



A NEUMANN JÁNOS  
SZÁMÍTÓGÉP-  
TUDOMÁNYI  
TÁRSASÁG  
LAPJA

Ára: 30 Ft

**1986**  
július







# MÁR MINDENKI TUDJA!

A megoldás:

## PRINTER BASIC

COMMODORE—64-re

MIKROSZÁMÍTÓGÉP  
MAGAZIN  
AJÁNLATA:



**AZ ELŐZMÉNYEKET MEGTALÁLJÁK:**

a PRINTER BASIC termékismertetőjét magazinunk 1986/3-as, a forgalmazókat az 1986/4-es és az 1986/5-ös számunkban.





## A NEUMANN JÁNOS SZÁMÍTÓGÉP- TUDOMÁNYI TÁRSASÁG LAPJA

A kiadvány  
a Tudományos-  
és Informatikai  
Intézet  
együttműködve készül

A szerkesztőbizottság  
vezetője:  
Kovács Győző

E számunkat  
szerkesztették:  
Bakos Tamás  
(programozástechnika)

Broczkó Péter  
(hírek)

Kovács Győző  
(levelezés)

Lindner László  
(sakkprogramozás)

Petróczy Judit  
(könyvek)

Simonyi Endre  
(klub)

Vadkerti János  
( $\mu$ programok)

Varga András  
(iskola — számítógép)

A szerkesztőség  
munkatársa:  
Kardos Zsuzsa

PR menedzser:  
Pálhalmi Vali

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

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

Terjeszti a Magyar Posta  
Előfizethető  
bármely postahivatalban,  
a kézbesítőknél,  
a Posta hírlapüzleteiben  
és a Hírlapelőfizetési  
és Lapellátási Irodánál  
(Budapest V., József nádor tér 1.  
Postacím: 1900 Budapest)  
közvetlenül  
vagy postautalványon,  
valamint átutalással  
a HELIR 215—96162  
pénzforgalmi jelzőszámra.  
Megjelenik havonta  
Példányonkénti ára 30,— Ft  
Előfizetési díj:  
egy évre 360,— Ft  
fél évre 180,— Ft



Szika Lapnyomda  
Budapest (86-3285)  
Felelős vezető:  
Csöndes Zoltán vezérigazgató

INDEX: 25629  
ISSN 0236-6088

Címképünk:  
A HT 3080C új iskolaszámítógép,  
amelyet a Híradástechnika  
Szövetkezet gyárt



### Tartalom

$\mu$ 86? $\mu$ 87?	2
Alma Materek és a számítástechnika	10
Assemblerek, cross-assemblerek	11
Adok — veszek — cserélek	21
A minőségügy közügy	23
A tús mátrixnyomtatók	24

### ISKOLA — SZÁMÍTÓGÉP

Mi van a gép belsejében?	4
Vizsgálatok a „vizsgálatok” körül	5
Eljárás a DATA POINTER felhasználására	6

### DIÁKROVAT

Fényceruza ZX—Spectrumhoz	7
Úrhajós játék	8
A bolygók bolyongása	9

### PROGRAMOZÁSTECHNIKA

BASIC és gépi kód	14
Gépi kód-szerkesztő 1 kb-ajtos ZX81-re	15

### $\mu$ PROGRAMOK

Öt gép vizsgálata egy programmal	18
Morze adóprogram	19
Négy ötlet	20
APPEND program	21
Négykarakteres PRINT rutin	22

### JÁTÉKPROGRAMOK

$\mu$ KLUB	30
PTA—4000. Rendszerismertető	34
ZX81 buszcsatlakozó	35
Ki ad magyarázatot?	36

### PIAC

Az árhaború újabb hulláma	37
---------------------------	----

### AZ OLVASÓ ÍRJA

FÓRUM	39
-------	----

### SAKKPROGRAMOZÁS

Bitek és figurák	44
------------------	----

KÖNYVEK	46
---------	----

HÍREK, ÉRDEKESSÉGEK	48
---------------------	----



# μ'86! μ'87?

*„Mennyire szerettem volna én is kísérletezni gyermekkoromban! De nem volt sem könyv, sem ember, sem iskola, aki megtanított volna rá.”*

(ÖVEGES JÓZSEF)

Sokért nem adtam volna, ha tudomány népszerűsítési példaképem, Öveges professzor ott lehetett volna a μ'86-on, és alkalmam lett volna végigkísérni a kiállításon, megmutatni a számítógép-amatőrök munkáit, leültetni és meghallgatni diákbarátaim beszámolóit, élvezni a szakma és az érdeklődő közönség találkozását. A ma diákjainak a számítógép a nagy kaland, az a szellemi kihívás, ami Öveges József gyermekkorában az érdeklődő tehetséges diáknak a fizika volt. A μ'86-on azt szerettük volna bemutatni, hogy aki ma az informatika iránt érdeklődik, az megtalálhatja a könyvet, az iskolát, az embert és nem utolsósorban a gépet is, amellyel — nem is akármilyen szinten — kísérletezhet. Ezt bizony nagyon jó lett volna Öveges professzornak megmutatni.

Amikor a Budapesti Tavasz Fesztivál vezetőivel elhatároztuk, hogy a μ'86-ot megrendezzük, akkor többen is voltak, akik örültségnek minősítették a vállalkozást. Azal érveltek, hogy az emberek, ha jó idő lesz, akkor kirándulni mennek, ha rossz, akkor pedig bolondok lesznek kimenni a vásárvárosba, ázni-fázni közelebb is tudnak. Arról nem is beszélve — mondták —, hogy a nyakunkon a BNV, ezért ha baráti alapon összejön 10~15 kiállító, akkor már boldogok lehetünk, ha meg is nézi őket néhány száz látogató, akkor biztosak lehetünk benne, hogy az elvárhatónál is nagyobb sikert értünk el. Pontosan azok a szavak csengtek vissza, amelyeket Kiss Imre a BTF igazgatója idézett első találkozásunk alkalmából, ti. valami hasonlót mondtak a Budapesti Tavasz Fesztiválra is nem-

csak az ellendrukkerek, de talán még a jóakarató szimpatizánsok is; végül a Budapesti Tavasz Fesztivál igazi siker lett. Mi nem a BNV-vel versenyző szakmai kiállítást akartunk rendezni, annál sokkal többet, egy olyan — a bevezetőben már említett — találkozót, amelyen „egy asztalhoz ültetjük” a szakma legjobb képviselőit és a nagyközönséget.

Régen mondogatom — és azért vagyok nagyon boldog, mert a μ'86 is ezt igazolta —, hogy a számítástechnikai szakvállalatoknak egyre inkább a közönség felé kell fordulni, ti. a hagyományos piacon egyre inkább növekszik a kínálat és egyre kisebb a kereslet.

Ebben a folyamatban — ha jól látom a hazai számítástechnikai helyzetet — semmi új és különös nincs, csak annyi, hogy a különböző intézmények számítástechnikai szakemberei ebben a pillanatban számítógépekkel is, szoftverrel is többé-kevésbé el vannak látva.

Az elmúlt néhány évben felfutott a számítógépgyártás, már nemcsak néhány nagyobb intézmény (VT, SZKI, SZTAKI), de kisvállalkozások (Műszertechnika Ksz., Controll Kiszövetkezet stb.) is kínálják professzionális termékeiket a piacon. Nem biztos, hogy a μ'86-on bemutatott számítógépekből a tendenciákra egyértelműen következtetni lehet, az azonban megállapítható volt, hogy mind a gyártók, mind pedig az alkalmazók többsége az IBM PC kompatibilis utat választotta, tehát professzionális számítógép környezetében létrejött az a szintű szabványosítás, amelyet elsősorban a felhasználók — nagyszámítógépes szinten — évek óta hiába óhajtottak.

Miután több, egymással kompatibilis számítógép is kapható — lehet, hogy csak képzelődtem de — olybá tűnt, a szoftverpiac is megélné. Én ezen azért nem csodálkozom, mert nem egyszer tapasztaltam, hogy különösen a „tőkeszegény” kisvállalkozások nem kezdtek nagyobb szoftvertermékek fejlesztésébe (a szellemi kapacitás egészen biztosan megvolt hozzá), mert nem is reménykedhettek, hogy egy-egy termékükből 2-3 darabnál többet képesek lennének eladni. Ma — miután a személyi számítógépek nagyobbik része egymással kompatibilis — a vállalkozó kedv is növekszik,

ami még talán a szoftvertermék-keresletet is megnövelheti. Nem egy kisvállalkozás, leányvállalat és gm ajánlotta programcsomagjait Proper 16, Variter X, Microcontrol és más IBM PC kompatibilis gépekre, vállalva nemcsak a termék üzembe állítását, de a szakemberek és nem szakemberek be tanítását is.

A kiállításon bemutatott programválaszték meglehetősen széles volt, több szövegszerkesztőt láttunk, adatbázis-kezelő rendszereket, nagyon sokféle grafikai programot, IBM PC gépekre vállalati alkalmazási programokat, különböző programnyelvek fordítóprogramjait, adatátviteli emulátorokat, különféle számításokat megoldó programokat stb. Viszonylag kevés oktatóprogramot találtam, de talán az volt a legfeltűnőbb, hogy a bemutatott programok javarésze a szakembereknek szólt, kevés öntanító jellegű terméket ajánlottak, olyat, amelyet 1-2 óra gyakorlás után azok is kezelni tudtak, akik még számítógéppel sohasem találkoztak.

Általában az volt az ember benyomása, hogy az eladók az ismert alkalmazási területekre koncentrálnak (pl. vállalati gazdálkodás) és itt remélnek nagy üzletet, és nem próbálnak különleges felhasználási feladatokat megoldani.

Szeretném ismételtelen aláhúzni, hogy mindezek ellenére a μ'86 számomra legfontosabb tanulsága az volt, hogy a kiállítók többsége a nem szakember látogatóknak próbálta meg rendszereit eladni. Én azt hiszem, hogy ma nálunk ez a fajta eladói magatartás még meglehetősen új és szokatlan, ez a fajta piaci tevékenység pedig rendkívül nehéz. Először is több „barátságos” (user friendly) alkalmazói program kell, sokkal több, mint amennyit a kiállításon láttunk. Én azt hiszem, hogy él bennük (kiállítóknak) valamiféle rossz hagyomány, amelyen még néhány rutinos cég sem tudott (talán nem is akart) változtatni. Észrevehető volt ugyanis, hogy a kiállító szakemberek egy része nem tudta egyszerű tömönatokkal, szakmai zsargon használata nélkül bemutatni és elmondani, hogy az általuk ajánlott rendszerek mit tudnak, mi az előnyük és azokat hol a legcélszerűbb alkalmazni. Ezen a standokon a látogató ezután nem állt meg, csak átsétált, míg ott, ahol jó és a



ajánlanak. A Skála Sztráda ötlete, hogy a C 64-eket összekapcsolja egy információgyűjtő rendszerbe, mint szellemes felhasználás, elfogadható — ha nem is nagyon olcsó — megoldásnak látszik. Még mindig jobb, ha a gépeket így használják, mint ha a gép kezelői a „Csillagok háborúját” vívják munkaidő alatt.

Az iskolaszámítógép pályázat nyertesei közül két gépet állítottak ki, a TV-Computert, ami a Videoton gyártmánya és az új HT számítógépet (Híradástechnika Szövetkezet). A kapott információk szerint mindkét gép iránt mind a tanárok, mind pedig a diákok igen érdeklődtek. Azt hiszem, a gyártók hasznos információkat szerezhetnek a látogatóktól, elsősorban a gép alkalmazásával kapcsolatban. Úgy gondolom, hogy a COSY sajnálhatja, hogy nem vett részt a PRO—PRIMO-val a kiállításon.

A piac egyszerű törvénye, hogy ahol nagy a kínálat, ott az eladók részben az árakkal, részben a szolgáltatásokkal próbálnak saját maguk számára előnyös piaci helyzetet teremteni. Amint ez várható volt, részben már elkezdődött az IBM kompatibilis személyi számítógépek árának csökkenése, részben az eladók inkább a nagyobb teljesítményű XT, sőt AT kompatibilis modelleket ajánlják, és a vevők ezt is keresik az alap PC típus helyett. Elsősorban a szakmai közönség fogadta jól a lokális hálózatok jelentkezését (IBM PC alapon Mikrosystem és SZKI, illetve Commodore 64 adat-előkészítő hálózat Skála-Sztráda), nem tudom, hogy az alkalmazásban való néhány éves lemaradás-e az oka, de a lokális hálózat iránt azért a keresletnek a jelenleginél sokkal nagyobbak kellene lenni.

Ma még kuriózum volt a Műszertechnika KSZ 16/32 bites számítógépe, nem tudom, hogy milyen mértékben fogja az alkalmazókat aktivizálni, lesz-e olyan sikeres, mint ma az XT vagy az AT.

Számomra nagyon érdekes volt néhány kisvállalkozás kísérlete, hogy megpróbálják a vállalatoknál levő Commodore 64-es gépeket hasznosítani. Azt hiszem ma már minden közületi vásárló látja, hogy annak idején a C 64 olcsósága volt a csábító erő, de a gép végül is alkalmatlan volt komoly termelési feladatok megoldására. Nem erre tervezték. El kell ismerni, hogy még ma is a legjobb házi számítógép — ez sem kevés —, de mosolyogni kell, amikor komoly gazdasági szakemberek arról beszélnek, hogy ilyen gépekkel fogják elkészíteni a „vállalati komplex vezetői információs” rendszert. Nyilvánvaló, hogy a vevő intézmények szaladnak a pénzük után és biztosan szívesen veszik, ha a már említett kisvállalkozások valamilyen ésszerű alkalmazási javaslatot

A találkozó fel szerettük volna mérni a társadalom informatizálása program és mozgalom eredményeit is, ezt a célt szolgálták az iskolai számítástechnika eredményeiről szóló anketok, valamint a tanár-diák alkalmazási bemutatók, amelyek a szerény várakozást sokszorosán felülmúlva, igen sok tanárt és diákot „csábítottak ki” a találkozóra.

Két vitáról is meg kell emlékezni. Az egyik a számítógépes amatőrizmus, illetve a klubok jövőjével foglalkozott. Az volt a vélemény, hogy mind a két mozgalom, ha jól szervezzük, akkor igen nagy jövő előtt áll és mindkettőnek a támogatása — a számítástechnikai társadalmi program végrehajtásához, de fogalmazhatunk úgy is, hogy az elektronizálási program sikeréhez feltétlenül szükséges. A számítógépeket építő amatőrök száma napról napra növekszik, eredményeiket a kiállítás egyik leglátogatottabb standján mutatták be. Egyes eredményeik megjelentek már az iskolaszámítógép pályázat gépeiben, egészen biztos, hogy kellő odafigyeléssel — mint annak idején a rádió-amatőrizmus — komoly segítséget adhatnak az iparnak és főleg az alkalmazói szférának is.

A klubokat mint mozgalmat érdemes tovább szervezni, ez volt a vitán részt vevők egységes véleménye. Ami a klubok céljait illeti, sokan mondták, hogy a klubmozgalom lehet az iskolán kívüli informatikai oktatás alapszervezete, például oly módon — mondom én is —, hogy például erre a mozgalomra épülhet az informatikai távoktatás. A klubok bonyolíthatják a konzultációkat, a beszámoltatásokat, esetleg a vizsgákat is. Ez persze azt is jelenti, hogy a szervezet költségeinek egy részét az állami költségvetésből kellene finanszírozni.

Egy másik vélemény szerint — Népmű-

velési Intézet — a klubok célszerűen kisvállalkozásokká alakulhatnak át, oktatnának is, vállalkoznának is, és ebből teremtenék meg a fenntartásukhoz szükséges anyagi alapokat.

A vita nem döntött el semmit, pontosabban azt kifejezte, hogy a klubmozgalmat fejleszteni kell, nem szabad hagyni, hogy elaludjon. Nem volna célszerű az sem, ha mereven és bürokratikus irányításon, meg kell tartani a kezdeményezőkézséget és azt a sokszínű tevékenységet, ami ma a legtöbb klub működését jellemzi.

Nehéz abbahagyni a µ'86-ról készült beszámolót, sokat lehetne még írni a számítógépes zenei bemutatóról, a számítástechnika-történeti kiállításról, a µ'86 vendégeiről, sikeres és kevésbé sikerült programokról.

Az NJSZT Ügyvezető Elnöksége, a kiállítás résztvevői, a tevékenységükre figyelő intézmények és főhatóságok általában pozitívan értékelték a találkozót, arra bátorítanak bennünket, hogy legyen folytatás, jövőre is legyen µ'87.

Azon gondolkozunk, hogy nem volna-e érdemes — az idej érdeklődést látva — a megkezdett úton eggyel továbblépni. Megtartani a találkozó jelleget, de emellett meg kellene szervezni néhány tudományos egyesülettel, a Hungexpoval és a Budapesti Tavasz Fesztivál vezetőivel együttműködve a szocialista országok tavaszi elektronikai szakkiallítását. Az érdeklődés megvan, a szakértelem biztosítható, a vásári tapasztalatot a BNV szervezői hozzák, talán a döntésre sem kell sokat várni.

Én azt hiszem, hogy ebben az irányban kellene továbblépni, és akkor a találkozók nemcsak fontos társadalmi funkciót töltenének be, hanem a magyar, de a szocialista országok elektronizálásának az ügyét is szolgálnák, és ezzel együtt az elektronika fejlődéséhez annyira fontos nemzetközi együttműködés továbbfejlődését is biztosítanák.

KOVÁCS GYŐZŐ



## Mi van a gép belsejében?

1960-ban az akkor három éve működő kibernetikai szakkörünk két vezető egyénisége — ma neves fizikusok — megépítette a szakkör második játékgépét, a Csodamalmot. A többiek kíváncsian nézték a bemutatót. A 35 jelfogót tartalmazó játékgép az egyszerű malom, vagy ahogy Angliában ismerik, a Nulla és kereszt játékot játszotta élő ellenféllel. A cél három, egy egyenesbe eső mező elfoglalása volt a 3 × 3-as játéktéren. A játékot jó taktikával mindig legalább döntetlenre lehet vinni. A nézők és játékosok csakhamar tapasztalták, hogy a gép jó taktikával játszik, mert vagy kikaptak, vagy legfeljebb döntetlent tudtak elérni.

Az egyik fiúnak az az ötlete támadt, hogy megpróbálja becsapni a gépet. A játékszabályokat áthágva, egyszerre két helyet foglalt el, két gombot nyomott le a gépen. Az egész nézősereg, a gép két tervezőjét és építőjét is beleértve, érdeklődve várta, hogy mit válaszol a gép, mert erre az esetre a tervezők nem gondoltak. A gép elkezdett a szokott módon ketyegni — az egyes jelfogók meghúzását külön-külön is hallani lehetett —, majd eloltotta az élő játékos szabálytalanul kigyújtott két zöld égőjét, és ugyanabban a sorban, szintén szabálytalanul, kigyújtotta a maga három piros égőjét, az eredmény táblán pedig kiírta: „Győztem!”.

Ehhez hasonló élmények adtak erőt és kitartást a 60-as években sok diákomnak ahhoz, hogy tíznél több olyan szellemes játékgépet építsenek jelfogók felhasználásával, amelyek méltán nevezhetők a mai robotok elődeinek. Több gép részletes leírása megjelent a Tankönyvkiadó által 1970-ben

és 1972-ben kiadott Néhány kibernetikai játékgép című könyvben.

Hála a ma már hároméves középiskolai és az idén megindult általános iskolai számítógépes programnak, az ország egész ifjúsága részesül a számítógéppel való személyes találkozás élményében. A számítógépet faggatva annak olyan képességeire is rájönnek, amelyek nincsenek leírva a kezelési útmutatóban, sőt amelyekre talán még a tervezők sem gondoltak. A jó műszaki érzékkel rendelkező fiatalokban pedig felbúrad a kíváncsiság, hogy hogyan működik a gép, mi van a belsejében, hogyan lehetne ilyet építeni?

A számítógép műszaki vonatkozásaival, a hardverrel való foglalkozásnak kezdetben több akadálya volt. A programozás önmagában is elegendő munkát adott tanárnak és diáknak. Sok iskolában csak egy gép volt, amit joggal féltettek, hiszen az egyéves garancia csak a fel nem nyitott gépre szólt.

Ma már ezek az akadályok megszűntek, sőt új szempontok is sürgetik a hardverrel való ismerkedést. Az új iskola-számítógépek többet tudnak, mint az előzőek. Finomabb a grafikájuk, sok mindent lehet rájuk csatlakoztatni: nyomtatót, hajlékonylemezt, botkormányt stb. Féltő, hogy a korábbi gépek háttérbe szorulnak, pedig műszaki szempontból még sokat tanulhatnánk segítségükkel. Kezdjük el tehát minél hamarabb!

Miben tudok segíteni és mit tudok ajánlani azoknak a diákoknak és tanároknak, akik most kezdenek foglalkozni a számítógép hardverével? Kezdőknek olcsó és jó tankönyvek állnak rendelkezésre. Például



Zóránd Tibor: Gyakorlati feladatok a technika órákon; Technika A; Technika C; Technika D III. o. és Technika D IV. o. A Z80-as mikroprocesszor assemblerével való ismerkedésre akár Sztróky Kálmán hasonló című könyvét, akár az OPI kiadásában megjelent, Számítógép a fizikatanításban című könyvet ajánlhatom.

Ezután jöhetnek az egyszerűbb csatlakoztatások, a többit pedig a mindennapi élet szükségletei hozzák magukkal. Szükségessé vált például az iskolaszámítógép csatlakoztatása a meglévő nyomtatóhoz. Más iskola egy évig is járt utána, mire végre nyomtatni tudott. Mi nekifogtunk, és megcsináltuk.

Hasznos, szükséges és érdekes is az iskolákban akár a technikaórákon, akár szakköri keretben a számítógép műszaki felépítésének megismerése. Ha minden apró cseprő műszaki hiba esetén a szervizhez fordulunk garanciaidőn túl is, akkor sokba kerül a számítógép fenntartása, és sok idő is kárba veszik, amíg ott van a gépünk. Az otthon végzett javítás idő- és költségmegtakarítást jelent. De ennél is fontosabb, hogy diákjaink ügyesednek és tapasztalnak.

Az egyre szaporodó gépek gyártása, fenntartása sok szakembert igényel. Szinte létkérdés ma minden ország számára, hogy az elektronikában a kor színvonalán legyen és maradjon. A nálunk is folyamatban levő elektronikai programhoz nyújtunk nagy segítséget, ha fiataljaink elektronika iránti érdeklődését felkeltjük, és állandóan ébren is tartjuk.

A 20-as években a rádió gyors fejlődése sokat köszönhetett az amatőröknek, akik a rádióval kedvtelésből foglalkoztak. Vezetésük minél több diákunkat úgy, hogy kedvtelésből is foglalkozzon a számítógéppel. Ez nemcsak a közösség szempontjából lesz hasznos, hanem az egyén saját életét is gazdagítja.

KOVÁCS MIHÁLY  
tanár





# Vizsgálatok a „vizsgálatok” körül

A különböző vizsgálatok a programok leggyakoribb eljárásai közé tartoznak. Legtöbbször nem helyettesíthetők, elkerülhetetlenek, és sokszor lassítják kritikusan a programot. Fontos tehát a lehetséges gyorsítási fogások használata.

A probléma általánosan úgy vetődik fel, hogy adott  $n$  db logikai állítás:  $A_1, A_2, \dots, A_n$ , és ezek kiértékelése után a program különböző ágakon folytatódik. Jelölje  $i$  ( $0 \leq i \leq n$ ) az igaz állítások számát. A leggyakrabban előforduló helyzetek a következők:

- I.  $i = n$  minden állítás igaz
- II.  $i = 0$  minden állítás hamis
- III.  $i = k$  ( $1 \leq k < n$ ) pontosan  $k$  igaz állítás van
- IV.  $i \geq k$  legalább  $k$  igaz állítás van
- V.  $i \leq k$  legfeljebb  $k$  igaz állítás van

Például ha egy  $p$  természetes számról el akarjuk dönteni, hogy prím-e vagy sem, akkor 2-től  $\sqrt{p}$ -ig minden prímmel és  $p$ -vel oszthatósági vizsgálatot kell végeznünk. Ez I. vagy II. típusú feladat. Ha viszont „póker” programunk közben kell eldönteni, hogy egy leosztásban (öt lap) van-e négy egyforma figura, akkor a III. esettel állunk szemben. Itt  $n = 10$  és  $i = 3$ .

A fenti öt esetet általánosan megoldja az alábbi programrészlet:

```
5 I=0
10 IF A1 THEN I=I+1
20 IF A2 THEN I=I+1
.....
10n IF An THEN I=I+1
```

és most  $I$  kiértékelésével a kívánt mederbe terelhetjük a programot.

Minden egyszerűnek tűnik, pedig a problémák csak most következnek. Ez a program mindig  $n$ -szer vizsgál, holott lehetséges, hogy már egy hamis érték is feleslegessé teszi az összes hátralevő vizsgálatot. Ha  $p$ -nek találtunk egy osztót, akkor már nem prím, és póker sem lesz olyan leosztásban, ahol már van három különböző figura!

## Minden igaz, vagy minden hamis

Ez az első két eset gyakran előfordul, de nehezen gyorsítható. Folytatódjon az „IGAZ” ágon a program, ha minden állítás igaz, és a „HAMIS” ágon, ha nem mindegyik igaz.

```
10 FOR J=1 TO N:IF A; THEN 30
20 "HAMIS"
30 NEXT: "IGAZ"
```

Ez a program egyetlen felesleges vizsgálatot sem végez, a legelső hamis állításra leáll. A helyzet teljesen hasonló a II. típusnál is.

Van valami azonban ezeknél az eseteknél is, amellyel időt takaríthatunk meg. Tekintsük a következő programrészletet:

```
10 INPUT "KÍSÉRLETEK SZÁMA"; N
20 FOR J=1 TO N
30 A=INT(100*RND)0(+1):B=INT(100*RND)0(+1):C=INT(100*RND)0(+1)
40 IF A+B>C AND A+C>B AND B+C>A THEN 60
50 ?A; B; C;"-BŐL NEM SZERKESZTHETŐ HÁROMSZÖG!": NEXT: STOP
60 ?A; B; C;"-BŐL SZERKESZTHETŐ HÁROMSZÖG!": NEXT: STOP
```

A program véletlenszerűen választott számokról dönti el, hogy lehetnek-e egy háromszög oldalai; ezer kísérlethez 47 másodpercre van szüksége.

Ha a 40-es sor helyett  
40 IF A+B<C THEN 50  
42 IF A+C<B THEN 50  
44 IF B+C>A THEN 60  
sorokat írjuk, akkor a futási idő 5 másodperccel csökken. Sok kicsi sokra megy!

## Vadászat nem szintiszta igazságokra

Most a II., IV. és V. típusú vizsgálatokkal foglalkozunk. Adott az  $A_1, A_2, \dots, A_n$  logikai állítás, és folytatódjon a program az „IGAZ” ágon, ha közülük pontosan  $k$  ( $1 \leq k < n$ ) igaz, és a „HAMIS” ágon különben. Tegyük fel továbbá, hogy minden állítás  $\frac{1}{2}$  valószínűséggel igaz vagy hamis. Nézzük a következő konkrét programot:

```
10 INPUT "ÁLLÍTÁSOK, IGAZ Á. SZÁMA"; N,K: DIM C(N)
20 FOR J=1 TO N:C(J)=INT(N*RND)0(+1):NEXT
30 ?"A TÖMB TELE!"
40 I=0:FOR J=1 TO N:IF C(J)2=INT)C(J)2 THEN I=I+1
50 NEXT:IF I=K THEN ?"IGAZ":STOP
60 ?"HAMIS":STOP
```

A program egy véletlen tömbben számolja meg a páros számokat, és ha ez a szám az előre megadott  $k$ , akkor „IGAZ”, különben „HAMIS” kiírással megáll. A futási idő csak  $N$ -től függ, az 1. táblázat szerint.

Az idő egyenesen arányos a  $C(N)$  tömb nagyságával. Hogyan gyorsítható ez a vizsgálat? Nyilván úgy, hogy ha  $I$  már nagyobb mint  $K$ , akkor ne folytassuk. Helyettesítsük a 40-et a

```
40 I=0:FOR J=1 TO N:IF C(J)2=INT)C(J)2 THEN I=I+1: IF I>K THEN 60
```

sorral. A várt gyorsulás nem maradt el, ezt mutatja a 2. táblázat.

n	t(s)
10	0,183
100	1,833
1000	18,492

1. táblázat

K \ N	10	100	1000
1	0,097	0,088	0,116
5	0,191	0,243	0,273
8	0,192!	0,348	0,353
50	—	1,826	2,021
80	—	1,873!	3,206
500	—	—	18,830
800	—	—	19,183!

2. táblázat

K \ N	10	100	1000
1	0,088	0,078	0,109
5	0,185	0,235	0,243
8	0,191!!	0,338	0,347
50	—	1,741	1,963
80	—	1,746!	2,913
500	—	—	17,781
800	—	—	18,151!!

3. táblázat

Módszer	t(s)
I.	13,33
II.	10,02
III.	9,96
IV.	4,55

4. táblázat

Az idő most  $N$ -től kevésbé,  $K$ -től viszont annál jobban függ. A látható rendet csak azok a  $K-N$  párok bontják meg, amelyeknél  $K$  közel esik  $N$ -hez, és így a ciklusnak vége lesz, mielőtt találna  $K$  páros számot. Ez reális, hiszen nem várható el, hogy például 100 véletlen szám között legyen 80 páros!

A vizsgálat tehát gyorsult a kis  $K$  számoknál, lassult viszont a nagyoknál. Hogyan lehetne a javulást általánossá tenni?

A 40-es sor módosításával minden páros szám esetén egy új vizsgálatot iktattunk be, és ez csak viszonylag kis  $K$  számok esetén fizetődik ki. Ráadásul, ha  $K \approx N/2$  körül van, vagy annál is nagyobb, akkor jó ideig felesleges is az  $I > K$  vizsgálat. Azt kellene megoldani, hogy csak akkor vizsgáljuk  $I$  és  $K$  viszonyát, amikor már „érdemes”. Ez egy valószínűségszámítási feladat! Adott  $N$  logikai állítás, ezek mindegyike  $\frac{1}{2}$  valószínűséggel igaz vagy hamis. Mennyi annak a valószínűsége, hogy a  $J$ -edik lépésben megtaláltuk a  $K+1$ -edik igaz állítást? Ekkor ugyanis a további vizsgálatok feleslegesek. A probléma megoldása:

$$p(J,K) = \frac{\binom{J-1}{K}}{2^J} \quad 1 \leq K < N \text{ és } K < J \leq N$$

Gondoljuk ugyanis meg: ahhoz, hogy a  $J$ -edik lépésre legyen  $K+1$  „igaz”, az kell, hogy az előző  $J-1$ -ben pontosan  $K$  legyen,



## HT-1080Z

### Eljárás a DATA POINTER felhasználására

és „igaz” legyen a J-edik is. A kapott képlet persze túl bonyolult ahhoz, hogy egy programon belül számolgassuk — amit nyerénként a révén, azt elvesztenék a vámon —, de nekünk elég a  $p(J,K)$  valószínűség maximumhelyét megkeresni. Ilyen kettő is van:  $J=2K$  és  $J=2K+1$ . Már a  $J=K+1$ -től kezdenek „összejönni” a  $K+1$  darab igaz állítások, aztán a valószínűség növekszik, majd a  $J=2K+1$  után megint csökken. Az  $I>K$  vizsgálatokat érdemes tehát csak a  $K$ -adik és  $2K$ -adik lépés között elkezdeni.

Módosítsuk a programot a következőképpen:

```
40 I=0:H=INT(1.5*K):FOR J=1 TO
  H:IF C(J)(2)=INT(C(J)(2)) THEN
  I=I+1
42 NEXT
44 FOR J=H+1 TO N:IF
  C(J)(2)=INT(C(J)(2)) THEN I=I+1:
  IF I>K THEN 60
```

A többi sort nem változtatjuk.

A program most csak  $\frac{3}{2}$ -ed  $K$ -tól vizsgálja az  $I>K$  relációt. Az eredmény a 3. táblázatban látható.

A futási idők tovább javultak, elsősorban nagyobb  $K$  számok esetén. A módszer hátránya, hogy értelmét veszti, ha  $K$  nagyobb az  $N$   $\frac{3}{2}$  részénél. Ekkor az  $I>K$  vizsgálatokat elkezdhethetjük már a  $K$ -adik lépésben!

Végül nézzük a negyedik módszert, amelynél nincs szükség az  $I$  változóra, tehát a vele kapcsolatos vizsgálatokra sem. Példánkban legyen  $N=4$ ,  $K=2$ , tehát adott  $A_1, A_2, A_3, A_4$  logikai állítások esetén a program az „IGAZ” ágon folytatódik, ha két állítás igaz, és a „HAMIS” ágon egyébként:

```
11 IF A1 THEN 22
12 IF A2 THEN 23
13 IF A3 THEN 24
14 GOTO "HAMIS"
22 IF A2 THEN 33
23 IF A3 THEN 34
24 IF A4 THEN "IGAZ"
25 GOTO "HAMIS"
33 IF A3 THEN "HAMIS"
34 IF A4 THEN "HAMIS"
35 GOTO "IGAZ"
```

Itt a program szerkezete számolja az igaz állításokat, tehát nincs szükség az  $I$ -re. Nagyobb  $N$ -re és  $K$ -ra persze rengeteg sort kellene leírni, de olyan nyelvekben, ahol van kiszámított GOTO vagy — és ez még jobb — GOTO (kifejezés), néhány sorban megoldható a vizsgálat.

A 4. táblázat a négy ismertetett módszer gyorsaságát hasonlítja össze egy olyan programban, ahol a feladat 100 darab totószelvény kiértékelése. Most  $N=14$ , a nyertes szelvényeknél pedig  $K \geq 10$ .

A leírt módszereknél mindig  $K$  darab igaz állítás volt az elágazás feltétele (III. típusú vizsgálat). A fentiek kevés módosítással érvényesek a „legalább  $K$ ” és „legfeljebb  $K$ ” (IV., V.) esetekre is. A programok és mérések C64 gépen, a TI belső órával készültek.

A HT hátránya, hogy a DATA beolvasás BASIC programban közvetlen utasítással nem irányítható. Az alábbi eljárás lehetővé teszi a DATA utasítássorozatok tetszés szerinti sorának beolvasását.

A RAM periféria ellenőrző blokkjában a 16 639 és a 16 640-es tárcímen (40FFH és 4100H) található a DATA POINTER, mely a READ utasítással utolsóként olvasott karakter utáni bájtra, elválasztó karakterre mutat. Ennek köszönhető, hogy a beolvasást RESTORE nélkül onnan folytatja, ahol abbahagyta. Ha a DATA POINTER-t becsapjuk, átállítjuk, akkor a következő beolvasás az általunk választott helyen kezdődik.

A feladatot megkönnyíthetjük, ha az adatokat úgy rendezzük, hogy az egyszerre beolvasandók egy sorba kerüljenek. Mivel a program írása közben a DATA sorok kezdőcímére szükségünk van, célszerű ezeket a program elejére írni, így ezek menet közben nem változnak. Ha a DATA POINTER-t a beolvasandó sor kezdőcíménél 1-gyel alacsonyabb címre állítjuk, a következő READ utasítás az általunk kiválasztott sort fogja beolvasni. (A példaprogram 70-es során beállított pointerrel a 80-as és 90-es sorban a 60-as sort olvastatjuk be.) Célszerű első adatként a beolvasandó adatok számát megadni, így biztosítható az egész sor és csak a sor beolvasása.

A példaprogram az első 5 sor tetszés szerinti soronkénti beolvasását és kiírását teszi lehetővé. Tessék kipróbálni! Kiírásnál a sor a sorszámmal kezdődik, előtte a beolvasandó adatok száma van. A programból \*-gal lehet kilépni.

A kulcs a 60-as sorban található, ide raktam a sorok kezdetét jelző címekeket. Ennek elkészítési menete a következő:

1. Beírjuk az első 5 sort.
2. Monitor üzemmódba lépünk a SYSTEM \*?/12 710-zel.
3. Begépeljük a D 42E9 parancsot, NEWLINE NEM KELL!
4. A következőket látjuk:  
42E9 FB 42 0A 88 20 35—2C 31 2C stb.  
Ebből 0A hexadecimálisan a sor sorszá-

ma (10 DEC), a 42 FB pedig a következő sor kezdetét mutatja. 88 a DATA, 20 a space stb.

5. Jegyezzük fel a 42FB-t, és X után gépeljük be D42FB-t. 42FB 0B 43 14 00 88 20 stb. adódik.

6. Ezt az eljárást folytatva, megkapjuk az első 5 sor kezdőcímét:

```
42E9H decimálisan bájtonként 66 233
42FBH decimálisan bájtonként 66 251
430BH decimálisan bájtonként 67 11
4319H decimálisan bájtonként 67 25
4331H decimálisan bájtonként 67 49
```

X és B után BASIC-üzemmódban beírhatjuk a 60-as sort. Az első 5-ös a beolvasandó adatok száma (a 90-es sorba páronként olvastatunk be), ezután az előbb számított decimális értékek következnek. A 70-es sor beírásához újra monitor üzemmódba kell mennünk és D4331-re.

4331 43 43 32 00 88 20 stb. jön ki.

A 4343H a 60-as sor kezdete, decimálisan 67 és 67, ez elé kell állítani a DATA POINTER-t, ezért kerül a 16 639-re 66. A 16 639-re az LSB kerül! A 90-es sorban beolvasott M1 és M2 értékek (ha 10-nél több, dimenzionálni kell!), a 120-as sorban K megválasztása után beállítják a DATA POINTER-t a kiválasztott sorra.

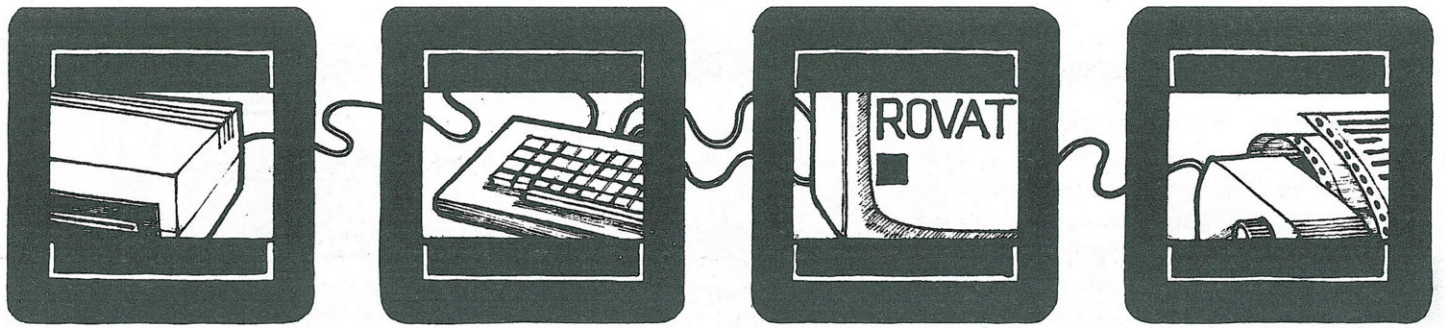
```
10 DATA 5,1,2,3,4,5
20 DATA 4,2,3,4,5
30 DATA 3,3,4,5
40 DATA 7,4,3,2,1,0,-1,-2
50 DATA 5,5,6,7,8,9
60 DATA 5,66,233,66,251,67,11,67,25,67,
  49
70 CLS:POKE 16 640,67:POKE 16 639,66
80 READ N:FOR I=1 TO N
90 READ M1(I),M2(I):NEXT I
100 INPUT " MELYIK SORT KERI-
  (1-5)";K$:IF K$="" THEN END
110 K=VAL(K$):IF K<1 OR K>5
  THEN 100
120 POKE 16 640,M1(K):POKE 16 639,
  M2(K)-1
130 READ L:FOR J=1 TO L
140 READ X:PRINT X::NEXT
  J:PRINT:GOTO 100
```

MAROSVÁRI SÁNDOR

COMMODORE 16-hoz 64 k-s memóriabővítés beépítése 1 nap alatt  
A BASIC memória ötszörösére n.ö.: 12277-ről 60671 bájtra  
Grafikus üzemmódban: 48383 bájtt BASIC+ 2 k szabad memória  
Állítható memórianevelés, kls fogyasztás  
Videoadapter beépítése C16-ba  
(csak videomagnó-csatlakozós tévékhez):  
jobb képminőség, automatikus átkapcsolás  
KÉRJEN RÉSZZLETES TÁJÉKOZTATÓT! ISKOLÁKNAK ÁRENGEDMÉNY!

EL—KA GM Tel.: 346-933 LEVÉLCÍM: 1141 Bp. Lipótvár u. 59.





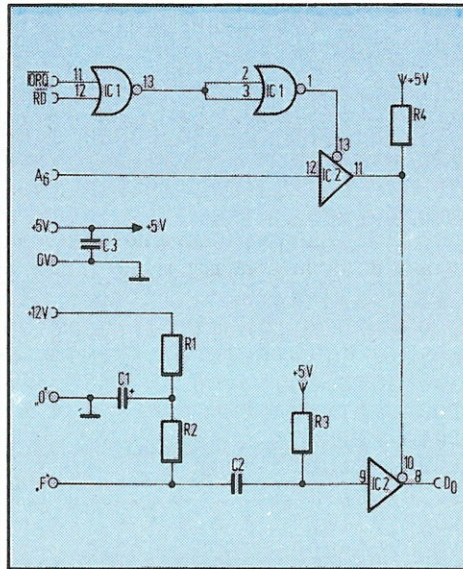
## FÉNYCERUZA ZX-SPECTRUMHOZ

Ez a kiegészítő berendezés alkalmas arra, hogy az általunk elképzelt ábrákat, rajzokat viszonylag egyszerű módon juttassuk a gép memóriájába. Ha például vaktérképet akarunk készíteni, akkor ez szoftveres úton hosszadalmas munkát igényelne, mivel az egyes pontokat koordinátáként kellene megadni. Ezzel szemben a fényceruzát a tévé képernyőjén mozgatva, bonyolultabb ábrát is könnyen tudunk készíteni.

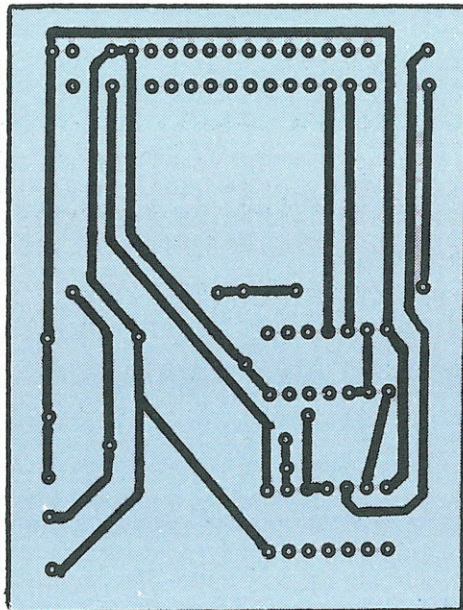
A fényceruza két részből áll. Az egyik maga a ceruza, a másik az illesztő áramkör, melynek elvi kapcsolási rajza az 1. ábrán látható. Az áramköri kapcsolásból megállapítható, hogy a Z80-as mikroprocesszor engedélyező jelére (IDRQ és az RD), valamint a címsz A<sub>6</sub>-os jelére megvizsgálja a képernyőre helyezett fototranzisztort, és a vizsgálatok eredménye az adatbusz D<sub>0</sub>-val jelölt kivezetésére kerül.

A 2. ábrán látható nyomtatási rajz alapján elkészítjük a nyomtatott áramköri lapot, amelyet egy oldalon foliózott lapra rajzolunk. Az interfész csatlakozóját a 3. ábra alapján a DS 2582-ből alakítjuk ki. A be nem rajzolt lábakat a zárlat elkerülésére többen kicsípjük. Az ábrán látható helyre az érintkezők eltávolítása után kis bakelitlapot helyezünk, az egyértelmű csatlakoztatás céljából. Ezek után a 4. ábra alapján betelepítjük a nyomtatott áramköri lapot. Vigyázzunk az alkatrészek helyes beültetésére, például a C1-es kondenzátor polaritására. Az IC-eket lehetőleg foglalatba helyezzük, ezzel megkönnyítjük az esetleges javításokat. Az elkészült illesztőegységhez már csak a fototranzisztort kell hozzákapcsolni.

