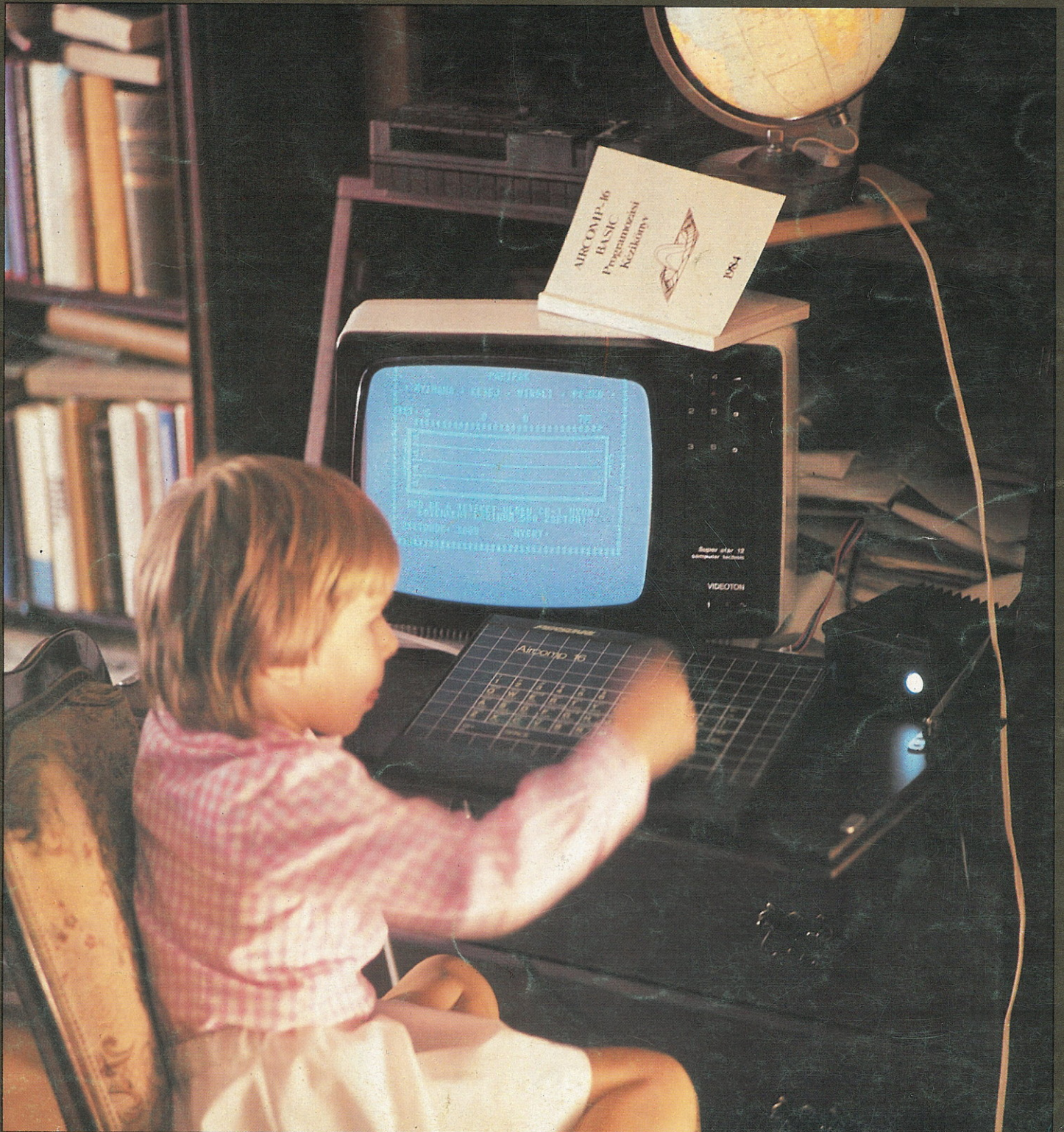


A NEUMANN JÁNOS SZÁMÍTÓGÉP-  
TUDOMÁNYI TÁRSASÁG LAPJA

1984/5



MIKROSZÁMÍTÓGÉP  
MAGAZIN



**IZOT 0220 M2 mikroszámítógép**  
**Kulcsra kész rendszerek.**

**OKISZ** Szervezési  
és Számítástechnikai Vállalat





## A Neumann János Számítógéptudományi Társaság lapja

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

**A szerkesztő bizottság  
vezetője:**  
Kovács Győző

**Munkatársak:**

Broczkó Péter  
(hírek)  
Buday György István  
(személyi számítógépek)  
Garádi János  
(agyafúrmány)  
Jakab Ágnes  
(ember-gép kapcsolat)  
Kovács Győző  
(az olvasó írja)  
Lindner László  
(sakkprogramozás)  
Nacsa Sándor  
(termékismertető)  
Pataki Ernő  
(programozástechnika)  
Petróczy Judit  
(új könyvek)  
Pogány Csaba  
(alkalmazástechnika,  
tanfolyam)  
Sieben Nándor  
(százlábú)  
Simonyi Endre  
(klub)  
Takácsy Ildikó  
(favágás)  
Varga András  
(iskola - számítógép)  
Vass Nándor  
(alkalmazások)  
Votisky Zsuzsa  
(játékprogramok)

**Felelős szerkesztő:**  
Könyves Tóth Pál

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

**Kiadja:** a Lapkiadó Vállalat  
**Felelős kiadó:**  
Siklósi Norbert vezérigazgató  
**Kiadóhivatal:**  
Budapest VII., Lenin krt. 9-11.  
**Postacím:**  
1906 Budapest, pf. 223.  
Telefon: 429-350, 221-285

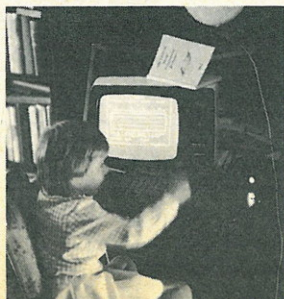
**Terjeszti:** a Magyar Posta.  
Előfizethető  
bármely postahivatalban,  
a kézbesítőknél,  
a Posta hírlapüzleteiben  
és a Posta  
Központi Hírlap Irodában  
(Budapest V.,  
József nádor tér 1.,  
postacím: 1900 Budapest)  
közvetlenül  
vagy postautalványon,  
valamint átutalással  
a PKHI 215-96161  
pénzforgalmi jelzőszámra.  
**Előfizetési díj:**  
egy évre 168,- Ft,  
fél évre 84,- Ft.

**Szedte:**  
a Nyomdaipari Fényszedő  
Üzem  
(847655/09)

**Nyomás:**  
Petőfi Nyomda, Kecskemét,  
Külső Szegedi út 6.  
(84.41780)  
Telefon: 20466  
**Felelős vezető:**  
Ablaka István igazgató

**ISSN**  
0236-6088

**Címképünk:**  
Az AIRCOMP 16 személyi  
számítógép.  
Personal GT 2040 Budaörs  
Molnár Pál utca 1.



## Tartalom

Tehetséggondozás	2
Miniprogramok	13
Miből lesz a cserebogár?	27
Adok - veszek - cserélek	27
Kell-e nekünk számítógép-múzeum?	38
Kinek lesz igaza?	41

### ISKOLA - SZÁMÍTÓGÉP

Lelkes tanárok - tehetséges diákok	4
KöMaL gyakorlatok	6
Beírás - futás közben	7

### TANFOLYAM

Alapozás VI.	8
--------------	---

### PROGRAMOZÁSTECHNIKA

Programozási fogások	14
Rendezések Commodore 64-en	15

### TERMÉKISMERTETŐ

Üzleti grafika	16
Hivatásos amatőröknek	19
Tényleg olyan rossz gép?	21

### ALKALMAZÁSOK

Személyi számítógép a közgazdasági munkában	26
---	----

### µKLUB

Építsünk számítógépet! IV.	30
Chip-problémák	32
Mátrixnyomtató-illesztés	33

### JÁTÉKPROGRAMOK

A negyedik helyezett játékprogram	34
Az ötödik helyezett játékprogram	35

### FAVÁGÁS

Függvényábrák és if nélküli definíciók	37
--	----

### ALKALMAZÁSTECHNIKA

Gyorsítási fogások	39
--------------------	----

### AZ OLVASÓ ÍRJA

HÍREK, ÉRDEKESSEGEK	44
---------------------	----

### SAKKPROGRAMOZÁS

Lépésről lépésre	45
Nemzetközi konferencia Londonban	46

### KÖNYVEK

	47
--	----

# Tehetséggondozás

**„Amit mi nevelésnek nevezünk, az nem egyéb, mint az agy programozása abban a korai szakaszban, melyben az még formálható. Az emberiség jövője a neveléstől függ, attól a programozó rendszertől, amely tetszés szerint változtatható. Az emberi történelem lényegében ennek a programozásnak a fokozatos változását tükrözi, és hogyha bárki összehasonlíttja magát egy vad emberevővel, beláthatja, hogy az egyetlen lényegbevágó különbség kettejük között abban az eltérő nevelési programban van, melyen mindketten keresztülmentek. Ebből az következik, hogy a nevelés az emberiség egyik legfontosabb tevékenysége.”**

(Szent-Györgyi Albert)

Július van, 40 °C árnyékban, forgatom az ÉS legújabb számát. Azután elolvasom ismét, a korábbi példányokat is, azzal a szándékkal, hogy még egyszer átnézzem a tehetséges gyermekek érdekében kirobbant sajtóvita anyagát. A ... tehetségvédelem, ... elit-képzés, ... tehetségevel és más hasonló címmel megjelent cikkek olyasmivel foglalkoznak, ami a számítástechnikai közvéleményt is mélyen érinti. Ezért úgy vélem, hozzá kell szólnunk a vitához, még akkor is, ha a hozzászólás talán a vita lezárása után jelenik meg.

Talán annyit jegyeznek meg előljáróban, hogy a vita már jellegzetesen magyar. Az alpmotívum (kell-e a tehetséges gyerekekkel törődni) már eltűnt; egyre több a személyeskedés (Bekekataizmus és társai). A vita stílusát nem követném, megpróbálom a kérdést más oldalról közelíteni.

1. Azt hiszem, hogy nem lehet vita tárgya, kell-e a tehetséges gyerekeket támogatni. Minden országnak, főleg a kis országoknak, ahol az abszolút számokat tekintve valószínűleg kevesebb az igazán tehetséges gyerek, mint egy nagy országban, a társadalom „kutya kötelessége” a tehetségesek kiemelése és gondozása.

2. Hol kell a feladatot megoldani? Az elit-iskolákban? Már a fogalom is furcsa. A vitázók cikkeket olvasva az ember azt hinné, hogy egy közép- vagy általános iskola úgy lesz elit-iskola, hogy valaki elhatározza: „az X iskola pedig szeptember 1-től elit-iskola lesz!” Az ember hajlamos azt hinni, hogy az elit-iskolákat szervezni kell. Holott nem hatósági engedély kell az iskola minőségi munkájához, hanem kiváló és lelkes tanárok, úgy is mondhatnám, hogy tanáregyéniségek; ha jól érttem, ezt mondja Laki Pál is.

3. Mít tudjon az elit-diák, akit az elit-iskolából az elit-tanár bocsát újtárra? Az elmúlt néhány évben a megítélés – enyhén szólva – néhány alkalommal alaposan változott. A mai elit-diák mindent kiválóan tud (kitűnő rendű), de valószínűleg egyetlen tárgyat sem igazán. Semmiképpen sem elit-diák az olyan, aki igen kiváló például matematikából és fizikából, esetleg csúcsteljesítményekre képes a számítástechnikában, de közepesnél nem jobb a humán tárgyakból. Persze a fordítottja is igaz; egy fióka Ady Endre vagy Szép Ernő, ha nem hozza a jelet a természettudományi tárgyakból, akkor

első nekifutásra valószínűleg nem jut be a bölcsészkarra, olyan a hátránya a felvételi rendszerünkben.

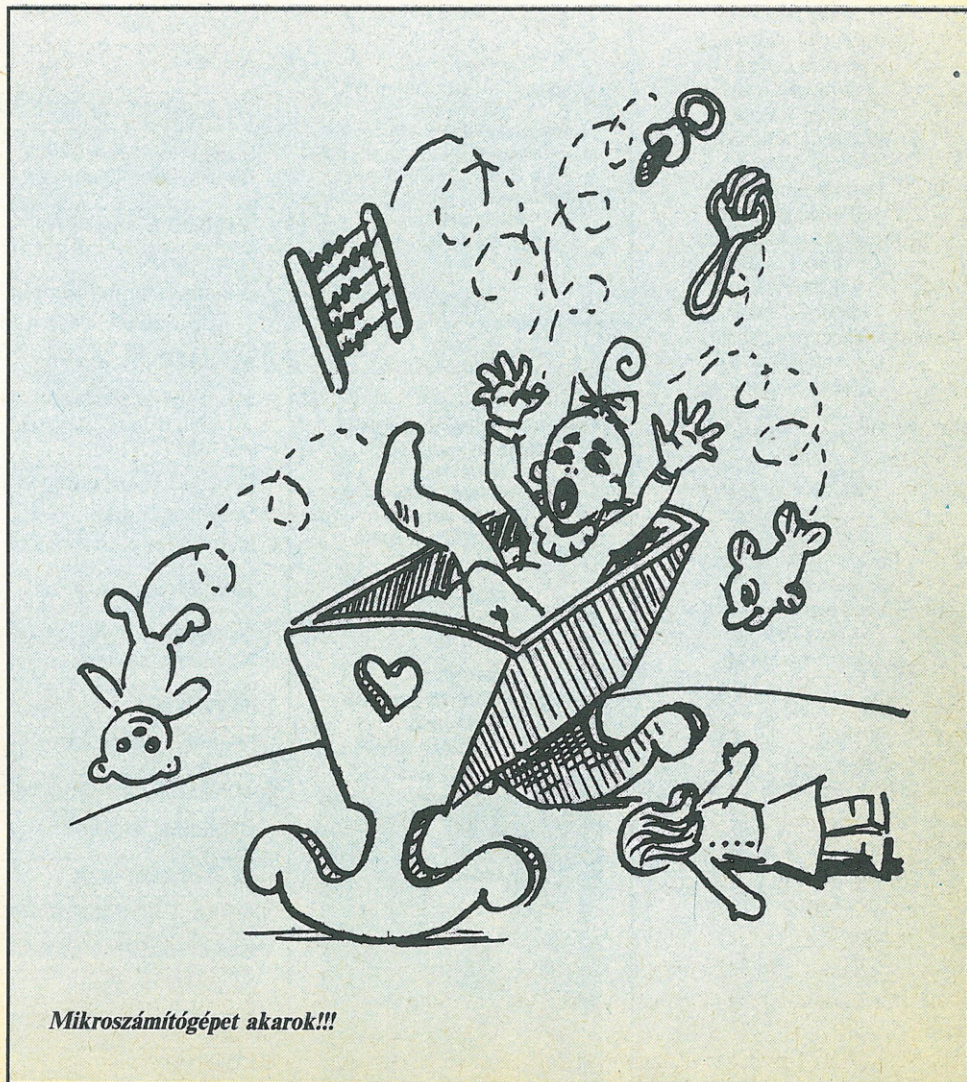
4. Ha tetszik, ha nem, észre kell vennünk, hogy az elmúlt 30–50 évben körülöttünk a világ nemcsak politikai értelemben változott, de – hála a telekommunikációs eszközöknek, illetve a technika fejlődésének – megváltoztak, vagy ha úgy tetszik, eltolódtak a tudásszintek is, az ifjabb generáció javára. Az már közhely, hogy egy általános iskolás gyerek többet tud ma a tudomány eredményeiről, ha csak a tévé ismeretterjesztő előadásait nézi, mint a negyvenes-ötvenes években mi tudtunk az érettségi után.

Arról pedig a saját tapasztalatomból tudok számtalan példát hozni, hogy egy általános iskolás diák (nem beszélve a középiskolásokról) annyi számítástechnikai ismerettel, amivel a hetvenes évek közepén (!) diplomát lehetett szerezni, ma egy diák-válogatóversenyen sem vehetne részt.

5. Ha az iskolarendszert tekintjük, akkor viszont meg kell állapítanunk, hogy a közoktatásra a társadalomban bekövetkezett változások gyakorlatilag nem hatottak. A középiskola ma is olyan, mint fél évszázada volt. Nem lehet változásnak tekinteni a fakultációt; a jelenlegi kötelező, szabadon választott formája csak tüneti kezelést jelent, de nem gyógyítja a kórt. A közoktatás ma nem gyógyszeres kezelést, hanem operációt igényel.

6. Azt hiszem, hogy a mai középiskolásokat úgy kellene tanítani, mint azelőtt az egyetemistákat. Azt kellene tanulniuk, ami választott pályájukhoz szükséges, és megengedni, hogy a jövődől szakmájukhoz közvetlenül nem használható tantárgyakat vizsgakötelezettség nélkül, csak hallgathassák.

Lelki füleimmel már hallom a „feszítsd meg őt” kiáltást, azoknak a kórusát, akik végre elérték, hogy érettségi vizsgát kell tenni humán és természettudományi tárgyakból egyaránt, majd az ebből számított pontokat egyformán viszi magával a bölcsész- és a mérnökjelölt a felvételi vizsgára. Én azt hiszem, az általános



Mikroszámítógépet akarok!!!

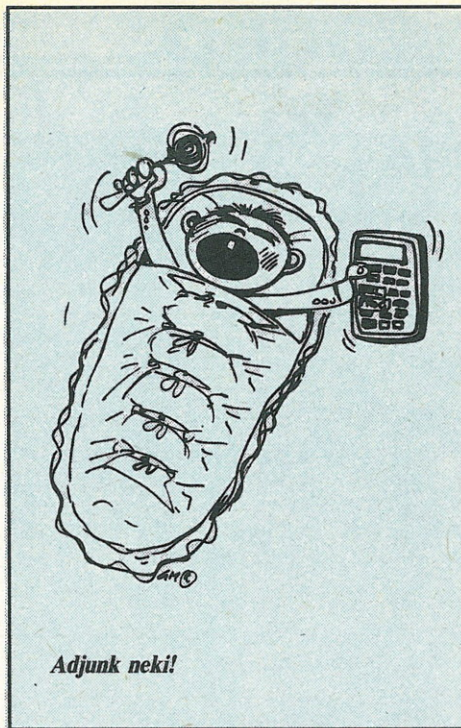
műveltség ilyen erőszakolt megszerzetése ma már elfogadhatatlan. Nem hiszem, hogy azt a gyereket, aki például kiváló matematikus, szereti és érti a fizikát, ragyogó programokat ír az iskolai számítógépeken, a műveltség szent nevében arra kell kényszeríteni, hogy a megfelelő érettségi pontszám érdekében jelesre vizsgázzon a közismereti tárgyakból is. Én az ilyen módon való műveltségterjesztést formalitásnak érzem; legfeljebb statisztikai adatnak jó, valóságos eredményt így elérni nem lehet.

Talán nem hat szerénytelenségnek saját példámmal érvelni az elmondottak mellett. Amióta az eszemet tudom, mérnöknek készültem, és sohasem lelkesedtem a humán tárgyakért. Mégis gimnazista korom legszebb emlékei a délutáni irodalmi órák Béla Pál tanár úrral, aki megismertette velünk Babitsot, Móriczot és a magyar irodalom nagyjait (akkor ugyanis nem volt önképzőkör, azt az illetékesek károsnak ítélték). De ugyanúgy felejthetetlenek az esti lemezhallgatások Pataki igazgató úr irodájában, Létay tanár úr (fizikus!) avatott irányításával. Igaztalan lennék, ha nem szólnék Bodrogi tanár úrról, aki az iskolai énekkarban plántálta belénk a magyar népdal máig is élő szeretetét. Ezeket az órákat tanáraink délután tartották, a részvétel nem volt kötelező, mégis sokan áldoztuk szabad időnkét – most már tudom – humán műveltségünk alapozására. Ha a negyvenes évek végén úgy is lehetett, ma így kellene tanítani. Persze ehhez valóban kiváló pedagógusokra van szükség.

7. Nem elég ma már egy szakembernek, ha csak az egyetemen kezd választott szakmájával foglalkozni. Például a számítástechnikában olyan magas a követelmény, hogy az erre a pályára készülő diáknak a középiskolában is sok időt kell szakmai ismeretek megszerzésére fordítani. És itt van a „középiskola csapdája”! Ha nem teszi, akkor valószínűleg kiválóan ért sok mindenhez, jelesen érettségizik valamennyi tárgyból, valószínűleg fel is veszik az egyetemre, de hátránnyal indul a szakmai versenyben. Ha a szakmával foglalkozik, és ne adj’ isten elhanyagol egyes tárgyakat, akkor majdnem biztosan megjósolható, hogy pályája megtörik, ha mégoly kiválóan is szerepel a felvételin, a gyenge érettségi bizonyítvány miatt esetleg csak egy év múlva kerülhet be az egyetemre.

8. Azt mondják, hogy az Országos Középiskolás Tanulmányi Versenyt azért találták ki, hogy a tehetségeseknek legyen lehetőségük a kiugrásra. Számítástechnikából csak a szakközépiskolások versenyezhetnek, pedig a nem szakközépiskolákban is rengeteg tehetséges gyerek gyötri a számítógépeket. Ők azért nem indulhatnak, mert a számítástechnika nem órarendi tárgy a középiskolákban. Talán ezért tanulják olyan sokan és lelkesen! És hányan tudnak „kiugrani” évente? Hárman, tízen? A verseny megnyeréséből származó előny egyenlő-e az érettségi bizonyítvány heterogén eredményéből származó hátránnyal? Aligha. Így a tehetségesek kiemelésére a megoldást máshol kell keresni.

9. Nem foglalkoznék a szellemi munka társadalmi értékelésével, a bér- és jövedelemviszo-



nyokkal. A vitában részt vevők a mai lehetetlen helyzetet minden oldalról bemutatták.

Nem foglalkozom vele azért sem, mert a számítástechnikát bolondul szerető és nagyon értő gyerekeket túlnyomórészt nem az anyagiak, hanem a szakma szépsége vonzza a pályára, ezért nem szeretném, ha elkallódnának. Pedig, ha nem jutnak be az egyetemre, ha nem kapják meg a szükséges szakmai irányítást és kritikát, akkor előbb vagy utóbb ez lesz a sorsuk, és akkor se általánosan, se szakmailag nem lesznek művelt emberek.

10. Az az érzésem, hogy a jelenlegi inkább formális, mint tartalmi egyetemi felvételi rendszert egyelőre nem lehet megváltoztatni. Pedig ne jó, ti. ha jó lenne, valószínűleg nem lenne évről évre olyan sok rosszul tanuló, esetleg lemorzsolódó egyetemi hallgató. Áthidaló megoldást javasolok: legyen kétlépcsős a felvételi rendszer, és így kapjon szabad utat az egyoldalúan képzett tehetség is, függetlenül attól, hogyan tanult a középiskolában. A számítástechnika köréből szeretnék példát mondani. Ismerek olyan középiskolás diákokat, ők nyerek általában a Neumann Társaság, a KISZ, a számítástechnikai intézetek versenyeit, akik a számítástechnika különböző területeit már a középiskolában is egyetemi szinten megismerték. (Feltételezem, hogy hasonló tehetséges diákok akadnak a kémiában, fizikában, de a közgazdaságtanban, sőt még talán az orvosi biológiában is.) Maradva a számítástechnikánál, kapják meg a jogot az egyetemre, hogy egyes fakultásokon előfelvételt tarthassanak, amelyre természetesen bárki jelentkezhet. Az előfelvételi anyaga jóval magasabb legyen, mint a középiskolás anyag, véleményem szerint az egyetem I-II. évfolyamán tanított tantárgyakból lehetne vizsgázni. Én biztos vagyok benne, hogy néhány tíz diák, talán a legodaválók, sikeresen vennék az akadályt. Ezeket a diákokat függetlenül attól, hogyan érettségiztek, fel lehetne

venni az egyetem adott szakára. Sőt, ha a vizsgán igazán kiemelkedő tudást mutatnak, akár a II. vagy (horribile dictu!) III. évfolyamra is beiratkozhatnának. (Manapság ugyanis nemcsak bizonyítani kell az egyetemeken az előírt anyag megfelelő ismeretét, de végig is kell ülni az éveket.) Tudom, ezzel a lehetőséggel az első évben fel nem vett diákok, ha egy évig keményen tanulnak, behozhatják az elvesztett évet, de miért baj ez? Ha megpróbálunk eredményre optimalizálni, tehát, hogy kiválóan képzett fiatal szakemberek kerüljenek ki az egyetemekről, akkor ilyen módszerekkel valószínűleg hamarabb célt érhetünk.

11. Végül néhány szót a diplomásokról. Évek óta van szerencsém főiskolásokot állami vizsgáztatni. Nem sok, de nem is kevés az egyik évről a másikra éppen csak átcúsúzó, évenként több utóvizsgát is összegyűjtő hallgató. Nyilvánvaló, hogy az ilyen gyenge tanuló az állami vizsgán is éppen csak átcúsúzik, és kap 2, esetleg 2,5 átlagú bizonyítványt. Nem biztos, de nagyon nagy a valószínűsége, hogy a főiskolákon és az egyetemeken folyamatosan gyenge eredményt elérő hallgatók a szakmában sem állják meg a helyüket; előbb vagy utóbb az adminisztratív beosztások felé iparkodnak, tehát elvesznek a szakma szempontjából.

Meg kellene próbálni az egyetemi diploma erkölcsi tekintélyét is visszaadni. Ha rajtam múlna, akkor valószínűleg nem engedném diplomázni azokat a hallgatókat, akik nem értek el 3,5-ös vagy 3-as átlageredményt az egyetem utolsó két évében. A műszaki életben ismert az üzemmérnöki képesítés. Kapjon ilyen bizonyítványt az, aki gyengébben tanult az egyetemen, vagy az is, aki diplomázott, de ott nem ért el jó eredményt.

Amikor erről szakembereknek beszéltem, azal érveltek, hogy „akkor a szegény hallgató végigülte az egyetemet, és a végén nem kap egyetemi diplomát”. A válaszom: az egyetem is verseny. A sportban is így van, hogy a helyezetteket minősítik – akik lemaradtak, rosszabb az eredményük, azok nem kerülnek be a döntőbe.

Egy ilyen intézkedéssel lenne „hajtás” az egyetemen, csak a tehetséges érvényesülne, a kevésbé tehetséges pedig alacsonyabb képesítéssel kerülne ki az Életbe. Ma sajnos az a helyzet, hogy aki egyszer bekerült az egyetemre, az el is végzi. Ha akarja, ha nem.

Zárszó. Nem vagyok az oktatásügy szakembere, így nemigen tudok a tehetséggondozásra biztos recepttel szolgálni. A számítástechnika talán a legjobb példa a tehetségekkel való rablógazdálkodásra, vagy inkább a tehetségekkel való foglalkozás szükségességére.

A fenti gondolatokkal – most jó a közhely – nem törekedtem teljességre. A teljes megoldást az oktatásügy szakembereinek kell megtalálni. Mi – a vitázók – csak jó, rossz ötleteket adhatunk a probléma egyes elemeinek megoldására. A személyeskedés árt az ügynek, csak a tehetséges gyerekekért aggódó, józan gondolatok vezethetnek el a megoldáshoz. Ezért kell a vitát folytatni!

## NEMCSAK OKTATÁSRA

# Lelkes tanárok – tehetsége

Mikor 78-ban egy PTK-1096-os gépecske megjelent iskolánkban, tudtuk, hogy valami megindult; de azzal is tisztában voltunk, hogy a számítástechnika napi gyakorlata helyett e gép „bűvölése” csak ízeletese az igazi számítástechnikának. Együttal megfogalmaztuk az azóta is megdönthetetlen alaptörvényünket, mely szerint akárminck van, az még mindig kevés.

A következő évben lett egy HUNOR 252-es asztali számológépünk, amelyen megtanultuk, hogy számolni billentyűk nyomogatásával is lehet. (Ne tessék mosolyogni! Ebben az időszakban komoly műszaki emberek még azért nem vettek vagy vétettek zsebszámológépet, mert nem hitték, hogy ezek a „játékszerek” valóban 24 műveletre képesek.)

1980-ban kaptuk az első számítógépet. A HT-680X színes grafikai egységével ma is laboratóriumunk legimpozánsabb jelensége. Hónapokig vallattuk, csodáltuk, nem tudtunk betelni vele. Gyűjtöttük a gyér számítástechnikai szakirodalmat és az ősz hajszálakat, hiszen autodidakta módon kellett megtanulni a gép nyelvét. 1981-ben telexillesztéssel a programok kiírása is megoldódott, de ez sem tudta feleltetni alaptörvényünket: egy gép nem gép.

1981-ben vettünk 15 darab PTK-1050-es kalkulátort, és azt játszottuk, hogy egyszerre sokan, külön gépen dolgoznak. Az eredmény: kicsit akadózva, de már programnyelven beszéltünk. És a lényeg: fél osztálynyi diák kezébeadtunk számítógépet. (Számítógépnek nevezük azt a gépet, amely különféle műveleteket egy előre meghatározott program szerint elvéggez.)

1982. Commodore VC-20. Profimunka. Kezes, intelligens gép, melyet egy bővítő modulal 1983 tavaszára rajzolni is „megtanítottunk”. Ez év szerzeménye továbbá egy RA-03-as rajzdigitalizáló, amely fényes jövő elé néz a mi rajzos szakmánkban.

A számítástechnika, a számítógép, ez a megállíthatatlan folyamat közben iskolánk falain kívül is érezette hatását, és 1983-ban megkaptuk az első iskolaszámítógépünket.

Pár hónappal ezelőtt kiíró, modulbox és hajlékonylemez tette komfortosabbá Commodore-unkat.

A dolgok „kemény” oldalához tartozik a számítógép-laboratórium kialakítása 1980-ban, majd három év múlva egy nagyobb helyiségbe költöztetése.

Az eddig elmondottak iskolánkban kb. 2 millió forintjába kerültek – kivéve természetesen az iskolaszámítógépet –, tehát teljes egészében saját fejlesztésünk a számítástechnikai apparátus.

### Oktatás

A HT-680X üzembe helyezésével egyidőben meg kellett fogalmaznunk, hogy mire lehet és mire akarjuk a mikroszámítógépet használni.

Mi sem voltunk kivételek, mi is kezdtünk annak idején bosszankodni, amikor léptenyomon azt hallottuk, hogy lassan már a kaskukorélást is számítógép vezérli. Nyilván a propaganda hibája is, hogy ha valamilyen szerkezetben megjelent egy mikroprocesszor, már rögtön számítógépről beszéltek.

Két dolgot szerettünk volna kezdetben: szokjanak hozzá a gyerekek ahhoz, hogy bizonyos számításokat számológéppel lehet, illetve kell elvégezni, és hogy egy komoly számítógép birtokában feltérképezzük azokat a területeket az oktató-nevelő munkában, ahol a gépet alkalmazni tudjuk.

A PTK-1050-eseket kezdetben (és azóta is) főképpen a műszaki-szakmai tárgyak számítási feladatainál használtuk. Természetesen első pillanatban éreztük a „bűvös kocka”-effektust. Logarlécen és függvényábrán nevelkedett tanáraink szépen feldolgozták a feladatot, programot készítettek, majd elhűlve tapasztalták, hogy diákjainknak szűk fél óra elég volt arra, hogy a miniszámítógépet kézbe vegyék és virtuóz módon kezeljék. E sorok szerzője egy órán keresztül szerkesztette egy géptermi mérés értékelő számításának programját, amelyet másnap egy 17 éves diákja úgy „fésült” meg, hogy a programlétszámát húsz százalékkal csökkentette.

Természetesen nekünk is meg kellett küzdenünk azzal a közismert aggálllyal, hogy a géppel való számolás – pláne kész programokkal – a gyerekek számolási készségét visszafejleszti. Itt azért még nem tartunk. De egy mérési sorozat részeredményeit „gyalog” kiszámítani egy fél délután munkája volt eddig, és a végső eredmény – például egy jelleggörbe megjelenése – már szinte érdektelenségbe fulladt.

A PTK-1050-esekkel gyorsan átugrottuk ezt a didaktikai állóvizet. A gond csak az volt, hogy diákjainkat majdnem erőszakkal kellett a következő órára elküldeni, mert nem akarták a gépet letenni. És ez már több volt, mint hasznos játék.

Természetesen minőségileg más problémát jelent a programozható zsebszámológépek órai alkalmazása és egy olyan nagy teljesítményű gép „hadrendbe” állítása, mint a HT-680X. Még nem volt meg ez a gép, amikor a vele megoldandó feladatokat már átgondoltuk. Nem arról volt szó, hogy mindenre „ráhúzzuk” a számítógépet. De iskolánk jellege eleve olyan, hogy természetudományi és műszaki tárgyaink szinte sugallják a számítógép alkalmazását.

Úgy éreztük, hogy felkészülten fogadjuk tehát első gépünket. Kialakult közben egy olyan csoport a kollégák között, amely érdeklődött a számítástechnika iránt. Ez az érdeklődés az óvatos ismerkedési szándék és a teljes erőbedobással való tanulás, kísérletezés, operátori gyakorlatyszerzés szélsőségei között mozgott. Egy dolog viszont közös volt: a gép jelenléte inspirált minket az egyes feladatoknál arra, hogy átgondoljuk, miként lehet bevonni azt a megoldásba.

Természetesen nem tekintettük megváltónak, csodaszernek. Tudjuk, hogy nem dolgozik helyettünk. Lehetőségeit csak kemény, hosszú tanulás eredményeként tárja fel. Nem mindenki vállalta ezt az árat. Konkrétan: az első eredményeket tanár kollégánk (jelenleg a számítástechnikai labor vezetője) és két diákja érte el. Minden dokumentáció, segítség nélkül először BASIC nyelven írták a programokat, amelyek között már kezdetben volt kifejezett oktatóprogram is. Sokszor figyelmeztette őket a pedellus zord tekintete, hogy későre jár.

Szívós munkájuknak terjedt a híre. A szakfelügyelet a futó fizikai, matematikai, műszaki programok megtekintése nyomán szorgalmazta, hogy más intézetek szakos tanárai is ismerkedjenek meg ezekkel az új módszerekkel, lehetőségekkel. Több továbbképzést tartottunk ezután, és tartunk ma is. Egyre markánsabb formában működik a számítástechnikai szakkör, felszínre hozva több diákunk tehetségét.

Nemsokára már tanfolyamok rendezését is tudtuk vállalni, természetesen csak a számítástechnika alapjai címen, hiszen a VC-20-szal kettőre növekedett gépparkunk operátori gyakorlásra nem adott lehetőséget. Ha az iskolaszámítógép-program keretében hozzáánk került HT-1080Z-t további gépek követik, és szándékunk szerint tovább fejlesztjük laboratóriumi munkát, akkor elérhetjük az operátorképzés minimális tömegszerűségét.

Az egységes alapú, középiskolára épülő, naplapi tagozatos technikusképzésben (V. évfolyam) e tanévtől kezdve a tananyagba is beépítettük a számítástechnika oktatását. Igaz viszont, hogy a javaslatunk elfogadása után előírt és az automatika tantárgyba beépített számítástechnikai elméleti anyag részéhez kapcsolódó, heti háromórás számítógépes laboratóriumi gyakorlat csoportbontásos megvalósításához jelen pillanatban 13 gépre lenne szükségünk.

Néhány szó eredményeinkről. Egy volt diákunk már számítógép-programozóként dolgozik, egy másik pedig sikeresen felvételizett a KLTE matematikus szakára. Az első országos számítástechnikai versenyen két harmadikos tanulónk I. dicséretben részesült. Három jól működő szakkörünk mellett elismerésnek tekintjük, hogy iskolánkat szakközépiskolai számítástechnikai továbbképzési bázisnak tekint a megyei tanács, és a mi feladatunk a szakközépiskolai tanárok ez irányú továbbképzése.

### Kutatás

Gépeink teljesítőképességét és intelligenciáját ismerve tudjuk, hogy luxus lenne ezeket csak számítástechnikai feladatok elvégzésére és a számítástechnika oktatására használni. Ezért különféle programok készítése mellett kezdettől fogva kutatjuk a számítógép alkalmazásának egyéb lehetőségeit is. Korán kiderült, hogy bizonyos hardverfeladatok megoldására is fel kell készülnünk. Néhány már

## s diákok

megvalósult és tervezett fejlesztést említünk az alábbiakban.

A HT-680X színes televíziós számítógéprendszer. Az első lépés tv-kamera szinkron illesztése volt a géphez. Az ehhez szükséges szinkronleválasztó készüléket is mi fejlesztettük ki és építettük meg. Céljainkhoz át kellett alakítanunk a tv-kamerát is. A szükséges program birtokában így szinte korlátlan lehetőség nyílt különféle ábrák fénykép minőségű megjelenítésére a központi egységen keresztül.

További nagy eredmény volt az ún. „vissza-színesítő” program alkalmazása, amellyel a fekete-fehér kamerával felvett kép eredeti színeit tudjuk rekonstruálni a képernyőn. Az így felvett képeket tetszés szerint nagyíthatjuk, kicsinyíthetjük, tárolhatjuk teljes egészükben vagy egyes részleteikben. Mivel a számítógép valódi grafikai funkciójában négyféle vonalvastagsággal, 64 színárnyalatban tud rajzolni, a megjelenített ábrák, függvények rendkívül pontosak és esztétikusak. Az így készített sok-sok ábrát évek óta használjuk különféle oktatási programjainkban.

Az ábrák programmal vagy tv-kamerával való bevitelén túl – megoldva az RA-03-as rajzdigitalizáló illesztését – rajzról, térképről is tudunk képet bevinni a központi egységbe.

Egy sor fizikai mennyiség mérésére és közvetlen értékelésre nyílt mód azzal, hogy különféle jeladókat készítettünk, amelyeket a Commodore VC-20-szal csatlakoztatunk. Így gyorsulásmérést, dermedési görbék felvételét, dugattyús gépek indikálását stb. végezhetjük számítógéppel.

Kutatjuk a HT-1080Z képernyőjén a programmal készített és kívülről bevitt képek monitorozásának megoldását.

A műszaki feladatok mellett kidolgoztunk egy nyelvtanulást segítő programot is, amelyet pályázatra küldtünk be, és el is fogadták.

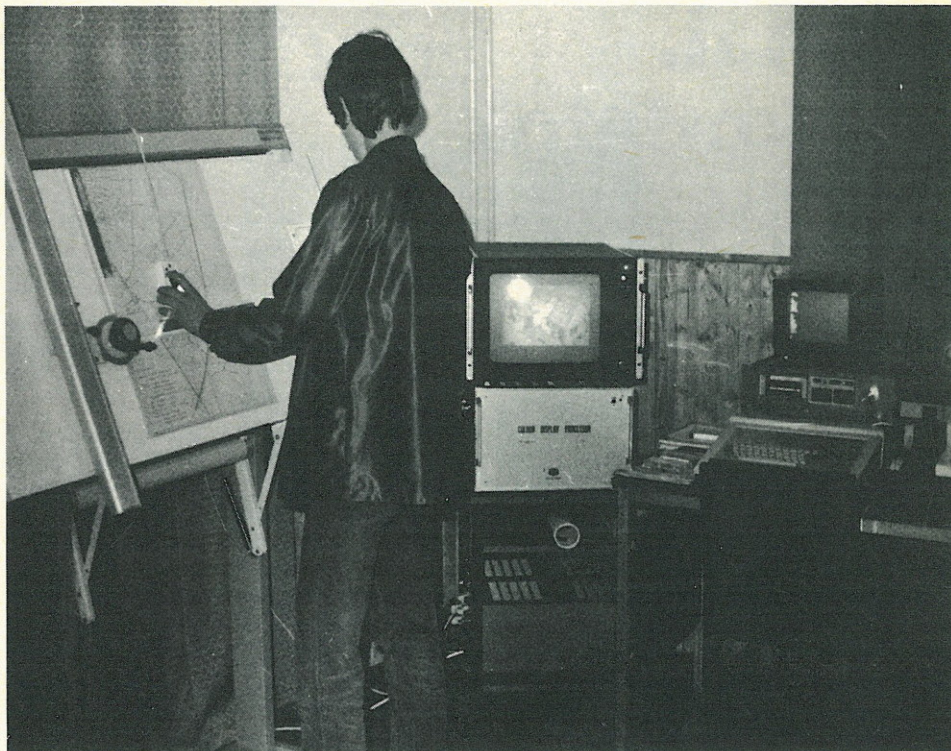
A mikroprocesszorok azonban nemcsak számítástechnikai munkánkban vannak jelen. Iskolai tanműhelyünkben 1980 óta üzemel CNC esztergagép, melyen természetesen tanulóink dolgoznak. Későbbi szerzeményünk egy CNC marógép, amelyen a mindennapi munka mellett a további alkalmazási lehetőségeket vizsgáljuk, és a működési paramétereket mérjük be a gyártó cég felkérésére.

Dolgozunk a számítógépes technológiai tervezés és szimuláció fejlesztésén, amellyel ugyancsak a tanműhelyi munkát készítjük elő.

További terveink között szerepel szaklaboratórium létrehozása a pneumatikus és hidraulikus technika oktatására, ahol szintén fontos szerepet kap a számítógép.

A fentiekből kiderül, hogy mit szeretnénk, de az is, hogy mit nem.

Részt vettünk a múlt év végén Budapesten rendezett országos középiskolai számítástechnikai kiállításon. Tenyérvényi gépcsodák, tisztes iskolaszámítógépek, bizonytalan származású szupergépek között azt próbáltuk meg tudni,



A HT-680X és a rajzdigitalizáló berendezés

mire használják ezeket. Nos, feltételezve, hogy az ott futó programok egyúttal az adott gépek fő alkalmazási területét mutatják, az eredmény enyhén szólva elgondolkoztató. Több száz gép közül egy-kettő mint valamilyen mérés demonstrációs vezérlőberendezése dolgozott, néhányon iskolai feladat megoldását láttuk, de a gépek 90 százalékán játékprogram futott. És ami nagyon elgondolkoztató: a játékoknak szinte kizárólag az a lényege, hogy ki, kit, hogyan győz le, semmisít meg.

Farizeuskodás lenne letagadni saját játékprogramjainkat. Persze hogy vannak nekünk is, de ezek addig voltak érdekesek, amíg megírták őket a tanulók. Tapasztaltuk, hogy ha értelmes feladatot tűzünk ki, az legalább annyira érdekli őket, mint a játék.

### Hírünk

Napjaink slágere a számítógép. Nem lehet divatcikk, múltó mánia, hiszen még mielőtt személyi számítógép formájában emberközébe került volna, már ezerszer bebizonyította, hogy fokozza a haladási sebességet minden területen. A számítástechnikára vonatkozó párhuzarozat megvalósítási szakaszában sokan voltak kíváncsiak eredményeinkre. Többször jártak nálunk a Tudományszervezési és Informatikai Intézet munkatársai, akikkel termékeny eszmecserét folytattunk. Hírünket bizonyítja, hogy több megbízatást kaptunk már számítástechnikai feladatok megoldására „külső” cégektől.

Szűkebb környekünkön egyébként két iskola számítástechnikai munkája a meghatározó. A miskolci Földes Ferenc Gimnáziumé (ők előbb kezdték) és a miénk. Ismertjük, becsüljük egymás munkáját, de az alkalmazás szempontjából különböző utakon járunk. Nem örülünk viszont annak, hogy meggyénkben tudomásunk

szerint sehol nincs több, mint a „kincstári” HT-1080Z, és még ennek használatáról sincs információnk.

Itt tartunk ma. A számítástechnika oktatásának és alkalmazásának iskolánkban eddig elért eredményeiről, a megtett útról villantottunk fel néhány képet.

Illó szólnunk végezettel azokról is, akiknek személy szerint a legtöbb érdemük van abban, hogy idáig eljutottunk. Diákjainkról: Kiss Csabáról, Dallos Józsefről, Nagy Alfrédre, Béres Zsoltról, akik különféle területeken, versenyeken, pályázatokon öregbítették iskolánk jó hírét. A KLTE Számítástudományi Intézetnek munkatársairól: Fazekas Gáborról és Papp Zoltánról, akiktől az indulástól kezdve magas szintű, önzetlen, baráti, szakmai segítséget kaptunk. Igazgatónkról, dr. Kálmán Andrásról, aki a szakmai tanácsokon túl a szűkös lehetőségek ellenére is biztosította munkánk anyagi és személyi hátterét. És végül Varga László kollégánkról, akinek tulajdonképpen mindazokat az eredményeket köszönhetjük, amelyekről szoltunk. Ő az, aki az első időkben tanári munkáján túl, majd a számítógép-laboratórium függetlenített vezetőjeként rengeteg ötlettel állt elő, fáradhatatlanul képezve magát oktatta kollégáit és tanulóinkat. Ő a lelke, motorja az ÜGY-nek.

Mert hiába következett be a világon a személyi számítógépek forradalmi áttörése, az új elfogadtatásához – úgy tűnik – mindig szükség lesz jó értelemben vett megszállott emberekre, akik az óvatos többséget megmozgatják. Ideje, mert súlyos lemaradásokat kell pótolnunk.

MERÉNYI JÓZSEF

mérnök-tanár

Zalka Máté Gépípari Szakközépiskola  
Miskolc

## KöMaL gyakorlatok MEGOLDÁS ZX81-EN

A Középiskolai Matematikai Lapokban ki-tűzött gyakorlatokat gyakran ZX81-es számí-tógépem segítségével oldom meg; hosszadalmas számításoktól, rajzolástól, függvényábrázolás-tól kímél meg. Olykor a helyes eredmény „meg-sejtésében” is segít. Erre szeretnék egy példát bemutatni.

Gy. 2192. (KöMaL 1984. 4. szám, 173. oldal). Egy  $8 \times 8$ -as sakktablán a mezőket úgy festjük át, hogy egy lépésben egy sor és egy oszlop – tehát összesen 15 mező – színét változtatjuk. Ha kezdetben valamennyi mező fehér volt, elér-hető-e, hogy a sakktabla a szokásos módon legyen kifestve?

