ezeket kihámozni a programból.

Valóban az 1984/6. számban a kétértelmű pontossággal kapcsolatos programban sajnos a # jel lemaradt, ezért a két program látszólag azonos. Begépelve két különbözőt. A kétértelmű pontosság azt jelenti, hogy minden lebegőpontos számítás kétértelmű amennyi jegyre számít. Ezt a Primó és a HT-1080Z nem teszi.

Az ON GOTO utatitástal kapcsolatban téved, mert a BASIC nyelvi eredetije szerint sem így volt.

Önök és sok más olvasónak is figyelmébe ajánlom: cikkem írásához csak a kézikönyv állt rendelkezésemre, ezért hivatkoztam csak arra. Azóta megjelent könyvet természetesen ekkor még nem vehettem figyelembe, mert tartalmukat nem ismertem.

**Kissné Nyeste Rita, Tiszavasvári,**

**Élmunkás u. 11. 4440**

Úgy emlékszem, részletesen nem volt megadva (a Primó gépkönyvében — A szerk.), hogy a CTR billentyűt a különböző betűkkel együttesen lenyomva, hogyan lehet a Primóval a képernyőn „játszani”. Kikeresve a megfelelő CHR\$ kódot, az utatitások programja vihető.

A következő betűk a CTR billentyűvel együtt lenyomva a képernyőn az alábbiakat eredményezik:

CTR +	EREDMÉNY	CHR\$ (KÓD)
C	világos képernyő	3
S	sötét képernyő	3
O	függőleges írás	15
W	a függőleges írást vízszintesbe viszi	23
G	hibajel hangot ad	7
L	Törli a képernyőt	12
B	nyújtott (magnóval) betűforma	2
R	a nyújtott betűformát normál betűformába viszi	18
H	szívfaléle egy karaktert töröl	4
D	sávban írás (pl. világos képernyőn a CTR billentyűt és a D betűt együttes lenyomása után a felírt sötét sávban világos betűkkel jelenik meg)	4
E	a sávban írást megszünteti	20
T	aláhúzás	5
U	az aláhúzást megszünteti	21
Q	fél sort emel	17
P	fél sort süllyeszt	16
A	az írást új sorban kezdi	13
N	ugyanabban a sorban a sor elejére viszi a kurzort	13
M	megfelel a RETURN billentyűnek	

A betűk együttesen is alkalmazhatók. Ha tehát a CTR billentyűvel együtt lenyomjuk a B-t, a D-t, az E-t, egymás után, az eredmény — például sötét képernyő — világos sávban sötét felírt, magnóval betűformával, aláhúzás.

A CTR + Q-val a képernyőn a kurzort föléli, a CTR + P-vel lefelé mozgathatom közvetlenül.

Köszönjük levelét, de amit ír, jórészt valóban benne van a gépkönyvben, és részben a  $\mu M$  1984/6. száma-

ban. Tekintettel azonban arra, hogy ugyanez másokat is érdekelhet, a táblázatát közöljük.

Most pedig lássuk a többi levelet.

**Szós Sándor, Szeged,**

**Gyöngyvirág u. 18/A 6723**

Vásároltam egy ZX-Spectrum 48 k-s számítógépet, és megvettem hozzá a kézikönyve magyar változatát, valamint a TV-BASIC könyvet is. Azt hittem az ember, hogy e két könyv alapján el lehet sajátítani a számítógép-tudományt. Lehet, hogy ez igaz is, de nekem nem megy. Kérem Önöket, tájékoztassanak arról, milyen szakirodalom kell megvásárolnom ahhoz, hogy használni tudjam a gépet. Ehhez nyilván matematikai alapokra van szükség, de én egyelőre csak azt kívánám megtudni, hogy miért nem tudom futtatni a TV-BASIC-ben írt Sinclair programokat, amikor helyesen ütöm be azokat. Vagy csak hiszem, hogy helyesen?

Van egy 8 éves fiam. Nem utolsó sorban miatta is érdekel a számítógép. Hogy magyarízzák neki, ha én sem értek hozzá?

Azt javaslom, hogy keresse meg Szegeden az NJSZT Csongrád megyei szervezeteit, a Technika Házában. Dr. Makay Árpád vagy Paku Sándorné, a szövetség titkárai, egész biztosan segítenek a problémák megoldásában.

**Király László, 5230 Sömmerdő Fichte Str. 15.**

**Pf. 5-03, DDR**

Az NDK-ban élek, itt is dolgozom. Kb. 4 éve foglalkozom mikroelektronikával. 3 hónapja tudok itthon programozni. Nem kis nehézség is anyagi befektetés árán sikerült egy C64-et vásárolnom.

Az Önök újságját is megrendelém, ha lehetne, de a bürokrácia gátat vet. Még a Magyar Kulturán keresztül sincs rá lehetőség. Kérem: szívesen lennélek azokkal, akiknek jobbikja a programozás. Az sem baj, ha nem C64, hanem másfajta számítógépük van.

Mindent megteszünk, hogy a  $\mu M$  a külföldön élő magyarokhoz is elkerüljön, de eddig nem sok siker kísérte igyekezetünket. Felhívását szívesen közöljük.

**Szilvássy Péter, Budapest,**

**Hársfa sétány 15. 1203**

Örömmel fogadtam az 1985/3. számban megjelent repülésszimulációs programok leírását. A magam részéről az eddigi játékok közül ennek vettem a legnagyobb hasznát. Régebben kiseleztem a Flight Simulation programot, mert háromnegyed órányi repülés után képtelen voltam a leszállásra. Most, hogy már tudom, mit és hogyan kell csinálni, újból felveszem, és megszerzem mellé a Fighter Pilot-t is. Érdemes az ilyen és ehhez hasonló programleírásokkal foglalkozni.

Van egy ZX-Spectrumom, és szeretném megtanulni a Spectrum assemblert. Jó lenne, ha indítanánk egy cikkorozatot, amely ezzel a témával foglalkozik. Ha akkora rá szerző is hely is a lapban, megköszönöm.

Sok sikert és valamivel pontosabb megjelenést kívánok a Mikromagazinnak!

Igen, ha akadna rá szerző... Ami a pontosabb megjelenést illeti, a nyomda kezében vagyunk. Ezzel kapcsolatban figyelmébe ajánlom a Kortárs 1985. szeptemberi számát.

**Faragó Zsolt, Budapest,**

**Nagymező u. 12. 1065**

2. osztályos gimnazista vagyok. Szeretnék néhány megjegyzést fűzni a laphoz. Összehasonlítottam az 1984/4. és az 1985/3. számukat. Nos, a fejlődés szerintem szembetűnő: az újság színesebb, érdekesebb lett. Mégis az az érzésem, meg jobb lenne, ha valahogy megpróbálnák tömöríteni: nagyon sok üres hely, folt van az oldalakon. Színt adna az újságnak, ha még több rövid cikket, apró érdekességet hozna. Felelesek nagyon találtak néhány képet (például 1985/3. szám, 18-19. oldal).



Végül lenne egy javaslatom: olyan rovatot kellene indítani, amelyik egy-egy újabb számítógépet mutatna be, attól függetlenül, hogy a magyar írjak zsebéhez valószínűleg sohasem lesz elég olcsó. De végül is ideje lenne, hogy végre megismerjük egyik-másik gép belesjét!

*A szerkesztők annak örülnek, ha az olvasók észreveszik a fejlődést. Mi szeretjük a képeket és az ábrákat, amelyeknek méretét az határozza meg, hogy valami be akarunk-e mutatni részletesen, vagy sem. A javasolt rovat már megindult, legfeljebb nem rovatnak nevezük. Már eddig is nagyon sok gépet bemutatunk (Proper, Primo, IBM PC és mások). Ezt a hagyományt folytatni fogjuk.*

Geges József, Budapest

Szakasítás Árpád út 41. 1119

Minden bizonnyal sokan kezdik így levelüket: „Nem vagyok számítástechnikai szakember, mégis érdeklődve olvasom az Önök lapját...”. Magam is így vagyok ezzel. Mindig találok benne olyan cikkeket, olvasói véleményeket, amelyek kielégítik a kíváncsiságomat. Véleményem szerint éppen ettől jó az Önök által szerkesztett magazin. Sokak érdeklődését képes felkelteni, lehetőséget terem az ismeretek széles skálán való bővítésére, nyilvánosságot ad a jó tapasztalatoknak, hasznos tanácsokkal szolgál.