A TIL 81 típusú fototranzisztor emitterét a nullával, kollektorát az F kivezetéssel kössük össze, lehetőleg hajlékony, árnyékolott, kb. 1,5 m hosszú vezetékkel. A bázisra



1. ábra



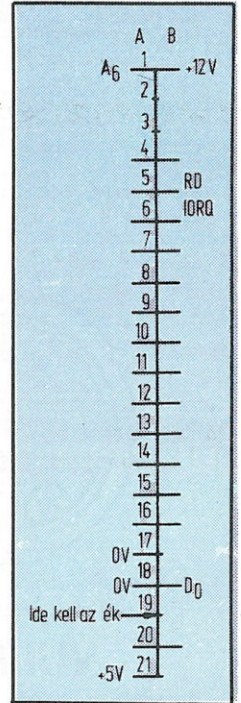
2. ábra

nincs szükségünk, ezért kicsíphetjük. A fototranzisztort egy kifogyott filctoll hegyébe helyezük.

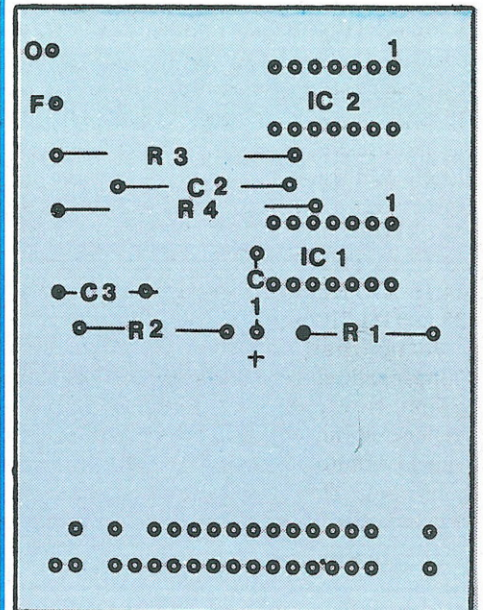
A kész illesztő áramkört a Spectrum feszültségmentes állapotában helyezük fel a gépre.

### Alkatrészjegyzék

- R1 1,2 kohm
- R2 470 kohm
- R3 470 kohm
- R4 12 kohm
- C1 22 μF tan-  
tál
- C2 10 nF
- C3 10 nF
- IC 1 74LS02
- IC 2 74LS125
- TR TIL-81  
fototranzisztor  
DS 2582  
csatlakozó



3. ábra



4. ábra

A fényceruza működéséhez szükség van szoftverre, melynek elkészítéséhez — választóriték és bélyeg ellenében — segítséget nyújtok.

KÁLI CSABA  
budapesti Piarista Gimnázium



# KRÁTER

A programmal háromdimenziós kép perspektivikusan ábrázolható síkon.

```

PRINTM KRATER
PRINTM *****
PRINTM @ LUKACS JANOS
POKE USR "3",5: POKE USR "
PRINT AT 10,5: FLASH 1, "ALL
PRINT "A MAGNOT!"
DIM B(255)
FOR X=1 TO 112: LET A=X+1
=600*(SIN(X/8)/X)+5: NEXT X
BORDER 1: PAPER 1: INK 5
CLS: PRINT AT 21,2: "A FUTA
SI IDŐ KÉ: 28 PERC"
FOR Y=10 TO 175 STEP 5
111: LET FF=Y
112: FOR X=0 TO 255
113: LET F=Y+A*(SQR((Y-87)*(Y-87)
+X*(X-128)*(X-128))+1)
114: IF B(X+1) < F THEN PLOT X,F:
PLOT X,(F+FF)/2: LET FF=F: LET
B(X+1)=F
115: NEXT X: NEXT Y
116: PRINT " VOLT"
117: PAUSE 0: BORDER 7: PAPER 7:
INK 0
    
```



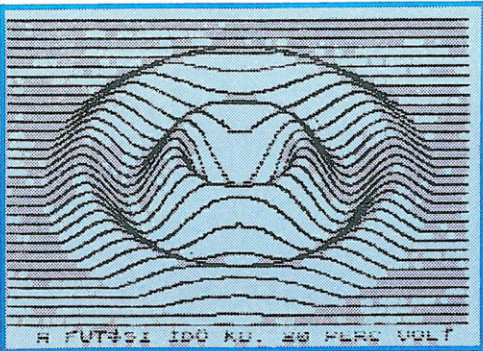
# VULKÁN

A fenti programmal megrajzolt kép — színek nélkül — lerajzolható a képernyőről.

LUKÁCS JÁNOS

```

5 CLS:REM "VULKAN" rajzolasa Lukacs Janos
10 DIMA(500),F(256)
20 FOR X=1 TO 112:A(X+1)=600*(SIN(X/8)/X)+5:NEXT X
100 FOR Y=10 TO 175 STEP 5
110 FF=Y
120 FOR X=0 TO 255
130 F=Y+A*(SQR((Y-87)*(Y-87)+X*(X-128)*(X-128))+1)
140 IF F(X+1) < F THEN SET(X,F):SET(X,(F+FF)/2):FF=F:X+1)=F
150 NEXT X:NEXT Y
160 PRINT@90,17,"VULKAN"
    
```



# Űrhajós játék

A játék lényege, hogy úgy manőverezzünk űrhajókkal, hogy se az ellenséges űrhajóknak, se a falnak ne ütközzünk. Lőni is lehet, de csak korlátozott számban, ezért takarékoskodni kell a lőszerrel. A program HT-re készült és CLOAD"U"-val, vagy egyszerűen csak CLOAD-dal lehet behívni.

```

5 REM *** READY? 32000 ***
10 CLS:FANDCM
20 CLEAR 100
40 POKE 16526,0:POKE 16527,125
50 DATA 33,1,60,17,0,60,1,0,4,237,176,33,63
60 DATA 60,17,64,0,6,16,54,32,25,16,251,201
70 FORA=0 TO 24:READ E:POKE 32000+A,E:NEXTA
90 PRINT"KAPCSOLJ AT FEL KEPERNYORE !"
95 Q1=800:GOSUB 3000
100 CLS:PRINT@3,"* KEFSZ INFORMACIOT *":INPUT A$
110 IF A$="I",1000
120 IF A$<>"N",100
125 A1$=CHR$(141)+CHR$(143)+CHR$(140)
130 REM * FCFKCGPAM *
140 B$=CHR$(136)+CHR$(174)+CHR$(142)
150 X=450:Z=287:F=30:CLS:C=14400
160 PFINT@256,STRING$(32,191);
170 PRINT@704,STRING$(32,191);
180 PRINT@X,A1$;
190 IF PEEK(G)=8,GOSUB 1200
200 IF PEEK(G)=16,GOSUB 1300
210 IF PEEK(G)=128,GOSUB 1400
240 PFINT@X," ";:Y=USR(C):L=L+1
255 IF PEEK(15363+X) <> 32,2000
260 PRINT@X,A1$;
270 D=RND(4)+1:W=RND(3)-2:IF Z+W*64 < 0,W=-W
280 IF Z+W*64+448 > 1024,W=-W
285 Z=Z+W*64
290 PRINT@Z,CHR$(191);CHR$(24);STRING$(7,26);CHR$(191);
295 IF L/8=INT(L/8),PRINT@Z+D*64,B$;
300 GOTO 190
1000 PRINT"*****"
1010 PRINT" "
1020 PRINT" U R F A J O F "
1030 PRINT" "
1040 PRINT"*****"
1050 Q1=800:GOSUB 3000:CLS
1060 PRINT" FEL ES LE A NYILAKKAL"
1070 PRINT" MOZOGHAT"
1080 PRINT" SPACE - TUZ"
1085 PRINT:PRINT" CSAK 30 LOSZFERE VAN !"
1090 PRINT"ONNEK MINEL TOBB IDFIG KELL"
1095 PRINT"MANOVEREZNI E."
1100 PRINT"EA MEHET NYOMJON COMBCT !"
1110 IF INKEY$="",1110
1115 DIM A$(6):CLS
1117 A$(1)=CHR$(141)+CHR$(143)+CHR$(140)
1118 A$(2)=CHR$(141)+CHR$(142)+CHR$(140)
1119 A$(3)=CHR$(157)+CHR$(157)+CHR$(132)
1120 A$(4)=CHR$(141)+CHR$(157)+CHR$(132)
1121 A$(5)=CHR$(153)+CHR$(157)+CHR$(132)
1122 A$(6)=CHR$(141)+CHR$(141)+CHR$(132)
1124 FORA=1 TO 6:PRINTA$(A);" ";A:PRINT:NEXT
1130 INPUT "MELYIKET VALASZTOD (1-6)";A
1140 A1$=A$(A):GOTO 130
1200 PRINT@X," ";:X=X-64:IF PEEK(15363+X) <> 32,2000
1210 PRINT@X,A1$;:RETURN
1300 PRINT@X," ";:X=X+64:IF PEEK(15363+X) <> 32,2000
1310 PRINT@X,A1$;:RETURN
1400 A=X+4:R=R-1:IF F < 1,3010
1410 FORB=1 TO 5:PRINT@A+B-1," ";:PRINT@A+B,CHR$(140);:NEXT
1430 PRINT@A+B-1," ";:RETURN
2000 REM * UTKOZES *
2010 PRINT@X,CHR$(177);CHR$(138);CHR$(160);CHR$(163);
2015 Q1=1000:GOSUB 3000
2020 CLS:GOTO 2500
2500 PRINT"ON MOST HIBAZOTT."
2510 PRINT L-(30-R);"PONTOT ERT EL."
2520 PRINT:INPUT"CPAJT MEG JATSZANI";C$
2530 IF C$<>"I",END
2540 GOTO 100
3000 FORB=1 TO Q1:NEXT:RETURN
3010 PRINT@960,"** ELFOCYOTT A LOSZER ! **";
3020 GOTO 190
    
```





A program a játék elején kérésre kiírja a játékszabályokat, s ilyenkor mód van úrhajónk tetszőleges megválasztására. Ütközés esetén a program futása megszakad, kiírja az elért pontszámot, és kérésre újra folytatja a játékot.

#### A változók jelentése

Q1 — Az üres ciklust futtató szubrutin (3000-es sor) hosszúságát szabja meg.

A1\$ — Úrhajónk képe.

B\$ — Az ellenséges úrhajók képe.

X — Úrhajónk pozíciója (az úrhajó végére mutat).

Z — A fal kirajzolásának jelenlegi pozíciója.

R — A lőszer száma.

G — Értéke 14400, ami a billentyűzetfigyelést egyszerűsíti le.

D — A szembejövő úrhajó kirajzolására utal.

W — Értékétől függően emelkedik a fal vagy süllyed.

A — Az úrhajónk előtti karakterre mutat.

B — Ciklusváltozó.

L — Lépésszám.

A programban egy gépi kódú szubrutin található, ezért a program beolvasása vagy beírása előtt az alapállapotra (READY ?) 32000-t kell begépelni.

#### A lista magyarázata

40— 70 A gépi rutint helyezik el a tárban.

130— 300 Főprogram.

190— 210 Billentyűzet figyelése.

240— 260 Letörli az úrhajó képét, majd lefuttatja a gépi szubrutint, amely balra tolja eggyel a képet.

270— 290 A D változó mutatja meg, hogy a két fal közé hová tegye a szembejövő úrhajót. A W változó -1, 0, illetve 1 értéket vehet fel, s eszerint emelkedik, süllyed vagy halad egyenesen a fal. Z értéke pedig eszerint változik -64-gyel vagy 64-gyel, vagy 0-val. Ezután a 290-es sorban az eltolás miatt üresen maradt oszlopba kirajzolja a fal folytatását.

1000—1110 A szabályok kiírása, tájékoztatás.

1115—1140 Itt lehet megválasztani az úrhajót.

1200—1210 Az úrhajó emelkedése.

1300—1310 Az úrhajó süllyedése.

1400—1430 Lövés.

2000—2540 A futás befejezése, a pontszám kiírása, esetleg a program újrafuttatása.

BACZÓ TAMÁS

budapesti Piarista Gimnázium

## A bolygók bolyongása

A bolygók az égen az állócsillagokhoz képest általában előretartó, nyugati irányba mutató mozgást végeznek. Egyes bolygók azonban időnként megállnak, sőt egy ideig visszafelé tartó, retrográd mozgást is végeznek, majd egy hurkot leírva, folytatják előretartó mozgásukat.

Ezt a mozgást mutatja be és magyarázza meg programunk, mely a szemléletesség kedvéért kissé torzítja a valóságos arányokat.

KOVÁCS MIHÁLY  
tanár



```
10 CLS:PRINT" A BOLYGOK BOLYONGASA"
15 PRINT" =====":PRINT
30 PRINT" EGY BOLYGO KERINGESI IDEJE ANNAL NAGYOBB, MINEL ME
SSZEBB VAN A NAPTOL. KEPLER III. TORVENYE SZERINT:
T1[2 / T2[2 = R1[3 / R2[3"
40 PRINT" PL. A JUPITER KOZEL 12 EV ALATT KERULI MEG A NAPOT
ES TOBB MINT 5-SZOR MESSZEBB VAN A NAPTOL, MINT A FOLD.":P
RINT
50 PRINT" 11.862[2 : 1[2 = 5.703[3 : 1[3

60 PRINT" 140.8 = 140.8"
70 IF INKEY$="" THEN 70
80 CLS:PRINT" A FOLD ES A JUPITER KERINGESE":SET(22,23)
90 FOR A=-1.6 TO 4.7 STEP.2
100 SET(20*COS(A)+22,-10*SIN(A)+23)
110 SET(104*COS(A/12)+22,-52*SIN(A/12)+23)
120 NEXT A:PRINT@776,"FOLD";:PRINT@824,"JUPITER";
122 Y=23:FOR X=22 TO 115:SET(X,Y):Y=Y-.2:NEXT X
124 PRINT@455,"NAP";:PRINT@397,"A";:PRINT@296,"5*A";
130 IF INKEY$="" THEN 130
140 CLS:PRINT" A BOLYGOK AZ ALLO CSILLAGOK KOZOTT MIND EGY
IRANYBAN HALADNAK LASSAN ELORE. IDONKENT MEGALLNAK, SOT EGY ID
EIG VISSZAFELE IS MENNEK."
150 PRINT"EZT A JELENSEGET RETROGRAD, VISSZA FELE TARTO MOZGAS
NAK HIVJUK."
155 PRINT"A JELENSEG OKA P1. A BEMUTATOTT PELDANKBAN AZ, HOGY
AMIG A FOLD EGYSZER MEGKERULI A NAPOT, A JUPITER A PALYAJANAK
CSAK AZ 1/12- ED RESZET TESZI MEG."
160 IF INKEY$="" THEN 160
500 CLS
510 A=0.75*3.14: B=3.3*3.14
515 PRINT@8*64+6,"NAP";
520 PRINT@4*64+4,"FOLD";:PRINT@780,"JUPITER";:SET(14,23)
525 PRINT@862,"A JUPITER";:PRINT@927,"AZ EGEN";
530 FOR FI=A TO B STEP .2
540 'RESET(X1+14,-Y1+23)
550 X1=1.8*7*COS(FI): Y1=7*SIN(FI)
560 SET(X1+14,-Y1+23)
570 F1=FI/12+43/24*3.14
580 'RESET(X2+14,-Y2+23)
590 X2=1.8*14*COS(FI): Y2=14*SIN(FI)
600 SET(X2+14,-Y2+23)
610 SET(2.5*X2-1.5*X1+14,-2.5*Y2+1.5*Y1+23)
620 NEXT FI: I=I+1
630 IF INKEY$="" THEN 630
640 CLS:PRINT" FIGYELJUK MEG, HOGY A FOLD, A JUPITER ES A
JUPITER KEPE A CSILLAGOK KOZOTT MINDIG EGY EGYENESEN HELYEZKED
NEK EL."
645 IF I>1 THEN 700
650 IF INKEY$="" THEN 650 ELSE 500
700 CLS:PRINT" A BOLYGOKAT OSEINK IDE ODA 'BOLYONGANI' LAT
TAK AZ EGEN. EZERT NEVEZTEK EL OKET: B O L Y G O K -
nak."
```



# Alma Materek és a számítástechnika

## A hardver éppoly fontos, mint a szoftver!

A budapesti Piarista Gimnázium egyike azon középiskoláinknak, ahol igen sikeresen folyik a számítástechnika oktatása. Különböző versenyeken elért I., II., III. helyek, dicséret oklevelek, bemutatókon értékes díjat nyert programok tanúskodnak erről. A gimnáziumi számítástechnikai oktatás irányítója Kovács Mihály tanár úr, aki immár negyed évszázada „felelős” azért, hogy a diákok ilyen kiemelkedően szerepelnek a megmérettetéseken. *(Lapzártakor kaptuk a hírt, hogy az Ötlet „2 gépnyerő” pályázatán első helyezést értek el, és egy 64 k-s HT gépet nyertek. — A szerk.)*

— Hogy lehet az, hogy egy egyházi iskola feltűnést keltő sikereket ért el a természettudományok és a számítástechnika oktatása terén?

— A mi iskolánk — amely közel 300 évével a város pesti oldalának legrégebbi gimnáziuma — kezdetől fogva nagy súlyt fektet a természettudományok és a technika oktatására. Egyik elődünk például alig egy évvel a Montgolfier testvérek után, 1783-ban Magyarországon először indított útnak léggömböt. Rendünk alapítója, Kalozancius a XVII. században a római szegények számára létrehozta az első piarista iskolát azzal a céllal, hogy a gyakorlati életre neveljen. A tananyagban a legelső pillanattól kezdve fontos szerephez jutottak az úgynevezett reál tárgyak.

— Mikor és hogyan kezdődött a számítástechnika „helyi története”?

— Egyszer — még 1958-ban — a kezembe került egy Kibernetika című folyóirat, benne a jelfogó és csöves számítógépekről szóló cikkekkel. Ennek hatására beindítottunk egy kibernetikai szakkört.

— Mi volt a szakkör lényege, célja?

— Hogy a diákok középiskolás szinten és gyakorlati módon megbarátkozzanak a számítógépek világával. Akkoriban készült el a Műszaki Egyetemen az első hazai jelfogós számítógép, melyet az egyik tanulmányi kirándulásunkon megnéztünk. Ennek köszönhetően a gyerekek egymás után készítették az egyszerűbb jelfogós automatákat. A legsikeresebb egy ószszeadó-kivonó gép volt. Készítőjéről érdemes egy kicsit bővebben mesélni. Az akkor harmadikos fiú beült egy szakköri órára, amelyet két társa tartott. (Nálam egyébként mindig diákok tartják a foglalkozásokat azóta is; én csak segítők nekik, ha erre szükségük van.) Szakkör után jön hozzám, hogy ő tudna jelfogós ószszeadógépet szerkeszteni. El nem tudtam képzelni, hogy csinálná, erre ő már scicceli is. Kértem, hogy kivonni is tudna-e? Hát, ezen még nem gondolkozott. Másnap megint felkeres, hogy a gépe már kivonni is tudna, ha... Jól van, kértem, mi kell hozzá. Tíz jelfogó. Volt a postának egy roncsstelepe, kimentem, meg-



vettem a jelfogókat. A gyerek nekifogott, és egy hónap alatt megcsinálta a gépet. És működött! Az irodalomban máig sem találkozom ilyenekkel, tehát teljesen az ő kitalálása volt. Eltelt egy idő, megint jött, hogy hallott játékautomatákról, csinálna egy kártyázógépet. Két hónapig bütykölt rajta, de a gép nem akart működni. De nem azért, mert egyedül csinálta, hanem, mert nem fogadta meg a tanácsomat, hogy harminc valahány jelfogónál már kevés tudni, hogy mit akar az ember, itt már vagy 300 drótot kell kötözni. Nagy a hibalehetőség és nehéz megtalálni a hibát. A gyerek szétszedte, elkészítette az elrendezési rajzot és így építette meg újra. És a gép ment, igen jó eredménnyel játszott élő ellenfelekkel. Akkoriban, 1959-ben nagy feltűnést keltett, több cikk jelent meg róla, a tévé is bemutatta a „Logi”-t.

— Tudja-e, mivel foglalkozik most ez a diák?

— Hogyne, sok kedves, régi diákkunkal tartjuk a kapcsolatot. Perjés Zoltán most nemzetközi hírfizikus. A gravitációelmélettel foglalkozik. Mikor Angliában dolgozott, róla neveztek el a gravitációs tér első kvázi-részecskéjét. Egyébként a következő években is mindig akadt néhány szeniális diákkunk, akik olyan jelfogós automatákat készítettek, amelyek már mérnököknek is becsületére válhattak volna. Ezeket — Csodalom, Műegér, Permutax, Nyolcas kombinett stb. — országos kiállításokon és a tévében is bemutatottuk, könyv is jelent meg róluk.

— „Csak” játékgépeket csináltak?

— Mivel gyerekekről van szó, főleg. Az 1965-ben szabadalmaztatott „Didaktomat” nevű jelfogós gép azonban már gyakorlati célokat is szolgált. Segítségével a tanár az óra alatt bármikor ellenőrizhette, értik-e a tanulók az anyagot.

— Más iskolában lehetett volna hasznosítani az itteni tapasztalatokat?

— Én is gondoltam erre, ezért 1963-ban felajánlottam, hogy könyvet írok a témában. A könyv meg is jelent a Táncsics Kiadó gondozásában — öt év múlva! Az Akadémia és egy svájci kiadó együttműködésével német nyelvterületen is olvashatták, újabb két év múltán. Akkorra már egészen más képp írtam volna meg, hiszen a szakkörben is már más, jóval modernebb dolgokkal foglalkoztunk. Sajnos egy ilyen, rohamléptekkel fejlődő területen kifejezetten káros a hazai publikációs gyakorlat olykor elképesztő lassúsága. A nyomdai átfutások „írásmával” csak egyes ipari üzemek vetekedhetnek.

— Tudna mondani erre egy példát?

— Mi 1983 májusában megkaptuk az iskolaszámítógépeket. Vártam, hogy ha némi késedelemmel, de hamarosan jönnek az interfészek. Csak 1985 végén kezdtek megjelenni!

— Egy kis kitérő után ugorjunk vissza a „dalás időkhöz”. Eddig csak a hardverről esett szó. Mikor „kóstoltak bele” a programozásba?

— Már a 60-as évek elején több kollégámmal együtt elvégeztük a programozói tanfolyamot. Közülük jószereivel csak én maradtam meg a középiskolai oktatás mellett. Ez a jelenség néhány helyen nem kis problémát okozott. Hallottam olyan iskoláról, ahonnan a tanárok a tanfolyamot követően egymás után mentek el fő hivatású programozóknak! Érthető, hogy az ilyesmi egy oktatási intézmény vezetését nem éppen a számítástechnika buzgó támogatására sarkallja. Most, hogy az iskolaszámítógépprogrammal nagyot léptünk előre technikai szempontból, ideje lenne a személyi feltételek javítására is jobban odafigyelni. Folytatva választom az eredeti kérdésre; ott tartottunk, hogy részt vettem a programozói tanfolyamon. Ezután, 1965-től kezdve szakköri keretben tanítottuk a FORTRAN-t.

— Volt már akkor saját gépük?

— Nem, de sikerült módot találnunk, hogy másutt kipróbálhassuk a programjainkat. Nagy fordulatot 1974 hozott, amikor kaptunk egy Hewlett-Packard gépet, ami akkor nagy szám volt. Még nagyobb lett a lelkesedés, amikor 1979-ben hozzájutottunk egy TRS-80-hoz, egy éven belül pedig a „párjához”. A két mikrogepen szinte éjjel-nappal folyt a munka. Így értük el, hogy a szakkörökben előbb a tanulók haromada, később pedig több mint fele el-sajátította a BASIC-et, és képességei szerint megtanult programozni. Csapataink a különböző versenyeken elért eredmények alapján az iskolának 1982-ben egyszerre három gépet nyertek. Jelenleg hét gépen dolgozunk. Különösen sokat várak a Videoton most beszerzett TV-Computerétől, amely szerintem hozzávetőleg annyit tud, mint a Commodore és a Spectrum

együtt. Azt hiszem, hogy a fiúk a meglévő technikai lehetőségekkel tovább fogják folytatni a hagyományos sikereket.

— Elégedett tehát a technikai feltételekkel?

— Nagyjából igen. De hadd említsem meg, hogy amikor a Primo megjelent, igen örültem, hogy végre nálunk is lehet látni az üzletek kirakataiban iskolásoknak való gépeket, be lehet menni, meg lehet venni, haza lehet vinni. Reméltem, hogy nemsokára akár 3 hasonló gép közül válogathatunk. Hát ez egyelőre álom. Gond még, hogy nehéz beszerezni minden alkatrészt, pedig egy chipen is leállhat a gép. Probléma, hogy a csatlakozó kiegészítő részek igen drágák a középiskolák számára. Rettentően kellene floppyegység, amiből egy darab 30 ezer forint, ha kapható. Ezért is dolgozunk a nehézkes magánokkal a szakkörben.

— Hányan vesznek részt most a szakköri munkában?

— Negyven körül. Működik két kezdő, egy haladó és egy gépi kódú tanfolyam. A hardverek jelenleg a Spectrumokhoz építenek egy-egy csoportban botkormányt, illetve fényceruzát, egy csoport pedig a digitális elektronika elemeivel foglalkozik.

— Véleménye szerint mi az iskolai szakkörök jövője a számítástechnikai kultúra elterjesztésében? Találkoztam olyan vélekedéssel is, hogy tantárgyszerűen, kötelezően kellene oktatni.

— Nem hiszem, hogy ez jó lenne. Kár túlerőltetni. A számítástechnika éppúgy nem való mindenkinek, mint a rajz és sok más. Elégnek tartom a mostani gyakorlatot, hogy a 3–4. osztályban a tanulók a matematika- és a technikaórákon ismerkedjenek meg a műszaki vonatkozásokkal és a programozással. Az érdeklődőknek, a kirtartóbbaknak ott vannak a szakkörök, hogy alaposabban elmélyüljenek a témában, és kipróbálják képességeiket. Egyáltalán nem biztos, hogy belőlük számítástechnikai szakember lesz, nem is ez a cél. A szakkörök célja, hogy későbbi életpályájukon a fiúk magától értetődően és főleg hatékonyan használni tudják korunknak ezt a nagy technikai vívmányát.

— Mint nagy tapasztalatú, az ifjúságnak a számítástechnikát oktató tanár, milyen gondolatot tart fontosnak, amivel befejezhetnének beszélgetésünket?

— A fiúk már régóta sürgetik a számítógép-építést. Nagyon várják már, hogy az üzletekben legyen kapható „zacskós számítógép”. Volt ígéret, hogy 1984 végére ez így lesz. Ennek másfél éve. Pedig a hardver éppen olyan fontos, mint a szoftver! Véleményem szerint a közeli jövőben egyre inkább szükség lesz olyan szakemberekre, akik a számítógépekhez, a nyomtatott áramkörökhöz ugyanúgy értenek, mint a programozáshoz.

LACZKA MIKLÓS



# Assemblerek, cross-assemblerek

## 5. A rendszer működése II. — Táblák és adatterületek

A rendszer működéséről leírtakból kitűnik, hogy a fordítóprogram többfajta táblát, veremmemóriát használ, az adatok (utasítás-sorok) feldolgozásához pedig azok átmeneti tárolása érdekében puffertületekre van szüksége. A következőkben a rendszer ezen elemeinek felépítésével foglalkozunk.

A fordítóprogram által használt vagy felépített legfontosabb táblák:

- műveletikód-tábla (utasítástábla),
- címketáblák (modul- és lokális címketáblák),
- makródefiniációs tábla.

Ezekon kívül a rendszer még három kód-táblát: a regiszter-, a regiszterpár- és a hibáuzenatkód-táblákat használja. Ezek szerepét a megfelelő operátorok bemutatásakor tárgyaljuk.

Nézzük végig a táblákat, hogy mi a tartalmuk és milyen a szerkezetük!

### Műveletikód-tábla

A műveletikód-tábla annyi elemet — bejegyzést — tartalmaz, ahány gépi utasítása van annak a mikroprocesszornak, amelyre a (cross-)assembler program készül, hozzáadva a fordítóprogramban megengedett direktívák számát. Minden egyes bejegyzés azonos hosszúságú, és tartalmazza egy gépi utasítás vagy direktíva mnemonikus kódját (nevét); annak jelzését, hogy gépi utasításról vagy direktíváról van-e szó; gépi utasítás esetén: az utasítás tárgykódjának hosszát bajtokban, az utasítás bináris műveleti kódját, hogy létezik-e az utasításnak operandusa, és ha igen, akkor milyen típusú (lásd a 2. részt), a regiszteroperandusoknál megkülönböztetve, hogy forrás (source) vagy fogadó (destination) típusú-e (ez utóbbi szintén Intel sajátosság); direktíva esetén: hogy van-e operandusa a direktívának, és ha igen, akkor kötelező vagy választható a használata, egy elugrási címet, amely a direktíva-feldolgozó operátoron (szubrutinon) belül az adott direktívát feldolgozó programrészlet relatív címét adja meg.

Ezzel tehát a direktíva-feldolgozó operátor megvalósításának egy technikai elemét rögzítettük: az egyes direktívák feldolgozását a szubrutinon belül egy ugrási táblázat-tal kívánjuk vezérelni.

Az egy bejegyzésben tárolt adatokat összefoglalóan — a név kivételével — műveletikód-információknak fogjuk nevezni.

### Címketábla

Valamennyi címketábla azonos szerkezetű. Egy bejegyzés 8 bajt hosszú, felépítése az 1. ábrán látható. Az első 5 bajt a maximum 5 karakteres szimbolikus nevet tartalmazza. A flag a szimbolikus név típusára utal. Lehetséges értékeit és ezek jelentését részletezve is megadjuk:

- 1: címkéről vagy EQU-val definiált relatív azonosítóról van szó, amelyek értéke megváltozik, ha a szerkesztőprogram a fordítási kezdőcímtől eltérő kezdőcímmel szerkeszti a programot,

**Nyolcrészes sorozatunk a mikroszámítógépek assemblereiről, cross-assemblereiről szól. Célja, hogy a cross-assembler példáján keresztül megismertesse az olvasót az assembler programok működésével. A bemutatáshoz a rendszermodellezési eszközöket használjuk fel, s így készíttjük el az assemblerek működésének egy szabványosított algoritmusát, modelljét. E cikkünk a sorozat ötödik része.**

1. (Cross-)assembler és a rendszermodellezés
2. Az Intel 8080 assembly nyelv
3. A (cross-)assembler program mint rendszer
4. A rendszer működése I. — A fordítás két menete
5. A rendszer működése II. — Táblák és adatterületek
6. Az operátorkészlet I.
7. Az operátorkészlet II. — A rendszer kapcsolatábrája
8. Példák a rendszer működésére

- 2: EQU-val definiált azonosító abszolút operandussal (az azonosító értéke a korábban tárgyalt programáthelyezés esetében sem változik meg),
- 3: SET direktívával definiált azonosító relatív operandussal,
- 4: SET direktívával definiált azonosító abszolút operandussal,
- 5: hibás (nem SET-tel), többször definiált címke.

Az érték ilyenkor nulla lesz.

A fordító a címketáblát mindig egy puffertületen kezdi felépíteni. Két puffertület van fenntartva: egy a teljes program-modul (globális) címkéit tartalmazó modul-címketábla és egy a mindenkori lokális makrócímketábla számára. Ha a használt puffertület betelik, akkor tartalma háttértárra kerül, s a puffertület újból tölthetővé válik. A puffertületeknek két mutatójuk van: egy a terület kezdetére, egy pedig a pufferen belül a következő szabad terület elejére mutat. Ez utóbbi mutató egyben a táblába is bekerül, legelső elemként (2. ábra).

A lezárt címketáblában az utolsó bejegyzést egy speciális végjelzés követi.

### Makródefiniációs tábla

A makródefiniációs tábla a fordítás alatt levő programból hívható valamennyi makró nevének és a háttértáron való elhelyezkedésének kezdőcímét tartalmazza. A makródefiniációk forrásnyelvi tartalma két helyen fordulhat elő a háttértáron: az állandó, minden felhasználó számára hozzáférhető

makrók egy külön, ún. makrókönyvtárba kerülnek be állandó megőrzésre, a felhasználó által a saját programjában definiált és csak innen hívható makrók az első menet végén a forrásprogram többi részével együtt a lemez kijelölt (átmeneti) területére íródnak fel. Így a makródefiniációs táblában léteznek állandó bejegyzések — a könyvtári makrókhoz tartozók —, s léteznek időleges bejegyzések, az egy-egy program lokális makródefiniációihoz tartozók. Ez utóbbiak a fordítás befejeződése után elvesznek. A makródefiniációs tábla is felkerül a háttértárra, s a makrókönyvtár katalógusaként szolgál. A fordítás kezdetekor beolvasódik a központi tárra, s ekkor válik a fordítóprogram számára hozzáférhetővé.

Felépítése a következő (3. ábra). Első eleme egy mutató, amely a katalogizált makrók bejegyzései után következő szabad területre mutat. A tábla további elemei a makrónev- és lemezcím párokból álló bejegyzések.

### Veremmemóriák

A táblák bemutatása után tekintsük át a makrók kezelésével kapcsolatos veremmemóriák: a megszakítási hely és makróparaméterek adatait tartalmazó, valamint a makrócímketábla-veremmemória felépítését. Mindkét veremmemóriában egy bejegyzés egy-egy makrókifejtési szinthez tartozik. Ez a két verem logikailag összetartozik, s együtt alkotják a makrókifejtési veremtartat. Mint már láttuk azonban, a második menetben csak a makrócímketábla-veremre van szükség, ezért válik fizikailag kétfelé a makrókifejtési veremmemória.

A megszakítási hely és makróparaméterek veremében egy bejegyzés a következő adatokat foglalja magában (4. ábra):

— Az előző (megszakított) szint megszakítási helye: egy lemezcím, ami egy lemezkardot jelöl ki, s a rekordon belüli elhelyezkedést megadó mutatóból álló adat határozza meg a megszakított szint következő sorának (amely *elé* a makrókifejtés beékeződik) a helyét. Kivételt képez az az eset, amikor a megszakított szint a nulladik, és a fordítandó program nem lemezen, hanem valamilyen más bemeneti adathordozón (lyukszalagon, kártyán, mágnesszalagon) helyezkedik el, mert ilyenkor a bemeneti periféria működése ideiglenesen leállítható a megszakítás helyén, majd a beolvasás a 0. szintre való visszatéréskor folytatható. Ebben az esetben a két mezőbe speciális jelzés kerül.

— Az aktuális paraméterek: egymástól vessző választja el őket, az utolsó után speciális jelzés, például pont (.) jelzi, hogy vége van az aktuális paramétereknek.

— Közvetlenül ezután a formális paraméterek következnek, egymástól vesszővel elválasztva, és végül ponttal lezárva. E két utóbbi rész hossza változó.

— A bejegyzés elejére egy mutató, a bejegyzést megelőző bejegyzés (ha van ilyen) kezdőcíme kerül.

A makrócímketábla-verem definiálása-



kor egyszerűsítéssel élünk, nevezetesen fel-tételezzük, hogy minden makrókifejtésben az egy szinthez tartozó lokális szimbolikus nevek száma nem haladhatja meg a háttér-táron egy előre rögzített számú (K) fizikai rekordot tartalmazó területen elférő címke-táblába írható bejegyzések számát. Azaz, minden makrócímetábla rögzített, K rekord hosszúságú. (Például K=1 vagy 5.) Ilyen kikötés mellett a verem felépítése igen egyszerűvé válik (5. ábra), mert minden makrókifejtési szinthez két lemezadat, a címketábla első és az aktuális rekord-jának a hozzáférési címét kell csak megjegyezni. K=1 esetén egyetlen lemezcima-dat, az első rekordé, elegendő.

## Pufferterületek

A következőkben összefoglaljuk, hogy a fordítóprogramnak milyen fontosabb pufferterületekre van szüksége a működéséhez. A forrásprogram feldolgozása és a fordítás eredményeinek (tárgyprogram, lista) létrehozása ezeken az adatmezőkön végzett műveletek segítségével történik. A rendszer operátorai tehát elsősorban ezeket az adatmezőket olvassák, módosítják, és ezek jelentik az operátorok közötti elsődleges információk kapcsolatát.

Csak röviden utalunk az egyes puffert-erületek feladatára, szerepük igazán a rendszerműködés részletes elemzése során fog kiviláglani. A (cross-)assembler programunk legfontosabb pufferterületei tehát:

SPB1, SPB2, SPB3: három egyenlő hosz-szú, s egyetlen forrásnyelvi sor befogadásá-ra alkalmas terület.

DFB1: mágneslemezzel való, akár az eredeti forrásprogram, akár a lemezen tá-rolt markódefiníció elérése érdekében vég-zett olvasási művelet puffere. Hossza meg-egyezik a háttértáron a fizikai rekordhosz-szal.

DFB2: az első menetben a forrásprog-ramnak a lemezre való felírásához használt pufferterület.

DFB3: a címkekutatásnál használatos pufferterület, ide olvassa vissza a fordító-program a modulcímetábla már kész, le-mezre felírt rekordjait.

MLB: a modulcímetábla pufferterülete, ahol a lemezre még fel nem került címke-a-datok tárolódnak, míg egy rekordnyi mennyiség össze nem gyűlik.

DFB4: K=1 esetén a makrócímetábla pufferterülete: itt gyűlnek a címkeadatok a lemezre való felírás előtt, s a lemezről való visszaolvasás is e területre történik. Ha K>1, akkor a makrócímetáblák kezelésé-hez ugyanúgy két különböző pufferterület-re van szükség, mint a modulcímetáblá-nál.

OBJB: az elkészített tárgyprogram-be-jegyzések pufferterülete, innen kerülnek ki a kívánt kimeneti perifériára.

LPB: a fordítási lista egy sorának puffer-területe. E területen állítja elő a fordító-program a tárgyalt listamezők kitöltésével soronként a nyomtatandó listát.

## További korlátozások

Mielőtt rátérnénk a rendszer működésé-nek részletes bemutatására, még néhány ki-egészítő feltételezést és megszorítást tes-zünk, hogy a program működése egysz-e-rűbben követhető legyen. Természetesen ezek nem szükségszerűen meglévő korláto-

zások lesznek; a programozó eltekinthet al-kalmazásuktól, ha programját általánosab-bá kívánja tenni. De ekkor rendszere egy-ben bonyolultabbá is válik.

A szűkítő feltételeket három csoportba oszthatjuk. Egy részük a fordító kétmen-tes voltából, más részük a (cross-)assembler program futtatására rendelkezésre álló tár méretének korlátosságából, harmadik ré-szük pedig a háttértáron fenntartott mun-katerületek nagyságának korlátosságából fakad.

A fordító kétmenetes voltából követke-zik a fejezet elején a makrókkal kapcsolat-ban tett megszorításokon kívül, hogy min-den esetben, amikor már az első menetben szükség van az operandusmező kiértékelé-sére (például IF, DS, ORG direktívák), a mezőben csak már korábban definiált szim-bolikus nevek szerepelhetnek.

Minthogy a (cross-)assembler program tárfoglalását lehetőleg minél kisebbé kíván-juk tenni — anélkül, hogy például a makró-használat lehetőségeiről lemondanánk —, az előbb tárgyalt veremmemóriák méretét úgy tarthatjuk kézben, hogy előírjuk a mak-róhívások egymásba skatulyázásának maxi-mális szintjét és ezzel a MIND rendszervál-tozó legnagyobb megengedett értékét, vala-mint rögzítjük a makrókban előfordulható paraméterek maximális számát is.

A mágneslemezen a különféle munkate-rületek és adatállományok számára előre lefoglalt vagy a (cross-)assembler program által, működése elején lefoglalandó terü-letek véges nagysága pedig meghatározza a makródefiníciók (könyvtári + időleges), az egy programban szereplő összes külső és belső makróhívások, valamint a program-ban használt modulcímkék maximális szá-mát. E két utóbbi korlátozás a makrócíme-táblák, illetve a modulcímetábla számá-ra fenntartott lemezterületek előre megha-tározott méretének következménye.

## A részletes rendszerműködés

Mindezek után vizsgáljuk meg részlete-sebben a rendszer működését! A korábban elmondottakból csak annyit ismételünk meg, amennyi feltétlenül szükséges a teljes-séghez.

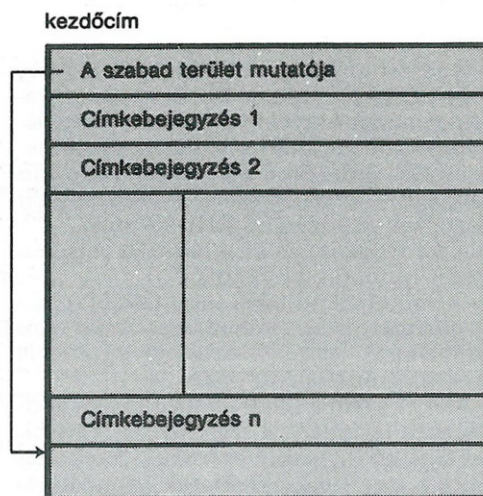
A (cross-)assembler program a hívási pa-raméterek feldolgozásával indul: a rend-szer meghatározza a bemeneti és a kimene-ti perifériákat, a mágneslemezen munkate-rületet foglal le a címketáblák, a kifejett forrásprogram számára. Majd behívja a tárba a makrókönyvtárból a makródefiníci-ós táblát, beállítja a különböző pufferterü-letek mutatóinak, a rendszerváltozóknak, indikátoroknak stb. a kezdeti értékét.

## Az első menet

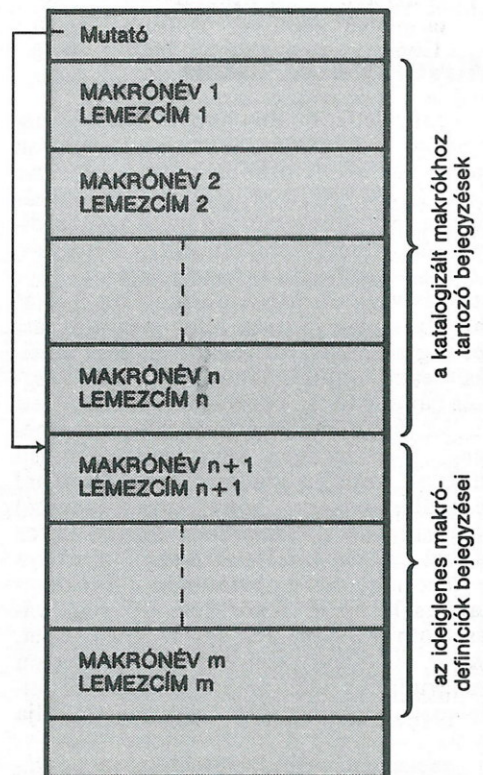
Ezután kezdődik a tulajdonképpeni első fordítási menet. A rendszer beolvas egy sort a bemeneti perifériáról az SPB1 terü-letre. Ha számítógépünkön mód van arra, hogy ezt a B/K műveletet aszinkron mó-don hajtsuk végre, azaz a fordítóprogram tovább tud haladni azalatt is, amíg a B/K művelet végrehajtódik, akkor gyorsíthatjuk a fordítóprogram működését azzal, ha az egyes sorok feldolgozását és a következő sor beolvasását időben átlapolva végeztet-jük el. A továbbiakban feltételezzük, hogy az olvasási művelet ilyen aszinkron módon

C	I	M	K	E	Flag	Érték
---	---	---	---	---	------	-------

1. ábra. Címketábla-bejegyzés



2. ábra. Címketábla

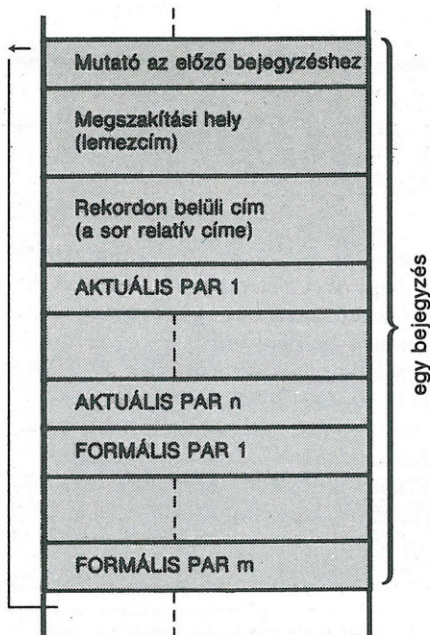


3. ábra. Makródefiníciós tábla

elvégezhető. Így a fordítóprogram a B/K befejeződése után a beolvasott sort átviszi az SPB2 területre, és újból indítja az olva-sási műveletet.