Az előírt színezést a következő programmal végezhetjük el:

```
10 DIM A(8,8)
20 FOR F = 1 TO 8
30 FOR G = 1 TO 8
40 PRINT AT F,G;
   CHR$(136-8 * A(F,G))
50 NEXT G
```

11	13	15	17	22	24	26	28	31	33	35	37	42	44	46	48
51	53	55	57	62	64	66	68	71	73	75	77	82	84	86	88

A program alkalmazása nélkül igen bonyo-lult lett volna a feladat helyes eredményét kiki-sérletezni. Sok sakktablát kellett volna rajzolni, amíg a helyes megoldásra rájövünk, és közben nagyon kellett volna figyelni a megfelelő színe-zést is. Már egy mező helytelen színezése is felborította volna a megoldó stratégiát, és utó-lag nehéz lett volna visszakeresni, hogy melyik mezőt festettük hibásan.

Számítógéppel meg egy jó programmal mindez kiküszöbölhető. A program elkészítésé-hez először definiálnunk kell egy  $8 \times 8$ -as töm-böt (10. sor).

A 20–60. sorokban két ciklust szerveztünk, amivel a több elemeit jelezzük ki. A fekete, illetve fehér mezőket (ezek a képernyőn szürkék lesznek, és így élesen elválnak a képernyő fehér alapszínétől) a tömbben 1, illetve 0 jelöli. Ha a tömb valamelyik elemének tartalma 1, azaz a  $(136-8 * A(F,G))$  aritmetikai kifejezés értéke 128, akkor a 128 kódú grafikus karakter (fekete négyzet) jelenik meg. Ha a tömb tartalma 0, akkor pedig a  $CHR$(136-A(F,G)*8) = CHR$(136)$  és így a 136 kódú grafikus ka-rakter (szürke négyzet).

A program a 70. sorban várja a változtatni kívánt sor és oszlop számát. A 80. és a 110. sor az esetleges hibás adatbevitelt védi ki. A 90. és

```
60 NEXT F
70 INPUT A$
80 IF LEN A$ <> 2 THEN GOTO 70
90 LET I = VAL A$(1)
100 LET J = VAL A$(2)
110 IF I <> 1 OR I > 8 J < 1 OR J > 8
   THEN GOTO 70
120 FOR F = 1 TO 8
130 LET A(I,F) = 1-A(I,F)
140 LET A(F,J) = 1-A(F,J)
150 NEXT F
160 LET A(I,J) = 1-A(I,J)
170 GOTO 20
```

A program kiszolgálása során be kell írni a változtatni kívánt sor és oszlop számát, például 47, ami a 4. sort és 7. oszlopot jelenti.

A kitűzött gyakorlatra igenlő választ adha-tunk; egy helyes kiszínezés a következő. Kiszí-nezünk minden páratlan sorban minden párat-lan oszlopot és minden páros sorban minden páros oszlopot. Azaz:

a 100. sorban értéket adunk az I és a J változóknak: I tartalmazza a sorindexet, J az oszlopindexet.

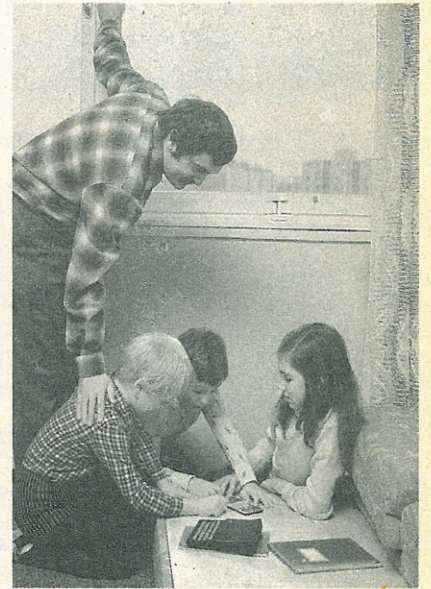
A 120–160-ig tartó programrész végzi el a feladat feltételeinek megfelelő színezést, azaz a változtatni kívánt sorban és oszlopban a fekete mezőket fehérre, a fehéreket feketére cseréli. A cserék után a tömbben az 1-es tartalmú ele-mek tartalma 0, a 0 tartalmúaké pedig 1 lesz. A változtatást az  $f(x) = 1 - f(x-1)$  periodikus függvény végzi. Ebben a ciklusban a változta-tott sor és oszlop metszéspontjában lévő elem színét kétszer cseréltük meg, vagyis maradt az eredeti szín. A feladat követelményei szerint ennek is változnia kell, amit a 160. sor biztosít.

Végül a 170. sor visszaugrat a kijelző rutinra. A kijelzés és a változtatás elég lassú, ezért a programot érdemes gyors üzemmódban (FAST) futtatni. A 70. sor INPUT-jánál amúgy is kijelzésre kerül a képernyőfájl.

A kitűzött feladat  $M \times N$ -es sakktablára is általánosítható, ahol M és N páros számok. Ekkor természetesen némi értelemszerű változ-tásra szorul a program. A megoldást bizzuk az olvasóra.

KOVÁCS LAJOS  
I. oszt. gimn. tanuló  
Nyíregyháza

### Családi kör



## Már ők is tudnak BASIC-ül

Szegeden a TIT Városi Szervezete és a Balázs Béla Úttörőház bentlakásos BA-SIC nyelvi tábort rendezett 1984. június 20-tól 25-ig, illetve július 2-től 7-ig, álta-lános iskolai 6., 7. és 8. osztályos tanulók részére. A táborozáson Baranya, Bács, Csongrád és Pest megyéből mintegy 210 gyerek vett részt.

A szakmai irányítást a Juhász Gyula Ta-nárképző Főiskola számítástechnikai ok-tatói végezték, a számítógépes gyakorlato-kat a főiskola oktatói, valamint általános és szakközépiskolai tanárok vezették.

A turnusonként megtartott 9 elméleti órához – melyen a BASIC nyelv alapfo-galmaival, a programírás szabályaival is-merkedtek meg a tanulók – 18 számító-gépes gyakorlati óra csatlakozott hét gyakorlati helyen; így egy-egy gépre há-rom tanuló jutott. A JATE Kibernetikai Laboratórium, a főiskola, az Úttörőház és a 600. sz. Szakmunkásképző a tábor rendelkezésére bocsátotta személyisza-mítógép-parkját. A résztvevők progra-mozhattak és játszhattak is ABC-80, Commodore 64, ZX80, ZX81, ZX-Spect-rum, HT-1080Z, HT-2080Z és Wang 2200 számítógépeken.

Az intenzív tanulást gondosan szerve-zett kultur- és sportprogram könnyítette meg a gyerekek számára, akik a turnus végén emléklapot és jutalomkönyvet kaptak.



## Beírás - futás közben

Különböző problémák megoldása során előfordul, hogy egy olyan BASIC aritmetikai kifejezés értékét kell kiszámíttatnunk a program futása közben, amelyet az indításkor még nem ismerhettünk. Ennek több oka lehet. Például: a gép választ vár tőlünk, amelyre egy számot kellene beírni. A beírandó szám a programban szereplő BASIC változók aktuális értékének valamilyen függvénye. A felhasználónak azonban nem kell ismernie a változók értékét, csak a függvényt. Megoldható lenne a dolog az aktuális értékek kiírásával, a függvény helyettesítési értékének külön kiszámításával, de ez sok hibalehetőséget rejt.

Egy másik példa: egy alapprogramban néhány értékadó utasítás aritmetikai kifejezését meg kell változtatni a program különböző célú felhasználása miatt. Mivel a HT-1080Z nem ismeri a MERGE parancsot, programok keverésére csak hosszadalmas módon lehet rábírn. Jó megoldás lenne, ha a BASIC kifejezés sztringként bevihető lenne egy fájlból.

Ilyen és más esetekben jelenthet segítséget az alábbi eljárás, amelynek alap gondolata, hogy a BASIC aritmetikai kifejezést komplett módon átadjuk egy sztringváltozónak, például K\$-nek. Ezt a kifejezést ezután egy előre meghatározott tárcímre egy utasítássorszám és egy BASIC értékadó utasítás „=” jele után rendre, alkalmas módon POKE utasításokkal lerakjuk. Ha ezután a programot kilistázzuk, a megfelelő sorszámon az általunk bevitt kifejezést találjuk, mint „rendes” BASIC értékadó utasítást. Ha pedig kilistázás helyett ezt a programrészletet szubrutinként meghívjuk, rendelkezésre áll a kifejezés értéke.

A feladat megoldásához néhány alapvető szabályt be kell tartani.

Az értékadó utasítást a programban olyan helyen kell elhelyezni, amely elé már semmiféle utasítást nem szúrunk be. Ezen utasítás előtt levő programrészletben semmilyen változtatást nem szabad végrehajtani, mert különben utasításunk a tárban elmozdul, és később rossz helyre POKE-oljuk az aritmetikai kifejezést.

Fontos még, hogy elegendő hely maradjon a tárban az egyenlőségjel után. Ajánlatos csupa „.” karaktert írni, mert ezt a gép nem veszi figyelembe. Legcélszerűbb az 1. program szerint eljárni.

Ha nincs semmi ez előtt a tárban, akkor így a 30. utasítás „=” jele a 17155 tárcímre kerül. Erről könnyen meggyőződhetünk, de célszerű még más egyebet is megvizsgálni.

Ismert, hogy a gép az egyes kulcsszavakat ún. token formájában, vagyis egybájtos bináris számként rögzíti. Írassuk ki egy egyszerű BASIC kifejezés tárban elhelyezését. A 30. utasítás után most rendes programlépésként írjunk egy értékadó utasítást (2. program).

A 3. programmal megvizsgálhatjuk, mi van a tárban, melynek eredményeként a 4. listát kapjuk.

A program rendre kiírja a tárcímet, a bájttartalmát és a tartalomnak megfelelő karaktert. Feltűnik, hogy a beírt BASIC kifejezés épen visszakerül a képernyőre a harmadik oszlopban, csak a műveleti jelek helyett vannak más számok, mint a nekik megfelelő ASCII kód. Pontosan ezek a műveletek tokenjei. Megvizsgálhatjuk ezzel a módszerrel bármely matematikai függvény lepakolási szabályát is. Itt a sza-

```
10 GOTO50
20 REM ELEJE
30 AA=:
:
:
:
40 RETURN
50 REM FOPROGRAM
```

### 1. program

```
30 AA=2*A/(23-5*B)+1
```

### 2. program

```
50 REM FOPROGRAM
60 FORI=17129TO18000
70 A=PEEK(I)
80 PRINTI,A,CHR*(A)
90 IF INKEY*="" THEN 90
100 NEXT
```

### 3. program

.		
.		
.		
17141	147	
17142	32	
17143	69	E
17144	76	L
17145	69	E
17146	74	J
17147	69	E
17148	0	
17149	239	
17150	67	C
17151	30	
17152	0	
17153	65	A
17154	65	A
17155	213	
17156	50	2
17157	207	
17158	65	A
17159	208	
17160	40	(
17161	50	2
17162	51	3
17163	206	
17164	53	5
17165	207	
17166	66	B
17167	41	)
17168	205	
17169	49	1
17170	58	:
17171	58	:
17172	58	:
17173	58	:
17174	58	:
.		
.		
.		

### 4. program

```
10 GOTO50
20 REM ELEJE
30 AA=:
:
:
:
40 RETURN
50 REM FOPROGRAM
60 CLEAR300
70 M*="+*/" :K=17155
80 INPUT "KEREM AZ ARITMETIKAI
KIFEJEZEST";K$
90 FOR I=1 TO LEN(K$)
100 A$=MID*(K$,I,1)
110 FOR J=1 TO 4
120 IF A$=MID*(M$,J,1)
THEN A=204+J : GOTO 150
130 NEXT
140 A=ASC(A$)
150 POKE K+I,A
160 FOR J=0 TO 1 : NEXT
170 NEXT
180 POKE 17155+I,58
POKE 17156+I,147
190 GOSUB 30 : PRINT AA :
STOP
```

### 5. program

bály egyszerű: a műveleti jelek helyére a token rakja, minden mást karakterenként pakol.

A tokenek:

```
205 +
206 -
207 *
208 /
147 REM
```

Ha tehát azt szeretnénk, hogy a sztringkifejezésből BASIC aritmetikai értékadó utasítás legyen, a K\$ kifejezést némileg át kell alakítanunk. Minden műveleti jel helyére a saját tokenjét kell berakni.

Erre alkalmas az 5. BASIC program (csak műveleti jelekre).

A program működése egyszerű. A K\$ sztringet karakterenként megvizsgálja. Ha műveleti jelet talál, akkor a megfelelő token-t írja a helyére, egyéb esetekben az ASCII kódot. Külön kell említeni a 160. sort. A ciklusból kiugrást a HT-1080Z gépen nem lehet mindig büntetlenül megcsinálni. Bizonyos esetekben olyan hibajelzést kapunk ennek hatására, hogy NEXT utasítást használtunk FOR nélkül. A hibajelzés azonban nem ott mutatkozik, ahol elkövettük a hibát, hanem sokkal később. A fenti módon ez elkerülhető.

Nem volt szó eddig arról, hogy mi kerüljön a 30-as sor végére. Ide egy : után a REM token-jét célszerű betenni. Ezt végrehajtja a 180-as utasítás.

Végül, mielőtt a számító szubrutint meghívjuk, gondoljunk arra, hogy szintaktikailag hibás kifejezés esetén se függesszük fel a program futását. Erre szolgál a hibakezelő rutin, amely ilyen esetben megismételteti a sztring beírását.

NYIRATI LÁSZLÓ

Ybl Miklós Szakközépiskola  
Székesfehérvár

THEISZ GYÖRGY

József Attila Gimnázium, Székesfehérvár

## Alapozás VI.

**A számítógép egymással „kommunikál” operátorok összessége. A gépet alkotó operátorok egymással „beszélgetnek”. E beszélgetés azonban nem kedélyes csevegés, nem véleménycsere, hanem minden esetben olyan informálás és informálódás, ami az informálódó operátor viselkedését törvényszerűen alakítja. Az operátorcsapat tagjainak beszélgetésében nincs humor, nincs érzelem, sőt még értelem, fontolgatás sem. A robotok „beszédjükkel” hidegen, élettelen gépként, szigorúan megszabott keretek között irányítják egymás munkáját. A kapott információ mindig automatikusan hat. A robotoknak nincs szabad akaratuk.**

**A robotok közlési folyamatai szigorúan szabványosak. Ez megkönnyíti a számítógép-építést, és bizonyos esetekben (de nem mindig!) a gép működésének a megértését is. A közlési folyamatok megértése szükséges ahhoz, hogy megértsük a gépek „lelkivilágát”. Ennek érdekében folytatjuk most a robotok kommunikációjával kapcsolatos vizsgálódásainkat.**

### Mikor mi számít és mi nem?

Felvetettünk kérdéseket az üzemmódváltással, az üzemmódváltással kapcsolatban. Például azt, hogy *egy ferdén felfutó él melyik pontját kell „indító” időpontnak tekinteni?* Vagy *van-e a fel- meg a lefutó él meredekségének határa, melynél laposabb él már hatástalan?*

E kérdések a robotok „beszédjére” és a „beszéd” más robotokra való hatására vonatkoznak, ezért központi jelentőségűek. Hozzálatunk tehát ezek megválaszolásának előkészítéséhez.

Azzal kezdjük, hogy nyomtatékosan felhívjuk a figyelmet arra, hogy senki se reménykedjen egyszerű megoldásban. A népszerű számítástechnikai irodalom e fontos kérdéseket vagy elkerüli, vagy jelentéktelennek feltüntetve, hamisan egyszerű választ ad rájuk. Az operátorok, a robotok között mikor mi számít, mikor minek milyen hatása van, erre általános érvényű válasz nincs és nem is lehet; ezt minden konkrét gép esetében a számítógép tervezője és a programok készítője határozza meg.

A számítógépben zajló jelenségek megértését szinte lehetetlenné teszi egy egyébként könnyen tisztázható kettősség. A számítógép matematikai precizitású eszköz – a matematika oldaláról nézve. A megvalósítás oldaláról nézve azonban ugyanolyan, tapasztalatokat hasznosító gép, mint a többi. A gép működésének megértéséhez a tapasztalatlanok, a naivak keresik a matematikai egzakttságot. Ezt nem találják, mert itt ilyen nincs. Sőt még tisztességesen megírt szakácskönyvet sem találnak. (Ez azonban lehetne.)

A gép matematikai folyamatokat utánoz fizikai folyamatokkal. A fizikai jelenségeknél azonban a miértekre sokszor csak az az egyedül egzakt válasz, hogy „ez a tapasztalat”. Ehelyett azonban sokszor félmagyarázatokat és álmagyarázatokat adnak, amik csak megtévesztésre alkalmasak. A helyzetet még az is nehezíti, hogy a gyakorlati szakember el sem tudja képzelni, hogy miért kellene – a teljes precizitás kedvéért – olyan esetekkel foglalkoznia, ami-

nek semmi gyakorlati haszna nincs. A gyakorlati szakember magyarázata így – matematikai szempontból – szinte sohasem egzakt. Az ő célja nem az, hogy minden esetet – még az érdektelenekeket is – gondosan végigelemezze, hanem, hogy jól működő, gyors gépet alkosson. Annak érdekében, hogy ezt a súlyos problémát megoldjuk, nézzünk szembe vele. Kiderül, hogy a szembenézés már majdnem megoldás is. Szinte minden nehézség megoldódik, ha tudatosítjuk a szemléletmódbeli különbségeket.

### A matematikai és a gyakorlati megközelítés

Nagyon fontos különbség a matematikai és a gyakorlati szemléletmód között az, hogy a matematikust mindig a teljes eseményrendszer, a teljes lehetőségrendszer érdekli, a gyakorlati szakembert pedig csak a számára hasznos és káros események, illetve lehetőségek foglalkoztatják. A számítógépet alkotó operátorokhoz a matematikus úgy közelít, hogy teljes működéstörvényükre kíváncsi. A gyakorlati szakember viszont csak olyan esetekben használja az operátort, amely esetekben hasznos munkát kap tőle. A többi eset nem érdekli. Gondoskodik róla, hogy a „többi eset” ne fordulhasson elő, vagy ha előfordul, akkor ne használja a kialakuló eredményt. A gyakorlati szakember csak nagyon ritkán tudja, hogy milyen működés zajlik olyankor, amikor az számára közömbös. A matematikust ez is érdekli.

Egy jól jellemző példával szemléltetjük a gyakorlati szakember gondolkodásmódját.

A számolási munkát végző operátorok helyes működéséhez a feldolgozás (például egy összeadási munka elvégzése) előtt rendelkezésre kell állnia az összeadandóknak. Ez természetes. Ez a rendelkezésre állás azonban nem lehet akármilyen rövid időtartamú. Hogy a minimális időtartam pontosan mekkora, és hogy akár csak egy jól meghatározott konkrét esetben létezik-e egyáltalán, nagyon nehéz, valószínűleg megoldhatatlan kérdés. A gépszerkesztő gya-

korlati szakembert azonban ez nem izgatja. Gondoskodik róla, hogy a rendelkezésre állási idő minden esetben a szükségesnél egy kicsit hosszabb legyen (1. ábra).

### Az operátorok közötti kommunikáció sajátágai

Az operátorok közötti beszélgetésnek az emberek közötti beszélgetéssel rokon és attól eltérő tulajdonságai is vannak.

Ugyanúgy, ahogy az emberek között, az operátorok világában is ritka az olyan eset, amikor egyszerre csak egy beszél. Ennek oka az emberek között gyakran szereplésvágy és neveletlenység, az operátoroknál azonban az idő ésszerű kihasználása. Az emberek közötti kommunikáció szinte sohasem teljes mértékben kényszerítő erejű. Az operátorok között viszont mindig megtagadhatatlanul erős parancs jellegű, amit az operátor mindig teljes mértékben figyelembe vesz, és saját törvényeinek megfelelően, fontolgatás és érzelmek nélkül, automatikusan reagál is rájuk.

Az olyan társaságban, amelyben egyszerre többen beszélnek, mindenki mindent hall, de általában csak saját magára figyel, és lesi, hogy mikor juthat szereplési lehetőséghez. A robotok között mindegyik csak azt hallja, amire működéséhez szüksége van, csak arra figyel, amire kell, és ezt 100 százalékos „lelkiismeretességgel” teszi. Az operátorok nem lesik ugrásra készen a szereplési lehetőséget, hiszen mindegyik állandóan szerepel, akkor is, ha meg sem szólal; de nem hiúságból vagy hatalomvágyból, hanem mechanikusan determinált kötelességből vagy inkább kényszerből.

Az olyan emberi társaságban, amelyben mindenki mondja a magát, lehetetlen megállapítani, hogy mikor ki kihez beszél, mikor ki kire figyel. A robotok között ez a kérdés mindig egyértelműen rendezett.

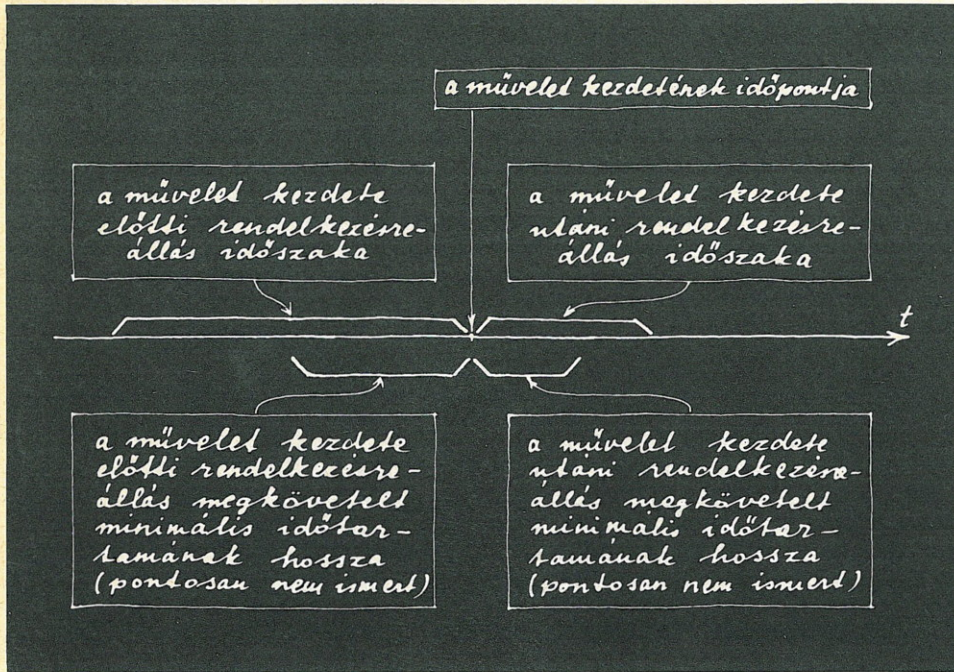
Ritka az olyan ember, aki egyszerre több másikkal tud hasznosan és értelmesen, különböző témákról eszmecsere folytatni. A robotok között ez gyakori és természetes. Egy robot egyidejűleg több másik beszédjét figyelheti és több másikhoz beszélhet is azalatt. Sőt két robot között egyidejűleg több különböző eszmecsere is folyhat.

A magában beszélő ember elmekórtanilag gyanús lehet. Még inkább gyanús azonban az, aki magának beszél, magának küld üzeneteket. A robotok esetében ez a jelenség nem gyanús, nem káros, sőt nem is ritka, még úgy sem, ha például a robot egyidőben saját magának több különböző témában küld üzenetet.

E furcsa de hasznos jelenségek felhívják a figyelmünket annak fontosságára, hogy figyeljük meg a különböző robottípusokat és tulajdonságaikat. Teljesség igénye nélkül néhány ilyen megfigyelést teszünk a következőkben.

### Robotfajták és robotnyelvek

Hogyan beszélgetnek az emberek? (Ma már olyan csúnyán, hogy erről jobb volna nem beszélni.) E kérdés elől azonban – bárhogy szeretnénk is – csak részben térhetünk ki. Az emberi beszélgetés jelenségének fő ága kétségtelenül a hangadás, és a hang felfogása, érzékelése. Most csak ezt a fő folyamatot vesszük figyelembe. A többi fontos jelenségtől, például a gesztu-



1. ábra

soktól, a mimikától, a témától, a szókinccstől stb. most eltekintünk.

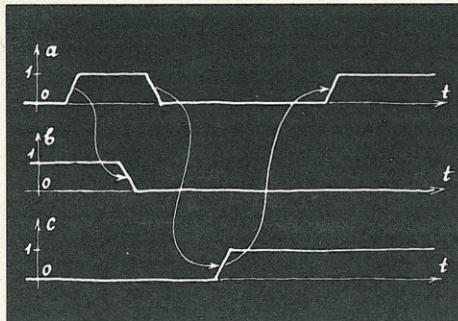
Az egyik ember tehát hangfolyamatokat (rezgéseket) bocsát ki magából. A másik pedig hallja, érzékeli ezeket. Tudjuk azonban, hogy a kommunikációnak – még emberek között is – vannak más eszközei is. Lehet kommunikálni írásban is. A süketnémák is tudnak „beszélgetni”, gesztusokból álló „beszéddel”, folyamatokkal. A látszólagos különbözőségek ellenére minden beszélgetés kibocsátott és felfogott információt hordozó folyamatokkal bonyolódik le.

Az emberek beszélgetése általában hangfolyamatok révén valósul meg. A robotok esetében nagyobb a változatosság. A számológép alkatrészeként működő robotok elektronikus folyamatok kibocsátásával és felfogásával valósítják meg beszélgetéseiket. (Újabbán a közönséges fény, a lézersugár, a hang is mind nagyobb szerepet kap.) A programok robotjainak, az operátorcsapat tagjainak beszélgetése azonban számok és jelek kibocsátásának és érzékelésének folyamatából áll.

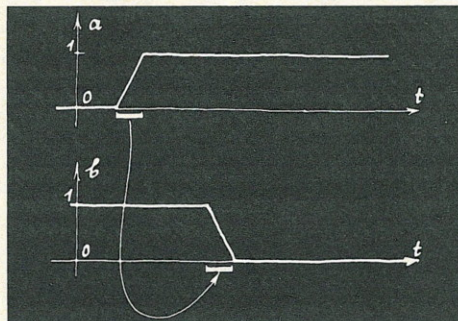
A gép alkatrészeként működő robotok esetében a számológép-tervező egyszer s mindenkorra eldöntötte, hogy melyik robot melyikkel beszélgethet. (Ez alól csak bizonyos kapcsolók a kivételek.)

A program operátorai azonban szabadabban. Sem kapó, sem adó kezük nincs összeforrasztva más operátorok kezeivel. E szabadabb operátorok megtehetik, hogy részben vagy egészben maguk válasszák meg azokat a helyeket, ahonnan kapó kezeikbe információt juttathatnak, és azokat a helyeket is, ahová adó kezeiken keresztül az információt áramoltatják.

Annak az operátorfajtának, amely a programokban szerepel, van egy nagyon érdekes tulajdonsága. Ezek adó és kapó kezeinek tartalma lehet operátor is. (Hasonlóan a teherautót szállító teherautóhoz, vagy a repülőgépet szállító repülőgéphez.) A számológépet alkotó építőelemek viszont olyan operátorok, amelyekre ez a tulajdonság nem jellemző.



2. ábra



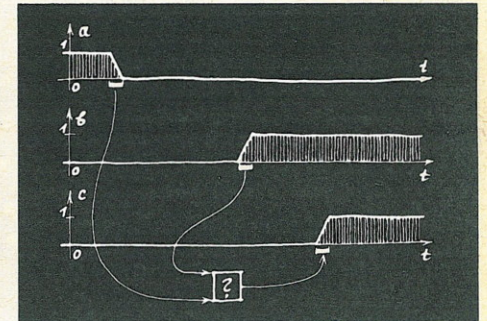
3. ábra

Van viszont ezeknek egy olyan tulajdonságuk, amivel nem minden programbeli vagy hozzá hasonló operátor rendelkezik. A számológépet alkotó operátorok bizonyos kezeik adó vagy kapó (fogadó, felfogó, érzékelő) jellegét is változtatni tudják. Természetesen nem önkényesen, hanem külső vagy belső parancsra. Sőt egyesek ezenkívül arra is képesek, hogy némely kezüket „semmilyen” állapotba hozzák, azaz, ha szükség van rá, sem adásra, sem kapásra nem használják azokat. Erre a bravúrra nem minden másfajta (például programbeli) operátor képes. Ez a tulajdonság olyan, mintha valaki a fülével beszélni vagy a szájával hallani is tudna, és ha kell, eltüntetné mindkettőt. (Meg-

jegyezzük, hogy a számológép építőelemeinek kivezetéseit nem kezeknek, hanem lábaknak szokták nevezni.)

Az ember és a különféle robotfajták egymás közötti beszélgetése rokon és eltérő vonásainak sok haszonnal járó taglalását a „hangerő” kérdéssel zárjuk. A társaságban levő ember hangját, ha sokan vannak, egy bizonyos távolságon túl nem lehet megérteni, mert elnyomja az általános zaj. Minden embernek van egy hangerőségtől és zajsinttől függő jellemzője, hogy hány embert győz még jól hallható szöveggel ellátni. A számológép építőelemeinek társasága ebben hasonlít az emberi társaságra. Minden robotnak, helyesebben minden robot minden adó kezének van terhelhetősége, táplálóképessége; ezt általában úgy méri, hogy hány más kapó kéz csatlakozhat egy adó kézre. Ez a szám sohasem végtelen. Sőt a tízet sem szokta elérni. (Ha egy adó kéznek például 20 kapó kezét kell táplálnia, ún. erősítő operátort kell közbeiktatnunk.)

A programoperátorok adó kezeinek terhelhetősége azonban elvileg végtelen. Ezek az operátorok ugyanis nem fizikai lények, és így mű-



4. ábra

ködésükhöz, illetve működtetésükhöz nincs szükség energiára.

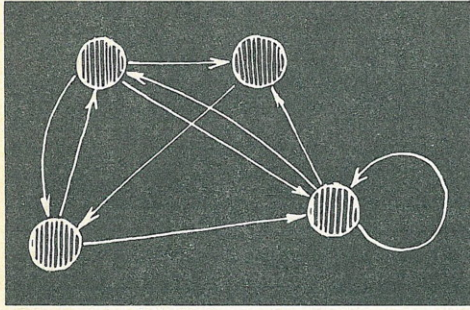
Végezetül pedig az operátorok körében lehetetlen a másik túlkibálása, és az is jellemző rájuk, hogy bármilyen finom információval is tápláljuk őket, sohasem mohók, nem csámcsognak, nem bőfognak, és nem eszik le az asztalterítőt.

## Ábrázolási szokások

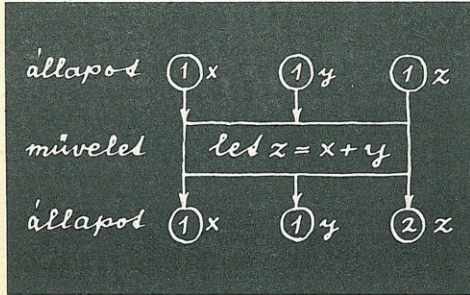
Most csak a további munkánkhoz nélkülözhetetlenül fontos néhány grafikus ábrázolási szokást érintünk. A kérdéskörre később majd még visszatérünk. Az elektronikus folyamatoknál a folyamat egy-egy jellemzőjét az idő függvényében ábrázolhatjuk, ha az például valós számmal megadható. Ilyenkor esemény lehet például az érték megváltozása, de adott ideig állandó volta is.

Események kapcsolatának jelölésére szokásos az ok–esemény jelétől az okozat–esemény jelére mutató nyíl használata. Mindig a kiváltó esemény jelétől indítjuk a nyilat, amelynek hegye a következmény–esemény jelen van (2. ábra).

Példaképp a számológépben tipikus értéktartományú folyamatokat mutatunk be. Az ábrán az  $a$  értékének 0-ról 1-re változása a  $b$  értékét 1-ről 0-ra viszi. Ugyanezt teszi a  $c$  értékével az



5. ábra



6. ábra

a folyamat 1-ről 0-ra esése, ami visszaviszi a értékét 1-re.

Érdeemes e nagyon hasznos és szemléletes módszert pontosabbá és hatékonyabbá tenni. Először is kifejezőbb és értelmesebb az eseményeket – mivel azok folyamatok – az időtengelyen jelölni (3. ábra). Másodszer pedig hasznos lehet az események közötti viszony pontosabb jelölése is. Az sem ritka, hogy egy eseményt nem egy, hanem több másik vált ki. Jó szolgálatot tehet ilyen esetekben is az operátoros jelölésrendszer (4. ábra).

Az operátoros módszer, szemben a 2. ábrán alkalmazottal, teljes körű információt képes nyújtani az események viszonyáról. Az persze nem mindig egyszerű feladat, hogy mikor mit kell az operátort jelképező téglalapba (a kérdőjel helyett) írni. (Például olyankor, amikor a kiváltó folyamat és a kiváltott folyamat időtartama nem egyforma.)

Operátoraink folyamatok kibocsátása és folyamatok felvétele (érzékelése) révén beszélgetnek. A beszélgetés eredménye fontos számunkra. Ez az eredmény mindig valaminek a változatlanul maradása és valamiknek a valamilyen módon való megváltozása lehet, más nem. Van olyan eset, amikor végessok lehetőséget különböztetünk meg, és arra vagyunk kíváncsiak, hogy ezek mikor hogyan következhetnek egymás után.

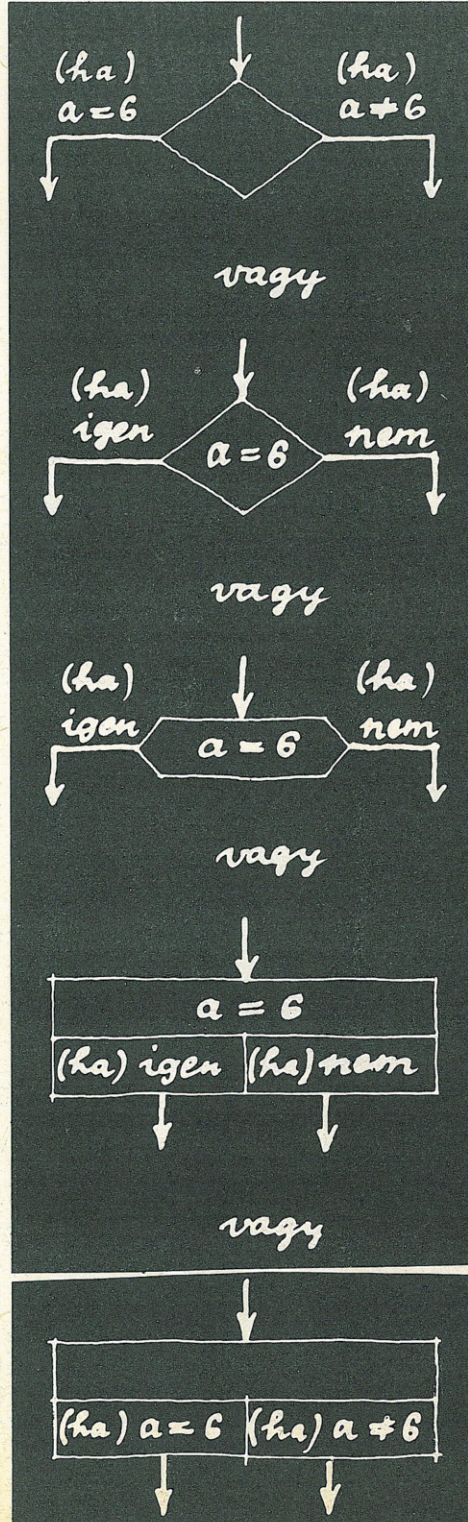
A pontosság érdekében meg kell mondanunk, hogy mit értünk itt lehetőségen. A válasz: mindig valamilyen állapotot. Ez lehet pillanatnyi és huzamosabb állapot is. Az állapot olyan jelenség, amelynek zajlása idején valamilyen jellemzője változatlan. Nemcsak nyugalmi állapot van, hanem valamilyen tevékenységvégzési állapot is. Amíg ez utóbbi zajlik, a tevékenységet meghatározó (mutató) jellemző értéke állandó. (Felhívjuk a figyelmet arra, hogy a köznyelvben előforduló „lehetetlen állapot” szóhasználat modoros túlzás, a matematikában pedig olyan állapot, amely sohasem fordulhat elő.)

A lehetséges állapotokat körökkel szokták jelölni. Egyik körtől egy másikig mutató nyíl

pedig azt jelzi, hogy a nyíl kezdőpontjánál levő kör által képviselt állapot után közvetlenül következhet az az állapot, amit a nyíl hegyénél levő kör jelöl. Az 5. ábrán van olyan állapot is, amely után újra következhet ez az állapot is. Itt megint olyan esettel van dolgunk, amelyben a matematikai és a gyakorlati felfogás élesen eltér egymástól.

A gyakorlat el szokta hanyagolni az ő szempontjából haszontalant. Úgy veszi, mintha nem is volna. Ezért a gyakorlati szakemberek matematikai modelljeiben legtöbbször csak az álta-

7. ábra

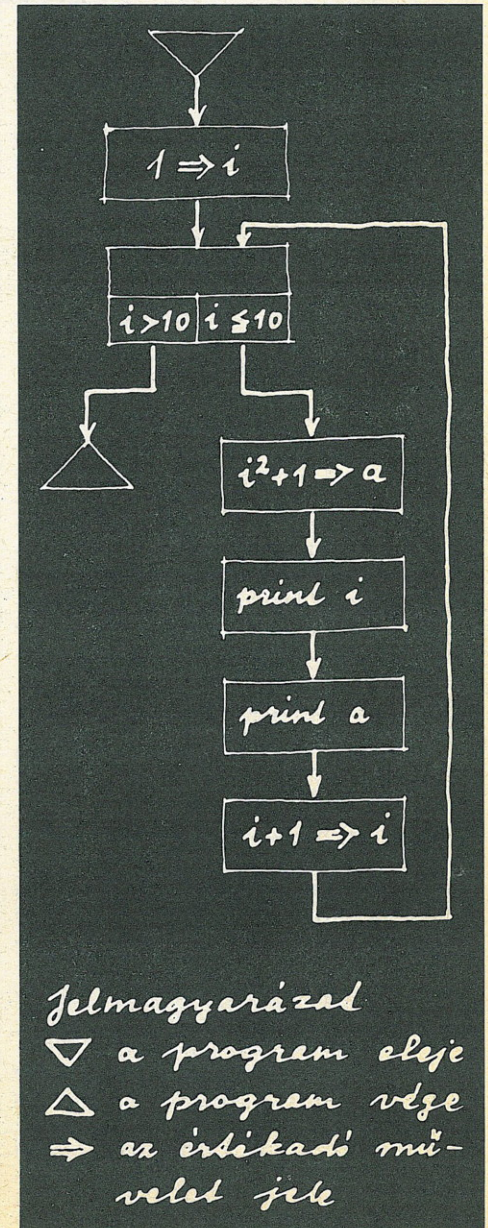


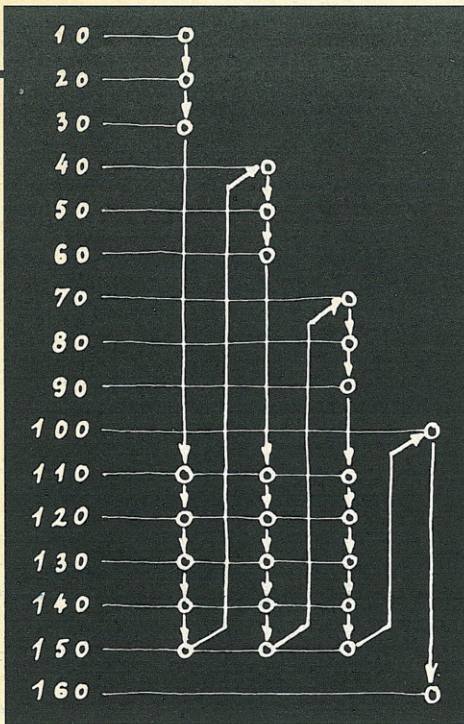
luk fontosnak tartott elemeket találjuk, csak ezeket veszik figyelembe. A most tárgyalt ábrát is ilyen csonkításokkal szokták használni. Az esetek többségében ugyanis egyik fontosnak tartott (figyelembe vett) állapot után nem rögtön következik a másik fontosnak tartott (figyelembe vett) állapot, hanem köztük egy átmeneti állapotra is sor kerül. (A 0-ból 1-be, illetve az 1-ből 0-ba jutás átmeneti állapotai jellegzetes példák erre.) Ezt a tényt figyelmen kívül szokták hagyni. Ez a felületesség azonban nem mindig engedhető meg.

Talán már sejthető, hogy operátorokkal itt is mindent meg tudunk oldani. Ez azonban kissé bonyolultabb kérdés, amire később esetleg még visszatérünk.

A nyilakra sok minden írható, például az állapotátmenetek ideje, az állapotátmenet törvénye, annak különböző jellemzői. Hasonlóan, a körökbe vagy melléjük írhatjuk az állapotok jellemzőit, például maximális, minimális, átlagos, pontos fennállási időtartamot, költséget vagy energiafelhasználást stb.

8. ábra





9. ábra

Számos esetben nincs szükség teljes állapotátmenet ábrára, csak annak egy-egy részletére. Ilyenkor mindig jelezni kell, hogy nem teljes ábráról van szó.

Programunkhoz mindig készíthetünk állapotátmenet ábrát vagy részábrát, illetve ezzel rokon táblázatot, mégpedig többféle módon is. Az egyik fontos gyakorlati eset a következő. Ebben az utasítások végrehajtásának folyamatait átmeneti állapotoknak vesszük (vagy foglalkozunk velük külön, vagy nem), két egymás utáni utasítás végrehajtása között tetszőleges, de egyértelműen meghatározott rekeszek, regiszterek tartalma nem változik, tehát felhasználható állapotjellemzésre.

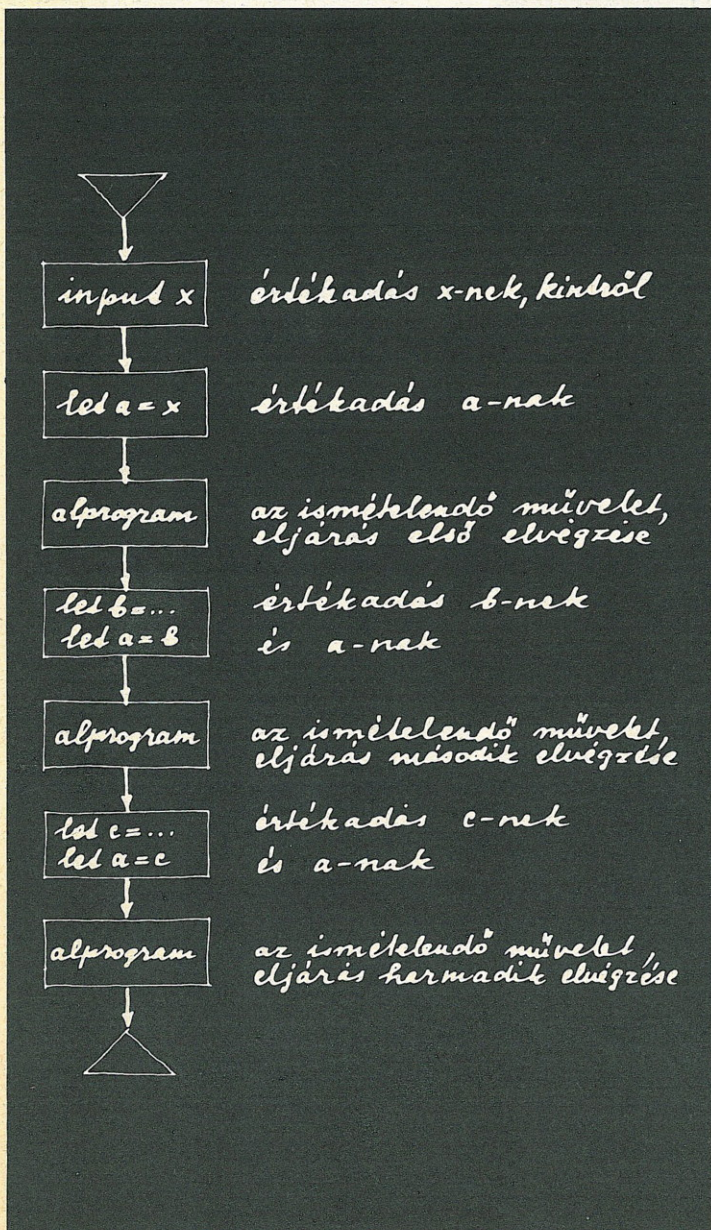
Ha például  $x, y, z$  a kijelölt rekesz, akkor, ha minden utasítás után beírjuk a `print x,y,z` utasítást, akkor a gép a program végrehajtása során minden lépésben tájékoztat „az állapotról”. Az  $x = 1, y = 1, z = 1$  által jellemzett állapotból például a `let z = x + y` utasítás végrehajtása után az  $x = 1, y = 1, z = 2$  által jellemzett állapot következik (6. ábra). Ebben az esetben

azonban csak a gép korlátai miatt mondhatjuk, hogy végessok állapottal van dolgunk. Azonban végtelensok állapotú rendszerekkel sem mindig reménytelenül nehéz a munka, sőt néha még jó ábrák készítése sem lehetetlen. Végtelensok állapotú rendszerek esetében azonban már formulákkal célszerűbb az állapotokra (is) vonatkozó összefüggések leírása; természetesen az operátorok itt is segítenek majd.

