



MIKROSZÁMÍTÓGÉP  
MAGAZIN

A NEUMANN JÁNOS SZÁMÍTÓGÉP-  
TUDOMÁNYI TÁRSASÁG LAPJA

**1986**

szeptember

Ára: 30 Ft



# Fiatalok! Figyelem!

Következő számunkban három hónapon át tartó rejtvenypályázatot indítunk! Ugyanakkor közöljük majd a megfejtések beküldési határidejét is. Ha több hibátlan megoldást kapnánk, kisorsoljuk azt a szerencsés nyertest, aki az alábbi — Spectrum számítógépekhez használható — kiegészítő eszközök közül kiválaszthatja a számára legmegfelelőbb nyereményt!



- INTELLIGENS JOYSTICK INTERFACE
- EPROMÉGETŐ és ELLENŐRZŐ (16K — 32K-hoz)
- KÜLSŐ PROGRAMTÁR
- UNIVERZÁLIS PRINTER INTERFACE
- FÉNYCERUZA
- MEMÓRIABŐVÍTÉS (16K → 48K)
- LAPOZÓS MEMÓRIA (48K → 80K)
- SPECTRUM — TON hangosító
- EGROMÉGETŐ (2716 → 27128)
- SPECTRUM ROM I. (módosított)

Fentiek megtekinthetők a DIGITÁL Számítástechnikai Szaküzletben! Cím: 1026 Bp., Szilágyi Erzsébet fasor 35.

**SOK SIKERT KÍVÁN A SZERKESZTŐSÉG!**



## A NEUMANN JÁNOS SZÁMÍTÓGÉP- TUDOMÁNYI TÁRSASÁG ÉS A KISZ KÖZPONTI BIZOTTSÁG LAPJA

**A kiadvány  
a Tudományos Szervezési  
és Informatikai  
Intézetnél  
együttműködve készül**

**A szerkesztőbizottság  
vezetője:  
Kovács Győző**

**E számunkat  
szerkesztették:  
Bakos Tamás  
(programozástechnika)**

**Broczkó Péter  
(hírek)  
Kovács Győző  
(levelezés)**

**Lindner László  
(sakkprogramozás)  
Petrőczy Judit  
(könyvek)**

**Simonyi Endre  
(klub)**

**Varga András  
(iskola — számítógép)**

**PR menedzser:  
Pálhalmi Vali**

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

**Kiadja az Ifjúsági Lap-  
és Könyvkiadó Vállalat  
Felelős kiadó:  
dr. Petrus György  
Igazgató  
Kiadóhivatal:  
1065 Budapest VI., Révay u. 16.  
Telefon: 116-660**

**Terjeszti a Magyar Posta  
Előfizethető a hírlapkézbesítő  
hivataloknál  
és a Posta Hírlapelőfizetési  
és Lapellátási Irodáján  
(1900 Budapest V.,  
József nádor tér 1.)  
vagy átutalással a 215-96 162  
pénzforgalmi jelzőszámra.**

**Megjelenik havonta  
Egy szám ára 30,— Ft  
Előfizetési díj:  
egy évre 360,— Ft  
fél évre 180,— Ft  
Külföldön terjeszti  
a Kultúra,  
1389 Budapest, pf. 149.  
és a Magyar Média  
1932 Budapest, pf. 279.  
86—253**



**Szikra Lapnyomda  
Budapest (86-3721)  
Felelős vezető:  
Csöndes Zoltán vezérigazgató**

**INDEX: 25 629  
ISSN 0236-6088**

**Címképünk:**

**A KODEX 2000  
szövegszerkesztő**



### Tartalom

μ'87	2
A minőségügy — közügy	12
Assemblerek, cross-assemblerek	13
A CAD alapelemei Commodore mikroszámítógépeken	21
Helyi hálózatok	24
Adok — veszek — cserélek	31
Barátunk, a számítógép	32
Ki ad magyarázatot?	40
Külföldön olcsó — belföldön drága?	46

### ISKOLA — SZÁMÍTÓGÉP

A Commodore 16 mint iskolaszámítógép	3
Az ismeretlen C16	3
Az 1986. évi „Nemes Tihmér” középiskolás számítástechnikai versenyről	5

### DIÁKROVAT

A HT—1080Z lemezrendszer utasításai és függvényei	8
Amit nem tud a HT-val	9
Nagy felbontású grafika	10

### PROGRAMOZÁSTECHNIKA

BASIC és gépi kód	18
Z80 programozási gyakorlatok	19

### μPROGRAMOK

Sztringváltozóban elhelyezett összefüggést kiértékelő szubrutin	27
GET # helyett INPUT #	29
Beviteli rutin	30
UNIN példaprogram	31

### μKLUB

Az Építsünk számítógépet sorozat egy hibájáról	33
PTA—4000 BASIC	34
Forgástestek és felületek	35
TURBO interfész ZX—Spectrumhoz	38

### SAKKPROGRAMOZÁS

Bitek és figurák	39
------------------	----

### JÁTÉKPROGRAMOK

AZ OLVASÓ ÍRJA	42
----------------	----

FÓRUM	44
-------	----

KÖNYVEK	45
---------	----

HÍREK, ÉRDEKESSEGEK	46
---------------------	----

# Még egyszer a μ '87-ről

„Az alkotó gondolkodás általában kérdések, feladatok, problémák felmerülésekor indul meg. Ilyen értelemben nemcsak az egész emberiség számára új felfedezéseket, művészi teljesítményeket stb. kell alkotásnak minősítenünk, hanem a mindennapi élet mindenkor adódó, az egyén számára új, apró problémáit is, pl. egy bevásárlási program tervezését (milyen sorrendben és milyen járműveken látogatjuk sorra az üzleteket a legrövidebb úton), egy erősítő tervezését, ahol csak az erősítés és a rezgésszámhatár más, mint az eddig ismert kivitelekben, egy reprodukáló művész feladatát stb.”

(Nemes Tihamér: Kibernetikai gépek)

A kedves olvasó joggal gyanakodhat, hogy elterveztem a dátumot és tíz hónappal korábban írtam meg a szerkesztőségi cikket. A kedves olvasó téved, az írás szándékos, a második Országos Mikroszámítógépes Találkozóhoz az ideinél jobb és hatékonyabb előkészítést szolgálja.

A μ'86-ról szóló hivatalos és nem hivatalos értékelések megítéslő módon elismerek voltak, ezért különös gondossággal szeretnénk a jövő évi találkozóra felkészülni, 1987-ben egy, az ideinél jobb és még több érdeklődőt vonzó programot szeretnénk összeállítani, ehhez kérjük az informatika alkalmazása iránt érdeklődők segítségét és támogatását.

Talán egyetértetek velem, hogy ismét, immáron harmadszor is meg kell rendeznünk „A számítástechnika mindenkiért, a számítástechnika mindenkié” kiállítást. A célt nem szeretnénk változtatni, ezen a kiállításon azokat az eszközöket, programokat, oktatási anyagokat, módszereket mutatjuk be, amelyeket kevés tanulóval, kevés szakértelemmel bárki gyorsan alkalmazhat. Az idein a szakma elismert nagy intézményei mellett sok kisvállalkozás, szövetkezet, leányvállalat és gazdasági munkaközösség vonult fel, főleg a fővárosból. Jövőre azt szeretnénk, ha a vidék is méltóan képviselve lenne a kiállításon. Úgy gondoljuk, hogy az NJSZT megyei területei szervezetei mutassák be a helyi vállalkozások eredményeit és így a μ'87 egyben országos tapasztalatcsereére is lehetőséget ad.

Változatlanul reméljük, hogy a társadalmi programok mellett meg tudjuk rendezni a szocialista országok első elektronikai vagy inkább informatikai szakkiallítását is. A tervezett együttműködés a Tavaszi Fesztiválért felelős IPV-vel, a kiállítások megszervezésében gyakorlott HUNGEXPO-val, az elektronizálási programért felelős főhatóságokkal és természetesen az érdekelte intézményekkel és vállalatokkal nem utolsósorban a Neumann János Számítógéptudományi Társaság szakembereivel egészen biztosan tovább növeli a kezdeményező Tavaszi Fesztivál tekintélyét.

Jövőre is meghívjuk az egészségkárosultak szervezeteit a kiállításra. Úgy gondoljuk, már az idein is bizonyítottuk, hogy a vakok és csökkentlátók, a mozgássérültek a számítástechnikában teljes emberként dolgozhatnak. Ma az a legnagyobb probléma, hogy erre viszonylag kevés a lehetőségük. Az igazi megoldás az volna, ha otthoni munkahelyeket lehetne kialakítani és a mozgásukban korlátozott emberek távbeszélő vonalra kapcsolt készülékekkel tudnák munkájuk eredményét a munkahelyükre küldeni. Egy távkapcsolatra alkalmas otthoni számítógéppel egyszerre több munkahelyen is összeköttetést lehet tartani, így az egészségkárosult emberek kevésbé éreznék, hogy kiszolgáltatott helyzetben vannak.

A vártnál sokkal több érdeklődőt vonzott az oktatási és általában az iskolai alkalmazási bemutató. Tanárok és diákok nagyszámú nézőközönségnek mutatták be, hogy a számítógép nagyon rövid időn belül az iskolai oktatás nélkülözhetetlen segédeszközévé fog válni. Az általános iskolák gépésztégye most kezdődik, a középiskolákban is egyre több tanár és diák vált az informatika „tudósává”, ezért valószínűnek tartom, hogy a jövő évi bemutatók során is sok érdekes alkalmazást láthatunk.

Szeretnénk újabb pályázatokat kiírni és versenyeket indítani. Feltételezzük, hogy a hardver és szoftver amatőrök részben továbbfejlesztik rendszereiket, részben újakat terveznek, így a 87-es munkák legalább olyan sikert aratnak, mint az ideiek.

A napi programot anketókkal, kerekasztal megbeszélésekkel zárjuk, felhasználva az alkalmat a számítástechnika aktuális problémáiról való véleménycsere.

Jövőre — talán ez lehetne a μ'87 jelmondata is — többet szeretnénk foglalkozni az informatika és a művészet kapcsolatával. Az a hangverseny, amelyet az idein az NJSZT zenei bizottsága rendezett, emlékezetes szép sikert aratott. A zenei bemutató nagyon sok érdeklődőt hozott ki a kiállításra, azok is eljötték, akiket a számítástechnika kevésbé, inkább a mai modern zenei törekvések érdekeltek. A Budapesti Tavasz Fesztivál rendezőinek azt javasoltuk, hogy 87-ben a fesztivált egy számítógépes-zenei hangversennyel lehetne nyitni, amelyen részben magyar szerzők, részben külföldiek — elsősorban Boulez — műveit mutatnák be. El szeretnénk hozni és meg is mutatni azokat a számítógépeket, amelyeket komponálásra, illetve zeneművek lejátszására konstruáltak.

A másik tervezett esemény a számítógéppel készült grafikai alkotások bemutatása. Az a szándékunk, hogy nemcsak hazai, de külföldi alkotók munkáit is kiállítjuk, ezzel is bizonyítva, hogy a számítógép nemcsak a tudomány és az üzleti élet, hanem a művészet számára is újabb lehetőségeket kínál. Biztosan sokan ismerik a kibernetikus szobrokat, másnéven idő- és térstruktúrákat, amelyek, ha úgy tetszik, informatikai alkotások. Szeretnénk bemutatni hazai képzőművészeink alkotásai mellett például Nicolas Schöffer magyar származású francia művész munkáit, aki a világon talán legelőször készített ilyen műveket.

Nagyon örülnénk, ha sor kerülhetne más műzsáktól ihletett alkotók (pl. irodalmárok) bemutatkozására is, nagyon szívesen fogadnánk újabb javaslatokat.

Ennek az írásnak bevallott célja, hogy a μ'87 megrendezéséhez munkatársakat találjunk, újabb ötleteket kapjunk, hiszen csak így válhat az esemény azzá, aminek a neve után válnia kell, a mikroszámítógépet alkalmazók igazi országos fesztiváljává.

KOVÁCS GYÖZÖ

# A Commodore 16 mint iskolaszámítógép

Lapunkban vita indult a C16 körül. Több olyan kérdés is felvetődött, amelyre nem közismert a válasz. Hogy a vitát szűkebb mederben tartsuk, most ilyen kérdésekre válaszolunk.

Hogyan lett a C16-ból iskolaszámítógép? A terveknek megfelelően a Tudományszervezési és Informatikai Intézet már 1984-ben elkezdett foglalkozni az általános iskolák számítógéppel való felszerelésével. A középiskolákban elterjedt HT-1080Z nem jött számításba, részben a szükséges mennyiség, részben ára, részben pedig szolgáltatásai miatt. A piacon akkor hozzáférhető áron, megfelelő szolgáltatásokkal csak a Primo állt rendelkezésre. Az iskolákban az első gyártási sorozat példányaival végzett kísérletek azonban megbízhatósági problémákat vetettek fel.

Ekkor kapóra jött egy ajánlat Commodore 16 gépek szállítására. A beszerzés nem igényelt devizát — a művelődési tárcának erre a célra nem volt devizája —, és a gépek minden eddiginél olcsóbbak voltak, ami azt jelentette, hogy a rendelkezésre álló keretből jóval több ilyen számítógép vásárlására tellett, mint amennyire korábban gondolni lehetett. A szakértők szerint a gépnek nem volt olyan hátránya, amely indokolta tette volna a rendkívül kedvező ár-teljesítmény viszony elutasítását. Így került be 2000 db C16 az országba. A devizakitermelést a Skála-Coop vállalta magára.