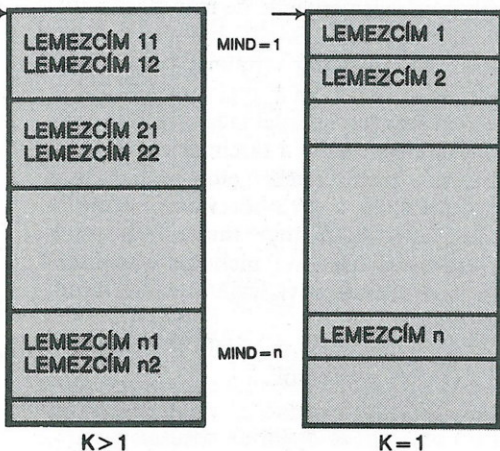
Az SPB2-re ávitott sort mezőkre bontja fel, s mezőnként folytatja a feldolgozást. Ha a sor teljes egészében megjegyzésként értel-mezendő, akkor nem foglalkozik a tartal-mával. Ugyanez az eset, ha a rendszer az AIF1 vagy az MK1iIF, illetve az MDEF ál-lapotok egyikében van, azaz egy korábbi IF direktíva feldolgozásakor a feltételes fordít-ás feltétele nem teljesült vagy makródefi-níció feldolgozása folyik, kivéve, hogy az egymásba ágyazott IF-ENDIF direktívák párosságának vizsgálata és a kiinduló álla-potba (A1 vagy MK1i) való visszatérés ve-zérlése érdekében az előforduló IF-EN-DIF, illetve az ENDM direktívákat (és csak ezeket) figyelnie kell a műveletkód-mezőben.





4. ábra A megszakítási hely és makró paraméterek veremtéra

a verem kezdőcím mutatója



5. ábra. A makrocímke-tábla-veremtár

Flag	Hossz	Tárgykód

6. ábra. Tárgyprogram-bejegyzés

Ha a sor se nem megjegyzés, se nem kihagyandó, akkor makrókifejtési állapot (MK1i) esetén előbb elvégzi — ha szükséges — a formális és az aktuális paraméterek cseréjét a különválasztott mezőkben, majd a megváltozott mezőket visszaírja SPB2-re.

A címkemző feldolgozása következik. Ha a mező nem üres, akkor a MIND értéke által meghatározott aktuális (modul- vagy makró-) címke-táblát vizsgálja meg a rendszer, hogy létezett-e már ilyen szimbolikus név. Ha nem, akkor egy, az 1. ábra szerinti bejegyzést készít a táblába. A feldolgozásnak ebben a pontjában még minden szimbolikus név címke név minősül, ezért 1-es flag s értéként a beültetési számláló értéke kerül bejegyzésre.

A műveletkód-mező feldolgozása során a rendszer előbb azt dönti el, hogy nem makróhívásra utal-e a mező tartalma. Ha igen, tehát a makródefiníciós táblában is szereplő makró név áll a mezőben, akkor elvégzi a makrókifejtés indításához szüksé-

ges és korábban már tárgyalt műveleteket: mélyíti a megszakítási hely és makróparaméterek, valamint a makrocímke-tábla-veremmemóriákat (4. és 5. ábra), bár a formális paramétereket még nem tudja kitölteni — erre majd a következő, a MACRO direktívát tartalmazó sor feldolgozásakor kerül sor —, s elindítja ennek a sornak, azaz a tárolt makródefiníció első sorának beolvasását DFB1-en keresztül SPB1-re. A rendszer a makróhívás hatására MK1i állapotba kerül, a MIND rendszerváltozó értéke eggyel nő.

Ha a fordítás bemeneti perifériája nem mágneslemez és a megszakított szint a nulladik, akkor a makróhívás indításával az előzőek szerint párhuzamosan az SPB1-re olvasandó következő forrásprogramsort az olvasási művelet befejeződése után el kell menteni, mivel lemeztől eltérő periférián az olvasási művelet nem ismételtető. Erre a mentésre szolgál az SPB3 puffertérlet. A makródefiníció olvasása csak e mentés elvégzése után indítható.

Kicsit részletesebben foglalkozunk a makrocímke-tábla-verem kezelésével. Létezik egy olyan LMCT-nek nevezett mutató, amelybe a (cross-)assembler program indításakor és a második menet kezdetekor is a következő értéket töltjük be: LMCT = a lemezen a makrocímkek számára lefoglalt terület előtt K rekordnyira lévő terület kezdőcíme. Akkor a makróhívás indításakor az alábbi műveletet kell elvégezni:

- ha a régi  $MIND \neq 0$ , akkor az előző szinthez tartozó aktuális makrocímke-táblát átmenetileg le kell zárni, és a lemeznek arra a területére kell kivinni, amelyet a veremnek a MIND által meghatározott bejegyzésében a második lemez cím (LEMEZCÍM<sub>2</sub>,  $i = MIND$ ) határoz meg;
- LMCT értékét a K rekordnyi területnek megfelelő értékkel növelni;
- és ezt az új értéket a makrocímke-tábla-veremnek a MIND új értéke által meghatározott helyére (mindkét címre) bejegyezni;

— a DFB4 puffertérleten a szabad terület mutatóját a puffertérletre kell állítani.

Ha a makró törzs feldolgozása során egy makrocímke-tábla-rekord betelik, a rekordot le kell zárni, a LEMEZCÍM<sub>2</sub> által meghatározott lemeztérletre ki kell írni, majd LEMEZCÍM<sub>2</sub>-t a következő lemezkód címére kell állítani.

A makrókifejtés befejeződésekor elvégzendő műveletek:

- az aktuális makrocímke-táblát véglegesen le kell zárni és a verem megfelelő eleme által kijelölt lemezcímre ki kell vinni;
- a makrocímke-tábla-veremben ki kell keresni az új (a réginél eggyel kisebb) MIND-értékhez tartozó makrocímke-tábla lemezcímét (LEMEZCÍM<sub>2</sub>), ha van ilyen;
- ezt a címke-tábla-rekordot vissza kell olvasni a DFB4 puffertérletbe;
- a címke-tábla első elemének segítségével be kell állítani a címke-táblába írt utolsó bejegyzés utáni szabad terület kezdetének mutatóját a puffertérleten.

Ha a műveletkód-mező tartalma nem makró név, akkor a műveletkód-táblát nézi végig a rendszer. Ha a kód gépi utasításra utal, akkor csak az utasítás hosszát állapítja meg. A direktívákat a korábban tárgyalt módon feldolgozza. SET és EQU direktíva esetén felülírja az aktuális címke-tábla utolsó bejegyzésének flag- és értékmezőit. Az IF, ENDIF, MACRO, ENDM, END direktívák hatására rendszerállapot-átmenet következhet, illetve következik be.

A forrásnyelvi sor, feldolgozásának végén, átmenetileg a háttértárra kerül fel. A rendszer megvizsgálja, hogy a címke-tábla puffertérlete nem telt-e be, majd az END direktíva beérkezésétől függően vagy lezárja a sorfeldolgozási ciklust a beültetési számláló szükség szerinti növelésével, a ciklus indikátorainak alapállapotba hozatalával, és kezdi az újat, vagy lezárja az első menetet a modulcímke-tábla lezárásával és háttérlemezre való kiírásával, a puffertérletek kiürítésével, a rendszerváltozók értékének az A2 állapotnak megfelelő módon való beállításával.

## A második menet

A második menetben a rendszer a lemeztől olvassa be a makrókifejtéssel bővített forrásprogramot. Egy sor olvasása SPB1-re és egy sor feldolgozása SPB2-n ugyanúgy átlapolva történik, mint az első menetben. Ugyancsak hasonlóan kezeli a rendszer a megjegyzéssorokat és a kihagyandó sorokat: tartalmukkal nem foglalkozik — kivéve az IF és ENDIF direktívákat a műveletkód-mezőben —, csak az esetleg készülő fordítási listára nyomtatja ki őket. A makródefiníció(k) forrásaival hasonló a helyzet: csak az ENDM direktívát keresi a rendszer, különben kihagyandó sorokként kezeli őket.

A műveletkód-mező feldolgozása újból a makróhívás eldöntésével kezdődik. Ha a mező tartalma egy makró név, akkor az első menet tárgyalása során ismertetett algoritmus szerint a rendszer meghatározza az új makrocímke-tábla lemezcímét, behívja a DFB4 területre, mélyíti a makrocímke-tábla-veremmemóriát, és bekövetkezik a rendszer új állapotba való átmenete (MK2i).

Ha a mező gépi utasítás kódját tartalmazza, akkor a műveletkód-információk és az operandusmező kiértékelése segítségével a rendszer meghatározza az utasítás tárgykódját 1, 2 vagy 3 bajt hosszúságban. A direktívákat a bemutatott módon értelmezi, s ha tartozik hozzájuk tárgykód, akkor kiszámítja azt. A tárgykód az OBJB puffertérletbe kerül egy flag- és egy hosszinformációval való kiegészítés után (6. ábra). A flag a szerkesztőprogram számára adja meg az utasítás vagy direktíva típusát (lásd később a vonatkozó operátor tárgyalásánál), a hossz pedig a következő flag adja meg a tárgykód hosszát bajtokban.

A feldolgozás végén történik a hibaüzenetek beállítása, és szükség esetén a lista nyomtatása. A rendszer működése során különféle ellenőrzéseket végez, és az előírások, korlátozó feltételek stb. megsértésekor hibát jelez. A hibák kétfélek lehetnek: felfedésük vagy a fordítás azonnali megszakítását vonja maga után, vagy a fordítás tovább folytatódik, és csak egy hibaüzenet utal a hiba jelenlétére. Az előző csoportba tartoznak az olyan hibák, mint például a megengedett szintet meghaladó egymásba ágyazott makróhívások. Az utóbbi csoportba tartozó hibára néhány példa: többszörös címke-definiálás, helytelen műveleti kód használata, nem definiált címke stb.

A feldolgozásnak ebben a szakaszában természetesen csak a második csoportba eső hibákról lehet szó.

Az END direktíva megérkezésekor a rendszer lezárja a fordítást, és visszaadja a vezérlést a (cross-)assembler programot hívó operációs rendszernek.

VÁRGEDŐ TAMÁS



## BASIC és gépi kód

Előző számunkban a gépi kódú programozás lehetőségeiről és korlátairól volt szó. Megismerkedtünk a gépi kódot betöltő BASIC programok egyik típusával, valamint egy egyszerű időmérő eszközzel, mellyel a BASIC-ben, illetve gépi kódban írt programok sebességét lehet összehasonlítani. Most a gépi kódú programrészek tárbeli elhelyezkedéséről és aktivizálásáról lesz szó.

### Hol helyezük el a gépi kódú rutinokat?

Erre a RAM-on belül számos lehetőség adódik, ezek közül ismertetek néhányat.

Leírásokban gyakori a szalagpufferben történő elhelyezés. A múlt alkalommal közölt rutinokat is a szalagpufferbe tettük. Ez a 828 ... 1019 (\$033C ... \$03FB) című tartományban helyezkedik el. A C16 típusú gépeken néhány bajtnyi — a felhasználás szempontjából lényegtelen — eltérés van. Itt óvatosságnak kell lenni, mert a mágnesszalagos egység (datasette) használata esetén a beviteli-kiviteli rendszer is ezt a területet használja.

A VC20 és C64-en van egy kevésbé népszerű, de sokkal biztonságosabb terület. A 679 ... 769 (\$02A7 ... \$02FF) tartományt használhatjuk szabadon. A C16-ra vonatkozóan erről a területről egyelőre sajnos nincs pontos információ.

A C64-en 49152 ... 53247 (\$C000 ... \$CFFF) között van egy 4 kilobájtos szabad terület. Itt már nagyobb programrészek is elférnek. Általában az ún. monitorprogramokat szokták itt elhelyezni.

Mindhárom géptípuson gyakran használt módszer az, hogy a BASIC programok rendelkezésére álló terület (az ún. BASIC-RAM) felső végéből vágják le a szükséges

darabot. Ehhez tudni kell, hogy az 55 ... 56 (\$37 ... \$38) című bajtokon egy mutatót találunk, mely a BASIC-RAM utáni első bajt címét tartalmazza, a mikroszámítógépeknél megszokott sorrendben, vagyis a kisebb helyértékű bajt megelőzi a nagyobb helyértékűt.

A gépi kód betöltése előtt ezt a címet — általában POKE utasítással a betöltőprogramból — a kívánt értékre kell beállítani. Erre példa a bemutatott program.

Előfordul, hogy nem a BASIC-RAM te-  
tejből, hanem az elejéről vágnak a gépi kódú rutin részére. Ez a módszer elég kényelmetlen, mert a mutatókat nem lehet a programból állítani, hanem a betöltőprogram háttértárolóról való betöltése előtt kell parancs módban POKE utasításokkal átírni.

A Commodore gépeken viszonylag ritkán használják azt a más gépeken gyakori módszert, hogy a gépi kódú részt egy REM utasításban helyezik el. A későbbiekben erre is látunk majd néhány példát.

Létezik még jó néhány különleges, egyedi megoldás; ezekkel szintén később találkozunk.

### Hogyan aktivizáljuk a gépi kódú rutinokat?

Ha a gépi kódú alprogram a helyére került, valamilyen módon meg kell hívni, hogy működni kezdjen. Erre most két módszert ismertetek: a SYS utasítást és az USR függvényt. A harmadik elterjedt módszerrel, az ún. beszúrásról majd később írok, ha már jobban megismertük a gépi kódú programozás titkait.

A SYS utasítás után a BASIC programban egy aritmetikai kifejezés áll, általában számkonstans vagy numerikus változó, melynek értéke az a gépi cím, amelyre a vezérlést akarjuk átadni. Ezt a címet a gépi rutin belépési pontjának nevezzük. A gépi rutin befejezése után a SYS utáni BASIC utasításra adódik a vezérlés.

A SYS utasítás használatakor paramétere-  
reket is adhatunk át a gépi kódú alprogramnak. Ez történhet paraméterlistával vagy a gép regiszterein keresztül.

Az USR függvény használata bonyolultabb. A BASIC programban, ahol a BASIC szintakszisa függvényhivatkozást enged meg, bárhol állhat. Formája: USR (kif.). Itt a „kif.” a függvény argumentuma, mely tetszőleges érvényes aritmetikai vagy karakterlánc-kifejezés lehet. Az argumentumot a gépi kódú részben lehet feldolgozni, ennek módját később ismerjük meg.

A gépi kódú függvény belépési pontjának címét a függvényhivatkozás előtt egy meghatározott helyen, az USR vektorban

kell elhelyezni. Ennek címe a három géptípuson más. Az USR vektorok címét az 1. táblázat tartalmazza, bár egyelőre még nem sokat tudunk kezdeni vele.

A gép bekapcsolását követően az USR vektorokban egy hibaüzenetet kiírató, a BASIC-ROM-ban található rutin belépési pontjának a címe van. Nézzük meg ezt a hibaüzenetet, például a következő utasítás bevitelével (akár program módban, akár parancsmódban):

```
PRINT USR(1)
```

### Paraméterátadás a regisztereken keresztül

A 6502-es mikroprocesszor regisztereit legközelebb ismertetem, de már most elárulok annyit, hogy a BASIC-ből a SYS utasítással négy regiszter tartalmát lehet beállítani: az A, X, Y, és P regisztereket. Ezeknek egy-egy tárcím felel meg. A SYS végrehajtása előtt ezekre a tárcímekre kell a regiszterek kívánt értékét elhelyezni POKE utasítással, és a SYS-ből való visszatérés után PEEK függvénnyel innen olvashatjuk ki azokat az értékeket, melyeket a regiszterek közvetlenül a visszatérés előtt tartalmaztak.

A regisztereknek megfelelő tárcímek a 2. táblázatban találhatóak.

### A programról

Mostani egyetlen programunk a BASIC-RAM végén történő elhelyezésre ad példát, egyelőre magyarázat nélkül. Hibátlan lefutása után jegyezzük fel a képernyőn megjelenő „SYS” utáni értéket. Utána POKE utasítással tegyünk gépünk A regiszterébe (lásd 2. táblázat) egy 0 és 255 közötti számot, majd SYS utasítással adjuk a vezérlést a gépi rutinra. Ha nem hibáztunk, akkor egy számjegy jelenik meg a képernyőn. Az eljárást megismételve próbáljuk megfejteni, mit csinál a gépi kódú rutin.

A betöltőprogram sikeres futás után NEW parancssal törölhető, de ne felejtjük el még futtatás előtt háttértárolóra kimenteni.

BARNA LÁSZLÓ

Géptípus	Decimális	Hexadecimális
	cím	
VC20	1 .. 2	\$01 .. \$02
C64	785 .. 786	\$0311 .. \$0312
C16	1281 .. 1282	\$0501 .. \$0502

1. táblázat. A USR vektor helye a különböző géptípusokon

Géptípus	A	X	Y	P
VC20	780	781	782	783
C64	780	781	782	783
C16	2034	2035	2036	2037

2. táblázat. A SYS utasításnál használható (decimális) regisztercímek

```
10 B=PEEK(55)+256*PEEK(56)-18
15 H=INT(B/256):L=B-256*H
20 POKE55,L:POKE56,H:CLR
25 B=PEEK(55)+256*PEEK(56)
30 FORI=BT0B+17
35 READA:POKEI,A:C=C+A
40 NEXT
45 IF C<2242 THEN PRINT"ADATHIBA":STOP
50 PRINT"A RUTIN SYS" B"UTASITASSAL HIVHATO"
55 DATA162,0,201,0,240,6,10,144
60 DATA249,232,16,246,138,9,48,76
65 DATA1A,255
```



# Gépikód-szerkesztő 1 kb-ajtos ZX81-re

Ezt a programot akkor írtam, amikor még nem készült el a memóriabővítésem. Akkoriban tapasztalnom kellett, mennyire kevés az 1 kb-ajt, különösen ha esetleg a képernyőt is teleírom. Ezért kényszerültem rá a gépi kódú programozásra. A gépi kódú programok helyfoglalása kb. harmada az azonos teljesítményű BASIC programokénak, és akkor takarékoskodnak megfelelőképpen a hellyel, ha minél több programot hívnak a ROM-ból.

Első próbálkozásaim izasztóak voltak: egyenként bepötyögtetni egy olyan hosszú REM sort, amelybe majd a gépi kódú programot elhelyezhetem. Ez persze a szükségesnél hosszabb nem lehet, tekintettel a memória méretére. Így született meg a gondolat: a program maga legyen képes nyújtani a REM sort, mindig egy karakterrel, amikor egy bájtnyi utasítást begépelek.

A program első sora a gépi kódot tartalmazó REM sor, második sora átadja a vezérlést az INICI címkével jelzett iniciáló rutinnak:

```
20 RAND USR 16857.
```

A program ekkor kiírja a „gyártó cég” nevét, és billentyűre vár. Bármely billentyű megnyomására RAMTOP-ot állít, feltölti önmagát oda, és NEW utasítást hajt végre. Most a vizsgálni kívánt programot LOAD utasítással betölthetjük vagy elkészíthetjük a REM sort, amely a szükségesnél rövidebb lehet. A program feltétlenül legalább kétsoros legyen, mert különben a gép végtelen listázási ciklusba kerülhet, és ilyenkor csak a RESET gomb vagy a csatlakozó kihúzása segít.

A programot most és újraindításakor is RAND USR 17290-nel indíthatjuk. Erre megjelenik a LIST OR WRITE felirat. W betűre programírásba lép, X betűre kilép BASIC-be, bármely más billentyűre listázásra ugrik. LIST és WRITE üzemmóddhoz FROM ADDRESS? felirattal kezdőcímet kér.

LIST üzemmódban kiírja a tárcímet, az ott található tartalmat hexadecimális formában és karakteres formában (Tony Baker nyomán). A kép alján megáll és billentyűre várakozik; L betűre folytatja a listázást, bármely más betűre a FŐCÍM nevű rutinra ugrik.

WRITE üzemmódban kiírja az aktuális címet, és két számjegyű hexadecimális számot vár, amelyet majd oda beír. Ekkor az inzert üzemmód még nem működik. A tartalom beírása után billentyűre várakozik. W betűre a következő címet írja ki, majd a fentieket ismétli. I betűre is kiírja a következő címet, hexadecimális tartalmat vár, de ezúttal egy karakternyival megnyújtja a program első BASIC sorát, ahol a munkába vett gépi kódú program van. Ez nemcsak REM, hanem PRINT is lehet; lényegében itt minden olyan BASIC utasítás állhat, amely után bármilyen karakter beírható, például 10 LET A\$ = „AAAABBBBCCCCAN”

Ekkor a gépi kód a 4086<sub>H</sub> címen kezdődhet.

A program nagyon óvatos. Írást csakis akkor engedélyez, ha W vagy I betűvel kérjük. Bármely más billentyű hatására kilép a WRITE üzemmódból.

Ha hosszú programot írunk, előfordulhat, hogy néhány sor megjelenítése után a gép END OF DISPLAY vagy OUT OF RAM hibajelzéssel megáll. A SZERKESZTŐ írása közben előfordult, hogy csupán négy sort tudtam kiírni, mert a programot önmagával írtam.

A program a LIMT számkonstanssal megjelölt cím feletti részt nem engedi írni vagy olvasni. Ez a már betöltött programnál POKE 17066 és POKE 17067 utasítással módosítható.

A program leírásában minden szám hexadecimális.

A PRT szubrutin az A regiszter tartalmát kétjegyű hexadecimális szám formájában kiírja.

A KEZD tároló először a listázás kezdőcímként szerepelt, de ezt a későbbiekben feleslegesnek tartottam. Jelenlegi szerepe csupán annyi, hogy a megnyomott billentyű kódját ideiglenesen tárolja (418B-nél látható).

Az ADDR tároló a mindenkori aktuális címet tartalmazza.

Az ADD2 tároló eredetileg az aritmetikai műveletek miatt volt szükséges, a program jelenlegi változatában felesleges; így a 409E-nél található utasítás is.

A LIMT a listázási határcímet tartalmazza.

A LIST rutin a fentiekben leírt listázást végzi. A LAP címkére ugrik a gép, ha újabb sort írunk ki, és ha a képernyő betelt. A XOR A utasítások szerepe, hogy szöközket helyezhessünk el a kiírt szövegbe. A 40BD-n kezdődő utasítás megvizsgálja, hogy nyomtatható karakter van-e az A regiszterben, és ugrik a PTH címkére. Ellenkező esetben törli az akkumulátort, tehát szöközt ír. A 40CA-nál kezdődő utasítások megvizsgálják az SPOSN rendszerváltozót, hogy a képernyő hányadik sorában tart a kiírás, mert az EDIT sorba való írás hibajelzést hozna létre. A 40D2-nél kezdődő utasítások a kiírási pozíciót a képernyő bal felső sarkába helyezik („lapdobás”). A JUMP címkére való ugrás jön létre, ha elértük a LIMT listázási határt.

A BLTYÜ nevű szubrutin érdekessége, hogy a program kétféle módon is hívja. A 40D8 sor például a RAMTOP fölé feltöltött címével hívja, a 41E8 sor viszont eredeti helyén látja még. Szerepe, hogy leolvassa a gép billentyűzetét, az eredményt dekódolja, és az akkumulátorba helyezze. A 40F6-on álló utasítás megvizsgálja, hogy számjegyet vagy vezérlőgombot nyomtunk-e meg. Ha 0...F tartományba eső számjegyet írtunk be, akkor ezt a gép kiírja. A DUPL címkénél a billentyűzetet még egyszer leolvassuk, mert így előzhető meg a

„ráduplázás”: egy számjegy kétszeri beolvasása.

A NÉGYS címkénél említettem meg, hogy assemblerem akkor még nem volt, de igyekszem betartani a szabályt, hogy a címkék nem lehetnek öt karakternél hosszabbak. A NÉGYS jelű szubrutin feladata, hogy négy számjegyű hexadecimális számot olvasson be a billentyűzetről, és az eredményt a DE regiszterbe helyezze. Az első két számjegyet például a 4110 sorban álló utasítás egyesíti a D regiszter alsó és felső négybitestetrádjában. A billentyűzet leolvasása alatt a DE tartalmát a veremtárba mentjük, mert a BLTYÜ megváltoztatja a DE tartalmát.

Két számjegyű szám beolvasásakor a KÉTSZ belépési ponton érjük el a szubrutint.

A WRITE rutin végzi a memóriairás feladatát. Első néhány utasítása megvizsgálja, hogy túlléptük-e a LIMT által megszabott értéket; ekkor a FŐCÍM-re ugrik, és kiírja, hogy LIST OR WRITE. 4152 soránál megvizsgáljuk, hogy betelt-e a képernyő. Ha nem, a NLAP („nem lapozunk”) címkére ugrik, ellenkező esetben törli a képernyőt. Ekkor a gép billentyűre vár; helytakarékosságból itt nem íratok ki semmit. Az I billentyű megnyomására végrehajtódik a 4167 utáni programrészlet. Ez először FAST-ba kapcsolja a gépet, majd a HL regiszterbe helyezi az első sor hosszát jelölő címet, és inkrementálja annak tartalmát.

Szerencsére a ZX81 olyan sorrendben tárolja a BASIC sor hosszát, ahogyan a 16 bites számokat. A sor számával ezt ilyen egyszerűen már nem lehetne megcsinálni. Következő lépésként a HL regiszterbe helyezük az aktuális beírás címét, majd meghívjuk a ROM-ból a MAKE ROOM rutint a ONE SPACE belépési pontján. Ez egyelőre följebb másolja az egész RAM tartalmát a HL-ben megjelölt címtől a STKEND-ig, és az érdekelt rendszerváltozók értékét inkrementálja (például DFILE). Utána SLOW-ba kapcsoljuk a gépet, és folytatódhat a programírás.

A FŐCÍM rutin fejléct ír a programba való minden egyes belépésnél. Működését már ismerttettem. Itt csak annyit említek meg, hogy a programkezelés logikája szerint előbb kell megkérdeznem, hogy LIST OR WRITE („állítmány”), mert zavaró lenne, ha indításakor a program csak annyit írna ki: FROM ADDRESS? Ezért kell a lenyomott billentyű kódját ideiglenesen tárolni.

A rutin után helyeztem el három üzenetet. Az STR1 és STR2 egymásba gabalyodva helyezkednek el, mert az adott szituációban (1 kb-ajtos gépben a RAMTOP fölött) a meghívott ROM rutin kettesével veszi az adatokat! Az STR3-at még RAMTOP alól kell kiírni, ezért karakterei normális sorrendben helyezkednek el. Ezt egyébként nem kell már áttölteni, hiszen elég, ha a program csak előszörre mutatkozik be.



Éppígy nem töltjük fel RAMTOP fölé az INICI rutint sem, hiszen ez is csak először szükséges. Működésekor, ha egy billentyűt lenyomtunk, a DE-be tölti a program új helyét, és onnan mindjárt be is írja a RAMTOP rendszerváltozóba. A program eredeti helyét a HL regiszterbe, hosszát a BC regiszterbe írjuk, majd az LDIR utasítással az egész programot a végleges helyére áttöltjük. Végül a NEW utasításra ugrunk, ez

törli az eredeti helyén levő programot, hogy újat vehessünk munkába.

Látható, hogy a program után üres helyek vannak még a tárban. Ez az eredeti veremtar helye. Ha oda programot írunk, majd kiadjuk a NEW parancsot, ez utóbbi elrontja az ott található adatokat, mégpedig azért, mert működése kezdetén még az eredeti veremtarat használja, s csak néhány lépés után tér át az új RAMTOP által meg-

Szerkesztő				4123 D1	POP DE
4082 F5	PRT	ORG 4082		4124 D61C	SUB A,28
4083 E6F0		PUSH AF		4126 83	ADD A,E
4085 1F		AND F0		4127 5F	LD E,A
4086 1F		RRA		4128 C9	RET
4087 1F		RRA		4129 ED5BA642	WRITE LD DE,(ADDR)
4088 1F		RRA		412D 2AAA42	LD HL,(LIMT)
4089 C61C		ADD A,1C		4130 ED52	SBC HL,DE
408B D7		RST 10		4132 3845	JR NC,F0C1M
408C F1		POP AF		4134 7A	LD A,D
408D E60F		AND 0F		4135 CD9342	CALL PRT
408F C61C		ADD A,1C		4138 7B	LD A,E
4091 D7		RST 10		4139 CD9342	CALL PRT
4092 C9		RET		413C AF	XOR A
4093	KEZD			413D D7	RST 10
4095	ADDR			413E 110000	LD DE,0000
4097	ADD2			4141 CD2343	CALL KÉTSZ
4099	LIMT			4144 7B	LD A,E
409B 2AAA42	LIST	LD HL,(LIMT)		4145 ED5BA642	LD DE,(ADDR)
409E 22A842		LD (ADD2),HL		4149 12	LD (DE),A
40A1 54		LD D,H		414A 13	INC DE
40A2 5D		LD E,L		414B ED53A642	LD (ADDR),DE
40A3 2AA642	LAP	LD HL,(ADDR)		414F 3E76	LD A,NEWLINE
40A6 A7		AND A		4151 D7	RST 10
40A7 ED52		SBC HL,DE		4152 FD7E3A	LD A,(SPOSN)
40A9 19		ADD HL,DE		4155 FE03	CP 3
40AA 3034		JR NC JUMP		4157 2003	JR NZ,NLAP
40AC 7C		LD A,H		4159 CD2A0A	CALL CLS
40AD CD9342		CALL PRT		415C CDF442	CALL BLTYÜ
40B0 7D		LD A,L		415F FE3C	CP "W"
40B1 CD9342		CALL PRT		4161 28C6	JR Z,WRITE
40B4 AF		XOR A		4163 FE2E	CP "I"
40B5 D7		RST 10		4165 2012	JR NZ,F0C1M
40B6 7E		LD A,(HL)		4167 CD230F	CALL FAST COMMAND
40B7 CD9342		CALL PRT		416A 217F40	LD HL, ELSŐ SOR
40BA AF		XOR A		416D 34	INC (HL)
40BB D7		RST 10		416E 2AA642	LD HL,(ADDR)
40BC 7E		LD A,(HL)		4171 CD9B09	CALL MAKE ROOM
40BD CB76		BIT 6,(HL)		4174 CD2B0F	CALL SLOW COM-
40BF 2801		JR Z,PTH		4177 18B0	MAND
40C1 AF		XOR A		4179 CD2A0A	JR WRITE
40C2 D7	PTH	RST 10		417C 11C343	F0C1M CALL CLS
40C3 3E76		LD A,NEWLINE		417D 010D00	LD DE,STR1
40C5 D7		RST 10		417F 010D00	LD BC,STRHOSSZ
40C6 23		INC HL		4182 CD6B0B	CALL PRINT STRING
40C7 22A642		LD (ADDR),HL		4185 3E76	LD A,NEWLINE
40CA FD7E3A		LD A,(Y+3A)		4187 D7	RST 10
40CD FE03		CP 03		4188 CDF442	CALL BLTYÜ
40CF 20D2		JR NZ,LAP		418B 32A442	LD (KEZD),A
40D1 D5		PUSH DE		418E FE3D	CP "X"
40D2 010000		LD BC,0000		4190 C8	RET Z
40D5 CDF508		CALL PRT AT		4191 11C443	LD DE,STR2
40D8 CDF442		CALL BLTYÜ		4194 010C00	LD BC,STRHOSSZ
40DB D1		POP DE		4197 CD6B0B	CALL PRINT STRING
40DC FE31		CP "L"		419A 3E76	LD A,NEWLINE
40DE 28C3		JR Z,LAP		419C D7	RST 10
40E0 C38A43	JUMP	JP F0C1M		419D CD0C43	CALL NÉGYS
40E3 CDBB02	BLTYÜ	CALL KEYB.SCAN		41A0 ED53A642	LD(ADDR),DE
40E6 2C		INC L		41A4 CD2A0A	CALL CLS
40E7 20FA		JR NZ,BLTYÜ		41A7 3AA442	LD A,(KEZD)
40E9 CDBB02	DUPL	CALL KEYB.SCAN		41AA FE3C	CP "W"
40EC 44		LD B,H		41AC CA3A43	JP Z,WRITE
40ED 4D		LD C,L		41AF C3AC42	JP LIST
40EE 51		LD D,C		41B2 31	DEFB "LF
40EF 14		INC D		41B3 2B	DEFB "F"
40F0 28F7		JR Z,DUPL		41B4 2E37	DEFB "I",DEFB"R"
40F2 CDBD07		CALL KEYB.DECODE		41B6 3834	DEFB "S",DEFB"O"
40F5 7E		LD A,(HL)		41B8 3932	DEFB "T",DEFB"M"
40F6 FE2C		CP "G"		41BA 0000	DEFB "sp",DEFB"sp"
40F8 D0		RET NC		41BC 3426	DEFB "O",DEFB"A"
40F9 D7		RST 10		41BE 3729	DEFB "R",DEFB"D"
40FA C9		RET		41C0 0029	DEFB "sp",DEFB"D"
40FB CDF442	NÉGYS	CALL BLTYÜ		41C2 3C37	DEFB "W",DEFB"R"
40FE 110000		LD DE,0000		41C4 372A	DEFB "R",DEFB"E"
4101 D61C		SUB A,28		41C6 2E38	DEFB "I",DEFB"S"
4103 07		RLC A		41C8 3938	DEFB "T",DEFB"S"
4104 07		RLC A		41CA 2A0F	DEFB "E",DEFB"?"
4105 07		RLC A		41CC 3F	DEFB "Z"
4106 07		RLC A		41CD 26	DEFB "A"
4107 82		ADD A,D		41CF 33	DEFB "N"
4108 57		LD D,A		41D0 00	DEFB "A"
4109 D5		PUSH DE		41D1 38	DEFB "sp"
410A CDF442		CALL BLTYÜ		41D2 34	DEFB "S"
410D D1		POP DE		41D3 2B	DEFB "O"
410E D61C		SUB A,28		41D4 39	DEFB "F"
4110 82		ADD A,D		41D5 3C	DEFB "I"
4111 57		LD D,A		41D6 26	DEFB "W"
4112 D5	KÉTSZ	PUSH DE		41D7 37	DEFB "A"
4113 CDF442		CALL BLTYÜ		41D8 2A	DEFB "R"
4116 D1		POP DE		41D9 01080B	DEFB "E"
4117 D61C		SUB A,28		41DC CDF508	INICI LD BC,0B08
4119 07		RLC A		41DF 11CC41	CALL PRINT AT
411A 07		RLC A		41E2 010D00	LD DE,STR3
411B 07		RLC A		41E5 CD6B0B	LD BC,STRHOSSZ
411C 07		RLC A		41E8 CDE340	CALL PRINT STRING
411D 83		ADD A,E		41EB 119342	CALL BLTYÜ
411E 5F		LD E,A		41EE ED530440	LD DE,PRT(U)
411F D5		PUSH DE		41F2 014C01	LD (RAMTOP),DE
4120 CDF442		CALL BLTYÜ		41F5 218240	LD BC,HOSSZ
				41F8 EDB0	LD HL,PRT
				41FA C3C303	LDIR JP NEW

Címke neve	Eredeti helye	Áttöltés utáni helye	
		1 kbájt RAMTOP-ja	16 kbájt fölött
PRT	4082	4293	7E92
KEZD	4093	42A4	7EA3
ADDR	4095	42A6	7EA5
ADD2	4097	42A8	7EA7
LIMT	4099	42AA	7EA9
LIST	409B	42AC	7EAB
LAP	40A3	42BA	7EB3
PTH	40C2	42D3	7ED2
JUMP	40E0	42F1	7EF0
BLTYÜ	40E3	42FA	7EF3
DUPL	40E9	42FA	7EF9
NÉGYS	40FB	430C	7F0B
KÉTSZ	4112	4323	7F22
WRITE	4129	433A	7F39
NLAP	415C	436D	7F6C
F0C1M	4179	438A	7F89
STR1	41B2	43C3	7FC2
STR2	41B3	43C4	7FC3
STR3	41CC	43DD	7FDC
INICI	41D9	—	—

Címkék táblázata

határozott helyre. A NEW által beírt adatok kb. a 43DB címre kerülnek.

A címketáblázatban felsoroltak valamennyi címke eredeti helyét és RAMTOP fölötti helyét is. A memóriabővítés megépítése után szükségesnek láttam a programot még egy változatban megírni, hogy a hosszú gépi kódú programokba is bele tudjak olvasni. A MONITOR csodálatos program, de 5 kbájt hosszú! Először a FORTH-ba is a SZERKESZTŐ segítségével olvastam bele.

Nos, a táblázatban közlöm a 16 kbájt tejeten elhelyezett SZERKESZTŐ címkéit is, ezeket a CALL utasításokban át kell javítani. Sajnos a ZX81 hardverstruktúrája miatt gépi kódú programot nem lehet a memória felső részében futtatni. (Megpróbáltam.)

Végezetül néhány szót a ROM-ból felhasznált rutinokról. Az RST 10 neve: PRINT A CHARACTER. Funkciója, hogy a képernyő aktuális pozíciójába kiírja az akkumulátorban elhelyezett karaktert. Mint a programból is látható, ez akár a NEWLINE is lehet (kocsi vissza—soremeles).

A 02BB KEYBOARD SCANNING leolvassa a billentyűzetet, és az eredményt a HL regiszterben hagyja. Ha nem nyomtunk meg semmit, a HL tartalma FFFF. Ez INC és JR Z utasításokkal könnyen tesztelhető.

A 03C3 NEW COMMAND kiolvassa a RAMTOP értékét, átrendezi a rendszerváltozókat, és teljes törlést hajt végre. A RAMTOP feletti tárt érintetlenül hagyja az említett kivételtől eltekintve (4 bájt).

A 07BD KEYBOARD DECODE a BC regiszterben várja a KEYBOARD SCANNING által beolvasott adatot, és a HL-be helyezi a tárnak azt a címét, ahol az illető billentyű jelentését elhelyezték. Ez a ROM elején található.

A 08F5 PRINT AT a BC regiszter által megjelölt pozícióba helyezi a képernyőírási rendszerváltozókat. A B regiszter a sor, a C regiszter az oszlop számát tartalmazza.

A 099B ONE SPACE rutin 1-et ír a BC regiszterbe, majd áttér a 099E MAKE ROOM végrehajtására. Ez a HL-ben megjelölt helytől kezdve a BC-ben megadott számú üres helyet hoz létre, és gondoskodik arról, hogy a tár egészét ennyivel följebb másolja a memóriában. Ha nincs hely, a rutin END OF RAM hibajelzéssel megáll.

A 0A2A CLS COMMAND törli a képernyőt, és a nyomtatási pozíciót a bal felső



A HT-1080Z felhasználók között közkezen forgó FORTH interpreternek — bár a szabványos Fig-FORTH egy nagyon jó implementációja — van néhány hiányossága. (Nem minden kö-zépiskolában meglévő zFORTH-ról van szó. Annál a leírt problé-ma nem áll fenn. — A rovatvezető.)

A BASIC-hez szokott programozók gyakran indítanak el végtelen ciklusokat, abban a tudatban, hogy a BREAK billentyű-vel az bármikor megszakítható. A TINY-FORTH-szal ez a meg-szokás nagyon sok kellemetlenséget okozhat, mert a RESET gomb megnyomása után a rendszer nem mindig indítható el. Ez akkor fordul elő, ha használtuk az 1. screen-t is. Ha mégis újrain-dul, a /19584-es indítási cím esetén elfelejti az eddig lefordított FORTH szavakat. Itt jegyzem meg, hogy a /19589 címtől lehet melegindítást végrehajtani!

Az alábbi gépi kódú programmal és a FORTH belső inter-preterének átírásával lekezelhető a BREAK billentyű, méghozzá a BASIC megszakításkezelésénél hatékonyabban. A BREAK megnyomására a FORTH program futása felfüggesztődik, ezután NEW LINE hatására folytatódik, CLEAR hatására megsza-kad. Ekkor a rendszer melegindítást hajt végre.

A gépi kód bevitelét a monitorral (/12710) oldhatjuk meg a legegyszerűbben:

```
M4C5E
3A 40 38      19550: LDA, (14400)
E6 04         AND 4
20 03         JR NZ, KI
56           TOVABB: LD D, (HL)
              EX DE, HL
E9           JP (HL)
3A 40 38      KI: LD A, (14400)
B7           OR A
20 FH        JR, NZ, KI
3A 40 38      NEWL: LD A, (14400)
E6 03         AND 3
28 F9        JR Z, NEWL
FE 01        CP 1
28 EC        JR Z, TOVABB
C3 C5 5D     JP 24005
```

A FORTH interpreter módosítása:

```
M4CC4
C3 5E 4C     JP 19550
```

Az M üzemmódból mindkét esetben az X paranccsal léphetünk ki.

Miután beírtuk, B-vel lépünk ki a monitorból, majd indítsuk el a FORTH interpretert. A BREAK billentyűt kipróbálhatjuk például a VLIST parancs kiadása után, de magunk is definiálha-tunk olyan FORTH szavakat, amelyek hosszabb idő alatt hajtód-nak végre. Ha sikerült jól beírni és hibátlanul működik, akkor az így átalakított TINY-FORTH-t a memória alsó részére betöltődő TBUG monitorprogrammal menthetjük kazettára a P 4C5E 6BA0 4C80 FORTH paranccsal.

SOLTI CSABA

# FORTH

E FORTH definíció segítségével már meglévő FORTH sza-vak visszafejthetők, tanulmányozható működésük. A (:CODE) utáni rész, illetve elsődleges szavak visszafejtésére nem alkalmas, ezt disassembler segítségével tehetjük meg.

A definíció kisebb módosítást igényelhet egyes FORTH rendszerekben. A C64 CALC—RESULT-jában az OBRANCH-nek az (IF), a :S-nek az EXIT felel meg. A definíció használata megkönnyíti a FORTH-szal való ismerkedést, illetve nem doku-mentált, működésükben nem ismert szavakról a visszafejtésen ke-resztül bővebb ismereteket lehet szerezni.

Az új szó nem létező definícióra az N. E. üzenettel, ha elsőd-leges definíciót akarunk visszafejteni, az M. C. üzenettel áll meg (QUIT).

Előfordulhat, hogy a program „elkalandozik” a memóriá-ban. Ennek oka, hogy a vizsgált definíció nem :S-sel, illetve (:CODE)-dal záródik, mint például az ABORT vagy a QUIT ese-tében. Megoldás lehet, hogy a :S és a (:CODE) mellett ezeket is mint a vizsgált definíció végét vesszük figyelembe. Hátránya az ilyen megoldásnak, hogy ha a visszafejtő definíciónk egy IF ágon talál QUIT-ot, leállítja a visszafejtést.

Jó tanács: a SPELL-lel konstansokat és változókat ne pró-báljunk visszafejteni!

KUBIK ATTILA

```
SCR # 1

: SPELL 0 -FIND CR 0= IF ." N.E." QUIT
ELSE DROP DUP DUP CFA @ = IF ." M.C."
QUIT THEN 2- THEN
BEGIN 2+ SWAP
IF DUP @ SPACE . 0 SWAP 0
ELSE DUP @ 2+ NFA ID.
DUP @ 2+ [ ' BRANCH ] LITERAL = OVER
@ 2+ [ ' OBRANCH ] LITERAL = OR OVER
@ 2+ [ ' (+LOOP) ] LITERAL = OR OVER
@ 2+ [ ' (LOOP) ] LITERAL = OR OVER
@ 2+ [ ' LIT ] LITERAL = OR SWAP
OVER 0=
IF DUP @ 2+ [ ' CLIT ] LITERAL =
IF DUP 2+ C@ SPACE . 1+ @
ELSE DUP
@ 2+ [ ' (." ) ] LITERAL =
IF 2+ COUNT OVER OVER TYPE SPACE +
2- 0 ELSE DUP
@ 2+ [ ' (:CODE) ] LITERAL = OVER
@ 2+ [ ' :S ] LITERAL = OR ( OVER
@ 2+ [ ' EXIT ] LITERAL = OR )

THEN THEN ELSE 0 THEN THEN ?TERMINAL OR
UNTIL DROP DROP ; ;S

OK
```

sarokba helyezi. Az 1 kb-ajos gépnél zsugo-rított képernyőfájl jön létre (24 darab NEWLINE), ami helytakarékossági szem-pontból számunkra igen lényeges. A rutin megváltoztatja a BC tartalmát.