Nem írnék, ha a fenti jó véleményem ellenére nem hiányolnék valamit. Ez pedig az a tény, hogy ismert objektívításuk ellenére kissé mostohanak bannak az Önök által nem „favorizált” géptípusok tulajdonosai-

val. Mindent számos megfontolásból érthetőnek és indokoltnak tartom, és bízom abban, hogy e „megkülönböztetés” nem a C64, VIC-20, Spectrum, ZX81, HT-1080Z vagy a Primótól eltérő géptípusoknak és tulajdonosainak szóló kritikájuk, hanem az a reális felismerés, hogy országunkban – bizonyos okok miatt – éppen az előbb említettek a legelterjedtebb típusok, és a lapnak ehhez kell igazodnia.

Véleményemmel és a következőkben leírt tapasztalataimmal nem szeretnék vitát provokálni. Szándékom szerint az Önök segítségével szeretnék lelket önteni mindazokba, akiket elkeserít, hogy számítógépükhez nincs megfelelő magyar szaktirodalom, használati utasítás, programfüzet, játékirodalom, vagy más, széleskörűen használható szoftver.

Igazukat látom abban, hogy az enyémhez hasonló gépek (8k-s LASER 210) tulajdonosai kissé magukra hagyottan kénytelenek megismerni mindazokat a számítógéppel kapcsolatos elméleti, gyakorlati, programozástechnikai alapokkal, amelyek elsajátítása nélkül elképzelhetetlen a számítógép cészerű működtetése. Igaz az is, hogy sokkal nagyobb fáradsággal, sok-sok türelemmel és kitartással lehet tuljutni a kezdeti nehézségeken, mégis óriási előnyét látom annak, ha valakinek van elég ambíciója a „maga választotta” úton járni.

Tapasztalásom szerint ugyanis az ilyen áron megszerzett ismeretek tartóssága és alkalmazása, az első egyszerű programok sikeres futtatása nagyobb örömet szerez, mint a „konzerv programokkal” való órákig tartó játék. Egy program kidolgozása, a hibák bogarászása, a véglegesnek tűnő, de soha nem a legjobb megoldások megtalálása hasznos, ismereteket

bővítő időtöltés és jó szórakozás. Egy géphez közel kerülni, a „lelket” megismerni nem a vele való játékunk lehet (bár ennek jelentőségét bizonyos korosztályok esetében kar lenne tagadni), hanem a programozással, a géppel folytatott közvetlen „kommunikációval”. Így érthető, hogy nem osztom azok véleményét, akik egy géptípus használhatóságát és képességét a mellé rendelt játékirodalom alapján ítélik meg. Sokan éppen azért akarnak tudni kevésbé elterjedt számítógépeken, mert az megunt játékká vált a kezükben. A hetekig nyújtott játékirodalom helyett egyre köriményesebb az új kazetták beszerzése, és nem könnyű egy új program önálló elkészítése sem. Ha más nincs, a megoldás ennél sokkal kezenfekvőbb. A más gépeken bevált programokat át kell dolgozni a saját géptípus képességeinek megfelelően. Ennek is megvan a maga szépsége és a gyakorlati haszna, mert a nagy pontosságú követelő munka közben meg kell ismerni a BASIC tájékoztatóit és egy más formában öntött logikai lánc kapcsolatait. Az szaktitukus és logikai elemzés az ismeretek bővítésének rendkívül gazdag lehetőségei.

Ily módon jártasság szerzhető a programozásban a programzseléméletben és a probléma megfogalmazásában, az egyedi gépi megoldások „frappanságának” megítélésében. A megszerzett újabb ismeretek próbája a gyakorlat, és ezekben elengedhetetlenül fontos a tapasztalatok összegezése (esetenként táblázatba foglalás), az okulás az elkövetett hibákból, melyekből nekem is akadatok szép számmal.

Rábédredek majd, hogy a mi gépünknek mi a hátránya és az előnye más gépekkel szemben, s kétségei alapos ismeretében, ha feladatainkkal, szándékainkkal összeegyeztethető, biztosan nem kívánunk megválni tőle.

Úgy tudom – elnézést, ha tévedek –, a LASER 210 Ázsiában a legnépszerűbb gépek egyike. Történelen fejlesztése és használhatóságának számos tapasztalattal történő kiterjesztése nem hiszem, hogy okot adhat e géptípus tulajdonosainak elkeseredésére. Mellette szól az is, hogy (egy magyar turista valutakeretéhez mérten is) szerény költségekkel olyan konfigurációba fejleszhető (sornyatmóltó botkormány, fénycruza, floppy stb.), amely versenyhársra lehet számos más, hazánkban elterjedt géptípusnak.

Kétségtelen, hogy az ismerkedés kezdeti nehézségein túl is lesznek pillanatok, amikor kilátástalannak ítéljük helyzetünket, de a magam tapasztalataiból valom, hogy ennek fő oka gyakorta abból fakad, hogy nem ismerjük eléggé gépünk.

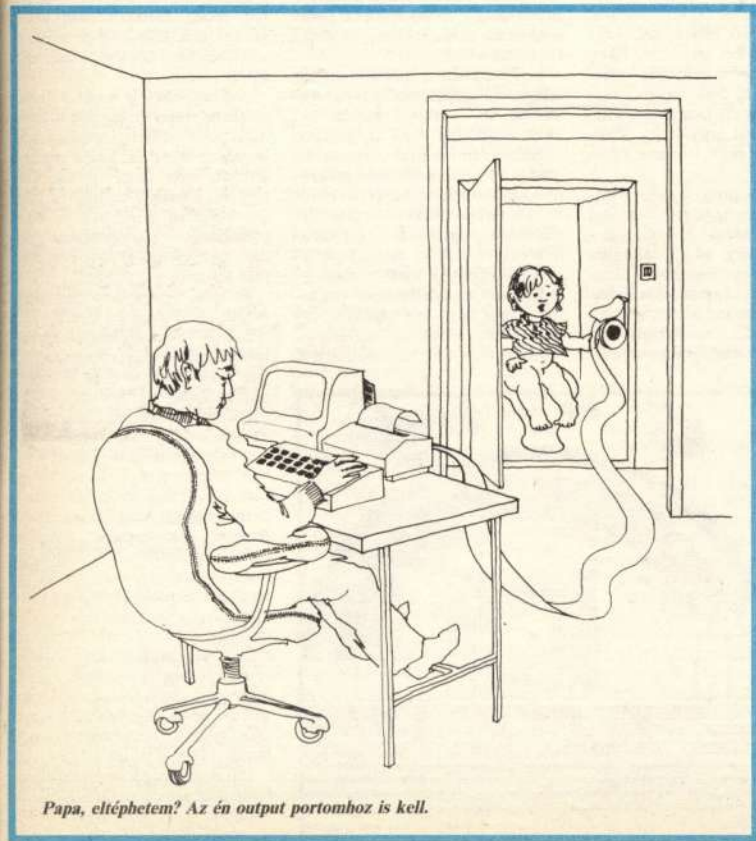
Az igazsághoz tartozik, hogy a LASER 210-zel való ismerkedésnek nem voltak nyelvi akadályai (angol, német, BASIC). Ma már bevaltonal sokat köszönhetek „ismeretségünknek”. Mindenekelőtt megbízható, pontos és kritikus bárátra lettem, amely kiabrándított a számítógépek önmagától mindent megoldó képességének illúziójából, rákényszerített szunnyadó ismereteim felrészítésének és bővítésének szükségességére, segítséget nyújt a munkában és örömet szerez az egész család számára.

Végezetül: jegyzetsoraimat nem a LASER 210, vagy annak tulajdonosai védelmében írtam. Tapasztalásaim függetleníthetők e géptípustól. Azokhoz kívánom szólni, akik örömeiket lelik a járattal utak felfedezésében. Mindehhez csupán biztatást szeretnék adni. Mindazon LASER 210 tulajdonosoknak pedig, akik nem rendelkeznek magyar nyelvű gépkönyvvel, egy nem hiteles és nem szakszerűen pontos, de jól használható fordítással készséggel segítséget nyújtok. Remélve, hogy ők is mihamarabb tuljutnak a kezdeti nehézségeken.

Szokásunktól eltérően a teljes levelét közlöm, mert az elsőtől az utolsó betűig egyetérték a leírattal. Azért azt még hozzátenném, hogy abszolút ismeretlen gépet csak az vegeyn, aki elég erőnek érzi magát a gép lelkének megismeréséhez, és vállalja, hogy a problémáikat egyedül, segítség nélkül kell megoldania. Ha valaki olyan gépet választ, amelyből lithon sok van, ráadásul még klub is működik, az „megsörölja az iskolapénzt”, igaz ugyan, hogy nagyon sok sikert hozó pillanatot is élvezhet.

Nagyon szeretném, ha az országban sokan megszerzenék maguknak azt az örömet, amelyről Geges József ír; ezt kívánja

KOVÁCS GYÖZÖ



Papa, eltéphetem? Az én output portomhoz is kell.