A gépeknek volt magyar ékezetes betűkészletük, de nem a nálunk szokásos elrendezésű billentyűzet. A magyar nyelvű felhasználói kézikönyv az eredeti angol alapján készült, amely sajnos nagyon sok hibát tartalmazott. Ezeknek csak egy részét sikerült kiszűrni a rendelkezésre álló rövid idő alatt. A többször hivatkozott kapcsolási rajz, monitorleírás a német kiadásban található, amelyet akkor nem sikerült beszerezni. Sajnálatos módon az eredeti kézikönyvben — és így a magyar nyelvűben is — hivatkozott *Programozói referencia kézikönyv* sokáig nem jelent meg angolul sem. Jelenleg kézirata alapján fordítják, de sajnos a legényesebb felhasználókat ez sem fogja teljesen kielégíteni. Ezért készül az ismert Becker-sorozatból a C16 ROM listáját is tartalmazó könyv fordítása is, mindez a Novotrade Rt-nél.

Az iskolák a gépeket a hozzájuk tartozó *Datasette* 1531 kazettás tárolóval, 3 db *Felhasználói kézikönyvvel*, 1 db bemutató kazettával (két játéka, egy rajzoló- és egy

sakkprogram), ezek leírásával, egy a C16 BASIC nyelvbe bevezető önképző tanfolyami jegyzettel (*Bevezetés a BASIC nyelvbe*, 1. rész. Novotrade) és az ehhez tartozó 2 db kazettával, 1 db *Hetedhét Commodore 16* könyvvel (Révbíró—Pál, Novotrade), valamint a gépre helyezhető *Emlékeztetővel* kapták meg.

A *Felhasználói kézikönyv* felhívja a használók figyelmét arra, hogy színes képet és hangot (fekete-fehér kép mellett is) csak kétnormás készülékek kapnak. Egyébként minden mai hazai gyártású tévé teljesíti e követelményt.

A C16 kiválasztásával egyidejűleg az Országos Pedagógiai Intézet irányításával megindult a kísérleti munka, egyelőre bizonytalan üzletekből vásárolt gépekkel. Többek között ennek eredményeként lehetett a megyei Pedagógiai Intézetek szakemberei részére ez év januárjában olyan tanfolyamot tartani, amely előkészítette az általuk tartandó megyei tanfolyamokat, és amelyen számos C16 segédprogram, bemutatóprogram és oktatóprogram cserélt gazdát. Ugyancsak ennek eredménye lett a Novotrade-nél megjelent *Feladatgyűjtemény C16 számítógéphez általános iskolásoknak*, illetve a *Tanári segédkönyv a C16 számítógéphez készült általános iskolai feladatgyűjteményhez* c. könyv. (Szerzőjük Bencsikné Takács Márta.) Az Országos Oktatástechnikai Központ *Mikroszámítógép-programok* sorozatának 3. kötete ugyancsak tartalmaz már C16-os programokat.

Az új elektronizációs oktatási program keretében meghirdetett hardverpályázat eredményeképpen — mint lapunk már közölte — az általános iskolai programban továbbra is számításba jön a C16, illetve a vele kompatibilis 64 kb-átos Commodore Plus/4. A vállalkozó szállítók biztosítják a devizát, a Commodore felelős vezetői szerint gyártják a jövőben is e gépeket. Várható ugyan a nyugati piacokon a szoftverkiadás bővülése, de oktatásunkban — az eddigiekhez hasonlóan — saját szoftverfejlesztésre vagyunk utalva. Ehhez jó alapot ad a 3.5 verziójú, igen kiváló BASIC, valamint a befejezéshez közeledő FORTH-fejlesztés és a már megszületett oktatási programok. De továbbra is várjuk olvasóink ötleteit, amelyek megkönnyíthetik a programfejlesztéseket.

DR. VARGA ANDRÁS

Tudományszervezési és Informatikai Intézet

## Az ismeretlen C16

# A KERNAL rutinok

A C16 lelkivilágának ismertetését a KERNAL rutinok leírásával folytatom. Ezek a rutinok a gép operációs rendszere szempontjából alapvető feladatokat látnak el. Különleges tulajdonságuk, hogy a legtöbb Commodore gépen azonos módon lehet használni őket. Hívásuk előkészítése, belépési címük és működésük lényegi része a különböző géptípusokon megegyezik egymással.

A belépési pontok a memória SFF lapjának felső címein helyezkednek el. Itt közvetlen és közvetett ugróutasítások sorozatát találhatjuk. Ezeket keresztül jut a vezérlés azokra a helyekre, ahol a tényleges végrehajtás történik.

A C16-on lévő KERNAL rutinoknak most kb. 1/3-át ismertem. Igyekeztem úgy válogatni, hogy az összefüggő rutinok lehetőleg ne szerepeljenek külön részben; egyedül a BSOUT-nál tettem kivételt.

Az egyes eljárások leírásánál eltértek az előző részben követett módszertől: nem adom meg külön, hogy az említett rutin mely címen hívható a C64 és VC20 gépeken, mert ez ebben az esetben felesleges. Néhány hivatkozott rendszerváltozó címmel is adós maradok. Most nem fontosak, később majd pótolom őket.

<b>RAMTAS</b>	<b>\$\$\$F87</b>	<b>65415</b>
Megvizsgálja a RAM-ot és beállítja az ezzel összefüggő rendszerváltozókat. (A VC20-on ez a KERNAL rutin hiányzik.)		
<b>RESTOR</b>	<b>\$\$\$F8A</b>	<b>65418</b>
Beállítja, illetve helyreállítja a \$0312—\$0331 (786—817) címtartományban lévő rendszervektorokat. A C64 és VC20 típusú gépeken ezek a vektorok a \$0314—\$0333 (788—819) címeiken található, azonos sorrendben.		
<b>VECTOR</b>	<b>\$\$\$F8D</b>	<b>65421</b>
A RESTOR által is kezelt vektorokról másolatot készít vagy a felhasználó által kívánt értékre állítja be azokat, a carry bit értékétől függően.		

Hívása előtt egy 16 bájtos terület címét kell az X/Y regiszterpárban elhelyezni, úgy, hogy az X-be a cím alacsony helyérté-

kü bájta, Y-ba a magas helyértékű bajt kerüljön. C=1 esetén a rutin a vektorok tartalmát az így meghatározott területre másolja, C=0 esetén pedig a fenti módon megadott címtől kezdődően a következő 16 bajt tartalmát a rendszervektorokba írja.

SCNKEY \$FF9F 65439

A lenyomott billentyűhöz tartozó ASCII kódot a billentyűzetpufferbe küldi. Ha híváskor nincs lenyomott billentyű, akkor háttástan. Általában a GETIN rutinnal célszerű együtt használni.

READST \$FFB7 65463

Az 1/0 állapotzó tartalmát az A regiszterbe tölti. (Ez a változó — mindhárom géptípuson — az \$90 címen található. Az állapotzó tartalma 0, ha az utójára végrehajtott beviteli/kiviteli művelet végrehajtása hibátlan volt, illetve 64, ha az olvasási művelet során a rendszer fájl-vége jelet észlelt. Minden egyéb érték átviteli hibát jelent.

BSOUT \$FFD2 65490

Az A regiszter tartalmát az aktív kiviteli csatornára küldi. Ennek részletezésére a következő alkalommal visszatérek. Alaphelyzetben az a karakter kerül a képernyőre, melynek ASCII kódja az A regiszterben van.

SETTIM \$FFDB 65499

Beállítja a 0 lapon lévő rendszeróra (angolul Jiffy Clock) tartalmát. Ez 1/60 másodpercenként számlál egyet. Az óra 3 bájtos, beállításá előtt az A regiszterbe kell tenni a kívánt érték legmagasabb helyértékű báját, az X regiszterbe a középső, az Y-ba a legkisebb helyértékű báját.

RDTIM \$FFDE 65502

Ez a rutin a rendszeróra aktuális tartalmát az A, X és Y regiszterekbe tölti, a SETTIM rutin leírásánál ismertetett sorrendben. A BASIC TI nevű fenntartott változójának az értékének megfelelő eredményt ad, eltérő formában.

STOP \$FFE1 65505

Megvizsgálja, hogy a híváskor a stop billentyű le van-e nyomva. Ha igen, akkor (a következők számban ismertetendő) CLRCH KERNAL rutin hívásával lezárja a megnyitott fájlokat és az állapotregiszter Z-bitjét 1-re állítja. Ha a stop billentyű nincs lenyomva, a Z-bit értéke 0 lesz.

A SYS utasítással vagy az USR függvénnyel hívott gépi kódú alprogramok futása alatt a stop billentyű lenyomása hatástalan szokott lenni. Ezt a C16-on a következő utasítások megfelelő helyre való beírásával kerülhetjük el:

...

JSR \$FFE1

BEQ \$867E

Megfelelő helyen olyan pontot értek, ahová a programrész futása során elég gyakran adódik a vezérlés.

GETIN \$FFE4 65508

A billentyűzetpufferben található következő karakter az A regiszterbe teszi, és a karakter eltávolításával aktualizálja a billentyűzetpuffert. Ha a puffer üres, az A regiszterbe 0-t visz.

Példa az SCNKEY és GETIN rutinok együttes használatára:

CIKLUS JSR \$FF9F ; SCNKEY  
JSR \$FFE4 ; GETIN

CMP # \$00

BEQ Ciklus

A programrész egy billentyű lenyomására vár. A lenyomott billentyűhöz tartozó ASCII kód az A regiszterbe kerül, a további feldolgozás során felhasználható.

UDTIM \$FFEA 65514

Megnöveli a 0 lapon lévő rendszeróra értékét. A gép hardver megszakítási (interrupt) rendszere hatvanad másodpercenként hívja ezt a rutint. Használata csak akkor célszerű, ha a hardver megszakítási rendszert SEI utasítással kikapcsoltuk.

SCREEN \$FFED 65517

Az X regiszterbe teszi a képernyő oszlopainak, az Y regiszterbe a soroknak a számát. Ez az egyes géptípusoknál állandó, nem veszi figyelembe a képernyőméret esetleges szoftver úton történő módosítását. Jelentősége a más géptípusokon is futtatható programok írásánál van, de ez — úgy tudom — nagyon ritka.

PLOT \$FFF0 65520

A carry értékétől függően beállítja vagy lekérdezi a kurzor pozícióját. Beállításnál a carryt 0-ra állítjuk (CLC-vel), az X regiszterbe visszük a kívánt kurzor sorának számát, az Y-ba az oszlop számát, azután hívhatjuk a PLOT rutint. Lekérdezésnél a carryt 1-re állítjuk, majd a PLOT hívása után az X/Y regiszterpárban megkapjuk a kurzor helyének sor/oszlop koordinátáit.

Fontos! A sorok és oszlopok számozása 0-tól kezdődik. Célszerű arra is ügyelni, hogy az X regiszterbe kerül a függőleges koordináta, az Y-ba a vízszintes, éppen ellenkezőleg, mint ahogyan azt a geometriában megszoktuk.

A PLOT rutint a C64 és VC20-as gépeken gyakran hívják BASIC-ből is a kurzor beállítására. A C16-nál ezt a CHAR utasítás felelősségé teszi.

BARNA LÁSZLÓ

## Önkritika a „Hardcopy C16-hoz” című cikkhez

A  $\mu$ M 1986. júniusi számában megjelent cikkben helytelen megállapítás volt, hogy a monitorszolgáltatások nyomtatásához külön hardcopy program szükséges. Természetesen ugyanolyan egyszerűen nyomtathatók mint a BASIC listák, a CMD utasítás alkalmazásával. Azért marad még elég eset, amikor szükség van a programra!

VARGA ANDRÁS

## Helyreigazítás

A lapunk 1986/5. számának 24—25. oldalán közölt  $\mu$ 86 riportban durva hibát követtünk el. A cikk írója Kozma Lászlónéval készített interjút és nem Kalmár Lászlónéval, ahogy riportjában írta. Gondatlanok voltunk, nem tudjuk megmagyarázni, hogyan történt. Kozma Lászlónétól és Kalmár Lászlónétól is elnézést kérünk.

A szerkesztőség

Felhívjuk kedves érdeklődő Olvasóink figyelmét, hogy az 1986/6-os (júliusi) számunkban a 21. oldalon közölt ATARI Szekcióval kapcsolatos helyes telefonszám:

128-266

(Hirschler Tamás)

**Tájékoztatjuk  
kedves megrendelőinket,  
hogy az oktatási  
programok forgalmazását  
1986. szeptember 1-jétől  
ismét intézetünk végzi.**

**Jelenleg a HT 1080Z  
számítógépekre készített  
oktatóprogramok  
széles választékával  
állunk az érdeklődők  
rendelkezésére.**

Címünk: 1372. Budapest,  
Postafiók 454.  
XI., Egri J. u. 1-9. „E” ép.  
Telefon: 453-197

**Tudományszervezési  
és Informatikai  
Intézet**

## Az 1986. évi „Nemes Tihamér”

# középiskolás számítástechnikai versenyről

Mint azt valószínűleg többen is tudják, a Neumann Társaság kezdeményezésére, más intézményekkel együttműködve, már 1980 óta szervezünk különféle számítástechnikai versenyeket középiskolások részére. Tavaly kapta meg a számítástechnika azt a jogot, melyet az iskolai szaktárgyak országos tanulmányi versenyének győztese már régebb óta élveznek: az első tíz helyezett mentesítik matematikából az egyetemi, főiskolai felvételi alól. Ugyancsak tavaly, az országos matematikai versenyek mintájára neveztük el versenyünket a kibernetika egyik legelső kiemelkedő hazai művelőjéről, terjesztőjéről, Nemes Tihaméról.