A 0B6B PRINT STRING 4 nevű rutin a DE-vel megcímezett, a BC-ben megadott hosszúságú karakterfüzért a képernyő aktu-ális kiírási pozíciójába írja. Mint látható, alkalmazása legalább 9 bájtot igényel, de az RST 10 esetén ciklust kell szerveznünk a kiírás lebonyolítására, s végeredményben az is hasonló hosszúságú lesz.

A 0F23 FAST COMMAND átkapcsolja a gépet FAST üzemmódba. Jelen esetben

ez azért előnyös, mert a MAKE ROOM ru-tin magát a képernyőfájlt is átmásolja.

A 0F2B SLOW COMMAND funkciója csupán annyi, hogy a gépet visszakapcsolja SLOW üzemmódba, s így láthatóvá tege a képet.

A programban szereplő két rendszervál-tozó szerepét az adott helyen ismerttettem: a 403A-n az SPOSN és 4004-en a RAM-TOP. Az első BASIC sor hossza a 407F-nél található.

DR. ZANA JÁNOS

### Irodalom

Ian Logan—Frank O'Hara: The Complete Timex TS 1000/  
Sinclair ZX81 ROM Disassembly  
Tony Baker: Mastering Machine Code on Your ZX81

Személyszámítógép-javítás, -karbantartás közületeknek, magánszemélyeknek. Egyedi megrendelés alapján kiegészítő berendezések gyártása. Pl. Sinclair fényceruza, joystick interface, oktatási intézményeknek kabinet kialakítá-sa.

Pásztor Ferenc, személyszámítógép-javító és -karbantartó kispáros. Szolnok, Mátyás király u. 2. V/3.





**PROGRAMOK**

# Öt gép vizsgálata egy programmal

Az azonos című, ez évi 4. számunkban közölt program C16, PRIMO és TVC változatával könnyen rajzolhatunk nagy felbontású spirálokat és sokszögeket.

## C16

Először be kell állítani, hány szögű sokszöget akarunk rajzolni (-50-től +50-ig). A „+” számoknál homorodik, a „-” számoknál domborodik az alakzat. A második adat a szögeket alkotó spirálok sugarát adja meg; minél nagyobb, annál pontosabb az alakzat, kisebb a sugár. A harmadik adat azt mutatja, hányszor forduljon körbe a

spirál (1-18). Végül meg kell adnunk, hogy az alakzat mozogjon-e, csökkenő sugarú legyen-e és a képernyőt kívánjuk-e előtte törölni.

Az ábrák paramétereinek értékei a táblából olvashatók le.

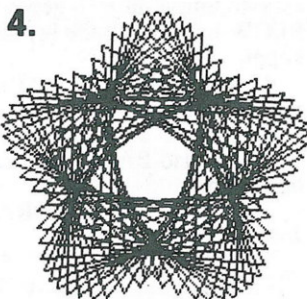
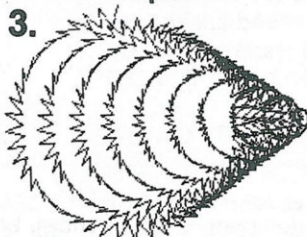
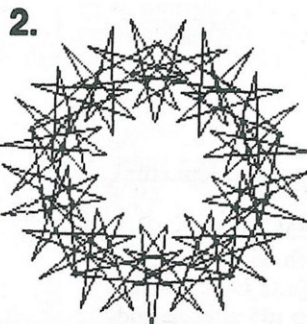
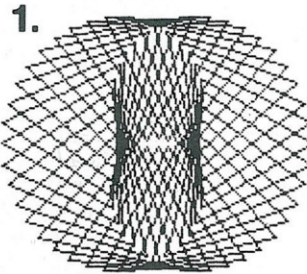
BALÁZS GYÖRGY

Változónév	K	R	F	X\$	Y\$	T\$
Rajzsorszám						
1.	34	28	1	N	N	I
2.	45	40	1	N	N	I
3.	48	60	10	I	I	I
4.	35	1	1	N	N	I

```

10 PRINT "C"
20 PRINT "#####SPIRALIZER"
30 FORT=1 TO 1000: NEXT
40 PRINT "C"
50 K=5
60 INPUT "SPEED (-50- 50)"; K$
70 K=VAL(K$)
80 IF K$="" THEN K=5
90 IF K<-50 OR K>50 THEN 60
100 K=K-1
110 R=35
120 INPUT "RADIUS (1- 60)"; R$
130 R=VAL(R$)
140 IF R$="" THEN R=35
150 IF R<10 OR R>60 THEN 120
160 R=R+13: S=1
170 INPUT "SPIN (1- 18)"; Z$
180 Z=VAL(Z$)
190 IF Z$="" THEN Z=1
200 IF Z<1 OR Z>18 THEN 170
210 A=1/2: IF Z>1 AND Z<9 THEN 280
220 SM=1: M=2
230 INPUT "MOVEMENT OR DECREMENT"; X$
240 IF LEFT$(X$,1) <> "M" AND LEFT$(X$,1) <> "D" THEN
0 M=0: SM=0: PRINT "NONE": GOTO 2
250 IF LEFT$(X$,1) = "M" THEN SM=0: GOTO 2
260 IF LEFT$(X$,1) = "D" THEN M=0: GOTO 2
270 PRINT X$
280 PRINT: INPUT "CLEAR (Y/N)"; T$
290 IF LEFT$(T$,1) <> "N" THEN PRINT "YES": E=0
300 IF LEFT$(T$,1) = "N" THEN PRINT "NO":
310 W=1: Z=140: IF M=2 THEN Z=80: IF A=1 THEN W=5: M=1: Z=140
320 IF SM-A=0 THEN W=5
330 IF A<1 THEN K=K+1
340 C=0.001: IF A<1/9 THEN M=M/2: C=C/2
350 J=R: I=80-R
360 X0=Z: Y0=20
370 FORT=0 TO 6.28314/ANNUM STEP 0.06283
380 IF SM THEN J=R#S: I=80#S-J: S=S-C
390 X1=Z+T#M-SIN(T)*J+SIN(T#K)#I
400 Y1=80-COS(T)*J-COS(T#K)#I: Y1=Y1+20
410 GOSUB 1000: X0=X1: Y0=Y1
420 NEXT
430 BE TKEY$
440 GRAPHIC0: GOTO 4
450 GRAPHIC1
460 IF LEFT$(T$,1) = "Y" AND E=0 THEN GRAPHIC1: I=I+1
470 DRAW1: X0, Y0 TO X1, Y1
480 RETURN

```



## PRIMO

A program két Primo változata közül egyik a grafikus programcsomag használatával, a másik anélkül készült.

HORVÁTH ZOLTÁN

```

5 'NEV:spiralizer@G
10 CLS
20 PRINTTAB(16)CHR$(5)"SPIRALIZER"CHR$(21)
25 FOR T=1 TO 1000: NEXT
35 PRINT: PRINT: K=5
40 INPUT "SPEED (-50 TO 50)"; K$
41 K=VAL(K$)
42 IF K$="" THEN K=5
45 IF K<-50 OR K>50 THEN 40
47 K=K-1
48 R=35
50 INPUT "RADIUS (1-60)"; R$
51 R=VAL(R$)
52 IF R$="" THEN R=35
55 IF R<1 OR R>60 THEN 50
57 R=R+13: S=1
60 INPUT "SPIN 1-18"; Z$
61 Z=VAL(Z$)
63 IF Z$="" THEN Z=1
65 IF Z<1 OR Z>18 THEN 60
70 A=1/2: IF Z>1 AND Z<9 THEN 200
80 SM=1: M=2
85 INPUT "MOVEMENT OR DECREMENT"; X$
90 IF LEFT$(X$,1) <> "M" AND LEFT$(X$,1) <> "D"
" THEN M=0: SM=0: PRINT: PRINT "NONE": GOTO 200
95 IF LEFT$(X$,1) = "M" THEN SM=0: GOTO 200
100 IF LEFT$(X$,1) = "D" THEN M=0: GOTO 200
200 PRINT: INPUT "CLEAR (Y/N)"; T$
205 IF LEFT$(T$,1) <> "N" THEN PRINT "YES":
E=0: 'PUT CLEAR OF HIRES
210 IF LEFT$(T$,1) = "N" THEN PRINT "NO"
220 W=1: Z=139: IF M=2 THEN Z=80: IF A=1 THEN
W=5: M=1: Z=122
230 IF SM-A=0 THEN W=5
235 IF A<1 THEN K=K+A
238 C=0.001: IF A<1/9 THEN M=M/2: C=C/2
240 J=R: I=79-R
250 X0=Z: Y0=171
260 'MAIN LOOP
270 FOR T=0 TO 6.28314/ANNUM STEP 0.06283
280 IF SM THEN J=R#S: I=79#S-J: S=S-C
290 X1=Z+T#M-SIN(T)*J+SIN(T#K)#I
300 Y1=112+T#M-COS(T)*J-COS(T#K)#I: Y1=Y1-20
310 GOSUB 1000: X0=X1: Y0=Y1
320 NEXT
330 A$=INKEY$: IF A$="" THEN 330
340 GOTO 35
1000 IF LEFT$(T$,1) = "Y" AND E=0 THEN CLS: E=1
1010 CMDJ: X0, Y0, X1, Y1, 1
1030 RETURN

```

```

5 'NEV:spiralizer@G
10 CLS
20 PRINTTAB(16)CHR$(5)"SPIRALIZER"CHR$(21)
25 FOR T=1 TO 1000: NEXT
35 PRINT: PRINT: K=5
40 INPUT "SPEED (-50 TO 50)"; K$
41 K=VAL(K$)
42 IF K$="" THEN K=5
45 IF K<-50 OR K>50 THEN 40
47 K=K-1
48 R=35
50 INPUT "RADIUS (1-60)"; R$
51 R=VAL(R$)
52 IF R$="" THEN R=35
55 IF R<1 OR R>60 THEN 50

```



```

57 R=R+13:S=1
58 INPUT "SPIN 1-18":Z$
59 Z=VAL(Z$)
60 IF Z$="" THEN Z=1
61 IF Z<1 OR Z>18 THEN 60
70 A=1/2:IF Z>1 AND Z<9 THEN 200
80 SH=1:M=2
85 INPUT "MOVEMENT OR DECREMENT":X$
90 IF (LEFT$(X$,1) <> "M" OR LEFT$(X$,1) <> "m") AND
ND(LEFT$(X$,1) <> "D" OR LEFT$(X$,1) <> "d") THEN
M=0:SM=0
92 PRINT:PRINT "NONE":GOTO 200
95 IF (LEFT$(X$,1) = "M" OR LEFT$(X$,1) = "m")
THEN SM=0:GOTO 200
100 IF (LEFT$(X$,1) = "D" OR LEFT$(X$,1) = "d")
) THEN M=0:GOTO 200
200 PRINT:INPUT "CLEAR (Y/N)":T$
205 IF (LEFT$(T$,1) <> "N" OR LEFT$(T$,1) <>
n") THEN PRINT "YES":E=0
210 IF (LEFT$(T$,1) = "N" OR LEFT$(T$,1) = "n")
) THEN PRINT "NO"
220 W=1:Z=139:IF M=2 THEN Z=80:IF A=1 THEN
W=5:M=1:Z=122.
230 IF SM=A=0 THEN W=5
235 IF A<1 THEN K=K+A
238 C=0.001:IF A<1/9 THEN M=M/2:C=C/2
240 J=R:I=79-R
250 X0=Z:Y0=171
260 'MAIN LOOP
270 FOR T=0 TO 6.2831/A*M STEP 0.06283
280 IF SM THEN J=R*S:I=79*S-J:S=S-C
290 X1=Z+T*M-SIN(T)*J+SIN(T)*K*I
300 Y1=112+COS(T)*J+COS(T)*K*I:Y1=Y1-20
310 GOSUB 1000:X0=X1:Y0=Y1
320 NEXT
330 A$=INKEY$:IF A$="" THEN 330
340 GOTO 35
1000 IF (LEFT$(T$,1) = "Y" OR LEFT$(T$,1) = "y")
AND E=0 THEN CLS:E=1
1010 GOSUB 1100
1030 RETURN
1100 'VONAL MűZo RUTIN
1110 IF X0=X1 AND Y0=Y1 THEN SET(X0,Y0):
GOTO 1155
1115 IF ABS(X1-X0) >= ABS(Y1-Y0) THEN ST=
(X1-X0)/ABS(X1-X0) ELSE 1135
1120 FOR XX=X0 TO X1 STEP ST
1125 YY=Y0+(XX-X0)*(Y1-Y0)/(X1-X0)
1130 GOSUB 1160:NEXT XX
1135 IF ABS(X1-X0) < ABS(Y1-Y0) THEN ST=(Y1-Y
0)/ABS(Y1-Y0)
1140 FOR YY=Y0 TO Y1 STEP ST
1145 XX=X0+(YY-Y0)*(X1-X0)/(Y1-Y0)
1150 GOSUB 1160:NEXT YY
1155 RETURN
1160 IF XX>255 OR XX<0 OR YY>191 OR YY<0
THEN 1670
1165 SET(XX,YY)
1170 RETURN

```

## TVC

A következő módosításokkal programunk Dragon-változata a TVC-n is futtatható.

20 PRINT AT 7,14,"SPIRALIZER";  
A 25. sor lesz a 30.  
A 40;50;60;85;200 soroknál INPUT helyett INPUT PROMPT és ; helyett : áll.  
A 90,95,100,205,210,1010. sorokban LEFT\$(változó, szám) helyett változó (: szám) lesz.

A 290. sor végére kerül : X1=4\* X1  
300 Y1=161+ COS (T) \*J+ COS (T)\*K) \*  
I : Y1= Y1-20:Y1=4 \*Y1

310 1000 helyett 1010  
1000 nincs  
1010 PCLS helyett CLS van.  
1020 PLOT, X0,Y0,X1, Y1  
K=5  
R=35  
Z=15  
D  
Y

eset teljesen más képet adott, mint a COMPUTE! c. újságban APPLE II gépre közöltek.

Ha a DRAGON-változatnál a következő módosításokat beírjuk:

A 270. sor elejére  
PMODE4,1:PCLS:SCREEN1,1  
1000 nincs

akkor a kép ugyanaz lesz, mint a TVC-n.  
SIMONYI ZSUZSA

## HOMELAB 2 (AIRCOMP 16)

# Morze adóprogram

A rádióamatőr-forgalomban jelentős feladatot tölt be Samuel Morse 1837-ben kidolgozott kódrendszere. Jelentősége abban van, hogy segítségével igen nagy távolságú kapcsolatokat lehet létrehozni annak veszélye nélkül, hogy a kiküldött jeleket a háttérzaj elnyomja.

Az amatőr adástechnikában az egyszerű kopogtatót már régen felváltották a gyorsbillentyűk, morzeírógépek és egyéb nagyobb teljesítményű berendezések. Napjainkban a számítástechnika rohamos elterjedésével lehetőség nyílt a továbblépésre. A számítógép nagy információátviteli kapacitása lehetővé teszi több oldalnyi gépelt szöveg egyidejű tárolását, gyors hozzáférést. A gép memóriájába bevitt adatok egyszerű programmal átkódolhatók morzejelekké. Ezt a feladatot hajtja végre ez a kis program.

A program három fő részre bontható. A BASIC programnak csak annyi a feladata, hogy ellássa a felhasználót a szükséges információval, és elindítsa a gépi kódú rutint. A BASIC rész működése egyszerű, ezért magyarázatára nem térek ki.

A gépi kódú rutin végzi a begépel szöveg átkódolását, majd a morzekód kiküldését a magnókimenetre. A rutin első része lefordítja a bevitt szöveget olyan speciális kódra, amit az adórutin külön vizsgálat nélkül képes értelmezni. Ha a leadandó szövegben olyan írásjelet talál, aminek a kódját az adó nem ismeri, jelzést ad, és a további fordítás megszakad. A lefordított szöveg a memóriába kerül, ahol az adórutin hozzáfér. A fordítás elvégzése után továbbadja a vezérlést az adóprogramnak.

Az adórutin végzi a lefordított kód morzejelekké alakítását, majd azok kivételét a magnókimenetre. A rutin addig vár, míg meg nem nyomjuk az alsó SHIFT gombot. Csak ekkor kezdődik az adás. Erre azért van szükség, mert a szöveg begépelésekor keletkező hangimpulzusok kijutnak a kimenetre. A rádióadó „élesztése” után már nem lehet további szöveget bevinni. Az alsó SHIFT gomb megnyomásával — zavaró jel nélkül — indul az adás. A felső SHIFT gomb segítségével az adás bármikor megszakítható.

A program harmadik részében van tárolva az a kódtáblázat, ami alapján a fordító dolgozik. Egy-egy morzejelet három bájttá ábrázol, amiből kettő vonatkozik a morzekódra. Az első bájtt az írásjel ASCII kódja, a második bájtt az elemi jelek számát tartalmazza, a harmadik bájtt a morzejel, ahol 0 bit = pont és 1 bit = vonás. Például a Q betű morzejele — — — A táblázatban a következőt találjuk: 51H 04H 0BH, ahol 51H a Q betű ASCII kódja, 04H: négy alapelemből álló jel következik és 0BH 00001011B, ami a Q betű bináris jelölése a morzejelek megfelelően.

A morzejelek bájtos tárolási módjából adódóan egy jel nem lehet nyolc alapelemnél hosszabb. Kivétel ez alól az a morzejel, amely nyolc alapelemenként periodikus. Mivel a szabványos jelkészlet ilyen kódot nem tartalmaz, ezért egyes bájtcsoportok második bájttja nem lehet 08H-nál nagyobb.

A program használatba vétele előtt be kell állítani a H és L változók értékét. A H változó adja a jelek hangmagasságát, az L változó pedig egy alapelem hosszát. A megadott számokat a rutin számlálóként használja, így az FFH érték eredményezi a legmélyebb hangot, illetve a leghosszabb alapelemet (pontot).

A BASIC program listája:  
10 HM=20480:PRINT "■!!", "MORSE ADOPROGRAM"  
20 PRINT "UTASITASOK:!"  
30 PRINT "LN=N JELHOSSZ (1-FFH)"  
40 PRINT "HN=N HANGMAGAS SAG(1-FFH)"  
50 PRINT "PN K=K ADASA N-SZER!"  
60 PRINT "(ALSO SHIFT)=ADAS MEGKEZDESE"  
70 PRINT "(FELSO SHIFT)=ADAS-MEGSZAKITASI!"  
80 PRINT "R=VISSATERES BASIC-BE!!"  
90 PRINT "AZ UTASITASOK FORMAJA MEGEGYEZIK A MONITOR UTASITASAIVAL"  
100 CALL HM:PRINT "■"

A gépi kódú program listája:

5000	32	3F	3F	CD	BE	01	CD	0B
	C9	18	F5	FE	52	20	02	E1
	C9	FE	48	20	05	7D	32	91
	40	C9	FE	4C	20	05	7D	32
	90	40	C9	FE	50	C0	E5	D5
	DD	E1	11	4D	52	DD	7E	00
	DD	23	FE	60	28	26	21	CD
	50	01	35	00	ED	A1	20	06
	0E	02	ED	B0	18	E7	23	23
	EA	3C	50	E1	F5	3E	0D	EF
	3E	20	EF	F1	11	BE	50	EF
	E7	C8	18	FB	12	3A	FE	3A
	1F	38	FA	32	3E	3E	21	4D
	52	7E	FE	60	28	2A	47	23
	4E	23	3C	28	1B	11	01	3C
	CB	09	30	02	1E	03	CD	9E
	50	11	01	3B	CD	9E	50	10
	EC	1E	02	CD	9E	50	18	D9
	11	07	3B	CD	9E	50	18	D1
	E1	2D	C8	E5	18	C8	E5	2A
	90	40	7C	12	3D	20	FC	15
	7C	12	3D	20	FC	14	2D	20
	F1	1D	20	EB	E1	3A	FE	3A
	E6	20	C0	E1	E1	C9	2D	54
	20	4E	45	4D	20	49	53	4D
	45	52	45	4D	60			

A hexadecimális kódtáblázat:

50CD	20	FF	FF	30	05	1F	31	05
	1E	32	05	1C	33	05	18	34
	05	10	35	05	00	36	05	01
	37	05	03	38	05	07	39	05
	0F	2E	06	2A	2C	06	33	3F
	06	0C	3A	06	07	3B	06	15
	2D	06	21	3D	05	11	27	06
	1E	2F	05	09	22	06	12	41
	02	02	42	04	01	43	04	05
	44	03	01	45	01	00	46	04
	04	47	03	03	48	04	00	49
	02	00	4A	04	0E	4B	03	05
	4C	04	02	4D	02	03	4E	02
	01	4F	03	07	50	04	06	51
	04	0B	42	03	02	53	03	00
	54	01	01	55	03	04	56	04
	08	57	03	06	58	04	09	59
	04	0D	5A	04	03	5B	05	16
	5C	05	04	5D	04	07	5F	04
	0C	2B	05	0A	23	06	2D	00



# Négy ötlet

Az alábbi, jól használható ötleteket Palotai Péter olvasónktól kaptuk.

## A VC20 is tud magyarul!

Ezt a programot azért készítettem, hogy a VC20 is tudjon ékezetes betűket írni. Programom rövidebb és egyszerűbb a más gépekre írt, ugyanezt a feladatot ellátó programoknál (1. lista).

```
90 REM * EKEZETES BETUK VC 20 *
100 PRINT "JVARJ!" : FOR T=0 TO 2047 : Q=PEEK(32768+T) : POKE 5120+T,Q : NEXT
110 POKE 36869,253 : PRINT "J"
120 POKE 36879,14
130 FOR M=1 TO 9
140 READ A : FOR T=0 TO 2 : READ B : POKE 5120+A*M+B : NEXT
150 IF M>5 THEN READ C
160 FOR V=3 TO 7 : F=PEEK(32768+(A-64+C)*8+V) : POKE 5120+A*M+V,F : NEXT
170 POKE 7680+M*2,A
180 NEXT
190 DATA 65,25,37,67,69,125,65,65,73,29,9,9,79,24,44,74,85,74,74,74
200 DATA 80,66,24,36,-1,111,66,90,36,-32,89,36,0,66,-4,119,36,36,66,-34
```

### 1. program

A karakterkészlet a ROM-ban a 32768-as címen kezdődik. Ha megváltoztatjuk a karaktertár kezdőcímét úgy, hogy a karakterkészlet a RAM-ban legyen elhelyezve, akkor beleszólhatunk a karaktergenerálásba. A karaktertár kezdőcímét a 36869 bajtra lehet állítani.

A program működése:

- 10 a karaktertárat áttöltjük a ROM-ból a RAM-ba az 5120-as címtől kezdődően
- 20 a karaktertár kezdőcímét átállítjuk 5120-ra
- 30 a képernyő színét állítja be
- 40–80 az új karaktereket generálja
- 90–100 az karakterekhez szükséges adatbázis

A programot RUN-nal indítva előbb várni kell, majd amikor megjelenik a képernyőn az összes hosszú magánhangzó, használni is lehet őket a következőképpen:

Á=SHIFT A

É=SHIFT E

Í=SHIFT I

Ó=SHIFT O

Ú=SHIFT U

Ö=SHIFT P

Ő=<COMMODORE BILLENTYŰ> P

Ű=SHIFT Y

Ű=<COMMODORE BILLENTYŰ> Y

A szerkesztő megjegyzése. A cikk címe helyesen az lenne, hogy „A VC20 is csak úgy tud magyarul, mint a C64, a Spectrum és a C16”. Ennek magyarázata, vagyis az, hogy miért lehetetlen a teljes és a magyar helyesírásnak megfelelő elhelyezkedésű betűkészlet előállítás, lapunk 1984/5. számának

31. oldalán, valamint az ÖTLET 1985. márciusi számának 25. oldalán található.

## Nagyított betűk

Nem árt, ha van egy olyan programunk, amely a betűket és karaktereket felnagyítja. Ezt főleg előadásokon, reklámprogramok létrehozásakor használhatjuk, de játékprogram írásakor is hasznos lehet (2. lista).

```
90 REM * NAGYITOTT BETUK VC 20 *
100 PRINT "J** TURELEM **"
110 POKE 36869,253
120 POKE 36867,25
130 FOR T=0 TO 2047
140 Q=PEEK(32768)
150 K=2*#T
160 POKE 5120+K,Q
170 POKE 5121+K,Q
180 NEXT
```

### 2. program

A programot RUN-nal indítva megjelenik folyamatosan egy felírás. Ez átíródik, majd a READY felirat jelenik meg, de már nagyban.

A karaktertár a 32768-as címtől kezdődik. Ha átállítjuk a kezdőcímet, akkor lehetőségünk van a karakterek megváltoztatására. A kezdőcím a 36869 bajton tárolódik. A nagyítást úgy végezzük, hogy a 8×8-as pontmátrixot 8×16-ra állítjuk át. Ez a 36867 bajton történik.

A program működése:

- 10 a karaktertár kezdőcímét állítja át, 5120-tól kezdődik így
- 20 a 8×16-os pontmátrixot hozza létre
- 30–80 a ROM-ban tárolt karakterbájtokat kiolvassa. Az 5120-as címtől kezdődően leírja az előzőleg kiolvasott értéket úgy, hogy 2 bajton (5120 és 5121) ugyanezt az érték szerepel. Így egy karakter 16 bajtot foglal el, nem 8-at.

## Törölt program visszahívása

Előfordul, hogy a megírt programot nem rögzítettük, és úgy töröljük ki a tárból. Mit tehetünk ilyenkor? Vagy újra írjuk a programot, vagy kikapcsoljuk a gépet és mérgeződünk. A következő eljárással visszahozható a program. A módszer lényege, hogy a gép NEW utasítás hatására nem a programot törli, hanem a programjelző regisztereket állítja nullára.

Ezek az alábbiak:

4097 bajt — az első sor jelzőszáma

4098 bajt — a programjelző (ha van program, 16)

4099 bajt — a programkezdő sorszám

Ha tehát vissza akarjuk írni a kitörölt programot, a 3 programjelző regiszterbe visszaírjuk a törölt tartalmat.

Két adatot könnyen megjegyezhetünk:

4098, 16  
4099, a programkezdő sorszám

Az első sor jelzőszámát úgy számítjuk ki, hogy a sorszámot követő utasítás vagy jel 7-et jelent, az utasítást vagy a jelet követő karakterek mindegyike 1-et jelent, a szóközt is beleértve. Például:

10 POKE 4098,16      10 A=52768  
7 1111111=15      7 111111=13

Ha kiszámoltuk a jelzőszámot, a 4097 bajtba beírjuk:

4097, jelzőszám

## Egy sokhasznú hardverötlet

Az előző példát felhasználva megoldhatunk még egy gondot. Van olyan eset, amikor egy programot elindítva, a program „elszáll”. VC20-nál ez legtöbbször a képernyő befeketedését jelenti, és ilyenkor a STOP RESTORE sem használható. Mégis van lehetőség a program visszahozására.

A felhasználói kapun egy tökéletes RESET van elhelyezve.

1 GND

3 RESET

E csatlakozók rövidre zárásával — ami kapcsolóval történik — a gép alaphelyzetbe kerül úgy, hogy a bent lévő program nem törlődik ki, de listázni nem tudjuk.

Eljárásunk az előző példa nyomán:

4097, jelzőszám (első sor jelzőszáma)

4098, programjelző (16)

4099, programkezdő sorszám

Ezek után ismét listázhatunk.



# APPEND program

A C16 kazettás egységének közismerten lassú működése miatt a nagyobb méretű programokat célszerű modulonként elkészíteni. Így az egyes programrészek kipróbálásával, módosításával kapcsolatos SAVE- és LOAD-műveletek viszonylag rövid idő alatt elvégezhetőek. Az alább ismertetett program (1. lista) az ily módon elkészült programrészek egyesítésére alkalmas.

Ha betöltése után RUN parancsot adunk, akkor az 1 REM sor tartalmának felülírásával két rövid, DATA sorokban tárolt gépi kódú rutin válik felhasználhatóvá. Ezek közül az elsőt a 9-es sor USR utasítása hívja. Hatására a változóábla kezdőcímét tartalmazó, az \$2D-\$2E címen található mutató értéke 2-vel csökken, majd betöltődik az \$2B-\$2C címre, amely a BASIC program kezdetének mutatója. Ezt követően a program futása a STOP utasítás miatt megszakad. A további teendőkről a képernyőfelirat tájékoztat:

```
LOAD, RENUMBER..., SYS 4123,
RUN
```

Ezt követve, először az egyesítendő programrészt kell betölteni. Ha megtörtént, akkor a sorszámegezés elkerülése érdekében újraszámoztatást kell végezni olyan kezdő sorszámmal, amely nagyobb a korábban betöltött program (első esetben az APPEND) legmagasabb sorszámanál. Ezután SYS 4123 parancsot adva, a második gépi kódú rutin hatására a BASIC kezdőcím az eredetire (\$1001) állítódik vissza, ezzel az újonnan betöltött program az előző után kapcsolódik. A következő modul betöltése előtt RUN parancsot kell adni, hogy a fentiekben ismertetett egyesítési folyamat értelem szerűen ismétlődhessen. A legutolsó modul elhelyezését követő SYS 4123 után DELETE 1-12 parancs következzen, amely az APPEND programot kitörli a gép memóriájából.

A gépi kódú rutinok könnyebb áttekinthetősége érdekében megadom azok disassemblerrel előállított listáját is. (2. lista).

PÁL LÁSZLÓ

## 1. lista

```
1 REM ##### APPEND C 16 #####
2 PRINT"##### LOAD , RENUMBER... ,";
3 PRINT" SYS 4123 , RUN #####"
4 IF PEEK(4104)=166 THEN 9
5 FOR I=4104 TO 4131
6 READ A:POKE I,A
7 NEXT
8 POKE 1201,8:POKE 1202,16
9 X=USR(0):STOP
10 DATA 166,45,164,46,130,200,1,136,202
11 DATA 130,200,1,136,202,134,43,132,44
12 DATA 96,169,1,133,43,169,16,133,44,96
```

## 2. lista

```
1000 A6 2D LDX #2D
100A A4 2E LDY #2E
100C 0A TXA
100D D0 01 BNE #1010
100F 00 DEY
1010 CA DEX
1011 0A TXA
1012 D0 01 BNE #1015
1014 00 DEY
1015 CA DEX
1016 06 2B STX #2B
1018 04 2C STY #2C
101A 60 RTS
101B A9 01 LDA #01
101D 05 2B STA #2B
101F A9 10 LDA #10
1021 05 2C STA #2C
1023 60 RTS
```

# ADOK – VESZEK – CSERÉLEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetések 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.

■ ZX-Spectrumhoz játékprogramokat cserélek. Válaszokat a programok listájával erre a címre kérek: Ötvös Tibor, Kecskemét, Tánácsics Mihály u. 7. 6000.

■ Commodore 16-os gépre irt játékprogramok cseréjéhez partnert keresek. László István, Békéscsaba, Máriássy u. 24.

■ ZX-Spectrum 48 k programokkal együtt eladó. Mészáros Károly, Budapest, Kerepesi út 1. 1087.

■ Sinclair Spectrum 48 k interfésszel, botkormánnyal, 700 db felhasználói és játékprogrammal, felhasználói programokhoz leírással, minitívével, DATA magnóval külön-külön is eladó. Levélcím: Kristófy Gyula, Solymár, Vasút u. 62. 2083.

■ Felajánlok megvételre külön vagy együtt: 1 db Commodore VC20 számítógépet, 10 E Ft; 2 db VC20 szuper bővítőmodult, nagy felbontású grafika+8 kb-ajt RAM, 5 E Ft/db; 1 db japán kismagnó háttértárat, külön illesztővel, 4 E Ft; 1 db VC20 BASIC programkazettát, 500 Ft. Magyar nyelvű gépkönyvet biztosítok! Kecskeméti Ottó, Miskolc, Leszich Andor u. 1. 3524. Telefon: 62-261.

■ Keresem a kapcsolatot Timex-Sinclair 2068-as számítógép tulajdonosaival. Tapasztalatcsere! Cím: 9023 Győr, Kassák L. u. 20. Gócsa Károly. Telefon: 22-178.

## A HCC Budapesti szekciói

(Az NJSZT azonos nevű szakosztályai)

**Apple Szekció:** Minden hónap első keddjén 3—5 óra közt a Neumann János Számítógéptudományi Társaságban (Bp. V., Báthory u. 16.) Vezető: Diebel Dietrich.

**Commodore Szekció:** Minden szerdán 3—6 óra közt a Budapesti Műszaki Egyetem F épületében (Bp. XI., Budafoki út. 8.) II. épületszárny II. em. 9. Vezető: dr. Simonyi Endre.

**Homelab Szekció:** Minden kedden 6—8 óra közt a Belvárosi Művelődési Házban (Bp. V., Molnár u. 9.) Vezető: Barabás Rezső.

**Sinclair Szekció:** Minden hétfőn 6—8 óra közt a Szellőző Művekben (Bp. XI., Építész u. 14.) Vezető: Hivessy Ferenc.

**Az Atari Szekcióval kapcsolatban felvilágosítást** Hirschler Tamás ad. (128-262-es tel.)

**Az „Építsünk számítógépet” és 68(x) x x Szekciókkal kapcsolatos felvilágosításokat** dr. Simonyi Endrétől (elérhető szerkesztőségünk telefonszámán minden hétfőn 8—14 óra közt, vagy az 556-245-es telefonszámon hétfőn és pénteken 5—7 óra közt) kaphatnak.



## Négykarakteres PRINT rutin

Akik szép játékot szerettek volna írni nagy figurákkal, azok észrevehették, hogy ha elmozgatták a több karakterből álló figurát, az csak úgy sikerült, hogy a figura csúnyán, homályosan vibrált. Ez a program egy 2 x 2-es karaktert állít elő a grafikus A, B, C, D, betűkből. Az óriáskarakterben a betűk a következőképpen helyezkednek el: bal felső karakter: A, jobb felső karakter: B, bal alsó karakter: C, jobb alsó karakter: D.

Ezt a óriáskaraktert a képernyő bármelyik helyére eltehetjük. A koordinátákat B- és C-regiszter tartalmazza, mégpedig B az y koordinátát, C az x koordinátát. A karakter színelemait a megfelelő vezérlésnél állíthatjuk be. Például ha a 220-as sorba ezt írjuk: LD A,1, akkor inverzbe írja a karaktert. A karakter koordi-

C350	10	ORG	50000
C350	20	ELE	ENT
C350	C381C3	30	JP
C353	3E10	40	PRINT
C355	D7	50	RST
C356	3E00	60	LD
C358	D7	70	RST
C359	3E11	80	LD
C35B	D7	90	RST
C35C	3E07	100	LD
C35E	D7	110	RST
C35F	3E12	120	LD
C361	D7	130	RST
C362	3E00	140	LD
C364	D7	150	RST
C365	3E13	160	LD
C367	D7	170	RST
C368	3E00	180	LD
C36A	D7	190	RST
C36B	3E14	200	LD
C36D	D7	210	RST
C36E	3E00	220	LD
C370	D7	230	RST
C371	3E15	240	LD
C373	D7	250	RST
C374	3E00	260	LD
C376	D7	270	RST
C377	3E16	280	LD
C379	D7	290	RST
C37A	78	300	LD
C37B	D7	310	RST
C37C	79	320	LD
C37D	D7	330	RST
C37E	7A	340	LD
C37F	D7	350	RST
C380	C9	360	RET
C381	0600	370	FEJ
C383	0E00	380	LD
C385	1690	390	LD
C387	0D53C3	400	CALL
C38A	0C	410	INC
C38B	1691	420	LD
C38D	0D53C3	430	CALL
C390	0D	440	DEC
C391	04	450	INC
C392	1692	460	LD
C394	0D53C3	470	CALL
C397	0C	480	INC
C398	1693	490	LD
C39A	0D53C3	500	CALL
C39D	C9	510	RET

Pass 2 errors: 00

ELE C350 FEJ C381  
PRINT C353

Table used: 45 from 176  
Executes: 50000

nátái a bal felső karakterre érvényesek. A D-regiszter az aktuális karakter ASCII kódját tartalmazza. Ezeket az értékeket is tetszés szerint változtathatjuk.

**A program működése.** A program elugrik a FEJ nevű főprogramra, ott beállítja a koordinátákat és az első karaktert, majd hívja a PRINT szubrutint. Itt az akkumulátorregiszterbe sorra betöltjük a megfelelő vezérlőkarakterek ASCII kódját, majd az RST 16 utasítással értelmezi a gép, ezután a vezérlőkarakterekhez tartozó értékekkel végezzük el ugyanezt a műveletet. Aztán visszatér a főprogramra, ami változtatja a koordinátákat és az aktuális karaktert, majd hasonlóképpen még háromszor elvégzi a fenti műveleteket. E program mintájára írhatunk tetszőleges karaktermátrixot, kiírató rutint.

Mivel a program nem tartalmazza a képernyőmegnyitást, a program meghívását a következőképpen végezzük: PRINT: RANDOMIZE USR cím. A program bárhová elhelyezhető. A rutint a GENS3 nevű assembler editor programmal írtam.

A listához magyarázatként annyit, hogy az első oszlop hexadecimálisan a memóriacímeket, a második oszlop szintén hexadecimálisan a programot, a harmadik oszlop a sorok sorszámát, a negyedik oszlop a címkeket, az ötödik pedig magát a programot tartalmazza.

A program mentése:

SAVE „file name” CODE cím, 78.

BOREK ANDRÁS



**PRO-KONTRA GM.**

1074 Budapest,  
Csengery u. 7. fszt. 1/a.  
☎ 417-893

### Cégünk ajánlataiból:

- 64 k-s memóriabővítő Commodore C16-hoz, 4000 Ft/db.
- Komersz személyi számítógépek és összes tartozékaik garancián túli javítása átalánydíjas szerződéssel is.
- FAST—VC—1541 kommunikáció-gyorsító rendszer a C64-hez (minden gép—floppy közötti és floppy belüli műveletet 4-12-szeresre gyorsít).

### Konfiguráció beszereléssel együtt 7000 Ft.

- Abszolút programvédelem C64-hez: cartridge-ben, műgyantával kiöntött EPROM-ba égetéssel, MÁSOLHATATLANNÁ teszi programjait.
- IEC buszról vezérelhető méréspontváltó egység 2 x 30 vagy 1 x 60 bemeneti, 2, ill. 4 kimeneti csatornával.
- Digitális kijelzésű elektronikus óra 8 x 14 cm-es digitmérettel, különféle színekben.
- Egyéb, közepes sorozatú fejlesztések igény szerint.

**Pro-Kontra Automatizálási,  
Műszaki Tanácsadó  
és Közvetítő GM.**

Budapest, Csengery u. 7. fsz. 1/A 1074

Tel.: 417-893

Levél cím: Budapest, Pf. 72. 1581



# A MINŐSÉGÜGY KÖZÜGY

## A számítógép-minősítés zsákutcai

A számítógép-minősítés feladatának megoldása során egy tetszetős és egyszerű formulával kezdjük a munkát. Az alaposabb vizsgálat azonban kiderítette, hogy ha megbízható minősítésre törekszünk, sokkal több jellemzőt kell figyelembe venni, mint ami a formulában szerepel. Nem túlzás, ha egy megnyugtatóan megalapozott minősítéshez száz vagy még több tényezőről kell kielégítő pontosságú információt megkövetelni. Ez a nagy szám pedig már más kérdéseket is felvet, amelyek közül kettő lényegbevágóan fontos.

Lehetséges-e egyáltalán tetszetős és egyszerű minősítő formulát szerkeszteni száz vagy még több jellemzővel?

Lehetséges-e egyáltalán ennyi adatot beszerezni? (Ha ezek egyáltalán rendelkezésre állnak valahol, vagy kimunkálhatóak valami módon.)

Az első kérdésre a felelet egyértelműen az, hogy ilyen sokváltozós formula legalábbis nehezen áttekinthető, tehát semmiképp sem lehet tetszetős és egyszerű. Minden fontos tényezőzt számításba vevő tetszetős és egyszerű formula tehát nincs. Ilyet ne is keressünk!

A második kérdésre is negatív a válasz. Még ha rengeteg pénzünk volna, akkor sem tudnánk minden adatot beszerezni, például egy új számítógéptípusnál az olyanokat, amelyek a tartós használatra vonatkoznak. (Mivel pedig általában tartós használatra szokunk számítógépet vásárolni, ezek a jellemzők egyértelműen a legfontosabbak közé tartoznak.)

A minősítésnek az az útja, amit eddig követtünk, elméletileg nyilvánvalóan zsákutca. Vannak azonban rövidebb és hosszabb zsákutcák. Vannak olyan hosszúak is, amiknek nem is látjuk a végét, és ezért nem is gondoljuk róluk, hogy zsákutcák.

Tanulmányozzuk egy kicsit a gyakorlatot! Rögtön rájövünk, hogy a gyakorlati ember bizonyos bölcsességre kényszerül. Az elméleti zsákutca közül a gyakorlat mindig a számára elérhető leghosszabb zsákutcaiba igyekszik bejutni.

A minősítés az élet kikerülhetetlen és állandóan elvégzendő művelete. Csinálni kell, akár zsákutca elméletileg az egzakt megvalósítás, akár nem. Adatok százait beszerezni — sok okból — lehetetlen. Adatok százaival sok változatot végigszámolni olyan kényelmetlen, hogy nem is csinálja senki.

És mégis folyik a minősítés, mindenhol és minden időben — nem ritkán egészen jól is.

Annak érdekében azonban, hogy a jó minősítések fortélyait megismerhessük, vagy egyáltalán csak észre is vegyük, szembe kell néznünk néhány káros előítélettel.

Az első előítélet az, hogy minden előítélet rossz. Ez nyilvánvalóan nem igaz. (Tipikus példa hibás előítéletekre.) Az előítélet kielégítő előzetes vizsgálat nélkül hozott ítélet. Lehet, hogy igaz, lehet, hogy hibás, lehet, hogy részben igaz, részben hibás.

Egy sok éve jó hírű gyár termékét — részletes vizsgálat nélkül, előítélettel! — jobbnak tartani, mint egy sok éve rossz hírű gyárét? Tegye a szívére a kezét az e sorokat olvasó gyakorlati számítástechnikus és nyilatkozzon! Megvetendő előítélet ez, vagy inkább szakmai tapasztalat?

A szakmai tapasztalatok alapján való döntések mindig előítéletek. Azok az álláshirdetések például, ahol 5 vagy 10 éves szakmai „gyakorlatot” követelnek, valójában nem a „gyakorlatra”, hanem a jó előítélő képességre tartanak igényt. Az ilyen szakember előző tapasztalataiból kialakított véleményei szükségképp előítéletek, és ha „objektív” információk beszerzésére nincs mód, az ő vagy más szubjektív véleményére kényszerülünk hagyatkozni, amit általában nem szívesen teszünk. Ennek oka egy másik helytelen előítélet: ami szubjektív, az szükségképpen hibás.

Manapság az objektívet a valóságossal, a megbízhatóval, a szubjektívet pedig az elfogulttal, a hibással, a valótlanul, a megbízhatatlannal azonosítják. Az objektív az igaz, a szubjektív a hamis szinonimája lett. (Talán mert túl sokszor bizonyultak hamisnak az emberek által származó információk?)

E hibás előítélet következetes és radikális alkalmazóinak eszébe sem jutott, hogy az embert és annak igényeit is érdeemben sóhoz kellene juttatni a gépek alkalmassága minősítése vizsgálatánál, hiszen az embert és igényét „szubjektív”-ként eleve nem egzakt, tudományosan kezelhetetlen-

nek tartották. Sajnos ez a szemlélet lett általános.

Az „objektív”-nek ebből, a hamis előítéletből származó téves értelmezéséből és kultuszából eredt aztán az a teljesen hibás gyakorlat, amely számítógépek alkalmasságának minősítésére, összehasonlítására, értékelésére szolgáló mérőszámokat szolgáltató formuláknak kialakításánál csak és kizárólag a számítógépek „objektív” jellemzőit szerepelteti. E tévhit szerint az alkalmasság, az alkalmazhatóság mértéke csak a géptől magától függ. Ez azonban nem így van. Nézzünk szerint, egy gyakorlatban valóban használható algoritmus (formula), amellyel egy konkrét gépnek egy konkrét felhasználó céljaira való alkalmasságának, megfelelőségének mértékét kívánjuk kiszámítani, csak olyan lehet, amelyben a gép jellemzői mellett helyet kapnak a felhasználói igények jellemzői is. Ha viszont egy minősítő formula (algoritmus) nem alkalmas konkrét gép, konkrét objektív és szubjektív igények kielégítésére való alkalmasságának mérésére, akkor joggal vonjuk kétségbe, hogy valamilyen értelmes célra egyáltalán használni lehet.