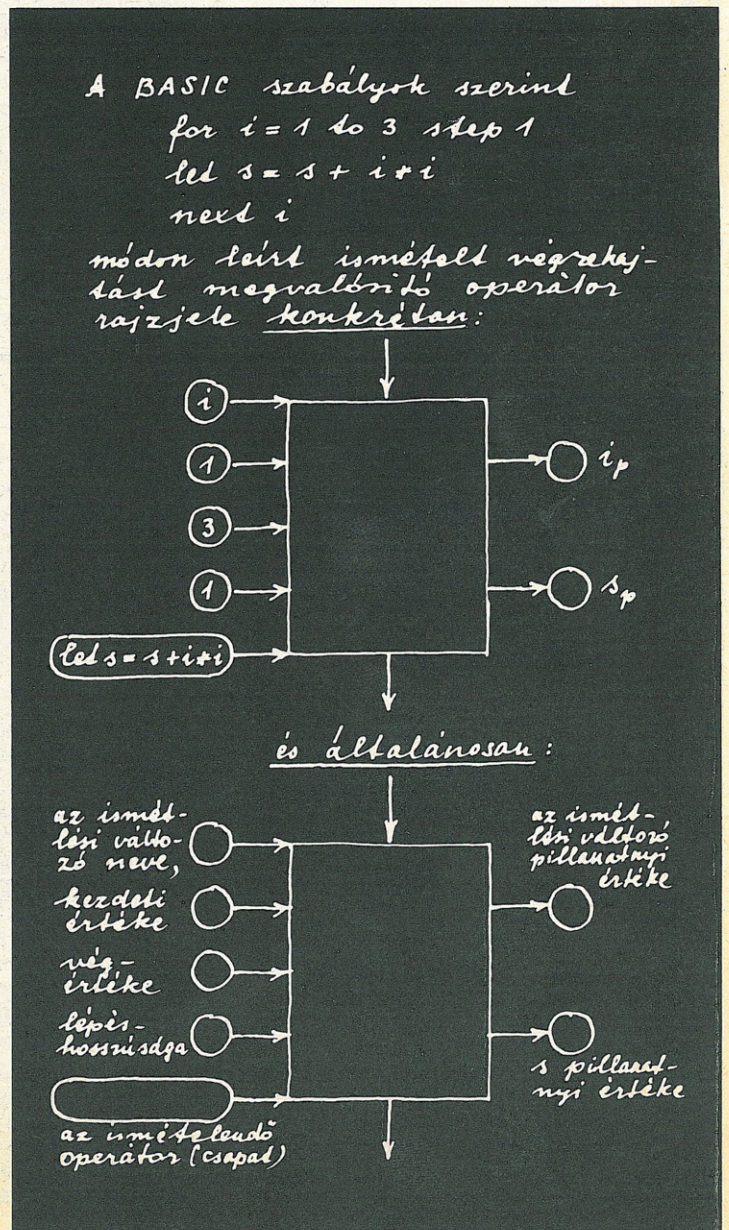
Fontos tudatosítanunk, hogy a legkisebb számológépnek is végtelensok állapota van, de a legnagyobb digitális gép leírásánál is mindig csak végessok állapotot szoktunk figyelembe venni. A lehetséges állapotok halmazát (vagy annál bővebb halmazokat) állapotternek hívjuk. A modellezett rendszer ebben az állapotterben tartózkodik, mozog, bolyong, ugrál. Az állapotter geometriai szerkezete és a rendszer benne lebonyolódó viselkedése fontos információk szoktak lenni.

Programok állapotátmenet ábrájának másik fajtáját kapjuk, ha a program egyes utasításainak végrehajtásait vesszük (tevékenységvégze-

10. ábra



11. ábra



si) állapotoknak. Ez egy olyan speciális kapcsolatábrához vezet, amelyen végigmenve számozási folyamatot (azaz egy működéstörténetet) kapunk.

Ennek az ábrázolásmódnak kialakult szabályai vannak. A közönséges értékadási műveletek elvégzésének állapotait például téglalap jelöli. A vizsgálat függvényében történő folytatási helykijelölő művelet (például  $\text{if } y = 6 \text{ then } 100 \text{ else } 200$ ) jele rombusz, „csúcsos téglalap” (hatszög) vagy speciálisan kialakított téglalap lehet (7. ábra). A nyilak (irányított vonalak) szerepe változatlan; a nyíl irányában közvetlenül következő állapot mindig egy lehetséges közvetlenül következő állapot.

Az ábrázolásnál problémát okoz a programokban levő ismételt műveletvégzés kezelése. Az ismétlés ugyanis egyetlen művelet. Ezzel szemben a

```
for i = 1 to 10 step 1
```

és a

```
next i
```

például két külön utasítássor.

Két módon szüntethetjük meg e látszólagos nehézséget. Ismétlődő operátor alkalmazásával, vagy pedig a program átírásával. Először az utóbbival foglalkozunk. A

```
10 for i = 1 to 10 step 1
```

```
20 let a = i*i + 1
```

```
30 print i
```

```
40 print a
```

```
50 next i
```

```
60 end
```

program kinyomtatott eredmény szempontjából egyenértékű a következő programmal:

```
10 let i = 1
```

```
20 if i > 10 then 80 else 30
```

```
30 let a = i*i + 1
```

```
40 print i
```

```
50 print a
```

```
60 let i = i + 1
```

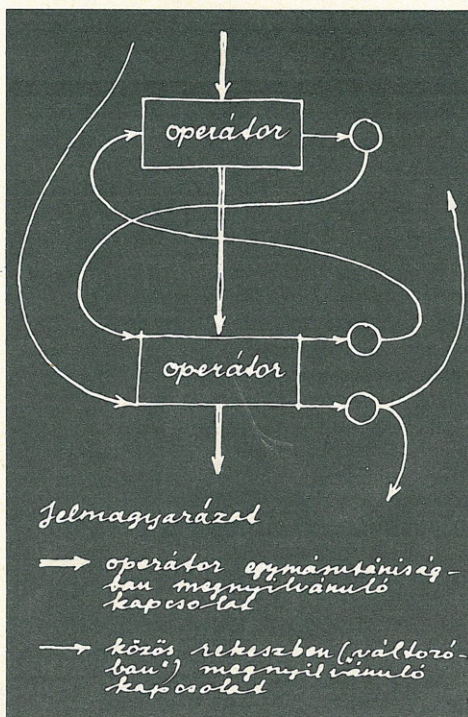
```
70 goto 20
```

```
80 end
```

Az utóbbi programnak viszont már gyerekjáték elkészíteni az állapotátmenet ábráját (8. ábra). A goto folytatási helykijelölő műveletet ábrázolhatjuk téglalappal is, de ezt el is hagyhatjuk; egyetlen nyilat (irányított vonalat) is használhatunk.

Az ismételt végrehajtásnak van egy másik gyakori módja is. Ez folytatási helykijelölés, úgy, hogy a kijelölt folytatási helyen folytatódik a munka addig, amíg egy visszatérési utasításra sor nem kerül. A visszatérési utasítás is folytatási helykijelölő utasítás, mégpedig olyan, hogy az előző folytatási helykijelölő utasítás sorának számát közvetlenül követő legkisebb sorszámú sort jelöli ki folytatási helyként.

Ezzel a lehetőséggel a BASIC programleíró nyelv elemeit változó bevezető részben, mint haladottabb lehetőséggel, még nem foglalkoztunk. Most azonban a tudáselmélyítés megkívánja, de önmagában is érdekes, hogy e hasznos eszközt megismerjük. Használatával ugyanúgy, mint a for-next esetében megoldhatjuk, hogy nem kell mindannyiszor leírni az ismétlődő programrészt, ahányszor sorra kerül. (Feladatunk gyakran nemcsak a nagy munkáigény miatt, hanem amiatt is megoldhatatlan



12. ábra

lenne, hogy nem mindig tudjuk előre megmondani, hogy mit hányszor kell majd ismételtetni a számológéppel.)

Célszerű, ha egy egyszerű példán szemléltetjük az elmondottakat.

```
10 input x
```

```
20 let a = x
```

```
30 gosub 110
```

```
40 let b = (3*x*x + x - 1) / (1 + 2*x*x)
```

```
50 let a = b
```

```
60 gosub 110
```

```
70 let c = (1 + x) * (1 + 2*x) * (1 + 3*x)
```

```
80 let a = c
```

```
90 gosub 110
```

```
100 goto 160
```

```
110 print a
```

```
120 print a*a
```

```
130 print a*a*a
```

```
140 print a*a*a*a
```

```
150 return
```

```
160 end
```

A programban háromszor kerül sor ugyanarra a műveletsorozatra, egy szám, négyzete, harmadik hatványa és negyedik hatványa kinyomtatására. A műveletek végrehajtási sorrendjét a 9. ábrán látható folyamatábrán szemléltettük. Ez az ábra működéstörténetnek is és állapotátmenet ábrának is felfogható.

A gosub az első, a return pedig a második folytatási helykijelölő utasítás. A gosub 110 például folytatási helykijelölés szempontjából egyenértékű a goto 110 utasítással. A return utasítás mindig a gosub 110 utasítás sorszámát közvetlenül követő sorszámú sort jelöli ki folytatási helyként. A gosub a goto subroutine (menj „alprogramra”) rövidítése. Az „alprogram” olyan programrész, olyan eljárás, amit általában többször kívánunk végrehajtani. Formai követelmény vele szemben csak az, hogy a végére return irandó. Végrehajtását pedig a go-

sub c utasítással indíthatjuk, ahol a c egész szám az alprogram (ismételendő programrész) első sorának száma. Ez az állapotátmenet ábra azonban jellegében eltér attól, amit az előbbiekben említettünk, abban a típusban (5. ábra) ugyanis minden állapot csak egyszer szerepelhet.

Az elmondottakból megállapítható, hogy az alprogramot használó ismétlés ábrájában az alprogramot annyiszor kell szerepeltetni, ahányszor a gosub a programban előfordul. (De nem szükségképp annyiszor, ahányszor a végrehajtás alkalmával sorra kerül.) Természetesen ésszerű az alprogramot grafikaiilag egyetlen operátorba összefoglalva ábrázolni (10. ábra).

Nem foglalkoztunk még az ismételt végrehajtás megoldásának ismétlődő operátor segítségével való lebonyolításával.

Az ismétlődő operátor tipikus példa az olyan operátorra, amelynek egy vagy több kapó kezében operátorok (sőt operátorcsapatok) vannak. Az operátor kapó keze, érzékelője matematikailag „független változóknak” vagy szerencsésebb kifejezéssel argumentumnak vagy operandusnak nevezhető. Az ilyen operátorok egyes operandusai tehát operátorok vagy operátorcsapatok.

Az ismétlődő operátornak „kezébe kell adni”, hogy mit ismétljen, és azt is kezébe kell adni, hogy ennek az ismétlésnek mik legyenek a jellemzői, különben nem fogja tudni, hogy mit kell csinálnia. Egy egyszerű példát mutat a 11. ábra az alábbi program esetére:

```
10 let s = 0
```

```
20 for i = 1 to 3 step 1
```

```
30 let s = s + i*i
```

```
40 next i
```

```
50 print s
```

```
60 end
```

Az operátorok és végrehajtásuk állapota között kölcsönösen egyértelmű megfeleltetés van. Így az imént kapott állapotátmenet-ábra-típus folyamat generálására, folyamat definiálására, folyamat leírására használható. Alkalmos a gép munkájának leírására, tehát definiálására. Ezért ez a módszer szemléletes programleíró „nyelvként” is használható. Az ezen a nyelven készített ábrát – helytelenül – folyamatábrának hívják, pedig nem ábrázolja a folyamatot, hanem generálására vagy definiálására használható. Célszerűbb ezért folyamatgeneráló vagy folyamatdefiniáló ábrának nevezni. (Azonban ez az elnevezés sem tökéletes, mert az ábra nem generál, nem definiál, hanem csak generálásra, definiálásra használható.)

Gyakran előnyös a tevékenységvégezési állapotátmenet ábrán (amely egymásutániságról informál kapcsolataira) más kapcsolatokat is ábrázolni, például azt, amely arról informál, hogy az operátoroknak mik a közös rekeszei, regiszterei („változói”). Ebben az esetben a különböző jellegű kapcsolatokat jelölő vonalakat jól megkülönböztethetően kell megrajzolni (12. ábra).

Az eddigi példánkban majdnem mindig egyszerre csak egy művelet folyt. A gépben azonban – mint tudjuk – mindig nagyon nagy számú művelet zajlik egyidejűleg. Ezek ábrázolási módjaival és a most kifejtettekre vonatkozó példákkal foglalkozunk majd a következőkben.

POGÁNY CSABA

# Miniprogramok

```

10 FOR k=0 TO 2*PI STEP 0.02
20 LET x=cos (k)+50
30 LET y=sin (k)+50
40 PLOT x+128,y+88
50 NEXT k

```

1. Az első program egy kört rajzol a képernyőre. Bár a ZX-Spectrumon találhatunk ennek a feladatnak megfelelő utasítást (CIRCLE), ezt azonban a későbbiek folyamán nem tudjuk majd alkalmazni. Programunk ugyan lényegesen lassabban működik, de a CIRCLE utasítással szemben több előnye is van, melyek később derülnek ki.

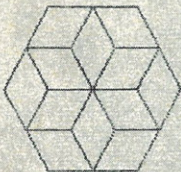
A kör ábrázolásához szükségünk van a sinus és a cosinus függvényekre. Mindössze arra vigyázzunk, hogy az argumentumot radiánban adjuk meg. A kör egyes pontjainak koordinátáit ciklusban számoljuk és rajzoltatjuk ki. Mivel a teljes szög  $2\pi$  radián, ezért a ciklus 0-tól eddig működik. A legideálisabb lépésköz 0,02. Hogy az így kapott 1 pontnyi sugarú kört ábrázolhassuk, 50-szeresére nagyítjuk. Középpontja a képernyő közepére kerül, ha felbontóképességének felét (128, illetve 88) adjuk a kiszámított koordinátákhoz. A kirajzolt kör sugara és helyzete a képernyő adta lehetőségeken belül a konstansokkal tetszőlegesen változtatható. Ha az x és y irányú szorzó nem egyezik meg, akkor ellipszist kapunk.

```

10 LET xx=50: LET yy=0
20 FOR k=0 TO 2*PI+.1 STEP PI/3
30 LET x=cos (k)+50
40 LET y=sin (k)+50
50 PLOT xx+128,yy+88
60 DRAW x-xx,y-yy
70 LET xx=x: LET yy=y
80 NEXT k

```

2. A kör helyett rajzoltassunk ki az ebbe a körbe írható szabályos sokszögeket. Vegyünk alappéldának egy hatszöget. Osszuk a kör területét hat egyenlő részre, és az így kapott pontokat rendre kössük össze egyenesekkel. Ezen elv alapján működik a program. Mivel most csak hat pont helyének meghatározására van szükség a teljes körön, ezért a lépés  $\pi/3$  lesz. A hatszög pontjait a DRAW utasítással kötjük össze. A program egy 50 pontnyi sugarú körbe írható szabályos hatszöget rajzol. Ha más sokszöget akarunk, akkor csak a lépés változik. Például: a háromszög  $2\pi/3$ , a négyzet  $\pi/2$ , az ötszög  $2\pi/5$  stb.

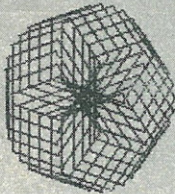


```

5 FOR t=0 TO 2*PI+.1 STEP PI/3
6 LET q=cos (t)+30
7 LET w=sin (t)+30
10 LET xx=30: LET yy=0
20 FOR k=0 TO 2*PI+.1 STEP PI/3
30 LET x=cos (k)+30
40 LET y=sin (k)+30
50 PLOT xx+(128+q),yy+(88+w)
60 DRAW x-xx,y-yy
70 LET xx=x: LET yy=y
80 NEXT k
90 NEXT t

```

3. A következő program az előző két-tőnek egy kombinált változata. A rajz síkban szemlélve egy csillag, térbe képzelve pedig sarokba állított kockák képét adja. A program az előző, most fel nem rajzolt szabályos hatszög csúcsaiba helyezi annak a körnek a középpontját, amelybe a fel nem rajzolttal azonos nagyságú hatszögeket rajzol.

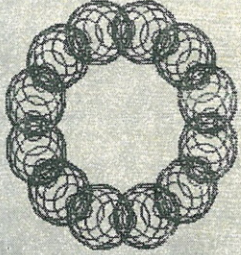


```

5 FOR t=0 TO 2*PI+.1 STEP PI/12
6 LET q=cos (t)+30
7 LET w=sin (t)+30
10 LET xx=30: LET yy=0
20 FOR k=0 TO 2*PI+.1 STEP PI/20
30 LET x=cos (k)+30
40 LET y=sin (k)+30
50 PLOT xx+(128+q),yy+(88+w)
60 DRAW x-xx,y-yy
70 LET xx=x: LET yy=y
80 NEXT k
90 NEXT t

```

4. A következő program egy fel nem rajzolt szabályos huszonnégyszög csúcsaiba helyezi annak a körnek a középpontját, amelybe szabályos ötszögeket rajzol. A két kör sugarának változtatásával más jellegű ábrákat kapunk, nem beszélve az ellipsziszre való áttérésről!

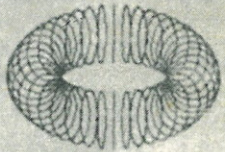


```

5 FOR t=0 TO 2*PI+.1 STEP PI/50
6 LET q=cos (t)+60
7 LET w=sin (t)+60
30 CIRCLE q+128,w+88,SIN (t+6)
*20
90 NEXT t

```

5. A gyöngysor programja egy kicsit más jellegű. Itt szinuszosan változó sugarú köröket helyezünk el egy fel nem rajzolt huszonnégyszög csúcsaira. A köröket, hogy programunk gyorsabb legyen, a CIRCLE utasítással rajzoltatjuk. Természetesen a ciklus lépését és a körök sugarát megadó konstansokat változtatva több kisebb-nagyobb gyöngyöt rajzoltathatunk, esetleg a köröket ellipsziszbe írt affin sokszög csúcsaira is helyezhetjük.



```

10 FOR t=0 TO 2*PI STEP PI/20
20 LET q=cos (t)+50
30 LET w=sin (t)+50
40 FOR t=0 TO 2*PI STEP PI/50
50 LET x=cos (t)+100
60 LET y=sin (t)+50
70 PLOT x+128+q,y+(88+w)
80 NEXT t
90 NEXT t

```

6. A tórusz programját az előzőek alapján már nem nehéz megérteni. Egy fel nem rajzolt ellipsziszre írtunk olyan köröket, melyeknek vízszintes sugara szinuszosan változik.

7. Ha a külső rajzoltató ciklust a fenti programoknál nem  $2\pi$ -ig visszük, hanem annak csak tört részéig, akkor is érdekes ábrákat kapunk.

E programok mintájára némi ötlettel bárki készíthet szép, érdekes képeket adó programokat.

ÉNEKES FERENC  
és tanítványai  
Képzőművészeti Szakközépiskola

Sorozatunkban ötleteket, módszereket adunk közre azok számára, akik már megismerkedtek mikrogépükkel, és/vagy azt munkájukban is használják. Reméljük, hogy e sorozat írásába ők is bekapcsolódnak; beküldött programozási fogásaikat – amennyiben valóban ötletesek – szívesen közzöljük.

## A verem felhasználási lehetősége

A BASIC interpreter egyetlen adattípust, a tömböt biztosítja a felhasználók számára. A tömbök segítségével a többi adattípus BASIC programmal modellezhető. A továbbiakban a verem felhasználási lehetőségeiről lesz szó (az angol stack terminológiát sokan zsáktárnak fordítják). A verem a gépi kód szintjén is igen hatékony, sőt nélkülözhetetlen eszköz; gyakran azonban az alkalmazói feladatok megoldása is megköveteli a verem, vagy legalább egyes tömbök veremszerű használatát.

A verem leginkább egy nagy bank páncélszekrényéhez hasonlítható, amibe mindenféle értékű dolgokat (adatokat) tehetünk. A páncélszekrényt őrző nyitja és csukja, és szigorúan ügyel arra, hogy abból csak az utoljára betett tárgyat (adatot) vehessük ki.

Ezzel az eljárással egész addig nincs is baj, míg csak olyankor akarunk kivenni valamit, amikor még van valami a páncélszekrényben, és csak akkor akarunk betenni valamit, ha még van üres rekesze. Ha a két szélsőséges eset valamelyike bekövetkezik, akkor a bank szervezésétől függően az őrző vagy udvariasan mosolyog, vagy halomra ló minket.

A fentiek elvégzésére az absztrakt programozási nyelvek a **push**, **pop**, **is-empty**, **is-full** utasításokat biztosítják. A **push**  $t_1$  utasítás a  $t_1$  kifejezés értékét betölti a verembe; a **pop**  $x$  utasítás a verem tetején levő értéket kivesszi és az  $x$  változóba tölti; az **is-empty** és **is-full** pedig a verem üres, illetve tele állapotát jelző logikai változók.

Ilyen típusú társzervezést BASIC-ben az **1. program** valósíthatunk meg.

A **10.** sor inicializálja a használt változókat, és létrehozza a  $V(\text{MAX})$  tömböt (a páncélszekrényt). A **GOSUB 1000** utasítás hatására a verem tetején levő elem az  $X$  változóba kerül, míg a **GOSUB 2000** az  $X$  értékét a verembe tölti. A két alprogram az **EMPTY** (üres) és a **FULL** (tele) változók értékét értelem szerűen állítja. A tömb nagyságát (**MAX**) a feladattól függően kell beállítani.

### 1. program

```

10 REM INICIALIZALAS
20 DIM V(MAX):VM=0:EMPTY=1:FULL=0
30 :
1000 REM POP X
1010 IF VM=0 THEN RETURN
1020 VM=VM-1:X=V(VM):FULL=0
1030 IF VM=0 THEN EMPTY=1
1040 RETURN
1050 :
2000 REM PUSH X
2010 IF VM=MAX+1 THEN RETURN
2020 V(VM)=X:VM=VM+1:EMPTY=0
2030 IF VM=MAX+1 THEN FULL=1
2040 RETURN
    
```

# Programozási fogások

A továbbiakban a fenti típusú verem szervezésére mutatunk példákat. A verem szervezése – a konkrét feladattól függően – minden esetben kicsit más.

## Hanoi tornyai

A Hanoi tornyai igen régi és közismert logikai játék. Egy pálcikán adott ( $N$ ) számú, egyre kisebb és kisebb korong található, amelyeket egy másik pálcára kell áthelyeznünk, ugyanebben a sorrendben. Ennek a műveletnek az elvégzéséhez egy további, harmadik pálcát áll rendelkezésünkre, mivel a korongok áthelyezésére a következő megkötést tesszük: egyszerre csak egy korongot tehetünk át egyik pálcáról bármelyik másikra, és kisebb korongra nem tehetünk nagyobb (lásd az **1. ábrát**).

Írjunk programot, amely megadja a szükséges lépéseket!

Kezdjük a megoldást csalással: egyszerre nem egy, hanem akárhány korongot áthelyezhetünk egyik pálcáról a másikra; de továbbra is vigyázzunk arra, hogy kisebb korongra nem tehetünk nagyobb.

Jelölje  $I \rightarrow J, D$  azt, hogy az  $I$ -edik pálcáról a  $J$ -edik pálcára  $D$  darab korongot egyszerre áthelyezünk. Például  $2 \rightarrow 3, 5$  azt jelenti, hogy a második pálcáról a harmadikra öt korongot átrakunk.

Így persze a feladatot könnyen megoldhatjuk, az egyetlen  $1 \rightarrow 2, N$  lépéssel. Most próbáljunk valamivel kevesebbet csinálni: maximum  $N-1$  korongot helyezhetünk át egyik pálcáról a másikra. Ebben az esetben a következőképpen járhatunk el:

- 1  $\rightarrow 3, N-1$
- 1  $\rightarrow 2, 1$
- 3  $\rightarrow 2, N-1$

Ezt az eljárást azután folytathatjuk. Az  $N-1$  korong áthelyezését átalakíthatjuk  $N-2$  korong áthelyezését tartalmazó lépésekké. Általában az  $I \rightarrow J, D$  lépést átalakíthatjuk három vele ekvivalens lépéssé:

- 1  $\rightarrow K, D-1$
- $I \rightarrow J, 1$
- $K \rightarrow J, D-1$

(ahol  $K$  a harmadik pálcák sorszáma  $I \neq K \neq J$ ). Az eljárást tovább folytatva végül csupa olyan lépést kapunk, amely csak egy korongot mozgat, azok pedig már szabályos lépések lesznek.

A **2. program** a fentebb vázolt, ún. rekurzív eljárást verem segítségével végzi el. Az  $I(100)$ ,  $J(100)$ ,  $D(100)$  tömbök elemeiben tároljuk az  $I \rightarrow J, D$  alakú lépéseket. Első lépésként (116.

```

100 REM *****
102 REM * HANOI TORNYAI *
104 REM *****
106 :
108 DIM I(100),J(100),D(100)
110 INPUT "ON:IN:PRINT":
112 :
114 REM KEZDOERTEKEK BEALLITASA
116 VM=0:I1=1:J1=2:D1=N:GOSUB 162
118 :
120 REM KOVETKEZO LEPEK KIVETELE
122 REM HA NINCIS MAR--> VEGE
124 GOSUB 172:IF E=1 THEN END
126 :
128 REM HA EGY KORONGOT KELL MOZGATNI
130 REM A LEPEK SZABALYOS ES KIIRJUK
132 REM HA NEM-->HARMON LEPESSSEL
134 REM HELYETTESITJUK
136 IF D=1 THEN GOTO 182
138 H=6-I-J
140 I1=H:J1=J:D1=D-1:GOSUB 162
142 I1=I:J1=J:D1=1:GOSUB 162
144 I1=I:J1=H:D1=D-1:GOSUB 162
146 :
148 GOTO 124
150 :
152 REM *****
154 REM * VEREMMUVELETEK *
156 REM *****
158 :
160 REM A LEPEK (I1,J1,D1) VEREMBE
TOLTESE
162 VM=VM+1:IF VM>100 THEN GOTO 166
164 I(VM)=I1:J(VM)=J1:D(VM)=D1:RETURN
166 PRINT "A VEREM TELE":END
168 :
170 REM A LEPEK (I,J,D) VEREMBOL VALO
KIVETELE
172 IF VM=0 THEN E=1:RETURN
174 E=0:I=I(VM):J=J(VM):D=D(VM)
176 VM=VM-1:RETURN
178 :
180 REM EGYETLEN SZABALYOS LEPEK
KIIRASA
182 PRINT I:"-->":J
184 GOTO 124
    
```

### 2. program

szor) az  $1 \rightarrow 2, N$  lépést tesszük a verembe. Ezt követően a **120–148.** sorokban levő ciklus segítségével végezzük el a lépések szabályos lépésekké való alakítását az előbbi helyettesítés felhasználásával. A veremben talált szabályos lépéseket kiírjuk, a szabálytalan lépést a fenti hárommal helyettesítjük. Mire a verem kiürül, az összes szükséges lépést kiírjuk.

(Feladat: a)  $N$  korong esetén legalább  $2^{N-1}$  lépésre van szükség. Miért? b)  $N$  korong esetén mekkora veremre van szükség? c) Tudunk-e nem rekurzív programot írni a feladat megoldására?)

## Nyolc királynő

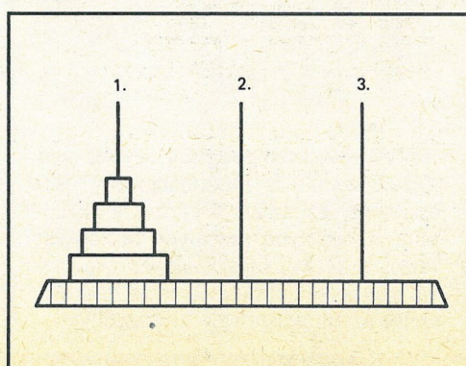
Írjunk programot, amely nyolc királynőt elhelyez a sakkasztalán úgy, hogy ne üssék egymást!

Fogalmazzuk át a feladatot: egy  $8 \times 8$ -as táblázatban nyolc pontot kell úgy elhelyezni, hogy bármely sorban, oszlopban és átlóban legfeljebb egy pont legyen.

A sakkasztala átlóit két részre bontjuk: lesznek felfelé menő és lesznek lefelé menő átlók. Ezeket a **2. ábra** szerint számozzuk meg.

A nyolc királynőt a következőképpen próbáljuk felállítani. Az elsőt az első oszlopba helyezzük el, a következőt a másodikba stb. A következő királynőt mindig először az első sorba kíséreljük meg letenni. Ha sikerül (azaz a már

1. ábra

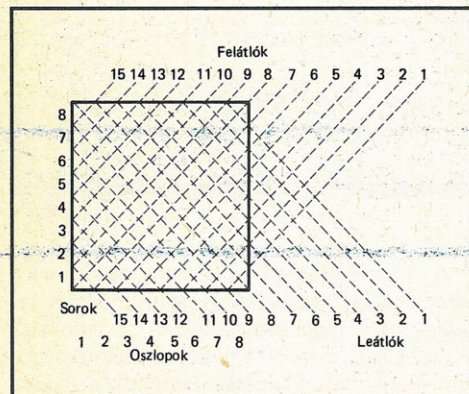




letettek nem ütök), lerakjuk. Ha nem sikerül, akkor próbálkozunk a második, harmadik stb. sossal. Ha semelyik sorba sem lehet lerakni, akkor féltre tesszük, visszatérünk az előző királynőhöz, és azt próbáljuk meg más (nagyobb számú) sorba elhelyezni. Ha ez sikerül, akkor újabb királynőt vehetünk, ha nem, akkor azt is le kell vennünk a tábláról.

Az ütések ellenőrzésére minden sorba, a fel- és lefelé menő átlóhoz hozzárendelünk egy-

2. ábra



3. program

```

100 REM *****
102 REM * NYOLC KIRALYNO *
104 REM *****
106 :
108 PRINT "J"
110 DIM S(8), J(8), F(15), L(15)
112 :
114 I1=1: VM=0: REM ELSO OSZLOP
116 :
118 IF I1=9 THEN GOTO 196
120 :
122 J1
124 GOTO 126
126 :
128 R
130 F
132 :
134 :
136 :
138 :
140 :
142 :
144 :
146 :
148 :
150 IF J1=0 THEN
152 REM NEM-->KOVETKEZO SORRA
154 J1=J1+1: GOTO 124
156 REM OSZLOP VISSZALEPES
158 REM AZ ATLOK ES A SOR FELSZABADITASA
160 GOSUB 188: I1=I1-1
162 IF E=1 THEN GOTO 204
164 GOSUB 212
166 F(F1)=0: L(L1)=0: S(J1)=0
168 GOTO 150
170 :
172 REM *****
174 REM * VEREMHUVELETEK *
176 REM *****
178 :
180 REM TOLTES A VEREMBE
182 VM=VM+1: J(VM)=J1: RETURN
184 :
186 REM VEREMBOL VALO KIVETEL
188 IF VM=0 THEN E=1: RETURN
190 E=0: J1=J(VM)
192 VM=VM-1: RETURN
194 :
196 REM A VEREM KIIRASA
198 GOSUB 188
200 IF E=1 THEN END
202 PRINT J1; " "; GOTO 198
204 PRINT "A FELADAT MEGOLDHATATLAN"
206 END
208 :
210 REM ATLOK INDEKENEK KISZAMITASA
212 F1=8+J1-1: L1=17-J1-1: RETURN
    
```

egy változót, amely jelzi, hogy a szóban forgó átlóban, illetve sorban áll-e már királynő. Erre az S(8), F(15), L(15) tömbök szolgálnak. Végül szükségünk lesz egy veremre, amely az eddig felállított királynők sorszámaát tartalmazza. Ez a V(8) tömb.

Nézzük most a 3. programot. A 114., 118. sorokban az első királynőt az első oszlop első sorába kíséreljük meg elhelyezni. Általában a 134-138. sorok ellenőrzik, hogy lerakható-e a királynő, vagy sem. Ha igen, akkor lerakjuk, ami a verembe való írást, a sor, illetve a kétfajta átló lefoglalását jelenti. Ha nem, akkor a következő sossal kísérletezünk. Ebben az esetben azonban egy már lett királynőt fel kell emelnünk. Ez a veremből való kivételt, a sor és az

átlók felszabadítását jelenti. Ezt végzik el a 160-166. sorok.

A program a gép sebességétől függően több, mint egy percig fut.

(Feladat: a) Írjuk át úgy a programot, hogy a végeredményt a sakkírás szabályai szerint A1, G6 stb. alakban írja ki! b) Módosítsuk a programot úgy, hogy egy 9 x 9-es sakk táblán helyezzen el 9 vezért! c) Módosítsuk úgy a programot, hogy az összes lehetséges felállítást megadja! d) A 134-138. sorok többféleképpen is átírhatók egyetlen sorrá. Hogyan? e) A verembe töltő programrész nem ellenőrzi, tele van-e a verem. Miért?)

DR. ÚRY LÁSZLÓ

## RENDEZÉSEK

# Commodore 64-en

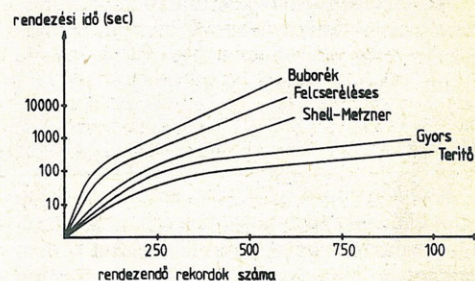
Adatok rendezésére több algoritmus is ismert. Ezek közül mutatunk be néhányat, a Commodore 64 BASIC nyelvén.

A rendező szubrutinok hívásához néhány paramétert kell megadni. A programot RUN 2000-rel indítjuk a 2060 sor szubrutin hívását változtatgatva. A kiírt rendezési idő az egyes algoritmusokon belül a rendezendő tételek számától és a rendezési kulcs hosszától (a tételek értékétől) függ.

Az egyes algoritmusok rendezési ideje a tétel szám függvényében a grafikonon látható. A grafikon is jól illusztrálja, hogy míg kis tételnelkül az egyszerűbb eljárások is megfelelők, addig nagy tétel szám esetén csak a kifinomultabb módszerek hatékonyak.

Rendező eljárásokról részletes elemzést például N. Wirth: Algoritmusok + Adatstruktúrák = Programok. (Bp. 1982. Műszaki Könyvkiadó) II. fejezetében olvashatnak.

JARABEK LAJOS



```

10 REM N = A RENDEZENDO REKORDOK SZAMA
20 REM A$(J) = A RENDEZENDO REKORDOK VEKTORA
30 REM *****
40 REM FELCSERÉLÉSEK RENDEZES
50 REM *****
60 FOR J=1 TO N-1
70 FOR K=J+1 TO N
80 IF A$(J)>A$(K) THEN TEMP$=A$(J): A$(J)=A$(K):
90 A$(K)=TEMP$
90 NEXT K
100 NEXT J
110 RETURN
200 REM *****
210 REM BUBORÉK-RENDEZES
220 REM *****
230 FOR J=N-1 TO 1 STEP -1: FIN=-1
240 FOR K=1 TO J
250 IF A$(K)>A$(K+1) THEN FIN=0: TEMP$=A$(K):
260 A$(K)=A$(K+1): A$(K+1)=TEMP$:
270 NEXT K: IF NOT FIN THEN NEXT J
280 RETURN
300 REM *****
310 REM SHELL-METZNER RENDEZES
320 REM *****
330 M=N
340 M=INT(M/2): IF M=0 THEN RETURN
350 J=1: K=N-M
360 I=J
370 L=I+M
380 IF A$(I)>A$(L) THEN TEMP$=A$(I): A$(I)=A$(L):
390 A$(L)=TEMP$: I=I+M: IF I>L THEN FIN=0
400 GOTO 360
500 REM *****
510 REM TERITO RENDEZES
520 REM *****
525 INPUT "REKORDOK ATL.HOSSZA"; LE
530 LE=64: U=91: Z=0: K=0: F1=0: TE$="": PL=0: PH=0
535 LE=LE+3
540 B=FR(0)/LE**N
550 IF B<2 THEN PRINT "KEVES A MEMORIA": STOP
560 IF B>4 THEN B=4
570 PL=PEEK(46): PH=PEEK(47)
580 DIM B$(B**N+30)
    
```

```

590 Z=B**N/(U-L)
600 FOR J=1 TO N: K=(ASC(A$(J))-L)*Z
610 IF B$(K)="" THEN B$(K)=A$(J): NEXT: GOTO 630
620 K=K+1: GOTO 610
630 J=1: FOR K=1 TO B**N+29: IF B$(K)="" THEN NEXT:
640 GOTO 650
640 A$(J)=B$(K): J=J+1: NEXT
650 POKE 46, PL: POKE 47, PH
660 FOR J=N-1 TO 1 STEP -1: FIN=-1
670 FOR K=1 TO J
680 IF A$(K)>A$(K+1) THEN FIN=0: TEMP$=A$(K):
690 A$(K)=A$(K+1): A$(K+1)=TEMP$:
700 NEXT K: IF NOT FIN THEN NEXT: RETURN
800 REM *****
810 REM GVORS RENDEZES
820 REM *****
830 DIM ST((LOG(N)/LOG(2)+4)*.1)
840 S=1: ST(1,0)=1: ST(1,1)=N
850 L=ST(S,0): R=ST(S,1): S=S-1
860 J=L: K=R: X$=A$(L+R)/2
870 IF A$(J)<X$ THEN J=J+1: GOTO 870
880 FOR V=1 TO 16: IF A$(K)>X$ THEN K=K-1: NEXT
890 IF J=K THEN J=J+1: K=K-1: GOTO 870
900 IF J<K THEN TEMP$=A$(J): A$(J)=A$(K):
910 A$(K)=TEMP$: J=J+1: K=K-1: GOTO 870
910 IF J<R THEN S=S+1: ST(S,0)=J: ST(S,1)=R
920 R=K
930 IF L<R THEN S=S+1
940 IF S=0 THEN S=0
950 RETURN
2000 INPUT "REKORD SZ. "; N
2010 DIM A$(N)
2020 J=RN(0-1)
2030 FOR J=1 TO N: A$(J)=CHR$(65+RND(1)*26)+
*****
2040 NEXT
2050 T=I
2060 GOSUB 10
2070 PRINT (T-I)/60
2080 FOR I=1 TO N
2090 PRINT A$(I)
2100 NEXT I
2110 END
    
```

# Termékiismertető

SOROK	OSZLOPOK				
	1. oszlopnév	2. oszlopnév	3. oszlopnév	4. oszlopnév	5. oszlopnév
1. sornév					
2. sornév					
3. sornév					
4. sornév					

1. ábra

MICROMAGAZIN TEST				
OSZLOP	S1	S2	S3	S4
JANUAR	3	4	1	8
FEBRUAR	5	4	3	1
MARCIVS	5	4	3	6
APRILIS	9	1	7	2
MAJUS	8	1	2	4
JUNIUS	3	3	2	1
JULIUS	1	1	2	-1
AUGUSZTU	3	4	6	1
SZEPTEMB	9	6	7	-3
OCTOBER	7	4	2	0
NOVEMBER	8	5	1	-2
DECEMBER	10	9	9	7

MEGJ.:  
1984.06.08

2. ábra

## Üzleti grafika

Azt a megtisztelő feladatot kaptam a Magazintól, hogy ismertessek és objektíven értékeljek egy olyan szoftverterméket, amelyet nem én írtam. Hogy ez milyen nehéz feladat, azt csak profi programozótársaim érthetik igazán. Ugyanis mások szemében a szálkát, ... Ugye értik?

Komolyra fordítva a szót, számomra a feladat mindenképpen szokatlan volt. A program kipróbálásához ugyanolyan feltételeket kértem és kaptam, mint egy átlag felhasználó (később – mint olvasni fogják – bele is estem egy olyan csapdába, amely minden kíváncsi felhasználóra leselkedik). Így nem kértem listákat, külön leírásokat, nem voltam kíváncsi a szerzők személyére, és a Commodore 64 számítógépet sem ismerem jobban, mint egy átlag felhasználó. Ennek ellenére egész sor ötletem támadt arra vonatkozólag, hogy a szóban forgó, nagyon figyelemreméltó programterméket hogyan lehetne továbbfejleszteni – hiszen a kibicnek semmi sem drága.

Hátrányomra volt viszont, hogy nem dolgozom olyan helyen, ahol ezt a programterméket kellene vagy lehetne használni, ezért gazdasági jelentőségét csak találgatni tudom.

### A termék külső jegyei

A program neve: Üzleti grafika, termelési, gazdaságossági mutatók ábrázolására és értékelésére szolgáló programrendszer.

A hardver konfiguráció:

- Commodore 64 számítógép
- grafikus lehetőségekkel rendelkező nyomtató (például SEIKOSHA GP 100 VC)
- egy darab VIC-1541 hajlékonylemez-egység
- monitor vagy színes televízió (fekete-fehér is használható)

A programtermék készítője: A rendelkezésemre bocsátott hajlékonylemezen ez a név,

illetve cím áll: ÁSZ INTERSYSTEM Számítástechnikai Szkcs. 1025 Bp., Törökvész u. 55.

A terjesztő: NOVOTRADE RT. 1136 Bp., Fürst S. u. 24-26.

A programtermék ára: 12 900 forint (beleértve a hajlékonylemezt is).

### A kiindulás és a termék ismertetése

A NOVOTRADE RT emblémájával megjelölt, izléses nyeltesekben vehettem át a programot tartalmazó, 5/4 hüvelykes hajlékonylemezt, és egy példányban a felhasználói dokumentációt. Külön kérésre kaptam Commodore-dokumentációt is. Félttem, hogy a program működtetéséhez alaposabb Commodore-ismertetek kellenek, de tévedtem. Ha valaki a Commodore gépet üzembe tudja helyezni, a programot is használni tudja.

Azt, hogy a programtermék kezelése mennyire egyszerű, mi sem bizonyítja jobban, mint hogy 4-5 óra gépidő-felhasználás tökéletesen elegendőnek bizonyult a viszonylag mély megismeréséhez. Ez idő alatt szisztematikusan végigpróbáltam azokat a helyzeteket, amelyeket kritikusnak véltem. Találtam néhány – a gyakorlati felhasználásban kis valószínűséggel jelentkező – hibát, amelyet remélhetőleg a cikk megjelenése idejére már kijavítottak a szerzők. A program jó kezelhetőségét bizonyítandó el kell mondanom, hogy a felhasználói leírást csak a program megismerése után olvastam el. Fel is fedeztem benne pontatlanságokat (fontos dolgok hiányát), de hangsúlyoznom kell, hogy semmiféle nehézséget nem okozott a termék kezelése.

Egy programtermék ismertetése nagyon nehéz: nyilvánvalóan a prospektusnál bővebbnek és a felhasználói leírásnál szűkebbnek kell lennie.

A programtermék általános célkitűzéseit könnyű megfogalmazni. Nagyon gyakran találkozzunk táblázatba foglalt adatokkal. A táblázat sorai és oszlopai mennyiségek sorozatait tartalmazzák, amelyeket éppen azért foglaltak egyetlen táblázatba, hogy egymás viszonylatában fejezzenek ki valamit. Az arányok (viszonylatok) számok formájában nem túlságosan kifejezőek, különösen akkor nem, ha a táblázat viszonylag sok elemet tartalmaz. Nos, ilyenkor segít a grafika és a (jó öreg) Descartes-féle koordinátarendszer. Egy jól sikerült grafikon sok olyan dolgot képes érzékeltetni, amelyet egy számsorozat nem. A programtermék arra vállalkozik, hogy táblázatokat jelenítsen meg. Ehhez több grafikai lehetőséget kínál.

Idézet a programtermék felhasználói leírásából: „Célja: A programrendszerrel vezetői döntésekhez szükséges grafikonokat, függvényeket ábrázolhat. Ezek adatait a program táblázatos formában várja. A megjeleníteni kívánt ábrához az adatokat tartalmazó táblázatot be kell vinni. A táblázatban a független változó (x) értékei és a függvényértékek szerepelnek.

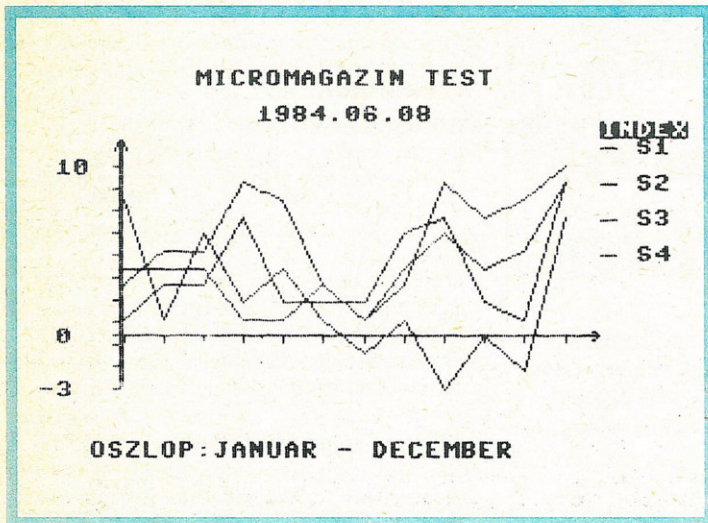
A grafikontervezés és -rajzolás gondjait veszi le a felhasználó válláról az ÜZLETI GRAFIKA programrendszer.”

Nem óhajtom elemezni az idézetet (ti. például azt, hogy a vezetői döntésekhez miért kellenek grafikonok), de úgy gondolom, bizonyos dolgok magyarázatra szorulnak. Ugyanis a felhasználói leírás meglehetősen gyengére sikerült (jellemző módon egyetlen ábra sincs benne, pedig grafikus programrendszert ír le), ráadásul az én példányom egy oldala hiányzik. Szerencsére, mint fentebb említettem, használatára szinte egyáltalán nincs szükség.