## LÉPÉSRŐL LÉPÉSRE

### Kitekintés

A sakkprogramozás a mesterséges intelligencia területéhez tartozik. Magas intelligenciát igénylő szellemi tevékenység megvalósítása a feladat; olyan tevékenység, amelyre még csak közelítőleg optimális megoldó algoritmus sem található. Sorozatunk utolsó részében vizsgáljuk meg, hogy mennyire általánosan érvényes a mesterséges intelligencia jelenlegi állására az, amit a sakkprogramok képesen megismertünk.

Egy korábbi részben említettük, hogy már 1959-ben jelentek meg komoly szakértőktől olyan jóslatok, amelyek szerint 1965-re a számítógép le fogja győzni a sakkvilágbajnokot. Körülbelül ugyanekkorra tettek azt is, hogy a számítógépek egyik nyelvéről a másikra jó minőségű fordításokat fognak készíteni, hogy el fog készülni az automatikus diktafon, amely egy elmondott szöveget képes lesz helyesen leírni, és hogy a számítógépek megbízható diagnosztikusokat fognak adni a betegségekrol.

E területek mindegyikén igen nagy erővel folynak a kutatások, már csak a dolog nyilvánvaló gazdasági jelentősége miatt is. Ennek ellenére nem hesszmegek a hiradások az elkészült nagyserű készülékekről, de továbbra is rendszeresen találkoznak olyan jóslatokkal, amelyek a nagyon közeli jövőre ígérnek ilyeneket.

A mesterséges intelligencia legtöbb ágának története nagyon hasonló a sakkprogramozáséhoz. Eleinte roppant egyszerű algoritmusokkal, pusztán a számítógép mennyiségi lehetőségeinek, gyorsaságának és pontosságának ügyes kihasználásával sikerült látványos eredményeket elérni. Egy bizonyos szint elérése után azonban a pusztán keresési elveken alapuló algoritmusoknak lényeges finomítása, bonyolítása sem hozott átütő eredményt. Sakkozásban például a minimax algoritmuson alapuló programok lényegében a mesterjelölt szintig tudtak eljutni.

A sakknál még viszonylag szerencés a helyzet, hiszen egy mesterjelölt eréjú program már üzleti-  
leg garantált siker, és az emberek nagy többséget teljesen kielégíti. A sakk ugyanis a legtöbb ember számára játék, csak a kiemelkedő tudásúak – azok sem mind – érik el a mesteri szintet. Anyanyelvét

azonban a legtöbb ember „mesterfokon” beszéli: tökéletesen érti, és már nagyon kis hibák is roppant zavarják. Egy „mesterjelölt eréjú” fordítóprogram éppen azt tudja nyújtani, hogy az általa létrehozott szöveg szinte olvashatatlan. Annak, aki a gép által lefordított szöveget lektorálja, több a dolga, mint ha maga fordítaná le az anyagot. Még világosabb, hogy hasonló a helyzet egy „mesterjelölt szinten” diagnosztizáló program esetében. Ezeket a területeken a programnak „mesterré” kell válnia.

Láttuk, hogy egy sakknagymester kb. 50–100 ezer, sakkal kapcsolatos sémát ismer. Valószínűleg más szakmákban is hasonló a helyzet. A mondott sémaszám nagyjából megegyezik a kiváló irodalmárokkal által ismert szavak és szófordulatok számával, vagy például a kínai írás teljes jelkészletének mennyiségével. A sakkmesterjelölt viszont már csak néhány száz vagy legfeljebb egy-két ezer sémát használ. Beszéd esetén ennyiféle fordulat segítségével csak nagyon egyszerűen tudnánk magunkat kifejezni, egy kicsit árnyaltabb fogalmazásra már nem lennénk képesek.

Azt is láttuk, hogy a szóban forgó sémák nem egyszerűen lexikai egységek (például szavak, kész mondatok vagy adott sakkállások). A sémák tartalmazzák a lexikai egységekkel kapcsolatos tudnivalókat is, használatuk nehezen megfogalmazható szabályait. Éppen ez okozza a mesterséges intelli-

genciának a legfőbb problémát: könnyű a szótár összes szavát beprogramozni, de ebből még nem lesz értelmes szöveg. Még akkor sem, ha a nyelvtani szabályokat is beprogramozzuk, tehát csak olyan mondatokat készít a számítógép, amelyek ezeknek nem mondanak ellent. Hasonlóképpen be tudunk programozni akár százezer mesterjátszma-állást is, de ebből még nem lesz mestereréjú sakkzó program.

A sakkhoz hasonlóan az utolsó tíz évben a mesterséges intelligencia többi ágában is egyre inkább áttértek a kutatók a séma elvű programok készítésére. Ennek oka, akárcsak a sakkban, az volt, hogy a kezdetben nagyon látványos eredményeket hozó egyszerű keresési algoritmusok (mint például a minimax) komoly korlátokba ütköztek. Mind az algoritmusok fejlesztgetése, mind a gépek gyorsabbodása már csak nagyon kevés többletet adott.

Napjainkban a mesterséges intelligencia legigéretesebb programjai az ún. *szakértő rendszerek*. Ezek működési elvei lényegében olyanok, mint amiket a sémák segítségével működő sakkprogramoknál láttunk. A szakértő rendszerek tartalmazzák a szakterületük tényanyagait és főbb fogalmait, ezenkívül sokféle szabályt, amelyek a sémáknak felelnek meg.

Ezeknek a szabályoknak egy része olyan, hogy segítségével egy adott helyzetben az egymással kapcsolatban álló tényeket, illetve

a fogalmak konkrét megnyilvánulási formáit ki tudják keresni. Számos szabály van arra is, hogy hipotéziseket tudjanak generálni, és azokat képesek legyenek ellenőrizni is. További szabályok vezérik, hogy miéle hipotéziseket próbáljon generálni a rendszer ahhoz, hogy egy konkrét feltett kérdésre válaszolni tudjon. E szabályok vezérlik tehát végeredményben a megoldás keresését.

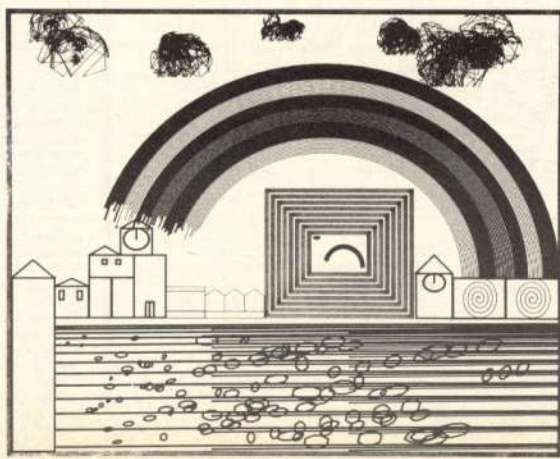
Jelenleg a legfejlettebb szakértő rendszerekbe (például a különféle betegségek diagnosztizáló MYCYN rendszer változataiban) kb. egy-két ezer sémát sikerült beépíteni. Ez nagyjából annyi, mint amennyit a sakkprogramoknál láttunk, és ami kb. a mesterjelölti erőhöz szükséges. A sakkprogramoknál ezen az úton valójában még alig sikerült ezt a szintet elérni, míg a diagnosztizáló szakértő rendszerek éppen túlszárnyalták azt a szintet, ameddig nagyon egyszerű módszerekkel korábban már eljutottak.

Annak, hogy a szakértő rendszerekbe ennél lényegesen több sémát építsenek be, ugyanazok a nehézségei, mint amiket a sakknál láttunk: igen nehéz egy idő után további sémákat találni, és főleg úgy beépíteni a rendszerbe, hogy összehangoltan dolgozzanak, például ne tegyenek tönkre korábban már jól működő részeket.

Ez nem egyszerűen mennyiségi probléma. Valahogy minden elég általános feladatra készült szakértő rendszer megtorpan ezen a szinten. Valószínű, hogy a továbblépéshez újabb szervező elvek felismerése lesz szükség, talán éppen az emberi gondolkodás működésének mélyebb megismerése révén.

A nagymester eréjú sakkprogram létrehozása tehát továbbra is jó modellfeladatnak látszik a mesterséges intelligencia számára. A fejlődés nehéz előre megjósolni, hiszen ugyanúgy, mint ahogy egy régebbi számunkban láttuk a minimax elvű programok fejlődési trendjének megtörését, az első séma elvű programok fejlődési trendje is megtörni látszik úgy néhány ezer séma körül. Több tizezernyi séma meghatározására és összcépitésére pedig egyelőre nem bontakozott ki új elv.