Ily módon tehát beszélhetünk a szubjektív igények kielégítéséről is. És ha portékánk megvásárlását az összes felhasználó szubjektíve a saját számára legjobbnak tartja, mi is legjobban utunk a sikerhez.

Ne kergessünk ábrándokat! Ne higgyük, hogy valamiféle általános, absztrakt alkalmasság létezhet. Csak konkrét gépek, konkrét (több-kevesebb mértékben szubjektív) célokra vonatkozó megfelelősége létezik. Nekünk konkrét feladatot (pl. gépvásárlást) kell megoldanunk. Mégpedig konkrétan, mert absztrakt feladatok absztrakt megoldásának értéke annál magasabb hatványa a zérusnak, minél általánosabb a „megoldás”.

Mi az, amit ezek után tehetünk?

Ha szerkesztünk egy olyan eljárást (formulát), amely a gépeket és az igényeket is figyelembe veszi (a kínálatot is és a keresletet is) megfelelően választott jellemzőikkel, akkor ennek gyakorlati használható-

sága — bonyolultsága miatt — rendkívül korlátozott lesz. Be kell érünk tehát azzal, hogy csak saját konkrét feladatunk megoldására való alkalmasság szempontjából foglalkozunk a gépekkel. Így a változók száma is lényegesen kevesebb lesz, és a számolási munkát is könnyen el tudjuk majd végezni. „Be kell érünk” — mondtuk. De hát lehet-e valakinek vagy valaminek más feladata mint az, hogy a saját feladatát minél tökéletesebben elvégezze?

E ponton nem tudjuk elfojtani azt a gyanút, hogy a túl sokat markolást, a túl általános feladatokkal foglalkozást egyese arra használják, hogy biztonságos távolságban maradjanak azoktól a konkrét feladatoktól, amik az ő feladataik lennének. Az „objektív”, különösen a nehezen vagy sehogyan sem beszerezhető „objektív” adatok igénylése jó eszköz arra, hogy a munka állása — a semmittevés — indokolva legyen, hiszen az „objektív adatok” hiányánál nyilvánvalóbb, munkát megghiúsító „objektív nehézség” nincsen. A túlzott objektivitásra törekvés pedig nem ritkán a tudáshiányt palástolja. Azok szoktak túl sok (és gyakran felesleges) „objektív” adatot igényelni, akiknek nincs kellő áttekintésük arról a területről, amelyen valamilyen feladatot meg kellene oldaniuk.

A helyénvaló és felelősségteljesen mérlegelt szubjektív ítélet és a felelőtlen hasraütéses döntés között óriási a különbség. Sajnos ma a legtöbb kár nem az — egyébként hibás — indokolatlanul „objektivitásra törekvésből”, hanem pont az objektív megalapozás elmulasztásában megmutatózó lelkiismeretlenségéből és tudatlanságból ered. Gyakran azt sem tudják, hogy mire vonatkozóan lehet és kellene objektív adatokat begyűjteni...

A következő konkrét lépésként tehát azzal kell foglalkoznunk, hogy hogyan szerkesztünk saját feladatunk megoldására alkalmas minősítési eljárásokat (formulákat).

Felkészülésként ajánljuk az olvasónak azt a gyakorlatot, hogy *hogyan adna tanácsot* a hozzá fordulónak, akik pl. gépkocsivásárlásban kérik a segítségét. Sok olyan hasznos dologra lehet e képzeletbeli feladattal való foglalkozás révén rájönni, ami közvetlenül hasznosítható a számítógépvásárlás (bérlet, bérletvétel) terén, de sok más területen is.



# A tűs mátrixnyomtatók

Az ókori Kínában már létezett a mai értelemben vett nyomtatás. A nyomtatni kívánt szövegeket, ábrákat fatáblákra faragták, majd ezekre festéket vittek fel, és papírra, textilre, bőrre stb. nyomva a táblát, vitték át azokra az információkat. Ez a megoldás művészileg ugyan magas színvonalú nyomásképet adott, de nagy hátránya volt, hogy hiba esetén nagyon nehéz vagy lehetetlen volt a szöveg javítása.

A XV. század közepén Gutenberg forradalmian új megoldással állt elő. Találmányának lényege az volt, hogy az egyes szavakat, sorokat különálló betűkből rakta össze. Így mód nyílt az esetlegesen előforduló szöveghibák egyszerű betűcserével történő javítására és a már használt betűk újbóli felhasználására is. A nyomtatási technológia ettől kezdve rugalmasabbá, könnyebben alakíthatóvá vált.

A nyomtatógépek cégek konstruktőrei napjainkban Gutenberg alapötletével analóg elvet hasznosítanak. Ha a szavak kisebb egységekre, betűkre bonthatók, miért ne volna lehetséges magukat a betűket is még további kisebb egységekre, „pontokra” bontani? Fölismerték, hogy bármilyen jel ábrázolható parányi, egymástól szabad szemmel alig megkülönböztethető, célszerűen elhelyezkedő pontok, (valójában kör alakú festékfoltok) halmazaként. A kérdés csak az, hogy hova kell a pontokat tenni ahhoz, hogy a kívánt jel minél könnyebben felismerhető legyen. E ponton közbeszóltak a technikai megvalósítás nehézségei, és a kérdésfeltevést úgy módosították, hogy lehet-e az összes jelet valamilyen egységes és egyszerű elv alapján, pontokból kialakítani. Kiderült, hogy ha a pontok lehetséges helyeit egy négyzetrácsnak egy téglalapon belüli rácspontjaira korlátozzák, a kitűzött feladat kielégítően megoldható. Az ilyen pontok úgy helyezkednek el, mint egy téglalap alakú mátrix elemei. A mátrixnyomtató neve is innen származik. Az 1. ábrán egy 8-szor 6-os „pontmátrixot”, a 2. ábrán pedig egy betűnek pontokból kialakított képét szemléltetjük.

A mátrix elrendezésű pontok többféle eljárással (például hő segítségével) vihetők papírra. Esetünkben ez vékony nyomótűk segítségével történik.

A tűs nyomtatók „lelke” a nyomtatófej. Több típusban és nagyságban gyártják. A fényképen egy nagyméretű fejet mutatunk be.

A nyomtatófej több részből áll (3. ábra). A fej belsejében kis vezetőkábeleket találunk. Ezek a vékony, kb. 0,1 mm vastagságú tűk vezetőkábelei. A tűk gyakori és nagyfokú igénybevételnek vannak kitéve, anyaguk ezért volfrám, amely elég kemény és rugalmas ahhoz, hogy a nagy mechanikai terhelést elviselje.

A tűk egymástól független, kellő időben való mozgását egymástól függetlenül vezérelhető elektromágnesek biztosítják, a rájuk adott rövid idejű feszültségimpulzusok segítségével.

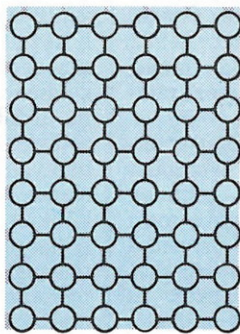
A tűk mozgása közvetlen és közvetett úton történhet. Közvetlen mozgáson azt értjük, hogy az elektromágnes húzóereje a tű végéhez szilárdan rögzített, lágyvasból készült, a tekercs belsejében elmozduló vasmagra hat.

Közvetett mozgatási mód esetén a tűket a nyomtatófej elektromágnes felőli végére

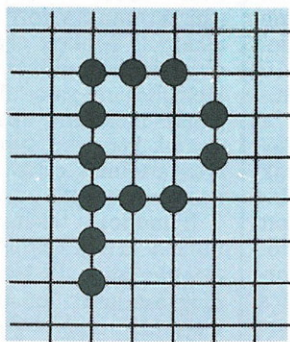
szerezt, egyik végükön rögzített, sugarasan elhelyezett fémlapocskák hozzák működésbe. Az elektromágnesek ezeket a lapocskákat húzzák meg, a tűknek pedig a lapocskák adják át a mozgást. A tűket mindkét esetben rugók térítik vissza alaphelyzetükbe. A tűk mozgáskor kb. fél milliméternyire futnak ki a fejből. Ez az úthossz elegendő ahhoz, hogy a festékszalagnak ütődve, nyomot hagyjanak a papíron.

A folyamatnak ez a része semmiben nem különbözik attól, mint amikor a hagyományos betűkaros írógépen leütve egy billentyűt, a kar végén levő betű hozzáüti (hozzányomja) a festékszalagot a papírhoz. Itt a fej előtt elvezetett festékszalagot a nyomtatófejből kiugró tűk (amelyek természetesen tompa végűek) nyomják hozzá a papírhoz.

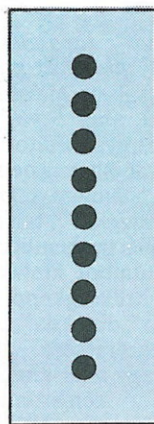
A tűket a fejből való kilépéskor igen pontosan kell megvezetni. A pontossági követelmény ezredmilliméter nagyságrendű, mert csak így biztosítható, hogy a tű mindig a megfelelő pontban érintse a festékszalagot. Az elmondottakból érzékelhető, hogy ennek a vezetőrésznek a pontossági



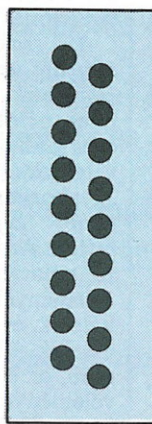
1. ábra



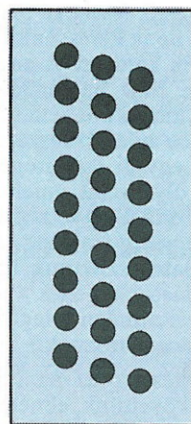
2. ábra



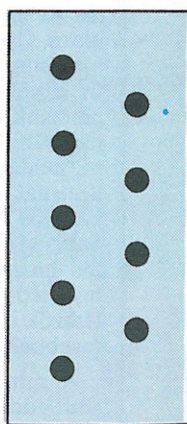
4. ábra



5. ábra

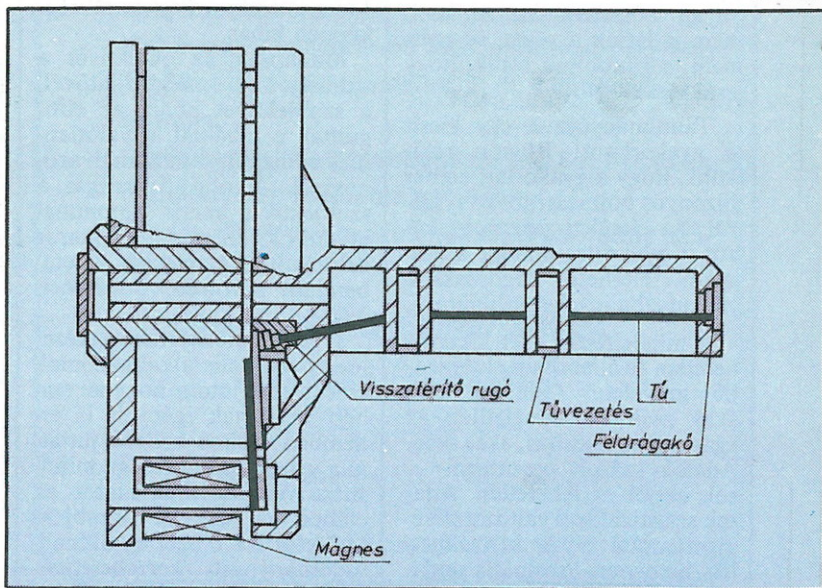


6. ábra



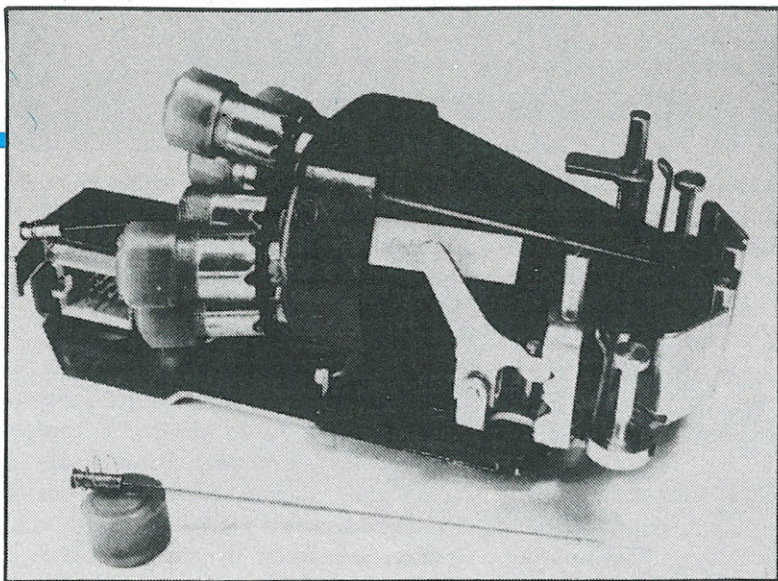
7. ábra

3. ábra



10. ábra





követelmény kielégítésén túl igen kopásállóknak kell lennie. A tűk vezetésére egy támasztólapocska szolgál, melynek anyaga a hagyományos órákból is ismert féldrágakő, a rubin. Igénytelenebb kivitel esetén a tá-

masztólapocska valamilyen kemény fémből is készülhet.

Ezzel a nyomtató mechanikai részének bemutatásával végeztünk is. A többi rész, a papírtovábbítás, a festékszalag-továbbítás

és a fejmozgatás terén lényeges újítás nincs. Jól érzékelhető a nyomtató mechanikai felépítésének lényeges egyszerűsödése. Ezt az egyszerűsödést az tette lehetővé, hogy számos, eddig mechanizmusokkal megvalósított művelet elektronikus eszközökkel vált elvégezhetővé. A legfontosabb és egyben legbonyolultabb — de mégsem bonyolult — funkció a tűk elektronikus vezérlése lett.

Az elektronika tárolja az egyes jeleket, azaz a nekik megfelelő bitmintázatot, ami a 2. ábra esetében a következő.

```

0 0 0 0 0 0
0 1 1 1 0 0
0 1 0 0 1 0
0 1 0 0 1 0
0 1 1 1 0 0
0 1 0 0 0 0
0 1 0 0 0 0
0 0 0 0 0 0

```

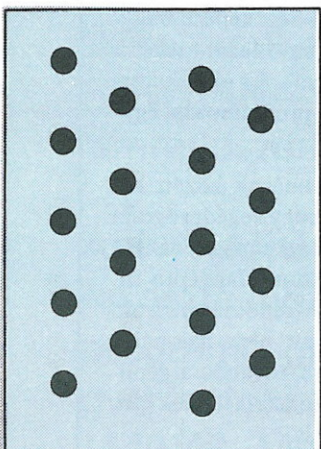
A nyomtató vezérlésében — igényesebb nyomtatóknál ez ma már általános — mikroprocesszor fogadja a központi egységből jövő információkat. Ellenőrzési, konverziós és kódolási műveletek után ez vezérli a tűmozgatást, és természetesen a fejnek és a papírnak a mozgatását is.

Ismerkedjünk meg kicsit közelebbről a vezérlés tevékenységével! Vegyünk egy egyszerűbb esetet, például, amikor 8 tű helyezkedik el egymás alatt, egy egyenes mentén. Egy betű (például a 2. ábrán levő) kialakításánál a vezérlés megkapja az információt, hogy P betűt kell nyomtatnia. Kikeresi a P betűnek megfelelő (tárolt) pontmintázatot, majd ezt oszlopról oszlopra haladva letapogatja, és az egyes oszlopok szerint működteti a tűket mozgató elektromágneseket és vezérli a fej helyzetének beállítását.

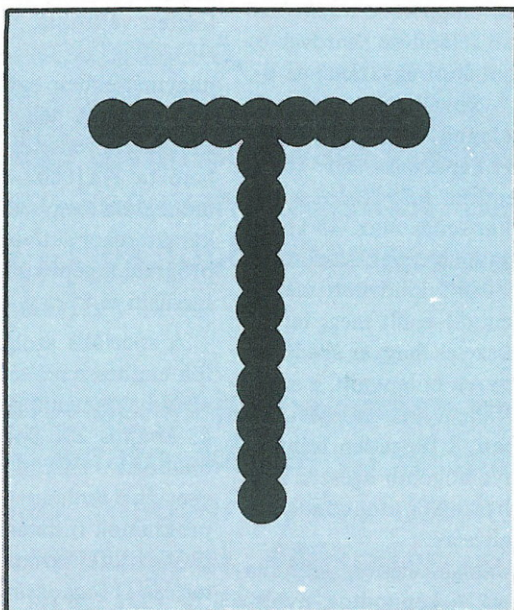
Az elv egyszerű, a működés gyors, csak az így kialakított betűk nem eléggé esztétikusak.

Sűríteni kellene a pontokat. A tűk (függőleges) sűrítésére egy határon túl azonban nincs lehetőség. A papírra kerülő túlnyomatok sűrítésére azonban mégis sikerült ötletes lehetőséget találni. Ugyanazon a soron többször végigviszik a nyomtatófejet, a későbbi végigvitelnél azonban egy kicsit lejjebb vagy feljebb állítják a fejet. Így olyan nyomtatási kép alakítható ki, mintha például nem 8, hanem 16 tű volna egymás felett. Belátható, hogy a pontok függőleges sűrítésének ily módon nincs korlátja. A vízszintes sűrítés pedig műszakilag olyan fejállítási, pozicionálási kérdés, amelynek megoldása nem okoz különösebb problémát.

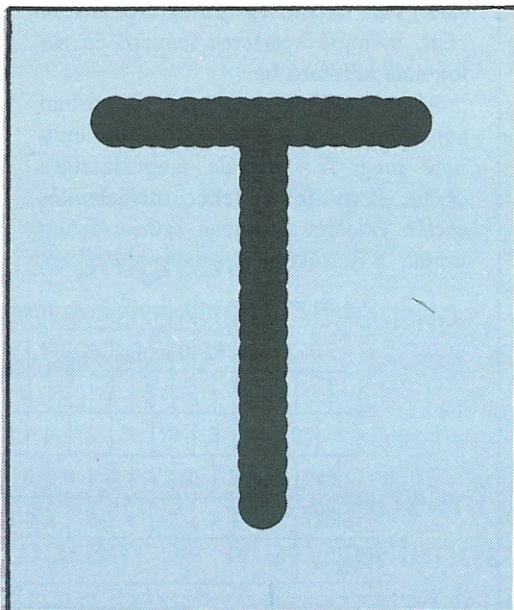
A fejnek a sorokon való többszöri végigvitele azonban számottevően csökkenti a nyomtatási sebességet, ezért a tervezők más megoldásokat kezdtek keresni. Új fejeket alakítottak ki, amelyek több tűoszlopot tar-



8. ábra



9. ábra



11. ábra



talmaznak. Sokféle túelrendezés van forgalomban. Ezek mind nagyon egyszerűek, egyik-másik azonban nem nélküli a szellemességet.

A 4.—8. ábrán öt túelrendezési típust mutatunk be. Ezek az adott sávzélességben rendre 9, 18, 27, 9 és 18 pontot képesek létrehozni a papíron, mégpedig a fej függőleges mozgása nélkül.

A 9.—11. ábrán egy T betű különböző sűrűségű nyomtatási képei szerepelnek. A 10. ábra pontsűrűsége a 9.-ének kétszerese, a 11. ábráé pedig a 10.-ének a kétszerese. Jól érzékelhető először a pontokból álló betű összefüggővé válása, majd pedig a határoló vonal kisimulása.

A tús mátrixnyomtatók műszaki jellegzetességeit áttekintve néhány felhasználással kapcsolatos kérdést kell még érintenünk.

Az első tús nyomtatót 1968-ban készítették egy asztali számítógéphez. 18 évvel ezután már számbavenni is lehetetlen a különböző tús nyomtatótípusokat. E termék — úgy tűnik — hosszabb időre befutott. (Örökös pozíciót jósolni neki nyilván senki sem merne. Máris komoly versenytársai vannak — lásd a  $\mu$ M 1986. februári számában a 24—25. oldalon szereplőket —, és nyilván lesznek még mások is a jövőben.)

Az egyes típusok teljesítményben, a szolgáltatott íráskép esztétikai értékében és ezektől függően az árban is nagyon különböznek egymástól. A legolcsóbbak 100—150 írásjelet nyomtatnak másodpercenként, a drágábbak ostromolják az 1000 írásjel/mp „bűvös” sebességhatárt.

A nyomtatási kép minősége is változó. Az olvashatóságot mindegyik eléri, vannak azonban olyanok, amelyekkel választékos igényű, nyomdai termék benyomását keltő oldalak is nyomtathatók, igényes tipográfia is módot nyújtanak. Vannak például különböző vastagságú, nagyságú és típusú betűk, használható görög, arab, sőt kínai írásjelkészlet is, és mindez különböző színekben.

A mátrixnyomtatók egy része nemcsak betűt nyomtat, hanem vonalakat, tartományokat is össze tud állítani pontokból. A technikai csúcs ma a színes, grafikus mátrixnyomtató. Ezzel már nemcsak hasznos műszaki ábrák, hanem hatásos „op-art” képzőművészeti alkotások is létrehozhatók. És mindez az elektronikának köszönhető.

Ne legyünk azonban igazságtalanok a mechanikával szemben sem! A legmagasabb színvonalú elektronika sem ér semmit, ha nincs körülveve hasonlóan magas színvonalú mechanikával. A tús nyomtatók mechanikája pedig nem akármilyen színvonalú finommechanika. Csupán érdekességként említjük, hogy egy nyomtatófej megtervezésekor nemcsak precíziós gyártási, különleges mérési problémákkal, hanem bonyolult differenciálegyenletekkel is meg kell birkóznunk. Ez utóbbi feladatot számítógéppel oldják meg, az eredményeket ki-nyomtatják, kirajzolják. Ennek pedig ma a legolcsóbb, legkényelmesebb, legelterjedtebb és állandóan fejlődő eszköze a tús mátrixnyomtató.

SZEBENSZKI SÁNDOR

## A Híradástechnika Szövetkezet új iskolaszámítógépe

A második iskolaszámítógép-pályázat — középiskolai kategóriában — helyezett gépe lett a HT3080C típusszámot viselő gép, melyet a Híradástechnika Szövetkezet gyárt. A pályázat eredményeképpen többek között ez a géptípus is hivatott arra, hogy a sikeresen megindult iskolai számítástechnikai program továbbviteléhez a megfelelő hardveralapot biztosítsa.

A pályázat ugyan nem tért ki erre, de igen fontos szempont, hogy a korábbi (HT1080Z/...) iskolaszámítógépre kidolgozott több száz program sorsa az új gép megjelenése után mi lesz. A benyújtott pályaművek között egyedül a HT-gép alkalmas arra, hogy változtatás nélkül fogadni tudja a régi gépre írt programokat.

Az új gép külső megjelenési formájában, mechanikai felépítésében nem változott meg. Belső felépítése (hardver és szoftver szempontjából egyaránt) az új, megnövekedett követelményrendszerhez igazodva átalakult. A Z80 alapú gép beépített RAM kapacitása 64 kb-ot, önálló videomemória 6,75 kb-ot, beépíthető ROM-kapacitás max. 48 kb-ot, mely a felhasználói igények szerint lapozható. A HT3080C kibővített tasztatúrával (74 billentyű) épült meg, tartalmazza a teljes ékezetes magyar ábécéhez szükséges karakterek billentyűit, a szabványban előírt billentyűelrendezés szerinti kialakításban. 8 független felhasználói (USER) nyomógomb egészíti ki a tasztatúrát a felhasználói programok kezelésének elősegítésére.

A nagy felbontású, színes, grafikus képezelés 192 x 256 képpontos, nyolcféle háttér- és rajzolószín alkalmazására van lehetőség karakterpozícióként, továbbá kétszeres fényerő és villogtatás állítható be.

Alfanumerikus kijelzőskor 24 sorban soronként 32 karakteres szöveg jeleníthető meg. A színskála megválasztása olyan, hogy fekete-fehér megjelenítés esetén lineáris elosztású szürke skálát mutat. Kijelzőként a rendszerhez kap-

csolható fekete-fehér vagy színes tv-vevőkészülék, fekete-fehér vagy színes videomonitor vagy RGB-monitor.

Az alkalmazott hanggenerátor a korábbi változatban már üzemelő 3 hang- és egy zajcsatorna programból történő kezelését megvalósító eszköz (11 oktáv, szelektív hangerő és burkológörbe programozhatóság). Az új gép széles körű periféria csatlakoztatási lehetőségekkel rendelkezik. Alapkiépítésben a Z80-bus csatlakozás mellett két 8 bites I/O csatorna, „Centronics” nyomtató csatlakozás és lokális hálózati csatlakozás lehetősége adott. Opcionálisan kiépíthető fénytoll, botkormány, speciális soros (Commodore), szabványos V24 és floppy drive csatlakozás.

A HT3080C szoftver felépítésében a korábbi BASIC interpreter továbbfejlesztett változata, új — teljes képernyő — editor és további, a felhasználást nagymértékben egyszerűsítő elemek találhatóak. A teljes ékezetes karakterkészlet közvetlenül a billentyűről kezelhető (a HT1080Z/64-hez képest nagymértékben leegyszerűsítve lehet ezeket a karaktereket aktivizálni). Az új monitorprogram a gépi kódú programozást egyszerűsíti le.

A speciális szolgáltatások között külön említésre méltó, hogy segédprogram alkalmazásával a gép alkalmassá válik a 48 kb-ot ZX Spectrum alapgépre írt, önállóan működő (rendszerprogram speciális szolgáltatásait nem használó) programok futtatására, továbbá a géptípushoz kifejlesztett perifériaelemek (Interfész I) fogadására is.

A HT3080C iskolaszámítógép kialakítása összességében illeszkedik a korábban megindított számítástechnikai oktatási programhoz, és messzemenően figyelembe veszi a továbbfejlesztési célkitűzéseket. Az új gép széles körű alkalmazásba vétele feltételezhetően enyhíteni fogja a program végrehajtását nehezítő géphiányt, és talán segít a kicsit heterogén géppark egységessé tételében is.

A HT3080C billentyűzete a magyar ékezetes karakterekkel

	F1	F2	F3	F4	F5	F6	F7	F8	BRK	CLR	Page	Tape
!	"	#	\$	%	&	'	(	)	0	Ö	Ü	Ó
1	2	3	4	5	6	7	8	9	0	Ö	Ü	Ó
!	Q	W	E	R	T	Z	U	I	O	P	Ö	Ü
CAP	A	S	D	F	G	H	J	K	L	É	Á	Ü
SHIFT	Y	X	C	V	B	N	M	<	>	?	+	=
USR	↑	↓										



# TPA, SZM4, Commodore-64 felhasználók figyelmébe!

*A programokat forgalmazza az*



*Alkotó Ifjúság Egyesülés*

*Számítástechnikai Iroda*

*Telefon: 112-666*

*Levélcím: 1519 Budapest, Pf. 330.*

*Telex: 22-7272*

**SELECT:** paraméterevezhető rendszerprogram. Alkalmas bármely op. rendszerben létrehozható, bármilyen elérésű adatfájlok adott logikai szempontok szerinti szétválogatására, gyors ad-hoc jellegű statisztikák készítésére, az adataiból kiküszöbölésére, javítási idő nagymértékű csökkentésére, BATCH üzemmódban is működtethető.

**MULTIFMS:** célja olyan software környezet biztosítása a TPA felhasználó részére, melyben könnyen készíthet Basic-Plus 2, FORTRAN-IV, illetve FORTRAN-IV-Plus többfelhasználós adatgyűjtő, rögzítő, lekérdező programrendszer(ek)e)t.

**AIE-HAHO:** univerzális párbeszédés, többterminálos ad-hoc lekérdező, naprakészre hozó, listázó, képernyő formában tervező és diagrammegjelenítő programcsomag. Magyar nyelvű utasításkészlettel, hibaüzenetekkel és magyarázatokkal rendelkezik. Működési környezet: PDP 11-es gépcsalád, TPA 1140/48, 440, JANUS, SZM4 és bármelyik RSX, DOS-RV, RSTS/E op. rendszer.

**Commodore-64:**

Létesítményjegyzék-készítő program

Rendelés és készletnyilvántartó programcsomag

„CROSS” keretszámító program

- Síkbeli keretszámító program
- Késedelmi kamat számítása
- Főkönyvi kivonatkészítő program
- Mérlegellenőrzés
- Expander programozási segédlet
- Gépipari tervezőprogramok



## . . . egy könyvet (Fertig: *The software revolution, North-Holland, 1985.*)

mely kétszáz, különféle típusú, 1984-ben erősen reklámozott rendszer elemzése alapján von le következtetéseket többek között a mikroszámítógépek piacára vonatkozóan. Az anyaggyűjtést 1984 végén zárták le, és arra számítanak, hogy a könyv mintegy öt évig megőrzi aktualitását. Az elemzés szerint a piac a következő szektorokra oszlik:

- 6502/BASIC alapú rendszerek,
- Z-80/Z-800/Z-8000/CP/M alapú rendszerek,
- 8088/8086/80286/MS-DOS alapú rendszerek,
- 80286 és 68000/UNIX és XENIX alapú rendszerek,
- 80386/Smalltalk-like rendszerek,
- Lisp (PROLOG) alapú rendszerek.

Egy-egy szektoron belül a piac „rétegek”-re osztható: integrált szoftvereszközök, operációs rendszerek, gépek, mikroprocesszorok. Minden egyes rétegre jellegzetesen más-más típusú eladók szakosodnak. A VisiCorp például elsősorban szoftvereszközökre, a Microsoft főleg operációs rendszerekre, az IBM elsősorban gépekre és az Intel leginkább mikroprocesszorokra specializálódott.

Az integrált szoftvereszközök 1982-ben jelentek meg a piacon. Ezek a piaci előrejelzés szerint fokozatosan felváltják, kiszorítják az egy-egy alkalmazásra erősen specializálódott, ún. „vertikális” termékeket és az egyedül megoldásokat. Az integrált eszközök első jellegzetes példájának a Lotus 1-2-3 tekinthető, mely jelentősen hozzájárult az IBM PC sikeréhez.

Az integrált szoftvereszközök fő jellemzői: (1) az összes eszköznek azonos stílusú, konzisztensen felépített felhasználói interfésze van; (2) több különböző eszköz ada-

mai egyidejűleg jeleníthetők meg; (3) az adatok bármikor könnyen mozgathatók az eszközök között; (4) kis és nagy mennyiségű adat egyformán könnyen mozgatható. A jelenlegi, integráltnak szánt termékekben ez a négy jellemző még ritkán található meg együtt. Az eszközök ilyen értelmű integrálódása azonban várhatóan az évtized végére uralkodó lesz.

A különböző funkciókat ellátó szoftvertermékek egységes rendszerbe történő integrálása többféle módon valósítható meg: vagy az operációs rendszer, vagy valamilyen magas szintű nyelv, vagy az adatbáziskezelő rendszer, vagy az alkalmazási programok keretében (az utóbbi példája a Lotus 1-2-3), vagy a közösen használt fájlok segítségével.

Az integrálás megvalósításának jövőbeli legvalószínűbb eszközei az operációs rendszerek lesznek. A UNIX és a Macintosh OS mutatja az ilyen lehetőségeket. A jelenleg használatos operációs rendszerek körülményes, utólagosan „rácsavarozott” kiegészítéseikkel ma még általában nem nyújtanak megfelelő támogatást valódi integrált szoftvereszköz kialakítására.

A piac összes szeletének bemutatására nincs módunk, ezért csak jelzésszerűen utalunk a házi számítógépek piacának várható alakulására, és kicsit részletesebben kitérünk az MS-DOS, illetve a Lisp (PROLOG) alapú két gépkategóriára (piacszeletre).

A házi számítógépek piaci részesedése 1984-től 1988-ig 38%-ról várhatóan 47%-ra nő. A hagyományos játékgépek gyártói (Atari, Coleco, Commodore stb.) megpróbálnak behatolni a nem játék jellegű házi számítógépek piacára. A videojáték-piacról

már megállapítható, hogy csak múltó szezőly volt, melyre nem lehet új iparágat alapítani.

A 8088/MS-DOS alapú rendszerek gyors elterjedése az IBM 1981. augusztusi, piaci belépésének köszönhető. 1984-ben már legalább 60 cég kínált kompatibilis rendszereket. A szoftverpiacon ugyanekkor több mint tízezer szoftvertermék volt kapható, és a számítások szerint jelenleg átlagosan minden tizenegyedik percben elkészül egy-egy újabb szoftvertermék erre a kategóriára.

A siker egyik fő forrása a „nyitott architektúra” volt, mely lehetővé tette a másolást és így azt, hogy a vásárló ne legyen kiszolgáltatva egyetlen eladónak. A Microsoft cég — amelytől a gép operációs rendszerét az IBM megrendelte — az MS-DOS operációs rendszerrel a 16 bites gépek piacának 60%-át hódította meg. Olyan mikroszámítógépes cégek, melyek nem voltak hajlandók hordozhatóságot garantáló szabványokat vagy nyitott architektúrát biztosítani a felhasználónak, visszaestek.

Az ebbe a kategóriába tartozó gépek Intel 8088, 8086 vagy 80286 típusú mikroprocesszorra épülnek, operatív táruk 64 kb-ot RAM-tól 1 Mb-ot RAM-ig terjed, 5 1/4"-os hajlékonylemezzel dolgoznak, 160 kb-ot vagy 320 kb-ot kapacitással. Az idők folyamán a kategória perifériakészlete bővült, és kiegészült többek között merevlemezzel.

Az IBM meglehetősen szoros termék- és árpolitikát diktált, melyet a kompatibilis típusok gyártóinak is követniük kellett: ha az IBM valamit változtatott, a kompatibilis gépek gyártóinak a túlélés érdekében gyorsan kellett reagálniuk.

Az IBM PC-k továbbfejlesztésének

# OATORG

Felajánlunk megvételre 1 db FELLER OCR-2002 típusú optikai bizonylat- és oldalolvasót, BPI mágnesszalagegységgel, valamint a berendezés tartozékait:

**1 db IBM gömbfejes írógép, 2 db OCR betűtípusú gömbfejjel, 1 db bizonylatbélyegző, valamint tartalékalkatrészek**

Érdeklődni lehet Perity László munkatársunknál  
Bp. V., Dorottya u. 6. Telefon: 184-055/1429-es mellék



kulcseleme az Intel 80286 típusú mikroprocesszor, mely a teljesítményeszek szerint jobb, mint a Motorola 68000-es (32 bites belső architektúra, 16 bites külső busz). A 68020-as lesz a Motorola valódi konkurens ajánlata. Az IBM PC/AT „szíve” az Intel 80286, melynek nagy tömegű előállítása kb. egy évvel megelőzte a Motorola 68020-ast. A 80286 ezenkívül kompatibilitást biztosít a 8088-cal, lehetővé téve ezzel az MS-DOS-ra írt hatalmas mennyiségű szoftver változtatás nélküli átvételét. Az IBM az Intel részvényeinek jelentős részét, 19%-át és ennek a mikroprocesszornak a gyártási jogát megvásárolta, és várhatóan a 286-os chipek nagy részét is fel fogja vásárolni. Jellemző adat, hogy például az Inteltől 1984-ben nyolcmillió darab 80186-os típust rendelt, miközben gyártási kapacitása csak egymillió körül volt. Hiány várható ennek alapján a 286-os chipek piacán is, mivel ennek a nagy sűrűségű chipnek a kihozatala alacsonyabb. Ha az IBM-nek sikerül rátennie a kezét a PC/AT érdekében a 286-os áramkörökre, akkor ezzel nehéz helyzetbe hozhatja a kompatibilis gépek gyártóit.

A 286-osnak is megvannak azonban a maga korlátai. A 16 bites beviteli/kiviteli adatút nem látszik elégségesnek a multitasking, valós idejű grafikus adatbevitel, egyidejű adattovábbítás stb. számára. Ennek ellenére úgy tűnik, hogy az IBM az évtized végéig kiköt a 286-os típus mellett, mielőtt valóban áttérne a teljesen 32 bites mikroprocesszorokra. Hogy mi lesz a valódi 32 bites operációs rendszerrel, azt csak találgatni lehet. A találgatást megkönnyítheti néhány cég, a Microsoft, az AT & T, az Apple/Xerox lépéseinek elemzése.

Az IBM jelentős erőfeszítéseket tett a PC forgalmazása érdekében: elsőként hirdett például személyi számítógépeket olyan általános, nagy példányszámú lapokban két teljes oldalon, mint a Time vagy a Newsweek. A repülőgéptársaságok saját kiadású utastájékoztató magazinjai valósággal hirdetési harctérre változtak. A hirdetések el-

sősorban a nagy számítóközpontok vezetőit vették célba, de felhívták a figyelmet arra, hogy a PC-k a kisebb vállalkozásokban is terjednek.

Az utolsó piaci szektorban helyezkednek el a legrágább és a legnagyobb teljesítményű gépek, melyek a mesterséges intelligenciával (AI-val) foglalkozó kutatói környezetből származnak. 1980-ig az AI-val foglalkozók még nem kínáltak gyakorlatban megvalósítható alkalmazásokat, illetve a kísérletek igen nagy számítógépes teljesítményt igényeltek. 1984-re ez a helyzet megváltozott: úgy látszik, egy sor üzletképes megoldás jött létre kevésbé költséges hardveren. A kutatók és a fejlesztők négy fő területet céloztak meg. Ezek a tudás-reprezentáció (szakértői rendszerek); a természetes nyelvek megértése; a beszédfelismerés és az alakfelismerés. E négy terület napjainkban művelhetővé válik mikroszámítógépek segítségével is.

A mesterséges intelligencia kutatása szorosán összekapcsolódott olyan magas szintű nyelvekkel, mint az 1950 óta fejlesztett Lisp. A Lisp versenytársa lett az Európából az 1970-es évek elején elindult PROLOG nyelv. A japánok az 5. generációs kutatási projektjük számára a PROLOG-ot választották.

A kereskedelmi szaksajtó sokat (egyesek szerint túl sokat) foglalkozik a mesterséges intelligenciával. Szinte minden, a számítástechnika jövőjével foglalkozó cikk a mesterséges intelligencia irányába mutat. Gyakran hangoztatott vélemény, hogy a mesterséges intelligenciára alapozott eszközök megengedik komplex problémák kevésbé strukturált megközelítését is, és ezzel segítik a termelékenység növelését.

## ...és árcédulákat is

(1986. márciusi, NSZK bolti árak)

Lapunk olvasói érdeklődnek házi számítógépek külföldi árai iránt. A szerkesztőbizottság vezetője egy olvasói levélre adott válaszában nem közölt konkrét árakat. Most igyekszünk ezt részben friss adatokkal pótolni (az árak nyugatnémet márkában értendők; az MNB március 19-i közlése szerint 1 DM = 20,37 Ft):

Commodore C128 típus (kezelési útmutatóval)	748,—
Színes monitor Commodore-hoz, C-1901 típus	1098,—
Kétoldalas hajlékonylemez-meghajtó Commodore-hoz, 1571-es típus	848,—
Mátrixnyomtató + Commodore csatlakozóegység, Microline-182 típus	899,—
Szövegfeldolgozó program Commodore-hoz. Textomat Plus típus	99,—
Commodore alapgép. C64 típus	498,—
IBM PC (16 bites, 64 k RAM-mal, lemezegységgel)	4448,—
Tasztatúra	698,—
33 cm-es monitor	698,—
Monitoradapter	728,—
IBM PC II. grafikus nyomtató	2148,—
Sinclair Spectrum (48 k)	298,—
Philips MSX központi gép (80 k RAM, 32 k ROM)	548,—
Fekete-fehér monitor	318,—
Kazettás egység	189,—
— KE —	

## Felajánlunk megvételre

egy kifogástalan állapotú Siemens 404/2 típusú,  
16 munkahelyes adatrögzítő rendszert,  
mely az alábbi egységekből áll:

**1 db 64 kb-átos központi egység,  
2 db mágnesszalagegység + vezérlés,  
2 db mágneslemezegység + vezérlés és konzol  
A berendezéshez komplett rendszerszoftvert is adunk!**

Érdeklődni lehet Perity László munkatársunknál, Bp. V., Dorottya u. 6. Telefon: 184-055/1429-es mellék

# DATORG



## A játékprogramozás technikája

### A RAJZOLÁS KEZDETE

A 10. listán látható részletben beállítjuk a képernyő-üzemmódot (maximális felbontás, az első képernyőmemória-lapon kezdés), letöröljük a képernyőt és bekapcsoljuk a grafikus üzemmódot, mégpedig úgy, hogy fekete alapon fehérrel rajzol. Ezenkívül ismét beállítunk néhány alapparamétert. Ebben a részben új típusú utasítás nem fordul elő.

### Az álló háttér megrajzolása

A 11. listában előállítjuk a fák képét, és ezeket három különböző helyen elhelyezük. A rajzok megjelenítése itt már nem igényli a rajzok előállítását, csak a fák képeinek jobb tanulmányozhatósága céljából azt négyszeresére felnagyítottuk. Itt is és a továbbiakban is az egyes programrészletek futtatása a RUN260 beadásával történik.

A 17. ábrán látható fa előállításához a programot a következőképpen kell módosítani:

```
295DRAW"516":GOTO310
311STOP
```

A három fát tartalmazó képrészlet (18. ábra) előállítása:

```
295,311 nincs
321STOP
```

A 300-as, 310-es, 320-as sorokban kihasználtuk a DRAW utasításnak azt a tulajdonságát, hogy a karaktersorozatokat kapcsolatos műveleteket ezen az utasításon belül is végre lehet hajtani, ezért a TREE\$ karaktersorozatot összekapcsolhatjuk a DRAW utasításban megadott különböző kezdőhelyekkel, és így háromszor — a képernyő különböző helyein — tudjuk megjeleníteni ugyanazt a fát.

A 12. lista a háttér további részeinek megrajzolását teszi lehetővé. Ezek a részletek a következők: ellipszisrajzolás, a horizont megrajzolása, és végül a leszállópálya kiképzése. A 19. ábrán látható részletet a következő programmodosításokkal állíthatjuk elő:

```
321 nincs
265GOTO330
391STOP
```

Itt új utasítás a CIRCLE, melynek formája: CIRCLE (x,y),r,c,h,s,e, ahol x,y a kör (vagy ellipszis) középpontjának koordinátái, r a kör (ellipszisének az x tengelyirányú metszet) sugara, c a szín, h az y, illetve

x tengelyirányú metszet hosszának hányadosa (ha ezt nem írjuk be, értéke egy), s a görbevonallal részleges megrajzolása esetén a kezdő hely a teljes görbevonallal szögfokban mért hosszának hányadosában megadva, az óramutató járásának irányában mérve, és kezdőértéknek az x tengely pozitív irányát véve (ha nem adjuk meg, értéke 0), e ugyanilyen egységekben mérve a vége a görbevonallal szakaszának (ha nem adjuk meg értékét, úgy ez egy). A felsorolt paraméterek közül az első három megadása kötelező, a továbbiaké nem. Amennyiben az utolsó háromból legfeljebb egyet elhagyunk, úgy ezt a vesszők között, az utolsónál a vessző után szám ki nem írásával jelezzük. Egynél több paramétert nem hagyhatunk el, ha utána paramétert megadunk. Ha nem kívánjuk megadni a második paramétert sem, akkor annak az alapértékét kell megadni. Ezzel az utasítással nemcsak kört, hanem ellipszist, kört, illetve ellipsziszrészletet tudunk rajzoltatni.

### Változó háttér

Az egyetlen sorból álló 13. listában véletlenszerű elhelyezésben rajzoljuk fel a háttérre a „csillagokat”.

A 20. ábrán bemutatott „csillagkép” előállításához programunkat a következő módon kell módosítani:

```
391 nincs
265GOTO400
401STOP
```

### Szövegrajzolás

A 14. listában közölt részlet a 21. ábrán látható feliratokat (üzemanyag, élők) rajzolja fel a képernyő tetejére. Ennek láthatóvá tételére és jobb olvashatóságának biztosítására a következőket írjuk be:

```
401 nincs
265DRAW"58":GOTO410
421STOP
```

#### 10. lista

```
260 PMODE4,1:CLS:SCREEN1,1
270 X=100:Y=70
280 MC=0:MF=0
```

Ebben a részletben újfajta utasítás nem szerepel, legfeljebb az, hogy a DRAW utasítást szövegek előállítására használjuk.

### További paraméterbeállítás

A 15. lista programrészlete a további paraméterek értékét állítja be.