A számítástechnikai versenyzetéseink immár ötves hagyománya, hogy kétfordulós, mégpedig oly módon, hogy az első fordulón számítógép használatát nélkül tesztjük próbára versenyzőinket. Ennek egyszerű magyarázata az a kényszer, hogy napjainkig még nem volt meg a reális lehetőség annak, hogy az országban egyazon időpontban géphez férjen 2-3000 tanuló. A második fordulón azonban mulhatatlanul szükségesnek tartottuk a gépen történő programfejlesztési képességek lemerését. Emiatt — amióta ez a lehetőség megvan — évről évre egy helyen összeszedünk nagyobb számú mikroszámítógépet (mégpedig kizárólag HT—1080, HT—2080 és Commodore 64 típusúakat, mivel a középiskolákban ezek terjedtek el), és a forduló téje adott feladatra program készítése. A gépek számának és a helyeségek férőhelyeinek korlátai természetesen egyben a második fordulóra juttatható versenyzők számának korlátait is jelentik. Tavaly például 50, idén 73 második forduló versenyzetését tudtuk megoldani.

Mielőtt az idei verseny körülményeiről, feladatáról szólnánk, néhány gondolatot felvázolunk a verseny céljairól, feladatairól. Ezek a célok összetettek, és a teljesség igénye nélkül ragadunk ki néhányat, melyet jelenleg már látunk és viszonylag jól meg tudunk fogalmazni.

— Közismert, hogy a számítástechnika iránt a tanulóknak egy nem elhanyagolható része spontán érdeklődik, szinte szenvedélyvel foglalkozik vele. Ennek a belülről jött motivációnak azonban megvannak a maga veszélyei is. Egyebek között például az, hogy a tanuló a számítógép használatának csak bizonyos szűk területeit veszik észre, csak ilyenekben merülnek el, emiatt érdeklődésük sőt készségük is, szakmai értelemben beszűkül. (Például csak játékok készítésére használják a gépet, csak a gépi kód és tárnipuláció trükkjeiben gondolkodnak, és távol maradnak a számítástechnika tényleges gyakorlati hasznosíthatóságának problémaköréitől.) *A verseny egyik feladata*

*bemutatni azokat a követelményeket, irányzatokat, amelyek a korszerű, jövőbe tekintő számítógép-alkalmazások területén mutatkoznak. Természetesen figyelembe véve az életkori és egyéb objektív korlátokat, leszűkítet, leegyszerűsített feladatokról van szó (csakúgy, mint például a fizikai példák esetében, ahol számos olyan zavaró tényezőről eltekintünk, melyeket valóságos számításonál nem szabad elhanyagolni).*

— A számítógép, maga a programnyelv, a gépeken használt felhasználói programok nem egyebek mint eszközök, különféle, rendkívül széles felhasználási területeken. Ez a tanuló részére két fő konzekvenciával jár: egyfelől tudatosulnia kell annak, hogy sem a gép, sem a gépi programozás, sőt a gépi játékok és egyéb hasonló programrendszerek nem tekinthetők céloknak, a velük való foglalkozás *csak ideig-óráig maradhat pusztá szórakozás*. A másik követelmény, hogy ha a tanuló már szerzett bizonyos jártasságot a géphasználatban, valamilyen programnyelv kezelésében, akkor szinte akarva-nemakarva feladatává válik más (nem számítástechnikai) felhasználók részére alkalmazási lehetőségeket teremteni a számítógépen. Ez nem csupán technikai követelmény, hanem egyben nevelési feladatot is jelent: *ki kell alakulnia* az ilyen tanulóknak a felhasználó iránti barátságosságnak, *annak a tulajdonságnak, amely elérébe megy a felhasználó igényeinek, töreksei részére a lehető legösszegebb és legkényelmesebb számítógép-alkalmazási helyzet kialakítására*. Mindez nemcsak a felhasználó problémáinak megértését jelenti, hanem a probléma rendelkezésére, algoritmusának meglátását is.

— Célja továbbá a versenynek a le mérés és vetélkedés, ami számítástechnikai vonatkozásban legalább kétféle szempontból érvényesül: 1. Le kell mérnünk azt a *technikai készséget*, ami a gép, a programnyelv, illetve egyéb *szoftvereszközök kezelésében* nyilvánul meg. Ez természetesen *conditio sine qua non*, de semmiképpen sem a döntő kritérium a mérésnek. 2. Vizsgálat tárgyává kell tenni azt a *tapasztalatot, öltetéseget, amivel a számítástechnikai eszközöket a gyakorlatiasan megfogalmazott alkalmazás, megvalósítás szolgálatába képes állítani*. Hangsúlyozom, hogy ez az alkalmazás való irányultság a legmarkánsabb kritérium a mérésnek, ezt értékeljük legjobban. (Más kérdés, hogy az ehhez megfelelő feladatok kitűzése középiskolások részére korántsem egyszerű dolog.)

— A versenynek — különös módon — van egy olyan célja is, mely a számítástechnikát tanító középiskolai tanárookra vonatkozik. Mi úgy látjuk, hogy bár igen sok, lényegesnek mondható lépés történt a számítástechnikát és alkalmazásait oktató taná-

rok képzése és továbbképzése terén, a mai helyzet e tekintetben még nem megnyugtató. A szakmai, szakdidaktikai területen a részükre nyújtott segítség kevés. Ennek okaival természetesen itt nem foglalkozhatunk, de a verseny lehetőséget ad egyfajta segítségre, amit mi célunknak is tekintünk. A versenyek feladatai mindkét fordulónban tájékoztatják, segítik a tanárokat abban, hogy melyek azok a szaktárgyi problémakörök, melyekre tanulóikat felkészítsék. Milyen színvonalat, milyen mély problémátlátást, milyen bonyolultsági fokot követelhetnek meg a legjobb tanulóktól, milyen módszereket használjanak az ismeretek lemerésében, az egyes készségterületek milyen súllyal értékeljék stb. A verseny értékelése párosuljon egyfajta visszajelzéssel a tanár részére, mely akár egyetértést, akár vitát vált ki, csak haszonnal járhat mind a tanári munka, mind a versenyzetők felfogása terén. Mindez természetesen csak akkor válhat valóra, ha mi magunk is olyan színvonalon és igényességgel tudjuk versenyzetéseinket kitűzni, hogy az elmondott céloknak megfeleljenek. Úgy érezzük, ezen a területen még bőven akad tennivalónk.

Az elmondottakból világos, hogy kerülni szeretnénk az egyes gépekre szűkült ismeretterületeket, az egyes gépek BASIC-jeinek speciális reprezentációira vonatkozó tudás mérését, elismerését. Még inkább kerülni kívánjuk a gépi-kódos és assembly szintű programozási ismeretek használatát, értékelését, beleértve a PEEK és POKE utasításokkal történő tárnipulációkat, melyekről tudjuk, hogy ezekben szinte valamennyi gép alapvetően különbözik.

Az alkalmazás-orientáltság természetesen tartalmazza a részben szoftver, részben hardvertechnikát feltételező berendezésszabályozási, illetve vezérlési területeket is. Mi eddig ilyen nem foglalkoztunk, de gondoljuk, hogy előbb-utóbb aktuálisá válik majd egy ilyen témakörben folyó versenyzetés is, mely a megfelelő feltételrendszerben alakítja ki céljait, módszereit. Már ma is szép számmal akadnak „chip-bütykölő” diákok, akikkel egy ilyen jellegű verseny résztvevői már biztosíthatnak látszanak. Ez a profil azonban nem azonos azzal, amit a „Nemes Tihamér” verseny felvállalt.

Az 1986. évi „Nemes Tihamér” Országos Középiskolai Számítástechnikai Tanulmányi Verseny (ez a hivatalos neve) megrendezésében három intézménynek volt meghatározó szerepe: a Neumann János Számítógéptudományi Társaságnak, a Művelődési Minisztériumnak és a KISZ KB-nek. A feladat nagyjából úgy oszlott meg, hogy a gazdasági alapokat és az iskolák eljárásának szabályozását az MM biztosította, a számítástechnikai szakmai munkát az

- a dolgozatok 34,7%-ánál saját értékelésünk egyezett a tanári értékeléssel,
- a dolgozatok 35,2%-ánál említett a tanár által megítélt pontszámot, átlagosan 2,55 pontértékeléssel (a legnagyobb emelés 10 pont volt),
- a dolgozatok 30,1%-ánál csökkentettük a tanár által megállapított pontszámot, átlag 2,37 pontértékeléssel (a legnagyobb csökkenés 7 pont volt).

Az újravizitott dolgozatok általunk megállapított pontszámát tekintettük mérvadónak. Ebből következett, hogy a tanulók pontszám szerinti rangsorolása is kissé más lett, mintha azt az eredeti tanári pontozás alapján készítettük volna el. A pontszám szerinti rangsorolási listát, kizárólag a tanuló nevével, iskola megjelölése nélkül, egy célna készített számítógépes program állította elő.

Mindezek után kerülhetett sor arra, hogy megállapítsuk, kik juthatnak a döntőbe. Ehhez természetesen ismernünk kellett azokat a korlátokat, melyek idén megszabták a második fordulóban részt vevők számának felső határát. Elsősorban a Budapesti Műszaki Egyetem segítőbizottságának köszönhetően, mintegy 40 db HT és 15 db C64 állt a versenyzők rendelkezésére, három e célra alkalmasan berendezett teremben, mindezt lehetővé téve a programlisták kinyomtatását is. A döntőbe jutottak választhattak (még a döntő időpontját megelőzően, postai értesítés alapján), hogy a két géptípus közül melyikben kívánunk versenyezni. A termék úgy álltunk rendelkezésünkre, hogy két turnusban bonyolíthattuk le a döntőt: turnusonként kb. 36 tanulóval. Ennek előnye volt, hogy ekkora létszám (végtül is 73 tanuló) eddig még soha sem juthatott döntőbe hazánkban. Hátrány volt viszont, hogy az időbeli megkötöttség miatt egy turnusnak maximum két és fél óra áll rendelkezésére feladata elkészítéséhez.

Miután a pontszám szerinti rangsorolásban pontosan 72 olyan tanuló volt, aki 59 vagy annál magasabb összpontszámot ért el, őket soroltuk a döntőbe. De azért nem jártunk el mechanikusan, mert valamennyi 58 és 59 pontos dolgozatot újra elővett a bizottság, és kollektíve ellenőrizte, újraértékelte a második javítást is. Ennek során egyetleneget tanuló volt, akit 58 pontról 59 pontra felértékelünk. Így jutott végül is 73 tanuló a döntőbe. Közülük 34 volt budapesti, és egyáltalán nem volt közbüti lány. A turnusokat úgy állítottuk össze, hogy a 8 órákos kezdő csoportba a budapestieket, illetve a Budapest környékieket soroltuk, és a 11-kor kezdő csoportba kerültek a vidékiek.

Természetesen már jóval a döntő előtt megkezdődött a kiadandó feladat kidolgozása, megfogalmazása. A szakmai előkészítés fázisai közül ezt tartjuk a legnehezebbnek, legigényesebbnek. Mert túlmenően azon az előzőekben már kifejtett szemponton, hogy ti, egy a gyakorlatból vett feladat kerüljön ide, számos egyéb szempont, korlátra is ügyelnünk kellett. Mindezek előtt arra, hogy ez a gyakorlati feladat ne követeljen meg olyan elméleti háttérismertelmet, melyet nem várhatunk el. Legyen a feladat olyan, hogy több, jól elkülöníthető részfeladatot és ezzel együtt jellemző jelje-

lag. Az értékelés megbízhatósága elengedhetetlenül tette a második javítást. Hangsúlyozni szeretném azonban, hogy ez kizárólag az értékelés objektivitásának javítását szolgálta és fogja szolgálni; mi senkit sem minősítettünk javítási hibái miatt, még kevésbé adunk ki ilyen információkat bárkiről. Az igazi az lenne — ezt mi is világosan látjuk —, ha a javítási differenciákról a tanárok részére visszajelzést adnánk, természetesen indoklással együtt. Annak ma egy nagyon egyszerű, de igen nyomós oka van, hogy nincs így: nem győzzük az adminisztrációt, sem idővel, sem energiával. Ha a jövőben jobban sikerül megszervezni a verseny adminisztrációját (aminek ilyen objektív feltételei is vannak), akkor erre is sor fog kerülni.