Illő ezzel kapcsolatban még néhány szót szólni az ún. ötödik generációs számítógépekről. Éppen az elmondottak alapján személy





## ISMÉT MAGYAR SAKKPROGRAM A NEMZETKÖZI ARÉNÁBAN

Ismét – mert hiszen 1974-ben, a Stockholmban rendezett első számítógépes sakkvilág bajnokságon szerepelt a magyar PAPA, Rajna György és Almási Béla programja. Hardvert nem vihettek magukkal, a versenyen kapott CDC Cyber 73 típusú számítógép túl lassúnak bizonyult, sorozatosan átlépték a gondolkodási időt. David Levy, a verseny főbírája sajnálattal írta, hogy nagy kár volt érte, mert az előzetes információk alapján az esélyesek között tartotta számon ezt a programot. Még sajnálatosabb volt, hogy a kitűnő sakkmester és a kutatómérnök nem fejlesztette tovább programját; nem mutatkozott kellő érdeklődés a számítógépes sakk iránt, és hiányzott a munkához az idő, az anyagi fedezet.

Egy évtizednek kellett elteltie ahhoz, hogy újabb sakkprogram készüljön Magyarországon, Kovács Attila egyetemi hallgató, programozó és a Honvéd mesterjelöltje szorgalmának, kitartásának eredményeképpen. Commodore 64 típusú készülékre írta programját a Novotrade Rt. támogatásával, és a Magyar Sakk Szövetség Számítógép Bizottsága benevezte a szeptemberben Amsterdamban rendezett, 5. Mikroszámítógép Sakkvilág bajnokságra. Az amatőrök csoportjában a 4. helyre került; egy nyert és egy döntetlen játszmat könyvelhetett el.

Ez természetesen nem kimagasló eredmény, de a körülmények figyelembevételével jelentősnek kell tartanunk. Mert az a tény ma sem változott, hogy a hardvert tekintve a program a legnehezebb helyzetben volt. Bár Commodore 128-ason futott a versenyen, mégsem küzdhetett egyenlő feltételekkel a kivétel nélkül nagyobb teljesítményű – részben 16 bites processzorral ellátott – számítógépekre írt programokkal. Például egy holland program ellen döntetlenül végződött játszmájában nagyszerűen kidolgozta a nyert állást, de az i-re a pontot viszonylagos lassúsága miatt nem tudta rátenni.

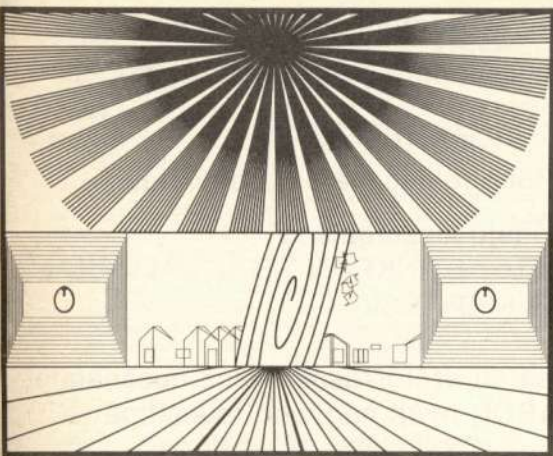
Kovács Attila ez év elején kezdte munkáját. Egyedül végezte el, és hogy jól fogja meg a dolgot, aligha vitatható. Áttanulmányozta Da-

vid Levy nemrég megjelent könyvének azt a részét, amely az állásértékeléshez alkalmazott matematikai függvény felállítására hoz fel példát. Ebben a különféle sakkbeli kritériumok – az anyagi erő, a királyok helyzete, a tiszték hatóereje, a gyalogok formációja stb. – matematikai pontértékekkel szerepelnek, s ezek kellő súlyozásától függ, hogy végül is a program a vizsgálandó, létre jöhető hadállásokkal helyesen értékeli-e. Nos, Kovács Attila megállapította, hogy a Levy-féle koefficiensnek csppet sem reálisak. (Ez persze nem véletlen; a kiváló sakkmester-programozó mintát adott, de senki sem kívánna meg tőle, hogy az általa legjobbnak ítéltet bocsássa közre.) A magyar versenyző „helyettesíté” tehát ezeket az értékeket, saját maga döntötte el, hogy mely tényezőket milyen súllyal vegyen számításba a program. Amint mondja, nem „anyagias” programot készített, a pozíció javítása érdekében szívesen hoz anyagi áldozatot is.

Arra törekedett, hogy a minimax eljárás működését, ahogy a lapunkban megjelent cikksorozatban olvasta, gyorsítsa. Gondot okozott számára, hogy az alfabéta algoritmusban melyik fázisban váltson alfáról bétára – a teljes elemzésről a szelektálásra –, s főként, hogy melyek legyenek a szelektálandó lépések, tehát azok, amelyeket tovább, mélyebben elemez. Kézenfekvő, hogy a sakk adására, az útre vagy a vezér megtámadására adott válaszokat szelektálni kell, továbbá azokat is, amelyek az értékelés során a legmagasabb pontszámú hadállásokhoz vezetnek. De rájött, hogy a legrosszabb lépéseket – például azokat, amelyek a saját királyt mattvesztélynék teszik ki, vagy nagyobb anyagi veszteséghez vezetnének – ugyancsak szelektálni kell, hogy ezeket eltávolítsa, ismétlődés-kor ne kelljen újból időt tölteni velük.

Az elmondottak persze csak kis szemelvények a fiatal magyar programozó gazdag gondolatvilágából, amely remélhetőleg a mostaninál még jobb, tökéletesebb sakkprogram készítéséhez vezet el.

LINDNER LÁSZLÓ



szert meg lehetőségen szkeptikus vagyok a dologban. Az ötödik generációs projekt elindító nagyon szépen megfogalmaztak olyan fontos célokat, amiket már huszonöt éve ugyanígy megfogalmaztak, csak akkor még nem hitték, hogy ennire nehéz lesz őket megvalósítani. A megvalósítás módjáról viszont az ötödik generációs számítógépek apostolai is csak annyit tudtak mondani, hogy az eddigi elveket megvalósítjuk még jobban, még gyorsabb számítógépeken.

Kétségtelenül fognak ötödik generációs számítógép címén új eredmények születni, hiszen ha egy szakértő rendszer olyan szűk speciális területen ad tanácsot, amire senki sem hajlandó magát mesterre túlspecializálni, akkor már mesterjelölt erejéig is igen hasznosnak bizonyulhat. A sakk azonban eléggé érdekes, komplex játéknak látszik ahhoz, hogy jelenlegi tudásunknak még ellenálljon.

### Hogyan játszunk a sakkszámítógép ellen?

Úgy tűnik, hogy ha sakkprogrammal kerülünk szembe, akkor még egy jó ideig minimax algoritmusra alapozott programra készüthetünk fel. Ezek pedig, mint megtudhattuk, jóval gyengébben szakkoznak, mint amilyenek látszanak. Amiórt mégis olyan eredményesek néha egész jó sakkoköz ellen is, az egy érdekes pszichológiai csapda.

A legtöbb ember első gondolata az, hogy ha gép ellen játszik, akkor ki akarja használni, hogy az ellen-

fele gép: nincs intuíciója, nincsenek ötletei. Ezért élesen játszik, elbonyolítja az állást azzal, hogy egy nehéz állásban inkább talál valami jópofa megoldást, mint egy gép. Valójában éppen ezzel nyújtja a lehető legtöbb esélyt a gépnek, hiszen ilyenkor a legvalószínűbb, hogy néhány lépéses távon döntő előnyhöz lehet jutni. És néhány lépéses távon a gép mindent lát. Ilyenkor nem azért játszik zseniálisan a gép, mert észreveszi a csodás eszmét, hanem mert az a lépés fogja adni a minimumok maximumát. A csodás eszmét csak mi vesszük észre, amikor a gép megjáttszja ellenünk.

Paradox módon a gép ellen az a legeredményesebb játéktípus, amit az ember képesnek nevez: a szolid játék, hosszú távú kis előnyök elérése, lehetőleg egyszerű állás mellett. Az egyszerű állás a gép számára éppoly bonyolult, mint bármely más. Az ember viszont jobban át tudja tekinteni, messzebb tud benne előre tervezni. Ráadásul egyszerű állásban valószínűleg, hogy komoly hátrányba kerülünk. Komoly hátrányban pedig a gép ellen jóval kisebb az esély, mint egy ember ellen: a gép pontosabban számol, nem fárad el, nem éhez meg, nem unja egyszerűen végigjátszani a nyert állást.