### Szövegháttér és rajzjelrajzolás

A 16. listában a 22. ábrán látható fehér sávot és felfelé mutató nyilat állítjuk elő. A módosítások:

```
265GOTO460
471STOP
421 nincs
```

### Kezdődik a játék

A 17. lista egyetlen utasítása a bevezetés befejeztével a játék során állandóan ismétlődő ciklusba ugratja a programot.

GALINA FERENC

```
290 TREE$="R1E1U11H1L1H2U2R1L2U1
E1U1L1U1E1U1E1U1E1R2F1D1E1U1E1R2
F2R2D1U1R1F1D1G2R1D1F1G1D2L1D2U1
L2D1G1R1G1D10F2L9;BR6BU13;L2U1;B
R2BU1;U1R1;BR2BU1;U2R1;BL4;D2H2;
BU2;L2U1G2;BD3;R2F2L1;G1;BU11BR3
;R2G2F1R1"
300 DRAW"BM30,165;" + TREE$
310 DRAW"BM55,170;" + TREE$
320 DRAW"BM205,171;" + TREE$
```

#### 11. lista

```
330 CIRCLE(230,175),20,,.2
340 LINE(0,0)-(256,192),PRESET,B
350 LINE(0,160)-(32,160),PSET
360 LINE(39,160)-(57,160),PSET
370 LINE(64,160)-(207,160),PSET
380 LINE(214,160)-(256,160),PSET
390 DRAW"BM94,160;G10R55H10"
```

#### 12. lista

#### 13. lista

```
400 FORS=1TO100:PSET(RND(256),RND(90)+22):NEXTS
```



## PRIMO

### Fordulj meg!

```
410 DRAW"BM10,10;U4R2L2U3R3;BR2;
D7R4U7;BR2;R3L3D3R2L2D4R3;BR2;R3
L3U7"
420 DRAW"BM215,16;D4R2BU4BR2D4U4
BR2D2F2E2U2BR2R2L2D2R1L1D2R2BU4B
R3R2L2G1F1R2F1G1 L2"
```

14. lista

```
430 FU=190
440 PC=1
450 DI=3
```

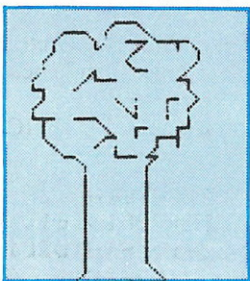
15. lista

```
460 LINE(30,4)-(189,9),PSET,BF
470 DRAW"BM88,12;F2H2G2E2D4"
```

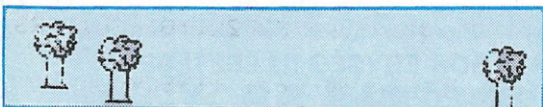
16. lista

```
480 GOTO570
```

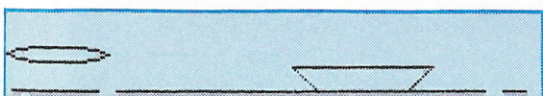
17. lista



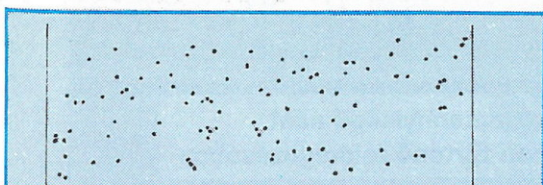
17. ábra



18. ábra



19. ábra



20. ábra

21. ábra

LIVES FUEL

22. ábra

A képernyő két oldalán egy-egy függőleges vonal van. A két vonal között vízszintes irányban jobbra és balra, változó sebességgel futó pontokat kell a vonal elérésében megakadályozni. A pontok az egy karakter beírását követő hangjelzés után irányt és sebességet váltanak. A pontokból álló mezőny több forduló után széthúzódik, és

ezért a vonal elérésének megakadályozása fokozatosan nehezedő feladat.

A számítógép a játék során olyan mérőszámot képez, amelynek növekedése a feladat nehézségéhez igazodik. A játék célja a nagy értékű mérőszámok elérése.

SOMOGYI GYÖRGY

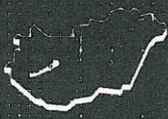
```
1 REM FORDULJ MEG
3 PRINT"A KEPERNYON FUTO PONTOKAT"
4 PRINT"EGY KARAKTER BEIRASAVAL"
5 PRINT "A VONAL ELERESE ELOTT KELL"
6 PRINT"ELLENKEZO IRANYBA KULDENI."
7 PRINT : PRINT "A JATEK EGY KARAKTER" : PRINT
  "BEIRASAVAL KEZDHETO EL."
8 REM A$= URES $ A 9. SORBAN
9 A$=INKEY$ : IF A$="" THEN 9
12 DIM X(16),M(16)
13 FOR D=1 TO 16
14 X(D)=120
17 NEXT D
18 PRINT CHR$(12)
20 FOR Y=5 TO 180
22 SET(10,Y)
24 SET(240,Y)
26 NEXT Y
28 M=1
46 FOR D=1 TO 16
47 REM U$(<) URES $ A 48. SORBAN
48 U$=INKEY$ : IF U$("<)" THEN M=M*-1 : BEEP 20,300
49 IF N(>)M THEN M(D)=2*RND(10)
50 M=M*-1
54 X(D)=X(D)+M*M(D) : X=X(D)
56 S=X-M*M(D) : F=X+M*M(D)
58 IF S<0 THEN S=1
59 IF F<0 THEN F=1
60 IF S>255 THEN S=254
61 IF F>255 THEN F=255
62 IF X<0 THEN X=2
63 IF X>255 THEN X=243
159 RESET(S,D*10) : RESET(F,D*10)
160 SET(X,D*10)
165 IF X<10 OR X>240 THEN 500
167 NEXT D
168 MU=MU+0.5
169 N=M : ER=ER+FIX(MU) : PRINT CHR$(6) : PRINT$ 0,0,ER
170 GOTO 46
500 PRINT "ELERTEM A VONALAT!" : ER=0 : MU=0,
632 INPUT "JATSZUNK MEG? (I/N)";V$
634 IF V$="I" THEN 13
```



# EUROPEAN GO CHAMPIONSHIP AND CONGRESS BUDAPEST '86

碁

SPONSOR  
SOFTINVEST



jul. 19. - aug. 2. ELTE KOLLÉGIUM BUDAPEST XI., BUDAÖRSI ÚT 95-101.

## A GO

A világ tekintete — és a miénk, magyaroké is — gyakran fordul Japán felé. Ezért is tarthat különleges érdeklődésre számot a nyár közepén Magyarországon első ízben megrendezésre kerülő GO Európa-bajnokság. A go-ról és az eseményről beszélgettünk dr. Vékey Károly 3 danos mesterrel, aki kétszeres magyar bajnok és a Magyar Go Egyesület elnöke, valamint Göndör András, 5 danos mesterrel, aki hatszoros magyar bajnok.

Kérésünkre Göndör András ismertette a go-játék legalapvetőbb elemeit. A játék több mint 4000 éves múltra tekint vissza. Kínai eredetű, ma Japánban űzik a legmagasabb szinten. Két játékos játsza, sok tekintetben hasonlatos a sakkhöz. A go-tábla 19x19 metszéspontjának valamelyikére felváltva helyezik el a játékosok fehér, ill. fekete színű kőveiket. A kővel (kövekkel) szomszédos metszéspontokat összekötő vonalakat a kő „életeinek” hívják. Ha az ellenfél körbeveszi a követ (kőcsoportokat), akkor azok elveszítik „életeiket”, így levehető a tábláról. A játék célja kettős: terület-szerzés és minél több ellenséges kő leütése.

Vékey Károly elmondta, hogy a világon kb. 30 millióan játsszák a go-t. Európában a századforduló táján vált ismertté, a II. világháború után kezdett szélesebb körben elterjedni. Magyarországon Erdős Pál matematikus vezette be. Aktív élet a 70-es évek közepén alakult ki. Eddig összesen mintegy 30 ezer go játékkészlet talált gazdára, de rendszeresen csak kb. félezen játszzák a játékot. Közülük — saját költség terhére — 40-50-en vesznek részt nemzetközi versenyeken.

Ennek ellenére igen eredményesen szerepelnek játékosaink. A XXX. GO Európa-bajnokság rendezési joga elismerése a magyar go-élet aktivitásának. Ez az első, szocialista országban rendezett EB, amin valószínűleg részt vesznek — első ízben — a jelentős játékerőt képviselő szovjet versenyzők is.

Az EB július 19-től augusztus 2-ig tart az ELTE Budaörsi úti kollégiumában, és rangját jelentősen

emeli majd a távol-keleti profi játékosok jelenléte. Az Európai Go Szövetség is ez idő alatt tartja évente szokásos ülését.

Legutóbb az I. Országos Mikroszámítógépes Találkozón láthatta a nagyközönség, hogy miként művelik a go-t a gyakorlatban. Joggal felmerül a kérdés, hogy milyen kapcsolat lehet a go és a számítástechnika között? A válasz: a go számítástechnikai vonatkozásairól először rendezünk konferenciát. Ennek során bemutatjuk a jelenleg létező go-programokat. Érdekes, hogy játékosaink nagy része matematikus vagy számítógépes szakember.

A go és a számítástechnika kapcsolatát jelzi az is, hogy a tavalyi EB fő sponzora az IBM volt, az idei EB-t pedig a SOFTINVEST karolta fel, nagy terhet levéve a szervezők válláról. A SOFTINVEST tette lehetővé a go részvételét a Mikroszámítógépes Találkozón.

Érdekes megvizsgálni, hogy milyen szinten játszik a számítógép go-t ma, amikor sakkbán már jelentős játékerőt képvisel. Göndör András véleménye szerint egy átlagos tehetségű játékos egy év alatt jut el arra a szintre, hogy megverje a ma legerősebb go-programot. Ennek az az oka, hogy a sakkhöz képest később kezdtek foglalkozni a go „gépesítésével”, így ezen a téren az eredmények még kezdetlegesebbek. A másik gond az, hogy a sakkot 8x8-as táblán játsszák, és viszonylag sok szabály korlátozza a bábuk mozgását. Egy játszma 40-50 lépésből áll. A go-tábla viszont 19x19-es, és általában 200 lépésig tart egy játszma. Ez már önmagában is megnehezíti a go programozását. Ide vág Erdős Pál véleménye: „a go minden bizonynyal a legbonyolultabb, polinomiálisan algoritmizálható játék”. A jövő valószínűleg sok újat hoz majd ezen a téren.

Az érdeklődők egyébként kedd esténként a Gorkij Könyvtár pinchehelyiségében, csütörtökönként az Eötvös Klubban ismerkedhetnek a go-val. Azokat is szívesen látják, akiket a go mint matematikai probléma érdekel.

AZ ORSZÁG MINDEN RÉGIÓJÁBAN ÉGSZI

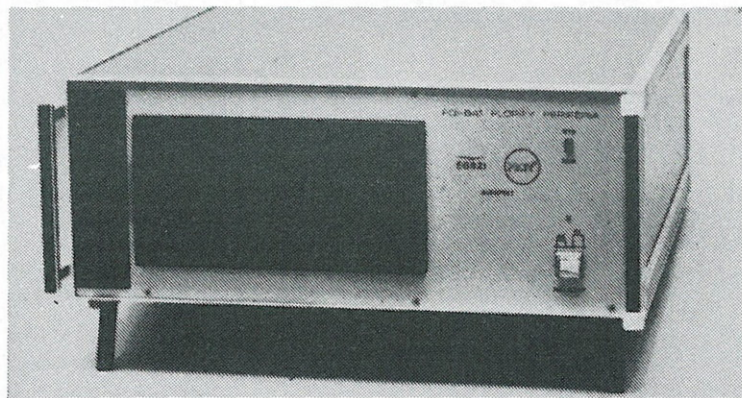
Az Építésgazdasági és Szervezési Intézet

## HARDVERAJÁNLATA



KÉPERNYŐS KONZOL  
ÉSZR számítógépekhez

### FLOPPYILLESZTŐ EGYSÉG



Az ESZR (és kompatibilis) nagyszámítógépek és mikroszámítógépek közötti offline kapcsolathoz

### MÁGNESSZALAGOS EGYSÉG ILLESZTÉSE IBM PC/XT-hez



Nagy mennyiségű adat  
mikrogépen történő feldolgozásához

TOVÁBBI TÁJÉKOZTATÁS:  
ÉGSZI Fejlesztési Főosztály  
RUTTKAY György főosztályvezető  
Telefon: 460-797

Cím: Budapest XI., Bartók Béla út 152.



# EGY JÓ DÖNTÉS

## AZ IBM KOMPATÍBILIS

# PROPER—16

### professzionális személyi számítógép

### BŐVEBB PROPER-VÁLASZTÉK!

**KORÁBBAN AZ SZKI-TÓL VÁSÁROLT GÉPEIBŐL SZEMÉLYI SZÁMÍTÓGÉPES HÁLÓZATOT ALAKÍTHAT KI.**

**RENDKÍVÜL JÓ ÁR/TELJESÍTMÉNY VISZONY!**

**A PROPER—16-OS HÁLÓZAT MEGTARTJA SZEMÉLYI JELLEGÉT, DE BIZTOSÍTJA A KITERJESZTETT ERŐFORRÁSOK KÖZÖS HASZNÁLATÁT. 12 TERMINÁLIG ALACSONYABB ÁRON NYÚJT NAGYOBB TELJESÍTMÉNYT, MINT EGY MINISZÁMÍTÓGÉP. NINCS SZÜKSÉG KÜLÖN SZEMÉLYZETRE, KÜLÖN GÉPTEREMRE, KLIMATIZÁLÁSRA!**

**NAGYOBB KATEGÓRIÁJÚ GÉPEKKEL ON LINE, ILLETVE OFF LINE KAPCSOLAT LEHETSÉGES.**



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

**Információ: SZÁMÍTÁSTECHNIKAI INFORMATIKAI  
FEJLESZTŐ  
LEÁNYVÁLLALAT (SCI—L)  
1011 Budapest, Iskola u. 10.  
Telefon: 350-180**



# PTA-4000

## Rendszerismertető

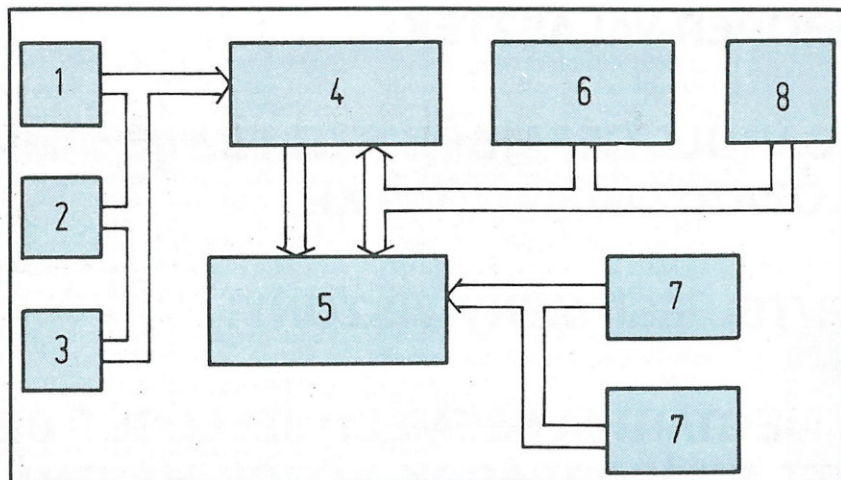
Az 1984 végéig megvalósított egységekből álló rendszer blokkvázlata látható az 1. ábrán. Az egységek: 1. 4 k RAM-bővítő; 2. 8 k RAM-bővítő; 3. programmodul; 4. PTA-4000; 5. négy színű grafikus nyomtató; 6. ún. szoftverkártya; 7. kazettaillesztők; 8. soros és párhuzamos illesztő.

A rendszer legfontosabb egységének, a PTA-4000-nek a blokkvázlatát a 2. ábra mutatja. 1. CMOS dekóder, a 74139 megfelelője, típuszáma

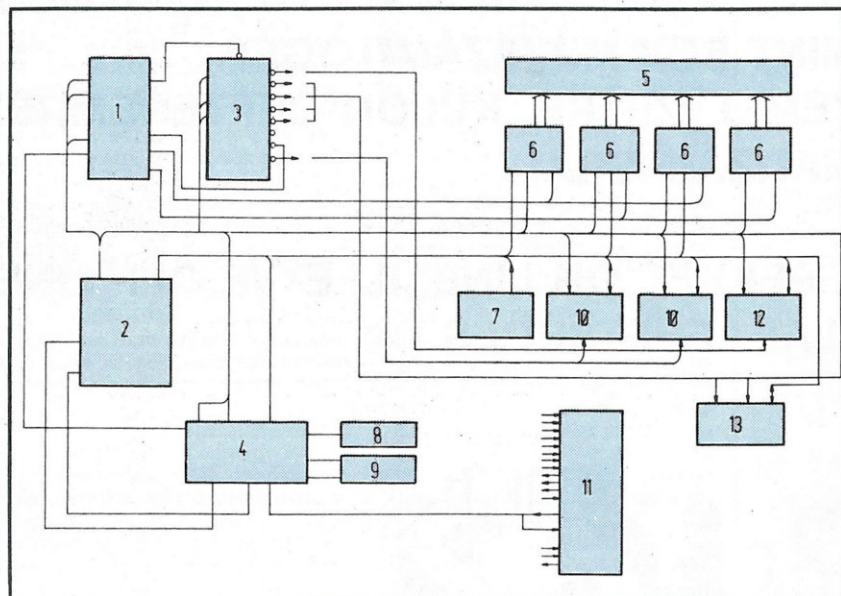
típuszám SC61328F; 8. hangkeltő körök; 9. számláló/időzítő, típuszám uPD1990AC; 10. 0,5 k RAM-ok, típuszám TC5514; 11. 60 vonalas csatlakozó; 12. felhasználói RAM, típuszám 6116; 13. 40 vonalas csatlakozó.

használt RAM-terület megoszlása a következő:  
7600H – 77FFH képernyőpuffer, a rendszer karakteroszorozat-változó

7900H – 79FFH rendszer-változók, rendszer-memória  
7A00H – 7AFFH rendszer-változók, a BASIC verme  
7B00H – 7BFFH karakteroszorozat-, ki/bemeneti pufferek  
A ki/bemenetek címei:  
F004H osztó, újabeállító;  
F005H U-regiszter kimenet;



1. ábra



2. ábra

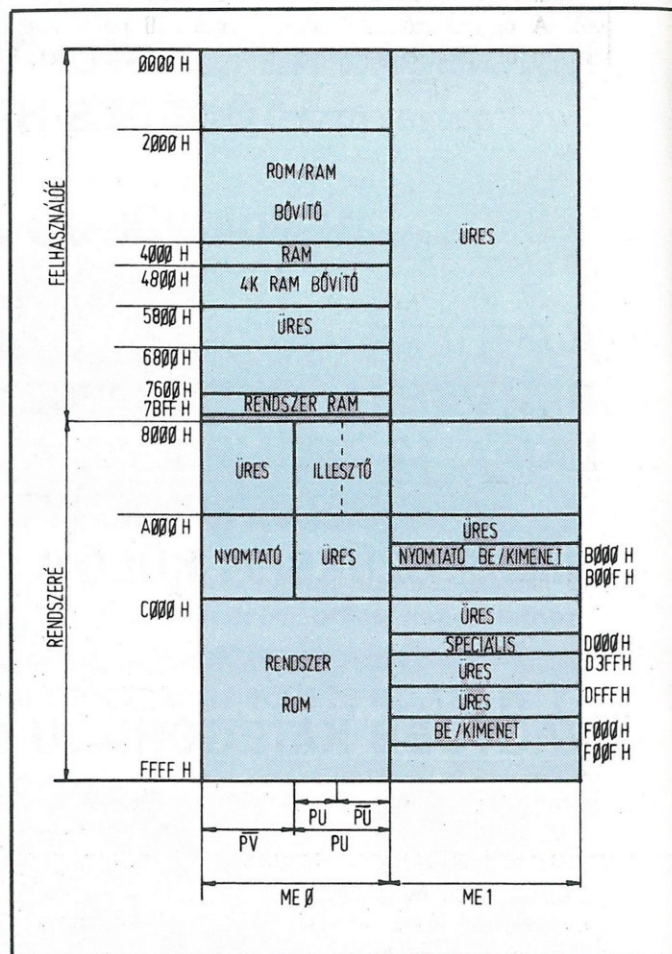
TC40H139F; 2. CPU LH5801; 3. CMOS dekóder, a 74138 megfelelője, típuszáma TC40H138F; 4. perifériakezelő LH5811; 5. folyadékkristályos kijelző, típuszáma LF8082GE; 6. kijelzőhajtók, típuszám SC-882G; 7. ROM,

A PTA-4000 memóriaterképe a 3. ábrán látható. Az első lap felső 32 k területén a PV- illetve  $\overline{PV}$ -bit értékétől függően két lapfél választható ki, PU-, illetve  $\overline{PU}$ -bit értékétől függően pedig ebből az alsó 8 k osztható meg. A rendszer által

7800H – 78FFH a rendszer verme, rendszer-memória, a rendszer-karakteroszorozat változó

F006H soros átvitel; F007H osztó; F008H C-port; F009H G-regiszter; F00AH MSK-regiszter; F00BH IF-regiszter; F00CH A-port irány; F00DH B-port irány; F00EH A-port; F00FH B-port.

DR. SIMONYI ENDRE



3. ábra



# ZX81 buszcsatlakozó

Ha gépünkhez bármilyen kiegészítést akarunk csatlakoztatni, szükségünk van egy buszcsatlakozóra. Célszerű a kiegészítést rövid szalagkábellel vagy egyéb hajlékony vezetékkel csatlakoztatni, mert így sok érintkezési hibát kerülhetünk el.

A ZX81 buszcsatlakozója közvetlenül a NYÁK-on kialakított, kétoldalas, ún. direkt csatlakozó. A csatlakozási pontok száma  $2 \times 23$ , ezek kiosztása a táblázatban található. A csatlakozó „A” oldala van felül (alkatrészoldal). A csatlakozó elkészítésének menete

zetégeket. A vezetékek legfeljebb 30–40 cm-esek legyenek.

7. Vizsgáljuk meg, nem lettek-e zárlat (ónátfolyás), és hogy az érintkezők tiszták-e. A vezetégeket igazgassuk el, és egyéni furfanggal tegyük esztétikussá a köteget.

*Néhány tanács az elkészült csatlakozóhoz*

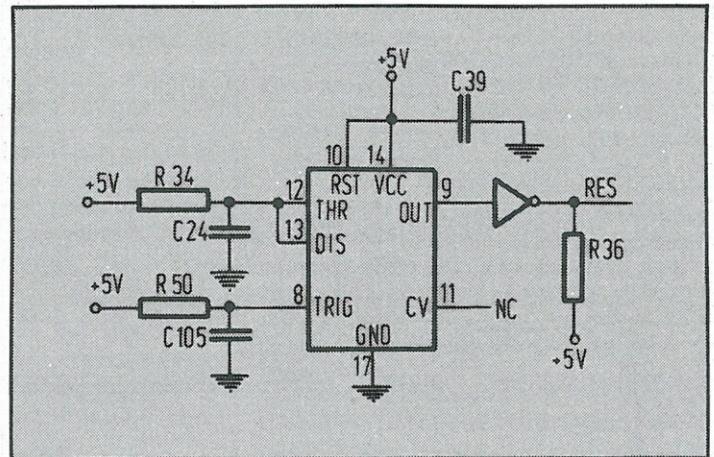
Lehetőség szerint minél kevesebbet mozgassuk a csatlakozót.

A csatlakozó megbontását és összedugását csak kikapcsolt állapotban végezzük.

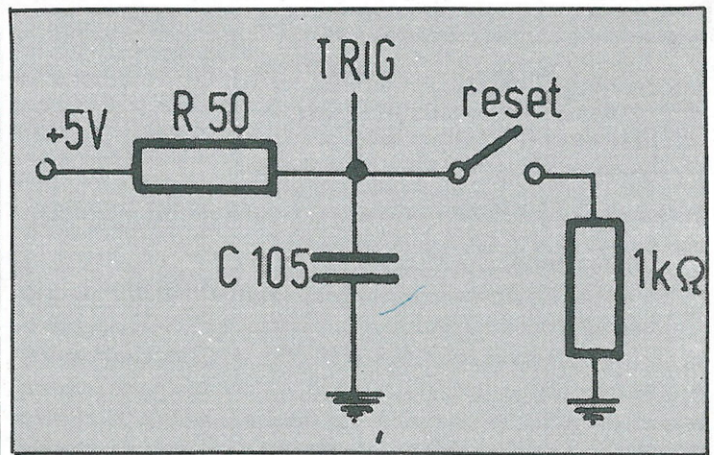
A csatlakozót, különösen üzem közben, ne érintsük meg.  
NAGY GÁBOR

# C64 hardver reset

Bizonyára sok érdeklődő olvasó szívesen látja, hogy a különböző programozási fogások után most a gép belsejébe is belekukkanthat. Aki találkozott már ABC-80-as vagy HT-1080Z számítógéppel, észrevehette, hogy a gépen van egy ún. reset gomb is. Ennek segítségével a gép tetszőleges helyzetből alapállapotba állítható. Ezt a funkciót a C64-esen a STOP és RESTORE gombok egyidejű lenyomásával valósíthatjuk meg. Gyakorlottabb felhasználó



1. ábra. A reset áramkör



2. ábra. Módosított reset áramkör

1. Veszünk egy tetszőleges, 23 pontnál nagyobb, kétoldalas direkt NYÁK-csatlakozót. (Pontosan ekkorát nem fogunk kapni!)

2. A csatlakozó két végéből levágunk annyit, hogy a megmaradt rész pontosan  $2 \times 23$  érintkezőt tartalmazzon.

3. Mindkét sorból kiemeljük a harmadik érintkezőt. Ezt fogóval, erős és jól irányzott nyomással lehet megtenni. Ügyeljünk a többi érintkező épességére!

4. A kiemelt érintkezők helyére egy szorosan illeszkedő szigetelőlemezt ragasztunk be. A lemezke ne lógjon ki a csatlakozóból, és a többi érintkezőt ne kenjük össze a ragasztóval.

5. Az ábra alapján elkészített NYÁK-ba forrasszuk be a csatlakozót.

6. A szükséges csatlakozási pontokhoz forrasszuk be a ve-

	A	B
1	D7	5V
2	RAMCS	9V
3	—	—
4	D0	0V
5	D1	0V
6	D2	0
7	D6	A0
8	D5	A1
9	D3	A2
10	D4	A3
11	INT	A15
12	NMI	A14
13	HALT	A13
14	MREQ	A12
15	IORQ	A11
16	RD	A10
17	WR	A9
18	BUSAK	A8
19	WAIT	A7
20	BUSRQ	A6
21	RESET	A5
22	MI	A4
23	RFSH	ROMCS

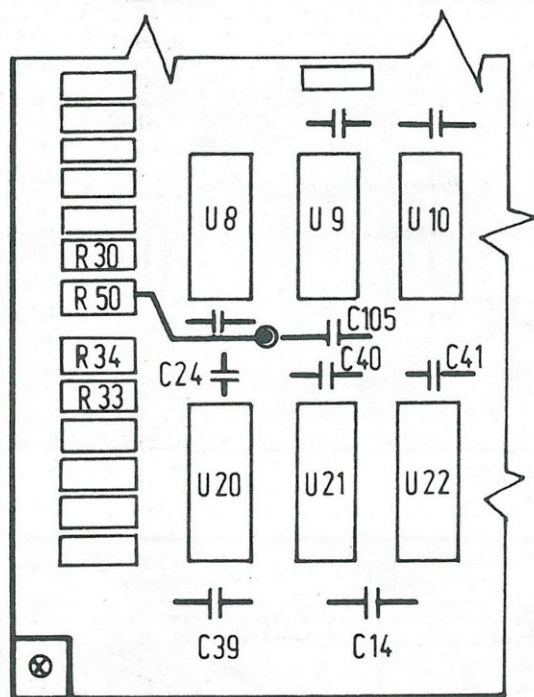
náló azt is tapasztalhatta, hogy ez a próbálkozás nem mindig vezet eredményre, mivel ez a lehetőség programból nagyon egyszerűen letiltható. Gyakrabban használható módszer a hardver reset alkalmazása.

A C64 áramköreinek és perifériáinak is van RES vezetéke. Ezek a vezetékek egy közös RES vonalra csatlakoznak (1. ábra), melynek aktivizálásakor, vagyis szintre juttatásakor minden, a vonalon lévő egység alapállapotba kerül. Ez a vezeték a soros buszon (drive csatlakozó) a bővítő és felhasználói porton keresztül is elérhető. A RES és GND csatlakozók pillanatnyi rövidre zárása a gép és a csatlakoztatott perifériák kezdeti, ún. inicializált állapotba jutását eredményezi.

Biztonságosabb és a gyári előírásoknak is jobban megfelelő megoldás a bekapcsolási reset felhasználása. Ennek ismertetése



előtt vizsgáljuk meg kicsit jobban a bekapcsolás pillanatát. A RES jel előállítását egy NE556-os időzítő áramkör végzi. A tápfeszültség bekapcsolásakor a C105 kondenzátor az R50 ellenálláson keresztül feltöltődik. Ez alatt az idő alatt a C24 kondenzátor is feltöltődik az R34 ellenálláson keresztül. Ez a folyamat addig tart, míg a C105 kondenzátor feszültsége eléri a tápfeszültség  $\frac{1}{3}$ -át. Ez adja az ún. trigger-jelet, mely az időzítést elindítja. Ekkor a C24 kondenzátor a 13-as lábón keresztül kisül. Ez két folyamatot indít el. Az órákimenet (9-es láb) +5 V-ra kapcsol, és az IC figyelő állapotba kerül. Ha az ismételtelen feltöltődő C24 kondenzátor feszültsége eléri a tápfeszültség  $\frac{1}{3}$ -át, az órákimenet ismét alacsony szintre kerül. Így egy pozitív impulzust állítottunk elő. Ezt egy inverteren átengedve, máris a megfelelő RES jelhez jutunk. Az impulzus szélességét a C24 és R34 ellenállás határozza meg.



3. ábra. A C64 NYÁK bal alsó sarka. A bejelölt ponton csatlakozhatunk a C105 és R50 közös pontjához

A működést ismerve a célunknak megfelelő átalakítás igen egyszerű. Mint kiderült, az egész folyamatot a C105 kondenzátoron keletkező trigger-feszültség indítja el. A C105 kondenzátor kisütésével a trigger-jel megismétlődik, amely a folyamat elindulásához és így újabb resethez vezet (lásd a 2. ábrát). A kisütést 330 ohm és 1,5 kohm közötti ellenálláson keresztül végezhetjük el. Az ellenállás a 3. ábra alapján könnyen beépíthető.

Figyelem! A hardver reset használatakor BASIC programunk elvész.

A Magazinban már közölt POKE 2050,1:SYS42291: POKE45,PEEK(34):POKE46,PEEK(35):CLR sor beírásával és végrehajtásával lehet előcsalogni.

NAGY VILMOS

## Nemzetközi Számítástechnika — Internacia Komputado

Az 50 országban olvasott magazin példányonként 30 forintért kapható az SKV Könyvesboltban.  
Bp. II., Keleti Károly u. 10.  
Telefon: 158-018

## Ki ad magyarázatot?

A Mikroszámítógép Magazin 85/3. szám 32. oldali felhívására ragadtam írógépet és Primót. Szerintem a válasz arra a kérdésre, hogy át tudjuk-e írni a Primo 0—127 karakterkészletét? nem!! A magyarázat a következő.

A Primo a képernyőre a karaktereket a 35C7—360F címen levő főkiírás-vezérlő által hajtja végre. Ehhez természetesen több járulékos rutin is tartozik, többek között a tényleges karakterkiíró a 379C címtől, de a főkiíró a fent említett címen van. Most nézzük, hogyan végzi a feladatát ez a rutin.

35C7 PUSH IY  
35C9 PUSH HL  
35CA PUSH DE  
35CB PUSH BC  
35CC SUB 1E

Itt vonja le a beérkező ASCII kódból a 30 értéket, hogy megtudja, vezérlő karakter-e.

35CE PUSH AF  
35CF LD HL, 4047

35D2 JR C, 3610

Ha vezérlő karakter (0—30), elugrik 3610-re. Ezért nem lehet ezeket átírni.

35D4 LD A, HL  
35D5 AND 18  
35D7 CALL NZ, 369F

35DA BIT 6, IX+05

35DE CALL NZ, 3798

35E1 LD IY, 31F7

Itt tölti be a 32—127 kódúak kezdőcímét a ROM-ba. Mivel ez fix cím, megváltoztatni lehetetlen.

35E5 POP AF  
35E6 PUSH AF  
35E7 CP 62  
35E9 JR C, 35F1  
35EB SUB 62  
35ED LD IY, 404B

Ezzel szemben ez már a RAM-ból kérdezi le, hogy hol kezdődik a 128—255 kódúak képe. Ezért csak ezeket a kódokat lehet átírni. A rutin további elemzése főlőseges, ugyanis ezek voltak a problémák.

35F1 ...

Lenne még egy lehetőség arra, hogy átírjuk a karakterkészletet: a 16833—34—35 címek a RAM-ban, ahova a BASIC fordító akkor „néz ki”, ha az A regiszter tartalmát akarjuk kiírni. De a Primo-ROM tervezői jobbnak látták, ha azt az elágazó rutint használják karakterkiírásra, amely „kinéz” erre a címre, mielőtt bármit is csinálna. Pedig csak annyit kellett volna tenni, hogy ahol a tárban CALL 0015 vagy CALL 35C7 van, oda CALL 032A-t kellett volna égetni. Ezzel nemcsak a fenti problémát lehetne megoldani egy saját kiírórutinnal, de sok más lehetőséget is megengedne a Primón.

Ezért, ha kérhetem a gép fejlesztőit, hajtsák végre ezt a változtatást! Megéri!

FEHÉR CSABA

**COMMODORE 16** (C-64, VC-20) számítógépe alkalmasá válik bármilyen elektromos készülék vezérlésére, ha bedugaszol a serial bus csatlakozóba egy SBCS-1 tip. illesztőt. Az illesztő 15 db független — BASIC-ből ki-be kapcsolható — TTL kimenetű áramkört tartalmaz. Kérjen tájékoztatót!

**Ára: doboz és hálózati trafó nélkül 799 Ft**  
Megrendelhető: levélben. Cím: 2030 Érd 1, Pf.1.



## AZ ÁRHÁBORÚ ÚJABB HULLÁMA

### Tramiel mindig győz?

Az első „olcsó” amerikai számítógép bevezetésével — a VC 20 típusú — Tramiel kezdte meg az árháborút. Ezzel a géppel, és azzal, hogy a gép árát úgy tudta sokszorososan csökkenteni, hogy még mindig nyereséges maradt, sikerült az olcsó számítógépek piacáról eltávolítani a Texas céget. (És csaknem teljesen az Atari céget is.) Az igazi nagy sikert az ezt követő C64 jelentette. Egyes becslések szerint ebből több mint hárommillió darabot sikerült eladni és úgy, hogy maga a termék a sokszori árcsökkenés ellenére (az eredeti árak mintegy ötödére) még mindig nyereséges maradt. (Egyesek szerint a Commodore cég problémáit az okozta, hogy ezt a nyereséget számos — üzleti sikert el nem ért — gépre költötte el.) Amikor Tramiel-nak távoznia kellett a Commodore cég éléről, állítólag megfogadta, hogy visszaüt. A visszaütéshez eszközként megvásárolta az — egyébként általa tönkretett — Atari céget.

A visszaütéshez egyrészt a korábbi cégek sikergépének a konkurenseit hozta létre, illetve futtatta fel, másrészt konkurenst igyekezett létrehozni az Apple Macintosh gépnek. Az utóbbival konkuráló gépe elsősorban abban különbözött attól, hogy ára lényegesen alacsonyabb volt (ez vitatottan a termék minőségének rovására ment).

Egy másik „húzás”, ami most kezd „beindulni”, hogy a Commodore cég legsikeresebb területét, Európát vette célba.

Az új árháború területe földrajzilag jelenleg Európa (különösen az NSZK), felhasználási terület szempontjából pedig az irodai rendszerek. Ezekben belül megjelent a „filléres” kategória. Itt indította meg májusban a nagy rohamát az Atari cég.

Az általuk kínált legolcsóbb rendszer a következőkből áll:

- Atari 800 XL (64 K RAM), valamilyen olcsó monitor, Atari 1050 (127 K hajlékonylemez egység), Seikosha GP100AT (50 karakter/sec sebességű grafikus A4 méretű papírral dolgozó mátrixnyomtató), szövegfeldolgozó programrendszer, melynek ára 1140 DM! (Ugyanez, csak a számítógép a korszerűbb és készeres memóriaméretű 130XE típusú 1340 DM.)

A többiek ilyen alacsony árral képtelenek eddig dolgozni.

Mik vannak eddig a piacon? A Sinclair rendszer: egy Spectrum (48 K RAM), egy olcsó monitor, egy hajlékony lemezegegyes-meghajtó (3,5”), lemezegegyes (720 K!), Seikosha GP50S (40 karakter/sec sebességű keskeny papíru nyomtató! 1350 DM ugyanez a 128 K memóriájú 68008 típusú mikroprocesszorú QL-el 1450 DM). Commodore rendszer, mely áll a C64-ből, egy olcsó monitorból, a VC1541-ből (168 K lemezegegyes), Seikosha GP500VC nyomtatóból (mely hasonló tulajdonságú az Atari rendszerrel említett) 1590 DM. A Schneider rendszer, mely áll CPC6128 típusú rendszereségből, és egy Brother M-1009 típusú nyomtatóból 1850 DM. (Angliában már kapható a 8256 elnevezésű rendszerük, aminek az ára márkára átszá-

molva mintegy 1600, és egy 256 K RAM memóriájú számítógépből, egy 180 K méretű lemezegegyesből, egy közel levélminőségű nyomtatóból és egy szokatlan felbontóképességű 90 x 32 karakter-monitorból áll.)

**Összefoglalva:** az árháború az 1000—1500 DM körüli árú (az egészen kis vállalatok számára tehát könnyen elérhető, sőt a háztartások számára is), teljes rendszerek megjelenése a házi számítógép-piac nagy fellendülését fogja valószínűleg eredményezni. (Az 1000—1500 DM hivatalos árfolyammal számolva mintegy 20—30 000 Ft-ot jelent!!)

(A jobb minőségű, de lényegesen más szolgáltatásokat nem nyújtó területen a Commodore a C-128D gépét — egybeépítve egy 128 K RAM memóriaméretű több operációs rendszerű mikroszámítógépet és egy 336 K lemezegegyes —, valamint egy Seikosha SP1000VC típusú, közel levélminőségű nyomtatót 2920 DM összegért árulja, míg az Apple által ajánlott rendszer egy Apple IIc — egy 128 K memóriaméretű beépített lemezegegyesű és egy beépített nagy felbontású monitorral, hasonló típusú nyomtatójú rendszere 2800 DM.)

A másik terület, ahol az Atari támad, a nagy belső memóriájú gépek területe. Itt rendkívül alacsony árakkal igyekeznek a konkurenciával „végezni”. Az árszerzés: egy 520 ST+ mikroszámítógépből (1 M!), egy lemezegegyesből (360 K), egy nagy felbontású (640 x 400) monitorból álló rendszer az operációs rendszerből és a kiterjesztett LOGO, valamint az ugyancsak kiterjesztett BASIC-ből álló alapszoftverrel 2000 DM! (A legolcsóbb IBM-kompatibilis rendszerek ára kb. 2600 DM, ha 640 K memóriaméretű a gép, 2 x 360 K lemezegegyes tartalmaznak, monitort, és színes grafikus kártyát.)

Személyes tapasztalatokat szereztem az új Atari rendszerrel és a Sinclair QL rendszerrel. A tapasztalatok kedvezőtlenek voltak. Ennek valószínűleg nem az az oka, hogy a két rendszer ne lenne jó minőségű, hanem valószínűleg az, hogy „leértékelt” géprendszereket igyekeznek eladni. (A kedvezőtlen tapasztalatokat 2, illetve 3 rendszerrel szereztem. Az első típusból a két rendszer közül az egyik csak bejelentkezni tudott, semmi egyebet, míg a második megfelelően működött, de a gyártó cég rendkívül kevés anyagot ad a gépnek, így annak tulajdonságai ez alapján nem voltak vizsgálhatók. Ennek a fantasztikus rendszernek a hivatalos árfolyamon átszámított ára tehát kevesebb, mint 40 000 Ft!! A második típusnál mindhárom rendszer képtelen volt a lemezegegyesről bármiféle programot hibátlanul betölteni.)

A C16 ára jelenleg 125 DM (ami hivatalos árfolyamon átszámítva, mintegy 2000 Ft). Úgy hirdetik, hogy a ZX81 óta ilyen olcsó gép még nem volt. Az alacsony árak köszönhetően a sikerlistán jogynéhány helyet előrébb lépett. Ez az ár természetesen már kiárusítási ár.

DR. SIMONYI ENDRE

## Pro-Kontra GM ÖNMAGÁRÓL

1982-ben 5 fővel alakult meg a cégünk, nem túlságosan konkrét üzleti elképzelésekkel, amelyekből vajmi kevés vált valóban üzletté, mintegy fél évig.

A dolgok akkor kezdtek komolyabbra fordulni, amikor tevékenységünket az elektronikai fejlesztésre próbáltuk koncentrálni, főleg a személyi számítógépekhez kapcsolódó egyedi vagy kisseriás hardver- és szoftvertermékek kihozatalával. Rövidesen olyan termékeket mondhattunk magunkénak, mint a C64-hez kapcsolható numerikus tasztatúra, a PROFI-COPY másolóprogram a hozzá tartozó — EPROM másolását, illetve a tartalom másolását lehetetlenné teszi! — cartridge-dzsel, továbbá néhány egyszerű kiegészítő eszköz.

A kisseriás gyártásból származó előnyök és hátrányok széles skáláján tapogatózva, néhány csúfos kudarc — és persze anyagi veszteség — meg némi sikerélmény után megtanultunk olyan alapvető szabályokat, amelyeket valószínűleg már mások is tudnak, talán régebben is tudták nálunk, de mi „vérrel megfizetett” árunak tartjuk tudásunkat.

Döntéshozó szisztémánkat egyszerűsítve, de a közös érdek elsőbbségét figyelemmel tartva, létszámunkhoz — jelenleg 14 fő — képest elég széles, alább összefoglalt termék- és szolgáltatásvalasztékot hoztunk létre:

- digitális kijelzésű elektronikus óra (kijelzője saját szabadalmunk!),
- IEC buszról vezérelhető univerzális mérés-pontváltó,
- ipari folyamatirányítás segédberendezéseivel kapcsolódó elektronikai egységek,
- két- vagy többfunkciós gombokkal felépített numerikus klaviatúra a C64 géphez (a Novotrade Rt. forgalmazza, COMFORT néven),
- FAST VC 1541 műveletgyorsító rendszer a C64 konfigurációhoz,
- 64 k-s RAM bővítő a C16 géphez,
- és lesz még néhány, részben a tervezőasztalon, részben a gyártástechnológia kérdéseire még választ váró termék.

Szolgáltatásaink például a hazánkban elterjedtebb személyi számítógépek és kiegészítő egységeik fizető javítása, az elektronikai építészeti, épületgépészeti szaktanácsadás, villamos biztonságtechnikai szolgáltatások végzése (KÖZÉPFESZÜLTÉGŰ hálózatokon is!).

Ami a fejlesztéseink főbb gondjait illeti, ezernyi mására rimelnek: importalkatrész-tökehiány, alvállalkozóink szerződésszegése — amiért még mindig mi tartozunk köszönettel —, és főleg, hogy a nap csak 24 órából áll. A számítástechnika hazai elterjedésében és közkeletűvé változtatásában mindenesetre egy kicsi szeletet szeretnénk magunkénak tudni.

Bár tagja lettünk a Kamara kisvállalkozói tagozatának s még néhány szervezetnek, ezúton is közzé tesszük: szívesen együttműködnénk más kisvállalkozókkal, cégekkel, amelyek — ifj. Marosán György után szólva — a magyar „Silicon-Valley” létrehozását nem abszurdumként, hanem egy valamikori, jövőbeni célként kezelnek.

GERGELY GYÖRGY



## KEDVES OLVASÓINK!

Egyre több olyan tartalmas levelet kapunk, amelyekből sajnálók részleteket kihagyni; így most kevesebb levélből, hosszabb részleteket közlünk.