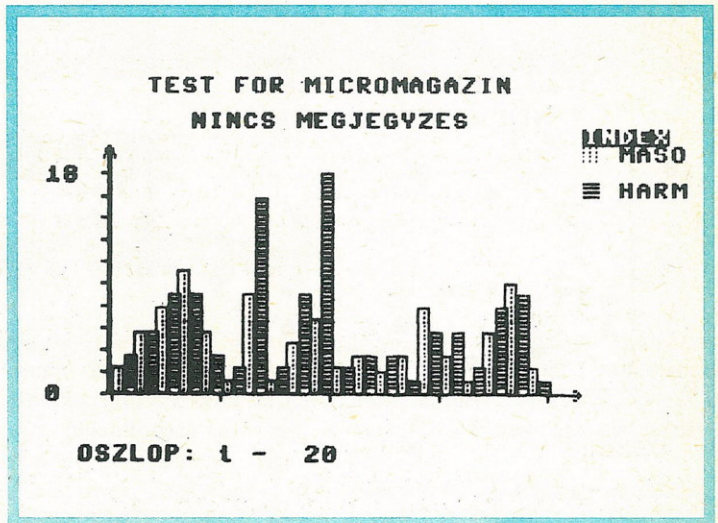
### A programrendszer bemenete

A bemenet (input) egy táblázat, amelynek megnevezései az 1. ábrán találhatóak.

Az ábrán látható sémára hivatkozva meg lehet adni az „oszlopok” helyett egy nevet (hónapok, évek, telephelyek stb.), amelyet a program által küldött „x neve” üzenetre adott válasszal definiálunk.



3. ábra

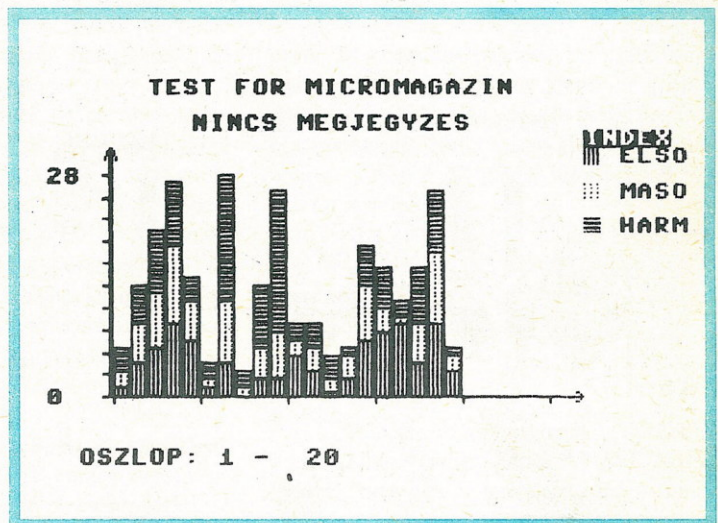


5. ábra

OSZLOP	ELSO	MASODIK	HARMADIK
1	1	2	3
2	4	5	5
3	6	7	8
4	9	10	8
5	7	5	3
6	1	1	2
7	4	8	16
8	0	1	2
9	2	4	8
10	2	6	18
11	5	2	2
12	3	3	3
13	1.3	1.8	2.9
14	2	3	1
15	7	7	5
16	8	3	5
17	9	1	2
18	4	5	7
19	9	9	8
20	3	2	1

MEGJ. NINCS MEGJEGYZES

4. ábra



6. ábra

A „táblázatnév” és a „megjegyzés” max. 39 karakter lehet.

Legfeljebb 15 adatsort adhatunk meg. Minden sornak adhatunk nevet (amennyiben ezt nem tesszük meg, úgy az 1. sor „Y1”, a 2. sor „Y2” stb. elnevezést kap automatikusan).

Az oszlopok „elnevezésére” két módszer adott. Az elsónél a programrendszer feltételezi, hogy minden sor egy-egy meghatározott függvény függvényértékeit tartalmazza. Tehát az oszlopok „nevei” tulajdonképpen az x tengely diszkrét pontjai által reprezentált számértékek, amelyek azonos lépésközzel követik egymást. Ehhez meg kell adni a kezdő és a végértéket, valamint a lépésközt. Ez a három adat meghatározza az oszlopok számát.

A második lehetőség, hogy az oszlopoknak „szöveges” nevet adunk (például január, február, ...), amely „szöveg” természetesen szám is lehet. Ezeket a neveket felsoroljuk. Az így felsorolt értékek azonos lépésközzel követik egymást a kirajzolt grafikonokon.

A táblázatot háromféleképpen lehet a programrendszernek megadni: közvetlenül begépeléssel; névvel, amennyiben egy ilyen táblázatot előzőleg a hajlékonylemezen eltároltattunk a programrendszerrel; idegen programok kime-

neteként (outputjaként), amennyiben azok hajlékonylemezes kimenete betartotta a felhasználói leírásban megadott konvenciókat.

A táblázat javítható, szűkíthető, törölhető (sajnos bővítési lehetőséget nem fedeztem fel). Ezek az editálási (update-olási) lehetőségek viszonylag kényelmesek is.

Az egyetlen kritikai megjegyzésem az, hogy nem tudom, vajon az „idegen” programoktól elvárt konvenciók egyeznek-e bármely kész és népszerű Commodore 64 programtermék lemezes kimenetével (például CalcResult elektronikus feladatlap programcsomag). Ugyanis erre nem találtam utalást a leírásban.

### A táblázat nyomtatása

A táblázatot nem „természetes” módon nyomtatja a rendszer. Abból indulhattak ki az alkotók, hogy az oszlopok száma általában lényegesen több, mint a soroké, tehát a táblázatot „fordítva” nyomtatja, azaz az oszlopok sorokká, a sorok oszlopokká váltak. A 2. ábrán egy olyan táblát láthatunk kinyomtatva, amelynek sorai: S1, S2, S3, S4, míg oszlopait „szövegesre” definiáltuk, rendre január, ..., december. A tábla címe: „Micromagazin test”, a megjegy-

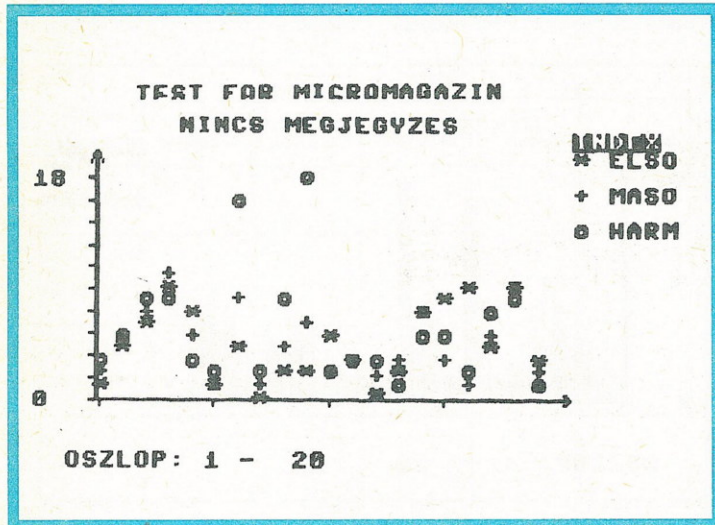
zés pedig a készítés dátuma. A 3. ábrán az ún. „vonalas ábrázolás” látható erre a táblára. Az „x-független” változó elnevezése: „oszlop”.

### A programrendszer grafikai kimenete

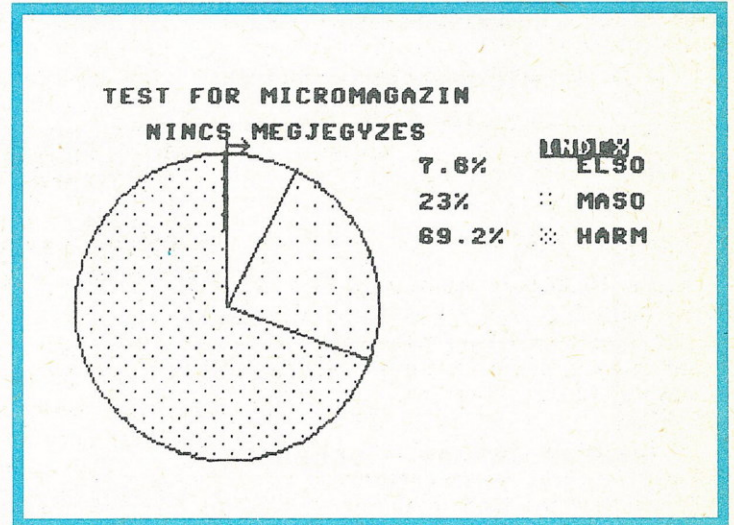
E fejezet címében indirekten utalok arra, hogy nemcsak grafikai kimenete van a programrendszernek. Minden megjelenített ábrát el lehet tároltatni a hajlékonylemeze, ahonnan az bármikor előhívható és megjeleníthető. Ezzel a kérdéskörrel a továbbiakban nem foglalkozom, egyszerűen tudomásul veszem, hogy ebben a figyelemreméltóan kompakt rendszerben „természetesen” ezt is lehet.

Minden grafika, ami képernyőn megjelenik, ki is nyomtatható. A képernyőn megjelenő ábrák színességét a nyomtatásban különböző raszterezéssel helyettesíti a programrendszer. Éppen ezért használható fekete-fehér képernyővel is. Ez azonban sajnos nem mindenhol biztosítható. A 3. ábrán látható „Vonalas ábrázolás” esetén például ez nincs így.

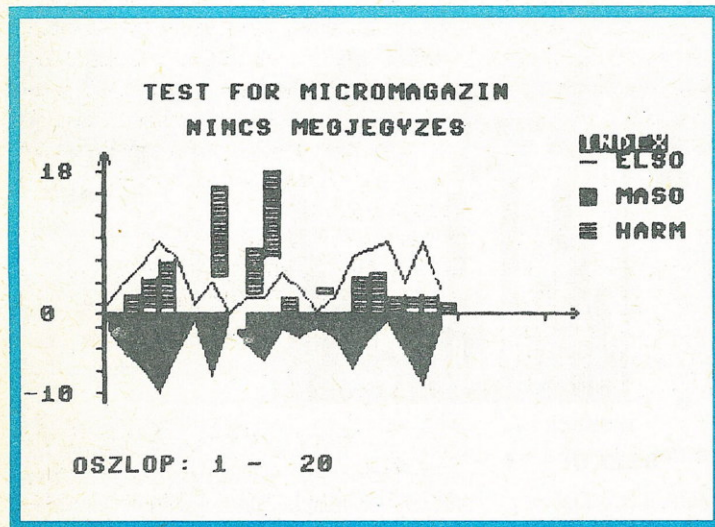
Minden sor egy-egy ábrázolható függvény. Vannak olyan esetek is, amikor a függvények kapcsolatát kell ábrázolni. Először foglalkozunk azokkal az esetekkel, amikor a függvé-



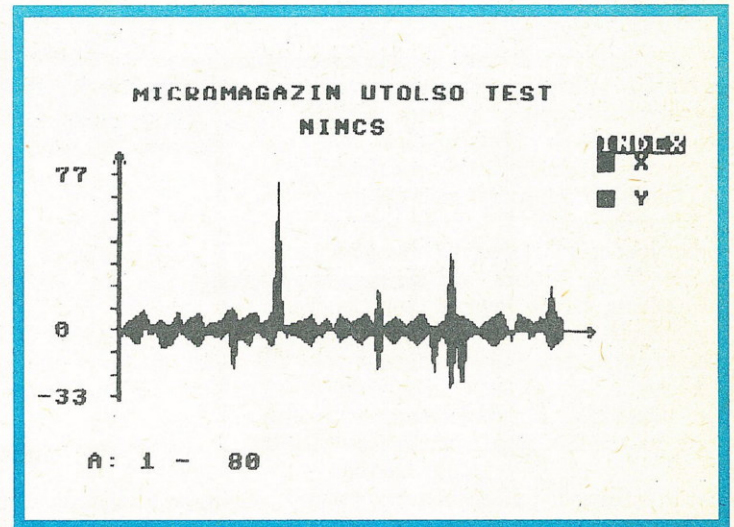
7. ábra



9. ábra



8. ábra



10. ábra

nyek önmagukban érdekelnek bennünket. Ilyenkor vagy eleve egysoros a táblázatunk, vagy annak egy konkrét sorát akarjuk ábrázolni.

Lehetőség van kitöltött (8. ábra x tengely alatt és 10. ábra), oszlop (5. ábra), karakteres (7. ábra), vonalas ábrázolásra.

Amennyiben több sort akarunk együtt megjeleníteni, de nem kívánunk oszlopokra vonatkozó összefüggéseket összevontan ábrázolni, akkor két megoldás közül választhatunk: a) Minden függvényt (minden sort) ugyanolyan módon akarunk ábrázolni. Rendelkezésre áll a vonalas, oszlop, karakteres ábrázolási mód (3., 5., 7. ábra); b) A sorokat az x tengely mindkét oldalán különböző vagy egyforma ábrázolásban kérhetjük (8. és 10. ábra). Egyenként minden függvényre ugyanazok a lehetőségek, mint az egyedüli függvényekre.

Az egyes függvények közötti összefüggések összevont ábrázolására is két lehetőség kínálkozik. Az egyik az osztott oszlop (6. ábra), amely az egyes sorok megfelelő oszlopait egymásra helyezi, így az oszlopok összegeit ábrázolja. Korlátja, hogy nem lehet a táblázat ábrázolt részében negatív érték. A másik lehetőség a kördiagram (9. ábra), amely egy oszlop érté-

keinek százalékos megoszlását ábrázolja, azaz a kiválasztott oszlop összegéhez viszonyítva le rajzolja a százalékos megoszlást. Nem értem, miért nem megengedett a kördiagramos ábrázolás egy sorra vonatkozólag. Az is érdekes lehet, hogy egy soron belül milyen súllyal szerepelnek az egyes értékek.

A képernyőn való ábrázolás csinosítására többféle lehetőség szolgál: az ábrákba bárhova szövegeket lehet írni, a képernyőn megválasztható a felirat színe is; az ábraelemek színeit vagy a raszterezést cserélni lehet; cserélhető a háttérszín; el lehet tüntetni az ábrák alól a menüfeliratokat (amelyek a nyomtatón eleve nem jelennek meg); mód van az index eltüntetésére is (az ábrákon inverzképpen jelenik meg); és végül megválasztható a nyomtatás példányszáma.

## Egy példa

A továbbiakban nézzünk egy példát. Először magát a táblát (4. ábra), kiemelve a kördiagram által ábrázolt oszlopot, majd az „oszlopábrázolás”-t (5. ábra). Az utóbbi esetben csak két sort láthatunk, mert a harmadik sor nem fért el (ezt

a programrendszer közölte). Ezután nézzük az ún. „osztott oszlop” ábrázolást, az ún. „karakteres ábrázolás”-t (7. ábra), majd a „kombináció”-t (8. ábra) és a „kördiagram”-ot (9. ábra).

Hogy fogalmat alkothassunk az ábrázolás felbontóképességéről, tekintsük meg végül a 10. ábrát. Az x tengely felett az „x”, alatta az „Y” függvény. Az oszlopok neve „A”.

## Korlátok, megjegyzések

Az Üzleti grafika terméknek vannak korlátai (és hibái is, amelyeket már remélhetőleg kijavítottak). E korlátok közül a legnagyobb az, hogy a képernyőn és így a papíron megjelenő kép mérete is meglehetősen korlátozott. Ezért bizonyos dolgokat már nem lehet ábrázolni (ilyenkor a program jelzi, hogy „túl sok adat”).

A másik észrevételem, hogy a rendszer szinte sugallja, hogy be kellene ágyazni azt egy word-processing programba. Az ábrák A/4-es lapokra valók, és nyilvánvaló, hogy méretüknél fogva leggyakrabban szövegbe ágyazva kerülnek felhasználásra.

A rendszerben vannak felesleges dolgok (például az ún. karakteres ábrázolás), ugyanakkor

értelmes módon tovább lehetne bővíteni (a sokra vonatkozó kördiagrammal, függvények vagy/és kapcsolatainak ábrázolási lehetőségeivel stb.).

Mindenképpen kifogásolom, hogy a felhasználó csak a program újratöltésének árán tud kilépni minden megkezdett műveletből.

## Összefoglalás, értékelés

Kedves olvasó! Ha a bemutatott grafikonábrázolási lehetőségekre szüksége van, vegye meg a NOVOTRADE RT-től ezt a programot! Őszintén írom, nagyon élveztem a vele való „játékot”, hiszen tényleg játszva tanultam meg használatát.

A rendszer nagyon kompakt, rendkívül könnyen kezelhető, nagyon elegáns mind megjelenésében, mind funkcióiban. Ritkán találkozhatunk olyan programmal, amelyben ennyi megvalósított ötlet van, amely a maga szűk területén ennyire komfortos és elegáns. Majdnem természetes, hogy vannak ötleteim arra, hogy mit kellene elhagyni belőle, mit kellene másként csinálni és mivel kellene bővíteni. Hangsúlyozom azonban, hogy alapvetően csak felsőfokon nyilatkozhatom a szerzőkről; nagyszerű dolgot csináltak ebben a kategóriában.

Szólnom kell a dokumentációról is. Meglepően (vagyis ismerve a hazai dokumentálási színvonalat, nem is meglepően) primitív kiállítású, pontatlan, egyáltalán méltatlan a termékhez.

Ha magát a terméket osztályoznom kellene, ötöst adnék rá. Egy jól meghatározott területet fed le teljesen, rengeteg ötlettel, úgy, hogy kezelését a számítástechnikában járatanok is órák alatt elsajátíthatják. Igazi személyiszámítógépprogram ez.

## Epilógus

A számítástechnika olyan, mint a KRESZ. Ha valami nem tiltatik, akkor (az ésszerűség határain belül) azt szabad csinálni.

Megkísértem kilistázni annak a hajlékonylemeznek a katalógusát, amelyen a programot kaptam. Nem hagyta magát. Mivel seholy sem tiltották meg, elkezdtem keresgélni a lemezen, és találtam is egy „E”-vel kezdődő nevű programot, amelyet betöltve, egy igen hosszú BASIC programot listázhattam ki a képernyőre. Utána elindítottam ezt a programot, amely hosszú futás után „Nana, kis kíváncsi!” (vagy valami hasonló) szöveggel lepelt meg, és a lemezen lévő értelmes program megsemmisült. Az ezek után listázhatóvá vált katalógus egyetlen nulla hosszúságú eleme az „Ez cégünk ajándéka” lett. Fanyar humor, mint mondják!

A kérdésem csak az lenne a „ravaszág” (vagy inkább szemtelenség) kiagyalójához: miért nem tiltják meg azt, amit csináltam? Például a következőképpen:

```
„Felhívjuk a figyelmet arra, hogy tilos a  
LOAD ”E*”,8
```

```
RUN
```

műveletsort végrehajtani, mert megsemmisül a lemezen lévő programtermék”.

Úgy vélem, ez lenne a korrekt dolog.

SIMON IVÁN

## ZX-SPECTRUM

# Hivatásos amatőröknek

A világ jelenlegi mikroépgyártása meglehetősen bonyolult képet mutat. Szinte naponta kerül piacra egy-egy új géptípus. Így a személyi számítógép kategória egyre több altípusra hasad szét. Ma már vannak professzionális, hobbi, asztali, házi stb. számítógépek. Ezeket a termékcsoportokat nem is igen lehet pontosan körülhatárolni. Egy professzionális gépet ugyanis nemcsak a tárméret, a sebesség különbözteti meg a játékgépektől, hanem megbízhatósága, perifériáinak jobb használhatósága, jobb szoftvere és így tovább.

Mindazonáltal a fentiek figyelembevételével két nagy csoportba sorolhatjuk a személyi számítógépeket: a professzionális és a házi számítógépek csoportjába. A Sinclair cég ZX-Spectrum nevű gépe jellemzői alapján kétségkívül az utóbbiak, a hobbi, játék-számítógépek közé tartozik. Ez természetesen nem jelenti azt, hogy apró adatfeldolgozási vagy tudományos számítások nem végezhetőek vele, csupán arról van szó, hogy az ilyen jellegű felhasználása korlátozott.

Saját kategóriájában a Spectrum meglehetősen elterjedt gép. A CHIP című nyugatnémet számítástechnikai magazin bestseller listáján az ötödik helyet foglalja el. Csak a Commodore és az Atari gépei előzik meg (lásd az 1984. júliusi számot).

Ha a Sinclair cég gépeit szemügyre vesszük, azt látjuk, hogy a Spectrum logikus továbbfejlesztése a korábbi ZX81 gépnek. Ennek első verziója szinte még zsebszámológép. A cég azonban szeretne betörni a nagyobb gépek piacára is. Ezt mutatja, hogy a Spectrum után kifejlesztett gépe, a QL már bizonyos professzionális vonásokat is mutat. A Sinclair gépeinek versenyképességét egyébként elsősorban olcsóságuk biztosítja.

A Spectrum méretei olyanok (kb. 24 × 15 cm és mindössze 3 cm vastag), hogy akár aktatászkában is hordozható. Alapvető perifériái – tévékészülék és kazettás magnetofon – már hétköznapi eszköznek számítanak. A gyártók mindent megtettek, hogy bizonyos határokon belül a gép a legtöbb tévé- és magnetofontípussal működtethető legyen.

## A hardver

A Spectrum Z80A típusú mikroprocesszort tartalmaz, amelyet 3,5 MHz-cel működtetnek. Tára 16 kb-ot ROM (ez tartalmazza a BASIC

rendszert) és – a vevő igényeitől függően – 16 vagy 48 kb-ot RAM. Mindez nem is hangzik rosszul, tekintve a gép Nyugaton olcsó árát, de egy számítógép használhatósága nemcsak ezektől függ.

A gép beépített PAL rendszerű színkeverőt tartalmaz a tévékészülékhez. Ezért nálunk csak a kétnormás színes készülékkel kaphatunk valóban színes képet. Beépített interfész vezérli a mágneskazetta be/kimenetét is. Minden egyéb perifériát a be/kiviteli porthoz kell csatlakoztatni megfelelő interfészen keresztül (például RS232, NET). Az alapgép nem drága ugyan, de egy tisztességes konfiguráció kiépítéséhez nemcsak a perifériákat, hanem interfészeket is be kell szereznünk.

A gép beépített programozható hanggenerátort tartalmaz. Ezen egyszerű dallamok is eljátszhatók, mert az időtartam és a hangmagasság szabályozható.

## A billentyűzet

A ZX-Spectrum billentyűzete meglehetősen mesterkéltnek tűnik. Ennek oka az, hogy a gyártók szerették volna elérni, hogy minden BASIC alapjelet, függvénynevet stb. egyetlen billentyű lenyomásával lehessen a gépbe bevinni. Ezek száma a kis- és nagybetűkkel, egyéb jelekkel együtt olyan nagy, hogy csak meglehetősen bonyolult shift rendszer segítségével lehet egy szokványos billentyűzeten elhelyezni.

E rendszer lényege az, hogy különböző beviteli módok vannak, és az egyes beviteli módokban ugyanazon billentyű lenyomása más-más jel bevitelét jelenti. A négy beviteli mód mellett a szokásos kisbetű-nagybetű shiften (CAPS SHIFT) kívül még egy shift gomb (SYMBOL SHIFT) alkalmazható. Mindez a billentyűk számát mintegy megtizenkétszeresíti. Ennyi lehetőségre természetesen nincs szükség, így a rendszer még komplikáltabbá válik.

A gép egy villogó betűt alkalmaz kurzorként, amely megmutatja, hogy milyen beviteli módban vagyunk. A „betű” beviteli módban azonban más kurzorbetű mutatja a kis-, illetve a nagybetűk bevitelét, így a felületes szemlélő azt hisheti, hogy öt beviteli mód van. Az is bonyolítja a dolgot, hogy a rendszer bizonyos helyzetben automatikusan vált beviteli módot. Szokatlan, hogy a beviteli mód váltását egyik esetben két shift billentyű egyidejű lenyomásá-

# Terméki mertető

val végezhetjük, holott már az írógép ahhoz szoktatja az embert, hogy egy „shift” billentyű csak egy „jel” billentyű lenyomásával együtt értelmes.

A helyzet persze nem olyan rossz, mint ahogy az a fentiekből kivehető. A tervezők igyekeztek annyi józan praktikumot belevinni a rendszerbe, amennyit csak lehetett, így némi tájékozódás után, egy kis gyakorlattal a billentyűzet kezelését a felhasználó nem érzi tehernek.

## BASIC

A Spectrum alapvető programozási lehetősége a BASIC rendszer. Ez, néhány specifikumtól eltekintve, nem különbözik sokban a szokásos BASIC rendszerektől, és egészében véve jól kiépítettnek tekinthető. Bőséges művelet- és függvényválasztéka jól használhatóvá teszi. Sajnos nincs benne IF-THEN-ELSE utasítás, és PRINT USING sincs. Ami azonban leginkább hiányzik, az a valós-egész megkülönböztetése. A felhasználó akkor is kénytelen az öt bájtot lefoglaló, sok műveleti időt felemésztő lebegőpontos számokat használni, ha kis értékű egész számokkal akar dolgozni, mint például tömbök indexelésénél.

A BASIC programok bevitelére és javítására szolgáló rendszer – noha nem tekinthető full screen editornak – jól használható és kényelmes.

## A képernyőkezelés

A képernyőt a gép két területre osztja. Középen helyezkedik el a munkaterület, és ezt körülvézi egy keret (BORDER). A keretnek csak a színét állíthatjuk be BASIC-ből, egyébként csak a magnetofonkezelő utasítások használják. A felhasználható munkaterület 24 sorban soronként 32 karaktert tartalmaz. Minden karaktert egy 8 × 8-as pontmátrixszal rajzol ki a gép.

Lehetőség van arra is, hogy saját grafikus jeleket tervezzünk és használjunk programból és klaviatúráról egyaránt. Így ékezetes betűk, görög betűk stb. állíthatók elő. A képernyő legalsó két sora fenntartott, ide a BASIC hibajelzései kerülnek, a szövegszerkesztő használja stb.

A fennmaradt 176 × 256 képpontból álló részt képpontonként szabadon kezelhetjük BASIC-ből a következő feltételek mellett: minden 8 × 8-as karakterhelyhez tartozik négy attribútum. Ezek egy bájton helyezkednek el, és azt adják meg, hogy milyen háttérszínen milyen színű képpontokat tartalmaz a karakterhely, kell-e a karaktert villogtatni és milyen a fényerősség. A háttér és a képpontok számára 8–8 szín alkalmazható (PAPER és INK), és két fényerősség van. Ez például azt jelenti, hogy egy karakterhelyen háromféle szín nem alkalmazható. Ezek a korlátozások bonyolult grafikához ügyes programozást kívánnak.

A BASIC segítséget nyújt egyenes szakaszokból és körívekből álló ábrák rajzolásához, de ábrák vagy ábrarészletek elmozdításához nem. Ez hatékonyan csak gépi kódú programmal oldható meg.

A karakteres ki- és bevitel (PRINT és INPUT) jól használható.

## A perifériák

A legfontosabb perifériákról már beszéltünk (tévékészülék, magnetofon). A Sinclair Microdrive bizonyos hirdetésekben „microfloppy”-ként szerepel, jóllehet működési elve végtelenített mágnesszalag-kazettán alapszik, amit igen miniatűr kivitelben valósítottak meg. Ez jól látszik átlagos elérési idejéből, ami 3,5 mp. Kapacitása kazettánként 85 kbájt. A ZX81-hez is csatlakoztatható ZX Printer sajátossága, hogy csak speciális (és meglehetősen drága) papírral használható. Nyomatási szélessége 32 karakter.

Ezekon kívül még sok gyártó perifériáit alkalmazhatjuk (elsősorban nyomtatókat). A hazai választék azonban elég szerény. Ha ilyesmit vásárolunk, győződjünk meg arról, hogy szükség van-e valamilyen interfészre.

Sajnos ezek a perifériák nem teljesen üzembiztosak. Ennek oka a gyártási nehézségekben rejlik. Mechanikus periféria biztonságos működtetéséhez bizonyos mérethatárok alatt rendkívüli problémákat kellett megoldani. Semmi kétségem sincs afelől, hogy a jövőben ezen a területen még nagy fejlődés várható.

A fentiekben kívül még sok periféria csatlakoztatható a Spectrumhoz, például botormány (joystick), hangosbeszélő (loud speaker) stb. Ezeknek komoly alkalmazásokhoz nincs közülük, inkább üzleti okokból készülnek.

Itt hívjuk fel a figyelmet arra, hogy kapható külön billentyűzet is, amely nagyobb és stabilabb, mint az alapgépen lévő. Aki tehát sokat nyúzza a gépét, például sok adat begépelését igénylő számításokat végeztet vele, nem árt, ha beszerez ilyet.

## Alkalmazási lehetőségek, szoftver

Mint már említettük, a Spectrum elsősorban játékprogramok futtatására készült. A szoftverjegyzékek is erről tanúskodnak. Az 1984. májusi Sinclair User például a következő választékot kínálja 48 kbájtos gépekhez: 70 darab kalandjáték, 84 Arcade-játék (gyors reflexeket kívánó játéktípus), 9 szimulációs játék, 28 stratégiai játék és 21 klasszikus játék (dáma, sakk stb.). Ezzel szemben 9 fordítóprogram van (minőségüket ne firtassuk!), üzleti, ügyviteli program pedig 27 darab. (Ezek nagy része is valószínűleg trivialisásokat old meg.) Az aránytalanság jól látható.

A fordítóprogramok közül a BETA BASIC-nek vannak jó paraméterei: IF-THEN-ELSE utasítást és WHILE ciklust tartalmaz, van ben-

ne PRINT USING, és még sok egyéb utasítással, függvénnyel bővíti a Sinclair BASIC-et. Az egész fordító kevesebb, mint 6 kbájt, mivel ki tudja használni a ROM-ban elhelyezkedő BASIC interpreter rutinjait.

Kétféle FORTH fordítót is készítettek, minőségükről nincs információnk.

A magazinok sokféle tanítóprogramot is hirdetnek; ezeket azonban nem érdemes számba venni, mert senkit sem fenyeget az a veszély, hogy Spectrumjától tanul meg franciául írni vagy gitározni. Inkább csak középiskolai szemléltetésre alkalmasak.

Ami a reális alkalmazási lehetőségeket illeti: ügyvitelgépesítésre a Spectrumot legfeljebb egy kiskereskedő alkalmazhatja (tulajdonképpen ez is valami); az adatfeldolgozási lehetőségeket erősen korlátozza a háttértárak kis kapacitása és a nem 100 százalékos megbízhatóság. Kisebb tudományos számításokra alkalmasnak mondhatjuk a gépet. Ilyen alkalmazások elsősorban a csillagászati számítások, mérnöki számítások, valamint néhány szűz, esetleg ezres minta alapján statisztikai próbák végzése, regressziószámítás (kiváló alkalmazási lehetőség!), és kisebb lineáris programozási feladatok megoldása.

Mindezekhez kaphatók különféle programcsomagok. 48 kbájtos géphez például 48 × 48-as mátrixoperációkat végző programcsomag, polinomok gyökhelyeit kereső program, numerikus integrálást végző program, többféle regressziószámítást és statisztikai próbákat végző programcsomag, lineáris programozási programcsomag. (Az utóbbi még 50 változós, 30 feltételes feladatot is megold. Ez persze csak bizonyos numerikus feltételek között lehet igaz, amit a hirdetések nem közölnek. De ha arra gondolunk, hogy egy átlagos étrendsámítási problémát a Spectrum e programcsomagjával már meg lehet oldani, akkor némi tiszteletet kezdünk érezni iránta.)

## Üzemeltetési Jótanácsok

Valamennyi mikrogép üzemeltethető szobai körülmények között. De ha hosszú élettartamot és üzembiztos működést kívánunk, semmi esetre se tegyük ki gépünket szélsőséges viszonyoknak. Nehéz pontos hőmérsékleti határokat vagy páratartalomra és egyéb körülményekre vonatkozó előírásokat adni, de figyeljünk a gépre, ne tartsuk olyan helyen, ahol napsugárzás, túlzott nedvesség vagy egyéb rongáló tényező károsíthatja.

A Spectrum nem tartalmaz beépített hűtőegységet, ezért kerülni kell a több órás használatot. A perifériákat ne tegyük egymás tetejére, mert szellőzésük nem biztosított!

Ha vigyázunk gépünkre, sokáig találjuk örömminket benne, sok érdekes dolgot tanulhatunk „kettesben”.

MÓCSI ZOLTÁN

## A ZX81

# Tényleg olyan rossz gép?

Nem! Határozottan állítom, hogy nem rossz! Az igaz, hogy nem túl megbízható, de az emberhez közel álló, kicsi, egyszerű. BASIC-je is jó, könnyen megérthető, megtanulható.

### Az alapgép

A ZX81-nek 16 k ROM-ja és 1 k RAM-ja van tárbővítés nélkül. A 16 k ROM 1-16513-ig címezhető és a PRINT PEEK utasítással leolvasható. A BASIC terület 16514-17538-ig terjed.

A gép felbontóképesége 63×42 pont. Egy képernyőn 22 sor és 33 oszlop jeleníthető meg.

### A perifériák

A géphez egy fekete-fehér háztartási televíziót csatlakoztathatunk. (Színeset nem érdemes, mert a gép csak a feketét és a fehéret ismeri.) A programokat magnókazettán tárolhatjuk, és ehhez kell két – EAR és MIC nevű – jack dugócsatlakozóval rendelkező magnó. Ezenkívül csatlakoztathatunk nyomtatót, pótmemóriát (utóbbit már egy magyar cég is gyárt, MEMOPACK néven). A géphez kapható kazettás gyorsmásoló, mely a programok kazettára való kivitelét vagy kazettáról való beolvasását gyorsítja.

Mint már említettem, a ZX81 egyszerű, kis teljesítményű gép, és valószínűleg ezzel tervezői is tisztában voltak. Azért elégedtek meg annyival, amennyit tud.

### Megbízhatóság

Ebben a kérdésben nagyon eltérőek a vélemények. Aki angol gyártmányú tárbővítést használ, az tudja, hogy a gépet emelgetni, tologatni, meglökni nem szabad, mert a program rögtön „elszáll”, azaz kimegy belőle. Ilyen esetben áramtalanítani kell a gépet, majd újra bekapcsolni, és lehet előlről kezdeni a program írását. Aki viszont a magyar gyártmányú tárbővítést ismeri, vagy aki tárbővítés nélkül használja a gépet, az már jobban meg lehet a ZX81-gyel elégedve.

A gép nagy hibája, hogy nagyon gyorsan felmelegszik, és lassan hűl. Ez azt jelenti, hogy másfél-két órai használat után legalább fél órát pihentetni kell.

Ezek alapján gépünket igazán nem mondhatjuk túlságosan megbízhatónak. Ha azonban jobban megismerjük, akkor egy kicsit „megpuhulunk” vele szemben, és bármennyire is megbízhatatlan, akárhogy is 1/2-es osztályzatot kapott az Ötlet vállalatjában a megbízhatóságra, mégis szívesen használjuk, játszunk vele.

### A billentyűzet

A gépnek tulajdonképpen nincs is billentyűzete, csak a tetejére „rárakoztatták” a gombokat, és az alá egy fóliaérintkezőt helyeztek. Ha hozzáérünk az egyik felfestett billentyűhöz, akkor

elméletileg a megnyomott gombon feltüntetett karaktert a gépnek el kellene raktároznia a memóriájában.

De ez nem olyan egyszerű, mint gondolnánk, mert biztonsági okból a fóliaérintkező és a billentyűzetnek kinevezett lap között maradt egy kis távolság. Hogy mi ez a biztonsági ok? Az, hogy ha megsérül a billentyűzet lapja, akkor legalább a fóliaérintkező maradjon épségben, mert azt kicserélni vagy javítani nem lehet! És mi ennek a következménye? Az, hogy nem elég a billentyűkhöz hozzáérni, hanem meg is kell őket nyomni, ha azt akarjuk, hogy a gép megértse, amit mondunk neki.

### A csatlakozók

A gépen öt csatlakozót találunk. Az egyik a tévével, másik kettő a magnóval, a negyedik a hálózattal köti össze a gépet. Ebből a négyből három jack dugó, a negyedik egy normál VHF csatlakozó. Arra mindenki rájöhét, hogy a tévével való kommunikáláshoz kell a VHF csatlakozó, a másik három pedig a magnóhoz és a hálózati csatlakozáshoz. Az ötödik csatlakozóba kell bedugni a memóriabővítést.

### A nyomtató

A nyomtató égetéssel változtatja el az eredetileg világosszürke, hőre és érintésre érzékeny papírt. A nyomtatót a gépen három utasítás tudja vezérelni:

- A COPY utasítás hatására a ZX81 kinyomtatja azt, ami a képernyőn van.
- A LLIST utasítás hatására kinyomtatja a gépen tárolt programlistát.
- Az LPRINT-et ugyanúgy lehet használni, mint a PRINT utasítást, csak nem a képernyőre, hanem a papírra nyomtatja ki az utána álló adatokat, szövegeket.

Több nyomtatót is csatlakoztathatunk egymás mögé. Ha a nyomtatás közben pótmemóriát is akarunk használni, akkor azt az utolsó nyomtató mögé csatlakoztassuk.

### Kazettás adattárolás, programtárolás

A ZX81 nem tud adatokat kazettán tárolni, de egy gépi kódú program segítségével ezt a problémát meg lehet oldani. Programokat a SAVE utasítással lehet elmenteni, kazettára kiírni. A kazettára kiírt programokat a LOAD utasítás segítségével olvashatjuk vissza.

Vigyázzunk, hogy a programbeolvasásnál csak az EAR csatlakozóba dugjuk be a drótot a magnón és a gépen is. Programok elmentésénél (SAVE) a MIC csatlakozóban legyen a drót a gépen és a magnóban is.

### A kazettás tárolás megbízhatósága

Ebben a címben is benne van a „megbízhatóság”, és ebben a tekintetben a ZX81 soha sem

állt túl magas szinten. De itt az embernek kell figyelmesnek lennie, mert ebben az esetben a gép nem hibázik. (Megfelelő legyen a hangszín, a hangerő, a drótok jó helyre legyenek bedugva, a LOAD vagy SAVE utasítás megfelelő legyen.)

Összefoglalva azt mondhatjuk, hogy a gép megbízható, csak az ember nagyon könnyen ejtethet hibát.

### Szerkesztés

A gép EDIT gombjával egy sort tudunk kiemelni javítás céljára. Hogy melyik sort írja ki a programból a gép javításra? Íme egy példa:

```
10 INPUT A
```

```
20 PRINT A
```

```
30 <GOTO 10
```

Itt a 30. sorban van „<” jel. Ez mutatja meg, hogy ezt a sort fogja a gép kiírni, ha megnyomjuk az EDIT gombot. Ezt a kis jelet a „↑” gombbal föl, a „↓” gombbal lefelé lehet mozgatni. Ha már kiemeltük a sort, akkor a „←” és a „→” gombbal tudunk a sorban törlés nélkül mozogni. Törölni a shift és a nulla lenyomásával lehet.

### A gép BASIC-je

A gép BASIC-je, mint már említettem, egyszerű, könnyen használható, de nagy hibája, hogy lassú. Érdekes a PRINT AT (az ABC-80-on PRINT CUR) utasítás. Ezzel a képernyőn bárhol írathatunk valamilyen karaktert, szöveget vagy változó értékét. Ezenkívül a gép nagy előnye, hogy a GOTO utasítás után akármilyen változót elfogad.

A gépen nincs DATA, és ezzel együtt READ és RESTORE utasítás, nincs ON, GOTO utasítás, DELETE automatikus újrásorszámozás és ELSE utasítás sem. A DATA utasítást helyettesíthetjük a LET X\$ = segítségével. Ez után karaktereket, számokat írhatunk be, és ezt beolvashatjuk a LET A (vagy A\$) = X\$ segítségével. Utána meg kell adnunk, hogy X\$ hányadik karakterétől hányadik karakteréig akarjuk a változó értékét beolvasni.

Az ON, GOTO utasítás helyett jó lehetőség a GOTO utasítás utáni változó használata. Ezt a változót a megfelelő sorszám értékére átírhatjuk, amikor kell.

A DELETE utasítás helyett semmi nincs, ezért kénytelenek leszünk minden sorszámot begépelni, és utána a NEWLINE gombot megnyomni. Az ELSE utasítást ugyancsak nem lehet helyettesíteni, ezért a program írása kicsit bonyolultabbá válik a THEN és az ELSE ág elkülöníthetlensége miatt.

Hát ennyit a ZX81-ről, amely minden ellenkező hírral ellentétben nagyon emberközeli, könnyen kezelhető, gyorsan tanulható, kicsi és méreteivel arányos tudásszinten álló gép.

FAZEKAS TAMÁS  
7. oszt. ált. isk. tanuló

# a COMPUT-80

univerzális mikroszámítógép családot hatékonysága, széles alkalmazási területe jellemzi.

Hatékonyságát korszerű architektúrájának, elektromos és mechanikus modularitásának köszönheti. Széles alkalmazási területét a nagy megbízhatóság, gazdag perifériaparkja, valamint kiterjedt alap és alkalmazási programrendszerei biztosítják.

## a COMPUT-80

mikroszámítógép a legkorszerűbb áramköri elemekből, korszerű technikai és technológiai megoldásokkal épül fel.

– E tulajdonságai méltán keltik fel a számítástechnika felhasználóinak érdeklődését.

## a COMPUT-80

mikroszámítógép modularitásából ered, hogy optimálisan alakítható ki a feladat megoldásához legjobban igazodó számítógép konfigurációs, akár több munkahelyes (felhasználói) változatban is. Ez a tény teszi lehetővé a COMPUT-80 mikroszámítógép univerzális felhasználását és cél (feladat) orientált mikroszámítógépes rendszerek létrehozását.

### a COMPUT-80

mikroszámítógépek különösen hatékonyan használhatók az ipar, mezőgazdaság, orvos-egészségügy, oktatás stb. területeken, az alábbi alkalmazásokban:  
– adatrögzítés, adatgyűjtés, adatbázis kezelés, adatfeldolgozás, szövegszerkesztés, szöveg feldolgozás, programfejlesztés, (képi) tervezés, oktatás, tájékoztatás, folyamatvezérlés, nyilvántartás, műszaki-tudományos számítások, ügyvitel, terminál, intelligens terminál stb.

Segítünk Önnek az igényéhez legjobban alkalmazkodó mikroszámítógépes konfiguráció kialakításában a kívánt feladat megfogalmazásában és az alkalmazói programjainak kidolgozásában, vételében.

Vállaljuk továbbá:

- kulcsrakész rendszerek szállítását
- oktatást
- meglévő rendszerének bővítését, továbbfejlesztését
- üzembe helyezést, karbantartást.

Az Ön részére szállított **COMPUT-80** mikroszámítógépes konfigurációhoz magyar nyelvű felhasználói dokumentációt és 1 éves garanciát biztosítunk.

Vállaljuk továbbá a garancia időn túli szervizt is!

### a COMPUT-80

mikroszámítógép megrendelhető: RAMOVILL Sz. Sz. V. Bp. Üllői út. 69. Tel.: 142-477, 343-125



1982EV 12.31=I ALLAPOT ALA

## Bérleménynyilvántartás mikroszámítógépen

TANACSI KEZD E6

165

2839

27

1983

100

1993

03 12 FELOTT

6

2

8

Belátom, ez a cím csak keveseket hoz lázba – következzen hát néhány adat, amely talán rávilágít a munka fontosságára, egyben utal a feladat nehézségeire, szakmai vonatkozásaira és – ha nem időszerűtlen manapság illyet emlegetni – érdekességére, szépségére is.

Ma Magyarországon mintegy 760 ezer lakás van állami tulajdonban; Budapest lakásállománya ebből körülbelül 430 ezer, amelyet 15 kerületi Ingatlankezelő Vállalat kezel. Ilyen tömegű (és értékében is figyelemre méltó) ingatlanállomány kezelése, fenntartása, a vele való gazdálkodás az ingatlanok adatainak teljes körű birtoklása nélkül elképzelhetetlen. Egyetlen lakásról csaknem 350 adatot kell nyilvántartani, a fő azonosítóktól kezdve a méret- és berendezési tárgy-adatokon keresztül, egészen a lakbért befolyásoló különböző tényezőkig (egyaránt kitérve a lakótelepi garzon és a kertés villa specifikumaira).

Az országos adatállomány természetesen nagyszámítógépen van: a Fővárosi Építőipari Üzemgazdasági és Ügyvitelszervezési Iroda (FÜTI) R-40-én. Az adatok gyakran változnak: például felépül egy lakótelep, lebontanak egy arra megérett blokkot, vagy éppen gázfűtést szerelnek egy eddig cserépkályhával fűtött lakásba. Ezek a változások a hagyományos rend szerint adatlapon kerülnek a bérleményeket kezelő kerületi Ingatlankezelő Vállalatok illetékes házkezelőségtől az adatrögzítőkhöz, onnan pedig a nagygépre.

Az egész lakásállományt érintő gazdálko-

dási döntésekhez kellő tájékoztatást adnak a nagygépes feldolgozások. A házkezelőség operatív tevékenységéhez – ügyfélszolgálat, tervszerű felújítások, karbantartások megszervezése és irányítása, váratlan hibák elhárítása – viszont már nem felel meg a hagyományos kartonos nyilvántartás.

Ezt a hagyományos nyilvántartást hivatottak felváltani a vidéki IKV központokban és a budapesti házkezelőségeken üzembe állított, illetve a jövőben üzembe álló mikroszámítógépek. A bérleménynyilvántartás céljára kihelyezett gépek kivétel nélkül hazai gyártmányú, 8-bites, floppy diszkes háttérrel ellátott berendezések. A többkötetes adatállományokat Pascal nyelven írt programok kezelik. Tekintettel arra, hogy a felhasználók zöme semmi számítástechnikai előképzettséggel nem rendelkezik, a programok részletes menükínálattal, gyakori gép-gépkezelői párbeszédrel látják el feladatukat.

Azzal, hogy gépesítik az adatok kezelését, a mikroszámítógépek sokrétű szolgáltatással segítik az ingatlankezelést: *ügyfélszolgálati* célokra közvetlenül hozzáférhetővé teszik a házkezelőségeken a bérlemények adatait. A reklamáló vagy csupán tájékozódni szán-

dékozó bérlő akár saját szemével meggyőződhet lakása képernyőn megjelenített adatainak helyességéről.