Következő szakmai megbeszélésünkön a benyújtott választékból — egyes feladatokat még újra módosítva — meghatároztuk az első forduló végleges feladatsorát. Ezt követte a feladatsor értékelési szempontjainak, a feladatok, illetve részfeladatok súlyozásának, a pontszámozásnak kialakítása. A maximális pontszám az első fordulóban idén 66 volt. Az iskolából történő továbbküldés feltétele az lett, hogy a tanuló elérje az összpontszám egyharmadát, és ezen belül legyen legalább egy hibátlan megoldása a nem szakfíjezését értelmező kérdések közül. Az elkészült feladatsort és javítási útmutatót mágnés adathordozóra rögzítettük, onnan nyomtatva küldtük szét a bizottság tagjainak további javításra, stílusjavításra. A kifejezetten gondos munka ellenére két komolyabb hiba került az anyagba, egyik a feladatkiütésébe, a másik a tanári javítási útmutatóba. Az e hibákból eredő tanulói, illetve tanári értékelési hibákat természetesen megfelelő módon figyelembe vettük az újraértékelésnél. Az első forduló feladatsorát itt nem közöljük, mivel ezt minden versenyző iskola megkapta. Amikor elkészült ez az anyag, úgy éreztük, meglehetősen nehéz feladatsort állítottunk össze, tartottunk attól, hogy a maximális munkahibájára estünk. Mint később kiderült, ebben bizony tévedtünk.

Az első fordulóban összesen kb. 1750 tanuló vett részt (becsült adat, mert kérdésünk ellenére az iskolák egy része nem jelezte vissza, hogy hány nem továbbított dolgozat készült el), ami mintegy 900 tanulóval kevesebb a tavalyi összlétszámnál. A versenyben 273 iskola vett részt, melyekből 47 volt budapesti.

A továbbküldött dolgozatok száma (tehát azoké, akiknek pontszámja a felső kétharmada esett) 1031. Ezeket 257 iskolából (ezen belül 45 budapesti) küldték el. Külön felhívom a figyelmet a következő adatra: az 1031 tanuló közül mindössze 26 volt lány. A beküldött tanulói munkák adminisztratív beiktatása során láttuk, hogy azok az általunk vártnál lényegesen jobban sikerültek, igen sok a nagyon magas pontszámot elért tanuló (a tanári értékelést figyelembe véve). Ezért úgy határoztunk, hogy a bírálóbizottság azokat a dolgozatokat javítja újra, amelyek a tanári értékelés alapján a pontszámokat tekintve a felső egyharmadba esnek, azaz 45 vagy annál magasabb az összpontszámuk.

A beküldött dolgozatokból 426 tanuló jutott a felső egyharmadba, közülük 168 budapesti és a 426-ból két lány tanuló

NJSZT végezte, az anyagok terjesztésében és az adminisztráció jelentős részében pedig a KISZ KB segítségével élvezhettük.

A szakmai munkát egy ún. bírálóbizottság végezte, melynek közvetlen irányításával az NJSZT Oktatási Szakszolgálata volt megbízva. Mindazonáltal a bírálóbizottság meglehetősen nagy szakmai munkája mellett sem mentesülhetett egy sor szervezési, adminisztrációs feladattól. Ezen felül idén a verseny egész szervezése késve indult az 1985 végén lezajlott társulati választások miatt, de egyes elmaradásokban magunk is ludasak voltunk, így a határidők betartása olykor rendkívüli erőfeszítéseket kívánt meg. Az elmondottak, meg nyitottságunk okán felsorolom a bírálóbizottság tagjait: Deák-Jahn Gábor BME, Hanák Péter BME, Hunyady István BME, Köhégi János ELTE, Mohai Tamás BME, Somogyi László Ped. Szaksz., Székely Jenő Budapesti Tanárképző Főiskola, Tagányi György BME, Zámbo Viktor BME és Zsakó László ELTE. A bizottság vezetője e sorok írója volt. Mint látható, a bizottságba olyan, a számítástechnikai szakmához és a középiskolai oktatáshoz, sőt a fiatalokhoz közel álló személyeket válogattunk össze, akik saját munkájukban, érdekeikben összeférnek a versenyzetektől elvárt pártatlansággal. Ez utóbbi az oka, hogy aktív középiskolai számítástechnikai tanár nem került a bizottságba, hiszen egyébként nagyon sok olyat ismerünk közülük, aki felette érdemes lenne erre a feladatra.

A szakmai munka első fázisa egy olyan anyag elkészítése, amely nagy vonalakban vázolja a verseny céljait, továbbá azokat az alkalmazási területeket (leszűkítésekkel, korlátokkal együtt), amelyekben a tanulókötő bizonyos felkészültséget kívánunk meg. Erre már ez évben is kísérletet tettünk, szeptemberben egy ilyen tárgyú anyagot. Úgy gondoljuk azonban, hogy ez valószínűleg még több lépésben javítandó, finomítandó, sőt egyes részeiben módosítandó lesz.

Az első forduló feladatainak kitűzése céljából a bizottság tagjaitól feladatjavaslátokat kértünk. A benyújtott írásos anyagokat sokszorosítottuk, és a teljes szeriát minden tagnak újra szétküldtük. Természetesen mielőtt ezek az anyagok elkészültek volna, egyeztetés, elvi szempontokat kialakító megbeszélést tartottunk, melyben tisztáztuk, hogy mit is kívánunk meg ettől a teszt jellegű versenyfokozattól.

A követelmények részben szakmaiak voltak, mint például: ne legyenek a feladatok gépre, BASIC-reprezentációra irányultak (amennyire ez lehetséges), de ha már mégis elkerülhetetlen, akkor egyaránt legyenek értelmezhető HT és C64 típusokon. Legyenek a kérdések gondolkodtatóak, vagy legalábbis adjanak fel értelmezési problémákat általánosan használt terminusokra nézve.

A követelmények másik része technikai volt: legyen a teszt viszonylag gyorsan javítható, a javítás könnyen áttekinthető. Ennek fő oka, hogy korábbi szokásainkhoz híven, nemcsak a tanárok, hanem mi magunk is kijavítottuk minden dolgozatot. Úgy gondoljuk, erről is nyíltan kell szólnunk. Mind a korábbi, mind az ideai tapasztalatok azt bizonyították, hogy a tanári javításokban meglepően nagy százalékban fordulnak elő hibák, nagyobb részt elnézésekből kifolyó-



imémszinteket tartalmazza, hogy az értékelés egyértelmű legyen. Igénytelen a feladat bizonyos rendszerszemléletet, adjon lehetőséget logikai egységek megállítására és ezzel együtt egy modul szerkezetű program elkészítésére, illetve az ennek megfelelő programlista-dokumentálásra.

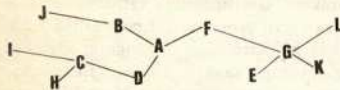
Mindebből nyilvánvalóan következik, hogy egy nagyobb méretű feladatot kell kitűzniük (és nem több kisebbet), de ez azért mégse legyen olyan nagy, hogy az adott időkeretben ne lehessen jól elkülönülő színvonalak elérése alapján értékelni. Úgy képzeltük, miután az első forduló feladatainak nehézségi fokát túbecsültük, hogy ebben a második fordulóban az egymást követő részfeladatok közül az utolsónak valóban nehezebb kell lennie. Viszonylag hosszú kollektív bizottsági munka eredményeként készült el egy olyan feladat, melynek első része adatrendezésekből és listázásokból áll, második része egy fa-gráf útvalon történő (nem optimalizáló) szállítási feladat, harmadik része egy csak néhány hurkot tartalmazó hálón történő (ugyan-csak nem optimalizációs) feladat. A kitűzött feladat a következő volt:

**Feladat:** Tervezz és írj programot a Neumann János Számítógéptudományi Társaság tagjainak nyilvántartására, amely a számítástechnikai szakfolyóiratok szétkülésére és a szállítási útvalonak meghatározására szolgál.

#### Tárolandó adatok:

- név (max. 25 betű, ékezetes betűk nem szerepelnek)
- város (számuk max. 16, jelük egy betű az (A, P) tartományban)
- kért újság (vagy a Számítástechnika vagy a Mikroszámítógép Magazin rendelhető)

Az újságokat vasúton szállítják. A vasúti összeköttetéseket a felhasználó adja meg. A hálózat városokból, mint csomópontokból és ezeket összekötő vonalszakaszokból áll. A hálózat megadásának és tárolásának módját te határozd meg. Egy példa a lehetséges összeköttetésekre:



#### Megoldandó részfeladatok:

- A) A tagok adatainak nyilvántartása:
1. Az adatbázis létrehozása legalább 30 tagra
  2. Új tag felvétele
  3. Tag törlése
  4. Tag adatainak módosítása
  5. A tagok névsor szerinti listája minden hozzá tartozó adattal
  6. A tagok listája újságoként külön-külön, ezen belül városok szerint, abc-sz. sorrendben
- B) Szállítással kapcsolatos feladatok:
1. A vasúti hálózat modelljének létrehozása a hozzá tartozó kezelőfunkciókkal. Ezek a funkciók a választott beviteli módtól függenek, így ezeket is neked kell meghatároznod.
  2. Újságoként külön-külön lista készítése arról, hogy melyik városba mennyi újságot kell küldeni, és ebből hányat kell továbbítani a következő állomásokra. Feltevétezzük, hogy mindkét újságot ugyanott, az „A” városban nyomtatják, és hogy „A” városból minden más városba el lehet jutni egy és csakis egy útvalonon.

3. A feladat ugyanaz, mint az előző pontban, de a két újságot bármelyik két különböző városban nyomtathatják, és néhány város több úton is elérhető.

#### Kiegészítések:

- A listákat képernyőre kell írni.
- Az adatbázisok háttértárolását nem kell megoldani.
- A program a lehető legjobban legyen dokumentálva, feleljen meg az általános formai követelményeknek, és legyen barátságos a felhasználóval.
- A program első sora egy REM sor legyen, és tartalmazza a következő adatokat: név, iskola, helység.
- A programodat két példányban rögzítsd.
- Beadandó a kazetta vagy lemez és minden, a könnyebb megértést segítő dokumentáció, vázlat.

A feladat kitűzésekor az alábbi elképzeléseink voltak:

- az A1–A6 részfeladatok rutin jellegűek, ezeket mindenki meg fogja oldani.
- Igazi próbatételt, ami a mezőny szét-húzza, a B1 és B2 feladatok fognak jelenteni. Előnyük lesz azoknak, akik valamit ol-vastak már gráfokról, de azok is meg fog-ják tudni oldani, akik most találkoznak elő-szór a témával.
- A B3 feladatot csak a legjobbak lesz-nek képesek megkezdni, befejezésére nem számíthatunk.
- Háttértárolás a géptípusok erős kül-önbözősége miatt nem kívánható meg.
- A dokumentáltságon elsősorban a magyarázó, illetve a programot tagoló REM sorok használatát értjük, és természetesen nem kérünk valamiféle kezelési le-írást melléklektünk.
- A javításhoz egy meglehetősen részletes értékelő pontozási rendszert készítettünk, melyben sokféle válaszlehetőséggel számol-tunk. Pontozási rendszerünk részfeladaton-ként és bizonyos minőségi tulajdonságok-ént az alábbiak szerint alakult ki:

- A1-re maximum 2 pont
- A2-re maximum 4 pont
- A3-ra maximum 4 pont
- A4-re maximum 5 pont
- A5-re maximum 7 pont
- A6-ra maximum 5 pont
- B1-re maximum 5 pont
- B2-re maximum 15 pont
- B3-ra maximum 15 pont
- Programszerkezet maximum 5 pont
- Barátságosság maximum 5 pont
- Adat- és programbiztonság maximum 5 pont

A program szerkezetén a program modu-láltsága, tagoltsága, a programlista olvashatósága, áttekinthetősége értendő. A barátságosság a felhasználóval való kapcsolat minőségét jelenti, a menük kezelhetőségét, az egyes részfeladatokból történő továbblépés képernyőn való dokumentálását, a program befejezhetőségét stb. A biztonság részben az adatokra vonatkozik, amennyiben védi a programot a feladat kitűzése szempontjából abszurd adatok bevitelétől, illetve védi a programot olyan beavatko-zásoktól, amelyekre az gépi hiba jelzéssel megállhatna.

Mint utólag rájöttünk, ezzel a feladattal viszont kissé túllőttünk a célon, aminek azonban részben az is oka volt, hogy a fel-

adat méretéhez mérten a rendelkezésre álló idő rövid volt. A döntő oknak mégsem ezt tartjuk, hanem azt, hogy szó sem volt arról, hogy az A1-től A6-ig terjedő feladatok rutin jellegűek lettek volna, ellenkezőleg, a többségnek már ez is gondot okozott. A B3-ba senki bele sem fogott, sőt a B2-t sem fejezte be senki. A legjobbak azonban — kb. a döntőbe jutottak 25%-a — valami-lyen formában tervet készítettek a B2-re, és a legjobb 10 ennek egy részét programban realizálták. Itt tehát az időfaktor nagyobb súllyal esett latba. Ha figyelembe vesszük hogy a feladatok összpontszámából a B3 pontjait minden további nélkül elhagyhat-juk, akkor 62 maximális pontszámmal szá-molva (és a feladat túlméretezettsége miatt ezt kell realísának tartanunk) az első tíz helyezett összpontszámáról és relatív készen-léti fokairól azt mondhatjuk, hogy azok 41-től 36-ig terjedtek, azaz 66%-tól 58%-ig értékelt minőséggel sikerültek. Hozzá kell azonban mindehhez tenni, hogy még a leg-jobbak pontszámaiból is hiányzott az utol-só három minőségi értékes szám — nem túl nagy ráfordítással elérhető — pontszáma.