Somodi Zsolt, Szeged,

Vértói u. 5. 6724

1985. április eleji válaszában azt írta a játékprogramokkal kapcsolatos kérdésekre, hogy kb. júniusban megjelenik egy könyvük, melyben csak játékprogramokat fognak közölni. Nos, én megvártam a júniust, de azt mondták, hogy ilyen könyv nincs még forgalomban, és így van ez még máig is! Szeretném, ha megírná, hogy mikor, hol, mennyiért, és egyáltalán: lehet-e majd kapni.

Most februárban az egyik haverom kapott egy SEIKO DATA—2000-es „zsebszámítógépet”, de nem kapott hozzá semmiféle dokumentumot. Ehhez kérném az Ön segítségét. Biztosan ismeri ezt a típusú gépet, és tudna nekem küldeni róla egy felhasználói leírást. Nagyon kérem, küldjön egyet, vagy szerezzen nekem. Előre is köszönöm!

Múlt évi levelében azt írta nekem, hogy játékprogramokat nem küldenek. Azt szeretném megtudni, hogy úgy sem, ha küldök cserébe programokat? Vannak a sulimban készült, és ismerős körből való programjaim!

1. A könyv a Műszaki Kiadó gondozásában megjelent. Címe: „Csupa játék ZX-Spectrumra”, ára 28,— Ft. Kapható a könyvesboltokban.

2. A SEIKO DATA 2000-sel kapcsolatban nem tudunk segíteni. Csak ismételni tudom: olyan számítógépet érdemes venni vagy kapni, amelyből sok van az országban.

3. Játékprogramot nem küldünk, de nem is cserélünk. Ha van közölhető eredeti programja, azt szívesen fogadjuk, és ha jó, közöljük.

Kiss Imre, Győr,

Kilián u. 6. 9026

Az 1986. évi 1. számot február megjegyzéssel látták el, ugyanakkor a 2. oldalon arról tudósítanak (örömmre), hogy évente 12-szer jelenik meg 1986-tól a Mikroszámítógép Magazin. Ez hogyan lesz lehetséges? Mindenesetre remélem, hogy nem késtem

le a január feliratú számról (bár a februárít januárban vettem), itt inkább tévedésről van szó. De azt hiszem, egy számítógép programja sem hibátlan mindig; a lapjuk 1986. évi programja vajon milyen lesz?

Az év elején nemcsak a havi megjelenésre tértünk át, hanem nyomdát is változtattunk, ezért jelentettük meg az első számot februári jelzéssel. Nem mulasztott el egyetlen számot sem, a tizenkettedik számot is megjelentetjük.

Szieberth András, Székesfehérvár,

Sebes I. u. 17/B 8000

9 évig úsztam, 1 éve pedig öttusázom, így talán érthető, hogy sokat foglalkoztat a gondolat: hogyan lehet a számítástechnikát alkalmazni a sportban? Mikor kaptam egy Spectrumot, írtam egy edzésnapló-programot, mely összesíti a távokat, eredményeket, s lehívhatók az eddig megtörtént események. Mostanáig kitűnően működik. Nemrég ismerőseim megkerestek, hogy írjak egy városi iskolai úszóprogramot, mivel kézzel elég nehézkes számolni. A kezdeti nehézségeken túljutva a program jelesre vizsgázott. A versenyidő majdnem felére csökkent, s a versenyjegyzőkönyv megírása, sokszorosítása pillanatok műve volt.

Kérdésem: tolmácsolhatnám-e lapjukon keresztül azt a felajánlásomat, hogy bármely tömegsportrendezvény eredményszámítását számítógépre viszem, bármely sportegyesület vagy város kérésére, ingyen. Esetleg egy cikk vagy cikksorozat keretében ismertetném eddigi tapasztalataimat, közölném a programokat.

Mivel csak Spectrummal és egy Alpha-com grafikus nyomtatóval rendelkezem, olyanok segítségét kérem, akik a többi általánosan elterjedt gépre át tudnák írni a programokat.

Ajánlatát máris közreadom, a cikket pedig, jó illusztrációkkal, programokkal várjuk.

Vados Tamás, Budapest,

Regöly u. 6. 1182

Rendszeres olvasója vagyok a Mikromagazinnak. Amikor megveszem a lapot, egyből a µprogramokhoz lapozok, s néha — hangsúlyozom, hogy csak néha! — csalódva állapítom meg, hogy a programoknak csak két-három oldal jutott.

Az olvasói rovatot böngészve úgy vettem észre, hogy programokat az olvasók is küldhetnek be. S minthogy ez csak feltevésem, azt kérném Szerkesztő Úrtól, írja meg nekem, hogy ez valóban igaz-e, és hogy játékprogramokat is lehet-e beküldeni. Azt is szeretném tudni, hogy papíron vagy valamilyen mágneses adathordozón (floppy,

kazetta) juttassam-e el a szerkesztőségbe, és hogy ezeket visszakapom-e? Én mindenestre már most elküldöm a programot részletes leírással, hogy ha ebben a formájában megfelelő, akkor fel tudják használni.

Így igaz, a µMagazinban közölt programok nagyobbik részét az olvasók küldik, csak a kisebbik részét „rendeljük”. A programok kipróbálásához szükségünk van vagy a mágnesszalagra vagy a hajlékony mágneslemeze, a közléshez pedig a nyomtatott programlistára.

Auer Gábor, Tiszapüspöki,

Arany János u. 4.

Elnézésüket kérem a zavarásért, de úgy érzem, hogy a VC20 számítógép tulajdonosait egy kicsit elhanyagolják. Szerintem vagyunk elegendően ahhoz, hogy kiérdemljük a törődést. Gondolok itt a programozási ötletektől a gépi kódú leírásokig. Amennyiben szükség lenne rá, a későbbiekben küldenék gépi kódú programozáshoz ismertető anyagot, valamint a VC20-ra programozási ötleteket.

Ízelítőnek egy rövid programötlet, melynek segítségével a képernyő bármely pozíciójára tetszőleges szöveget vagy megtervezett figurát megjeleníthetünk:

```
10 INPUT "X,Y"; S,O
20 POKE 214,S:REM SOR SZÁMA
30 POKE 211,O:REM OSZLOP SZÁMA
40 SYS 58640
50 PRINT„§”
```

A VC20-ra valóban kevés programot közlünk. Lehet, hogy olcsósága miatt sokan vásároltak VC20-at, de azt hiszem, ma már egyre kevesebben teszik, hiszen az ennél nagyobb gépek is elérhető áron kerülnek forgalomba. Azt sem tagadjuk, hogy előnyben részesítjük azokat a jó és érdekes programokat, amelyek a hazai gyártású számítógépeken futtathatók.

Fábián István, Érsekvadkert,

Rákóczi u. 176. 2655

Kissé meglepett, hogy az 1986/1. szám februári keltezéssel jelent meg. Mivel ez évben 12 számot ígérnek, ezért valószínűleg nincs elég hónap! Javasolom, hogy vezessünk be egy új hónapot: „PROGREMBER” néven.

De félre a tréfával. Nem ez az egyetlen ok, amiért írok. Az 1985/6. számban a µprogramok rovatban megjelent UNIN-rutin nem egy hibát tartalmaz, és bár néhányat sikerült megtalálnom, a rutin még mindig működésképtelen. Szerintem nem érdemes közölni olyan programlistát, amelynek működését nem írják le eléggé részletesen, mivel így lehetetlen rájönni a hiba okára. A lista egyébként is gyanús, mivel a C64 nem fogad el 80 karakternél hosz-



szabb sorokat (210, 340, 495, 540-es sor). A lista széle is hiányzik (105, 140, 495, 540-es sor).

A C64-re létezik az OXFORD PASCAL fordító. Nekem megvan, de sajnos minden dokumentáció nélkül. Azt hiszem, nemcsak nekem lenne hasznos, ha erről közölnének egy leírást, mert a BASIC-nél nem szeretnék leragadni. A FORTH-ról is csak elviekben szólnak, de egyetlen gyakorlati leírás sem olvasható róla. Ezzel is próbálok dolgozni, mivel sikerült megszereznem az eddig egyetlen magyar nyelvű könyvben leírt DATATRONIC AB-féle FORTH programot. A screen editorban megszerkesztett és lemezre írt programot visszatöltés után sem értelmezi az interpreter, és a szótárban sem jelenik meg az így definiált új szó. A sort még sokáig lehetne folytatni. Szerintem nem elég bizonygatni, hogy bizonyos nyelvek mennyivel jobbak, mint a BASIC, és utána továbbra is csak BASIC programokat közölni. A C64-re már nem a BASIC az egyetlen implementált nyelv, mégis a szoftverek döntő többsége ezen a nyelven íródik, és aztán lefordítják pl. AUSTRO COMP vagy AUSTRO SPEED BASIC-fordítókkal, vagy PETSPEED-del. Sajnos ettől még a BASIC BASIC marad.

A  $\emptyset$ Magazin sokat tehetne azért, hogy a jó minőségű fordítóprogramok alkalmazásával a mikrogepeken ne csak BASIC nyelven programozzanak azok, akiknek lehetőségük lenne más nyelven is dolgozni, és ebben csak a birtokukban levő programok kezelésének ismerete gátolja meg őket. Jó példa erre a PASCAL Spectrumra és C64-re az LSI ATSZ által kiadott könyv. A C64 részben a szerző elismeri, hogy a PASCAL-64 csapnivaló, mégis ezt ismerteti, és az OXFORD-PASCAL-ról említést sem tesz, pedig a C64 ASSEMBLY LSI-ATSZ 1985. szerzője ezt a fordítók közül igen komolyan minősíti. Az Ipari Informatikai Központ által kiadott C64 Software alkalmazói segédlet meglehetősen borsos áron szintén a PASCAL-64-et ismerteti. A  $\mu$ Magazin közölhetne ilyen információkat is gyakrabban az eddigienél, hasonlóan az Easy-Script ismertetéshez, támaszkodva az NJSZT tagságának ismereteire.

*Gondolom, érdekes levelét valamennyien szívesen olvastuk. A PROGEMBER jó ötlet, gondolkodni fogunk a megvalósításán. Az UNIN-rutin hibáiról szóló javítás azóta már megjelent. Ami a többi programnyelvet illeti: rövidesen FORTH tanfolyamot indítunk; úgy véljük, hogy ez lesz az a magas szintű programozási nyelv, ami a BASIC után széles körben el fog terjedni. Ma a BASIC nyelvű programok közlését az indokolja, hogy ez az egyetlen olyan magas szintű nyelv, amely valamennyi hazai személyi számítógépen futtatható. Úgy tudjuk, hogy olvasóink gyakran átírják a más gépekre közölt programokat a saját gépükre, lapunkat így ötletforrásnak használják. Csak ismételni tudom, hogy a  $\mu$ M elsősorban a hazai gyártású gépeket részesíti előnyben. Különösen az iskola-számítógépekre figyelünk, de nem feledkezünk meg pl. a C64-tulajdonosokról sem, bár nem fogunk csak a Commodore gépekkel foglalkozni.*

# FÓRUM

*A szerkesztőség postájában nem először találunk olyan leveleket, amelyek „országos” érdeklődésre tarthatnak számot. Egyre több olvasónk szól hozzá a Primóval kapcsolatos írásokhoz, ami érthető is, hiszen sem a hazai elektronikai ipar első házi számítógépeért aggódó felhasználó, sem pedig a gyártó számára nem közömbös, hogy mi lesz ennek a gépnek a sorsa.*

*Hasonló érdeklődés mutatkozik az egyelőre legolcsóbb és azonnal szétkapkodott Commodore 16-os gép iránt is, hiszen nem kevesen vannak, akik azt mondják, hogy felesleges volt szaporítani az amúgy is túl nagy számítógép-választékot egy újabb és a nyugati piacon sikertelen géppel, míg mások éppen az ellenkezőjéről szeretnék mindenkit meggyőzni, hogy „végre valamit tett a kereskedelem is az irreálisan magas számítógéparak letörése érdekében”.*

*Kíváncsiak vagyunk olvasóink véleményére, hiszen szinte az egész ország figyelmét kíséri az elektronikai programot, ami ma már társadalmi ügy és amelynek sikeréhez — talán olvasóink véleményével is — hozzá tudunk járulni.*

*A vitát Szabados István és Mohila Károly levelével indítjuk. Az elmondottakról olvasóink véleményét várjuk.*

KOVÁCS GYŐZŐ

## Viták a Primo körül

Már készen volt egy levelem a Primo-furcsaságokkal foglalkozó cikkkel kapcsolatban, mikor kézhez vettem 1986. februári lapszámukat. Így sajnos levelem egy része aktualitását veszítette. Ezzel szemben akadt egy sereg más észrevételem.

A Primo körüli vitában úgy érzem, mindkét félnek (szerkesztőnek, olvasónak) van valami igaza. A szerkesztővel egyetértek abban, hogy a Primóhoz adott felhasználói kézikönyv alig több a semminél. Ráadásul a benne lévőek egy része sem egészen egyértelmű és világos a laikus számára. Véleményem szerint a géphez mindenképpen tartozéként kellett volna adni a külön (és milyen sokára, és milyen drágán) kapható hardver- és szoftverleírást. Ez sok felhasználót megkímélt volna a kísérletezéstől és a bosszúságtól.

Abban viszont Gaál Tamással értek egyet, hogy egy számítógép mellé teljesen laikusként leülve, csupán egy kézikönyvtől nem várhatjuk, hogy bevezessen a számítástechnika rejtelmeibe. A szerkesztőnek sem egészen ez volt a problémája azt hiszem, mindössze a gyártó igen gyatra végköszölgésére akart rávilágítani. Persze lehet, hogy tévedek. Az előbbi mindenesetre úgy érzem, sikerült.

Véleményem szerint egy adott számítógépről sok száz, sőt ezer oldalt is lehetne írni, és így is biztosan sok minden hiányozna még belőle. Úgy érzem, mindenki igényét lehetetlen kielégíteni. Gaál Tamás írja: A számítógép figyelmet is követel a programok írása során. Nos, én ezt egy kicsit kiegészíteném. Nemcsak figyelmet, hanem precizitást is, és nemcsak a programírás, hanem a levélírás is. Meggondolatlanúgnak tartom egy-két kísérlet esetleges sikeréből — különösen anélkül — általános érvényű kijelentést tenni, amikor a jelenség szoftver- és hardverkörnyezetét nem ismerjük kellőképpen. Azt hiszem, mielőtt világgá kiáltjuk valamely „felfedezésünket”, nem árt kissé körüljárni a témát, hogy állításunk más körülmények között is megállja-e a helyét. Lehet, hogy maximalista vagyok?

Jakó János hozzászólásával folytatom. A képernyőterület a Primóban mindig a beépített memóriaterület végcíme mínusz 6143 bájtánál, vagy ennél 8 k-val kisebb címen kezdődik. E két lehetőség között a felhasználó választhat. Bekapcsoláskor a nagyobb cím érvényes, de a  $\emptyset$ -63 perifériaregiszter 3. bitjét  $\emptyset$ -ra változtatva a kisebb cím lesz érvényes. Ahhoz azonban, hogy ez tartós legyen, a regiszter tükörbájtánján (16443 memóriacím) tartalmában is  $\emptyset$ -ra kell változtatni a 3. bitet. Ez a tükör azért van, mert az adott perifériaregiszterbe csak írni lehet (és csak egyszerre a 8 bitet), a kiírt értéket visszolvasni nem. Mivel a regiszter minden bitje más-más funkció megvalósítására szolgál, mindig szükséges tudnunk, hogy milyen értékek vannak azokban, amelyek nem akarunk változtatni. Ezt lehet meg tudni akkor, ha a kiírt értéket a 16443 memóriacímen eltároljuk. Ennek megvalósítása például a következő lehet: POKE16443,PEEK(16443) AND 247: OUT 1,PEEK(16443). Ezért nem helyes,

amit Gaál Tamás írt a RESET hatástalanításáról. A lényeg ui. a perifériaregiszter adott bitjének megváltoztatása. Az általa írt módszer viszont nem mindig eredményezi azt. Például ha az általa adott utasítást programból adjuk ki, és a program további részében nincs olyan utasítás, amelyek végrehajtása közben a processzor a  $\emptyset$ -63 perifériacímre adatot ad ki, akkor a RESET nem hatástalanítódik.

Itt jegyzem meg, hogy a Primóban a RESET gomb semmilyen hatással sincs a processzor NMI jelére. Gaál Tamás ezt nem írta volna, ha ismerné a gép hardverét.

A folyamat ugyanis a következő. A processzor NMI lábára egy kétbemenetű NAND kapun keresztül rá van vezetve a tévéképköltés jele. Ez a processzornak másodpercenként 50-szer NMI-t okoz. Az NMI esetén a végrehajtó program először 1-gyel növeli a 16445—16447 címekeken elhelyezkedő 3 bájtos számlálót (ez a Primo órája). Ezután lekérdezi a  $\emptyset$ -63 bemeneti (csak olvasható) perifériaregiszter 1. bitjét, megtudja, hogy a RESET gombot nyomjuk-e vagy nem. Ha nem nyomjuk, akkor vége az NMI rutinnak, és visszatér a megszakított programba; ellenben ha nyomjuk, akkor melegtartot hajt végre. A NAND kapu másik bemenetére a  $\emptyset$ -63 kimeneti perifériaregiszter 7. bitje megy, ezáltal az NMI-t tiltjuk, ha értéke  $\emptyset$ , és így lesz hatástalan a RESET.

E kitérő után térjünk vissza a képernyőproblémához! A leírtakkal csak azt tudjuk elérni, hogy a kép az adott memóriaterület tartalmát ábrázolja. Ahhoz, hogy írások a karakterek erre a területre íródjának, a képernyőt kezelő programmal is közölni kell, hogy erre a területre írjon. A képernyőn kijelzett terület kezdőcímeinek felső bájtját a Primo a 16458-as címen tárolja, ezért ezt is át kell írni a megfelelő értékre. Az alsó bájt fixen  $\emptyset$ , ezért nincs külön tárolva. Ennyi azonban még nem mindig elég a boldogsághoz. Az ábrából látható, hogy amennyiben kellően nagy karakterterületet foglalunk le a CLEAR nn utasítással, úgy a processzor veremtára az alsó képernyőterületre fog kerülni. Ilyen esetben áttérve az alsó képernyőre, és ott például egy CLS-t végrehajtva megsemmisítjük a veremadatokat, és a gép az első gépi RET(urn) utasításnál „elszáll”, vagyis a  $\emptyset$  címre ugrik, ami a Z80-nál a bekapcsoláskori indulási cím. Ez az a helyzet, amelyet Jakó János úgy fogalmazott, hogy a gép kikapcsolja magát.

Nagy valószínűséggel hasonló helyzet állhat elő a rendszer RAM címeinek találomra történő átírásakor is. Itt azonban nem biztos, hogy ez bekövetkezik. A lehetőségek száma gyakorlatilag végtelen. Előállhat például olyan helyzet, hogy a gép nem reagál semmire, se BRK-ra, se RESET-re. Van azonban valami, amire még ekkor is reagálni fog: a tápegység kikapcsolása. Egyben biztosak lehetünk: a szakértelem nélkül nem „piszkált” gépet a billentyűzetről lehetetlen elrontani. Legfeljebb ki kell kapcsolni, és addigi munkánk esetleg elveszett.



Ami a PEEK-POKE-ot illeti: a 32767-nél nagyobb címeznél a cím kettes komplementjét kell megadni, azaz a cím - 65536-ot. A 16454-es cím tartalma nem a billentyűzet érzékenységét, hanem a sebességét állítja. A billentyűzetet lekérdező rutin ugyanis egy billentyűt egymás után 128-szor kérdez le, s ha csak egyszer is nyomva találja, akkor úgy tér vissza az őt hívó programba, hogy az adott billentyű nyomva van. Két lekérdezés között azonban szünetet tart, s e szünet hosszát szabja meg az említett cím tartalma. A 0-ra pedig azért lesz lassú, mert az 256-nak számít az adott rutinban. Ebből következik, hogy nem 5-6, hanem 256 fokozata van a sebességnek, legfeljebb az emberi érzékelés képtelen ilyen finom felbontásra. Gaál Tamás szerint az említett cím tartalmánál 10-255-nek kell lennie. Ezt úgy értettem, hogy más érték viszont nem lehet, ami nem igaz. Esetleg csak én értettem így?

Az Olvasó írja rovatban (1985/6. szám) Kissné Nyeste Rita levelében hibás egy-két adat. A CTR+X nem a képernyőt törli, hanem csak az utóljára beírt sort, úgy mint a SHIFT+←. A belső kódja pedig 24, de ezt a PRINT CHR\$(24): utasítással nem lehet kiváltani, mert a Primóban a SHIFT+← funkció másként van szervezve. A Q és P nem fél sort, hanem fél karaktert vagy 1/3 sort emel, illetve süllyeszt.

Egy kis kiegészítés. A CTR billentyű a mellé nyomott billentyű Primo szerinti ASCII kódjából 64-et kivon, a ↓ pedig 128-at hozzáad. A két-öt együtt is használható. Ezek segítségével így egész BASIC szavak vihetők be a gépbe, de sajnos a képernyőre nem íródik ki, hanem helyettük 128-151-ig a CHR\$( )-nek megfelelő jel, utána pedig értelmetlen ponthalmaz. Az interpreter azonban „látja” az alapszó tokenjét, és végrehajtja azt. Egyes sűrűn használt alapszavaknál azért még így is praktikus. Programba írva ily módon az utasítást, listázásnál már kiíródik az alapszó. Sajnos akkor is az alapszó íródik ki, ha idézőjelek között írjuk, és szándékunk a megfelelő speciális karakter bevitelére volt. Ez utóbbi csak a listázásra érvényes, a program futásakor a speciális karakter lesz kiírva. EDIT parancs után viszont így nem lehet dolgozni, mert ekkor a ↓ más funkciót jelent. Lehetőség van egyébként, belepiszkálva a rendszer RAM-ba, arra, hogy kiíródjék az egész alapszó. Ennek ismeretése azonban már nem fér bele egy hozzászólásba.

Az ismertető 1. ábrájával és Gaál Tamás írásának erre vonatkozó részével kapcsolatban véleményem a következő. A Primo billentyűzeténél a gyártó a felhasznált IC egy olyan jelenségét használja ki, amely a hőmérsékletváltozás hatására változik. Ez a jelenség ugyanis nem tervezett tulajdonsága az IC-nek, így nincs is specifikálva. Egy 20 °C-on kiválóan működő billentyűzet a nyári 30-35 °C-on vagy esetleg 10 °C-on (ilyenkor ugyan nem kellemes programozni) esetleg „bedobja a törülközőt”. Ezen ugyan segíthet az érzékenység megváltoztatása, de teljes megoldást a HALL-érzékelős billentyűzet jelent.

A 2. képpel kapcsolatban a kézikönyv 40. oldalán a következő olvasható: „... a rendszer a karakterfüzerekkel végzett műveletekhez szükséges átmeneti tárolóterületet is a szövegtárolóból veszi el, így a programban használt karakterfüzerek együttes hosszánál nagyobb tárolóterület kell kijelölni.” Ez látszólag magyarázat lenne a hibára, de az adott példában a hibajelzés után kiírva a még szabad karakterterületet (?FRE(KS)) 332-t kapunk válaszul, aminek elegendőnek kellene lennie még egy párszori ciklus végrehajtásához. Erre egyelőre sajnos én sem tudok magyarázatot adni.

Ellenvetésem van viszont az ismertető cikk

hiányosságokat felsoroló listájának 1. pontjával kapcsolatban. Nem tudom, hogy dr. Simonyi Endre milyen körülmények között tapasztalta az ott írt hibát, mert az én gépem (1984. novemberi) nem fordult elő. Az 5. pontot nem értem, nálam a SQR(0)=0. Az EXP-et illetően a kézikönyv 39-40. oldalán olvasható a Primo szám-ábrázolási tartománya. Mivel ez véges érték, azt hiszem, nyilvánvaló, hogy az EXP argumentuma sem lehet tetszőlegesen nagy.

Markó György FORMA-1 játéka nagyon ötletes. Egy kicsit sok benne viszont a „%” jel, amit az 5. sorba írt DEFINT A-Z: utasítással elkerülhetünk. Ekkor azonban a 15. sor C=0 utasítását írjuk át a 30-as sor elejére.

Nógrádi Előd írására reagálva, nagyon kell fékezni magamat a jelzők használatakor.

1. a) Nem értem, miért van hasznunkra a billentyűzet letiltása akkor, ha le akarjuk állítani a programot? A Primónál ezt egyébként nem lehet megtenni.

1. b) A szerző biztosan nem tudja, hogy a javasolt cím átírása mit jelent. A 16408-as cím ugyanis a Primo NMI rutinjának „ablaka”, ahol egy JP 0057H utasítás áll. Ő ezt, gondolom hasra ütésre, átírja LD BC,0057H-ra. Véletlen, hogy ez után egy másik, de nem használt ablakban szintén JP0057H utasítás áll, így a gép visszatér oda, ahová egyébként is menne. A baj csak az, hogy megváltozott BC értékkel. Ez kiszámíthatatlan hatással jár, mivel az NMI rutin nem használja a BC regiszterpárt, ezért nem menti el annak értékét. Azt viszont nem lehet tudni, hogy a processzor éppen hol tart, amikor NMI-t kap. Lehet, hogy olyan helyen, ahol nem okoz hibát egy más BC értékkel történő visszatérés. Azonban mivel NMI másodpercenként 50-szer van, biztos, hogy hamarosan a program felborulásához vezet ez a dilettáns változtatás. Ki lehet próbálni.

2. A javasolt programot beírtam, azzal az eltéréssel, hogy a 40-es sor NEXT-jét kettőspont után a 30-as sorba írtam. Az eredmény hibás működés volt. A BASIC szabvány szerint pedig egyforma a két program. Nem tudom, Nógrádi Elődnek mi a valódi befehéretés és mi nem, azonban kívül mit jelent a „lassú” és a „viszonylag gyors” szó? A programot az eredeti formájába beírva ugyanis, az 10 mp alatt ugyanazt végezte el, mint amit a PRINT CHR\$(3) átalam már nem mérhetően rövid (kb. 0,1 mp) idő alatt végzett. A -I-t már ki sem próbáltam.

3. Miért nem lehet az említett címek mindegyikét átírni? Az általa említetteket ki írja át gyakran, és vajon mire? Egy kis magyarázat, hogy ne csak kérdezzek. Az említett címek a Primóban végrehajtandó programmal nem rendelkező alapszavak ugróablaja. Jelenleg itt a hibajelző rutinba vezérlő ugróutasítás van. Tetszésünk szerint ezek bármelyikét átírhatjuk arra a címre, amelytől kezdve mi elhelyezzük az adott alapszóra általunk gépi kódban megírt végrehajtó programot. Ezek aktivizálásánál azonban be kell tartani egy szabályt. Azokat az alapszavakat, amelyek tokenje a 128-187 tartományba esik, használhatjuk BASIC utasítás kezdésére, míg a 187 feletti csak más BASIC alapszó után lehetnek. Például THEN, LEN, RND stb.-vel nem kezdődhet utasítás.

4. Első mondata hibátlan. Annak megvalósítását azonban csak azok kezdjék el, akik egy picit konyitanak a mikroprocesszorok világához. A LIST-et kipróbáltam. Elsőre nem találtam hibásnak, de ez még jelent semmit.

Az előzmények után remélem megbocsátják nekem azt a „trehányságot”, hogy nem volt kedvem megnézni a ROM 06CCH, illetve a 2. pontnál a 3EBCH címen kezdődő gépi kódú programot.

SZABADOS ISTVÁN

## A PRIMO

igazán nagyszerű kis gép lehetne, ha a benne rejlő lehetőségeket ki tudná használni az egyszeri tulajdonos. Úgy tűnik azonban, hogy ez (ki tudja miért?) nem érdeke a fejlesztőknek. Úgy tudom, hogy a mai napig várat magára például az szoftverfüzet megjelenése, pedig éppen egy éve ígértetik már. Szintén sajnálatos, hogy igen kevés program jelenik meg nyomtatásban, és azok is eléggé egyszerűek. A Demo kazettán nem ezt ígérték! Hiányzik egy jól tanulható, Primóhoz írt, gépi programozással foglalkozó forrásmunka is, a BASIC bizonyos feladatokra túlságosan lassú. Maradna a FORTH, csak ne lenne olyan drága a beszerzése!

Szegény ember vízzel főz, kénytelen voltam megpróbálkozni más gépre írt programok átírásával. Annak ellenére, hogy ez sem könnyű, legalább nagyon tanulságos. Programok tényleg bőven találhatóak egyes könyvekben, átírásuk azonban megfelelő kitalálást igényel, és közben más gépek lelkivilágának ismeretére is nagy szükség van. Kérdés, hogy elvárható-e a tulajdonosok többségétől az ilyen fokú elmélyedés más gépek használatában. Ráadásul a Primo esetében a komoly munka eredménye az eredeti programhoz képest lehangoló — nem színes, és a bonyolultabb megoldások miatt lassabb is. A gépi kódú programok átírása viszont teljességgel lehetetlen a tárterület felosztásának ismerete nélkül.

Igaz, marad még egy lehetőség az előrelépésre, ha levelezéssel jut az ember új ötletekhez, és kazettán programokat cserél. Sajnos, a vidékiek számára sokkal nehezebb közelebbi klubot találni, ahol ezt elérhetnék. A kazettaküldés pedig rizikó, nekem már kettő is eltűnt a postán. Ezzel kapcsolatban lenne egy javaslatom: a mikroklubok listáján szerepeljenek a géptípusok is, hogy könnyebb legyen partnert találni.

Az itteni Primóval van még egyéb probléma is. Mivel az első sorozatból származik, sok gondot okoz a billentyűzet érzékenysége (jobban mondva érzéketlensége) és az, hogy egy idő után a festék szemérmetlenül lepattogzik róla. Úgy tudom, hogy az újabb gépeken azóta több módosítást is bevezettek, van már profi billentyűzet, illetve az érintős gépek érzékenysége állítható, és botkormány-csatlakozást is beépítettek. Úgy tudom, hogy elkészült a színes kártya, és készül az emberi hang is. Persze kérdés, hogy ezek a plusz funkciók mikor lesznek elérhetőek, hol, mennyiért szerelik át, illetve be. Az is érdekelne (gondolom ezzel sem vagyok egyedül), lesz-e korszerűbb BASIC-je a gépnek, ami igazán ráférne.

Úgy tudom, hogy például az Eprom túlságosan is hasonlít a HT gépére, olyan HT rutinok is vannak benne, amelyeket a Primo nem is használ. Ha új Eprom készül, akkor mit tegyen az ember a „rég” Primójával, átépítik, kicserélik vagy dobjuk talán el? Biztosan érdekelné a Primós sorstársakat, ha a magazin közölné a fejlesztők (gyártó?) válaszait ezekre a kérdésekre, mondjuk „Mit kezdjek a Primómmal” címszó alatt.

### Primo tárfelosztás 64 k-s gép esetén

Vektorváltók				
0	4000	43EA	C800	FFFF
16 K ROM		BASIC program	képernyő alsó	képernyő normál
Rendszer	RAM	Skálárváltók	Z80 processzor veremtár	karakterterület



## A C16 — Tarpán

Második témám a C16-os. Itt Tarpán és a szomszédos Gulácson ugyanis a múlt év vége óta 3 gép is található az általános iskolában. Igaz csak „található”, de nem üzemel. A gépvételt egyetlen indok motiválta: az, hogy olcsó volt. Teljesen tájékozatlanul vették a gépeket, nem olvasták a tájékoztatókat, már csak azért sem, mert a szaklapok létezéséről sem volt fogalmuk. Így állt elő a jelenlegi helyzet: 3 gép + magnó van — színes tévé, tárbővítő, hozzáértő pedagógus nincs. Még elkésőbb az, hogy mindezt ez évben még kilátás sincs, mivel kimerült a pénzügyi keret, és amúgy sem könnyű olcsó színes tévét beszerezni! A gépvásárlásról értesülve ugyan rögtön felajánlottam a segítségemet, a Primóra épülő tudásomat is igyekeztem azonnal továbbfejleszteni a Felhasználói kézikönyv és a Hetedhét Commodore 16 segítségével. Igazán nem rajtam múlt, hogy a szerzett tudással magam sem vagyok elégedett, mivel a forrásokból amúgy se sok minden derül ki, különösen akkor, ha az ember meg van fosztva a géppel való kísérletezés lehetőségétől is. Felajánlottam a meglevő és tanul-

mányozásra érdemes magyar nyelvű szakkönyveimet és folyóirataimat addig is, míg az iskola a kapható irodalmat beszerzi (ha beszerzi). Mit mondjak, az egyik iskola egyáltalán nem lelkesedett a segítségért, sőt lehet, hogy egyenesen gyanúsak találtak. Azóta is várják, hogy a fekete-fehér tévéen megszólaljon a gép (!). Lehet, hogy színekre is számítanak?

Mivel a gépek megvannak, most már ezekkel kellene valahogyan boldogulni. Mivel ebben már van némi gyakorlatom, kézbe vettem az ügyet az iskola helyett. Úgy gondolom, ebben önök is segíthetnek, ha másképpen nem is, legalább a problémák közzétételében, majd csak akad valaki, aki tud esetleg segíteni. Már utaltam a Felhasználói kézikönyv hiányosságaira. Így akad jó néhány dolog, amire eddig nem találtam megoldást: hogyan oldható meg például: 1. az előtörlés kikapcsolása; 2. az ütközésvizsgálat; 3. a karaktergenerálás; 4. a képernyőtartalom kimentése; 5. az automata programindítás; 6. a MERGE funkció; 7. a BREAK letiltása; 8. a botkormány használata (bekötési rajz is jól jönne).

Persze elképzelhető, hogy a válaszoknak legalább egy része megtalálható a „Commodore 16 Programozói Hivatkozási Kézikönyv”-ben, melyet a Felhasználói kézikönyv többször is említ, de azt sajnos elhallgatja, hogy honnan szerezhető be! Mivel tudomásom szerint 3 ezer C16-ossal van több december óta, vagyunk elegen, akiket mindez nagyon érdekelne.

Talán furcsállhatják, hogy miért éppen ezekre a problémákra várok választ, de a Primón a botkormányproblémán kívül már ismerem ezeket a megoldásokat, és mi tagadás, annyira hozzászoktam, hogy szeretném a C16-oson is használni.

\* \* \*

Végére hagytam talán a legfontosabb közlendőt. Valamikor régen elindult egy országos akció, melynek az iskolák televíziós készülékekkel való ellátása volt a célja. Javasolom, hogy ezúttal színes készülékekről, esetleg +mikrogépekről legyen szó. Természetesen minél több üzem, intézmény, vállalat bekapcsolódását kell célul kitűzni. Eddig is akadt már példa arra, hogy közületek iskoláknak számítógépeket vettek. Sajnos, éppen a mi környékünkön nem teng túl az ilyen gesztus, kevés az ipari vállalat, a helyi község éppen veszteséges. Talán nagyobb sikerre lehet számítani, ha szervezett lesz az akció.

Bizom benne, hogy elgondolkodnak javaslatom, és esetleg a Magazin elvállalja, hogy felhívásként közreadja az ügyet. Más kérdés, hogy lesz-e, aki az akció szervezését magára vállalja. Egyébként javaslatommal levélben megkeresem a Videotont is, ugyanis ők mind a színes tévét, mint a mikrogépet gyártják. Reklámra is költenek, gondolom nem keveset, talán megérné nekik ajándékozni is.

MOHILA KÁROLY  
gyógyszerész

## Mibe kerül a 16 bit?

Legutóbbi számaikban sokat foglalkoztak 16 bites számítógépekkel, főleg persze az IBM PC-vel. Ha a fejlődés Magyarországon is arra fog haladni, amerre a nyugati szomszédoknál, akkor pár éven belül ezek és az ezekkel kompatibilis gépek lesznek túlnyomó többségben. Az írottakból úgy tűnik, hogy magánszemély, illetve kisvállalat soha nem juthat ilyen teljesítményű géphez, hacsak nem egy lottónyeremény árán. Így sokan lemondanak a 16 bitről egy jó esetben 128 kb-ajtos és esetleg CP/M operációs rendszerű 8 bites PC javára.

Valóban drágább a 16 bit? A válasz: attól függ, kinek. Mindazoknak, akik valamilyen betűkombinációra vágnak a pléhdobozon (bárki összerakhatja a kedvenc 3-6 betűjét), valóban nem olcsó. Aki viszont a PC-t magánimport útján (persze ha van valutája) hozza, az a normál ár tört részéből beszerezheti. Sok keresgélés sem kell hozzá, a legtöbb elektronikai üzletben hozzá lehet jutni a megfelelő építőelemekhez, amelyek forrasztás nélkül, egyszerű csavarhúzóval összeszerelhetők.

Miből is áll egy 16 bites asztali számítógép?

1. Az alap NYÁK az Intel 8088 proceszorral és 256—640 kb-aj RAM-mal, továbbá rengeteg IC és 7-8 bővítőhely
2. Egy tápegység, 100—150 watt
3. Egy floppy-controller kártya
4. Floppymeghajtó(k)
5. Monitor-controller kártya
6. RS232c és Centronics interfész kártya

Az egész bele van csavarozva egy fémdobozba, aminek az elején a billentyűzet lóg, a tetején meg a monitor áll.

Nézzük az árakat:

1. 390 és 420 nyugatnémet márka között RAM nélkül vagy 500 márka ugyancsak RAM nélkül, de 4,7 és 7 MHz között átkapcsolhatóan.
2. 230—300 márka, de nem hiszem, hogy tápegységet nyugatról kell hozni.
3. 120—140 márka, amire 2 meghajtó kapcsolható.
4. 330—400 márka (japán, legtöbbször Mitsubishi, 2×40 sáv, 360 k).
5. 190—200 márka a „color graphic” RGB + BAS (video) + fényceruza-kimenet, 330—400 márka a „monochrom graphic” TTL kimenet.
6. Egyenként 150, illetve 100 márka. Kombi kártyaként (2×RS232, 1×Centronics, óra, 256 k RAM-nak hely) 300—350 márka az árak.

Billentyűzetet 230—250 márka közötti áron lehet szerezni. 64 k RAM (4164, 150—200 ns) 25 márka, helyenként kevesebb.

Ha a dobozt, amelynek ára 130 márka, és a tápegységet nem számítjuk, akkor egy PC 512 kb-aj RAM-mal, „color” és „kombi” kártyával, 1 floppyval, tasztatúrával 1760—1960 márkából megvan. Nemcsak azért jobb a „color” kártya, mert a repülőgép-szimulátor is fut rajta, hanem lehetővé teszi olcsó, igaz egyszínű monitorok csatlakoztatását is. Ez az ár megfelel kb. egy

Commodore 128-nak vagy egy Amstrad/Schneider CPC 6128-nak, de lehetne sorolni ebben a 8 bites kategóriában még tovább is. A különbség mindössze a 16 bitben és a többlet 384 k RAM-ban van.

A szoftver természetesen MS-DOS, de ezúttal a PC-DOS-ról ismert BASIC interpreter nélkül. Akinek a BASIC nagyon kell, kénytelen külön megvenni. Sokkal ajánlatosabb azonban egy Turbo-Pascal fordítót venni. Ezzel két legyet ütethünk egy csapásra. Először is lesz egy jó szövegszerkesztő programunk (Wordstar, igaz nem minden funkcióval), másodszor is egy Pascal-fordítóprogram, ami a szabvány Pascalon kívül számtalan speciális PC-funkciót kínál. A Turbo-Pascal 3.0 kb. 300 márkába kerül, de mint „használt” szoftver (működnek már szoftver-„bizományok” is!) 100—150 márkáért is kapható.

Természetesen a 14% többletértékadó mindebből visszajár. Ami még jár, de nem vissza, az a vám, és ez ügyben ajánlatos érdeklődni a 64 k fölötti PC-k importjával kapcsolatban. Ha ugyanis a belföldi forgalmi ár alapjául a magyar 16 bites PC-k árát veszik, akkor kár volt ezeket a sorokat elolvasni.

ATTILA LADÁNYI  
NSZK

Szerk. megj.: A cikkben említett megegyező konfiguráció hazai ára: 442 000 Ft (lásd a Novotrade hirdetését a HVG 1986. január 11-i számában).



# Összeköttetést keres?

Számítógépek összekötésére alkalmas  
különbéféle kábeleket vásárolhat üzletünkben:

- Párhuzamos printerkábel (PC-k és Centronix felületű nyomtatók összekötésére)
- Null MODEM kábel számítógépek egymással való összeköttetésére
- Terminálkábel PC-k és terminálok összeköttetésére
- Különböző monitorkábelek (típusoknak megfelelően akár rendelésre is)

Címünk: 2c  
Számítástechnikai Szaküzlet



1136 Budapest, Balzac u. 35.  
Telefon: 402-954



**ORTEKON TERVEZÉSI ÉS FEJLESZTÉSI gazdasági munkaközösség**

1036 Bp., Árpád fejedelem útja 69.

Telefon: 154-250, 887-861

Bányai

**Vállaljuk:**

\_\_\_\_\_

\_\_\_\_\_

— műszerek, villamos és elektronikus ipari termékek konstrukciós és technológiai tervezését, komplex felszerszámozását a sikeres „0” sorozatig;

— elektronikus (digitális) vezérlőegységek, szabályozókészülékek és -rendszerek tervezését és gyártását;

\_\_\_\_\_

\_\_\_\_\_

— CNC-vezérlés komplett felújítását;

— a gyártás-előkészítés és elszámolás irányítási és adatfeldolgozási feladatainak megvalósítását több munkahelyes számítógéprendszer alkalmazásával.

\_\_\_\_\_

\_\_\_\_\_

**A megfelelő referenciákról személyesen adunk tájékoztatást.**



*ajánlja*  
**IBM PC/XT és azzal kompatibilis  
gépeken alkalmazható  
szoftvereit**

*Újdonság!*

## INFOMIX

Szabad, azaz kódolatlan — akár szöveges — információk tárolására és rendkívül rugalmas visszakeresésére alkalmas szoftver. Párbeszédés formában — számítástechnikai ismeret nélkül — építheti és kérdezheti le információbázisát (pl. jogszabályok, könyvek, jelentések, előterjesztések stb. nyilvántartását).

## dACCESS III.

*a SOFTWARE '86 első díjas terméke!*

Szabványos dBASE III. fájlokat és indexeket kezelő host-nyelvi relációs adatbázis-kezelő rendszer. Fájlmánipulációs szolgáltatásai megegyeznek a dBASE III. megfelelő szolgáltatásaival. (Host-nyelv: BASIC, C, FORTRAN stb.)

**Várjuk érdeklődésüket!**

1024 Budapest II., Petrezselyem u. 6.  
Telefon: 351-950



**SYSTEM  
SZERVEZÉSI  
VÁLLALAT**

Budapest, Szt. István krt. 11. 1055

**Megvételre felajánljuk  
ICL 1903/A típusú,  
„O”-ra leírt számítógépünket,  
tartalékalkatrészekkel  
és az alábbi perifériákkal:**

- lyukszalagállomás
- kártyaolvasó
- EDS transzportok  
(7,25 Mbájt)
- EDS kontrollerek
- mágnesszalagegység  
4 deck
- mágnesszalagegység  
2 deck
- konzol asztal + fotel
- lyukszalagtekerő (Facit)
- konzol írógépek
- papírszalaggyűjtő
- motorgenerátor
- motorgenerátor-vezérlő
- GEORGE 1—2 operációs-  
rendszer-program
- mágneslemezek (7,25 és 60  
Mbájt)
- beállító mágneslemez

## MEGEGYEZÉSES ÁRON.

Érdeklődni lehet a 401-539 telefonszámon Eóry Örs főosztályvezetőnél.  
Telex: 22-6975  
Levél cím: 1363 Bp. 502. Pf. 33.



**A**z elmúlt alkalommal elkezdtek a bonyolult, többváltozós értékelő-függvény egyes komponenseinek tárgyalását, beszéltünk a sakkjátékban a figurák anyagi jelentőségéről és mozgékony-ságáról. Most egy-két statisztikai adatba szeretném beavatni az olvasót, amit eredetileg egy amerikai folyóiratban publikált Eliot Slater, és David Levy is közölt a The Chess Computer Handbook c. könyvében.

## A mozgékony-ság jelentősége

Slater 1949-ben híres versenyek és páros mérkőzések játszmáiból 78-at választott ki véletlenszerűen, amelyeknek közös tulajdonsága, hogy mindegyik a 40. lépés előtt, küzdelemben fejeződött be.