Elősegítik az önálló *házkezelőségi gazdálkodást* azzal, hogy pontosan és gyorsan számíthatóvá teszik a lakbéreket; nyilvántartják a berendezési tárgyakat, s ezzel támogatják a tervszerű cserék ütemezését, és kialakítják azokat az alapokat, amelyekre felépíthető az IKV szintű gazdálkodás számítógépes támogatása.

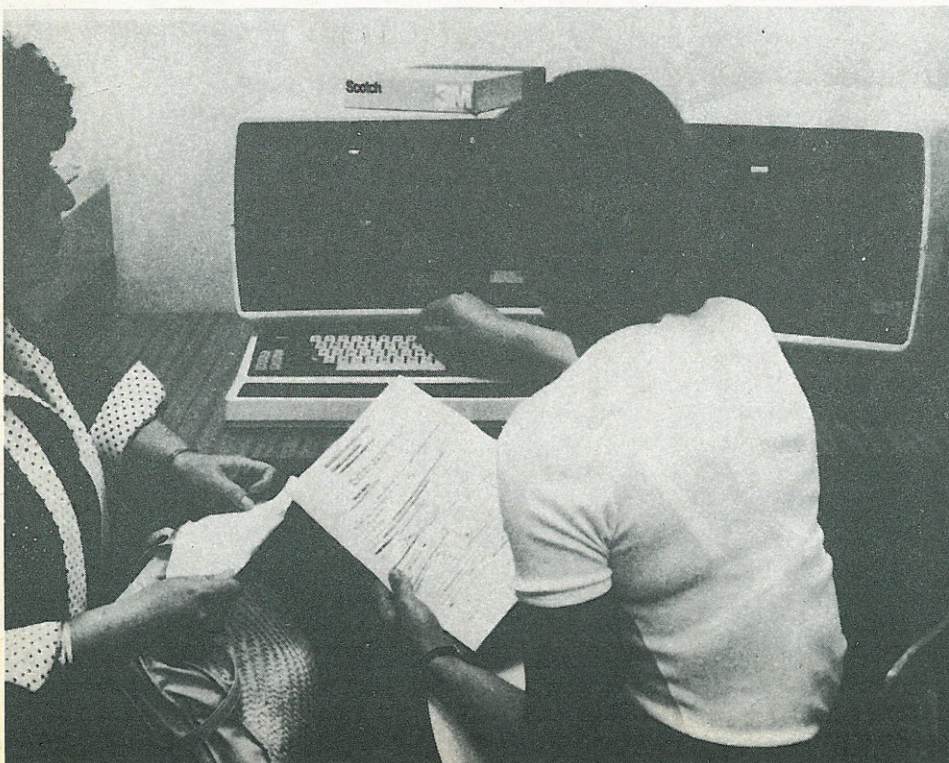
A házkezelőségi lakásállomány különböző szempontok szerinti feldolgozásával szükség szerint *információkat* szolgáltathatnak az IKV és a Fővárosi Tanács számára az üresen álló lakásokról, azok várható beköltözhetőségi időpontjáról, a társbérletekről, valamint a jogcím nélküli lakáshasználók által elfoglalt lakások jellemző adatairól.

A mikrogépek egyben korszerűsítik az *országos bérleménynyilvántartási* rendszer aktualizálásának folyamatát; az országos állomány ily módon nagyobb biztonsággal szolgáltat magasabb szintű döntések előkészítéséhez adatokat és összesített információkat.

A mikroszámítógépes programrendszer fejlesztése nemrég szolgáltatta az első eredményeket a legalapvetőbb állománykezelő és -feldolgozó programok formájában. Ezekre alapozva, most a speciálisabb felhasználói igények kielégítését szolgáló modulok készülnek. A felhasználók szintén mostanában barátkoznak a gépekkel és programokkal – a kezdeti idegenkedést, tartózkodást egyre több felhasználónál szakmai érdeklődés váltja fel. Figyelemre méltó, hogy a rendszerfejlesztés kezdeti időszakában tapasztalt tartózkodás ellenére a felhasználók rendkívül segítőkészek bizonyultak, jelentős segítséget nyújtottak a szervezői munkában.

Az eddigiekből bizonyára kiderült, hogy a munka neheze még hátravan: a rendszer igazi tesztjét a mindennapi használat fogja jelenteni, különösen azok a folyamatok, amelyekben a mikrogép közvetlenül lakossági információszolgáltatást fog végezni; másrészt belátható időn belül létre kell hozni a nagyszámítógép és a mikrogépek között az on-line kapcsolatot, hogy teljes mértékben kiaknázható legyen az az információtechnológiai kapacitás, amely a kihelyezett mikrogépek és a központi nagygép formájában – egymástól egyelőre fizikailag szétszórta – létrejön.

(X)



# MELYIK LÉSZ AZ ÉV GÉPÉ?

Pontosítunk: az év hazai gyártású mikroszámítógépe az, amire várjuk Olvasóink szavazatát a lapban található levelezőlapon. A választás megkönnyítésére közreadjuk az általunk ismert gépek legfontosabb adatait. A cél, hogy minél szélesebb körből nyerjük a felhasználók véleményét, megismerjük

tapasztalataikon alapuló értékítéleteiket.

Néhány szót az értékelésről: az Olvasók véleményén kívül a gyártónál is végzünk kicsit részletesebb felmérést, majd erőnk és segédesszökeink függvényében benchmark programokkal próbálunk objektív összehasonlítást végezni. A felmérésekből leszűrhető

következtetéseket közreadjuk, mely hasznos lehet egy jövőbeli mikroszámítógép-vásárló vagy -alkalmazó számára. Az eredményeket áprilisban közöljük, a szavazóknak előre köszönjük, hogy részt vesznek akciónkban.

AZ ÓTLET

ÉS A MIKROSZÁMÍTÓGÉP MAGAZIN SZERKESZTŐSÉGE

TÍPUS	GYÁRTÓ DES	MIKROPROC. TÍPUS	BT	MEMORIA	OPERACIOS REND.	PROGRAMNYELVEK	EV	AR
71777	EMG	AM2901 + 3X8053	16	128/92	SAJAT/GRAFIKUS	B	1982	800 000
72222	UMDS	Z8018090	8	64/..	SAJAT/CP/M	B/F/P/PLZ	1983	1 040 000
AGRINFO 100	LIGNIFER	U 880 (Z80)	8	64	AGRINFO (CP/M KOMP.)	A/B/F	1984	500 000
AIRCAMP-16	BOSCOOP	U 880 (Z80)	8	16/8	MONITOR	A/B	1982	27 000
AIRCAMP-64	BOSCOOP - PERSONAL GT	U 880 (Z80)	8	64/16	POST-DOS (CP/M KOMP.)	A/B	1984	250 000
AX-II	ALK. I.F.J. EGY.	ROCKWELL 6502	8	48-256/..	APPLE-II KOMP. CP/M A/B/C/F/P		1984	105 000
COMPUT-80	COMPROJECT GMK	8090 V. Z80	8	64-512	CPS-DOSY (CP/M)	B/F-IV/C/P	1982	200 000
CTX 30	COMPUTEXT	Z80	8	64/8	MSYS (CP/M)	A/B/F/F	1984	350 000
EMG-777	EMG	8085	8	16-144	GRAFIKUS	A/B	1981	875 000
FLOPPYLINE E	VILATI	F 8	8	4-64	SAJAT	A/P	1981	..
FLOPPYMAT E	VILATI	F 8	8	4-64	SAJAT	A/P	1981	..
FLOPPYMAT I	VILATI	F 8	8	4-64	SAJAT	A/P	1981	..
FLOPPYMAT SP	VILATI	F 8	8	4-64	SAJAT	A/P	1981	..
GD-80	MTA SZTAKI	TORB	16	16-256	GRAFIKUS	B/F	1981	..
HT 680X	HTSZ	SZM 601 (M6800)	8	64-512/8-384	EDITOR-ASSEMBLER	1981	..	350 000
HT 680X CDP	HTSZ	SZM 601 (M6800)	8	64-512/8-384	GRAFIKUS	B/F	1981	..
HT-10802/64	HTSZ	U 880 (Z80)	8	16/16	MONITOR	A/B	1983	58 000
IPT 002	TRITON GMK	Z80	8	16-64/8	MONITOR	A/B	1984	19 900
JONATHAN	NJSZT-HCC	ROCKWELL 6502	8	48	APPLE-II KOMP.	B/P	1982	..
KOMAT	MEV	..	8	64	ARANKOR-ELL.	..	1982	..
LABSYS 80	LABOR MIM	8090 V. Z80	8	64-256	MSYS (CP/M KOMP.)	B/C/F	1983	300 000
M80X/SZM 1625	SZKI	U 880 (Z80)	8	64/6-12	CP/M KOMP.	B/F/P/C-NYELV	1982	300 000
MFB	MTA SZTAKI	U 880 (Z80)	8	64	12K MODULARIS	..	1981	..
MSP-80	COMPUTEXT	8080	8	8-64/8	SYMON (CP/M-SZERU) P	..	1981	..
MICKEY 80	LSI ATSZ	U 880 (Z80)	8	16-32/8-16	MONITOR	A/B	1982	34 000
MICROCONTROL	CONTROL EL.T.	U 880 (Z80)	8	64/..	MC-DOS (CP/M); ISIS A/B; F/P; FORTH	..	1982	130 000

Model	Year	Processor	Memory	Storage	OS	Language	Price
MICROTEST 2	1981	8080	8	..	..	ARANKOR-ELL.	1981 ..
MIKROP 60 SDS	1983	8080	8	3	..	MONITOR	1983 40 000
MINIMOD	1984	U 880 (280)	8	32-54/32	..	MSYS (CP/M KOMP.) A/B	1984 410 000
MOD 81	1982	280 V. 8080	8	64-512	..	CP/M ES MF/M B/P/CLSP	1982 250 000
MOD 81M	1982	280 V. 8080	8	64-512	..	CP/M ES MF/M A/P/B/CLSP	1982 330
MOTRA	1982	SZM 601 (M6800)	8	64	..	..	1982 ..
N68	1981	SZM 601 (M6800)	8	128/10	..	2 K MONITOR	1981 ..
NEZ 215	1984	..	8	64/8	..	A/B/PLC	1984 ..
OBC 64	1983	SZM 601 (M6800)	8	48	..	MONITOR	1983 215 000
OBS 6802	1980	SZM 601 (M6800)	8	12-24/32	..	FEJLESZTO	1980 ..
OL 622/1	1980	8080	8	4/4	..	FEJLESZTO	1980 ..
ORDAG	1982	NINCS	12	32	..	OS/I	1982 4 000 000
PRIMO	1984	U 880 (280)	8	16-48/16	..	MONITOR	1984 15-24 000
PROCOM 16	1984	..	16	1024-4096	..	RSX-11	1984 ..
PROFESSOR	1981	M68000	16	256-16384/16-32	..	UNIX:ORISIS/CP/M-68K A/B/P/C	1981 200 000
PROLOCOM	1983	8085	8	64/24	..	FOLYAMATSZAB. TRANSIT-85	1983 800 000
PROPER16/ES1830	1984	8085	16	1024/48	..	PROPOS-16 B/P/P	1984 900 000
PROPER16W/E1833	1983	8088	16	832/46	..	PROPOS-16 B/P/P	1983 600 000
PROPER8/ES21800	1983	U 880 (280)	8	16-256/2-32	..	PROPOS (CP/M KOMP.) B/P/P	1983 20 000
PTA 4000/KA 150	1982	SHARP	8	4-16/3.5	..	MONITOR	1982 180 000
ROSY-80	1980	U 880 (280)	8	64	..	RODOS (CP/M KOMP.) A/B/F/P/PLM	1980 900 000
RPT-80	1981	8080	8	64	..	RPS (UPM - CP/M KOMP.) A/B/PLM	1981 ..
SAM-85 MIKRO	1981	8085	8	..	..	..	1981 ..
SAM-85 MIKRO-L	1984	8085	8	..	..	..	1984 450 000
SAM-85 MIKRO-P	1981	8085	8	..	..	..	1981 ..
SAMIS	1984	8085	8	64	..	FIDOS (CP/M KOMP.) A/B/P/PLM	1984 65 000
SIMON-68	1982	SZM 601 (M6800)	8	16-64/1-56	..	FLEX/CP/M KOMP. ..	1982 75 000
SLK-80	1982	U 880 (280)	8	32	..	MSYS (CP/M KOMP.) B/C/F	1982 163 000
SLK-80A	1983	280	8	64-128/12-16	..	MSYS (CP/M KOMP.) A/B/C/F	1983 130 000
SYSTEM	1982	280A	16	1024	..	NETTY (CP/M KOMP.) A/B/C/F/P	1982 10 MILLIO
SZM 52/10	1981	NINCS	8	40/24	..	RSX:ETSS:MMT2 A/B/C/F/P	1981 236 000
TAP-34	1984	8080	8	..	..	SAJAT	1984 400 000
TM 16	1984	MS8000	16	256-1024/..	..	TRSDOS:MSYS/CP/M-68K A/B/F/P	1984 1 400 000
TPA 11/440	1982	AM 2900 + 280	16	512-4096	..	FOBOS: DOS-RV A/B/C/F/P/ADA	1982 3 700 000
TPA 1148	1982	NINCS + 280	16	512-4096	..	FOBOS: DOS-RV A/B/C/F/P/ADA	1982 2 500 000
TPA JANUS	1983	K1501VM1: 280	16	64-255	..	FOROS: DOS-RV A/B/C/F/P/DIBOL	1983 640 000
TPA QUADRO	1983	AM 2900+ 3X 280	12	128 KSZO	..	OS/H:RTS/H/DOS/H B/F/FOKAL/OPAL	1983 650 000
TPA-L/123H	1982	AM 2900	12	128 KSZO	..	OS/H:RTS/H/DOS/H B/F/FOKAL/OPAL	1982 700 000
TR-80	1980	8085	8	64/8	..	IRDS (ISIS):FDOS B/F/P/PLM	1980 450 000
TRANSMIC 16	1984	M68000	16	256-1024/..	..	CP/M-68K	1984 800 000
TRNSMIC 8	1983	280	8	32-512/16-64	..	TRSDOS:MEMDOS:CP/M A/B/F/P	1983 400 000
TRIS	1984	8085	8	64	..	TRIS:IFDS:CP/M A/B	1984 500 000
TRSZ-80	1982	U 880 (280)	8	..	..	..	1982 ..
TV COMPUTER	1984	U 880 (280)	8	32-64/8	..	MONITOR	1984 16 000
TZ 80	1981	Z80	8	32-512/..	..	TRS-DOS:NEWDOS:CP/M A/B/F/P	1981 400 000
UNIFROG	1982	TMS 3900	16	64-200/32-64	..	SAJAT	1982 550 000
VARYTER	1982	280A	8	64-256/2-24	..	NETTY (CP/M KOMP.) A/B/F/P	1982 250 000
VIDEOPLEX 3	1980	NINCS	16	68-132	..	CSOP. ADATROGZ. ..	1980 6 000 000
VOLAN	1984	Z80	8	64	..	ADATROGZITESI ..	1984 ..
WPC	1983	U 880 (280)	8	16-56/8	..	UP/M (CP/M KOMP.) A/B/F/P	1983 200 000
WPC	1982	U 880 (280)	8	64/16	..	CP/M	1982 250 000
WT 16	1984	8088 + 280	16	256	..	CP/M: CP/M-66	1984 599 000
WT 20	1981	8080	8	64/8	..	SAJAT	1981 1.5-2 M
WT 30	1982	8085	8	16-96	..	CADC	1982 2-4 M
WT 52	1984	M58000	16	512-2048	..	SCS (UNIX)	1984 ..

Megjegyzés: BT = bitek száma; memória = RAM min-max (ROM min-max Kbájtban); A = Assembler, B = Basic, C = COBOL, F = FORTRAN, P = PASCAL; árak forintban.

Amikor kicsik intézik a nagyok dolgát

## Személyi számítógép a közgazdasági munkában

Több mint fél éve érdekes kísérlet kezdődött egy szervezési intézet közgazdasági osztályán. Célja a személyi számítógép (esetünkben egy Sinclair-Spectrum) közgazdasági munkákban történő felhasználhatóságának bizonyítása. A kísérlet eredményessége annyira egyértelművé vált, hogy a készített programok – egy negyedéves próbafeldolgozás után – 1984-től „élesben” üzemelnek, azaz ezeken a területeken a kézi kigyűjtéseket, számításokat megszüntették.

### Árkalkuláció

A KALKULAL program az intézet módszertani kiadványai árkalkulációját készíti el. Elvégzi az aikalmazható árak összehasonlítását, elősegíti az árak tervezését, és 11 kalkulációs tétel szerinti modellezést tesz lehetővé.

### Naturális utókalkuláció

Naturális utókalkuláción értjük az intézet vállalkozásaihoz kapcsolódó, többféle – természetes mértékegységű – ráfordítások (géporák, szellemi órák stb.) rögzítését, költségelosztási célokra történő összesítését, a vállalkozások gazdasági értékelését (fedezet, számlázottság stb.). Ezeket a feladatokat eddig kézi gyűjtésekkel, a nyilvántartások folyamatos vezetésével végeztük.

A NUTRIA nevű naturális utókalkulációs nyilvántartási rendszer az önálló költségügyi, főosztályi adatbázisok kezelésére, naprakészen tartására épül. Egy főosztályi adatállomány terjedelme: 98 munkaszám, egy munkaszámra 19 jellemző. A NUTRIA rendszer programjai között egyaránt található adatállományt generáló és módosító program, rendszeres hozzátöltést biztosító program, továbbá a bármikor lekérdezést irányító program. A főosztályi adatállományokat sorosan feldolgozva, intézeti összesen adatokat kaphatunk.

A rendszer elkészítését megkönnyítette a Spectrum gépnek az a programozási lehetősége, hogy az adattömböket a programoktól külön (de programozottan) lehet kimenteni és beolvasni, tehát tárolni is.

HEGNEVEZES	FT
1. SZELL. RAF. SZUK.	138443
2. SZELL. KULONKSG.	1441
3. NYOMDA. SZUK. KSG.	1426
4. SZUK. KSG. OSSZ.	1477
5. INT. ALT. KSG.	199
6. TELJES SAJ. KSG.	199
7. IDEGEN ALVALL.	199
8. RAFORD. OSSZ. *	200
9. NYERESÉG	3359
10. NETTO BEVETEL.	46
11. ERTEK. KULONKSG.	46
12. NYOMDA. RAYAGSG.	4032
13. BRUTTO BEV. **	273511
14. TETELSZAM ***	300
15. EGYSEGAR ****	1367

Ezt a rendszert 1984. január 1-től üzemeltetik. Fő előnyei, hogy zárási időszakokban időt takarítunk meg; folytonos lekérdezési lehetőség áll rendelkezésre az összes vállalkozás helyzetére; a változó értékelési igényeknek megfelelően, nagyon rövid idő alatt készíthetők a tipizált adattömböket különböző szempontok szerint lekérdező és értékelő programok; az osztály munkája szervezettebbé válik, és az egész rendszer „osztályon belül” van.

### Nyereségelosztás

A nyereségelosztás a tárgyidőszak végén – mivel addigra a befolyásoló tényezőkre vonatkozó döntések megtörténtek, és végrehajtásuk vonzatai is ismertek – mechanikus folyamat. A tárgyidőszak elején vagy év közben azonban egy-egy jelentős beruházási vagy bérkifizetési döntés előtt a nyereség-szükséglet érdekeltségi alapok oldaláról történő vizsgálata – a vállalati jövedelemszabályozás mai rendszerében – egyre fontosabb tervezési feladat. Erre a célra készítettük el az 1984. évre vonatkozó általános vállalati jövedelemszabályozási rendszer programozott modelljét. A NYIR (nyereségelosztást irányító optimalizáló program-) rendszer segítségével a felhasználó szakember modellezheti: a bruttó részesedési alap alakulását különböző létszám, bérfejlesztési, nettó részesedési alap kombinációkban; a nyereségági fejlesztési alapforrás alakulását a felhasználási célok kombinációiban; az adózatlan nyereség alakulását az érdekeltségi alapok különböző kombinációiban; az osztatlan érdekeltségi alap alakulását a befolyásoló tényezők különböző kombinációiban.

Az ismertetett eredményváltozókat egyenként 3-3, független változóként kezelt befolyásoló tényező és 5-7, futásonként megadható

JEL	HEGNEVEZES	ERTEK
0	NT. RESZ. ALAP (LEFT)	1388
1	OSSZLET SZAM (FO)	356
2	BERSZAM. FEJL. (%)	1
3	BERSZAM. TIP. [1-4]	1
4	JÖVEDEL. INT. (%)	1
5	BEDOLG. BER (LEFT)	33700
6	BAZ. LETSZAM (FO)	33700
7	BAZ. BER (LEFT) (FO) (EV)	65000
8	BERPREF. (LEFT)	0

paraméter által generált döntési térben lehet a NYIR rendszer segítségével modellezni. A befolyásoló tényezők értelmezési tartományát a felhasználó adja meg.

Kiürítható az 1000 eredményértéket tartalmazó döntési tér (10-10-10 lépés a független változóknak), továbbá grafikus megjeleníthető az eredményváltozó és egy befolyásoló változó 20 értékére vonatkozó összefüggés.

A bruttó részesedési alapot számító bérszabályozási rutin mind a négy alapformában (teljesítménytől függő, központi A, központi B, központiba saját kérésre átsoroltak) működik, és ezzel a NYIR rendszer alkalmazható minden, az általános vállalati jövedelemszabályozás alá tartozó szervezetnél.

A NYIR rendszer az általunk készített programok közül a legjobban felhasználó orientált, egymástól független rutinok halmazát tartalmazza, melyet a felhasználó saját céljainak megfelelően fűz össze. A rendszert felhasználtuk az 1984. évi gazdálkodás tervezésénél. Alkalmazási területén a korábbi egydimenziós tervezést megszüntette, mivel segítségével kifejezhető az intézet releváns döntési terve.

### Tapasztalatok

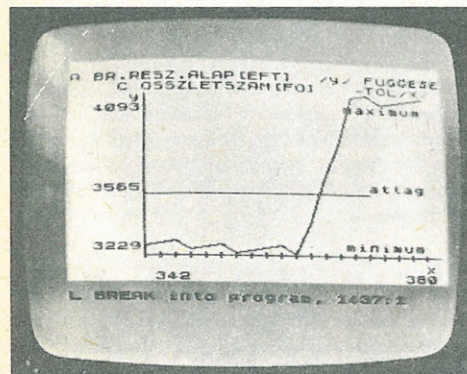
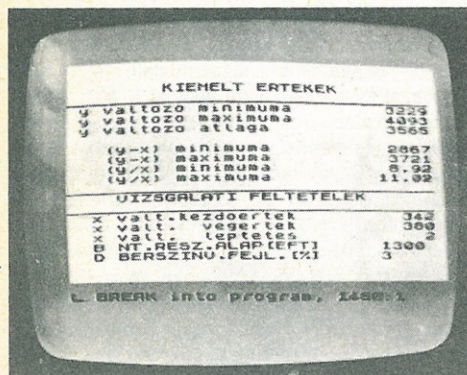
Önmagában a személyi számítógép csak lehetőség a személyre orientált számítástechnikai munka megteremtésére. Ennek tartalmát a rendelkezésre álló szoftver minősége lényegesen befolyásolja. Jó példaként a NYIR rendszer hozható fel. A számítógép és a szoftver a gazdasági szakember munkaeszközévé, „szerszámává” vált, aki a napi céljainak megfelelően használja azokat. Ez a munkamódszer azonban feltehetően, hogy a felhasználó egymaga látja el a rendszerszervezői, programozói, operátori, archiválási, dokumentálási feladatokat – bár ezek

elvégzéséhez elegendő egy általános számítástechnikai ismeretszint, amely ma már mindenképp a társadalmi kultúra szerves részét képezi.

A perszonalis jelleghez bizonyos ártértékelések is tartoznak. Nem lesz fontos, hogy a felhasználó által megírt program a legrövidebb, leggyorsabb legyen; elegendő az is, hogy az adott cél elérését a felhasználó rendelkezésére álló idő alatt biztosítsa.

Nem biztos, hogy az a periféria lesz a legkedvezőbb a felhasználónak, amelyet utasításokkal nyitni, címezni, zárni kell, viszont az átviteli idő a legrövidebb. Számunkra teljesen kielégítő volt a Spectrumhoz kapcsolt egyszerű kazettás magnetofon is, amelyre a NUTRIA rendszer adatállományait egy-két perc alatt lehetett kimenteni egy nagyon egyszerű utasítással.

A személyi számítógép használatának gazdaságossága az előbbieket miatt véleményünk szerint elsősorban nem a géptől, hanem a felhasználó embertől függ. Ez a gazdaságosság nem közvetlenül jelentkezik, hanem az egész gazdál-



kodási folyamat szerveztségének javulásával fizetődik ki.

A személyi számítógépek a gazdálkodó szervezet bármely szintjén használhatók: a döntéskészítést megalapozottabbá, a nyilvántartásokat és működést programozottabbá teszik, s ennek pozitív hatása révén az egész szervezet dinamizálhatják.

Ezért kezdtünk hozzá a személyi számítógépek felhasználásához intézetünk közgazdasági osztályán.

DR. BÓDIS BÉLA

## Miből lesz a cserebogár?

„Mi lesz ebből a gyerekből?” – kérdezik elszörnyedve a szülők, ha valami turpisságon érik fiukat vagy lányukat. Burt Sloane szülei azonban csak elcsodálkoztak, amikor fiuk telefonzsinórt vezetett a szobájába. Pedig a kérdés jogos lett volna – a fiú egyike volt azoknak a „számítógépbetyároknak”, akik éjnek idején egy személyi számítógép vagy bérelt terminál és némi fantázia segítségével, az USA országos távközlési hálózatán keresztül „betörnek” idegen nagyszámítógépek adatairaiba.

13 éves korában mutatta meg neki valaki, hogyan lehet saját számlát nyitni a MIT gépén. Majd tanfolyamokra járt, további ismereteket szerzett. Legnagyobb bravúrnak azt az esetet tekintti, amikor egyszer leállt az operációs rendszer, miközben ő éppen vonalban volt. Ez a biztos lebukást je-

lentette volna, ha nem sikerül újraindítania a rendszert. Később nem tudott ellenállni a kísértésnek, hogy egy kissé átírja a rendszert, mert úgy érezte, hogy a felhasználók túlságosan kevés erőforráshoz jutnak a rendszer „urához” képest. Ennek a vége természetesen az lett, hogy leleplezték.

Nem sokkal ezután szoftveresként elhelyezkedett a Silicon Valley cégnél. Ma már nem ér rá betyárkodni. „Ehhez gyerekek kell lenni, akinek rengeteg ideje van. És mostanában egyre nagyobb a biztonság a gépekben.” – mondta a Newsweek riporterének. 20 éves. Napi 8 órát dolgozik, elalvás előtt pedig számítógépes játékok készítésén töri a fejét. Saját cégét tervezi, amely – természetesen – szoftvervédelemmel fog foglalkozni.

– ab –

## ADOK – VESZÉK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100.- Ft, magánszemélyeknek az első sor 50.- Ft, minden további sor 20.- Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

• A VILATI értesíti felhasználóit, hogy 1984. július 1-től a Floppymat SP mikrogepevel együtt szállított alapszoftverkészletét a VT-20, TAP-34 kompatibilis Ügyviteli PASCAL nyelv futtató rendszerrel egészíti ki, a berendezés árának változatlanul hagyása mellett.

A PASCAL fordító rendszert a SZÁMREND (1125 Bp. XII., Szarvas Gábor u. 58-60., T.: 165-883) és a DIANA GMK. (Bp. I., Attila u. 33., T.: 368-842) forgalmazza.

A fordító rendszerhez PASCAL nyelvű editor, file utility programok, abstract file-kezelő, decimális I/O programcsomag, matematikai függvények programcsomag, néhány egyéb programcsomag, valamint komplett dokumentáció is tartozik.

• CSERÉRE FELKÍNÁLOM a következő alkatrészeket:

INTEL 8080A 1 db  
8212 1 db  
8228 (kiforrasztott) 1 db  
8255A (kiforrasztott) 1 db

TMS 2708 4 db  
MC 8502 1 db  
µA 723 (DIL tok) 2 db  
µA 723 (L tok) 2 db  
Szükségem lenne:  
Z80-A (MK3880-4) 1 db  
2716-35 4 db  
4116-3 max. 8 db  
8 MHz kvarc 1 db

Cím: Molnár Máttyás, Bp., Victor Hugo u. 25-27. 1132. T.: 338-305/406 mellék (8-15 óráig).

• FELAJÁNLUK MEGVÉTELRE, illetve további értékesítésre 1 db COMMODORE 8250 LD duál FLOPPY számítógép-egységet. Vételára: 350 000.- Ft. Pest Megyei Víz- és Csatornamű Vállalat, 2040 Budaörs, Komáromi u. 16. Telefon: 852-322, Ügyintéző: Balázs Gáborné



**MEOR**MEOR  
MEOR**MIBOR**MIBOR  
MIBOR**FAHIR**FAHIR  
FAHIR**HÁTIR**HÁTIR  
HÁTIR**MODELL' P**MODELL' P  
MODELL' PMIKROSTART  
MIKROSTART  
MIKROSTARTALLOC' 64  
ALLOC' 64  
ALLOC' 64**KERKOMP**  
KERKOMP  
KERKOMP**KERKOMP**  
KERKOMP  
KERKOMP**KERKOMP**  
KERKOMP  
KERKOMP

A címben szereplő nevek, rövidítések nem egy újabb (XX. századi) nyelvújítás kitalálmányai, hanem a KERSZI mikrosoftver termékeinek fantázianevei.

A Kereskedelmi Szervezési Intézet, mint a belkereskedelem bázisintézete – annak ellenére, hogy a tevékenységének nagyobb volumene nagyszámítógépes szervezés és feldolgozás körébe esik –, megalakulása óta folyamatosan végzi és fejleszti a kis- és középgépes, 1982 óta pedig a mikroprocesszoros adatfeldolgozó gépekre alapozott szervezési tevékenységét.

A megnövekedett gazdálkodási követelmények következtében egyre sürgetőbb az igény a vállalati belső információrendszer hatékonyabb működése, közelebről az információ „forgási idejének” felgyorsítása iránt. Ehhez járul még a kötelező statisztikai és beszámoló rendszer előírásainak naprakész és pontos teljesítése. A többoldalú követelményeknek megfelelő szakemberutánpótlás hiánya, és eleve az ügyviteli dolgozók csökkenő létszáma miatt szinte már lehetetlen jól megfelelni.

A fent vázolt ellentmondást a gépi adatfeldolgozás igyekszik feloldani. A nagyszámítógépes feldolgozás nem minden vállalatnál tudott meghonosodni (főleg kiskereskedelemben), mivel az adatok mennyisége ezt nem kényszeríti ki, és a hazai technikai lehetőségek által biztosított kötegel (batch) típusú feldolgozás viszonylag lassú, és nem érvényesül kellőképpen az operativitás. Ezekből az alapvető tényekből ered, hogy a kereskedelem széles területén inkább a saját üzemeltetésű középgépes és mikroszámítógépes technika terjed el.

A számítástechnikai fejlesztésnek ma – különösen a belkereskedelemben – a fejlesztési eszközök nagymérvű csökkenése a legnagyobb gátja. A valóban irodai adatfeldolgozásra gyártott mikrogépek hazánkban indokolatlanul drágák.

A KERSZI mikrosoftverjeinek több mint a fele ezért a SINCLAIR Spectrum személyi számítógépre készült, amely gép a beszerzési árának nagyságrendje miatt már minden vállalat számára elérhető.

**MEOR** munkaügyi nyilvántartó rendszer, ami munkaügyi, személyzeti és egyéb speciális szervezési feladatok (pl. telefonkönyv-készítés) személyi számítógépes megoldása.

**MIBOR** munkaidőbeosztást optimalizáló rendszer, mely jól alkalmazható a személyzet munkaidőbe-

osztása különböző variánsainak kidolgozására, a munkaidő pontos elszámolására, statisztikák és elemzések készítésére.

**A FAHIR** relációs nyilvántartó rendszer megoldja a határidős feladatok, levelezés, szerződések, rendelések, részhatáridők különböző szempontok szerinti ütemezését, végrehajtását és pontos követését és ellenőrzését.

**A HÁTIR** hálós tervezési-irányítási rendszer alkalmazásával egy beruházás, üzembe helyezés, át-szervezés, reklamáció, bármilyen nagyszabású rendezvény sikeresebb és gyorsabb. A rendszer lehetőséget nyújt és elősegíti a háló logikai tervezését, idő (Gantt-) diagramok készítését.

**A MODELL'P** mikroszámítógépes rendszer gyors segítséget nyújt a vállalati gazdálkodás rövid, közép- és hosszútávú tervezéséhez, prognosztizáláshoz, optimális vállalati magatartás kidolgozásához.

A KERSZI softverfejlesztésének jelenlegi fő technikai bázisa a COMMODORE 64 személyi számítógép, melynek beszerezhetősége vagy bérelhetősége egyre szélesebb körű és a vállalatok részére elérhető nagyságrendű. Mivel a gépre szervezett folyamatok nem tipikusan ügyviteli és ezáltal szakmára orientált feladatok, az alkalmazási terület kiterjed a belkereskedelem valamennyi vállalatára, sőt más népgazdasági ágazatára is.

A vállalatok szívesen alkalmaznak szakmai szinten egységes rendszert, mely azonos géptípusra készült, bevezetése gyors, egyszerű és olcsó. Ennek jegyében készítette el a COMMODORE 64 személyi számítógépre a

**MIKROSTART** programcsomagot, mely magában foglal egy munkaügyi nyilvántartó és információs rendszert, állóeszköz-nyilvántartás és -elszámolási rendszert, jövedelemérdekeltségű egységek nyilvántartását, bolti rentabilitási adatgyűjtő rendszert, energiatarifáló és -értékelő programot.

**ALLOC '64** állóeszköz-nyilvántartási rendszer lényege az értékcsökkenés automatikus elszámolása és az állomány különböző vetületű lekérdezésének, egyes egyedi és gyűjtött információk gyors lekérdezésének új lehetősége, melyeket a manuális nyilvántartás csak nehezen vagy egyáltalán nem tud szolgáltatni.

A rendszerben megoldott feladatok: egyedi analitikus nyilvántartás, bruttó és nettó érték vezetése,

értékcsökkenés elszámolása, leírt állóeszközök listája és az állomány automatikus átvezetése, leltárkiértékelés egységenként, állóeszközök kimutatása egységenként, típusonként, lejárató idő és elhasználódási fok, valamint a főkönyvi számlák szerint, állóeszköz készletváltozások átvezetése, főkönyvi feladás automatikus elkészítése.

A KERSZI mikroszámítógépes rendszereinek harmadik technikai bázisa a robotron A 5120 és 5130-as mikrogépek. Erre készült el a KERKOMP 83, amely olyan általános rendszermodell, melyet minden hazai kiskereskedelmi vállalat a bolti elszámoltatáson alapuló számviteli és információs folyamatok gépi feldolgozására alkalmazni tud. A hazai kiskereskedelem általános problémája – és a felhasználói igényeket ez alapvetően orientálja – hogy a kiskereskedelem alapfolyamatai igen nagy manuális adatfeldolgozási igénnyel jelentkeznek a számviteli és elszámoltatási munkában, és ez a hagyományos módszerekkel szinte már megoldhatatlan. Az új üzemeltetési formák mint a jövedelemérdekeltségű és szerződéses üzemeltetés, az eddig viszonylag egységes adatgyűjtési – és feldolgozási, valamint az információs és adatszolgáltatási folyamatokat szétbontották, az értékelés és adatszolgáltatás bonyolultabbá vált 1983. jan. 1-től.

A KERKOMP alapvető célkitűzése, hogy egyszerű adatbevitellel valamennyi összefüggésbe hozható folyamat ismételt adatrögzítés nélkül feldolgozható legyen, ezáltal növelje az adatfeldolgozás biztonságát és sebességét és csökkentse a manuális kiszolgáló munkát.

A KERKOMP kiskereskedelmi áruforgalmi adatfeldolgozási és információs rendszer alapfunkciói: bolti bizonylatok rögzítése, boltkönyvelés, bolti áruforgalmi jelentés készítése, napi és havi főkönyvi feladások, vállalati és bolti cikkszoportos beszerzési információk lekérdező rendszere, leltárelszámoltatás, cikkszoportos mérlegkör, bolti árrés elemzése és kimutatása, bolti készletvezetés, készletelemzés, forgási sebesség kimutatása, szállítói beszerzés és árrés kimutatása, cikkszoportos teljesítésfigyelési rendszer, szállítói számlák automatikus gépi likvidációja, belső szállítások automatikus gépi likvidációja, szerződéses boltok gépi nyilvántartása, automatikus pénzforgalmi határidőfigyelési rendszer, jövedelemérdekeltségű egységek speciális adatfeldolgozási és készletvezetési modulja, árrés-tömeg figyelése, automatikus EL-ÁBÉ- és készletszámítás, főkönyvi feladás.

# KERSZI MIKRO- SZOFTVEREK

## SPECTRUM 48-ra:

### MEOR

- MUNKAÜGYI NYILVÁNTARTÓ RENDSZER

### MIBOR

- MUNKAIDŐBEOSZTÁST OPTIMALIZÁLÓ RENDSZER

### FAHIR

- RELÁCIÓS NYILVÁNTARTÓ RENDSZER

### HÁTIR

- HÁLÓS TERVEZÉSI-IRÁNYÍTÁSI RENDSZER

### MODELL'P

- VÁLLALATI TERVEZÉSI-PROGNOSZTIZÁLÁSI MODELL

## COMMODORE 64-re:

### ALLOC-64

- ÁLLÓESZKÖZ NYILVÁNTARTÁSI RENDSZER

### MIKROSTART

- JÖVEDELEMÉRDEKELTSÉGŰ EGYSÉGEK NYILVÁNTARTÁSA

- BOLTI RENTABILITÁSI ADATGYŰJTŐ RENDSZER

- ENERGIA TARIFÁLO ÉS ÉRTÉKELŐ PROGRAM - STB.

## ROBOTRON A 5120 és 5130

## MIKROSZÁMÍTÓGÉPEKRE:

### KERKOMP - KISKERESKEDELMI

### ÁRUFORGALMI ADATFELDOLGOZÁSI

### ÉS INFORMÁCIÓS RENDSZER



BUDAPEST XIII., DÓZSA GYÖRGY ÚT 150.



**PRODUKTORG**

Szervezési  
Vállalat

feladatát tűzte ki,  
hogy a vállalatokat,  
kisebb termelőegységeket  
„testükre szabott”  
felhasználói programokkal  
ellássa.

Programjaink a vállalati műszaki, gazdasági és nyilvántartási ügyvitel egészét felölelik, segítik a vezetést a gyors informálódásban, a dolgozókat a napi munkájukban.

Programjainkat a legkorszerűbb kis- és mikroszámítógépeken lehet futtatni - Commodore 64, M08X, Proper 8, TAP 34 stb. -, ezáltal a szükséges befektetési költség nem okoz komolyabb megterhelést a vállalatoknak.

Egyes esetekben lehetőséget biztosítunk a programjaink használatához szükséges számítástechnikai elemek kölcsönzésére, bérletére is.

Az alábbi programok állnak rendelkezésre, illetve kidolgozás alatt vannak:

- számviteli programok: analitikus könyvelés, főkönyv, nyersmérleg,
- készletnyilvántartás: rendelés-nyilvántartás, raktárgazdálkodás, értékesítés, számlázás,
- munkaügy: személyi nyilvántartások, bérelszámolás,
- termelésnyilvántartást támogató programcsomagok,
- műszaki programok: optimum számítás, műszaki adatnyilvántartás.

A fentiek a legáltalánosabban használt programcsomagok, de ezen kívül rugalmasan alkalmazkodunk a folyamatosan változó hazai szoftver- és hardver igényekhez és a vállalati egyedi kívánásokhoz.

A programok kezelésére, használatára a vállalat dolgozóit betanítjuk, és kb. negyedéves intenzív próbafeldolgozást végzünk a vállalat telephelyén a vállalat dolgozóival együtt. Munkánkért felelősséget vállalunk.

Fő irányelvünk a gyorsaság és a pontosság, ezért a gyors vállalati eredményt garantáljuk.

Jelszavunk: kis gépekkel is megvalósíthatók nagy értékű programok és komoly vállalati eredmények.

Bővebb tájékoztatásra keressen fel bennünket.

Címünk:  
PRODUKTORG Szervezési Vállalat  
1251. Budapest, II., Fő u. 68.  
Számítástechnikai Fejlesztési és  
Kiszámítógép-Alkalmazási Iroda  
Telefon: 350-169, 154-090/408, 323, 178 mellék.

## Építsünk számítógépet! IV.

Ismételten felhívjuk az olvasók figyelmét arra, hogy további információkat a szerzőtől kaphatnak.

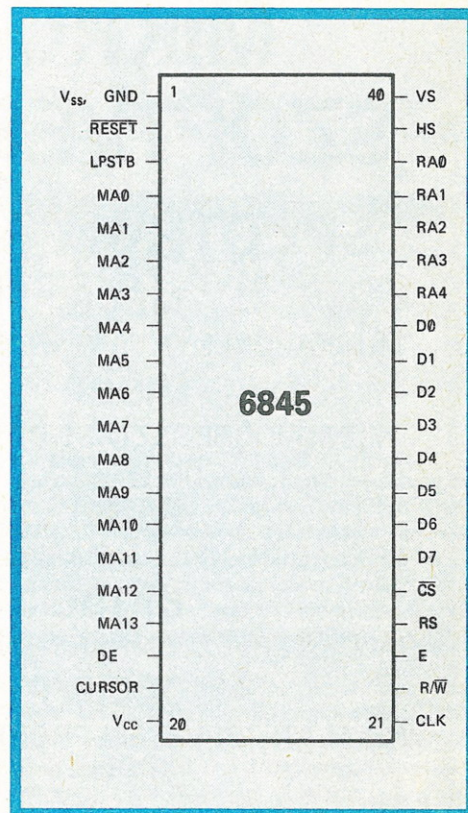
Ebben a részben először az alkatrészjegyzéket, majd a képernyőhajtó integrált áramkört ismertetjük.

Az 1. táblázatban megadott alkatrészek nagy része – legfeljebb némi utánjárással – kiskereskedelmi forgalomban kapható, néhány azonban nem. Ez utóbbiakból a szerző rendelkezik néhány darabbal, amit az első építőknek hajlandó önköltségi áron átadni. A felsorolt alkat-

1. táblázat

Típus	Darab	Gyártó	Megnevezés
6803,6303	1	Motorola, Hitachi	mikroszámítógép/mikroprocesszor
2764	1-2	számos cég	8 kbájt EPROM
74LS373	1	számos cég	nyolcas latch-típusú tároló
74LS20	1	számos cég	2 × 4, ÉS-NEM
74LS155	1	számos cég	kapuzott, 2 dekódoló/demultiplexer, 2-ről 4-re
74LS367	5	számos cég	hatos, négy bites puffer, three-state
7407	2	számos cég	hatos, jelkövető
LM339	1	Nat. Semi.	feszültség-összehasonlító
74LS139	3	számos cég	két, dekódoló/demultiplexer 2-ről 4-re
8T26,6880	6	Motorola, Bulgária	4, buszhajtó, three-state
7805	2	számos cég	+5 V, 1 A feszültségszabályozó
6845,6505	1	számos cég	képernyőhajtó
46845			
2716 (2732)	1	számos cég	2 (4) kbájt EPROM
2114	8-14	számos cég	1 k × 4 bit RAM
6116	1-8	számos cég	2 k × 8 bit CMOS RAM
74LS157	3	számos cég	kapuzott, négy multiplexer 2-ről 1-re
74157	2	számos cég	kapuzott, négy multiplexer 2-ről 1-re
74166	1	számos cég	8 bites léptetőregiszter párhuzamos/soros
7475	2	számos cég	kapuzott, négy, latch-típusú tároló
7486	3	számos cég	4 × 2, kizáró VAGY
7400	2	számos cég	4 × 2, ÉS-NEM
74LS04	3	számos cég	hat, NEM
74LS163	1	számos cég	4 bites számláló, törölhető
7420	1	számos cég	2 × 4, ÉS-NEM
74LS20	1	számos cég	2 × 4, ÉS-NEM
74LS74	1	számos cég	két, tároló
74LS21	1	számos cég	2 × 4, ÉS
74S32	1	számos cég	4 × 2, VAGY
7485	4	számos cég	4 bites összehasonlító
74LS138	1	számos cég	dekódoló/demultiplexer, 3-ről 8-ra
74LS138	2	számos cég	dekódoló/demultiplexer, 3-ről 8-ra
LM323K	1	számos cég	+5V, 5A feszültségszabályozó
7402	1	számos cég	4 × 2, VAGY-NEM
5 MHz	1	számos cég	kvarckristály
15.323 MHz	1	számos cég	kvarckristály
0.1 mfd	25	számos cég	tárcsa kondenzátor

Szükségesek még: ellenállások, csatlakozók stb. és egy 80 VA 7-8 V tápegység



1. ábra

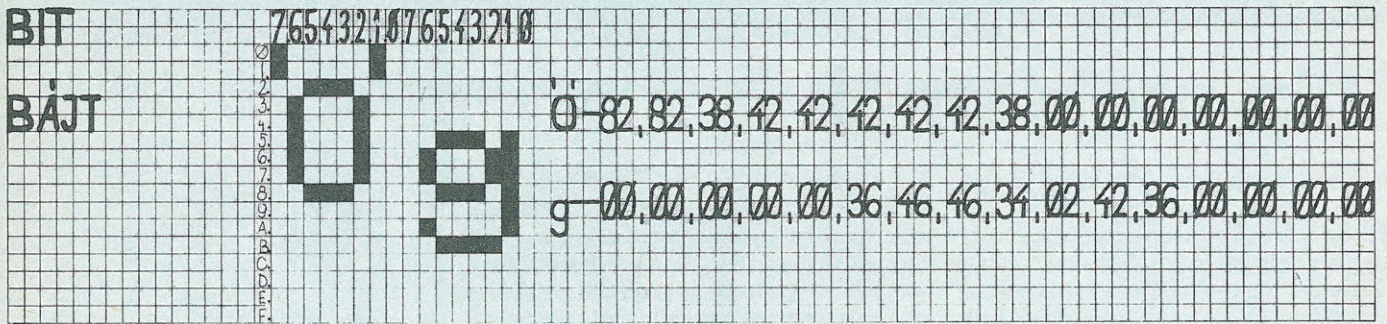
részek közül némelyik a minimális konfigurációnál elhagyható, a nagy tárcapacitású változatnál azonban továbbiakat is használni kell.