Ha valaki hajlamos akármennyire sülyedni, csak hogy megverje a számítógépet, akkor megnézheti, hogy melyik megnyitásméleti könyvet ködölték be a gépbe – nincs az a könyv, amelyben ne lenne sajtóhiba. És hol van még a mesterséges intelligencia attól, hogy a gép gyanút tudjon fogni!

MÉRŐ LÁSZLÓ



# NEM KÉSETT EL?

Megtakarítja idejét, a fáradságos rutinszámításokat elvégzi az IBM KOMPATÍBILIS PROPER 16 professzionális személyi számítógép.

MINISZÁMÍTÓGÉP TELJESÍTMÉNY  
egy felhasználós PROPER 16-ot több felhasználásra bővítheti.

Több felhasználó egyidejű kiszolgálása  
M08X, COMMODORE, VT 52 GÉPEKKEL, ASZINKRON TERMINÁLOKKAL  
HELYI HÁLÓZAT KIÉPÍTÉSE

Személyi jellegű, de biztosítja a kiterjesztett erőforrások közös használatát.  
12 MUNKAHELYIG ALACSONYABB ÁRON nyújt nagyobb teljesítményt.

## RENDKÍVÜL JÓ ÁR/TELJESÍTMÉNY VISZONY

PROPER 16 W 742 000,— Ft

PROPER 16 mega 995 500,— Ft

PROPER 16 m 391 000,— Ft

HELYI HÁLÓZAT TAGJAKÉNT

VILÁGSZÍNVONALÚ PERIFÉRIÁK  
RGB 133 SZÍNES MONITOR, RADIX 200 NYOMTATÓ, LT 880 PLOTTER  
NAGYGÉPES KAPCSOLAT

Nincs szükség külön személyzetre, külön gépteremre, klimatizálásra.

RÖVID HATÁRIDŐVEL SZÁLLÍTUNK  
REGIONÁLIS SZERVIZ HÁLÓZAT: SCILCO

EZ A MA TECHNIKÁJA!

Szki 

 Sci-L

 Scitel

INFORMÁCIÓ

1011 Bp. I., Iskola u. 10.



## **Bodor Tibor- Gerő Péter: Commodore 64 programozásának gyakorlata 1. Alapismeretek**

(Budapest, 1985. SZÁMALK,  
160 oldal. Ára: 55,— Ft.)

Az ismert szerzőpáros új könyvében arra törekszik, hogy az olvasó a Commodore 64 kezelésén és a BASIC nyelv alapfogalmain kívül a programozási gondolkodásmódot is elsajátítsa. Bár ez nem helyettesíti a személyes gép-használat tapasztalatait, de megkönnyíti ezek megszerzését azok számára, akiknek ez az első találkozásuk a számítástechnikával.

A könyv az elemi kezelési ismeretektől kiindulva, fokozatosan ismerteti meg olvasóját az egyre összetettebb programozói fogásokkal is. Szerepel benne a Commodore 64 központi egysége és a leggyakoribb kiegészítő egységek: a Datasette kazettás egység, az [54]-es lemezegység és az MPS 801-es nyomtató kezelési leírása. A Commodore 64 BASIC nyelvének alapjai a programozási funkciók szerinti csoportosításban szerepelnek.

A könyv anyagának elcsajjásítása az előismeretekkel nem rendelkező kezdőt is képes teszi arra, hogy a Commodore 64-es konfigurációt összeállítsa, kész programokat használjon, másoljon, kezeljen, elvlasson, kisebb mértékben módosítson, és egyszerűbb saját programokat készítsen. A könyvben szereplő listák és képernyő-felvételek sok esetben korábbi gépkönyvek, leírások hibáit is korrigálják. A témához tartozó fontosabb számítástechnikai szakkifejezések magyarázatát külön részekben foglalják össze a szerzők. Az eligazodást tárgymutató is segíti.

## **Böck István: ZX81 BASIC és ASSEMBLER**

(Budapest, 1985.  
Műszaki Könyvkiadó,  
181 oldal. Ára: 44,— Ft.)

Az elmúlt években hazánkban is rohamosan megnőtt a személyszámítógéptulajdonosok száma. Házi használatra elsősorban a ZX81 és a ZX-Spectrum típusok terjedtek el. Ez a könyv a ZX81 programozásáról és — példákban keresztül — alkalmazási lehetőségeiről szól.

A könyv célja kettős: egyrészt bemutatni a BASIC nyelven való programozásnak néhány olyan módszerét, amely szinte kizárólag személyi számítógépen alkalmazható — újabb lehetőséget adva a hagyományos eljárásokhoz —, másrészt a BASIC programozás határain túllepve bevezetni az olvasót a gépi kódban való (ASSEMBLER) programozásba. Az ismertetett módszerek

természetesen nemcsak ZX81-re alkalmazhatók, a példaprogramokon azonban kisebb változtatások szükségesek, ha más típusú gépen akarjuk futtatni őket.

A könyv egyaránt szól kezdőknek és haladóknak, nemképp a gépkönyvet is igénykíséri pótolni. Az első két fejezet a ZX felépítését, utasításkezelését mutatja be. A harmadik fejezet a személyi számítógépek BASIC programozási technikáját ismerteti, megmutatva a BASIC programozás határait. A gépi kódban való programozásba a IV.—V. fejezet nyújt betekintést. A IV. fejezet a Z80 (ill. a Z80A) mikroprocesszort mutatja be, az V. pedig az Assembly programozásra ad példákat. A függelék a gép használatát megkönnyítő táblázatokat kívül tartalmazza a ZX81-hez kapható házaí és külföldi programok jegyzékét.

## **Dr. Kovács Magda: Egyszerűen a MIKRO- SZÁMÍTÓGÉP-ről**

(Budapest, 1985. LSI ATSZ,  
322 oldal. Ára: 156,— Ft.)

Újabb kiadvánnyal bővült a magyar nyelven, a számítástechnikával most ismerkedők számára készült oktató-ismeretterjesztő könyvek sora. A kritikus csak a mű átnevezésétől való kockázatja meg a könyv ilyen módon való besorolását, mivel sajnos sem az Előző, sem a Bevezetés nem árul el semmit arról, hogy milyen céllal és kiknek íródott a könyv, pedig ez a két tényező határozza meg elsődlegesen azt, hogy milyen tematikus felépítés és kifejezési módszert alkalmazzon a szerző.

A nyolc nagyobb szerkezeti egységből álló művet három fő részre osztjuk. Az első rész (I. és II. fejezet) a mikro-számítógépek általános bemutatásával, fejlődésével, alkalmazásával, a mikroprocesszoros technika alapfogalmaival egyetűségi megismerteti az olvasót. Az itt közzétett ismeretanyag befogadását hátráltatja a témák kiérett rapszodik elrendezése (pl. miért tartozik az *Alapfogalmak* körébe a *Mikroprocesszor családok alkalmazásának műszaki-gazdasági értékelése?*). A kis terjedelmbe bezsűfolt információ- és adathalmaz nehezíti a használt fogalmak és összefüggések elmélyült megértését, ami pedig az igazi tudás és ismeret megalapozásához nélkülözhetetlen.

A második rész (III., IV., V. és VIII. fejezet) a mikroprocesszorok és mikrogepek hardver- és rendszertechnikai kérdéseit tárgyalja. Az olvasó itt könnyen zavarba jöhet, mert nem tudja, hogy valamiféle általánosított mikroprocesszor-architektúrával ismerkedik-e meg, vagy — ellhanglata a típus — egy adott termékcsalád adott felépítésén keresztül vezeti be a szerző az általános rendszertechnikai alapelveket, vagy netán minden létező mikroprocesszor és mikrogep az ismertetett mó-

don működik. Vajon minden gépcsaládban ilyenek a vezérlőegység jelei (IV.1. táblázat)? Vajon minden processzor vagy tároló IC ilyen lábkiosztással rendelkezik? Az olvasó értetlensége esetleg tovább fokozódik, amikor a IV. fejezetben, amelynek címe: *Programozási alapismertetek*, valami egészen mást talál. Mert a textusalkotózatás, az utasítások időzítése, a mikroutasítások és a mikroprogramok nem a programozási alapismertetek körébe tartoznak, hanem egyedül csak az az alig több mint egy oldal, amit az assembler és a memonik fogalmak kapnak. Ezek viszont megkeresnek ebben a fejezetben?

Ezzel az utóbbi témával foglalkozik a harmadik fő rész (VI. és VII. fejezet). A programozásnál kapcsolatban két dolgot fejt ki a szerző részletesebben: a hexadecimális jégyekkel leírt (majdnem)-tárgyprogramot és az assembly nyelvet. (A könyv ugyan „assembler nyelv”-ről beszél, de ma már erre az árnyalt megkülönböztetésre a nyelv és a fordítóprogram között nem sokan érzékenyek.)