Az első tíz helyezett:

1. Erdei Zsolt  
Árpád Gimnázium, Budapest
2. Huszár Péter  
Széchenyi Gimnázium, Sopron
3. Paller Gábor  
Teleki Blanka Gimnázium, Buda-pest
4. Mazán Zsolt  
Szilárd Áron Gimnázium, Kiskun-halás
5. Makay Géza  
JATE Ságvári Gyakorló Gimnázium, Szeged
6. Bélyky Zsolt  
Herman Ottó Gimnázium, Miskolc
7. Jánossy Zoltán  
Lovassy L. Gimnázium, Veszprém
8. Novák István  
Berze—Nagy Gimnázium, Gyön-gyös
9. Mamrovics László  
Árpád Gimnázium, Budapest
10. Pallagi László  
Vörösmarty Gimnázium, Érd

A felsoroltakon kívül két NJSZT különdíjat és három KISZ különdíjat osztottunk ki. A különdíjasok közül kiemelem Habony Zsolt (Földes Gimnázium, Miskolc) munkáját, aki a B2 feladat megoldásában egyedül alkalmazott igen szellemesen egy rekurzív módszert.

Nem kívánunk foglalkozni itt a verseny szervezésének, adminisztrálásának problémáival, noha bőven volna erről is mit írni. Pusztán annyit, hogy akik neheztelnek ránk az ebbeli fogyatékoságokért, azok joggal éreznek így, és tölük ezúton is elnézést ké-rünk. Egyetlen mentségünk, hogy igen fá-rasztó ráfordítások ellenére sem értünk el e területen áttűtő sikereket. Többek között ebben is van még mit javítani munkánkon.

A verseny szakmai, szervezési tapasztalatai igen sok, most még végig sem gondolt konzekvenciával szolgáltak. Kettőtől azonban mégis említést kell tennem.

— A tanulók iskolai és autodidakták számítástechnikai felkészülése ma ott tart, hogy a legjobbak meglepően magas prog-

ramutatisás-haszánlati, programozásté-  
 nológiai színvonalra jutottak el. Nem keve-  
 seknél határozottan felismerni szakköny-  
 vek ismeretét, egyfajta olvasottságot. Mind-  
 ennek az az eredménye, hogy jól ismerik a  
 számítástechnika szakkifejezéseit, értelme-  
 zésüket is beleértve, magas szinten kezelik  
 azokat az építőelemeket, melyekkel egy  
 professzionálisnak minősíthető programo-  
 zó is dolgozik. Ugyanakkor azonban ezek  
 egymástól elszigetelt, alig-alig összekapcsolo-  
 dó ismereteket jelentenek akkor, ha a fel-  
 adatban a probléma lényegét kell meglátni,  
 ha tudatosan végig kell járni az algoritmus-  
 keresés útját. A tanulók jelentős része  
 előbb kezd programot írni, és csak aztán  
 — amikor a megírt program valamilyen  
 szempontból hibásnak bizonyul — kezd a  
 rendszeren gondolkodni. Maga a rendszer-  
 átlátás is gyenge oldaluk. Könnyű lenne itt  
 ítéletet mondani, de ez a szemlélet sem ta-  
 nulóban, sem tanárban nem alakulhat ki  
 pusztán attól, hogy sokat gyakorol a gépen,  
 és megtanulja a BASIC nyelv használatát.  
 Ma hazánkban a legtöbb olyan számítá-  
 stechnikai szakember, aki ismeri az alkalmaz-  
 ás-orientált programkészítés szakterületét  
 a gyakorlatból, saját szakmai tapasztalatain  
 keresztül fejlődött az a, ami. Olyan szak-  
 könyv, amely középiskolás szinten a fő  
 eljárások, módszerek áttekintésével juttatná  
 el használatját egyfajta „általános felhasználó  
 programkészítési kultúra”-hoz, ma még  
 nemigen akad. A célt mégis fontosnak,  
 az irányában való haladást a szakmai képzés  
 meglehetősen széles körben elkerülhet-  
 lennek tartjuk. A szellemi tartalék ma a  
 problémák mögött rejlő rendszerek átlátá-  
 sában és a hozzá kapcsolódó alkalmazási  
 algoritmusok megtalálásában, felfedezésé-  
 ben rejlik. Erre kell nevelnünk.

— Végül pedig törleszttem adósságotam  
 a lány tanulók számítástechnikai motiválá-  
 sának problémájáról kialakult véleményem  
 vázolásával. A félreértések elkerülése érde-  
 kében előrebocsátom, hogy nem elsősor-  
 ban a női egyenjogúságról kialakult nézete-  
 im miatt tartom ezt a helyzetet súlyos baj-  
 nak. Hanem azért, mert teljesen nyilvánva-  
 ló, hogy az adminisztráció, banki, kereske-  
 delmi stb. munkák területén néhány éven  
 belül nem lesz olyan munkakör, amelyben  
 ne kellene számítógépet kezelni, használni.  
 Persze az ehhez szükséges ismeretek sem  
 lesznek azonos színvonalúak, hiszen példáu-  
 l egy szövegkezelő rendszer használatát  
 jól meg lehet tanulni a BASIC programo-  
 zásban való teljes tudatlansággal is. De  
 mekkora előnyben lesznek azok az admini-  
 strátorok, pénzügyi, kereskedelmi, ál-  
 lamgazgatási előadók, akik valamit tudnak  
 is arról, amit nap mint nap használnak. Az  
 élet sorozatban produkál olyan váratlan  
 igényeket, melyekre a pusztán billentyű-  
 használatra megtanított ügyviteli alkalmaz-  
 zottak képtelenek lesznek reagálni. Az al-  
 alkalmazkodás csak azoknak lesz birtoká-  
 ban, akik egy kissé belelátnak abba a rend-  
 szerbe, amit használnak. Mire lesznek ké-  
 pesek a számítástechnikai alkalmazások  
 iránt közömbös, felkészületlen nők ebben  
 a követelményarázatban? Mire lesz képes az  
 iroda, hivatal, vállalat a megfelelően felkés-  
 zült (és motivált) ügyviteli stb. dolgozók  
 nélkül? Itt látom a veszélyt.

Befejezésül köszönjük a lelkes tanári  
 munkát, a segítőkész iskolavezetési tevé-  
 kenységet, ami nélkül ezek az eredmények  
 nem jöhettek volna létre.

DR. ADA-WINTER PÉTER  
 és a bírálóbizottsági munkaközösség

## HT—1080Z A lemezrendszer utasításai és függvényei

Aki már látta a lemezrendszer utasítás-  
 készletét, az tudja, hogy ezeket az utasítá-  
 sokat akkor sem fogadja el a gép, ha a ru-  
 tincímét átírjuk. Például ha a CVI szót pa-  
 rancsként gépeljük be — azzal az álnok céllal,  
 hogy így lépjen a gép monitormódba,  
 hiszen a kezdőcímet átírtuk —, akkor ép-  
 pen úgy ?SN ERROR-t kapunk, mintha át  
 sem írtuk volna.

Próbáljuk meg azonban az A=CVI utasítást  
 begépelni! Ha mindent ügyesen gé-  
 peltünk, akkor a gép monitor módba ke-  
 rült. A magyarázat, gondolom természetes.  
 A CVI nem utasítás, hanem függvény. Te-  
 hát alkalmazására ugyanazok a „törvé-  
 nyek” vonatkoznak, amelyek például a  
 SIN, COS, VARPTR stb. függvényekre.  
 Törvényen azt az átfogó szintaktikai sza-  
 bályt értem, hogy függvény sor elején nem  
 állhat; vagy az egyenlőségjel jobb oldalán,  
 vagy egy print utasítás után, mint kírando  
 elem kell hogy legyen.

Ezeknek az utasításoknak a típusát tar-  
 talmazza az alábbi táblázat.

Utasítás	A rutin kezdőcíme a RAM-ban	Típus
CVI	4152	C
FN	4155	B
DEF	4158	C
CVD	415B	A
EOF	415E	C
LOC	4161	C
LOF	4164	C
MKIS	4167	C
MKSS	416A	C
MKDS	416D	C
CMD	4173	A
TIMES	4176	B
OPEN	4179	A
FIELD	417C	A
GET	417F	A
PUT	4182	A
CLOSE	4185	A
LOAD	4188	A
MERGE	418B	A
NAME	418E	A
KILL	4191	A
&	4194	B
LSET	4197	A
RSET	419A	A
INSTR	419D	B
SAVE	41A8	A
LINE	41A3	C
USR	41A9	C

Jelmagyarázat A: parancs  
 B: függvény  
 C: kiértékelő függvény

A Madách Gimnázium  
 klubja



## LEMEZOLDAL- JELÖLÉS

A 1541-es meghajtóval rendelkező fel-  
 hasznalók általában lemezeiket „kivágják”,  
 és a második oldalt is felhasználják pro-  
 gramtárolásra.

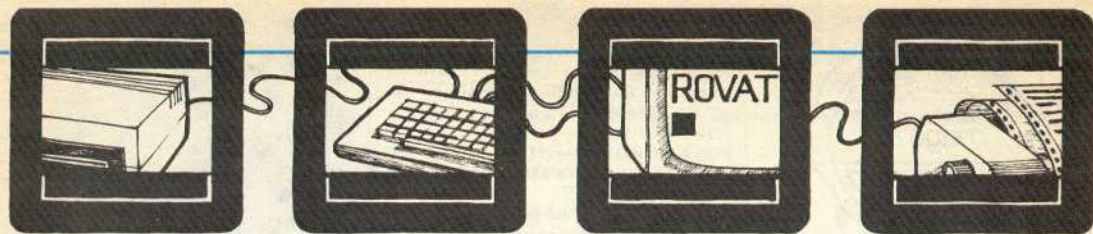
Hogy melyik a második oldal? Ez a szá-  
 mitógépeknek teljesen mindegy. Nekünk  
 viszont tudni kell egy kívülről ráragasztott  
 címkéről vagy a lemez más nevéből, ID-jé-  
 ből. Az alábbi program felhasználásával  
 mód nyílik egy olyan jelölésre is, amelynél  
 a lemeznev és a lemez ID is ugyanaz ma-  
 radhat mindkét oldalon.

Az ID után található egy 2A, a formatá-  
 lás típusára utaló jelölés. Ennek B-re vagy  
 A-ra (teljesen mindegy, lényeg, hogy két  
 karaktert tartalmazzon) változtatása a le-  
 mezen lévő programokat nem érinti. Ez az  
 átírás hajtható végre a program segítségével.  
 Nekünk csak a jelölést kell beírni, s  
 ahogy a lemezegységen a piros LED elaz-  
 szik, a módosítás kész is. Az új tartalom-  
 jegyzék ellenőrzése előtt (LOAD”S”) 8) kap-  
 csoljuk ki a gépet, ellenkező esetben nem  
 érzékelni a változtatást.

A programot futtathatjuk akár olyan le-  
 mezen is, ami már foglalt, de vigyázni kell  
 a pontos beírásra.

BÁRTFAI BARNABÁS  
 Ságvári Endre Szakközépiskola  
 Székesfehérvár

Ø REM ATOMSOFT B.B.  
 1Ø OPEN 1,8,2,””:OPEN 15,8,15  
 2Ø S=18:SZ=Ø  
 3Ø P=165  
 4Ø INPUT”OLDALJEL”:M\$  
 5Ø PRINT”15,”B—R.”:2:Ø,S;SZ  
 6Ø PRINT”15,”B—P.”:2:P  
 7Ø PRINT”1,M\$;  
 8Ø PRINT”15,”B—W.”:2:Ø,S;SZ  
 9Ø CLOSE 1 : CLOSE 15



## Amit nem tud a HT-val...

... azt mással kell megpróbálni. Természetesen a VAL-függvényről van szó. Az iskolaszámítógép kedvelői bizonyára irigykedve gondolnak a Spectrum, a ZX81 vagy az Aircomp (Homelab) remek VAL-függvényére. Ezek a gépek ugyanis képesek kiszámolni egy sztringben megadott kifejezés értékét, ami persze tetszőlegesen bonyolult lehet. Ha például az Aircompba beírjuk a PRINT VAL ("1+2") parancsot, akkor 3-at kapunk eredményül. Ha viszont a HT-nél kísérletezünk hasonlóval, a kiírt szám az 1 lesz, hiszen a gép csak a számokat értelmezi.

Az 1. listán látható gépi kódú program ezt a hiányosságot orvosolja. Mivel magát a VAL-t kissé bonyolult lenne átírni, ezért a nem használt CVI-függvénynek adunk jelentést. A program a következő feladatokat látja el:

1. Átvizsgálja a sztringet a 7F00H címre.
2. Tokenizálja/kódolja a műveleteket.
3. Kiszámolja és visszatér az aktuális értékekkel.
4. Helyet foglal és átírja a CVI ugrótábláját.

Bár a program elért volna az ugrótábla helyén is, egy ilyen kulcsszó használata „végzetes” lehetne. EDI-vel lefordítva azonnal lehet használni a megújult VAL (CVI) függvényt. Írjuk tehát be a gépbe: PRINT VAL ("2\*3"), CVI ("2\*3"). Az eredmény 2 és 6 lesz. A különbség ugye óriási!