A 6845 igen teljesítőképes képernyőhajtó. Alkalmos:

- fekete/fehér és színes képernyő kezelésére,
- különböző szintű intelligenciájú terminálok kiszolgálására,
- alfanumerikus, félgrafikus, grafikus megjelenítésre,
- alfanumerikus üzemmódban az összes ismert formátum (például 64 × 16, 80 × 24, 72 × 64, 132 × 20) megvalósítására,
- csak +5 V tápfeszültséget igényel,
- közvetlen TTL-szintű be- és kimenetek kezelésére,
- oldal, sor, karakterek hardveres mozgatására,
- a fénypont formátumának, a fényerőnek a szabályozására,
- fényceruza közvetlen csatlakoztatására,
- a képernyőmemória a 6845 és a mikroprocesszor között multiplexelhető, és így ez külső (DMA) készülék által is feltölthető, illetve a mikroprocesszor által is átírható,
- a karakterek normál és fél sor készletteléses módban való kijelzésére,
- a képernyőmemória 16 kbájt méretű kiterjesztésére,
- a karakterek maximális befoglaló mátrix méretének 32 sorig történő kiterjesztésére,
- két belső regiszterének összekapcsolásával 512 kbájt méretig terjesztett grafika létrehozására.

A felsoroltokról részletesebb információk találhatóak a 6845 adatlapján. Az igen széles körű variálhatóságot 19 programozható regiszter vezérlésével valósíthatjuk meg. A programozás a rendszer adatbuszán keresztül történhet.





2. ábra

2. táblázat

Regiszternév	Regiszter-tartalom
Horizontális végigfutás	98
Horizontális kijelzés	64
Horizontális szinkronozás	78
Szinkronozási szélesség	5
Vertikális össz sor	27
Vertikális karaktorsor	5
Vertikális kijelzés	16
Vertikális szinkronozás	20
Összenyomott, ill. kettőzött karakter	0
Max. kijelzett sor karakterenként	11
Karakterkijelzés kezdő sora	0
Karakterkijelzés végső sora	0
Fénypontkezdő cím felső bájt	0
Fénypontkezdő cím alsó bájt	0
Képernyőkezdő cím felső bájt	0
Képernyőkezdő cím alsó bájt	0
Fényceruza-regiszter	0

Az 1. ábrán látható a 6845 lábkiosztása. A csatlakozások funkciók szempontjából öt csoportba oszthatók: kapcsolat a processzorral, képernyővezérlés, memória/karakter-generátor címzés, újraindítás, egyéb.

A processzor a nyolcbites adatbuszon keresztül cserél információkat a 6845-tel (D0-7, 33.-26. láb). A 6845 kiválasztása a CS (25. láb) és a RS (24. láb) segítségével történik. Az információcsere órajelét az E (23. láb), irányát a R/W (22. láb) szabályozza.

A képernyővezérlés a képernyő-szinkronozás VS, HS (40., 39. láb) és a kijelzés-engedélyezés DE (18. láb) jeleiből áll.

A használt tár a 14 memória- (MA0-13, 4.-17. láb) és 5 oszlop (RA0-4, 38.-34. láb) címvezetékekkel választódik ki.

Az újraindítás a RESET (2. láb) vezetékre adott alacsony jelszintű jellel történik.

Az egyéb csoportba a CURSOR (19. láb), az órajel (CLK, 21. láb), a fényceruza-bemenet (LPSTB, 3. láb), a tápfeszültségek (V<sub>CC</sub>, V<sub>SS</sub>, 20., 1. láb) tartoznak.

A 6845 vezérlésére szolgáló belső regiszterek leírása a következő.

● Címregiszter: a többi belső regiszternek a processzor által történő megcímzésére szolgál.

● 10 időzítést vezérlő regiszter: horizontális végigfutás-, horizontális kijelzés-, horizontális szinkronozás-, szinkronozási szélesség-, vertikális össz sor-, vertikális karaktorsor-, vertikális kijelzés-, vertikális szinkronozás-, összenyomott, ill. kettőzött karaktermód-, maximális kijelzett sor karakterenként regiszter.

● 4 fénypontvezérlő regiszter: a karakterkijelzés kezdő és végső sorát kijelölő regiszterek, a fénypontnak a memórián belüli címét kijelölő regiszterek.

● Képernyő-kezdő cím regiszter.

● Fényceruza-regiszter.

A 2. táblázatban ismertetjük az egyes regiszterek tartalmát gépünkben. A 2. ábrán bemutatjuk egy karakter megvalósítását gépünk kijelzőmódjában. Ez a 8 × 12 pontból 7 × 9 mód alkalmas az összes magyar betű megfelelően elhelyezett és megfelelő méretű megjelenítésére, vagy más karakterkészlet néhány napos munkával történő megtervezésére és előállítására is.

DR. SIMONYI ENDRE

## A μKlub: mozgalom

Mint már hírt adtunk róla, a televízióban a közeljövőben BASIC tanfolyam indul, melynek végén a SZÁMALK vizsgabizottsága előtt bárki programozói vizsgát tehet. A tanuláshoz szükséges számítógépes háttér megteremtésére az NJSZT μKlubokat szervez.

Ennek érdekében felhívással fordultunk mindazokhoz az intézményekhez, sőt magánszemélyekhez is, amelyeknek (akiknek) számítógépük van, hogy csatlakozzanak a μKlub mozgalomhoz, és megszabott feltételek mellett bárkinek díjtalanul engedélyezzék, hogy termináljukon vagy személyi számítógépükön programozhasson.

A felhívásra eddig az alábbi iskolák és más intézmények jelentkeztek:

- Karikás Frigyes Gimnázium, Szakközépiskola és Kollégium, Fonyód
- Kaposvári Mezőgazdasági Főiskola
- Számítógépes Rendszereket Értékesítő Közös Vállalat, Budapest
- Nehézipari Műszaki Egyetem, Gyártástechnológia tanszék, Miskolc

- József Attila Városi, Megyei Művelődési Központ, Salgótarján
- Építésgazdasági és Szervezési Intézet, Budapest
- Garay János Gimnázium, Szekszárd
- Szolnok megyei Művelődési és Ifjúsági Központ, Szolnok

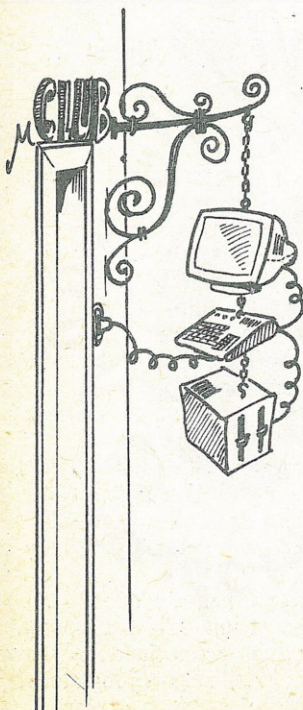
Idézet az egyik levélből: „Már egyre többen – bár még nem elegen – értik meg, hogy végérvényes lemaradásunk elkerülésére az egyetlen lehetőség a mikroelektronika, a számítógépes kultúra társadalmasítása – különös tekintettel az ifjúságra.”

Reméljük, hogy sokan osztják ezt a véleményt, és minél többen segítenek e nagy feladat megvalósításában.

A jelentkezés még nem zárult le. Köszönjük az eddig felajánlott támogatást, és várjuk újabb μKlubok csatlakozását a mozgalomhoz.

PÁRIS GYÖRGY  
a Tudományszervezési és  
Informatikai Intézet  
igazgatója

KOVÁCS GYŐZŐ  
a Neumann János  
Számítógéptudományi Társaság  
főtitkára



## COMMODORE 64

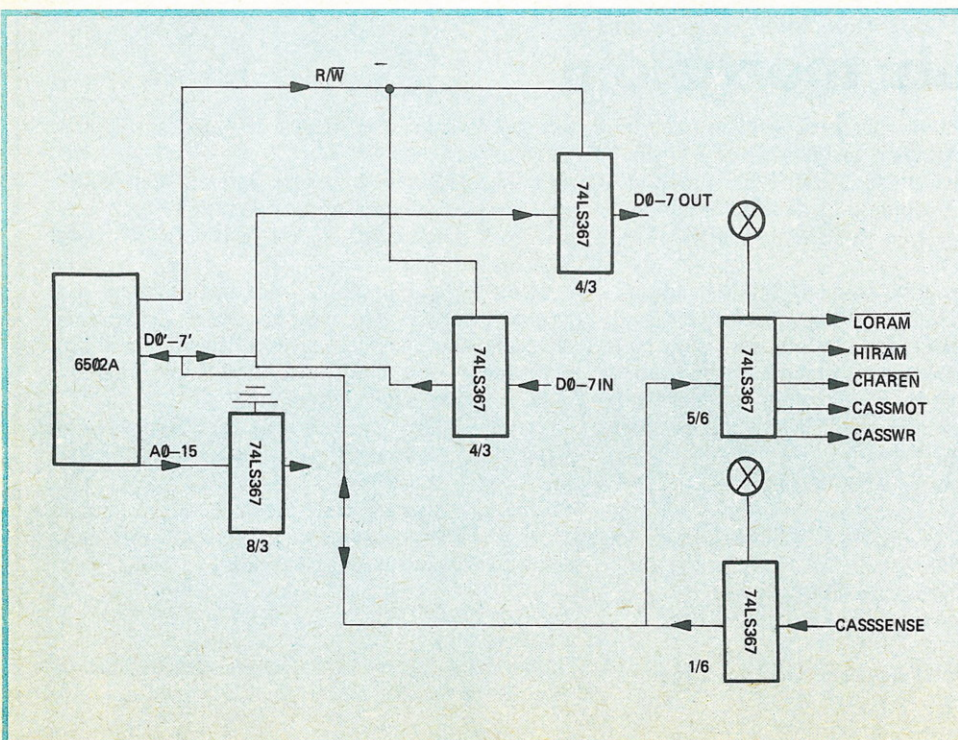
# Chip-problémák

Lapunk eddig minden számában megjelent egy cikk a HCC Commodore Szekciójától. Ezekben a különböző Commodore-gépek használatát megkönnyítő megoldásokat mutattunk be. Ezt a hardver-szoftver ötleteket ismertető sorozatot nyitottá kívánjuk tenni az olvasók számára is. Várjuk tehát ötleteiket!

A C-64 négy nagy integráltságú speciális áramkör, a 6510, 6526, 6566, 6581 köré épült. Ezek egyike sem kapható kiskereskedelmi forgalomban (Nyugaton sem). Az első kettő kényes alkatrész, erősen érzékeny a buszterhelésre. Sokszor elég a számítógéprendszer egyes egységeinek nem megfelelő sorrendű bekapcsolása, vagy egy, az átlagnál valamivel gyengébb tulajdonságú alkatrész, és vége.

Mi az oka ennek a legtöbb mikroszámítógépnél hiányzó nagy érzékenységnek? Az a sajátságos megoldás, hogy ezek az alkatrészek, a mikroprocesszor és a perifériacsatoló IC, a busz felől jövő túlterhelés elleni védelemre szolgáló ún. puffert is tartalmazzák. Mivel az ilyen túlterhelések gyakorlatilag elkerülhetetlenek, ezek a pufferek (amik viszonylag olcsók) aránylag gyakran mennek tönkre. (Ez az oka annak, hogy olcsóságuk és csekély lábszámuk – például a 74LS367 ára kb. fél USA dollár, lábszáma 16 – ellenére foglalatba szokták szerelni őket, és olyan helyre építik, ahol könnyű hozzájuk férni.) A C-64-nél a puffer tönkremenetele azonban nem valamelyik olcsó alkatrész cseréjét eredményezi, hanem a legdrágább (és ahogy már írtam), nem kapható alkatrészekét.

1. ábra



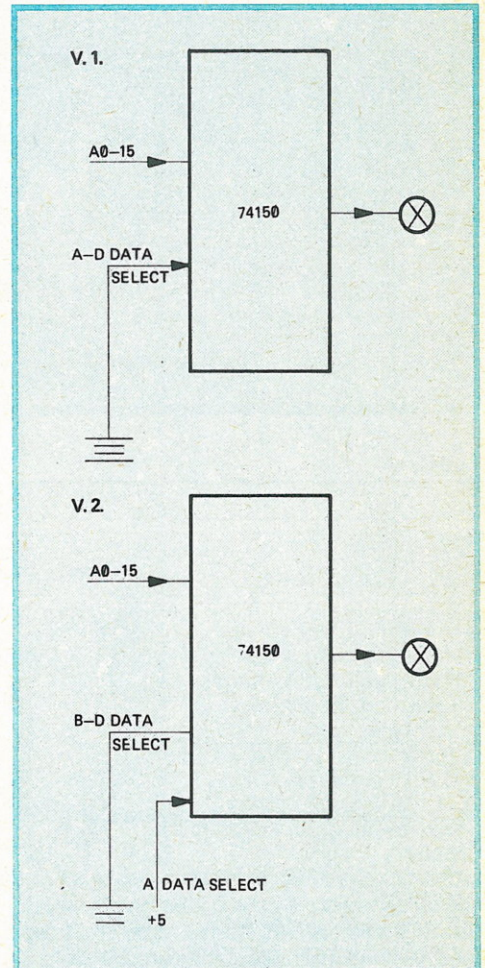
A Commodore cég ezen alkatrészek hibája esetén a szervizben cserél. Ezzel a lehetőséggel azonban a hazai tulajdonosok általában nem élhetnek. Az országban levő gépek egy részére van belföldi garancia, és ezeknél valószínűleg nincs alkatrészprobléma. Mit tehet azonban az, akinek a gépére nincs (vagy már nincs) garancia? Olyan híreket hallottam, hogy van valaki, aki hajlandó 8000 (nyolcezer!) forintért mikroprocesszort cserélni.

Azoknak, akiknek nincs erre ennyi pénzük, ajánljuk a következő megoldást. A 6510 típusú mikroprocesszor ebben a gépben helyettesíthető az 1. ábrán látható kapcsolással. A 6502A típusú mikroprocesszor Nyugaton kiskereskedelmi forgalomban (az NSZK-ban az ára kb. 22 márka), a 74LS367 és a 74150 pedig itthon is kapható.

A kapcsolás (2. ábra, a \* jelű résznél kapcsolódik az 1. ábrához) két változatot tartalmaz (V.1. és V.2.). Attól függően érvényes valamelyik, hogy a REFERENCE GUIDE 402. vagy 411. oldalán szereplő adat a helyes az adatrányregiszter (DDR) címét illetően (ugyanis az egyik szerint ez a  $0000_{16}$ , a másik szerint  $0001_{16}$ ). A nem jelölt vezeték a két mikroprocesszornál funkcionálisan azonosak.

A Klub, kellő számú érdeklődő esetén, vállalja a kártya megtervezését és elkészítését. Ugyancsak vállaljuk az USER PORT és a CASSETTE csatlakozók készítését, kábelcsatlakoztatását is.

DR. SIMONYI ENDRE



2. ábra

Kabalapályázatunkra érkezett



6. Somfai Géza, Budapest

# Mátrixnyomtató-illesztés

## Homelab II és Sinclair-Spectrum típusú mikrogépekhez

Ismeretes a mikrogépek inséges printerhelyezete, illetve ennek Sinclair-féle megoldási kísérlete: olcsó mechanikus szerkezet, drága papír, és lehetőleg mindent a szoftver csináljon. Az igazán jó nyomtató persze drága, és alig van lehetőség ezen a területen a házi fejlesztésre, kísérletezésre. A printerprobléma megoldásának egy lehetséges útját végigjártuk; a következőkben ezt ismertetjük.

Viszonylag kedvező áron (kb. 1000 osztrák schillingért) sikerült egy soronként 21 karaktert nyomtató kis mátrixnyomtató mechanikát beszerezni. A mechanika a következő vezérlési lehetőségeket nyújtja, illetve igényli:

- 7 darab nyomtató mágnes
  - motor ki- és bekapcsolása
  - papír soremelés – mágnes behúzásával vezérelhető
  - a nyomtatófej „home” pozíciójának jelzése
  - timing signal – a motor fordulatszámának jelzése
  - mechanikus festékszalag-továbbítás
  - tápfeszültség: 12 V, kb. 3,5 A.
- Normál, 57 mm széles papírszalagra nyomtat.

A gyártó cég a mechanikához vezérlőkártyát is szállít, amelyet azonban nem vettünk meg, csak a DPC 1 jelzésű, a mechanika kiszolgálására előkészített, programozott vezérlő chipet. Ez tulajdonképpen az Intel 8041 egychipes mikrogépének programozott változata, maszk-programozott karaktergenerátorral. Az általunk használt változatban egy ASCII szabványú karakterkészlet és egy japán írásjeleket tartalmazó karakterkészlet található (japán másodgyártmány volt!). Az I.8041-nek létezik EPROM-os változata is: I.8741, igény szerint változtatható karakterkészlettel.

A vezérlő elektronika lényegében ennek a chipnek a kiszolgáló áramköreit, továbbá a mágnesmeghajtó és motorkapcsoló áramköröket foglalja magában. Egy európa-kártya méretű kísérleti panelre valamennyi alkatrész felfér: egy 12 V-ról 5 V stabil tápot előállító táprész, a reset logika, a timing signal formáló áramkörre, a motorkapcsoló teljesítménytranszisztorok és a mágneseket kapcsoló, meghajtó és végfokot tartalmazó kapcsoló áramkörök. Az eredeti dokumentáció szerint itt japán gyártmányú

darlingtonokat ajánlatos alkalmazni, mi azonban beértük BC 301–303 és BD 239–240 kombinációval.

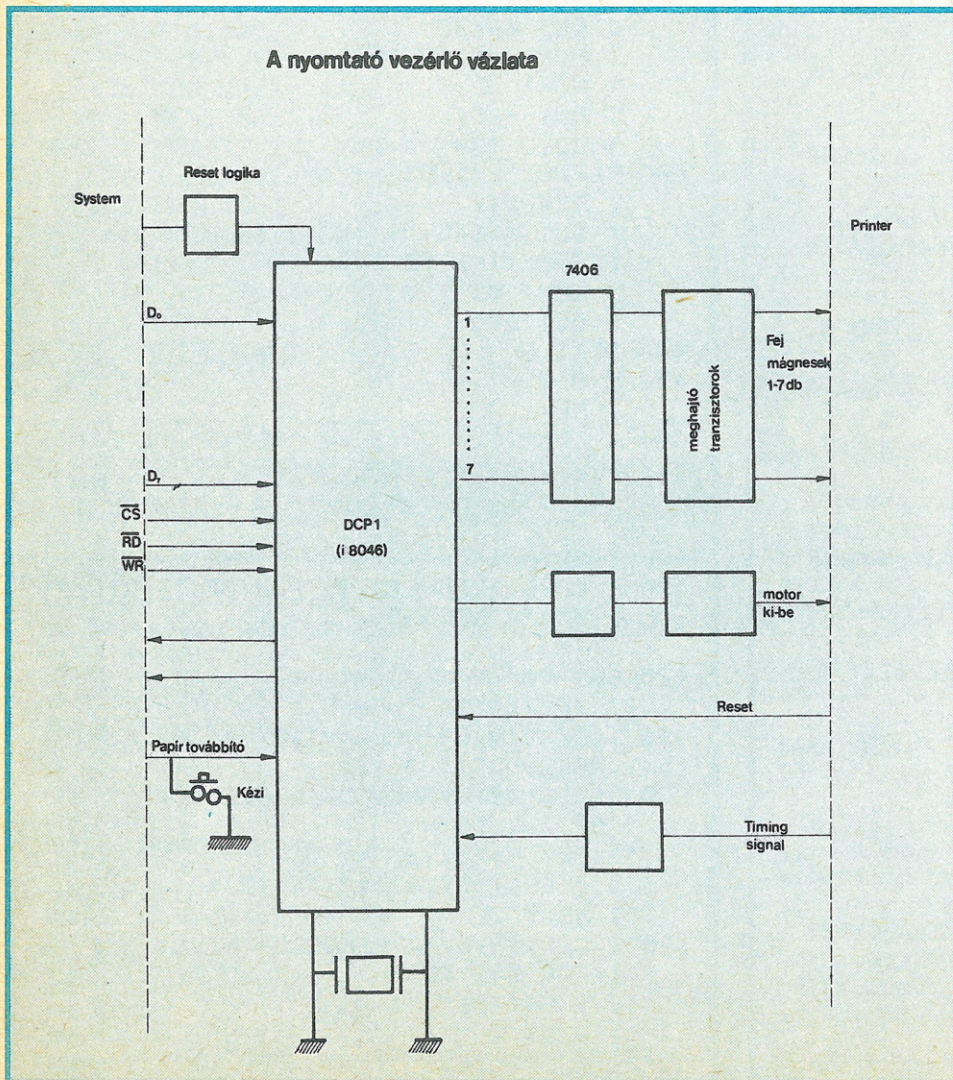
A DPC 1 vezérlő 6 MHz órajelről jár – a cég eleve nagy sebességre tervezte ezt a „szolgamikrogépet”. A host processzorral, a tényleges vezérlő géppel az adatbuszon keresztül kommunikál, a vezérlés a CS, RD, WR vonalak használatával valósul meg. Az adatbuszon keresztül a DPC 1 pufferebbe be kell írni a kinyomtatandó sor 21 karakterét, és egy nyomtató karakter elküldésével (A<sub>5H</sub>) ez kinyomtatódik.

A nyomtatás idejére, amíg a vezérlő a nyomtatással foglalkozik, az adatbuszon megjelenik egy státusz bájttal. Ennek kiértékelésével a fő processzor ellenőrizheti a nyomtató működését. Az itt megjelenő információ hardverváltozatban is rendelkezésre áll, a vezérlő Busy és Error lábain. Lehetőség van kézi papírtovábbításra (soremelés), de megfelelő vezérlő küldésével ez szoftverből is elérhető. Bekapcsoláskor három sor öntesztet végez a rendszer.

Az összeállított vezérlőkártyát és a mechanikát először Homelab II gépen próbáltuk ki, rövid gépi kódú vezérlő rutinnal, majd Sinclair-Spectrumhoz illesztettük. Itt külön problémát jelentett a CS jel képzéséhez használt címvezetékek csatlakoztatása. Mint ismeretes, a Sinclair-gépek sajátja a címbuszra kötött 1 kohmos felhúzó ellenállások. Erre a buszra közvetlenül nem lehetett csatlakozni a dekódoló logikával, az igazi megoldás buszmeghajtó lett volna. Átmenetileg kísérleti úton meghatározott, 8–10 kohmos ellenállással oldottuk meg a problémát. A vezérlő szoftver BASIC-ben készült, programok, adattömbök nyomtatására.

TÖLGYESI JÁNOS

A nyomtató vezérlő vázlata



### Kabalapályázatunkra érkezett



7. Havasi Miklós, Miskolc

Kabalapályázatunkra egyre több rajz, ötlet, javaslat érkezik. Ezért – miközben a legjobbak közlését folytatjuk – a határidőt 1985. május 31-ig meghosszabbítjuk. Kabaláikat, totemfiguráikat a szerkesztőség címére (1027 Bp. Fő u. 68.) továbbra is várjuk.

## Még mindig a játékprogram- pályázatról

Tőkei Gyula „Jégcsapolvadás” című programja nyerte el a negyedik helyet. A gép ismét ABC-80. Az első hat győztes program közül négy készült ABC-80-ra – igen szép teljesítmény az ABC-80 „tulajdonos” iskoláktól.

Az ötödik helyezett Perczes Zsolt több ügyes programot is írt az iskolaszámítógépre. Ezek közül az autóverseny nyerte el leginkább a zsűriző diákok tetszését.

A listák önmagukért beszélnek, úgyhogy az ismertetéstől mindkét programnál eltekintettünk.

### JÉGCSAP

```

20 DIM K$(10)
30 # CHR$(12)
40 POKE 31759,202,197,199,195,211,193,2
   OB,207
   # POKE 31746,170,20B,174,218,211,170:
   POKE 31744,170,20B,174,218,211,170
50 # CUR(5,2)"-DLVADNAK A JEGCSAPOK,"
60 # CUR(6,2)"-EGYRE GYORSABBAN."
70 # CUR(7,2)"-EL KELL KAPNI A CSEPPEKE
   T!"
80 # CUR(8,2)"-NEHA FELTAMAD A SZEL!"
90 # CUR(9,2)"-VABY LETDRIK EGY JEGCSAP
   ."
100 # CUR(10,2)"-DE AZT NE KAPD EL!"
110 # CUR(11,2)"-A-val BALRA,S-sel JOBB
   RA MEHETSZ."
120 # CUR(12,2)"-5 HIBANAL VEGE!"
130 # CUR(15,0)"HA KEZDHETJUK, NYOMD ME
   G A SPACE-t"#
135 IF INP(56Z)=160 THEN 140 ELSE 130
140 RANDDMIZE
150 # CHR$(12)
   # H=20
   # RZ=0
   # PZ=0
160 # CHR$(151)#" Jkkkkkkkkkkkkkkkkkkkkkk
   kkkkkkkkkkkkkkkkkkkkk "
170 A=16
180 FOR T=1 TO 6
   # # CHR$(151)" JJJJJJJJJJJJJJJJJJJJJJJ
   JJJJJJJJJJJJJJJJJJJJJ "
   # NEXT T
190 FOR T=1 TO 14
   # # CHR$(151)
   # NEXT T
200 # CHR$(151)"          ///"
210 # CHR$(151)"#####
   #####"
220 # CUR(23,0)"PONT:"#PZ#CUR(23,14)"HI
   BA:"#RZ#CUR(23,2B)"REKORD:"#MZ#
230 FOR P4=1 TO 1000
   # NEXT P4

```

```

240 FOR Q=1 TO 2
250 FOR B=1 TO 25
260 IF PZ>MZ THEN MZ=PZ
270 # CUR(23,0)"PONT:"#PZ#CUR(23,14)"HI
   BA:"#RZ#CUR(23,2B)"REKORD:"#MZ#
280 IF RZ=5 THEN 470
290 DZ=INT(RND*36)*2+5
300 FOR E=23 TO 3 STEP -1
   # IF DOT(E,DZ) THEN 310
   # NEXT E
   # GOTO 290
310 VZ=0
   # IF PZ<25 THEN 350
320 H1Z=INT(RND*6)
   # IF H1Z=5 THEN 610
   # IF (DZ<25 OR DZ>55) THEN 330 ELSE
   350
330 IF DZ>55 THEN 340
   # VZ=INT(RND*2)
   # GOTO 350
340 VZ=INT(RND*2)-1
350 FOR F=E TO 62
   # SEIDDT F,DZ
360 FOR G=1 TO H
   # NEXT G
370 BOSUB 530
380 CLRDOT F,DZ
390 DZ=DZ+VZ
400 NEXT F
410 IF DOT(F,DZ+VZ) THEN 430
420 RZ=RZ+1
   # OUT 6,135
   # FOR L=1 TO 1000
   # NEXT L
   # OUT 6,0
   # GOTO 270
430 CLRDOT F-1,DZ
   # PZ=PZ+1
   # OUT 6,145
   # FOR Y1=1 TO 100
   # NEXT Y1
   # OUT 6,0
440 NEXT B
450 H=H-5
   # NEXT Q
460 # CHR$(12)CUR(10,4)"EDDIG JOL BIRTA
   D, DE MOST FIGYELJ!":
   FOR Y=1 TO 2000
   # NEXT Y
   # GOTO 160
470 # CHR$(12)CUR(2,2)"PONTOD:"#PZ#CUR(
   2,25)"A REKORD
   #MZ
480 GET V$
   # IF PZ<>MZ THEN 510
490 # CUR(10,0)"MEGJAVITOTTAD AZ EDDIGI
   REKORDOT!"
500 # CUR(12,3)"NEVED?(B BETU)"#:
   FOR Z=1 TO B
   # GET K$(Z)
   # # CUR(16,Z*2+16)#K$(Z)#
   # NEXT Z
510 # CUR(16,0)"A REKORDER NEVE: "
   # FOR Z=1 TO B
   # # K$(Z)#" "#
   # NEXT Z

```

```

520 # CUR(18,21)"UJ JATEK?(I/N)";
# GET G#;
IF G#="I" THEN 140
# IF G#<>"N" THEN 520 ELSE END
530 IF INP(56%)=193 THEN 570
540 IF INP(56%)=225 THEN 600
550 IF INP(56%)=243 THEN 600
560 IF INP(56%)=211 THEN 580 ELSE 590
570 IF A=1 THEN 590
# OUT 6,81
# A=A-1
# # CUR(21,A+3) " #;
# # CUR(21,A)"/"#;
# GOTO 590
580 IF A=36 THEN 590
# OUT 6,81
# A=A+1
# # CUR(21,A-1) " #;
# # CUR(21,A+2)"/"#;
590 FOR M7=1 TO 5
# NEXT M7
# OUT 6,0
# RETURN
600 # CUR(13,15)"UPPER CASE";
# FOR D=1 TO 3000
# NEXT D
# # CUR(13,15)"
# GOTO 590
610 OUT 6,139
# FBZ=A+1
# FOR S3=1 TO 50
# NEXT S3
# FOR T1=INT(E/3) TO 19
# # CUR(T1,FBZ)"J"#;
620 GOSUB 530
# # CUR(T1,FBZ) " #;
# NEXT T1:
IF DOT(T1*3+3,FBZ*2) THEN 420
# PZ=PZ+1
# GOTO 440
630 END

```

```

210 SET(I,7):SET(I,35):SET(I+62,7):SET(I+62,35)
220 NEXT I
230 FOR I=1 TO 42
240 SET(I,14):SET(I,28):SET(I+81,14):SET(I+81,28)
250 NEXT I
260 FOR I=8 TO 34
270 SET(52,I):SET(72,I)
280 NEXT I
290 FOR I=1 TO 14
300 SET(62,I):SET(62,I+27)
310 NEXT I
320 FOR I=10 TO 114
330 SET(I,21)
340 NEXT I
350 X=5:Y=4:U=0:V=0:E=0
360 SET(X,Y)
370 IF POINT(X+1,Y)=-1 OR POINT(X-1,Y)=-1
OR POINT(X,Y+1)=-1 OR POINT(X,Y-1)=-1
THEN 460
380 IF X<9 AND Y=7 THEN E=E+1
390 IF E=10 THEN 580
400 A$=INKEY$
410 IF A$="D" THEN U=2:V=0 ELSE IF A$="A"
THEN U=-2:V=0 ELSE IF A$="Y" THEN U=0:V
=-1 ELSE IF A$="V" THEN U=0:V=1
420 RESET(X,Y)
430 X=X+U:Y=Y+V
440 SET(X,Y)
450 GOTO 370
460 FOR I=1 TO 500:NEXT
470 PRINTCHR$(23)
480 PRINT@64+3,"FALNAK UTKOZTEL!"
490 PRINT@8*64+3,"OSSZESEN";E;"KORT TETTEL MEG."
500 FOR I=1 TO 2000:NEXT
510 PRINT@64+3," "
520 PRINT@8*64+3," "
530 FOR I=1 TO 7
540 SET(62,I):SET(52,I+21):SET(72,I+21)
550 NEXT I
560 RESET(X,Y):GOTO 350
570 FOR I=1 TO 500:NEXT
580 CLS:PRINTCHR$(23)
590 PRINT@7*64,"A 10 KORT HIBATLANUL TETTED MEG."
600 PRINT"AKARSZ MEGEGYSZER JATSZANI ?"
610 IF INKEY$="I" THEN 130
620 CLS:PRINTCHR$(23)
630 PRINT@8*64+9,"VISZONTLATASRA !"
640 END

```

## AUTÓZÁS

```

10 CLS:PRINTCHR$(23)
20 PRINT TAB(11)"AUTOZAS !"
30 PRINT:PRINT"A KEPERNYON EGY PALYA FOG MEGJE-"
40 PRINT"LENNI.A BAL FELSO SAROKBAN LAT-"
50 PRINT"HATO PONT AZ AUTO,AMIT VIZSZIN-"
60 PRINT"TESEN AZ 'A' ES A 'D',FUGGOLE-"
70 PRINT"GESEN AZ 'Y' ES A 'V' BILLENTYU-"
80 PRINT"VEL LEHET MOZGATNI.NEKED 10"
90 PRINT"KORT KELL MEGTENNE HIBATLANUL!"
100 PRINT:PRINT"HA ELOLVASTAD, NYOMJ MEG EGY"
110 PRINT"BILLENTYUT!"
120 IF INKEY$="" THEN 120
130 CLS
140 FOR I=0 TO 124
150 SET(I,0):SET(I,42)
160 NEXT I
170 FOR I=1 TO 41
180 SET(0,I):SET(124,I)
190 NEXT I
200 FOR I=10 TO 52

```

## UTCÁN ÁT ...

A számítástechnika szolgáltatás. Különböző szolgáltatásokat: ptyolat, kultúra stb. utcán át vásárolhatunk. A számítástechnikához kapcsolódó eszközök, szellemi termékek esetében azonban ezt még nem szoktuk meg. Elsősorban azért, mert a számítástechnika alkalmazói a mikrogépek elterjedését megelőzően vállalatok, intézetek voltak. De napjainkban bevonult otthonainkba, egyre szélesebb kör, az utca embere is érdeklődik iránta. Most menjünk ki az utcára és be a boltokba.

**FOTO REVUE**  
Bp. V., Magyar u. 3. 1053

1982 májusában kezdte meg működését. Számítástechnikai forgalma évi 6 millió forint. Ennek mintegy 10 százaléka szoftvereladásból származik. Elsősorban külföldi alapszoftverrel foglalkozik (szövegfeldolgozó, BASIC fordító, grafikus szoftverek stb.). A kisebb és jól ismert gépeken, perifériákon kívül Commodore 720 (256 k) gépet is árul, 2x1 Mb-ajt duál hajlékonylemezzel, printerrel, kb. 1 millió forintért. Kapható IBM PC XT modell is, a kiépítettségtől függően 1,5-2,5 millió forintért.

# i 2716 EPROM PROGRAMOZÓ ADAPTER ZX 81 SZEMÉLYI SZÁMÍTÓGÉPHEZ

A készülék és a kifejlesztett program az i 2716-os EPROM beégetésére szolgál.

Használata számítástechnikai ismereteket nem igényel.

Üzem módválasztás menü szerint.

Széles körű szerkesztési, javítási lehetőségek.

Hexadecimális adatbevitel és ellenőrzése.

Az adatok behozhatók: billentyűzetről, master-EPROM-ból, magnetofonról egészben (2048 bájtt) valamint tetszőleges kis részenként. Egy új tartalom összeállításához a fenti eszközök tetszőlegesen sorrendben és számban vehetők igénybe.

Könyvtárkészítési lehetőség magnetofonon kész- és részprogramok számára.

A beírt, illetve visszaolvasott adatok tetszőleges címtől táblázatos formában ellenőrizhetők (hex. formátum). Alkalmas üres EPROM-ok feltöltésére, valamint ismeretlen tartalmak kiolvasására.

Beégetési idő: 103 s.

Javasolt felhasználási terület: főleg fejlesztési munkák során, ahol az EPROM tartalmát többször kell javítani, módosítani és a könyvtár adat lehetőségek igen jól kihasználhatók.

A készülék szolgáltatásai felveszik a versenyt a kapható nyugati típusokkal, ára (19 990,- Ft) csak tört része azokénak.

**HÍREX Híradástechnikai Vállalat**

**Nyíregyháza  
Tanácsköztársaság tér 15.  
Telefon: 11-313, 11-579**

## Tisztelt jövőbeni Felhasználónk!

Vállalatunk forgalmazza az

## A 6402-es TÍPUSÚ KISSZÁMÍTÓGÉPEKET

és biztosítja a gépcsalád komplex kiszolgálását

(garancia idő alatt és utáni javítást,  
a karbantartást,  
az alkalmazástechnikai és  
telepítési tanácsadást,  
a felhasználói oktatást  
stb.)

**Forgalmazzuk és szervizeljük:**

- vállalatunknál kooperációs együttműködés keretében szerelt svájci leprellofeldolgozó berendezéseket,
- a csehszlovák gyártmányú mikrofiche és mikrofilmolvasó berendezéseket, beállításmérőket,
- az NDK mikrofiche-olvasó és visszanyagító készülékeket, iratmegsemmisítő berendezéseket,
- lengyel nyomtatókat,  
és különféle számítástechnikai berendezéseket, egyedi megrendelések alapján.

## MINTABOLTUNKBAN

– Bp. III., Kerék u. 4. –  
(a Flórián Áruház közelében)

**nagy választékban kaphatók:**

integrált áramkörök, tranzistorok, diódák, trafók, relék, csatlakozók, elemek, gyengeáramú akkumulátorok, kazetták, műszerek, jelzőlámpák, biztosítékok, számítógép alkatrészek, kondenzátorok, mágneses adatrögzítők, személyi számítógépek.

Részletes tájékoztatással és szaktanácsadással készséggel állunk az Önök rendelkezésére. Forduljon hozzánk bizalommal!



## INFORMÁCIÓTECHNIKAI VÁLLALAT

Kereskedelmi főosztály

Bp. III., Kerék u. 6.  
Tel.: 803-294

## Függvényábrák

## és if nélküli definíciók

### JELMAGYARAZAT

^ HATVANYOZÁS

### AZ UTASÍTÁSOK

1. LET Y=(-ABS(X-2)+4+0.5\*(X-2)^2-ABS(0.5\*(X-2)^2+ABS(X-2)-4))/2
2. LET Y=(3-X+ABS(1-X))/2
3. LET Y=1.125\*X+0.5\*ABS(2-X)+0.5\*ABS(1.75\*X-2+ABS(2-X))
4. LET Y=(-SGN(X+1)\*(X+1)^2-(X+1)^2)/2+3
5. LET Y=2\*SGN(SGN(X-1)-1)+4
6. LET Y=3-ABS(ABS(ABS(X)-1)-1)
7. LET Y=-((X+2)^3+ABS((X+2)^3))/2+6
8. LET Y=((X-1)^3+ABS((X-1)^3))/2-1
9. LET Y=((X+2)^2-SGN(X+2)\*(X+2)^2)/2+2
10. LET Y=4\*SGN(SGN(X+1)+1)-1
11. LET Y=(SGN(X-2)\*(X-2)^2-(X-2)^2)/2+4
12. LET Y=X+(3\*X-6+ABS(3\*X-6))/2+(18-6\*X-ABS(18-6\*X))/2+(2\*X-8+ABS(2\*X-8))/2
13. LET Y=((X-2)^3-ABS((X-2)^3))/2+9
14. LET Y=5\*SGN(SGN(-X+1)+1)-2
15. LET Y=-((X+1)^3+ABS((X+1)^3))/2+1
16. LET Y=(SGN(X-1)\*(X-1)^2+(X-1)^2)/2+1
17. LET Y=3\*SGN(SGN(-X-1)-1)+4
18. LET Y=(13-3\*X-3\*ABS(1-X))/2
19. LET Y=X+ABS(X+1)
20. LET Y=2\*X+4-ABS(2\*X+2)

Az előző „Favágás”-ban if nélküli programokat kellett szerkeszteni bizonyos függvények helyettesítési értékeinek kiszámítására. Ezek a függvények azonban nemcsak gyakorlatozásra alkalmasak, hanem fontos gyakorlati szerepük is van. Jól hasznosíthatók más függvények (közelítő) előállításában.

Most néhány függvény ábráját közöljük (betűkkel jelölve) és if nélküli kiszámításukat vezérlő egy-egy utasítást (számozva). A feladat az, hogy meg kell állapítani, mely ábrák és utasítások tartoznak egymáshoz, azaz ugyanahhoz a függvényhez.

Az ábrákon az egység a négyzetháló négyzeteinek oldalhossza. A kezdőpont a tengelyek metszéspontja.

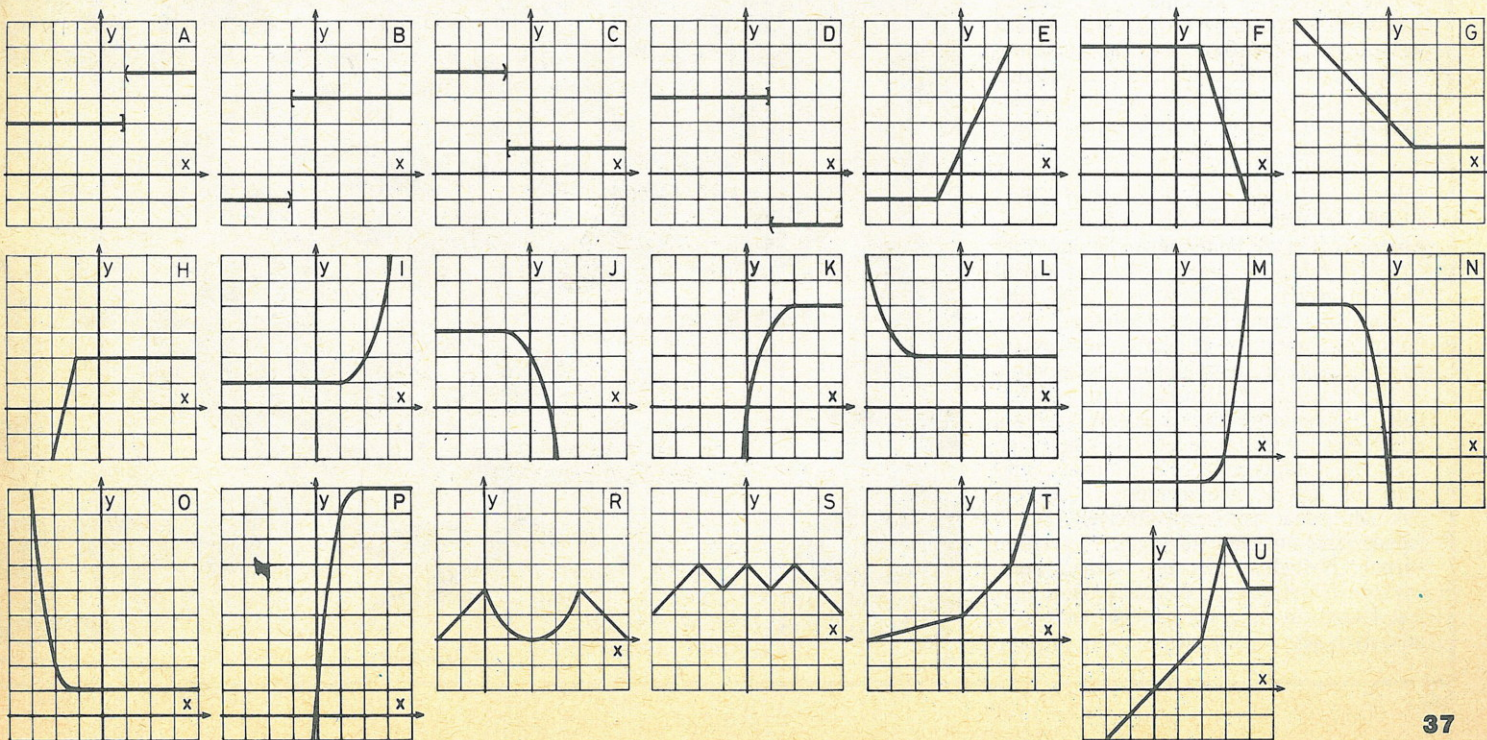
A szakadós függvények ábráján a zárt és nyílt intervallum zárójelét használjuk a határpontnak a vonalhoz tartozása, illetve nem oda-tartozása jelölésére.

Az első 16 ábra mindegyike két szakaszból áll, melyek négyesével azonos fokszámúak. Az első négy ábra nullad- és nullad-, a második négyes nullad- és első-, a harmadik négyes nullad- és másod-, a negyedik pedig nullad- és harmadfokú. Az első négyes kivételével mindegyik függvény folytonos. Érdekes az itt nem szereplő esetekkel is, azaz például a nem folytonos másod- és harmadfokú darabokból álló függvénnyel is foglalkozni. Ki fog derülni, hogy ezek, a most nem tárgyalt esetek mind előállíthatók azokból, amelyekkel most foglalkoztunk. Sőt más érdekes igazságok is megállapíthatók az előállíthatósággal kapcsolatban, amik majd még sorra fognak kerülni.

Akinek pedig még további favágásra is kedve van, írja fel az itteni formulák általános megfelelőit is!

A feladat és a megoldás ellenőrzését Takács Ferenc végezte.