Nézzük az 1. táblázatot, amelyet Slater úgy állított össze, hogy kiszámította az előbb említett 78 kiválasztott játszma mozgékony-sági értékeinek átlagát lépésenként, mind a győztes, mind a vesztes fél esetében. A rövidítések jelentése: GY — győztes; V — vesztes; GY-V — különbség; (GY-V):(GY+V) — relatív gyakoriság.

LÉPÉS	GY	V	(GY-V)	(GY-V):(GY+V)
0	20,0	20,0	0,0	0,000
5	34,2	33,9	0,3	0,004
10	37,5	36,0	1,5	0,020
15	39,7	35,2	4,5	0,060
20	38,9	36,4	2,5	0,033
25	39,6	31,9	7,7	0,108
30	35,6	27,7	7,9	0,125
35	31,7	23,2	8,5	0,155

1. táblázat

Világosan látszik, hogy a két fél bábjainak mozgékony-sági értéke a figuráknak a játszma elején való kifejlődése során megnő, és később a figurák lecserélődése során visszacsökken. Ha még tüzetesebben megvizsgáljuk, akkor látjuk, hogy a játszma vége felé a mozgékony-ság értékének visszaesése kisebb a vártnál. Ennek oka, hogy ha több figura van a táblán, akkor a sakktábla zsúfoltságából következik, hogy egy adott bábunak kevesebb legális lépése van. A figurák lecserélődése után pedig igaz, hogy kevesebb a figura, viszont nagyobb a mozgáslehetőségük. Így például egy vezér a játszma elején nem tudja kihasználni, hogy nyolc irányba léphet, míg a végjátékban sokszor előfordul olyan eset is, hogy a nyolc irányba 27 legális lépést tehet.

# Bitek és figurák

## Állásértékelés II.

Az 1. táblázat 4. oszlopa a győztes és a vesztes mozgékony-sági értékének különbségét tartalmazza, az 5. pedig a relatív gyakorisági értéket. A mozgékony-ság értékének alkalmazhatóságát már a megnyitásban is láthatjuk. Tekintsük a sakkjátszma kezdeti állását, és nézzük sorra, hogy az egyes megnyitó lépések milyen mozgékony-sági értéket képviselnek. (2. táblázat)

Mindjárt szembetűnik, hogy a sakkozók által játszott legnépszerűbb megnyitások mozgékony-sági értéke kimagaslik a többi közül. Ezek az e2-e4 és a d2-d4 lépések. Az is látható, hogy ugyanannyi a mozgékony-sági értéke az e2-e3 lépésnek, illetve a d2-d3 húzásé is megközelíti, de ez a sakkjátékban más szempontok figyelembevételével mégsem tartozik az erős lépésekhez. Ezt a sakkprogram is figyelembe veszi az értékelőfüggvény később ismertetett megfelelő komponensében.

8	1	1	1	1	1	1	1	1
7	1	2	2	2	2	2	2	1
6	1	2	4	4	4	4	2	1
5	1	2	4	8	8	4	2	1
4	1	2	4	8	8	4	2	1
3	1	2	4	4	4	4	2	1
2	1	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1
	a	b	c	d	e	f	g	h

1. ÁBRA

Az előző cikkben közölt Fischer-Szpaszkij játszma záróállásában arra láthatunk példát, hogy a mozgékony-sági előny ellensúlyozza az anyagi hátrányt. Az anyagnak a mozgékony-sággal való kompenzálását a különböző cseljátékok elemzésénél figyelhetjük meg jobban. Például a középcsiben a következő lépéssorozat után létrejövő hadállásban tisztán látható, hogy a világos két gyalog hátrányát figuráinak mozgékony-sága és aktivitása kompenzálja.

1. e2-e4 e7-e5
2. d2-d4 e5×d4
3. c2-c3 d4×c3
4. Ff1-c4 c3×b2
5. Fc1×b2...

A kialakult állásban szembetűnik, hogy világos mindkét futója aktív, a leghosszabb átlókon helyezkednek el, és a vezér is három irányban szabadon mozoghat, míg a sötét futók bezártak, és a vezér is az alap helyzetéből egy irányba léphet.

Más példák is adódnak cseljátékokban, amelyek a mozgékony-ság fontosságát mutatják.

A mozgékony-ság szerepe a végjátékban is nagyon fontos. Ezzel részben elérhetjük azt is, hogy figuráink szabadabbak legyenek, és harmonikusabban építsék fel az állást, vagyis kiegészítsék egymás hatósugarát, és ne hátráltassák egymást a szabad mozgásban. A mozgékony-ság értékének figyelembevételével a program automatikusan kerülni fogja a figuráknak a tábla szélén történő elhelyezését. Így ha például egy futó vagy egy vezér az a1-h8 átlón áll, akkor nem lép pontosan az a1, illetve a h8 mezőre, hanem egy olyanra, ahol egy második átlón is kifejtetheti hatóerejét, hacsak a más mezőre történő lépésre az értékelő-függvény többi komponense kedvezőle-  
nebb értéket nem ad. A mozgékony-ság túl-

8	1	2	3	4	4	3	2	1
7	2	5	8	7	7	6	5	2
6	3	6	8	9	9	8	6	3
5	4	7	9	10	10	9	7	4
4	4	7	9	10	10	9	7	4
3	3	6	8	9	9	8	6	3
2	2	5	6	7	7	6	5	2
1	1	2	3	4	4	3	2	1
	a	b	c	d	e	f	g	h

2. ÁBRA

2. táblázat

LÉPÉS	MOZGÉKONYSÁG
E2-E4, E2-E3	30
D2-D4	28
D2-D3	27
C2-C4, HB1-C3, HG1-F3	22
A2-A4, B2-B4, B2-B3, C2-C3	21
G2-G3, G2-G4, H2-H4	21
F2-F4, HB1-A3, HG1-F3	20
A2-A3, F2-F3, H2-H3	19



súlyozása a szükségesnél nagyobb anyagi áldozatokra készíti a programot, ha viszont a súlyozó tényező túl kicsi, a program nem fogja pozícióját anyagi áldozat árán javítani.

## Centrumellenőrzés

Mint az előbb láttuk, az értékelőfüggvény a mozgékonyági komponensében automatikusan figyelembe vette, hogy a figurák ne kerüljenek a tábla szélére. Mivel a mozgékonyág túlsúlyozása is fennállhat, ezért egy újabb komponenst is célszerű figyelembe venni, a centrumellenőrzést, amely az előbbihez hasonló hatást gyakorol a figurákra, de pontosítja az értékelőfüggvényt, és segítségével a mozgékonyág—anyag egyensúlya is pontosabban számítható.

A sakktábla központi mezői (e4, e5, d4, d5) értékesebbek a szélsőknél, mivel innen több helyre léphet egy figura, és ezeknek a mezőknek a megtámadása elfoglalásuk lehetőségét is magával hozza. Ezért fontos annak a mértéknek a megállapítása, hogy a játékos mennyire jól vagy rosszul ellenőrzi a tábla központi részét. Ennek legkézenfekvőbb módszere, hogy a sakktábla minden mezőjéhez egy-egy súlyozó tényezőt rendelünk, amely a tábla szélén kisebb, és a centrum felé haladva növekszik. Így a program minden mező megtámadását azonosítani tudja, és a támadó játékosnak az azonosított mező pontértékét adja. Ilyen mezősúlyozást mutat az 1. ábra.

Ez a súlyozás nem veszi figyelembe, hogy a szélső mezők sem egyenértékűek, azaz a sarki négyzet megtámadása kevésbé hasznos, mint a szélső vagy középső mezőké. Ennek figyelembevételével a mezősúlyozásra jobban differenciált táblát is készíthetünk, amely folytonosabb átmenetet biztosít a központi mezőktől a sarkokig. (2. ábra)

KOVÁCS P. ATTILA

### DIGITÁL

Számítástechnikai  
Szaküzlet újdonságai  
Spectrum számítógépekhez

KÜLSŐ PROGRAMTÁR ÁRA: 1500,— Ft

UNIVERZÁLIS PRINTER INTERFACE  
ÁRA: 4500,— Ft

FÉNYCERUZA ÁRA: 1900,— Ft

Budapest II.,  
Szilágyi Erzsébet fasor 35. 1026  
Telefon: 156-231

# SZÁMÍTÓKÖZPONT-VEZETŐK!

**Ne dobják ki sérült mágneslemezeiket!**

**Gyorsan, olcsón,  
6 havi garanciával megjavítjuk!**

Commodore 64 számítógépükhöz  
— az adatrögzítést megkönnyítő —  
kiegészítő numerikus klaviatúrát  
raktárról szállítunk, 1 éves garanciával.

Megvásárolható  
az Econorg 1. sz. Számítástechnikai szaküzletében:  
Bp. VI., Szinyei Merse Pál u. 1. Tel.: 127-628

Bp. Erzsébet u. 14. X. 59. 1045  
Telefonügyelet: Grósz Andor, 632-720 (9—17 óráig).

**PRODUKTORG**  
SZERVEZÉSI VÁLLALAT

**C64, M08X, IBM PC XT**  
kompatibilis rendszerekre  
ajánljuk programcsomagjainkat:

- *főkönyvi könyvelés*
- *költségelosztás*
- *állóeszköz-nyilvántartás*
- *rendelés-nyilvántartás*
- *béreلسzámolás*
- *adóelszámolás*

**Bővebb felvilágosítással  
készséggel állunk  
rendelkezésükre!**



## Válasz egy tankönyvbírálatra

Pogány Csaba megtámadta a tankönyvemet (*µMagazin*, 1986. március, 45. oldal). Ez rendben is van, joga van hozzá. Ha a *µMagazin* nem olvasná annyi diák, valószínűleg megvonnám a vállamat, nem válaszolnék. Így azonban azt kérem, aki esetleg hajlamos Pogány Csaba szavainak hitelt adni, olvasson bele a tankönyvembe alaposabban, esetleg olvasson bele Pogány ugyanezen számban található cikkébe, és ennek alapján döntse el, melyik a jobb. Pogány Csaba cikke mélynek, tudományosnak tűnhet, de üres fontoskodás, és én nem hiszem, hogy a *µMagazin* olvasói ezt várják az újságtól. Nem látom tehát, hogy veszi a merszet Pogány ahhoz, hogy megtámadjon, de megpróbálok kulturáltan válaszolni tévedéseire. (Nem érdemli meg.)

Mielőtt a könyv írásába kezdem volna, barátaim figyelmeztettek, hogy bármit írok is, éles támadásoknak teszem ki magam. Így azután nem is annyira Pogány Csaba támadása lepett meg, hanem az, hogy ilyen kevés és ennyire hozzá nem értő a támadás.

Természetesen a könyvben sok sajtóhiba van. Van néhány elírás, figyelmetlenség, és néha, bosszantó módon meg a programokba is bekerültek félrenyomatások (aminek az az oka, hogy a nyomtató másként tördelt, mint a nyomda). A valódi hibák túlnyomórészt eljavítható a következő kiadásra, és Pogány Csabának megköszöntem volna, ha segít ebben.

Pogány vádjai azonban többnyire alaptalanok, kinyilatkoztatás-szerűek, de az érvek hiányoznak mellőlük.

A lényeges vádpontok: „Nincs a könyvben munkára nevelés, tiszta didaktikai szerkezet, rendszeres megalapozott fogalomkialakítás, gyakorlati szemlélet, szerzői vagy lektori lelkiismeretesség.”

Először ezekre felelek.

Legtöbb technikai eszközünk, a fényképezőgép, az autó, a számítógép kompromisszumokkal vannak terhelve. Ha valaki egy tankönyvet ír, szintén számtalan kompromisszumot kell kötnie vágyaival, elveivel, lehetőségeivel, már csak a terjedelmi korlátok miatt is. Mielőtt egy ismeretterjesztő cikk vagy egy tankönyv írásába kezd valaki, el kell döntenie, kinek írja a könyvet. Én azokra gondoltam, akik a tankönyvből nem a számítástechnikai részleteket akarják megtanulni, hanem a számítástechnika alkalmazhatóságát. Már ez a réteg is legalább két csoportra bomlik: egy részre nem érdeklődik a számítástechnika mélységei iránt, a másik része még sokkal mélyebben érdeklődik, mint hogy ismereteit középiskolai tankönyvből szerezné meg.

Ennek megfelelően a bevezető rész után azt a koncepciót választottam meg, hogy elsődlegesen léteznek a megoldandó problémák, mi ezekkel „szembetalálkozunk”,

és megoldjuk őket, miközben lassanként elsajátíthatjuk a BASIC nyelv azon részét, amelyre éppen szükségünk van. Úgy is fogalmazhatnánk, hogy nem a szerszámhoz keresem a munkát, hanem a munkához a szerszámot. Ezzel persze vállalom azt, hogy programozás szempontjából néhol nehezebb részek után könnyebbek következnek. Mint ahogyan a klasszikus fizikaoktatásban is ez történt.

Pogány Csaba erre azt írja, hogy nincs a könyvben tiszta didaktikai szerkezet. Elfelejtji, hogy a didaktika arra való, hogy nehéz dolgokat aránylag könnyen megtaníthassunk. Nem az a kérdés tehát, hogy ő felismeri-e a didaktikai szerkezetet, hanem az, hogy a kijelölt anyag jól elsajátítható-e a könyvből.

És álljunk meg egy pillanatra ennél a kérdésnél! Éppen az lenne a legnagyobb baj, ha a könyv koncepciótlan volna. Ha tehát Pogány Csabában egy kicsi kis lényegrelátás is lett volna, akkor nem oldalszám szerint sorolja fel a vélt vagy valódi elírásokat, hanem a fontosabbal kezd: a koncepciótlansággal. Az persze más kérdés, hogy a könyv elején világosan megfogalmazom a koncepciómat, vagyis koncepciótlanság vádját helyett azt kellett volna kimutatnia, hogy helytelen a koncepcióm. Ja, és persze ÉRVEKKEL. Mert érveket nem nagyon látok a támadásban.

Természetesen könyvem írásában komolyan figyelembe kellett vennem néhány didaktikai szempontot is. Didaktikai kérdés volt, hogy minél hamarabb jussak el értelmes programok írásáig, anélkül, hogy közben blokkdiagramokkal, kettes szöveggel, a munkához felesleges definíciók tömkelegével elriasztanám az olvasóimat a tárgytól.

Didaktikai megfontolásból van beiktatva a könyvbe a 48–54. oldalig terjedő, könnyen olvasható rész. Célja: akik kissé nehezebben haladnak az anyaggal, itt elidőzhetnek, míg a gyorsabbak ezt a részt átugorhatják. A FOR-NEXT-es ciklusképzés fontossága miatt volt szükségem egy ilyen „könnyített indításra”.

Két szempontból is didaktikai lépés volt, hogy (szemben egy korábbi verzióval) a számelméleti problémákat hoztam előre. Kísérleteim során ugyanis kiderült, hogy egyszerűbbek az analízis-programok, bonyolultabbak a számelméletek, de az analízisoktatásban jóval bonyolultabb fogalmi kérdéseket oktatunk, és ez nemcsak a diáknak, de a tanárnak is nagyobb terhet ró a vállára. Hasonló az oka annak is, hogy a fizikai részben csak két problémára szorítottam.

Didaktikai kérdés az is, hogy egy leegyszerűsített nyelven fogalmaztam és építettem arra, hogy az olvasó meg akarja érteni, mit mondanak (nem pedig arra koncentrálnak, hogy hol lehetne azt félremagyá-

rázni). Didaktikai kérdés az is, hogy az egységes tudományt próbálok bemutatni, sok területre térek ki.

És végül, az egyik legfontosabb didaktikai elvem, hogy az olvasót a szellemes megoldások keresése felé irányítom. Azt tanítom, hogy kis gondolkodással sokkal jobban meggyorsíthatjuk programjainkat, mint a legbonyolultabb programozási trükkökkel.

Pogány Csaba „komoly” támadásokat intéz itt és máshol ellenem, mert a 64. oldalon „lefelejtetem” Weierstrass tételének kimondásában, hogy korlátos zárt intervallumon folytonos függvény az, amelyet tetszőleges pontossággal lehet polinommal közelíteni. Számombanom bűnömöt, de az országban van még néhány száz tanár és matematikus, akit analízisre oktattam, és aki tanúsíthatja, hogy az analízis ezen részét kitűnően ismerem, és a vizsgán kérlelhetetlen szigorral kérdezem vissza.

A 64. oldal említett kijelentése azonban az „Egy kis filozófia, egy kis történelem” című mesejezetben van két mondat közé odavetve, annak magyarázatára, hogy miért juthatott eszébe valakinek egy gépet gyártani hatodfokú polinomok értékeinek kiszámítására. Semmi sem kötelezett, hogy erre az állításra tegyem a hangsúlyt. Még ha egy matematika tankönyvről volna szó! De miért lennének itt, egy mese közepén precízebb, mint egy fizika tankönyv a „precíz részeinél”!

No de álljunk meg egy pillanatra! Hogyan jut eszébe Pogány Csabának az a sok ellenpélda Weierstrass polinomközelítési tételére??? Hiszen való igaz, hogy az  $[x]$ -et, a  $\sin x$ -et és még kismillió más függvényt lehetetlen polinommal tetszőleges  $\epsilon$ -ra  $\epsilon$  pontossággal közelíteni a  $(-\infty; +\infty)$  intervallumon, mert semmilyen „nem polinom” sem lehet: a különböző közelítések különbözősége korlátos lenne, így kontans. Ha azonban korlátos zárt intervallumot veszünk, akkor már az az elrettentő  $[x]$  vagy a  $[-1, 1]$ -ben mindenütt folytonos, sehol nem differenciálható függvény szintén tetszőleges pontossággal közelíthető polinommal.

Pogány Csabának igaza van abban, hogy mesélés közben lehagytam a tétel egyik feltételét. Az érvelése a sehol sem differenciálható és az analitikus függvényekről, a Peano-görbéről azt sugallja, hogy nem igazán érti, mit mond.

Az olvasók többsége most a fejét kapkodja: „engem a számítástechnika érdekel, és nem az, hogy a Pogány és a Simonovits el tudnak-e vitatkozni egymással ezekről a nem középszerű dolgokról”. És én egyetértek az olvasóval, könyvemben sehol sem törekedtem arra, hogy Pogány Csabának imponáljak, elsősorban arra ügyeltem, hogy egy széles ismeretanyagot közöljek a diákkal a számítás-

Dr. Urbán János, az Országos Pedagógiai Intézet matematikai osztályának vezetője így ír: „Sajnálattal olvastuk lapjuk, a *µMagazin* 1986. évi 2. (márciusi) számában Pogány Csaba aláírásával megjelent kifejezetten rosszulindulatú, durva támadást Simonovits Miklós »Számítástechnika« című könyve ellen. Megalapozott, érvelő, tárgyilagos, segítőkész bírálatának örültünk volna. Kérjük, hogy a válasz jogán a tankönyvszerző választás a lap közölte — az olvasók tárgyilagos tájékoztatása érdekében.” Természetesen eleget teszünk a kérésnek. (A szerk.)

technika precizitási szintjén (vagy néhol ennél sokkal magasabb szinten), és ekközben rámutassak a számítástechnika kapcsolatára a matematikával, a fizikával, a biológiával és általában mindennel, ami körülvesz bennünket.

A tranzisztorról és általában a technikáról...

Ha írok egy száraz, unalmas tankönyvet, telekarom hosszú és értelmetlen definíciókkal, formalizálom a számítástechnika minden részletét, elmagyarázom a különbséget a vektor és a tömb között stb., akkor talán (???) megnyerhetem volna P. Cs. rokonszenvét. Eleven, érdekes könyvet akartam írni, melyben váltakoznak a könnyű és nehéz részek, de amelyekből sokat tanulhatnak. Hogy érdekesebbé tegyem, és mert önmagában is igen fontosnak tartom, néhány technikai érdekességet is beiktattam. Később Babage-ról írok. Megírom: „A gép megépítése azonban a kor technológiája mellett nem sikerült.” Később, a 369. oldalon egy másik technológiai problémára irányítom rá a figyelmet. Ezt írom: „Amikor Muchly és Eckert a munkát elkezdték, mindenki látta a feladat fontosságát, de azt is látták, hogy a terv (elektronikus számítógép építése) elég kockázatos. Az elektroncsövek akkori minősége, a gép tervezett 18 000 elektroncsöve és a 100 000 lépés másodpercenként igen magas követelménynek számított akkoriban. Így a gép könnyen megbukhatott volna a megfelelő technikai feltételek, a megfelelő minőség hiányában. A számítógépek életben maradásában döntő fontosságú volt a tranzistorok megjelenése.”

Most jön P. Cs. Kiveszi a legutolsó mondatot, majd mellé írja: „Vaskos tévedés. Jól mutatja a szerző járatlanságát szakmai kérdésekben.”

Nem hiszem, hogy egy mondat-sorozatból jogos egy mondatot kiragadni!

Nem látom P. Cs. érvelését. Csak prófétai kinyilatkoztatást. Végül, MEGINT NINCS IGAZA, amit az alábbiakból láthatunk:

Herman H. Goldstine a Neumann J.-féle számítógép-építő csoport egyik fő embere írt egy könyvet „A komputer Pascaltól Neu-



mannig" címmel (melyet talán hamarosan a magyar olvasók is elolvashatnak). Ennek a „Háborús fejlemények” c. fejezetében, a 152. oldalon ezt írja az ENIAC-kal kapcsolatban: „A kockázat (a gép építésével) hihetetlen nagy volt...” Egy Brainerd nevű igazgatót idéz: „Rá kell azonban mutatnunk, hogy ez (a gép építése) egy fejlesztési kutatás, és nincs garanciánk arra, hogy a kívánt eredmény elérhető. Azonban reális az esélyünk.”

A gondot éppen a sok alkatrész, a sok rádiócső okozta. Ezért a tervezők abban a korban különösen magas minőségűnek számító elektroncsöveket rendeltek.

Az ENIAC megvalósult, majd Neumann elveivel elkezdték építeni az EDVAC-ot is. Itt is komoly technológiai problémákkal kellett megküzdniük.

H. Goldstine könyve 195. oldalán egy Neumann-levelet idéz, melyben Neumann, miután javasolja (bizonyos esetekben) a 8-as számrendszer alkalmazását, rátér egy számunkra talán meglepő kérdésre, két speciális újdonságra, a 6AK5 és 6AS6 rádiócsövekre hívja fel Goldstine figyelmét.

Amint fent leírtam, mutatja, hogy a technológiai problémák igen fontosak voltak az első komputer megépítésében. Nem mutatja, hogy miután sikerült megépíteni az ENIAC-ot, pusztán rádiócsövekkel nem lehetett volna messze továbbjutni.

Nem szívesen vitatkozom ezen, hiszen nem jól definiált a kérdés. Amit ma komputerrel csinálunk, annak elenyésző részét sem lehetett volna csöves komputerrel megtenni. Az egyik legszínvonalasabb tudományos-ismeretterjesztő lap, a Scientific American (melyet ma már magyarul is olvashatunk), az 1977. szeptemberi számát a mik-

roelektronikának szenteli. Izgalmas cikkek, még ma is érdemes volna lefordítani őket! W. C. Holton a következőket írja a „Mikroelektronikai áramkörök nagyfokú integrálása (LSI)” c. cikke bevezetőjében: „Már 1940-ben felismerték a nagy sebességű és kapacitású számítógépek szükségességét, de ezek teljesítményét korlátozta az elérhető elektronikai technológia, mely rádiócsövekre épült. Az első teljesen elektronikus komputert, azt ENIAC-ot a Pennsylvániai Egyetemen fejlesztették ki 1945-ben. 18 000 cső volt benne. Nagyobb gép építése nem volt praktikus, naponta 24 óra lett volna a meghibásodott elektroncsövek megkeresése és kicserélése. A komputert az idő előtti végétől a tranzisztor 1947-es feltalálása mentette meg.”

Bevallom, én jobban hiszek Holtonnak (és másoknak, akik ugyan-ézt írják), mint Pogánynak. Miért? Mert a) Holton érvel, b) érvelése egyenes folytatása Goldstine-ének, c) mert érvelése mindenkinek meggyőző, aki tudja, hogy pusztán a hőelvezetés mennyi gondot okozhat.

Hogyan idéz Pogány Csaba?

A 277. oldalon szerepel egy feladat molekulásúly-számításra. A feladat egyike azon feladatoknak, melyek megoldása jól illusztrálja a strukturált programozás fontosságát, ha csak úgy nekesünk, előzetes terv, alapos átgondolás, esetleg részegységekre szétzedés nélkül, feltehetőleg sok időt kell eltöltenünk a program belövésével. Az egyik veszély az, hogy az input elején vagy a végén elügyetlenkedünk valamit, így üres string-re alkalmazzuk mondjuk a mid \$(..)\$ függvényt. Eléggé el nem ítérlhető módon erre előre felhívom a diák figyelmét: „Vigyázzunk az

INPUT w\$ elején és végén levő zűrökre! Teszteljük a programot az alábbi vegyületekkel...” Pogány Csaba reakciója: Egy jellemző példa a szakmai pongyolásokat jópofasággal elütő nagyszámú megnyilatkozás közül. Nem értem: az a jópofaság, hogy a „zűr” szót használtam hosszabb leírás helyett? Mit „ütöttem el” vele?

Más. Használunk a matematikában egy „akkor és csak akkor” kifejezést. A tapasztalat azt mutatja, hogy bár a kifejezés teljesen világos, a használata középiskolában, legalábbis a nem tagozatos osztályokban gondot okoz: azokat a mondatokat, melyekben szerepel, nehezebben értik meg. Természetesen megpróbáltam elkerülni a használatát. Mármost, amikor valamit éppen definiálunk, akkor egy széles körben bevett és nem kifogásolható szóhasználat, hogy nem is használjuk az „akkor és csak akkor”-t, hanem „akkor”-ral helyettesítjük. Pogány ezt is kifogásolja, lásd megjegyzését a 86. oldalra. (Hétköznapos szóhasználat: azt javasolja, hogy a félreértés elkerülése végett ezután az apák így fogalmazzanak: „Fiam, akkor és csak akkor kapsz biciklit a születésnapodra, ha jegyeid átlaga legalább 4,7802 lesz.”)

Még egy félreidézés: A 329. oldalon félreérthetetlenül megírom, hogy a szótárt, a telefonkönyvet... adathalmazoknak kezeltem. Vitatható lenne, hogy helyes-e az ADATBÁZIS kifejezés ezen a helyen. Pogány ezt nem vitatja. A bevezető szöveg végén teszek egy megjegyzést: „Amikor adatról beszélünk, többnyire olyan feljegyzésre gondolunk, amelyiknek egyik része ismert, a másikat keressük.” Külön kiírom elé, hogy „Megjegyzés”, ott szerepel a szövegben, hogy „többnyire”, és persze a vak is látja, hogy nem a bal-

lisztikus programjaim indítási adataira gondolok.

Mi hiányozhat Pogány Csabának? Talán csak azt szeretne volna, ha odaírom: „Ebben a fejezetben, amikor adatokról beszélünk...” Engem még az bosszant, hogy a Pogány Csaba-típusú kritikusok elégedettek, ha valaki hosszasan fecseg a semmiről, de azt nagyon precízen teszi, viszont elkeseredetten tiltakozni kezdenek, ha valaki emberi hangon szólal meg.

... Nincs a könyvben munkára nevelés...” írja Pogány Csaba. Nem értem. Mondhatnám, hogy a 7. oldalon „megirtam”.

„Fontos tudatunkba vénsni, hogy a gép nemcsak zseniális játékszer, de komoly munkaeszköz is, oktató és kísérleti eszköz is.”

De nem erről van szó. Arról, hogy a könyv arra nevel, bármit, amit könnyebben oldhatsz meg gépen, mint gép nélkül, azt programozd be. Használjuk a gépet mindenütt, ahol értelmes a használata. Arra neveltem a diákokat, hogy szeressék ezt a tárgyat. Élvezzék.

Lehet persze, hogy ezzel lenvelem a munkáról a diákokat, mint pl. mindenki, aki a komoly fizikai munkát gépi munkával próbálja helyettesíteni. Ezt vállalom. De vállalja-e Pogány Csaba, hogy csak az a munka, amit utálunk? Vagy hogy például a differenciálegyenletek számítógépes megoldása nem munka? Szótárprogramok készítése nem munka? Valóban nem értem! Talán feladatokat nem adok eleget? De hiszen tele van a könyv könnyű, közepes és nehéz feladatokkal.

Pogány sok mindent kifogásolt tankönyvemben, de ezek egyike sem a lényeghez tartozott. Szerintem lényegében egyikben sem volt igaz, de még ha mindegyikben igaz lett volna, az sem minősítette volna a tankönyvet.

SIMONOVITS MIKLÓS

## MÉG MINDIG

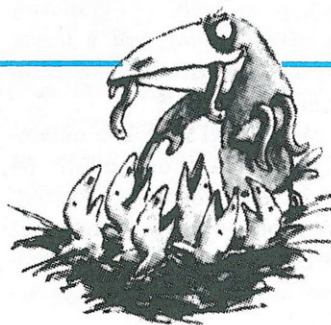
### Műsoron a számítógép!

1984 tavaszán indult a televízióban a Műsoron a számítógép c. sorozat, amelynek a célja az volt, hogy a tévénezők ismerkedjenek, barátkozzanak a számítógéppel és annak világával.

Ezzel egyidőben a Műszaki Könyvkiadó is megjelentetett egy könyvet, amely szintén a „Műsoron a számítógép” címet kapta. A könyvet eredetileg a — filmet is előállító — British Broadcasting Corporation adta ki, szerzői: R. Bradbeer—P. DeBono—P. Laurie.

A tévében dolgozók, nevezetesen Surányi Erika és Horváth János segítettek abban, hogy a könyv minél gyorsabban elkészüljön. Horváth János „sajátkezűleg” hozta — Angliából — az illusztrációkról a filmanyagot, így a lefordított könyv a filmsorozat indításakor már kapható volt a boltokban.

A könyv önmagában is igen értékes, magas színvonalú ismeretterjesztő mű, illusztráció anyaga igen gazdag: sok fényképpel, színes rajzzal, színes szövegkiemeléssel készült. A mű — miután rövid visszapillantást tesz a számítógépek történetébe — áttekintést ad arról, hogy a mindennapi életünkben vagy egyéb területeken hol használatosak és hol



használhatók majd a számítástechnikai eszközök; milyen számítógépek vannak; működésük min alapul. Leírja a feladatokat, amelyek számítógéppel elvégezhetők és azt, hogy ezeket a feladatokat hogyan oldja meg a számítógép. Komoly ismereteket könnyed formában tárgyal, s csak egy igen rövid BASIC „kóstolót” ad, kis eligazítást azoknak, akik a BASIC nyelvről is szeretnének valamit tudni.

A könyv végén szakkifejezé-

sek gyűjteménye található, amelyben a leggyakrabban előforduló szakszavak magyarázata van ábécébe szedve. Végül tárgymutató segíti az olvasót abban, hogy ha egy új számítástechnikai szakszóval találkozunk, meg tudja keresni annak jelentését.

Miért írunk most, 1986-ban erről a könyvről? Azért, mert ma is ugyanolyan friss, újszerű, ismeretei időtállóak. Kedves varjúfigurái változatlan érdeklődéssel kísérik végig a könyv oldalait.

A könyv olvashatóssága, sok illusztrációja miatt nagyon ajánlható iskolákban tanároknak, tanulóknak egyaránt. Javasoljuk, hogy a számítógépek iránt érdeklődők szintén olvassák el a könyvet. Érdemes!



## Hívószám-változások

A posta már évek óta alkalmazza a számítógépes beszéd-szintetizálást a gyakorlatban. Ezek között szövegű alkalmazások voltak, mint például annak a mondatnak a bemondása, hogy „A hívott szám megváltozott”. A számítógépes megoldás előnye ez esetben a magnetofonos módszerrel szemben, hogy nem kopik, és több hívószámra lehet rákötni.

A számítógépek számára testre szabottabb, igazi feladat az, amikor a beszéd-szintetizáló az üzenethez hozzáteszi a hívott fél új számát is. Erre a feladatra fejlesztette ki a Posta Kísérleti Intézet a Szegedi Postaigazgatóság megbízásából a HVA típusjelzésű automatát. 24, illetve 96 vonalas változatban készült el, ami azt jelenti, hogy ennyi „régi” vonalra lehet rákötni, hogy az új hívószámról tájékoztathassa a hívót. A posta általában a gyakran hívott, megváltozott központi számokra, például rendelőintézetek, nagyobb intézmények előző számaira köti rá ezeket az automatákat. A HVA számlálója vonalanként „méri” az automata „munkája” hatékonyságát. Amennyiben a hívások száma már nagyon lecsökken, a HVA adott vonalát át lehet kötni egy másik számra, vagy az egész HVA-t át lehet telepíteni.

Az elmúlt télen vetették be a HVA-kat először éles üzemben. Budapesten a Krisztina Központ átadásával járó nagy számváltozások okozta nehézségek csökkentésében vettek részt, sikerrel. Most nyáron Szegeden vannak jelentős telefonszám-változások, így az ország összes HVA-ját már Szegedre telepítették.

## Rádióamatőrök oktatása

Az MHSZ járszentlászlói rádióklubjában egy Commo-

dore 64 segíti az amatőröket. Az erre készített morzeoktató program önállóan és kitűnően tanítja a kezdőket, mivel nemcsak szabályosan, hanem ritmusosan is ad. A beprogramozott adás sebességét szabályozni lehet, és nagy előny, hogy a számítógép nem fárad, nem követ el hibát. A Commodore 64-et adóvevővel kapcsolták össze, így tudnak programozott szövegeket leadni.

## Oktatócsomagok

A korszerű, elektronikus oktatási eszközök nem csupán a hagyományos tantárgyak tanítására vethetők be hatékonyan, hanem önmaguk oktatására is. Ezt a módszert alkalmazza az LSI Alkalmazási Tanácsadó Szolgálat, amely egyidőben két oktatócsomaggal is a piacra lépett.

A *mikroszámítógép születésének titkai* című, 32 perces videokazetta áttekinti a hardver létrehozási folyamatát, a homoktól a mikroelektronikai elemgyártáson át egészen a mikroszámítógépek gyártásáig, továbbá a szoftverkialakítás legfontosabb elemeit. A vásárló a kazettához két tankönyvet is kap: az egyik az *Egyszerűen a mikroszámítógépről*, a másik pedig a *Népszerű elektronikai minilexikon*.

Az LSI ATSZ másik oktatócsomagja a Commodore 64 iránt érdeklődőknek készült. Van benne egy oktatólemez BASIC és Assembler programokkal, egy BASIC és egy As-

sembler tankönyv, valamint egy Commodore 64 információs kártya is.

Mindkét oktatócsomag izlées csomagolásban, egyenként 3500 forintért kapható.

## Kétszer gyorsabb

Népszerűek hazánkban az NDK gyártmányú mátrixnyomtatók, melyek elsősorban a MOM gyártáskooperációja révén kerülnek be az országba. Az idén két új nyomtatópáros jelent meg, melyek közös jellemzője, hogy az eddigi nyomtatók 100 jel/másodperces sebességével szemben 200-at nyomtatnak.

A K 6321 és 6322 típusok csak a behelyezhető papír szélességében különböznek: az egyikbe maximum 250, a másikba 420 mm széles papír fűzhető be. Az egy sorba nyomtatható jelek száma a betűszélességtől függően maximum 80–136, illetve 132–224 karakter lehet. A karakterek 9×7 pontos mátrixokból épülnek fel. Ennél szebb írásképet nyújt a 9×9-es betűmátrixot használó, szintén új nyomtatópáros, a K 6325 és a K 6326. Mind a négy új nyomtató a robotron kombinát sömmerdai gyárában készül.

## Növényvédelem

Az eddiginél is pontosabb és hosszabb távra szóló előrejelzésre lesz képes az idén a

MÉM Növényvédelmi és Agrokémiai Központja. 1986-ban már 12 növény károsítóinak fellépéséről ad rendszeres, friss tájékoztatást, és a távelőjelzésben 20 növényi kultúra 90-100 kártevőjére hívják fel a figyelmet szolgáltató hálózatukon keresztül.

A központ új szolgáltatásai közé tartozik az a mikroszámítógépes programrendszer, amelyet a búza és a kukorica, később a repce növénykárosítóinak előrejelzésére készítenek. A nagyüzemeknek a táblákon kell elvégezniük az előzetes adatfelvételt, majd a számítógépes program segítségével a korábbinál hatékonyabban védekezhetnek.

## IC – mint modem

Az amerikai SIERRA SEMICONDUCTOR CORPORATION komplett modemet készített egyetlen IC-be sűrítve. Az SC 11004 jelű IC ellátja a digitális és analóg funkciókat, és 300 vagy 1200 Baud sebességű szinkron vagy aszinkron üzemmódot biztosít. Tetszés szerint működtethető az amerikai BELL 212A szabvány vagy az európai CCITT ajánlás (V. 22.) szerint is. A modem ára 100 DM körüli. A modemhez az SC 1107 jelű áramkört hozzákapcsolva lehetővé válik a tetszőleges számítógéphez történő csatlakoztatás. Az SC 11008 vezérlő IC-t hozzákapcsolva soros interfészt valósíthatunk meg.

# NOVOTRADE

## Exportfejlesztési munkához

olyan szoftverfejlesztő szakembereket keresünk, akik a legújabb személyi és házi számítógépeket szívesen megismernék, és azokra üzleti szoftvertermékek fejlesztését vállalják.

**Jelentkezés a gyakorlat, referencia termék, az ismert mikroprocesszorok fajtáinak megjelölésével Verő Péternél vagy Séra Veronikánál.**

**Cím: NOVOTRADE RT.**

1136 Budapest, Fürst S. u. 24–26.

Telefon: 122-095, 122-047, telex: 22-7673



~~688 000,-~~  
**496 000,-**

## PC 20 LÍZING A NOVOTRADE RT-NÉL

**NEMCSAK PÉNZT, IDŐT IS MEGTAKARÍTHAT,  
HA ÜZLETÜNKET MEGLÁTOGATJA!**

**Az IBM kompatibilis számítógépek szoftverválasztéka széles körű,  
és különböző cégek forgalmazzák...**

### NÁLUNK MINDEZT EGY HELYEN MEGVÁSÁROLHATJA

- Főkönyv és folyószámla-könyvelési rendszer
- Anyaggazdálkodási rendszer
- Bér- és munkaügyi rendszer
- Állóeszköz-gazdálkodási rendszer
- Kereskedelmi rendszer
- Személyzeti rendszer
- Mérnöki program
- Matematikai és statisztikai alkalmazások
- SPEECH modul (beszédanalízis és szintézis)
- ÉKN nyilvántartó, számlázó programcsomag
- Kórházi nyilvántartó rendszer
- KLAUVGEN (ékezetes klaviatúrakezelő)
- Szarvasmarha-nyilvántartás
- Vetésterv-optimalizálás
- Műtrágya-szaktanácsadás
- TEXTMAN szövegfeldolgozó rendszer
- TEXTPRINT szövegfeldolgozó rendszer
- PERSTAT statisztikai programcsomag
- 256 kb-át operatív tár bővítése 512 kb-átra
- 384 kb-át operatív tár bővítése 640 kb-átra
- DACCESS III. I. 10. relációs adatbázis-kezelő
- FWINDOW 1. 10. ablakkezelési rendszer
- OCTOPUS relációs adatbázis-kezelő
- PROPOS-1 operációs rendszer
- PROMOS operációsrendszer-kifejlesztés (multi user)
- GRAFOS operációsrendszer-kifejlesztés (grafikus driverek)
- MENTOL mágneskazettára mentő-visszatöltő program
- BASIC nyelvi rendszer
- C nyelvi rendszer
- COBOL nyelvi rendszer
- BASIC oktatórendszer
- GRAFLIB fejlesztői könyvtár
- CREDIT grafikus editor
- PERDATIN adatgyűjtő-lekérdező rendszer
- Kommunikációs adapter
- VT 52 emulációs szoftver
- Lokális hálózati rendszer kulcsrakész telepítése

**Oktatásunk keretében  
a gyakorlatban megtanulhatja  
a COMMODORE PC 20 kezelését.**

**A lízingdíj magába foglalja az MSDOS 2.11 és GWBASIC szoftvereket,  
magyar nyelvű szakkönyvekkel.**

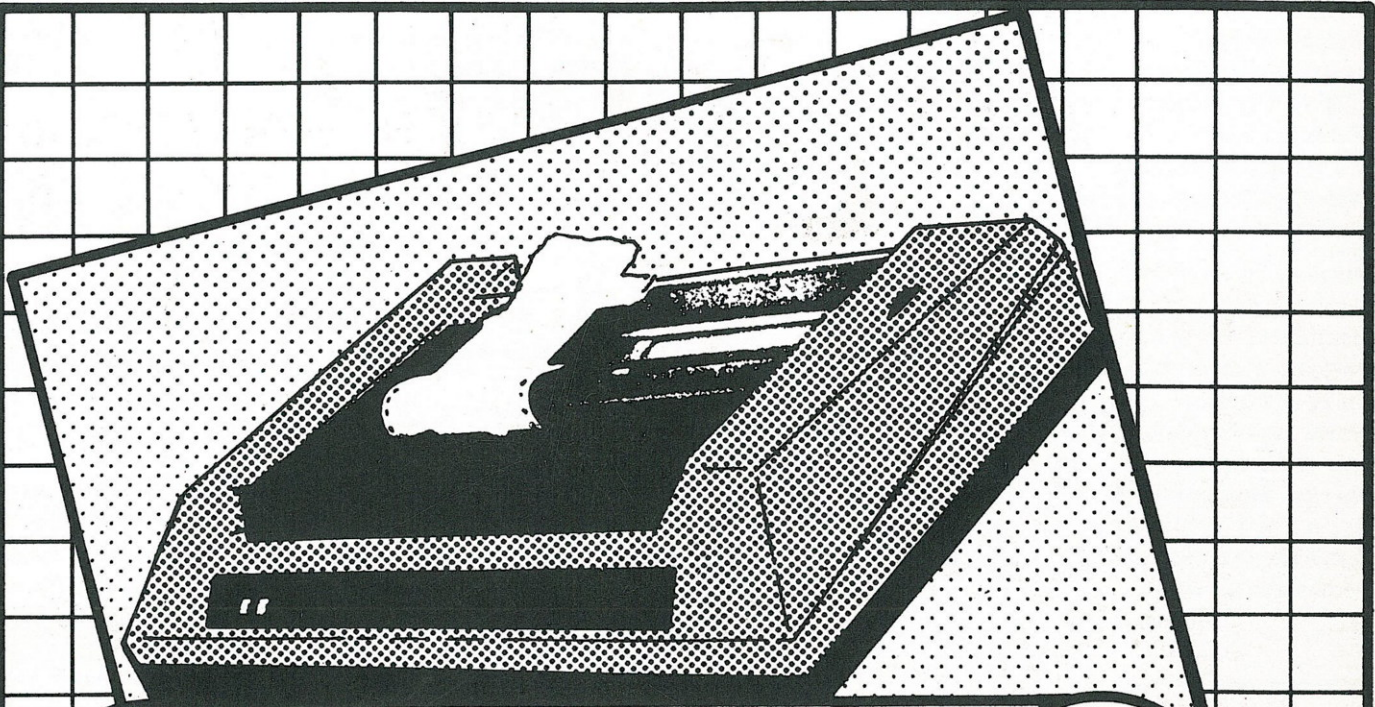
**Üzletünk ingyenes szaktanácsadással,  
információszereléssel várja az érdeklődőket.**

**NOVOTRADE**

Bp. XIII., Balzac u. 35.  
Telefon: 402-954. Telex: 22-5959  
Nyitva: hétfőtől péntekig 9-18<sup>h</sup>

**2c**





## FELADATA VÁLOGATJA

A számítógép munkájának eredménye leggyakrabban nyomtatásban jelenik meg. Nem könnyű eldönteni: a meglévő számítógéphez és az elvégzendő munkához melyik típusú nyomtató a legalkalmasabb.

Forduljon a Videotonhoz: gyártási programunkban minden feladatra talál megfelelő nyomtatót.

- VT 21200 (80 oszlopos mátrixnyomtató)
- VT 21400 (132 oszlopos mátrixnyomtató)
- VT 23000 (300 sor/perc szalagos nyomtató)
- VT 27061 (600 sor/perc sornyomtató)
- VT 27091 (900 sor/perc sornyomtató)

Felvilágosítás: 804-133

**A VIDEOTON PROGRAMJA – A JÖVŐ PROGRAMJA**

---