Hogy mire jó ez? Nos, vegyünk egy függvényábrázoló programot (2. lista). Eddig a legjobb megoldást az önmagát változtató program kínálja. A dolgunk most lényegesen egyszerűbb. A függvényt, esetleg többet is, sztringként lehet tárolni, és értékét a változóktól függetlenül bármikor kiszámíthatjuk. Vigyázzunk azonban, mert számolás közben ugyanúgy felléphetnek a különböző hibák (SN, FC, /θ stb. ERROR), tehát a BASIC programba be kell építeni a hibakezelést is. Ez az egyszerű ábrázoló a CVI lehetőségét és megkötezeit mutatja be, a bekért függvényt ábrázolja fél képernyőn. A CVI egyébként alig valamivel lassúbb egy direkt értékadásnál. A gépi kód betöltését és a helyfoglalást a 13θ-tól kezdődő rutin végzi, ezt kell tehát használni más programokban is.

PALLAGI LÁSZLÓ

Érd, Vörösmarty Mihály Gimnázium

```

1 ORG 7ED0H
2 LOAD 7ED0H
3 CALL 2A07H
4 JP Z,27F8H
5 LD E,A
6 INC HL
7 LD C,(HL)
8 INC HL
9 LD B,(HL)
10 EX DE,HL
11 INC HL
12 LD DE,7F00H
13 PUSH DE
14 PUSH DE
15 PUSH DE
16 CALL 29CFH
17 EX DE,HL
18 LD (HL),θ
19 POP DE
20 POP HL
21 CALL 1BCCH
22 POP HL
23 CALL 2337H
24 RET
25 ORG 4152H
26 LOAD 4152H
27 JP 7ED0H
28 ORG 40B1H
29 LOAD 40B1H
30 DW 7ECFh
31 END

```

1. lista

2. lista

```

10 GOTO130
20 CLS
30 PRINTθ, "F(X)=", INPUT#
40 CLS
50 OVERPROGRAMGOTO100
60 FORX=30T030
70 SET(X+31,23-CVI(AR))
80 NEXTX
90 GOTO30
100 E=ERR
110 IFEOSANDθ-19AHDθ-20THENRESUME20
120 RESUMEEXIT
130 POKE16561,200:POKE16562,126
140 CLEAR1000
150 POKE16723,209:POKE16724,126
160 FORK=3264T03249θ
170 REPLY:POKEK,Y
180 NEXTK:GOTO20
190 DATA205,7,42,202,248,39,95,35
200 DATA78,35,78,235,35,17,θ,127,213
210 DATA13,213,205,207,41,235,54,θ
220 DATA209,225,205,204,27,225,205
230 DATA55,35,201

```

## A Spectrum rendszerváltói

A Spectrum rendszerváltók megtalálhatók ugyan a gépkönyvben, de a kétsoros magyarázatokból általában nem lehet tökéletesen kikövetkeztetni a változó szerepét. Ezen a problémán igyekszem segíteni a teljesség igénye nélkül, mivel egyes változók csak a rendszer számára értékesek, a programozó szempontjából nincsen gyakorlati hasznuk. Ezeket az interpreter használja, megváltoztatásuk a rendszer összeomlást idézi elő.

Az általános kényeljm szempontokat kiszolgáló rendszerváltókat ismertetem. A táblázatban a cím hexadecimális formátumban is szerepel, mivel így gépi kódban könnyebben címezhető.

HEX	DEC	NÉV	FUNKCIÓ
5C08	23560	LAST-K	A legutóljára lenyomott billentyű kódját tartalmazza ASCII kódban. Kiválassza első a FLAGS változó 5. bitjét kell ellenőrizni. Ha ez 1, akkor a LAST-K-ban érvényes új kód van. Kiválassza után a FLAGS 5. bitjét θ-ba kell állítani.
5C09	23651	REPDEL	A változó azt az időt tartalmazza, ameddig nyomva kell tartani egy billentyűt ahhoz, hogy az ismétlési funkció (repeat) megkezdődjön. Ez az idő 1/30 másodpercben értendő. Kezdeti értéke 35. Gyakorlatiabb gépeltékek érdeme kisebbre változtatni, így gyorsabb lesz az adatszevitel.
5C0A	23562	REPPER	Azt az időt tartalmazza, ami az ismétlési funkcióban két billentyűütemítés között eltelik. Kezdeti értéke 5. Megváltoztatásával valószínűleg gyorsabb adatszevitelt lehet elérni.
5C36	23086	CHARS	A karaktergenerátor kezdőcíme tartalmazza. Új karakterkészlet kialakításához ez a cím a RAM-ba kell átirni, és innentől lehet az új karaktereket definiálni. Ezek a karakterkészletnek csak egy része, a 32 (space) és a 128 (θ) közöttiek.
5C38	23688	RASP	A figyelmeztető zümmögés magassága, amit túl hosszú sor végén ad a gép. Átirásával a jelenlegi kellemelebb hangot is el lehet írni.
5C39	23689	PIP	A billentyű lenyomásakor megszólaló hang magassága. Szintén kellemelebbé lehet tenni nagyobb értékek beállításával.

RUSZNYÁK GÁBOR



## HIBAIGAZÍTÁS

Lapunk 86/4. számában sajnálatos módon a 34. oldalon a Nagyító program ismertetése után egy ZX81-re írt nagy felbontású grafika program listája jelent meg. Most közreadjuk a program leírását.

A 35. oldalon levő Matematika program leírása mellett a 34. oldalon ismertetett Nagyító program listája található. A Matematika program listáját most közreadjuk.

## Nagy felbontású grafika

A program (listáját lásd a 86/4. szám 34. oldalán) csak közelítő pontosságú ábrát ad, ez azt jelenti, hogy egy bájtban a 256 vertikálisból csak valamivel kevesebb, mint a fele jelenhet meg. A program 4 sorral nagyobb képernyőt használ a szokásosnál, azaz 28 sorosat. A felső 4 sor mindig az eredeti kis felbontású displayfájl első 4 sora. A legalsó 4 sor pedig mindig nagy felbontású. E két érték között szabadon állítható be a két képernyő határa. A nagy felbontású displayfájl 256 x 192 pixel.

A programot a 4084 hexa címtől kezdve fordítottuk be egy REM sorba. Ezután írjuk be a mellékelt BASIC programot, mely a változókat definálja. Aztán mentjük ki a programot magnóra. Visszatöltés előtt vegyük le a RAMTOP-ot 6680H-ra és NEW-val töröljük a memóriát. Legelőször a CLS utasítást használjuk. Ez hozza létre a nagy displayfájl. Az utasítás jellegű változókat USR-rel az adatközpontok POKE-kal aktiváljuk.

Például: RAND UST CLS  
POKE CORDX,15

A DISPLAYB-val térhetünk át a nagy képernyőre. Ekkor a felső 4 sor BASIC-ből kezelhető. A többi nagy felbontású. Ettől kezdve a gép a hibajelzést felmásolja a képernyőre mindig látható tetéjére. Ez nagyban könnyíti a programozást.

A képernyő arányait a DISPLAYC-val változtathatjuk. Ez a CORDY adatot használja.

POKE CORDY, (4—24)  
RAND USR DISPLAYC

Visszatérni az eredeti képernyőhöz DISPLAYA-val lehet. Gyakran előfordul

ZX81-en, hogy hibás POKE-olás miatt szétesik a displayfájl. Ha még el a BASIC, akkor ez a DISPLAYD rutinnal kijavítható a képernyő törlése nélkül. A PRINT rutinnal egy bájtot vihetünk a képernyőre.

POKE CORDX, (0—31)

POKE CORDY, (0—191)

POKE CODEA, a karakter kódja

RAND USR PRINT

A PLOT rutin közelítő pontossággal rajzol a képernyőre. Szintén a CORDX—Y-t használja.

DRAW-val vonalat húzhatunk. Változói CORDX—Y,

DRAWX—Y a vektor koordinátái.

Található még a programban scroll mind a négy irányban:

UP, DOWN, RIGHT, LEFT.

Gyors QSAVE és QLOAD, mely a nagy felbontású képernyőt kezeli.

INVERT utasítás és EXX rutin, amely minden A kódú karaktert B kódúra cserél.

Formája:

POKE CODEA, .....

POKE CODEB, .....

RAND USR EXX

Végül a BORDER-el említeném, ami 3 pixel vastag keretet rajzol a nagy felbontású képernyőre.

PAPP ZOLTÁN

## MATEMATIKA Házi feladatok megoldására szolgáló program.

```
1 POKE 20609,80: LET mb=3000
2 LOAD "CODE"
3 INPUT "Név: Papp Zoltán © 1985":
10 LET a$="Bv"
20 LET p=0: LET b=0
30 DIM f$(5,64)
40 FOR x=1 TO 6: LET f$(x)="x"
NEXT X
50 DIM z$(mb): DIM w(1): LET w(1)=55
60 LET z$(1 TO 21)=a$
100 BORDER 6: PAPER 6: INK 0: BRIGHT 0: OVER ON: CLS
110 GO SUB 9000
120 PRINT TAB 11; BRIGHT 1;"Matematika"
130 PRINT
140 PRINT TAB 5; a$
150 PRINT : PRINT TAB 6; "Kalkulátor"
160 INPUT "egyenlet: "; a$
170 INPUT "állandó: "; b$
180 INPUT "függvény: "; c$
190 INPUT "szorzó: "; d$
200 INPUT "válasz: "; a$
210 IF a$=1 OR a$=2 OR a$=3 THEN GOTO 160
220 GO SUB 2+1000
230 GO TO 110
240 GO SUB 9000
250 GO SUB 9010
260 PRINT TAB 1; BRIGHT 1;"Kalkulátor"
270 IF p=1 THEN LPRINT "Kalkulátor"
280 CHR$ 13+CHR$ 13+a$+CHR$ 13
290 PRINT : PRINT "b=1020 eredmény: ret=15522"
300 INPUT c$
310 IF c$="" THEN GO TO 1060
320 IF c$="ret" OR c$="r" OR c$="r" THEN RETURN
330 FOR i=1 TO LEN c$: IF c$(i)=$"X" AND c$(i)!="": THEN GO TO 1110
340 IF c$(i)=$"RND" AND c$(i)=$"X" THEN GO TO 1110
350 IF "a+c(x)+b" AND c$(i)=$"X" OR c$(i)=$"R" THEN GO TO 1110
360 TO 1060
370 NEXT X
380 LET b=VAL c$
390 PRINT c$;"=";b
400 IF a$=1 THEN LPRINT c$;"=";b
410 GO TO 1060
420 GO SUB 9010
430 PRINT TAB 1; BRIGHT 1;"Házi feladatok"
440 PRINT : PRINT "a+xt2+b*x+c="
450 INPUT "a,b,c=? ";a1;" ";b1;" ";c1
460 PRINT : PRINT "(a1;b1)*xt2+(c1-b1)*x+(a1-b1)*b="
470 LET d1=1+4*a1*c1
480 IF d1<0 THEN PRINT "Nincs megoldás." : RETURN
490 IF d1=0 THEN PRINT "Egy megoldás." : PRINT
```

```
x=(b1-(2*a1)*RETURN
2000 PRINT TAB 1; BRIGHT 1;"Függvény"
2100 INPUT "Függvény neve? ";a$
2200 IF a$="" OR a$="f" THEN GO TO 2000
2300 LET c=CODE a$-95
2400 LET g$="FN "+a$+"(X)="
2500 PRINT PRINT g$+f$(1)
2600 INPUT "választás";(I/N)
2700 IF e$(i)="" AND e$(i)="" THEN RETURN
2800 INPUT "FN (X) = ";f$(i)
2900 PRINT "Helvett" : PRINT g$+f$(i)
3000 GO TO c+100+3100
3100 DEF FN a(x)=VAL f$(1)
3200 RETURN
3300 DEF FN b(x)=VAL f$(2)
3400 RETURN
3500 DEF FN c(x)=VAL f$(3)
3600 RETURN
3700 DEF FN d(x)=VAL f$(4)
3800 RETURN
3900 DEF FN e(x)=VAL f$(5)
4000 RETURN
4100 DEF FN f(x)=VAL f$(6)
4200 GO SUB 9010
4300 LET p=0
4400 PRINT TAB 1; BRIGHT 1;"Függvény ábrázolás"
4500 INPUT "Drágó? (0-250,0-170)";o$
4600 IF o$="250" OR o$="0" OR o$="0" OR o$="170" THEN GO TO 4000
4700 PRINT : PRINT "(X,OX);";"Y"
4800 INPUT "X(max) ? ";x$
4900 IF x$=<0 THEN GO SUB 1010
5000 GO TO 4000
5100 PRINT "Xmax=";x$
5200 LET a=x$(255-0x)
5300 PAUSE 30
5400 CLS
5500 PRINT AT 0,INT (0x/3)-1;"Y=";INT (175-0y)+a*100/100
5600 LET h$=STR$(INT (a+100)/1)
5700
5800 PRINT AT 20,INT (0y/8).30-L EN h$;X;"h$"
5900 PLOT 0;0; DRAW 255;0
6000 PLOT 0;0; DRAW 0;175
6100 LET s=10*INT (LN q/LN 10+2)
6200
6300 FOR i=INT (1-x$*0x/(255-0x))/f$(s)+f$(x) TO x$ STEP f$(s) PLOT (f$(s)+0x,0y+1); PLOT (f$(s)+0x,0y-1); N EN
6400
6500 FOR i=INT (1-x$*0x/(255-0x))/f$(s)+f$(s) TO x$(175-0y)/(255-0x) STEP f$(s) PLOT 0;1;f$(s)+0y; PLOT 0x+1;f$(s)+0y; NEXT I
6600 INPUT "Drágó? ";x$
6700 IF x$="0" OR x$="x1" OR x$="x0" OR x$="x0" OR x$="x1" OR x$="0" THEN GO TO 4100
6800 IF x$="x1" THEN LET x1=x$
6900 IF x$="x0" THEN LET x0=x$
7000 IF x$="x1" THEN LET x1=x$
7100 IF x$="x0" THEN LET x0=x$
7200 LET x2=x0+q+0x
7300 LET s=107-6
7400 LET x=x0
7500 LET k=VAL f$(s)+0y
7600 LET x2=x$+s
7700 LET x=(x2-0x)+q; IF x>x1 THEN EN GO TO 4500
7800 LET y=VAL f$(s)
7900 LET y2=0+q
8000 IF y2=0 OR y2>175 THEN LET s=1 GO TO 4200
8100 PLOT (x2,y)
8200 IF y2=0 THEN LET s=1 GO TO 4200
8300 LET s=g/(ABS (y2-k)/s)
8400 IF s=0 THEN LET s=1
8500 LET i=1
8600 GO TO 4200
8700 INPUT "Copy, Ismétlés, Függvény";a$
8800 IF a$="c" OR a$="c" THEN COPY GO TO 4500
8900 IF a$="i" OR a$="i" THEN GO TO 4100
9000 IF a$="f" OR a$="f" THEN GO TO 4000
9100 RETURN
9200 INPUT "Függvény asztaloptika";a$
9300 IF ABS (a$+q+0x-120)/125 THEN EN GO TO 4500
9400 PLOT a$+q+0x;0; DRAW 0;175
9500 GO TO 4500
9600 GO SUB 9000
9700 PRINT TAB 1; BRIGHT 1;"Szöveg"
9800 PRINT "a+xt2+b*x+c="
9900 PRINT "a,b,c=?";a$;b$;c$
1000 PRINT "a+xt2+b*x+c="
1010 INPUT "a,b,c=?";a1;" ";b1;" ";c1
1020 PRINT : PRINT "(a1;b1)*xt2+(c1-b1)*x+(a1-b1)*b="
1030 LET d1=1+4*a1*c1
1040 IF d1<0 THEN PRINT "Nincs megoldás." : RETURN
1050 IF d1=0 THEN PRINT "Egy megoldás." : PRINT
```

```

5055 IF V(1)+V-1 THEN LET W(1)+1
N VS GO TO 5060
5060 IF VS(V)=" " THEN GO TO 505
5070 NEXT V GO TO 5030
5075 IF V(1)+V-X+1>LEN Z$ THEN P
PRINT "Megtelt!" GO TO 110
5080 LET Z$(W(1)) TO W(1)+V-X)=VS
(X TO V-1)+CHR$ 13; PRINT CHR$ 8
+Z$(W(1)) TO W(1)+V-X-1)+CHR$ 13+
5090 LET W(1)=W(1)+V-X+1
5100 LET X=V
5110 NEXT X
5020 GO TO 5030
IF LEN W$#1 THEN GO TO 5030
5310 IF VS(2)="L" OR VS(2)="I" T
HEN LOAD "DATA W(1)"; LOAD "" DA
TA Z(1) GO TO 5500
5320 IF VS(2)="S" OR VS(2)="3" T
HEN INPUT "File név: " $ $ SAVE
$ $ DATA W(1); SAVE $ $ DATA Z(1);
GO TO 5030
5325 IF VS(2)="F" OR VS(2)="f" T
HEN GO TO 5600
5330 IF VS(2)="M" OR VS(2)="m" T
HEN INPUT "Mény képek? (1-7) $ $;
LET $b=$b+704; GO TO 50+( $b/350
0)+5330*( $b/3500)
5340 IF VS(2)="K" OR VS(2)="k" T
HEN GO SUB 1010; GO TO 5500
5350 IF VS(2)="E" OR VS(2)="e" T
HEN GO TO 5500
5360 IF VS(2)="P" OR VS(2)="p" T
HEN PRINT Z$(1 TO V(1)-1) GO T
O 5030
5370 IF VS(2)="N" OR VS(2)="n" T
HEN GO TO 50
5380 IF VS(2)="R" OR VS(2)="r" T
HEN RETURN
5400 GO TO 5030
5400 GO SUB 9010; PRINT Z$(1 TO W
(1)-2)+CHR$ 13+"; GO TO 5030
5500 LET W(1)=W(1)-2
5510 IF Z$(W(1))=CHR$ 13 THEN GO
TO 5030
5520 LET W(1)=W(1)-1; GO TO 5010
5530 LET W(1)=W(1)+1; GO TO 5500
9010 PRINT
9020 PRINT
9030 PRINT
RETURN
9050 INPUT "Printer ? (I/N) "; $
9060 LET P=$
9070 IF $="I" OR $="i" THEN LE
T P=1
9080 RETURN
9900 CLEAR : SAVE "MATEK" LINE 1
SAVE "BENK" CODE USE "R",130
9999 FOR X=0 TO 7 INPUT A: POKE
USR "T"+X,A: NEXT X