A „Forrásgrogramnak nevezük a hexadecimális számjégyekkel felírt programot” mondat kissé bizarrul hangzik, s bizonyára a szerző sem így gondolta. Az olvasót azonban furcsa érzés keritheti hatalmába a VI.2. és VI.2.1. fejezetet olvasva, hogy valóban e módszer lenne az, amely a „báris számjégyek sorozatával való programozás tökéletesítésének legnyilvánvalóbb” módja. S ha valóban „a nehéz munka a hexadecimális számjégyek alkalmazásával a mikroszámítógépre maradt”, akkor kíváncsiak vagyunk, hogy a maradék könnyű rész elvégzésére milyen gyakorlati módszert ajánlana a szerző egy, mondjuk csak néhány kilobájt nagyságú program elkészítéséhez és teszteléséhez. A következő — az assemblerről szóló — fejezetek szerencsére eloszlatják az olvasó kétségeit: létezik a tökéletesítésnek legalább egy további módja is.

Magyarországon is az elmúlt tíze-gyegnéhány évben számos assembler, ill. cross-assembler készült mikrogepekhez. Éppen ezért talán célszerű lett volna részletesebben (és pontosabban) foglalkozni ezzel a fontos témával. Az olvasó semmit nem tud meg az assemblerek belső működéséről, a szórványosan és hiányosan említtet direktívák — egy helyen „irányelveknek” (!) fordítva — tárgyalása még csak vázlatosnak sem tekinthető. Sőt még azok a szintaktikai szabályok sincsenek pontosan megfogalmazva, amelyek a VII. fejezet példáinak formáját meghatározzák.

E fejezetben egyébként a szerző szándéka bevallottan az, hogy egy képzetelbeli mikrogep utasításkezelését építse fel. Ezzel a didaktikai elvvel lehet egyetérteni vagy egyet nem érteni, de mindenképpen azt várjuk, hogy a szerző következetesen kitartson mellette. Az olvasóknak azonban túl gyakran támad az az érzés, hogy a szerző most nem vele együtt „találja ki” a legszük-

ségesebb vagy a leghasznosabban alkalmazható utasítástípusokat, hanem kész eseteket ismertet vele. Ha ez igaz vagy legalábbis részben igaz, az ez igaz végén utalni kellett volna a mintául vett típus(ok)ra.

Sajnos ebben a részben számtalan szakmai tévedéssel (pl. a bal és a jobb körbe léptetés nem ekvivalens a kettővel való szorzással, ill. osztással), hibás vagy felületes fogalmazási móddal, hibás hivatkozással, ábrával és példával találkozhatunk, és a szerző többször önmaga állításának is röviddel később ellentmond. Mindehhez járul, hogy a különböző címzési módok olyan zavarosan vannak bemutatva, hogy közöttük menthetlenül elvész az olvasó.

A szerző szerint „a miniszámítógépek programozók egy régi axiómája: 'Emlékezz arra, bármint csinálsz ma, az visszazárhat rád holnap'. De gondoljunk csak arra, hogy a mikroszámítógépek világában nincsen holnap...". Nem tudjuk, hogy ennek a könyvnek mi lesz a holnapi sorsa, de jelenleg formájában a didaktikai, szakmai és alaki hibái, tévedései miatt a kritikus nem sok jóvót tud neki jóslni.

Végezetül egy megjegyzés a kiadóhoz: a kiadványban nagy számban előforduló zavaró központosítási és gépelési hibák, az ábrák rajzolásának elkövetett figyelmenlenségük jelentős része a kézirat egyszerű átolvasásával kiküszöbölhető lett volna. (Csak egy példa: az irodalomjegyzék 24 olyan angol nyelvű címe közül, amelyben a „mikroprocesszor” vagy a „mikrogep” szavak jelennek meg, kilencben éppen ezek szerepelnek helytelen írással.) A könyvről borsos árát fizető olvasó megérdemelte volna ezt a „többletfordítást”.

VÁRGEDŐ TAMÁS

## **Számítástechnikai statistikai szekönyv, 1985.**

(Budapest, 1985.  
Statistikai Kiadó Vállalat,  
64 oldal. Ára: 12,— Ft.)

A népszerű statisztikai szekönyvcsalád új tagja az életünk minden területére betörő számítástechnika hazai alkalmazásáról nyújt átfogó tájékoztatást. Ismerteti a különböző teljesítményű, eredetű, életkorú és funkciójú számítógépek állományadatait és megosztását, a kapcsolódó gépi berendezéseket. Képet ad az alkalmazói tevékenység gazdasági eredményeiről és a programforgalomról. Erődes információkat tartalmaz a mini- és mikroszámítógépek elterjedéséről. Összehasonlító táblákat közöl a legfontosabb nyugat-európai adatokról, és prognosztizálja a következő években a miniszámítógépek és a szoftvertermékek várható piaci forgalmát.



## Negyedik generáció

A szakemberek előrejelzése szerint az elkövetkező két éven belül várható olyan „negyedik generációs” szoftverfejlesztések megjelenése, amelyek lehetővé teszik nagyon bonyolult, hibamentes, „testre szabott” programok készítését. Ezek előállításának időigénye hónapok helyett hetekre, egyes esetekben napokra zsugorodik.

Az így készült programok nem fognak többé előregéni. A generátorok segítségével dolgozó programozók képesek lesznek rendkívül gyorsan átvezetni a változásokat, ahányszor csak szükség van. Ez a karbantartási munkákat tízed-, esetleg századrészre csökkenti.

## Meghatszorozódik

Az amerikai Frost and Sullivan piackutató cég tanulmánya szerint az IBM gyártmányú személyi számítógépek és a velük kompatibilis mikroszámítógépek tulajdonosai a gép beszerzését követő év végéig átlagosan 1315 dollárt költenek szoftvertermékekre. Ebből 729 dollárt a beszerzésnél egyidőben, 586 dollárt egy éven belül fordítanak ilyen célra. Ugyanezen tanulmány szerint az IBM személyi számítógépekkel kompatibilis szoftver piacának volumene 1988-ra hatszorosára, 12 milliárd dollárra nő, elsősorban a beszerzést követő vásárlásoknak köszönhetően.

A vásárlások elemzése során bebizonyosodott a megfelelő dokumentáció fontossága: sok felhasználó nem vásárol mindaddig, amíg a dokumentáció alapján meg nem győződött arról, hogy a termék mire alkalmas. Érdekes kísérője a szoftverpiacnak a programkézikönyvek értékesítésének sikere is.

## Új robotron írógép

Új taggal bővült az NDK-beli robotron cég hazánkban is népszerű, intelligens elektronikus írógép-családjá. Az S 6130 típusú margarétafűző írógép egyesíti magában a hazánkban is kapható típusok előnyös tulajdonságait: az S 6011-től „örökölte”, hogy tárolója is van, s így a memóriában lehet javítani, az S 6010-nek megfelelően

pedig mikroszámítógéphez is csatlakoztatható nyomtatóként.

Az elegáns, szép formatervezésű S 6130 típusú írógép a V.24 illesztőn keresztül kapcsolható a mikroszámítógéphez. Nyomtatóként alkalmazva igen előnyös fejlesztése a gépnek, hogy automatikus papíradagolásra is képes.

## Festőrobotok

A gépjárművek lefűvéa porlasztott festékekkel közismerten az egyik legegészségtelegebb munka. Elsősorban azért jó hír, hogy az Ikarus gyár mintegy 2,3 milliárd forintért korszerűsíti festőcsarnokát. Itt a következő három évben mikroszámítógép vezérlésű automata gépsorokat és robotokat szerelnek fel, s meghonosítják a járműiparban alkalmazott legkorszerűbb festési eljárásokat. Az automata garantálják a korábbinál tartósabb, jobb minőségű festékesztést és a takarékos festékfelhasználást.

## A technika háza hálózat

Jelentős szerepet játszik a számítástechnikai kultúra terjesztésében Bulgáriában a technika háza hálózat. Ezeknek a nagyobb városokban épült műszaki központoknak a száma már 19-re emelkedett, és további 7 építése folyik. Valamennyi technika házában van mikroszámítógép-terem (lásd képünket) és több más laboratórium is; például a robotikai és folyamatszabályozási kísérletekhez is külön terem állnak rendelkezésre.