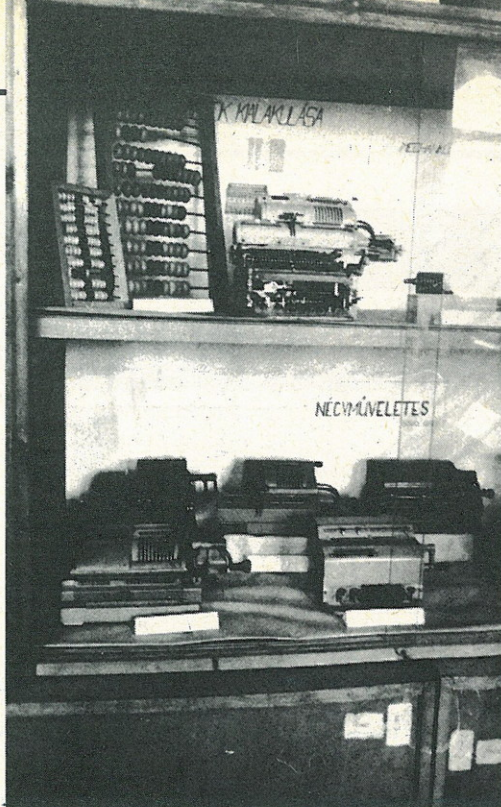
TAKÁCSY ILDIKÓ



**T**ermészetesen! – válaszolja a címben feltett kérdésre dr. Varga Károlyné, az ország egyetlen számológépmúzeumának éltere hívója. – Ahhoz, hogy a tinédzserek igazán értékelni tudják a számítástechnika csodáit, jól kell ismerniük a régi gépeket, szerkezetüket, működési elvüket. Azt hiszem, a huszonegyedik órában sikerült összegyűjteni néhány, kidobásra ítélt számológépet. Most már több, mint harminc, különböző életkorú és típusú számológépet őrzünk, amelyek egy része mai szemmel bizony kicsit komikusan korszerűtlen, pedig a maga idejében a legmodernebbnek számított. Mindent egybevetve: kell nekünk ez a múzeum!

Volna mit megőrizni: a számolási feladatokat ugyanis kezdettől fogva igyekeztek segédeszközökkel megoldani. Barangoljunk kicsit a régmúltban! A kezeken való számolással alakult ki például a tízes számrendszer, később a lábujjak bevonásával a húszas számrendszer. Az ókori kereskedő népek kavicsokat használtak, a rómaiak pedig az abacusnak nevezett számolótablát.

A számolástechnikában oly fontos helyértékrendszer a babilomi kultúrában jelent meg. A tizedesvesszőt a svájci Jobst-Bürgi találta



Bihari Tamás felvételei

## Kell-e nekünk számológép-múzeum?

fel, aki Rudolf német császár udvari órása volt Prágában; elterjesztése viszont a csillagász Kepler érdeme.

A XVII. század elején elkészítettek egy szorzó segédeszközt, az igazi logarlécet azonban 1627-ben találta fel Edmund Wingate angol jogász. Az első, számológépnek nevezhető szerkezet Blaise Pascal nevéhez fűződik. A híres francia matematikus tizenhét éves korában, 1641-ben szerkesztette meg fogaskerekekkel működő, összeadásra, kivonásra alkalmas szerkentyűjét, amely az összeadást úgy végezte, ahogy ma a telefont tárcsázzuk. Az első, szorzásra alkalmas gépet Leibniz készítette, harminc évvel később. Gépének jellegzetessége egy rögzített eredménymű, valamint az eltolható beállító- és számológép, amelyeket egy lépcsőshenger hajtott. Az osztás művelete azonban Leibniz gépe számára is kemény dió volt.

A számoló- és írógépek összeállításának több évtizedes darabjai a mezőtúri Teleki Blanka Gimnázium és Közgazdasági Szakközépiskola egyik tantermében láthatók. Az életképes ötletet dr. Varga Károlyné mondhatja magáénak, aki ügyviteltechnikát tanított az iskolában, és mint a régiségek nagy barátja, határozta el a lejárt szavatosságú, műemlék jellegű számológépek összegyűjtését. Felhívására vállalatok, állami gazdaságok, de főleg szövetkezetek jelentkeztek, s a múzeumot gazdagítandó, elküldték gépeiket.

Persze, korántsem teljes a gyűjtemény, hiszen sok helyütt rejtőzhetnek régóta nem használt irodai eszközök, amelyek csupán felfedezésre, múzeumi tagságuk elnyerésére várnak. Nyugdíjas pénzügyi szakemberek, egykori földművelésszövetkezeti irodai dolgozók bizonyára hasznos tippet tudnának adni ilyen gépek lelőhelyeiről.





## Gyorsítási fogások

A gyorsítási fogások köre nagyon tág. Lelkiismeretes és pontos számbavételükhöz egy egész kötet sem volna elég. Ezért itt természetesen csak néhány fontos típust említhetünk meg.

### Gyorsítás gyorsabb, egyenértékű műveletre cseréléssel

Ez a leggyakoribb, legegyszerűbb gyorsítási fogás. Általános szabály nincs, ez a terület a „vegyük észre” birodalmába tartozik. Főleg műszaki-tudományos számítások terén szokott lehetőség nyílni ilyen típusú gyorsításokra. Az azonosságok, nevezetes szorzatok ismerete feltétele annak, hogy e fogásokat alkalmazni tudjunk.

Elrettentésképpen közöljük a következő példákat.

Volt már programozó, aki a  $[0; \pi/2]$  tartományban táblázatot készített a  $\sqrt{1 - \cos^2 x}$  függvényről a  $\cos x$  négyzetének kiszámításával, minden lépésben ellenőrizve a gyök alatti értéket, nehogy negatív legyen, aztán pedig négyzetgyököt vont.

Akadtt olyan „kutató” is, aki sok ezer átvételenszám generálás segítségével gépi úton tapasztalatilag bizonyított be egy Poisson-eloszlásokra vonatkozó, általa nem ismert, de minden valószínűségelméleti tankönyvben szereplő, közismert télt.

Utolsó példánkat már egy művelt számítástechnikus „követte el”. Tankönyvből jól megtanulta, hogy polinom helyettesítési értékének kiszámítására a leggyorsabb út a Horner-elrendezés. (Azt is tudta, hogy ez „be van bizonyítva”). Ennek szellemében az  $x^3 + 1$  polinom helyettesítési értékének kiszámítását gondolkodás nélkül a Horner-elrendezés programozásával oldotta meg.

### Gyorsítás szervezéssel, az adottságok kihasználásával

Gyorsíthatjuk programunkat olyan esetekben, ha több olyan tennivaló (például kiszámítani való) van, amelyekben sok a közös elem. Ilyenkor gyakran érdemes úgy megszervezni a (számolási) folyamatot, hogy az azonosan ismétlődő tennivalót csak egyszer végezzük el.

Példa: az  $ab^2cd^2e^2f^3$  és az  $a^2b^2c^2de^2f^2$  kiszámítását érdemes úgy végezni, hogy az  $ab^2cde^2f^2$  értékét számíttjuk ki, ezt megszorozzuk  $df$  értékével és  $ac$  értékével.

### Gyorsítás a lefutási pálya átszervezésével

E fogás alkalmazásához elengedhetetlen az eljárásokat vezérlő esetek gyakoriságának ismerete. Enélkül megtörténhet, hogy az ilyen beavatkozással nem gyorsítani, hanem lassítani fogunk.

Ez a fogás átlagos gyorsítást eredményez. Előfordulhat, hogy egyes esetekben a program lefutása lassúbb lesz, mint a gyorsítás előtt volt. Mindenekelőtt tudatosítanunk kell, hogy

minden rendszer felfogható kiszolgáló rendszerként is. Így a számológépben zajló folyamatokat kiszolgálási folyamatokként is vizsgálhatjuk.

Programunkkal automatikusan definiáljuk egy kiszolgáló rendszer működésrendjét. Célszerű, hogy ennek minél nagyobb átlagsebességet adjunk.

Lássunk egy példát! Osszunk fel egy intervallumot 20 részre! Függhjön a számológéptől igényelt tennivaló attól, hogy egy bemeneti adat e részek melyikébe esik. A vizsgálatot a szakaszvégpontokkal való összehasonlítás segítségével kényelmesen elvégezhetjük, ha azokat növekvő vagy csökkenő sorrendben vesszük sorra. Világos, hogy így sok felesleges munkát fogunk végezni akkor, ha a bemeneti adatunk olyan, hogy egy középtáji részbe esésének valószínűsége 98,1 százalék és a többi 19 részintervallum mindegyikébe pedig csak 0,1 százalék. Ilyenkor érdemes először azt megvizsgálni, hogy nem a 98,1 százalékos gyakoriságú esettel van-e éppen dolgunk.

Ha ismerjük az értékek előfordulási valószínűségét a különböző szakaszokban, megszerkeszthetjük az annak a szakasznak a leggyorsabb megkeresését vezérlő programot, amelybe egy adott érték belesik.

Részintervallum-végpontok bal- vagy jobbszélről való sorra-vétele biztosan a leggyorsabb eljárás, ha az előfordulási valószínűségek monoton fogynak, illetve monoton növekszenek. Az egyenlő előfordulási valószínűségek esete mindkét előbbinek határeset. Ebből következik, hogy közel egyenlő előfordulási valószínűségek mellett érdemes a könnyen programozható, szélről induló keresésnél maradni. Egyéb esetekben azonban már a gépi adottságokat is figyelembe kell venni, és tér nyílik a programozó leleményességének is.

Ha az előfordulási valószínűségek nem egyformák, és nem monoton változnak, gondolhatnánk, hogy csökkenő valószínűségű sorrendben kell az intervallumokat sorra vizsgálni. Ez azonban nem így van.

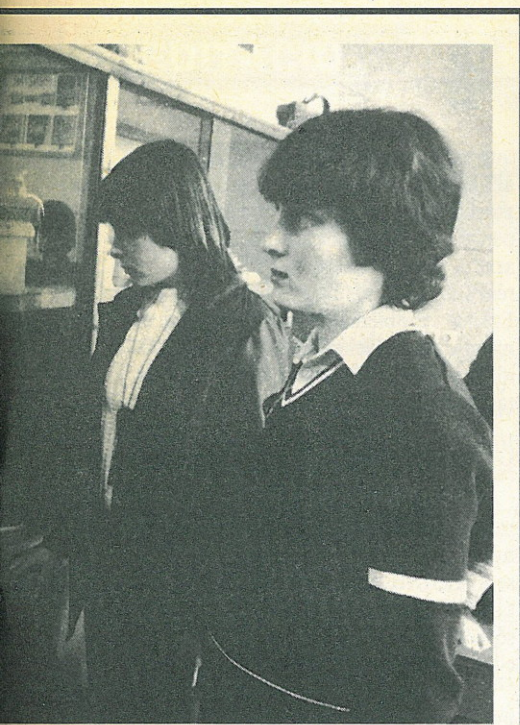
Az olvasóra bizzuk ennek a nagyon érdekes és hasznos problémakörnek a további vizsgálatát, azt javasolva, hogy a következő, egyszerűbb esettel kezdje a munkát.

Hogyan célszerű a vizsgálatokat felépíteni, ha a szóban forgó bemeneti adat az  $1, 2, \dots, k$  értékeket és csak ezeket veheti fel, rendre  $p_1, p_2, \dots, p_k$  valószínűséggel?

Ez és az előző speciális probléma szakaszonként különböző módon kiszámított függvények helyettesítési értékei kiszámításánál a gyakorlatban széles körben jelentkező, fontos kérdés.

Érdekes, de nehezebb eset, ha a géppel elvégezteni kívánt tennivaló több állandó számú értéktől függ, amelyeknek gyakoriságeloszlását ismerjük. Ebben az esetben több dimenziós (többváltozós) térbeli téglatestekbe esést kell vizsgálnunk.

Fontos gyakorlati eset az, amelyben ismerjük azokat a téglákat (ezek száma viszonylag nagy), amelyekbe biztosan nem esik érték, illetve pont.



S még valami. Szerencsés lenne kiegészíteni a múzeumi anyagot tablókkel, a számológépek történetét bemutató ismertetőkkal, szemléltető eszközökkel. Lenne miről mesélni...

Száz évnek kellett elteltie az első számológép megjelenése után, hogy megszülessen az a kerek órára hasonlító szerkezet, amely elvégzi a négy alapműveletet. Megalkotásának érdeme Philipp Mathaus Hahn württembergi matematikusé. Az angol Charles Babbage a múlt században már azon töprengött, hogyan szerkeszthetne gépet az összes matematikai probléma automatikus megoldására. Gépe a számításokat végző egységekből, a részeredményeket tároló szerkezetből és az ezeket működtető irányító és vezérlő automatákból állt. Az akkori idők legtökéletesebb számológépét mégis az orosz Csebisev készítette, s gépének alapelvét más feltalálók is alkalmazták. Századunk elejének legismertebb számológépét viszont a német Christal Hamann mérnök csinálta, aki a sokáig megoldhatatlannak tűnő osztás gondját vette le a gépszerkesztők válláról.

A vitrinekben az elvégezhető műveletek szerint csoportosítva sorakoznak a számológépek. Két-, három- és négyműveletesek, amelyek között kézi meghajtású, félautomata és automata egyaránt látható. Persze, nemcsak látható. Ebben az iskolai múzeumban érvénytelenítették a „mindent a szemnek, semmit a kéznek” hagyományos múzeumi szabályt, hiszen a szakközépiskolások ma is használják egyik-másik kiállított gépet. No, nem a gyűjtemény legrégebbi darabját, a hetvenéves Rheinmetall gépet, hanem a közel-múlt számológépeit. A használható Rheinmetall, Triumphok és Melitták karbantartásáról, üzembe helyezéséről rendszeresen gondoskodik egy irodagép-műszerész, akinek alkalmanként az Ascoták, Olivettik hiányzó alkatrészeit is el kell készítenie.

A rendhagyó iskolamúzeum berendezésében, a kollekcio kiállításában a szakközépiskolások is lelkesen segítkeztek. S ha a tananyag-adagok, ügyviteli programok, gazdasági feladatok megoldásakor nem is olvadnak el a gyönyörűségtől, mégis féltő gonddal óvják saját múzeumukban a számítástechnikai múlt egy darabkáját.

CSONTOS TIBOR

További fontos, de még nehezebb feladat minél gyorsabb vizsgálatot elvégző program szerkesztése akkor, amikor az igényelt tennivalók több adattól függenek, de ezeknek az adatoknak a száma nem állandó, és az értékelőszókra vonatkozó információink csak becslések, amelyeknek pontosságáról azonban minden szükségeset tudunk.

Hasonló jellegű az adatállományból való több ismérv, több jellemző szerinti keresés, ki-gyűjtés problémája.

Lássunk egy egyszerű példát. Ha a  $T_1$  és  $T_2$  tulajdonsággal egyidejűleg rendelkezőket keressük, és minden vizsgálat ugyanannyiba kerül (az időt illetően), akkor, ha sokkal több  $T_1$  tulajdonságú van, mint  $T_2$  tulajdonságú, érdemes a  $T_2$  tulajdonságúakat kiválogatni előbb, majd pedig ezek közül a  $T_1$  tulajdonságúakat.

Természetesen más a helyzet, ha egyidejűleg ugyanazt a tárat több processzor is használhatja. De a sebességgel kapcsolatos problémák ilyenkor (egyelőre még nem személyi gépeknél) csak egy-két területen egyszerűsödnek; általában inkább gazdagodnak és nehezebbé válnak.

## Gyorsítás ismétléssel és alkalmas redundancia bevezetésével

Ha egy eljárást alprogramként kezelünk, az alprogrammal közölni kell a független változóknak azokat az értékeit, amelyekkel dolgoznia kell. Ha sok a független változó, ez az adatforgalmi rész időigényes lehet, és ha az alprogram egyébként nem hosszú, és sokszor szerepel ugyan, de nem sok helyen, akkor érdemes lehet minden helyre külön-külön beírni (a „zárt szubrutin” helyett „nyílt szubrutin”-t alkalmazni), megtakarítva ezzel az eljárás bemeneti változóinak történő külön értékadásokat.

Ha van helyünk és a nagy sebesség fontos követelmény, adatállományainkat a keresési feladatok meggyorsítására „fellazíthatjuk”, előfeldolgozott állapotba hozhatjuk.

Álljon adatállományunk például háromkomponensű vektorokból. Készítsük el azokat az adatállományokat, amelyek ennek az első, a második majd a harmadik komponens szerinti növekvő sorrendbe rendezése révén állnak elő.

Ha feladatunk az, hogy időről időre ki kell gyűjteni és további feldolgozásnak alávetni az eredeti adatállomány olyan vektorait, amelyeknek hol ez, hol az a komponense adott korlátok közé esik, akkor e háromszorosra duzzasztott adatállománnyal lényegesen gyorsabban tudunk dolgozni, mint az eredetivel. (Megjegyezzük, hogy a gyorsításban itt a rendezésnek is fontos szerepe van; a rendezés az egyik leghatékonyabb előfeldolgozási módszer.)

## Gyorsítás vizsgálatok beépítésével

Ezt a gyorsítási típust példával szemléltetjük. Több, bonyolult, időigényes eljárást kell használnunk egy több tényezős szorzat tényezőinek kiszámítására. A szorzatot sokszor ki kell számítani. Ha tudjuk például, hogy az egyik tényező gyakran 0, és az is, hogy milyen gyakran, akkor érdemes lehet a programot úgy felépíteni, hogy először ennek a tényezőnek az értékét számoljuk

ki, és megvizsgáljuk, hogy az nem 0-e. Ha igen, nem kell a többi tényező kiszámítására időt fordítanunk. Mit érdemes tenni, ha ismerjük az összes tényező zérussá válásának valószínűségét?

## Gyorsítás a bemeneti adatoktól független, állandóan ismétlődő munka előzetes elvégzésével

Nem ritka az olyan, egyébként jól felépített program, amelyben van olyan rész, ahol minden futás alkalmával ugyanazt ugyanazzal az értékkel kell kiszámítani.

Az ilyen programot lehet úgy gyorsítani, hogy az előbb említett részt külön lefuttatjuk, és csak ennek eredményét tesszük be a programba, kihagyva belőle az időigényes, mindig ugyanúgy ismétlődő, ugyanazt eredményező részt.

## Gyorsítás több program elkészítésével

Vannak olyan feladatok, amelyekben, ha minden esetet egy programmal akarunk elintézni, a program nagyon bonyolult és lassú lesz. Az általános program helyett több speciális (eseti) program készítése meggyorsíthatja a munkát, különösen olyankor, amikor az általános célú program szolgáltatásai közül minden gyakorlati esetben csupán néhányra van szükség.

Statisztikai programoknál nem ritka az ilyen felesleges szolgáltatás. Megkapjuk például az átlagot, a terjedelmet, a mediánt, a szórást stb. olyankor is, amikor csak az átlagra és a terjedelemre van szükségünk.

Készült már programrendszer tudatosan úgy, hogy minden speciális igényre külön, csak arra a célra írt program álljon rendelkezésre. Ez az elv tárálódzatok révén ugyan, de lényeges időmegtakarítást tesz lehetővé.

## Gyorsítás kísérletezéssel

Az esetek többségében ma még nem tudjuk elméleti úton megállapítani, hogy ugyanarra a célra melyik program milyen sebességjellemzőkkel rendelkezik. Marad a mérés, a tapasztalat, a különböző eloszlású bemeneti adatokkal végzett próba futások megfigyelése és az ezek eredményeiből levont következtetés.

Foglalkoznunk kell tehát a futási időtartamok gyakorlati megállapításának módjaival is.

Az olyan gépeknél, amelyekben van beépített óra, és programozni tudunk különböző időméréseket, ez a feladat egyszerűen megoldható. A kisebb gépek egy része ma még nem rendelkezik ilyen kényelmes lehetőséggel. Ezekre vonatkozóan a következőket mondhatjuk.

Ha a gyár megadja a műveleti időket, akkor a gépi kódban írt programok futási idejét, még az átlagos futási időt is, több-kevesebb nehézséggel ki lehet számítani. (Más kérdés, hogy megéri-e ezt az egzakt utat választani.)

Magasabb szintű nyelvek alkalmazása esetén azonban a fordító-, illetve értelmező program szerkezetének ismeretére is szükség volna. Ez az út a gyakorlatban általában már nem járható. Marad a mérés.

Hogyan mérjük? Egyszeri futás mérésére a stopperóra csak akkor jó, ha a mérendő időtar-

tam elég hosszú. Kisebb időtartamok esetében általánosan használható fogás a program sokszoros futtatása együttes idejének mérése (stopperórával). Ilyenkor azonban gondolnunk kell arra is, hogy a sokszoros futást vezérlő programrész futási ideje is benne van a mért összidőben. Ezt a problémát lineáris egyenletrendszer-megoldással el lehet intézni, ha több, különböző idejű (különböző gyakoriságú) sokszoros futtatást végzünk.

## Gyorsítási eljárások egymás ellen

A sebességgel kapcsolatos gyakorlati alkalmazástechnikai fogások végére érve hangsúlyoznunk kell, hogy az említett néhány típussal közel sem merítettük ki a gyorsítási ötletek gazdag tárházát. Nem szoltunk arról a kellemetlen jelenségről sem, hogy egyik gyorsítási fogás a másik ellen dolgozhat, és ezt nem ritkán meg is teszi.

Vannak-e egyszerű, általános szabályok az ilyen esetekre, és ha igen, melyek azok? Ezt e sorok írója is szeretné tudni.

Az egész témakör összefoglalásaként – talán furcsa, de – azt mondjuk, hogy kerülnünk az öncélúságot. A program fő feladata általában nem az, hogy az eredményt a lehető leggyorsabban állítsa elő, hanem az, hogy egyáltalán szolgáltatassa. Vannak feladatok, ahol más szempontok, így például a gyorsaság is lényeges szerepet kaphat. Ezeket a feladatokat az átlagosaktól eltérő szelvényben kell megoldani, a programozás jellege is más lesz. Az öncélú sebesség megszállottai, ugyanúgy, mint a közutakon, a programkészítés terén is veszedelmek lehetnek. Gyakorlati haszon nélkül való feladványaikkal, bűvészműtávkal elterelhetik a figyelmet más, fontos szempontokról.

A sebesség fokozásának mestersége, a sebességgel kapcsolatos ismereteink ma még nagyon kezdetlegesek. A fejlődésnek minden bizonnyal nagy lökést fog adni a technológiai rendszerek számítógépes irányításának általánossá válása és a több, egymással egyidejűleg ugyanazon a feladaton is dolgozni képes feldolgozóegységgel (processzorral) rendelkező számológépek elterjedése.

POGÁNY CSABA

## Megoldások

Az „Agyafúrmány” 1984-es számunkban közölt feladataira helyes választ küldtek be:

I. korosztály: Helli Péter, Eger, VI. sz. általános iskola.

II. korosztály: Vadász Ágnes, Bp., I. István Közgazdasági Szakközépiskola; Veres Zoltán, Bp., József Attila Gimnázium.

IV. korosztály: Aradvári Tiborné, Bp.; Sipos Sándor, Pécs; Veres Jolán, Bp.

A „Favágás” feladatainak megoldása:

Az egymáshoz tartozó utasítások és ábrák: 1-R, 2-G, 3-T, 4-J, 5-A, 6-S, 7-N, 8-M, 9-L, 10-B, 11-K, 12-U, 13-P, 14-D, 15-O, 16-I, 17-C, 18-F, 19-E, 20-H, illetve A-5, B-10, C-17, D-14, E-19, F-18, G-2, H-20, I-16, J-4, K-11, L-9, M-8, N-7, O-15, P-13, R-1, S-6, T-3, U-12.

# KINEK LESZ IGAZA?

## Szekrény méretűek

Az 1965 közepén megjelent és 1972-ig gyártott IBM 360/40 típusú nagyszámítógép döntő mértékben hozzájárult a fejlett tőkés országok számítógépesítési folyamatának megindításához. ESZR programbeli megfelelője, a sorozatban 1972-től a hetvenes évek második feléig gyártott R20 számítógép, még ma is a hazai és szocialista számítógéppark gerincét képezi.

A nagy szekrények lenyűgöző méreteinek hatása alatt hajlamosak vagyunk elfelejteni, hogy milyen valós számítógép-teljesítményt nyújtottak és nyújtanak ezek a gépek. Nem árt néhány műszaki adatot felidézni.

A 360/40 processzorának ciklusideje 625 ns, aritmetikai-logikai egysége 8 bit (1 bájt) szélességű, operatív memóriájának ciklusideje 2  $\mu$ s (2000 ns). A beviteli-kiviteli műveletek vezérlésével foglalkozó szelektor és multiplex csatornák hardvereszközeit részben a processzor hardver (többek között az aritmetikai-logikai egység és a mikroprogramozott vezérlés) eszközeinek megosztott igénybevételével valósították meg. A berendezés háttértárrendszerének kiépítési alapegysége 7,25 Mbájtos lemezegység volt.

A rendszerek átlag 895 ezer dolláros eladási árához még jelentős járulékos többletköltséget jelentett a nagyméretű és speciálisan klimatizált gépterem megépítése, valamint a nagy létszámú kezelő és kiszolgáló személyzet felállítása. Az R20 számítógép esetében szinte egy az egyben hasonló a helyzet.

## Íróasztalra valók

Ezzel szemben áll az a mikromini (tehát nem személyi számítógép!), amely 0,5  $\mu$ s ciklusidejű (4 MHz) nyolcbites mikroprocesszorral (Z80A) működik, operatív memóriájának ciklusideje 1  $\mu$ s-nál kisebb, a hajlékonylemezek és a Winchester lemezek vezérlését nagy intelligenciájú vezérlő áramkörök autonóm üzemmódban ellátják stb. Egy ilyen tipikus rendszer max. 1 Mbájttal rendelkezik, Winchester lemez alapú háttértár-

rendszerének kiépítési alapegységei 10 és 16 Mbájtos (formátumozott) lemezegységek, amelyekkel több száz Mbájttal kapacitást el lehet érni.

Egy, az előbbi 360/40-nél (illetve az R20-nál) nagyobb kapacitású mikromini rendszer (0,5 Mbájttal operatív tár, 48 Mbájttal Winchester kapacitás, 6 db terminál, illetve 6 felhasználó, 2 db nyomtató, hajlékonylemez) vételára kb. 10 ezer dollár. Ez az ár egyúttal jóval fejlettebb alkalmazás-fejlesztési szoftverkörnyezetet is nyújt, mivel benne van az operációs rendszer mellett kétféle BASIC fordító, C és PASCAL fordítók, professzionális szövegdolgozó programcsomag, alkalmazás-generátor és adatbázis-kezelő rendszer, képernyőorientált elektronikus feladatlap (spreadsheet) és könyvelési programcsomag is. Tehát egy sor olyan alkalmazásfejlesztési eszközt megkap a felhasználó, amelyek annak idején ismeretlenek voltak, ma azonban a hatékony és célszerű fejlesztés elengedhetetlen eszközei.

Egy ilyen nagy kapacitású mikromini rendszer üzembe helyezése járulékos beruházást sem igényel, mivel a központi egység egy közönséges irodai íróasztalon elfér, csakúgy, mint a terminálok és a nyomtató berendezések. A berendezések üzemeltetéséhez légkondicionálás sem szükséges. A fejlett alkalmazásfejlesztési környezetben külső szakmai támogatással egész rövid idő alatt kialakítható a komplett és az erőforrásokat kihasználó feldolgozási rendszer, üzemeltetése speciális szakértelmet és személyzetet nem igényel, illetve bizonyos további feldolgozási igényeket maga a „naiv” felhasználó meg tud fogalmazni például az elektronikus feladatlap segítségével.

Nyugodtan mondhatjuk tehát, hogy alig több, mint 15 év alatt az Egyesült Államokban egy nagyobb számítógépes információs

rendszer kiépítésének és használatának költsége még névlegesen is két nagyságrenddel kisebb lett. Emellett a berendezések üzembiztonsága is legalább egy nagyságrenddel javult.

Ezt a mondhatni „drámai” fordulatot mindössze az utóbbi néhány év gyors egymásutánban következő eseményei eredményezték. Az váltotta és váltja ki, amit ma mikro-számítás-technikának szokás nevezni. Az LSI és VLSI áramköri technológia szinte ontja az elkészítően olcsó és egyre nagyobb teljesítményű processzor-, speciális vezérlő és memória-áramköröket. A nagy sorozatú és nagy teljesítményű elektronikus egységek igen alacsony ára óriási lehetőségeket nyitott és nyit a hasonlóan nagy sorozatú, megfelelően nagy teljesítményű, és hasonlóan olcsó árú mikro-perifériák (hajlékonylemez, Winchester, nyomtatók) offenzívájának. Az ilyenformán milliószámra megjelenő, nagy teljesítményű számítógépek megteremtették az alapját a tömeges szoftverpiacnak, ami az alkalmazásfejlesztést nagyságrendileg hatékonyabban támogató szoftvereszközökben és megbízhatóságuk mellett fantasztikusan olcsó árakban nyert kifejezést.

## Melyiket válasszuk?

Nem árt tehát emlékeztetni arra, hogy még a legegyszerűbb professzionális mikro-számítás-technikai eszközök között is vannak olyanok, amelyek igen kedvezően összehasonlíthatók a közelmúlt (és nálunk még mindig a jelen) „nagyjaival”. Azért is kell hangsúlyozottan emlékeztetni erre, mert a hazánkban kibontakozó „mikroszámítógépes láz” éppen nem a korábbiakkal összehasonlítható eszközök területén „szedi áldozatait”.

A sajtó és bizonyos szakmai körök által kifejtett szakmai propaganda akaratlanul is a házi

és félprofesszionális eszközöket (Sinclair, iskolaszámítógép, Commodore) teszi meg a mikro-számítás-technika sztárjainak. A professzionális eszközök területén a „professzionális személyi számítógép” (MO8X, Janus, Labsys, TAP 34, Varyter, Floppymat-SP stb.) az egyeduralgoló. Pedig a házi eszközöket a komoly számítástechnikai feldolgozások szempontjából primitívnek kell minősítenünk. A hazai „professzionális” eszközök pedig – egyértelmű perifériális és szoftverellátási „gyengélkedéseik” miatt – jelenleg az import félprofesszionális berendezések színvonalán állnak, vagy azt alig haladják meg.

Ennek ellenére piaci árak itt-hon eléri a megfelelő USA-beli ár tízszeresét vagy a nyugat-európai ár ötszörösét. Egy kész alkalmazási rendszer készítésének költsége akár félmillió forintot is kitehet, de semmi esetre sem kevesebb százezer forintnál. Tehát személyes feldolgozásokban való korlátozottságuk ellenére ezek az eszközök egyáltalán nem kis beruházást jelentenek az alkalmazóknak.

Hogy a fizetőképes keresletet magukhoz vonják, szinte minden cégnél (akár kisvállalkozásról, akár nagyobb vállalatról legyen szó) találkozni a szakmai felelőtlenség jeleivel. Olyan feladatok megoldására is ajánlják a kis kapacitású, félprofesszionális berendezést, amire az alig, vagy szélső esetben egyáltalán nem alkalmas.

Szinte előre látni azt a csalódási hullámot, amivel ennek következtében az elkövetkező években szembe kell néznünk. Végül még igazuk lesz azoknak, akik mindig is komolytalannak tekintették a mikroszámítógépeket, vagy például azok fő alkotóelemét, a mikroprocesszort. A házi és a félprofesszionális berendezésekben ugyanis gyakran előfordul ugyanaz a Z80A mikroprocesszor, amit korábban említettünk. Igaz, hogy ott más feladatokat is elláthat a központi processzor funkciók mellett, és így hasznos processzor kapacitása kisebb is lehet a műszaki jellemzők által meghatározottnál. A lényeg azonban nem is annyira ez, hanem inkább a számítógép kapacitását meghatározó többi tényező.

NACSA SÁNDOR

**A levelek őszinte örömeimre nagyon változatos témájúak, ezért szívesen állítom össze olvasóink színes gondolatainak összefoglalóját.**

**ifj. Laurik Pál, Jászberény,  
József Attila tér 6. 5100**

Számítógépem ugyan még nincs, de érdekel. Bizonyítja az is, hogy két könyvem van róla, és a  $\mu$  M, amiből több is lesz. Már sok mindent tudok, úgy érzem, csak nincs min hasznosítani, tanulni, kipróbálni. Most még nem lehet kilátásom rá, ugyanis építkezünk, de Önök szerint melyik az a legjobb, nem iskolai számítógép, amit a legérdemesebb megvenni, és ami sokat tud? Olyan választ szeretnék, ami megmondja, melyik külföldi vagy jó minőségű magyar számítógép a legjobb és legolcsóbb.

*Válaszom: nem tudom. Van olcsó gép, van jó gép, de a legjobb?! Erre a kérdésre Fordnak, az „autókirálynak” a válasza jut eszembe. Amikor megkérdezték tőle, hogy melyik a legjobb kocsi, „Az ÚJ!” mondta. Egyébként részletes levélben válaszoltam.*

**Szabó László, Békés,  
Teleky u. 105. 5630**

Tetszik az újság szerkesztése, kivéve pár rovatot. Az első rovat, amely nem tetszik, a Hírek, érdekességek. Van benne pár cikk, amely jó, de szerintem itt a számítástechnika legújabb találmányait kellene felsorolni, például a Sinclair cég új gépét, a QL-t.

A másik ilyen rovat az Új könyvek. Ezek az „új” könyvek annyira újak, hogy a könyvesboltból már régen elfogytak. Szóval inkább könyvlehetést kellene adni!

Végül harmadikként a Tanfolyamról szólnék. Ez ellen nincs nagy kifogásom, csak két részre bontanám: egyet a kezdőknek, egyet a haladóknak.

Lenne egy javaslatom is. A kit-be rakják a PRIMO-t! Ez valószínűleg olcsóbb lesz, mint a mostani ZX81-hez hasonló kit-gép.

*Biztosan elhiszi, hogy sokaknak ezek a rovatok is tetszenek, nem vagyunk egyformák. Ami a híreket illeti, vegyesen közlünk új „találmányokat” és olyat is, ami talán nem nagyon új, de érdekes. Sokan nem ismerik a már kiadott könyveket sem, ezekről az ismertetést nekik szánjuk. Lesznek előzetes híreink is, ha kapunk ilyeneket. A tanfolyam most a kezdőknek szól, a haladó szoftver- és hardveramatőröknek a többi rovat cikkeit ajánljuk. Ami a kit-et illeti, amikor ezeket a sorokat írom (július), még sikertelenül próbálkozunk, többek között a PRIMO-val is. De nem adjuk fel.*

**Huszár Attila, Budapest,  
Jós u. 7. 1039**

Kezdő vagyok a számítástechnikában, de szeretném tudásomat továbbfejleszteni. Főleg a játéprogramok érdekelnek nagyon, de az ilyen

program írásához több szakértelem kell, mint amilyennel én rendelkezem. Bár rövidebb játékokkal én is sikeresen próbálkoztam, de én nemcsak logikai, hanem „action game” programokat is szeretnék írni.

Kérdéseim: 1. A HCC-be beléphet haladó és kezdő egyaránt? 2. Hány hónaponként jelenik meg a lap? 3. Hol lehet a gépépítést elsajátítani? 4. Mi a HCC főbb (ha van ilyen) oktatási területe? 5. Hogyan lehet a HCC tagja lenni? 6. Mikor jelent meg az első szám?

*A kérdéseket sok más olvasó is felteszi. Dr. Simonyi Endre, a HCC elnöke válaszol:*

*1. A HCC-nek minden Magyarországon élő személy tagja lehet (külföldi pártoló tag lehet). 2. Kéthavonta. 3. A Klubon belül. 4. Erre a kérdésre a lap első számában bemutatkozó cikkünk már válaszolt. 5. A Klubba a következő ülésen lehet belépni, melynek időpontjáról a Neumann Társaságban kaphat felvilágosítást. 6. 1983 decemberében.*

**Szedmer József, Esztergom,  
Mártírok útja 17. 2500**

A 84/1. számhoz is hadd mondjam el véleményemet. Ismét hiányoltam, hogy nem tartalmaz olyan részt, ami a teljesen amatőrökhöz szólna. A másik gondom szorosan kapcsolódik ehhez. A programozást csak akkor lehet igazán megtanulni, ha van min gyakorolni is. Ma sajnos gép vásárlása nagyon kevés embernek adatik meg, munkahelyen vagy nincs, vagy nem engedik használni. Éppen ezért a házi gépépítés ebben nagy segítséget tudna nyújtani. Többek között ezért is vártam már az új számot, de sajnos ebben is csak ígéret van, amit én már fél éve megkaptam. Bízom benne, hogy a következő számban ez az ígéret már valóra válik. Különösen fontosnak tartanám azért is, mert mint a Számítástechnika folyóiratban megjelent, szeptembertől a tévé is indít BASIC tanfolyamot, amihez mindenképpen gép szükséges, viszont biztos vagyok abban, hogy sokkal többen fogják csak nézni a tévét, de géphez nem tudnak ülni. Ezért ha szeptemberre megépíthetnék a magunk gépet, azokat remekül tudnánk hasznosítani.

*Igaza van, de ... Vállalva a programozás és számítógép-kezelés, sőt -építés tanítását, a lap, mint általában a hasonló külföldi magazinok is, nagyobb terjedelemben közöl cikkeket a haladó amatőröknek. A tévétanfolyamot az NJSZT és a  $\mu$  M kezdeményezte; őszinte örömeinkre önzetlen támogatást kaptunk a tévétől, a SZÁMALK-tól és a TII-től. Ezért lesz tanfolyam és lesznek  $\mu$  klubok (lásd 1984. 3. szám), és talán lesz kit-számítógép is; erre vonatkozóan más levelezőimnek már válaszoltam.*

**Tóth György, Balatonalmádi,  
Dózsa György u. 26. 8220**

Én nem vagyok számítástechnikai szakember, de a téma érdekel. Géppel nem rendelkezem, de szeretnék egy megfelelőt venni. Valószínűleg nem vagyok egyedül, akit érdekelne a nálunk

beszerezhető PC-k összehasonlítása (teljesítmény, RAM, ROM, ár, grafika stb.). Olvastam egy új magyar gépről, a PRIMO-ról. Szeretném, ha írnanak róla: mit tud, mibe kerül, mikortól és hol lehet beszerezni.

*A PRIMO-ról a karácsonyi számban írunk.*

**Kovács László, Karcag,  
Zádor u. 8. 5300**

Sok kesergő utalást olvastam a gépek szakirodalma miatt. Ezzel kapcsolatban szeretnék egy jó hírt közölni. Az Ipari Informatikai Központban (Budapest V., Arany János u. 24. III. em. 305.) ZX81 és ZX-Spectrum magyar nyelvű gépkönyv vásárolható. A könyv ára 320 forint. Biztosan sokakat érdeklő információ, amit érdemes lenne mihamarabb közreadni.

*Köszönjük.*

**Mino Attila, 7. oszt. ált. isk. tanuló  
Szekezsárd, Kilitátó u. 3. 7100**

Ha megkérhetném Önöket, akkor legyenek szívesek és intézzék el, hogy előfizető lehessen a család. Ha nincs erre lehetőség, akkor kérem, hogy írják meg, hol lehet előfizetni az újságot. Azért nem tudtam előbb írni, mert az első példányt most kaptuk meg apukám testvérétől, aki Budapesten a SZÜV-ben dolgozik.

Szeretném, ha az újságban levő programok nagy része „hordozható” lenne, és ha az újság negyedévenként jelenne meg.

*A 84/1. számot elküldtük, a többi a Postán már előfizethető. Különösen meleg a szívem, ha ifjú „földim” levelét olvashatom, de a hordozhatóság ügyében vannak problémáink. Átirás vagy módosítás nélkül a programokat csak a megfelelő gépen lehet futtatni.*

**Mayer István János, Pápa,  
Béke út 6. 8500**

Szükségem lenne a HT karakterazonosító mómóriacímeire, azokra, amelyek átírásával új karaktereket lehet tervezni, például a helyett á-t stb.

*Hűséges levelezőm hosszú leveléből csak a kérését közlöm. A küldött programot köszönöm.*

**Vita Attila, Nyíregyháza,  
Tomba Mihály u. 11. 4400**

Egy általam írt programot szeretnék felajánlani Önöknek. A nyíregyházi Kossuth Lajos Gimnázium 4. osztályos tanulója vagyok. Egy éve foglalkozom a számítástechnikával, azóta élekhálok a gépekért (inkább élek). A KLTE programozó matematikai szakán szeretnék továbbtanulni. A nyelvek közül csak a BASIC-et ismerem, de azt nagyon.

*A programot köszönjük, átnézzük.*

**Lányi Tamás, Székesfehérvár,  
Vörös Hadsereg útja 92. 8000**

Mellékelten küldök Önöknek egy cikket a HT-1080Z számítógépre írt programmal. Amennyiben közölhetőnek találják, kérem, je-

lentessék meg valamelyik következő számukban.

Szeretném arra kérni Önöket, hogy ha nem kívánják leközölni, értesítsenek róla, mivel tudom, hogy a legtöbb folyóirat szerkesztősége nem őrzi meg és nem küld vissza kéziratokat.

Tíz évvel ezelőtt a Rádiótechnika szerkesztőségébe beküldtem egy akkor korszerű elméleti kérdést a gyakorlatban tárgyaló cikket. Fél évi türelmes várakozás után személyesen érdeklődtem a cikkemről. Előkerült egy fiókból, rajta a lektor véleménye: jó, leközölni! A mai napig nem jelent meg a cikk... Fél éve a Technika című lap szerkesztőségébe küldtem be egy másik programot a HT-1080Z gépre. A megfelelő számban, amelyben azt hittem, megjelenik, csak egy rövid közleményt találtam, amelyben kéri az olvasókat, hogy programokat printerrel kiírva küldjenek be.

*Igaza van. Mi a beküldött anyagokat, ha nem is jelentetjük meg, őrizzük, és nem küldjük vissza. Terjedelmünk korlátozott, ezért sok írás nem jelenik meg. Ezeket, főleg a programokat – ha sikerül – megpróbáljuk külön kötetben évente kiadni. A programlistát mi is printelve kérjük. Sok kézzel írt programot is kapunk, ezeket mi magunk íratjuk ki nyomtatón. Így a hátrányos helyzetben lévők problémáit megoldjuk.*

**Farkas Ferenc, Nagykanizsa,**

**Zemplén Győző u. 2/c. 8800**

Mivel hobbi alapon és szakmai téren is érdekel a számítógépek világa, szeretném a következőket megkérdezni: 1. Mi a címe annak a Stúdió '84 c. műsorban említett folyóiratnak, amely magasabb szinten foglalkozik a számítógépekkel, és hol szerezhető be. 2. Hol és mikor szerezhető be a mikroszámítógépek építéséhez szükséges alkatrészek. 3. Érdekelne, hogy a Lukács testvérek szabadalma alapján készülő gépek könyvéhez hol lehet hozzájutni. Ennek létezéséről egy napilapból értesültem, amelyben arról írnak, hogy klubtagok ennek alapján építik gépeiket.

*1. Információ – Elektronika, a Postán fizethető elő, 270 Ft/év. 2. Ha lesz kit-számítógép, egyébként a Skálában, a RAMOVILL-nál. 3. A Lukács-gépről a HCC Homelab szekciójánál érdeklődjön.*

**Szabó Attila, Keszthely,**

**Kossuth Lajos u. 10. 8360**

27 éves mozgássérült vagyok. Hobbim, kedvelt időtöltésem az elektronika. Újabban a mikroszámítógép keltette fel érdeklődésemet. Sajnos egy-két szakfolyóirat átböngészésén kívül nem tudtam egymagam előbbre jutni. Eddig nem sikerült részletesebb dokumentációhoz jutnom, amely alapján hozzákezdhetnék egy mikroszámítógép megépítéséhez.

A Klub segítségét szeretném kérni. Úgy hallottam, hogy tagjai közül már sokan építettek otthon mikroszámítógépet. Szeretnék hozzájutni egy ilyen elkészült mikroszámítógép doku-

mentációjához. Ha másképp nem, kölcsön. Ha készült fólia a panel fotózásához, akkor nagy segítség lenne számomra ennek a kölcsönzése is. Kérem a tisztelt Klubot, hogy kérésemet a lehetőségek szerint szíveskedjék teljesíteni.

*Levelét továbbítottam a HCC elnökéhez, dr. Simonyi Endréhez. Biztosan segítenek.*

**Farkas Zoltán, Érd,**

**Bagoly u. 3/b. 2030**

Két problémám van. Az első: iskolánkban működik egy Aircomp 16 számítógép, de nincs rajta INKEYS funkció, így nem tudunk rá irányítási játékprogramokat írni. Ha tudnának ilyen gépi kódú szubrutint küldeni, akkor küldjék el, de az is elég lenne, ha leírnák a billentyűzet perifériá címét, vagy a billentyűzet címzését a memóriában.

A második kérdés: ha tudnának olyan prospektust küldeni, természetesen pénzért, amiben a nyugat-európai országokban kapható számítógépek és perifériák (elsősorban hajlékonylemezek) ára szerepel DM-ben vagy USA dollárban.

*Levelét elküldtük a Personal GT illetékesének; lemezprospektust egyelőre sajnos nem tudunk küldeni.*

**Nagy Ottó, Nyíregyháza,**

**Arany János u. 29. 4400**

A március 16-án megrendezett játékprogram-pályázat döntője után a kerekasztal-beszélgetésen említette, hogy a SZÁMALK indít egy programozási tanfolyamot, amelyben felhasználják a televízió nyújtotta lehetőségeket, és amelybe szeretnék bevonni a széles néptömegeket. Arra kérem ezzel kapcsolatban, hogy tájékoztasson részletesebben a tanfolyamról, hiszen ha lehet, részt szeretnék rajta venni.

*Emlékszem beszélgetésünkre, és arra is, hogy programokat ígért. A tanfolyam januárban indul, a Rádióújság és a  $\mu$  M, de a Számítástechnika is közölni fogja, hogy kell jelentkezni.*

**Jenei Zoltán, Békés,**

**Meggy u. 16. 5630**

Időközben az Ötlet '84 is előállt egy kit-akcióval, bár az nem tartalmazza az alkatrészeket. Ezért maradok mégis az Önök kit-jénél, mert másképpen nem tudnám az alkatrészeket egyszerűen beszerezni. Remélem, a végleges kalkuláció már megvan, és meg lehet rendelni a gépet. De nem tudom, hol. Sejttem, hogy vagy a  $\mu$  M, vagy az NJSZT az illetékes. Kérem, írja meg, mennyibe fog ez kerülni, és hogy kinél kell megrendelni. Minél olcsóbb, annál nagyszerűbb!