```

```

44522tes 34101
44523 10
44524 10
44525 10
44526 10
44527 10
44528 10
44529 10
44530 10
44531 10
44532 10
44533 10
44534 10
44535 10
44536 10
44537 10
44538 10
44539 10
44540 10
44541 10
44542 10
44543 10
44544 10
44545 10
44546 10
44547 10
44548 10
44549 10
44550 10
44551 10
44552 10
44553 10
44554 10
44555 10
44556 10
44557 10
44558 10
44559 10
44560 10
44561 10
44562 10
44563 10
44564 10
44565 10
44566 10
44567 10
44568 10
44569 10
44570 10
44571 10
44572 10
44573 10
44574 10
44575 10
44576 10
44577 10
44578 10
44579 10
44580 10
44581 10
44582 10
44583 10
44584 10
44585 10
44586 10
44587 10
44588 10
44589 10
44590 10
44591 10
44592 10
44593 10
44594 10
44595 10
44596 10
44597 10
44598 10
44599 10
44600 10
44601 10
44602 10
44603 10
44604 10
44605 10
44606 10
44607 10
44608 10
44609 10
44610 10
44611 10
44612 10
44613 10
44614 10
44615 10
44616 10
44617 10
44618 10
44619 10
44620 10
44621 10
44622 10
44623 10
44624 10
44625 10
44626 10
44627 10
44628 10
44629 10
44630 10
44631 10
44632 10
44633 10
44634 10
44635 10
44636 10
44637 10
44638 10
44639 10
44640 10
44641 10
44642 10
44643 10
44644 10
44645 10
44646 10
44647 10
44648 10
44649 10
44650 10
44651 10
44652 10
44653 10
44654 10
44655 10
44656 10
44657 10
44658 10
44659 10
44660 10
44661 10
44662 10
44663 10
44664 10
44665 10
44666 10
44667 10
44668 10
44669 10
44670 10
44671 10
44672 10
44673 10
44674 10
44675 10
44676 10
44677 10
44678 10
44679 10
44680 10
44681 10
44682 10
44683 10
44684 10
44685 10
44686 10
44687 10
44688 10
44689 10
44690 10
44691 10
44692 10
44693 10
44694 10
44695 10
44696 10
44697 10
44698 10
44699 10
44700 10
44701 10
44702 10
44703 10
44704 10
44705 10
44706 10
44707 10
44708 10
44709 10
44710 10
44711 10
44712 10
44713 10
44714 10
44715 10
44716 10
44717 10
44718 10
44719 10
44720 10
44721 10
44722 10
44723 10
44724 10
44725 10
44726 10
44727 10
44728 10
44729 10
44730 10
44731 10
44732 10
44733 10
44734 10
44735 10
44736 10
44737 10
44738 10
44739 10
44740 10
44741 10
44742 10
44743 10
44744 10
44745 10
44746 10
44747 10
44748 10
44749 10
44750 10
44751 10
44752 10
44753 10
44754 10
44755 10
44756 10
44757 10
44758 10
44759 10
44760 10
44761 10
44762 10
44763 10
44764 10
44765 10
44766 10
44767 10
44768 10
44769 10
44770 10
44771 10
44772 10
44773 10
44774 10
44775 10
44776 10
44777 10
44778 10
44779 10
44780 10
44781 10
44782 10
44783 10
44784 10
44785 10
44786 10
44787 10
44788 10
44789 10
44790 10
44791 10
44792 10
44793 10
44794 10
44795 10
44796 10
44797 10
44798 10
44799 10
44800 10
44801 10
44802 10
44803 10
44804 10
44805 10
44806 10
44807 10
44808 10
44809 10
44810 10
44811 10
44812 10
44813 10
44814 10
44815 10
44816 10
44817 10
44818 10
44819 10
44820 10
44821 10
44822 10
44823 10
44824 10
44825 10
44826 10
44827 10
44828 10
44829 10
44830 10
44831 10
44832 10
44833 10
44834 10
44835 10
44836 10
44837 10
44838 10
44839 10
44840 10
44841 10
44842 10
44843 10
44844 10
44845 10
44846 10
44847 10
44848 10
44849 10
44850 10
44851 10
44852 10
44853 10
44854 10
44855 10
44856 10
44857 10
44858 10
44859 10
44860 10
44861 10
44862 10
44863 10
44864 10
44865 10
44866 10
44867 10
44868 10
44869 10
44870 10
44871 10
44872 10
44873 10
44874 10
44875 10
44876 10
44877 10
44878 10
44879 10
44880 10
44881 10
44882 10
44883 10
44884 10
44885 10
44886 10
44887 10
44888 10
44889 10
44890 10
44891 10
44892 10
44893 10
44894 10
44895 10
44896 10
44897 10
44898 10
44899 10
44900 10
44901 10
44902 10
44903 10
44904 10
44905 10
44906 10
44907 10
44908 10
44909 10
44910 10
44911 10
44912 10
44913 10
44914 10
44915 10
44916 10
44917 10
44918 10
44919 10
44920 10
44921 10
44922 10
44923 10
44924 10
44925 10
44926 10
44927 10
44928 10
44929 10
44930 10
44931 10
44932 10
44933 10
44934 10
44935 10
44936 10
44937 10
44938 10
44939 10
44940 10
44941 10
44942 10
44943 10
44944 10
44945 10
44946 10
44947 10
44948 10
44949 10
44950 10
44951 10
44952 10
44953 10
44954 10
44955 10
44956 10
44957 10
44958 10
44959 10
44960 10
44961 10
44962 10
44963 10
44964 10
44965 10
44966 10
44967 10
44968 10
44969 10
44970 10
44971 10
44972 10
44973 10
44974 10
44975 10
44976 10
44977 10
44978 10
44979 10
44980 10
44981 10
44982 10
44983 10
44984 10
44985 10
44986 10
44987 10
44988 10
44989 10
44990 10
44991 10
44992 10
44993 10
44994 10
44995 10
44996 10
44997 10
44998 10
44999 10
45000 10

```



# a FORTH nyelv kedvelőinek ajánlja

## IBM PC/XT és azzal kompatibilis gépekre

### MicFORTH fejlesztőrendszerét

Elsősorban az ügyviteli adatfeldolgozási alkalmazásokat támogatja. Tartalmazza a FORTH—79, FIG—FORTH, MVP—FORTH és a STRING—FORTH szavait, továbbá screen-editor, lebegőpontos számábrázolást, FORTH adatbázis-kezelőt, hibakeresőt és javítóprogramokat (a program tetszés szerinti helyen megszakítható).

## Commodore 64 számítógépre

### MAX—FORTH 85 fejlesztőrendszerét

Segédeszköz, programozáshoz és teszteléshez. A programozási időt harmadára csökkenti, a futást 50-szer gyorsabbá teszi, mint a BASIC-ben.

**Új címünkön várjuk  
érdeklődését!**

1137 Budapest, Kun Béla rakpart 8.  
(Újpesti rakpart)  
Telefon: 129-230

# A MINŐSÉGÜGY KÖZÜGY

## Minőség – minősítés – értéktétel

A minőségre vonatkozó ítélet mindig valaminek vagy valakinek a számára valamilyen szempontú megfeleléseket, értékeséget mér, értékeségről informál, értékeséget indikál. Fontos kérdések tehát, hogy kinek, minek számára, milyen szempontú, milyen célra való értékeségről van szó, és ezt hogyan mérjük.

Még ha ugyanazt bíráljuk is, más lehet az értékeségre (minőségre) vonatkozó ítélet aszerint, hogy kinek, minek számára alkottuk, aszerint is, hogy milyen célra, milyen szempontokból alkottuk, és végül hogy hogyan, milyen eljárással alakítottuk ki azt. Ebből azonban következik az is, hogy az olyasféle ítéletek, hogy ez vagy az „jó minőségű”, hiányosak és ezért értelmentlen, nemritkán szándékosan félrevezetőek is. A minőségre vonatkozó ítélet helyes szerkezete a következő. „X minősége Y számára, Z célra történő alkalmazásra, felhasználásra stb. V értékű (mérhető, mérészámu stb.).”

A minőségre, értékeségre vonatkozó ítéletet alkotni kikerülhetetlen és mindig mindenki szívesen létező által állandóan gyakorolt tevékenység, minden létezőnek fő funkciója. Ennek fontossága – noha szemből – nem kapja meg a kellő hangsúlyt. Az értékeségre vonatkozó ítéletalkotást külön tanítani kellene. Az életben való navigációnk sikere függ attól, hogy milyen, milyen minőségű, a minőségre, értékeségre vonatkozó ítéletalkotási tevékenységünk.