## Új lengyel mikrogép-család

Három gépkategóriát ölel fel az ELWRO cég új mikroszámítógép-családjá. A legkisebb elem az ELWRO 700, a házi számítógép kategóriába tartozik. Ennek mikroelektronikai bázisát a Z80-nak az NDK-ban gyártott funkcionális megfelelője képezi. Operatív tára háromféle változatban készül: 16, 32 és 48 kbájt méretben. A BASIC nyelven programozható gép periferiái a közönséges kazettás magnetofon és egy mátrixnyomtató. A gép ára 100 ezer zloty alatt van.

Elsősorban irodai alkalmazásokra szánják az ELWRO 600 jelzésű professzionális számítógépet, amelyet a hazánkban is már több ízben bemutatott ELWRO 500 típusú számítógép továbbfejlesztésekként alakítottak ki. Jelentősen csökkentek a gép méretei és ára is. A szoftverkompatibilitás biztosított, ugyanis ez a gép is a CP/M 2.2

verziójának adaptált változatát használja.

Az ELWRO 800 fontos jellemzője a modularitás. A két alapvető modul a 8 és a 16 bites, melyek a CP/M operációs rendszerrel, illetve az IBM PC-vel való kompatibilitást biztosítják. E modulok külön-külön vagy együtt is kiépíthetők számítógépre.

## Videoton TV-Computer

A Videoton Számítástechnikai Gyár forgalomba hozta az ideai BNV-n bemutatott TV-Computer-t. A 19 990 forintért kínált berendezés Z80 mikroprocesszoron alapul, 20 kbájt ROM és 32 kbájt RAM mellett 16 kbájtos, külön video RAM-ja van. Előnye a gépnek a betervezett négy darab bővítési kártyahely és a beépített összetett video- és RGB-kimenetek. Az utóbbiak segítségével professzionális monitorral is igen jól használható.





## 32 bites

### megamini

Ősszel mutatkozott be a nyilvánosság előtt a szocialista országok első 32 bites miniszámítógépe, a csehszlovák SZM 52/12. SZM 1505 jelzésű központi egysége teljes programkompatibilitást biztosít a VAX 11/780-nal. Ebből a 4 Gbájt operatív tárterület címzését lehetővé tevő gépből 1985 végéig még egy tucatnyi készül el, sorozatgyártása pedig 1986-ban kezdődik.

Mint ismeretes, Bulgáriában és a Szovjetunióban is folynak az e géppel kompatibilis számítógépek gyártási előkészületei.

## Minimális

### veszteséggel

A lap- és lemezszabászat optimalizálására szolgáló eljárás a termelés feladatból kiindulva olyan szabástérképet készít, amely a legkedvezőbb kihozatalt és minimális hulladékot ad. Az ezzel elérhető megtakarítás 25 százalék is lehet, de már néhány százalékos megtakarítás esetén is egy éven belül megtérülnek a ráfordítások.

A számítógépes program megadja a szabásterv számát, feltünteti a szabásterv szerint feldolgozandó táblák számát és méreteit, kiszámítja az adott szabásterv szerinti veszteséget. Az eredmények nyomtatásban és rajzos formában is megjeleníthetők. A CP/M vagy ezzel kompatibilis operációs rendszer alatt működtethető programot a Bütöripari Fejlesztési Vállalat dolgozta ki.

## Elektronikus

### levelezés

A munkájukhoz számítógépet használó üzletemberek az USA-ban 1984-ben több mint 100 millió alkalommal továbbítottak üzeneteket elektronikus levelezés útján, ami az előző évi forgalom mintegy kétszeresének felel meg.

Megvizsgálták az üzenet továbbítás költségeit: egy átlagos elektronikus levélre vetítve 86 dollárcent. Ez drágább ugyan a postai levélnél, melynek költsége 20 cent, sok esetben azonban az elektronikus levél mégis gazdaságosabb, például távolsági telefonbeszélge-

tést helyettesítve vagy körlevelek esetében.

Az elektronikus levelezés haszna azonban nem mindig mérhető dollárakban. Követett előlnei közé tartozik, hogy a levelek rendkívül gyorsan jutnak el a címzethez. Igen előnyös azok számára, akik nagyon elfoglaltak, sokat tárgyalnak, ezáltal nem lehet zavarni őket, és gyakran változtatják helyüket, tehát nem lehet őket megtalálni.

## Mérleg-

### ellenőrzés

A KSH SZŰV a PM Ellenőrzési Főigazgatósággal együttműködve programrendszer alakított ki a gazdálkodó szervezetek negyedéves – féléves – éves mérlegének mikroszámítógépes ellenőrzésére. A SZŰV szakemberei Commodore 64 számítógéppel szállnak ki a helyszínre, és néhány óra alatt elvégzik az ellenőrzést. A SZŰV országos hálózata lehetővé tette, hogy ezt a szolgáltatást minden megyére kiterjessék.

## Energia-

### szabályozás

#### a szállodákban

Az osztrák hitelből felépített szállodák korszerű számítástechnikai rendszerrel üzemelnek. Terminalokat helyeztek el a recepcióban, a pénztárban, az étteremben, a pénzváltásnál, sőt számítógép szabályozza az épületek energiarendszerét is. Az Atrium Hyattban, a Buda Pentában, a Fórumban és a Novotelben egy-egy Honeywell Delta 1000 típusú számítógép optimalizálja az energiafelhasználást. A géphez 1500 ún. adatpont (mérő, számláló, érzékelő, kapcsoló, vészjelző) csatlakoztatott, ameken át a gép figyelemmel kíséri a hőmérsékletet, a relatív nedvességtartalmat, méri a nyomást, a fogyasztást, megbíbasodás esetén automatikus hibajelzést, sőt szükség szerint vészjelzést is ad, továbbá az előre meghatározott értékeknek megfelelően kikapcsolja a fűtést, a hűtést, a légkondicionálást, a szellőzést.

A szállodák éves energiaköltsége 15–20 millió forint. Az energiamegtakarítást 10–20 százalékra becsülik, így a beruházás 2–3 év alatt megtérül.

## Fogpótlás

### mikrogéppel

François Duret, francia fogorvos a koronák, hidak és más fogprotézisek előállítására terén korszakalkotó újítást mutatott be: a fogpótlás elkészítéséhez alig egy órára van szükség.

A fogorvos tízezi kutatómunkával, más szakemberekkel közösen kifejlesztett egy apró videokamerát, amely néhány másodperc alatt háromdimenziós felvételt készít a beteg állkapcsáról és fogsoráról. Ezt a képet mikroszámítógép elemzi, és meghatározza a hiányzó rész körvonalait, majd a mikrogép által vezérelt szerszámmal elkészíti magát a fogpótlást. Az új módszer a hagyományosnál sokkal gyorsabb és pontosabb, így jelentősen javítja a szol-

gáltatás minőségét. Alkalmazásának következményeihez tartozik viszont az is, hogy teljes egészében kiküszöbölő a fogtechnikusi munkát.

## Gyógyszerek

### - programozott

#### receptre

Felújítják a Chinoin Gyógyszergyárnak azt a részlegét, ahol a hatóanyagból tablettákat, drázsákat állítanak elő. Ezen a munkahelyen mikroszámítógéppel vezérelt rendszerrel egészítik ki a gyártóberendezéseket, amelyek az előre beprogramozott receptúráknak megfelelően, emberi beavatkozás nélkül, nagy pontossággal mérik és adagolják a hatóanyagokat.

# μ '86

## ORSZÁGOS MIKROSZÁMÍTÓGÉPES TALÁLKOZÓ

A Budapesti Tavasz Fesztivállal egyidőben, 1986. március 14–19. között a BNV területén kerül megrendezésre az első Országos Mikroszámítógépes Találkozó. Célja a számítástechnika társadalmi méretű elterjesztésének segítése és a klub mozgalom továbbfejlesztése. Ennek keretében a következő rendezvényekre kerül sor:

- klubok bemutatkozása és a számítástechnikai amatőrök tapasztalatszerése. A HCC és a klubok keretében épített amatőr számítógépek kiállítása;
- a 2. Számítástechnika Mindenkié, a Számítástechnika Mindenkiért (SZMSZM) kiállítása. Ez a vállalatok, kisvállalkozások, szövetkezetek és GMK-k legújabb eredményeit kívánja felvonultatni, előtérbe helyezve az egyéni fejlesztési eredményeket;
- az otthoni felhasználásra, valamint bármilyen háztartási célú alkalmazásra készült amatőr szoftverek vetélkedője és bemutatója;
- a Szekszárdi Garay Gimnázium és a μM hagyományos játék- és oktatási program versenyén nyertes termékek közönségszemlájára;
- számítástechnika-történeti kiállítás az Országos Műszaki Múzeum és az NJSZT közös gyűjteményéből és még egy sor más eseményre, mint az elmúlt 25 év számítástechnikai tárgyu filmjeiből és videoanyagaiból készített válogatás vetítése, a számítógép és a zene kapcsolatának bemutatása, oktatási tapasztalatszerése stb.