*Július elején szerkesztem a rovatomat – ma még minden bizonytalan a kit körül. Remélem, mire a lap megjelenik, már minden biztos lesz.*

**Lendvay Balázs, Budapest,**

**Madách tér 7. 1075**

Csak egy észrevétel: egyik számukban olvastam a „Spectrum spektruma” című cikket. Mivel én is Spectrum-párti vagyok, és nekem is kilátás-

ban van egy ilyen számítógép, utánajártam ennek a klubnak. Megtudtam – és azt hiszem, ezt érdemes lenne közölni –, hogy a klub elköltözött. Új címük: Budapest XI., Kruspér u. 2-4. Ez a Műegyetem Kollégiuma. Itt is hétfőnként, este 6-tól 10-ig tartózkodnak.

*Köszönjük az észrevételeket is.*

**Juhász Katalin, Debrecen,**

**MTA ATOMKI, pf. 51. 4001**

Lehet, hogy a magyar számítástechnikai szaklapokat nem olvasom kellő rendszerességgel, de tudomásom szerint nem jelent még meg semmilyen ismertető az angol MEMOTECH cég 1983 októberében piacra került MTX 500, illetve MTX 512 típusú személyi számítógépeiről. Amennyiben szükséges, vagy értelmét látják, szívesen írok róluk ismertetőt valamilyen formában a  $\mu$  M számára.

*Várjuk a cikket.*

**Bagó Attila, Budapest,**

**Népszínház u. 38. 1081**

Nagy örömmel olvastam a hirdetések között a Diák-rovatról, ami szerintem is rendkívül észszerű kezdeményezés. A magam részéről nagyon szívesen besegítenék. Sőt ha lehet, részt is vennék a diák-szerkesztőség munkájában.

*Jöhét az írás. Aki dolgozik, az automatikusan tagja szerkesztőségünknek.*

**Rideg Péter, Szolnok,**

**Mikszáth út 17. 5000**

Szeretném, ha a lekötött programok mellett nagyobb teret kaphatna a PC-k alkalmazásának ismertetése, ami szerintem sokunknak adhatna újabb ötleteket ezen a téren.

*Várjuk a cikkeket, mi szívesen foglalkozunk az alkalmazással is!*

**Lindenberger Péter, Hatvan,**

**Horváth M. u. 5. 3000**

Egy kéréssel fordulok Önökhöz. Szeretnék házilag „barkácsolni” egy PRIMO számítógépet. A szerkesztőségtől azt kérdelem, hol lehetne meg tudni a számítógép leírását. Remélem, választ tudnak adni kérdésemre. Levelüket, ha mással nem is, az elkészülő számítógéphez programok küldésével hálálom meg.

*Levelét elküldtem a Microkey GM-hez, remélem, hogy segítséget kap a barkácsoláshoz.*

*Ezzel be is fejezem a levelekből vett idézetek bemutatását. Van egy állandóan visszatérő kérés: küldjenek olvasóink önálló cikkeket is. Önálló írásokat viszonylag keveset kapunk, így főleg a belső és külső munkatársaink írják a lapot. Ma a számítástechnika társadalmatisítása egyik legfontosabb célunk. Ennek elérését a  $\mu$  M akkor tudja segíteni, ha olvasóink is sokat írnak.*

KOVÁCS GYŐZŐ

## Kubai mikrogépek

Új személyi számítógép gyártását kezdték meg Kubában, CID-1408 néven. Az Intel 8080 mikroprocesszorral rendelkező gép operatív tára 64 kb-ot, operációs rendszere kompatibilis a CP/M-mel. Csatlakoztatható hozzá hajlékonylemezes tároló és nyomtató is.

Az Intel 8080 alapú mikroszámítógépek gyártásának már hagyománya van Kubában. 1977-ben készült el a Havannai Központi Számítástechnikai Intézetben (ICID) a CID-1400. Ezt egy szintén 8080 alapú, egykártyás mikroszámítógép követte, amely egy – különböző célorientált gépek kiépítését lehetővé tevő – moduláris rendszer alapját képezte. Ennek bázisgépe a MIKROCID-02 volt, amelyet 1980-ban vizsgáltak a szocialista országok miniszámítógép-rendszere (MSZR) elemeként, teljesítményénél fogva az SZM 50/40-1 kategóriába.

## Csak robotok

Csehszlovákiában a jelenlegi ötéves terv során 70 olyan integrált termelőműhely kifejlesztését tervezik, amelyekben az anyag és a félkésztermék szállítását teljesen automatizálják. Közülük 40 már elkészült.

Az automatizálásnak egy magasabb fokán állnak a termelést robotokkal végző rugalmas termelési rendszerek, amelyekből 1985-ig nyolcat valósítanak meg. Ezek közül egyetlenegy lesz komplex: a műhelyben a termelést a robotok teljesen önállóan, emberek nélkül végzik. A munkaszervezés úgy történik, hogy az egyetlen, nappali 8 órás műszak alatt a dolgozók előkészítik a következő 16 óra folyamatos és automatikus termelését. Ez a rendszer – amely csehszlovák és szocialista import alkatrészekből készült robotokkal működik – pár hónap múlva kezdi meg a termelést az észak-moráviai Olomouc városban, a TOS Szerszámgépgyárban.

## Teledata

Hazánkban már évek óta folynak a kísérletek a teledata rendszer meghonosításával. Ennek eredményeképpen a Számítástechnikai

Koordinációs Intézetben már működik a SZKITA nevű rendszer, melynek központi gépe egy ESZ-1015, és az ezen levő adatbázis a hozzá csatlakoztatott PROPER-család bármely tagjával lekérdézhető.

A kutatásban a SZÜV is részt vesz. Kaposvári számítóközpontjában ESZ-1022 számítógépen már fel is töltöttek egy adatbázist a Balaton környéki idegenforgalom kiszolgálását célzó információkkal: a szórakoztató programok helye és ideje, a különféle menürendek stb.

Az ORION az idén mutatta be a VTX-960 nevű teledata terminálját. A készülék műszaki paraméterei világszínvonalúak, ára viszont túlságosan magas: a rendeltetéstől függően 50–80 ezer forint.

## Számítástechnika

### a főutcán

Hazánkban az 1984-es év jelzi majd az első „főutcai” számítástechnikai boltok megjelenését. Várhatóan szinte egyidőben három intézmény is üzletet nyit, például a November 7. térnél és a Gellért térnél.

Ezek az üzletek a vásárlóhoz való közeledés jegyében nyílnak. Természetesen nemcsak számítástechnikai eszközöket (mikroszámítógépeket, perifériákat, adathordozókat) lehet majd itt vásárolni, hanem szoftvert is, és éppen az utóbbin lesz a hangsúly. Bárki arrafelé járva bemehet, kipróbálhatja a különféle gépeket és az azokhoz kínált felhasználói programokat. Így lehetősége lesz összehasonlításra, gondosabb kiválasztásra.

## A 16 bitesek

### inváziója

Még csak a tavalyi BNV-n jelentek meg az első szocialista gyártmányú, 16 bites mikroszámítógépek, a JANUS, a MIKROSZTÁR és a PROPER-16, az idei BNV-n pedig már több volt a bemutatott új 16 bites számítógép, mint a 8-bites.

A felállás a következő volt (zárójelben a gyártó cég): 8-bites gépek: Agrinfo-100 (Lignifer), Primo (Microkey), Transmic-8 (Műszertechnika GMK), TV-

Computer (VIDEOTON), azaz összesen 4 darab.

16 bites gépek: M 216 (RSZKICE FELIX), Procom 6 (SZKI), Professor (Comproject GMK), Proper 16W (SZKI), TM-16 (Műszertechnika GMK), Transmic-16 (Műszertechnika GMK), VT 16 (VIDEOTON), VT 32 (VIDEOTON), azaz összesen 8 darab.

Az utóbbi kategóriában feltűnő a Motorola 68000 mikroprocesszor gyakori alkalmazása (Professor, TM-16, Transmic-16, VT-32).

## Spectrum-

### újdonságok

Az Interface 1 lehetővé teszi max. 64 Spectrumból álló hálózat kialakítását, vagy a Spectrumhoz más perifériák csatlakoztatását, sőt modemen keresztül a telefonhálózaton át más számítógépekhez való csatlakoztatását is.

Az Interface 2 segítségével tölthetők be az új, különféle programokat tartalmazó ZX ROM minikazetták. Lehetővé teszi továbbá két botkormány csatlakoztatását is.

A végtelenített szalagot tartalmazó Microdrive egyszerűsíti és rendkívül felgyorsítja a beviteli-kiviteli műveleteket. Például egy 230 rekordot tartalmazó állomány beolvasása közönséges magnetofonról 170 másodpercig tartott, a Microdrive-ről pedig csupán 13 másodpercig.

Az árak angol fontban: az Interface 1 50, az Interface 2 20, a ROM-mikrokazetta 15, a Microdrive 50, a Microdrive-kazetta 5 fontba kerül.

## Rajzolás

### lézerrel

LÉZERGRÁF LG-1 típusjelzéssel az USA és Japán után hazánkban is gyártanak lézeres filmrajzoló berendezést. Pontossága, nagy felbontási képessége és gyorsasága következtében a berendezés kiválóan alkalmas nyomtatott áramkörök és IC maszkok filmjeinek előállítására, gépészeti és építészeti rajzdokumentációk készítésére, fényszex-észre újság formátumban is, to-

vábbá orvosi, légi és űrfelvételek képfeldolgozására.

Az alkalmazható maximális filmméret 500×600 mm, a felbontás 25 mikromilliméter, a rajzolási idő maximális filmméret esetén 8 perc. Az utóbbi paraméterhez összehasonlításként: a hagyományos fotoplotterek átlagos nyomtatott áramköri film rajzolási ideje 5–6 óra.

A berendezés tényleges alkalmazásához szükséges felhasználói programcsomagok közül elsőként a nyomtatott áramköri filmek készítésére alkalmas programok készültek el. A programcsomag a szabványosan kódolt nyomtatott áramkör leírásából (mely a vonalak kezdő- és végpontjainak koordinátáit, a vonalvastagságokat, mintázatokat és egyéb paramétereket tartalmaz) állítja elő tömörített formában a LÉZERGRÁF LG-1 közvetlen vezérlésére alkalmas vezérlő adatokat, a bittérképet.

## Rajz gép

MÜFO-PLOT néven a Központi Fizikai Kutatóintézet rajzgépet fejlesztett ki, melyet az iklandi Ipari Műszergyár gyárt. A hazai termépaletán mind ez ideig hiánycikknek számító új periféria számítógéphez csatlakoztatva alkalmas szöveggel kiegészített diagramok, rajzok, sík- és térgörbék, táblázatok megjelenítésére. Alkalmazása révén lehetőség nyílik a számítógéppel segített tervezés, mérésadatgyűjtés és adatfeldolgozás eredményeinek közvetlen, szemléletes ábrázolására, archiválására. Szerecsésen ötvözi össze az alfanumerikus nyomtatók szöveg megjelenítő képességét a nagy felbontású digitális rajzgépek grafikus szolgáltatásaival.

A megfelelő célorientált szimbólumkészlet kialakítását követően rendkívül sokrétű feladat ellátására alkalmas például a tudományos életben, a kutatásban, a gyógyászatban, a térképészetben, a mérnöki alkalmazásokban. A MÜFO-PLOT két írótollas, a közönséges Parker golyóstollbetéttel működik. Felbontóképessége 0,1 mm, visszaállási pontossága jobb, mint 0,02 mm. Sebessége minden irányban 5 cm/s. 336–400 mm széles lepelrelóra rajzol.

## LÉPÉSRŐL LÉPÉSRE

**Cikksorozatunk első részében bemutattuk a minimax algoritmust, amelynek segítségével elvileg minden véges játékra olyan program készíthető, amely a játékot hibátlanul játssza, tehát minden állásból a lehető legjobb eredményt hozza ki. Ott hagytuk abba, hogy ha ez ilyen egyszerű, akkor miért nem így csináljuk?**

### Kombinatorikai robbanás

Valóban, egyszerűbb játékokra a minimax algoritmus a játék teljes fáján ténylegesen végrehajtható; az ilyen játékokról nincs is több mondanivalónk. A sakk azonban ennél sokkal bonyolultabb. A meghökkenítő eredmény megér egy kis játékos számolást, már csak azért is, mert vegytisztán mutatja azt a jelenséget, amit a számítástechnikában kombinatorikai robbanásnak nevezünk, és ami gyakran a legsebbebb számítógépes eszméket is gyakorlatilag kivitelezhetlenné teszi.

Tegyük fel, hogy a Föld minden atomja egy-egy számítógép, ami csak sakkállásokat vizsgál, és egyet-egyét annyi idő alatt dolgoz fel, amennyi idő alatt a fény egy milliméter utat megtesz. Számoljuk ki, hogy egy ilyen szupergéppel mennyi ideig tartana a sakkjáték fájának kiemlézése.

Először is becsüljük meg, hogy legalább hány pontból áll a sakkjáték fája. Tapasztalat szerint egy átlagos sakkállásban kb. 30–40 lépés húzható. Vegyünk csak húszat, és tegyük fel, hogy egy sakkparti csak harminc lépésig tart. Ez 30 világos és 30 sötét lépést jelent, és mindegyiknél a fa csúcsainak száma meghússzorozódik. Tehát a fa legalább  $20^{60}$  csúcsból áll.

Két atom távolsága általában jóval  $10^{-7}$  mm fölött van. A Föld sugara kisebb 6400 kilométernél, így térfogata kisebb  $10^{30}$  köbmilliméternél, tehát a Földben jóval kevesebb, mint  $10^{51}$  atom van. A fény egy másodperc alatt  $3 \cdot 10^{12}$  mm utat tesz meg; vegyünk nagyvonalúan  $10^{13}$ -t. Elképzelt szuperszámítógépünk tehát egy másodperc alatt legfeljebb  $10^{64}$  állást tud megvizsgálni. Az átnézendő  $20^{60}$  állás összesen kb.  $10^{78}$  darab, tehát ennek feldolgozásához legalább  $10^{14}$  másodperc

szükséges. Ennyi másodperc több százézer évnél felel meg, azaz elképzelt (ugyancsak fejlett) számítógépünknek is több százézer évre lenne szüksége a sakkjáték fájának kiemlézéséhez. Pedig mindent igen durván, alulról becsültünk!

### A és B típusú stratégiák

Látjuk, hogy a sakkjáték teljes fájának kiemlézése reménytelen feladat, és ezen a technikai fejlődés sem segíthet. Így viszont a számítógépes sakkozás problémáját már egész másképpen szemléljük: nyilvánvalóvá vált, hogy a számítógéppel is a játék fájának csak egy elenyészően kicsi darabját tudjuk kiértékelni, és legfeljebb azt a feladatot tűzhetjük ki magunknak, hogy ezt a darabot lehetőleg okosan válasszuk meg, és lehetőleg precízen értékeljük ki.

A minimax algoritmust ettől még nem kell elvetnünk, csak át kell értelmeznünk a megváltozott körülményekre. A minimax algoritmus a fa egy kis darabjára is alkalmazható, csak tudni kellene, hogy a vizsgált farész végpontjaihoz milyen értékeket rendelünk. Ezek az értékek most már nemcsak 1,0 és -1 lehetnek, hiszen ha el tudnánk dönteni, hogy melyik álláshoz melyiket rendeljük, akkor nem is lenne szükségünk a fa vizsgálatára. Ebben az esetben egyszerűen megnéznénk, hogy melyik lépésünk után kell az ellenfélnek -1-et rendelnie a kialakult álláshoz, és ezt a lépést választanánk.

A minimax algoritmus minden nehézség nélkül elvégezhető akkor is, ha a vizsgált fa végpontjain tetszőleges értékek állnak. Így például a minimax algoritmus segítségével (megint csak elvileg) olyan játékokra is tökéletes program írható, amelyekben nem egyszerűen csak nyerni vagy veszteni lehet, hanem különféle mértékben lehet

nyerni vagy veszteni. Az algoritmuson semmit sem kell változtatni ahhoz, hogy erre az esetre is érvényes maradjon. A minimax algoritmus ebben az esetben megmondja azt, hogy melyik lépéssel tudjuk a legtöbbet nyerni (vagy a legkevésbé veszíteni), feltéve, hogy az ellenfél a lehető legjobban játszik.

Ha nem is tudjuk egy állásról biztosan megmondani, hogy az nyerő, döntetlen vagy vesztes, azt azért meg tudjuk becsülni, hogy az adott állás előnyös-e világosnak vagy sem, és hogy mennyire. Ezt a becsülő eljárást kiértékelő függvénynek nevezzük.

Egy egyszerű lehetőség a kiértékelő függvény kialakítására az, hogy csak az anyagi egyensúlyt számláljuk össze. A kiértékelő függvény meghatározásának módjait sorozatunkban később még visszatérünk; bizonyos értelemben ez a sakkprogram „lelke”. A most leírt „abszolút anyagi” kiértékelő függvényt azonban ne becsüljük le: ha egy program csak annyit csinál, hogy a játék fáját három lépésig előre teljesen felállítja, és elvégzi rajta a minimax algoritmust szerint a roppant egyszerű kiértékelő függvény szerint, nos, ez a program már nem fog nagyon gyengén játszani, és időnként meglepően jó lépéseket fog produkálni.

A minimax algoritmus azt mindenestre biztosítja, hogy a segítségével választott lépés a fa vizsgált darabjában a kiértékelő függvény szerint vett elérhető maximumot el is érje. Így például az előző bekezdésben leírt „program” azt garantálja, hogy mindig olyat lépjen, amivel három lépés múlva anyagiilag a lehető legjobban fog állni. Az persze előfordulhat, hogy olyat lép a program, hogy négy lépés múlva védhetetlenül hatalmas anyagot veszítsen – ezt nem láthatta előre, hiszen ez már kívül esik azon a világon, amit a program megvizsgált. Az is előfordulhat, hogy az ellenfél a következőkben védhetetlen matt-támadáshoz vagy vezérbevitelhez jut – ezt ugyanis a kiértékelő függvény nem vette figyelembe.

A sakkprogramozás egyik fő problémája tehát a kiértékelő

függvény alkalmas megválasztása. A másik az, hogy milyen darabját értékeljük ki a játék fájának. Ha egy adott gépen adott idő alatt ki akarjuk választani a következő lépést, akkor ezzel az is adott, hogy a kielemezendő fa legfeljebb hány csúcsból állhat. A kérdés az, hogy mi jobb: egy „szélesebb” fát sekélyebben, vagy egy „keskenyebb” fát mélyebben kiértékelni.

A számítógép-tudomány nagy klasszikusa, Shannon már 1950-ben foglalkozott ezzel a problémával, és a két lehetőségnek megfelelően kétféle alternatív stratégiát ajánlott. A típusú stratégiája a teljes fa vizsgálatát írja elő bizonyos mélységig (például 3 lépésre), B típusú stratégiája pedig minden pontban csak néhány, legjobbnak ígérkező lépést vizsgál (például az őt, a kiértékelő függvény szerint legjobbnak ígérkező lépésre vizsgálja mindegyiknél az őt legjobbnak ígérkező választ, ezekre ismét az őt-öt legjobbat stb.).

Az A típusú stratégia előnye, hogy a vizsgált mélységig biztosan nem hagy ki egyetlen lehetőséget sem. Ennek ára az, hogy nyilvánvalóan érdektelen lépések tömelegét vizsgálja meg, sőt idejének legnagyobb részét ilyen változatok elemzésével tölti.

A B típusú stratégia esetleg teljesen figyelmen kívül hagy fontos változatokat, de az adott idő alatt sokkal mélyebbre tud előre nézni, így gyakran lényegesen hosszabb kombinációkat is megtalál. A B típusú stratégia sokkal ígéretesebbnek látszik, mert nem néz meg minden buta lépést, mint ahogy az emberi játékos sem, de az elkészült sakkprogramok azt mutatták, hogy a „robosztus” A típusú stratégiák általában eredményesebbnek bizonyultak. Ennek okait a későbbiekben még elemezni fogjuk.

Az A típusú stratégiák külön előnye, hogy esetükben a minimax algoritmus gyakorlati végrehajtására számos ügyes matematikai trükköt sikerült találni, amelyek a számolás egy jelentős részét megtakarítják. Ilyen eljárásokat mutatunk be a sorozat következő részében.

MÉRŐ LÁSZLÓ

## Nemzetközi konferencia Londonban

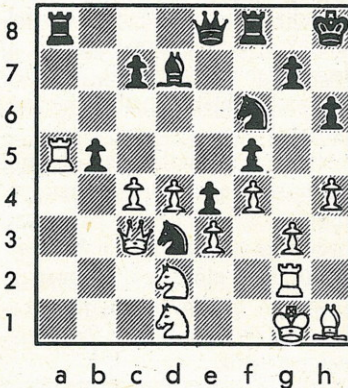
Az ez év áprilisában, Londonban megrendezett, negyedik „Advances in Computer Chess” (Fejlődés a számítógépes sakkban) konferencián elhangzott tizenegy referátum közül a szokásosnál kevesebb foglalkozott elvont, elméleti kérdésekkel, és több olyan akadt, amely bizonyos speciális célprogramok funkcióit vizsgálta. Csaknem valamennyiben felillant az ember és a gép számítás-módja közötti különbség, új gondolatokat, ötleteket sugallva a programozóknak arra, hogy milyen irányban célszerű tevékenységüket továbbfejlesztetni. E gondolatok közül ragadunk ki néhányat olvasóink számára is.

### Ember és gép – különböző feltételek mellett

Danny Kopec, az USA sakkmentere és a mesterséges intelligencia tanára, már az előző konferencián tesztet mutatott be az emberi és gépi lépésmegválasztás összehasonlításáról. A legkülönbözőbb játékerőjű sakkozók és számítógépeket választotta alanyként, és huszonnégy hadállást elemezett velük, amelyeknek a fele taktikai jellegű, a fele pozíciós, hadállást javító megoldást kíván. Kimutatta, hogy az ember a pozíciós megoldásokban nagy fölényben van, a taktikaiakat azonban a számítógép gyorsabban megtalálja; és minél nagyobb a játékerőjük, annál inkább közelítenek a gépek az emberi „gondolkodáshoz” a pozíció javításában is.

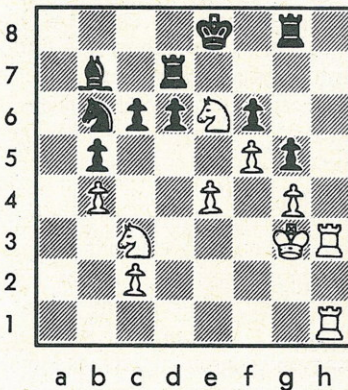
Kopec ezúttal igen érdekes módszerekkel kiterjesztette a tesztet. Új hadállásokat választott, összesen mintegy hetvenet, ismét kétféle típusúakat. A könnyebb érthetőség kedvéért egyet-egyét bemutatunk. Mindkettő igen híres játszából való.

### Bogoljubov – Aljechin



Sötét lép

### Lasker – Capablanca



Világos lép

Az előbbinek valóban fantasztikus taktikai megoldása 1. – b4 2. Bxa8, dxc3! 3. Bxe8, c2!! 4. Bxf8!, Kh7 és a c2 gyalog vezérré változik, matt-támadással fenyegetve. A másik hadállásban a nagy Lasker 1. e5! gyalogáldozattal, s 1. – dxe5-re 2. He4-gyel két huszárja domináló helyzetét kihasználva, döntő fölényre tett szert.

Talán meglepőnek tűnik, hogy Aljechin gyönyörű kombinációját a magas játékerőjű számítógépek könnyen megtalálják. De gondoljuk csak meg: sötét első, de még második lépése is eléggé plauzibilis. Az elképesztő harmadik lépésnél pedig már nem kellett messzire számolni. És akkor a számítógép csalhatatlan.

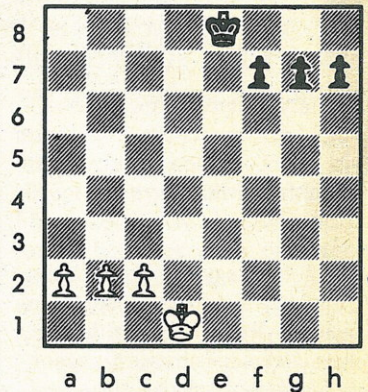
Kopec egyik érdekes ötlete volt, hogy hasonló játékerőjű elemző párokat választott, és azt vizsgálta, hogy együtt mennyire javul a telje-

sítményük. Kiderítette, hogy minden játékszinten lényegesen, s nagyobb mértékben az állást javító megoldásokban.

Ezután öt hadállás-csoport megoldására különböző időket adott. A teljesítmény – több gondolkodási időt adva – ismét a pozícionális, gyakorlatilag hosszabb távú terveknek javult erőteljesebben. Különösen érdekes volt ugyanezeknek a módszereknek az alkalmazása a számítógépeknél. (A tavalyi New York-i világbajnokságon szerepelt számítógépek zöme részt vett a tesztben.) A több idő (amit a mikrogepeknél „magasabb fokozatnak” neveznek) nem emeli olyan mértékben a játéknívót, mint az embereknél. A páros elemzést pedig a kétprocesszoros gépek működéséhez hasonlította (igazán szellemes gondolat!), és az eredmény a multiprocesszoros eljárásban rejlő nagy lehetőségeket igazolta. A témával egyébként két másik előadás is foglalkozott.

### Programozók figyelmébe: bábcsoportok

Hans Berliner professzor (Pittsburgh), a sakkprogramozás egyik kiemelkedő szakértője és Ivan Bratko, számítástechnikai kutatómérnök (Ljubljana) különböző oldalakról közelítette meg ezt a még eléggé új és kitűnő lehetőségeket nyújtó programozási „fogást”. Arról van szó, hogy egy-egy, több bából álló, logikailag összetartozó csoportot egy egységnek tekintenek. Ez azzal jár, hogy ennek a bábcsoportnak lényegesen több lépéslehetősége (-sorozata) van, mint egyetlen bábnak, de ez a szám a program számára korántsem túl magas (például az alapsoron egymás mellett álló három gyalogé, amelyek kettős lépést is tehetnek, 6<sup>3</sup>, azaz 216; ugyanakkor a lépérről lépésre hatványozottan növekvő sakk-fában kettővel kevesebb egység lépését kell tekintetbe venni, ami óriási időmegtakarítás. (De állhat egy-egy csoport hat-hét bából is.) Az alapsoron álló három-három szélső gyalog kitűnő példát nyújt erre.



### Aki indul, nyer

Ezt az állást döntetlennek tartották mindaddig, amíg Szén József magyar sakkmeister (1808–1857) be nem bizonyította, hogy a kezdő fél – ha jól játszik – nyer. A Barcza Gedeon szerkesztette Magyar Sakk történet 2. kötetében olvashatjuk, hogy Szén József Párizsban és Londonban – az akkori idők szokásainak megfelelően – fogadásokat kötött, hogy bárki ellen megnyeri ezt a játszmadállást. Nos, Berliner professzor bábcsoportokat alkalmazó programja ugyanezzel az anyaggal bármely olyan hadállásról, amelyben a két fél királyának és szélső gyalogjainak helyzete azonos, kimutatja, hogy adott esetben a legjobb játék mellett nyer, vagy veszít-e az induló fél.

### Feladványfejítő programok

E sorok írója is előadást tartott a konferencián a feladványfejítő programokról. Vázolta a speciális programok fejlődését, s igen nagy feltűnést keltett, amikor bemutatta Nemes Tihamér magyar mérnök 1949-ben és 1951-ben a Műegyetemi Közleményekben, illetve a Magyar Tudományos Akadémia Actáiban angol nyelven megjelent írásait: egy feladványfejítő, illetve a sakkozógép tervezetéről szóló tanulmányokat. Ezekről ugyanis a szakbibliográfiák egyáltalán nem



## Figyelem! Z80!

Kovács 273. Istvánnak egy napon eszébe jutott, hogy háztartásából már csak egy Univerzális Tohuvabohu hiányzik. El is ment a szaküzletbe, és fizetés után átvette az izlésesen becsomagolt árut. Hazaérkezvén rögtön kicsomagolta, gyönyörködött benne, majd szerette volna kipróbálni. Nézte a sok kiálló csövet, kart, nyomógombot, amelyek fölé számára érthetetlen jelek voltak festve. Elővette tehát a Használati Utasítást, és megdöbbenve tapasztalta, hogy abban ugyanolyan érthetetlen betűhalmaz található, mint az áhított készüléken.

Filozofikus alkat lévén, azon kezdett töprengeni, hogy miért nem tanult meg az iskolai szakköri foglalkozásokon annak idején Idegenül, de ha akkor nem is, legalább mielőtt a vásárlásra rászánta volna magát... Egy pillanatra az is eszébe ötlött, hogy talán az Illetékesek adhattak volna a berendezéshez jól lefordított Használati Utasítást, de ezt mint „gyakorlatilag elvi lehetetlen”-t, azonnal elvetette. Mit tegyen most? Talán a Szomszéd, az tud Idegenül. De ha az sem?

Helyettesítsük most Kovács úr helyébe a Felhasználót, az Univerzális Tohuvabohu helyébe a Z80-as mikroprocesszor családot, az Idegen helyébe az angolt, németet, orosz, a Használati Utasítás helyébe a katalógusokat, felhasználói kézikönyveket, a Szomszéd helyébe pedig – ha nincs jobb –, magunkat. Hozzátevé azt, hogy ebben az esetben a felhasználónak sokszor nehezebb a leíráshoz hozzájutnia, mint december 6-án banánnhoz.

Valószínűleg ehhez hasonló megfontolások vezették az Ipari Informatikai Központot, amikor elhatározta a Z80-as sorozat elkészítését és kibocsátását. Ez a tervezési segédlet hézagpótló a hasonló kiadványok hiányának szakadékaiban. Csak abbéli reményünket fejezhetjük ki, hogy ez csupán az első az ilyen jellegű szakmai kiadványok sorában.

A sorozat hét kötetből áll, amelyek közül az első magával a Z80 CPU-val (központi egység) foglalkozik. Megtalálható benne a CPU felépítése, szoftver-hardver jellemzői, és – bár igen szűken – szoftver-hardver alkalmazási példák is. Igen fontos vi-

szont, ahogy a felhasználó utasítás szinten is megismerkedhet a CPU működésével, és a különböző sebességű Z80 CPU-k legfontosabb jellemzőivel (a megjelenés óta már létezik egy még gyorsabb, a Z80 C is).

A második kötet két interfész áramkör, a Z80 PIO (programozható párhuzamos input-output port) és a Z80 CTC (számláló és időzítő áramkör) ismertetését adja, a CPU-hoz hasonlóan, de az alkalmazási példák sokrétűek, és sok, önálló tervezést elősegítő ötlet található benne a PIO-t illetően. A CTC-t ismertető részben ugyanez már nincs meg, pedig egyes periféria-illesztéseknél igen jól felhasználható időzítőként önmagában, vagy csak kis környezeti kiépítéssel.

A harmadik kötet a Z80 SIO (programozható soros input-output áramkör) leírását tartalmazza. Részletesen foglalkozik az egység különböző üzemmódjaival, funkcióival és programozásával. Ajánlást ad az áramkör felhasználóságára. Alkalmazási példa nem található benne, de a hatodik kötetben két, kifejezetten soros illesztési megoldás is olvasható a Z80 SIO felhasználásával.

A negyedik kötetből a Z80 COMBO (univerzális perifériaegység) áramkőről kapunk információkat. Ez az egység egy kis RAM-ot (0,25 kb-át), két időzítő egységet és egy soros input-output egységet tartalmaz. A leírás viszonylag részletesen foglalkozik az áramkör felépítésével, részarámköreinek funkcionális leírásával és programozásával. A két alkalmazási példa viszont nem tükrözi az áramkör felhasználói lehetőségeit – talán ez a rész többet is érdemelt volna.

Az ötödik kötetben a Z80 DMA (programozható, direkt adatátviteli egység), az MK 3805 óra/ram áramkör, a Z80 VCU (videojelvezérlő egység), az MK 50808 nyolccsatornás, nyolcbites analóg/digitális átalakító és az MK 5168 N-1 mikroprocesszor kompatibilis analóg/digitális átalakító leírása szerepel. A DMA egység leírásából csak az alkalmazási példák hiányoznak, pedig bizonyos külső perifériák hatékony kezelésénél, az azok közötti adatforgalom lebonyolításánál sok munkát és működési időt lehet ezzel az áramkörrel megtakarítani – igaz, ez csak nagy kiépítettségű univerzális készülék esetében elengedhetetlen, más esetekben ennél nagyobb kompromisszumra is hajlandók a tervezők és gyártók.

Az óra/ram áramkör egy rendszer-óra egység, amely célkészülékek, de számítógépegységek esetében is (például felhasználói időmérése) előnyösen alkalmazható. Leírása kielégítő, és alkalmazási példa is található rá a hatodik kötetben.

A különféle megjelenítő (display) funkciók ellátására kifejlesztett Z80 VCU ismertetője talán kissé tömör, de a kezdő felhasználó is sok ismeretet szerezhet belőle. Felhasználói példa erre az egységre ugyancsak a hatodik kötetben található. A két analóg/digitális átalakító ismertetője viszont igen szűkszavú, és egy digitális tervezéssel foglalkozó szakember számára beépítési szempontból semmit sem mond – azaz felhasználói példa nincs hozzájuk. A CPU-vezérelt általános rendszerterv ugyanis vajmi keveset mond a 105. oldalon.

A hatodik kötet alkalmazási példát, amelyről jórészt már szóltunk. Ami viszont figyelemre méltó, az egyrészt az Intel I.8086, illetve I.8088 mikroprocesszorral alapuló rendszerek Z80 perifériákkal történő felépítése, valamint a Z80 CPU alapú rendszer dinamikus RAM mezővel való ellátása konkrét tervezési ajánlásokkal. Hibájul róható fel ellenben a példák viszonylagos szűkkörűsége, és az eredeti katalógusokban szereplő példák egy az egyben való átvétele.

A hetedik kötet tulajdonképpen kiegészítője az elsőnek, hiszen a Z80 CPU utasításkészletének alfabetikus felsorolása. Könnyen, gyorsan áttekinthető, jól magyarázza az egyes utasítások hatását a végrehajtás során mind szoftver, mind hardver szinten. Lényeges hibája azonban, hogy nem tartalmaz 422 másik utasítást (például SLIA r, SLIA/IX + offset/r, LD rX, dd stb.), amelyek ugyancsak hozzátartoznak a CPU teljes működtető utasítássorához. Ezeket az érdeklődők megtalálhatják például az MC SOFT folyóirat 1982. januári számának 27. oldalán, rövid leírással együtt.

Érdekes szépséghibája a sorozatnak, hogy bár magyar nyelvű, a táblázatok, diagramok kivétel nélkül angol nyelven íródtak (másolódottak). Pedig ha már a szöveg magyar, nem valószínű, hogy ezeket ne lehetett volna lefordítani, és némely esetben láthatóbb, olvashatóbb formában közölni.

Ha mindezek rontják is a kiadvány összhatását, kézhezvételével mégis egy jól megszerkesztett sorozatot tudhatunk magunkénak.

TÖRLEY DEZSŐ

emlékeznek meg, és így szakmai körökben eddig nem tudtak róluk. Most – az előadás nyomán – fogja a nyugati szaksajtó ismertetni őket. (Lapunkban is visszatérünk rájuk.)

Az előadás fő mondanivalója, amit húsz feladványpélda támasztott alá, az emberi és gépi fejtes összehasonlítása és a fejítőprogramok gyakorlati hasznosítása volt. Ez utóbbit tekintve a leglényegesebb szempont, hogy a minden megfejtést, az előírt lépésszámon belül mattra vezető, minden folytatást kimutató programok a feladványok ellenőrzésében nyújtanak felbecsülhetetlen szolgálatot, de a szerzőket is segítik – már alkotás közben – műveik épségének ellenőrzésében (vagyis abban, hogy nincs mellékmegfejtésük vagy a témát tartalmazó változatokon kívül olyan megkerülés, amely a szerző intenciója mellett szintén célra vezet). Igen fontos a gyors megfejtés is, hiszen csak az „intelligens” – az értelmetlen kísérleteket eleve nem vizsgáló – programok képesek arra, hogy hosszabb lépésszámú feladványokat is belátható időn belül megoldjanak.

Az egyik első, kifogástalanul és kielégítő sebességgel működő programot Szálka Imre dolgozta ki a Számítástechnikai Koordinációs Intézetben, 1976-ban. Ennek az emberi fejtesével és más programokkal való összehasonlítására e sorok írója két tesztet hajtott végre, mindkét szer negyvennégy ötbábos szabadmattal. Az első teszt, amelyen Szálkát mellett két program vett részt, szemmel láthatóan erjesztőleg hatott a programozókra, mert a másodikban, 1981-ben már tizenhárom program teljesítményét sikerült összehasonlítani, s a legjobbak bámulatos fejlődést mutattak.

Az előadásban bemutatott feladványokra vonatkozóan a szerző a két legjobbnak bizonyult program (a finn *Mika Korhonen* és *Ilkka Blom* írta őket, egymástól függetlenül, Apple II számítógépre) és két mikro-sakkszámítógép, a Mark VI. és az Elite A/S (illetve „testvére”, a Prestige) fejteseit ismertette; a mikrogepek közül jelenleg csak ezek képesek a feladványokat – esetleges mellékmegfejtéseikkel együtt – megoldani, vagyis teljes egészükben kielemezni.

DR. LINDNER LÁSZLÓ

## Az OKISZ Szervezési és Számítástechnikai Vállalat figyelmébe ajánlja az IZOT 0220M2 mikroszámítógépet, a kis- és közép vállalatok irodagépesítésének korszerű eszközét

Az IZOT 0220M2 a Bolgár Népköztársaságban készül és kiforrott, korszerű konstrukciójával az irodai számítógépes feldolgozások jól bevált berendezése:

- Az íróasztallal egybeépített, formatervezett kialakítás eleget tesz a legkorszerűbb ergonomiai követelményeknek és jól beilleszkedik az Ön irodai környezetébe
- Központi egysége a széles körben elterjedt Motorola 6800 típusú, 8-bites mikroprocesszorral kompatibilis SZM 600 mikroprocesszoron alapszik
- Operatív tárkapacitása 56 Kbyte RAM és 5 Kbyte ROM, 48 Kbyte RAM opcionális tárbővítési lehetőséggel
- A közvetlen elérésű háttértárolás céljaira 2 db egyenként 250 Kbyte kapacitású hajlékony mágneslemez egység (floppy) szolgál. Egyszeres sűrűségű, egyoldalas és 8"-es lemezek cseréjével igen nagy méretű állományok is kezelhetők. Opcionálisan további két meghajtó csatlakoztatható a rendszerhez, a közvetlen háttértárolási kapacitást 1 Mbyte-ra növelve.
- A korszerű kialakítású, numerikus és funkcionális résszel is rendelkező billentyűzet a latin ABC-nek megfelelő karakterkészlettel rendelkezik
- A kezelő látómezőjében elhelyezett képernyő kellemes, zöld színű és tükrözésmentes képsővet használ. A 24 sorban megjeleníthető maximálisan 80 karakter az Ön legkényesebb igényeit is kielégíti.
- A berendezésbe beépített nyomtató a kiváló minőséget biztosító ún. margarétakeres (daisy wheel) elven működik. Így a *levegő minőségű nyomtatás* céljaira is jól használható az IZOT 0220M2. A maximálisan 132 karakter sorszélesség és az adott nyomtatási elv mellett magas, 30 kar/mp nyomtatási sebesség a legáltalánosabb nyomtatási szükségleteket is kielégíti. Opcionálisan kartonbehúzó előtét is alkalmazható. Szükség esetén külön mozaiknyomtató is csatlakoztatható.

Az IZOT 0220M2 hatékony *rendszer szoftver támogatást* nyújt az alkalmazások gyors és megbízható fejlesztéséhez, valamint a gép üzemeltetéséhez:

- MDOS operációs rendszere megfelel a Motorola MDOS 3.5-nek. Az operációs rendszer rezidens magból és igény szerint az operatív tárba kerülő ún. overlay részekből áll. Így a széles körű szolgáltatások ellenére az operációs rendszer mindössze 8 Kbyte RAM területet foglal el a tárban. A futtatandó programok kezelése és felügyelete mellett ezek a szolgáltatások a szekvenciális szervezésű file-ok használatát is biztosítják.

- Az alkalmazási programok géphasználati szempontból leghatékonyabb fejlesztését jól használható makroassembler és egy speciálisan gépközeli magas szintű nyelv, az MPL biztosítja. Ez utóbbi tulajdonságaiiban igen közel áll a széles körben ismert PL/1 nyelvhez.
- A fejlesztési idő és az elkészült alkalmazási program dokumentáltsága szempontjából leghatékonyabb nyelvi eszköz az IZOT 0220M2 COBOL rendszere. Ez igen közel áll a COBOL74 szabványhoz és egy sor igen hasznos bővítést tartalmaz. Saját file-rendszere segítségével az index-szekvenciális szervezésű file-ok kezelését is támogatja, speciális bővítésként pedig fejlett, interaktív képernyőkezelést tesz lehetővé. A nem adatfeldolgozási típusú alkalmazásokhoz opcionálisan rendelkezésre áll a FORTRAN-77 szabványnak megfelelő fordítóprogram is.
- Az operációs rendszer segédprogramjai sokrétű szolgáltatást nyújtanak. Egyrészt a szöveges információ bevitelét és a tárolt szöveg közvetlen kezelését támogatják. sorszerkesztő és teljes képernyő (ún. full screen) szerkesztő programok. A programfejlesztés során előálló bináris programmodulok közötti kapcsolat megteremtését az ún. kapcsolatszerkesztő program valósítja meg. A könyvtárkezelő program fejlett könyvtári szolgáltatásokat nyújt.

Az IZOT 0220M2 berendezés fejlett konstrukciója és magas színvonalú rendszer szoftver támogatása lehetővé tette, hogy az *irodai alkalmazás* legkülönbözőbb területein, így a könyvelésben, termelésirányításban, takarékpénztáraknál és biztosító intézményeknél egyaránt sikeresen alkalmazzák, akár *állandó, folyamatos üzemben*. A gyors használatba vételhez kész programcsomagok is feltétlenül szükségesek. Az IZOT 0220M2 mikroszámítógépre vállalatunk az alábbi *felhasználói programcsomagokat* ajánlja:

- Anyag, félkész- és késztermék nyilvántartási modul.
  - = Az adatfeldolgozás alapjául a forgalmi és készletállományok szolgálnak, folyamatos készletaktualizálás mellett utókalkulációs és főkönyvi szintű feladatokkal, elemzéseket támogató (készletérték átértékelés, elfekvő készlet, minimális készlet figyelése) és eredményadat-közlő elemekkel.
- Főkönyvi és folyószámla könyvelési modul.
  - = Az adatfeldolgozás két eleme a forgalmi és törzsadatállomány, az analitikus és szintetikus napi állapotra hozás után összesített kimutatásokat (főkönyvi ki-

vonat, mérlegadat, költségfelosztási adatok) szolgáltató elemekkel, folyószámla oldalon terhelések és jóváírások, késedelmi kamat számító, számla-egyezőség vizsgáló és egyenlegközlő információs elemekkel.

- Személyi és munkaügyi nyilvántartási modul.
  - = Alapadatként a dolgozók törzsállománya szolgál, kiegészítve a személyi és szakmai adatokkal, állandó jellegű bérszámfejtési adatokkal. A feldolgozás informatív elemei munkaügyi (FEOR számolás, szakképzettség, órabéres ill. teljesítménybéres és más szempontok szerinti) és személyi (egyedülálló, gyermekes, törzsgárdatag, nyugdíj jogosult stb.) elemzéseket adnak.
- Bérügyviteli modul.
  - = Alapadatként a dolgozók törzsadatai és a havi bértételek bérelemenkénti adatai szolgálnak. A feldolgozás informatív elemei a bérkarton, bérfizetési jegyzék, költség helyenkénti bérösszesítő és levonási jegyzék.

Ezzel korántsem merítettük ki az IZOT 0220M2 alkalmazási lehetőségeinek felsorolását. Csak megemlítjük, hogy tekintettel a Motorola Exorciser II illetve Exoterm 220-nak való megfelelésre, bővítéssel még mikroszámítógépes fejlesztő rendszerként is használható. Más számítógéppel gép-gép kapcsolat is könnyen kiépíthető.

Az OKISZ Szervezési és Számítástechnikai Vállalat *széles körű szolgáltatásokkal* segíti az IZOT 0220M2 hatékony alkalmazásba vételét:

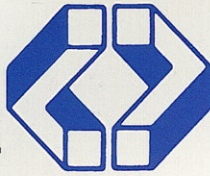
- Előnyös vételi lehetőségek
  - Rendkívül kedvező gépbérletezési konstrukció
  - Rendszeres hardver- és szoftverkövetés
  - Oktatás és támogatás a rendszer bevezetésében
  - Mintarendszerek kidolgozása és forgalmazása
  - Kulcsrakész rendszerek kidolgozása az Ön igényei szerint
  - Állandó vevőszolgálat
- További felvilágosítással, az Ön számára leginkább megfelelő feltételekkel, készséggel áll rendelkezésére az

**OKISZ** Szervezési  
és Számítástechnikai Vállalat

Budapest IX., Üllői út 47.  
1445 Budapest 8, Pf. 247.  
Telex: 22-6932. Telefon: 340-102

FEJLESZTŐ FORGALMAZÓ BÉRLETEZŐ (leasing)

**Szki**  
1015 BUDAPEST  
DONÁTI U. 35-45.



**Sci-L**  
1015 BUDAPEST  
ISKOLA U. 10.  
T.: 260-000



**Scitel**  
1015 BUDAPEST  
DONÁTI U. 35-45.



**proper 8      proper 16**

**NAGY TELJESÍTMÉNY + KIS MÉRET =  
PROFESSIONÁLIS SZEMÉLYI  
SZÁMÍTÓGÉP CSALÁD**

**MINDEN IGÉNYT KIELÉGÍTŐ ALAP ÉS ALKALMAZÓI**

**SOFTWARE**

**Széles körű szolgáltatások**

# Computerta