Az amatőrmozgalomban még felkutatnivaló alkatrészek szerzésének is helyet kíván biztosítani a találkozó. Ehhez azok szervezői jelentkezését várja a rendezőség, amelyek ilyen célra hasznosítható alkatrészekkel rendelkeznek.

Az SZMSZM kiállításra postafordultával lehet még jelentkezéseket beküldeni. Az otthoni és háztartási alkalmazásokra készült szoftverek vetélkedőjére 1986. január 15-ig lehet jelentkezni.

A Találkozó és az SZMSZM kiállítás rendezésének címe: Budapest 5., Pf. 240. 1368.



## VÁLASZOK A TUDÁSPRÓBA KÉRDÉSEIRE

1. A mínusz 10, amely 1010-nek felel meg  $(-10 = 1 - (-2)^2 + 0 - (-2)^2 + 1 - (-2)^1 + 0 - (-2)^0)$ .
2. Van. A mínusz kettes alapú számbázisú rendszerben a  $-2$  értékű 10 szám ellentétje a  $+2$  értékű 110 szám, és  $+2$ -nek más előállításra nincs.
3. Ha  $n$  páratlan, akkor több a pozitív, ha  $n$  páros, akkor több a negatív szám.
4. Van. Pl. a mínusz kettes alapú számbázisú rendszerben a páratlan darabszámú számjeggyel felírt számok mind pozitívak, a páros darabszámú számjeggyel felírtak pedig mind negatívak. (Természetesen minden szám előtt el kell hagyni az értéktelen nullákat.)
5. Igen. Pl. ha a törtvesszőtől számtott ugyanannyiadik helyen az egyik számban 0, a másikban 1 áll, és pont ezek változnak meg, a zérus 1-re az 1 pedig zérusra. (Feltételezzük, hogy a gép a paritáshiba észlelések nem áll le, és nem is végez a számokon semmilyen javítást.)
6. Igen, például 111 és 110 különbsége ugyanannyi mint 101 és 100 különbsége. (Feltételezzük, hogy a gép a paritáshiba észlelések nem áll le, és nem is végez a számokon semmilyen javítást.)
7. Igen, ha pl. az 111 szám 110-re az 110 pedig 111-re változik. (Feltételezzük, hogy a gép a paritáshiba észlelések nem áll le, és nem is végez a számokon semmilyen javítást.)
8.  $1010 < 111 < 100$ .
9. Igaz. Adjunk konstrukciót egy  $b_n$  és  $c_n$  sorozatra, ha ismerjük az  $a_n$  sorozat elemeit!
10. Egy folytonos időtartományú rendszerek modellezésére és a modellek

vizsgálatára készített gépi nyelv, mely a vizsgálni kívánt rendszert kapcsolatábrájával, illetve annak táblázatos leírásával adja meg. Hazánkban pl. a CDC-3300-as típusjelű gépen sok éve sikeresen használják.

11. Nem igaz, még akkor sem, ha a két 8 bites mikroprocesszor azonos típusú.

12. Nem, mert e nyomtatókban a lézersugár festékanyag felvételére alkalmas nyomóhenger felületét éri, a papírt közvetlenül nem.

13. Nem, hanem apró tintacseppeket juttat a papírra.

14. Lehet, több különböző színű tinta alkalmazásával.

15. Nem mind. Van olyan nyomtató, amelyben a piezoelektromos jelenséget használják, van azonban olyan is, amelyben a tintacseppecske elektromos impulzus által keltett hő következtében kialakuló gőzbuborék létrejötté, nyomásnövekedés ill. kitágulás révén "lövődik" a papírra.

16. Van. Az LCS-2400 típusjelű CASIO gyártmányú pl. ilyen.

17. Igen. Pl. az MP 9060 típusjelű BULL gyártmányban. Sebessége 90(l) lap percnként.

18. Igaz, a festékanyag papírra juttatását mágneses nyomóhenger végzi, a porfesték rögzítést azonban hővel oldják meg.

19. A helyes válasz c).

20. Igaz, de ennek semmi köze a lézerhez. Ez elsősorban a felhasznált festékanyagtól és a szóban forgó műanyagtól függ. Ugyanez a helyzet a tintacseppecs nyomtatókra. Alkalmas "tintával" ezekkel a nyomtatókkal nemcsak műnyagra, hanem még fémlemezre is lehet nyomtatni.

## KORSZERŰSÍTÉS FŐKÖNYVI RENDSZERÉT ADATFELDOLGOZÓ MIKROSZÁMÍTÓGÉPPLEL

ROBOTRON A 5120-as berendezés, ügyviteli alkalmazásra kiegészítve 550 000,— Ft-ért eladó, vagy 170%-os leasing áron 13 negyedéves egyenletes törlesztéssel átadó, főkönyvi könyvelési programcsomaggal együtt. Referencia helyen működés közben megtekinthető.

Munkaügyi és bérigényeléstől a magasrakrtár iránításig – egyedi igények figyelembevételével – számos programtermékünkkel állunk az érdeklődő felhasználók rendelkezésére.

Az igény megfogalmazásától az alkalmazói rendszerek betanításáig, a gépkiválasztástól annak bérbeadásáig teljes körű szolgáltatást nyújt Önnek a



Címünk: 1115 Budapest, XI. vagy 1501 Budapest  
Pf. 9.  
Ballagi Mór u. 14. II. kapu  
Telefonszámunk: 660-966/10 mellék

## Új termékünk a VT 20/IV. számítógépre kifejlesztett L/BASIC ADATBÁZISKEZELŐ RENDSZER.

- Egyszerre csak egy terminált üzemeltet? –  
*ez esetben nincs szüksége L/BASIC-re.*
- Kevés adattal dolgozik? –  
*a feladat megoldható L/BASIC nélkül.*

## HA IDÁIG ELJUTOTT, AZ L/BASIC AZ ÖN RENDSZERE!

Kiterjesztett BASIC utasításkészlete és a diszkprocesszorba ültetett adatbáziskezelője segítségével

- biztosítja több terminál egyidejű működését ugyanazokon az adatokon (record-lock),
- támogatja nagy adatbázisok gyors lekérdezését, módosítását, feldolgozását,
- naplózással véd az adatbázis épségét.

SZÁMÍTÁSTECHNIKAI MŰSZAKI FEJLESZTŐ KISSZÖVETKEZET  
Budapest XIV., Kőszeg u. 4. Telefon: 831-805 Ügyintéző: Maros Péter



# 100 feladat - 1 programmal

A CSB-901 programrendszert  
számítástechnikai ismeretek nélkül  
is azonnal tudja használni



Nagy számú  
referencia!

COMMODORE 64,  
VT-20/A, VT-20/IV, VT-16,  
MO8X,  
TZ-80,  
TRANSMIC 8,  
MULTI CENTER,  
APPLE II. számítógéprendszerein, hálózatain.

A CSB-901 nem igényel szakismeretet,  
speciális tanfolyamot,  
külön kiképzett számítógépkezelőt,  
használatát bárki gyorsan elsajátíthatja.  
Alkalmazásával órák alatt eredményt érhet el!

## TARTSA NYILVÁNTARTÁSAIT CSB-vel NAPRAKÉSZEN!

Fejlesztő: Automatika Ipari Kiszövetkezet  
„ALTECH”  
Alkalmazott Számítástechnikai Szakcsoport

Részletes felvilágosítás a 260-534-es  
telefonszámon.



**Kérjen referenciát, ismertetőket, bemutatót!**





## MA BIZTOSÍTJUK A HOLNAP TECHNIKÁJÁT

- ALACSONY SZÁMÍTÓGÉPÁRAK
  - ALACSONY ÁRFEKVÉSŰ KULCSRAKÉSZ RENDSZEREK
- Kulcsrakész rendszerek vásárlása esetén árkedvezmény a számítógépekre
- SZÁMÍTÓGÉPEK ELADÁSA ÉS LIZINGJE.
  - GARANCIÁLIS ÉS GARANCIÁN TÚLI SZERVIZ.

### KOMPUTER BOLTUNKBAN:

- állandó ügyfélszolgálat
  - szakszerű kiszolgálás
  - ingyenes hardver, szoftver tanácsadás
  - folyamatos bemutatók
  - szakkönyvek és szoftvercsomagok eladása
  - különböző számítógépek és programcsomagok oktatására jelentkezés (Wordstar, dBase II.III Lotus 1-2-3 stb.)
- Szolgáltatásainkkal állunk ügyfeleink rendelkezésére.