Hogyan alkossunk tehát helyesen ítéleteket a minőségre, az értékeségre vonatkozóan? Általános érvényű megállapítások, egyszerű szabályok nyilván nem létezhetnek, hiszen túl sok jellemzőt kell figyelembe venni. (Ismétléljük: például, hogy kinek, minek számára? milyen célból? milyen célra? stb.) A helyzet azonban nem reménytelen. Nagy, mindent magába foglaló szabály gyakorlati elérhetetlensége ellenére van sok kis lokális eljárás, ami jó, hasznos, kezelhető. Ezek azonban túl sokan vannak ahhoz, hogy elviselhető terjedelemben tárgyalhatók legyenek.

A legtanulságosabb az lesz, hogy néhány konkrét módszert mutassunk be.

Adott tehát a személy, a szervezet stb., aki számára készül az értékeségi, a minőségre vonatkozó ítélet. Az is adott, hogy minek a minőségét kell elbírálnunk. (Lehet szó példá-

ul Kovács úrról és tisztagyapjú öltönyökről egy konkrét áruház felsőruházati osztályán, vagy Kovács úr gyermekeiről és huszonegyezer forint alatti, új személyi számítógépekről ugyanannak az áruháznak egy másik osztálya készletéből, de lehet szó Kovács úr által kidolgozott beruházási javaslatokról, amiket saját vállalata számára készített.)

A minősítendőket jellemzőkkel leírva ábrázolhatjuk egy olyan térben, amelyek koordinátengyelein az egyes jellemzők skálái vannak. (Számítottáknál például az ár, a garanciális idő, a központi tár kapacitása bitekben stb.) Ezt a teret hívjuk *reprezentációs térnek*. Ennek pontjai (esetleg más alakzatai) egy-egy minősítendő, illetve választási lehetőséget reprezentálnak, illetve reprezentálhatnak. Az egyik pontot (vagy más reprezentáló alakzatot) *jobb*nak ítéltjük egy másiknál, ha a neki megfelelő választási lehetőség jobb mint a másik pontnak (alakzatnak) megfelelő. Így tehát pontok, alakzatok jószágáról, értékeségről, minőségéről is beszélhetünk. A minőség, értékeség tehát a reprezentáló pontokhoz, alakzatokhoz valamilyen módon hozzárendelt érték. Ilyen egyszerű az egész. Adott például egy alkalmasan szerkesztett többváltozós függvény, amelynek értelmezési tartománya egy többdimenziós tér vagy annak egy részhalma, értékkészlete pedig egy valós intervallum. A minőségre, értékeségre vonatkozó ítélet kialakítása semmi másból nem áll, mint ilyen függvényekbe való helyettesítésből. E függvény azonban nem akármilyen függvények. Egy-egy ilyen függvény megfelel a már említett sok kis lokális szabály valamelyikének. Ezeket a függvényeket sajátjaik teszik alkalmassá arra, hogy konkrét helyzetekben alkalmazhatók legyenek. Az ilyen minősítő, értékesítő számot szolgáltató függvények fontossága, gyakorlati értéke tehát rendkívül nagy. E függvények az operációkutatás „célfüggvényeinek” közeli rokonai. (Egy jó „célfüggvény” kialakításának jelentősége pedig közismert.)

Lássuk tehát, hogyan készíthetünk például saját magunk számára, valamilyen célra értékelő, minősítő függvényt! A sok lehetőségek közül bemutatunk

egy egyszerű és könnyű gyakorlati utat.

Vegyük a lehetőségek között a lehető (reális) legjobbat és legrosszabbat! Egy lehetőséget a legjobb lehetőséghez és a legrosszabbhoz való viszonya alapján minősítünk. Ezt is sokféleképp tehetjük meg. Képezünk például a minősítő számot úgy, hogy a minősítendőnek a legrosszabb lehetőségtől vett valamilyen távolságából levonjuk a minősítendőnek a legjobb lehetőségtől vett valamilyen távolságát!

Természetesen több „rossz” lehetőséget és több „jó” lehetőséget is kijelölhetünk, és ezek mindegyikét egyszerre szerephez juttathatjuk az értékelő függvényben. Alkothatunk továbbá a kijelölt jó és rossz lehetőségek (esetleg pontonként mértékben és előjelben is különböző „jószágú” és „rosszaságú”) tartományokat is. A bírált lehetőség (például pont) eltérését a rossz lehetőségek tartományától értelmezhetjük geometriailag, valamilyen távolsággal, de például integrálokkal, többek között potenciál jellegű integrálokkal is. Érdekes és gyakorlati szempontból nagyon nagy jelentőségű az, hogy az operációkutatás legtöbb célfüggvénye nagyon egyszerűen származtatható az említett, geometriai jellegű, eltéréseket (távolságokat) használó megfontolásokkal. Ez a tény – noha nem közismert – természetes: minden létező egyékszik saját tulajdonságainak, adottságainak, korlátainak megfelelően (a) rossz lehetőségektől minél távolabb és (a) jó lehetőségekhez minél közelebb kerülni, mégpedig egyidejűleg, valamilyen rá jellemző módon. Az értékelő függvények egymástól való szerkezeti és értékelési különbségeiben mutatkoznak meg az egyes minősítőik sajátjaik közötti eltérések.

A minősítendő, értékelendő egyes jellemzői fontosságát és értéktartomány-elhelyezkedését szempontjából elterelhetnek egymástól; ezeknek az eltéréseknek helyes figyelembevétele süllyőzással is, koordináta-transzformációval is, az értékelő függvény megfelelő kialakításával is megoldható. E műveletek is függnek természetesen attól, hogy a minősítési, értékelési ítélet kinek, minek a számára, milyen célra készül.

Legyenek egy minősítendő jellemzői  $j_1, \dots, j_m$  és minden

jellemző olyan, hogy nagyobb értéke a jobb. Minősítendőnk tehát egy  $(j_1, \dots, j_m)$  vektorral, illetve egy  $J$  ponttal reprezentálható,  $J = (j_1, \dots, j_m)$ .

A  $J$  által reprezentált lehetőséget egyszerűsége kedvéért pl. öt lehetőséghez viszonyítva fogjuk megítélni. Az öt lehetőség közül három  $(U, V, Z)$  legyen rossz, kettő  $(A, B)$  pedig jó. Egy értékelő függvényt például a következőképpen lehet definiálni.

$$f(J) = f(U, V, Z, A, B, J)$$

Konkrétabban például:  $f(J) = f(\rho_1(U, J), \rho_2(V, J), \rho_3(Z, J), \rho_4(A, J), \rho_5(B, J))$  ahol  $\rho_1, \dots, \rho_5$  valamilyen eltérés, például valamilyen távolság.

Ez az értékelőfüggvény-definiálási mód rendkívül egyszerű és numerikusan könnyen kezelhető, még olyan esetekben is, amikor  $A, B, \dots, U, V, \dots$  illetve  $J$  nem pontok, hanem egyenesek, síkok, kör, gömb vagy más tartományok.

Nem nehéz belátni, hogy alkalmazható ez a módszer olyankor is, amikor nem egy értékelendőre, hanem értékelendőkre egyösszegére vonatkozóan kell minősítő ítéletet kialakítani.

Egyszerűség kedvéért álljon  $n$  elemű a minősítendő csapat (pl. brigád) és legyen  $k+1$  vonatkoztatási pontunk,  $k$  darab rossz,  $l$  darab jó. (A minősítendő  $J_1, \dots, J_n$  a rossz pontok  $R_1, \dots, R_k$ , a jók  $A_1, \dots, A_l$ .) Egy egyszerű és numerikusan könnyen kezelhető lehetőség a következő.

$$f(J_1, \dots, J_n) = \sum_{i=1}^n \sum_{j=1}^k \rho_{ij}(J_i, R_j) - \sum_{i=1}^n \sum_{j=1}^l \sigma_{ij}(J_i, A_j),$$

ahol a  $\rho_{ij}$  és a  $\sigma_{ij}$  valamilyen eltérés (távolság) adó függvények.

Ha a „jó”, a „rossz” és a minősítendő pontok száma megszámlálható végtelen, végtelen sorokkal kell dolgoznunk. Nagyobb számosságok esetében – ha az ilyen feladat numerikusan kezelhető –, integrálok kiszámítását (értékének megbecsülését) kell elvégeznünk. Fontosnak tartjuk hangsúlyozni, hogy ezek, a látszólag csak matematikai érdekességgé, bonyolultnak tűnő esetek nem elméleti spekulációkból születnek, hanem gyakorlati igények kielégítésére készült egyszerű eszközök az értékelés, a minősítés gazdag módszertárából.

Gyakorlásul írja fel az olvasó annak a minősítő függvénynek a formuláját, amely  $n$  elemű csapatokat minősít,  $k$  rossz és  $l$  (különböző) jó lehetőségtől vett távolságosságú (eltérésességű) hányadosával!

# Assemblerek, cross-assemblerek

## 7. Az operátorkészlet II. – A rendszer kapcsolatábrája

### 16. operátor: operandusmező-kiértékelő (1. ábra)

Feladata: a műveletkód-információk meghatározta módon értelmezett operandusmezőt értékeli ki.

Regiszter, regiszterpár és/vagy aritmetikai kifejezésekből felépített opreandus(ok) értéket számítja ki. Ha több operandusa van az utasításnak, akkor az egymástól vesszővel elválasztott operandusokat egyenként határozza meg. Regiszter és regiszterpár típusú operandus(ok) esetén a memonikus regiszternévhez tartozó és a tárgykódba beépítendő bináris adatot az ún. regiszter-, illetve regiszterpár-kódtábla tartalmazza.

Egyéb típusú operandus a következő egyszerű kifejezésekből épülhet fel: hexadecimális, oktális, bináris, decimális számkonstans; a beültetési számláló aktuális értéke; ASCII konstans (karakter típusú konstans); címke; azonosító.

Egy címke vagy azonosító értékét az aktuális címke táblából keresi ki az operátor, tehát ilyenkor aktiválja a 8. operátort. Makrokifejtés esetén, ha az aktuális makrocímke táblában nem találja meg a keresett szimbólumot, akkor a modulcímke táblában folytatja a keresést.

A fenti egyszerű kifejezésekből a következő — csökkenő precedencia szerint megadott — aritmetikai és logikai műveletekkel alakítható ki egy összetett kifejezés: negáció, negatív előjel; szorzás, egész osztás, bitenkénti léptetés jobbra és balra; összeadás, kivonás; logikai szorzás; logikai összeadás, kizáró vagy művelet (NOR).

Az egy összetett kifejezésben szereplő műveletek és egyszerű operandusok száma korlátozott. A kifejezés értékének meghatározása az ismert inverz lengyel módszerrel történhet.

Az operátornak működése végén nemcsak az operandus(ok) értékét, hanem az operandusok számát is jeleznie kell, s utánia kell arra is, hogy az operandusok értéke relatív vagy abszolút-e. (Ezeket az indikátorokat a 18. operátor használja fel.) Címket és a beültetési számláló értékét tartalmazó

Nyolcreszes 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 készítjük el az assembler működésének egy szabványosított algoritmusát, modelljét. E cikkünk a sorozat hetedik 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

kifejezések lehetnek relatív típusúak, azaz olyan kifejezések, amelyek értéke a fordítási kezdőcím értékétől függ.

Amennyiben az operátor az operandusmező kiértékelése során bármelyik operandusnál szintaktikai vagy szemantikai hibát fedez fel, akkor beállít egy hibaindikátort, és az operandusmező, illetve az operandus értékeként nullát ad ki.

### 17. operátor: tárgy kód-meghatározó (2. ábra)

Feladata: a műveletkód-információk és az operandus(ok) értékének segítségével meghatározza a műveletkód-mezőben álló gépi utasítás tárgfolgalásának hosszát (mindkét menemben) és tárgykódját (csak a

második menemben). A tárgykódba beépíti az esetleges regiszter- vagy reiszterpár-kódot, illetve az egyéb típusú operandus értékét.

### 18. operátor: tárgyprogram-készítő (3. ábra)

Feladata: a korábban meghatározott tárgykódot kiegészíti a tárgy kód flag- és hosszadatokkal, s ennek alapján bejegyzést készít a tárgyprogram puffertérletén (OBJB). Előzőleg az operátor ellenőrzi, hogy a kiszámított hosszúságú bejegyzés elfér-e a puffertérletben. Ha nem, akkor lezárja a tárgyprogram-rekordot és kiviszi a kívánt kimeneti perifériára, és a puffertérletet a terület elejére állítja be.

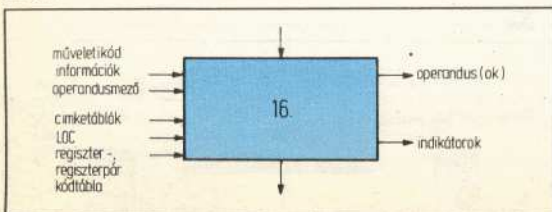
A tárgy kód flagek egy lehetséges kiosztása a következő:

- 1: gépi utasítás abszolút típusú operandusú vagy operandus nélkül,
- 2: gépi utasítás relatív típusú operandusú,
- 3: ORG direktíva,
- 4: END operandus nélkül,
- 5: END abszolút típusú operandussal,
- 6: END relatív típusú operandussal,
- 7: DB (mindig abszolút),
- 8: DW abszolút típusú operandussal,
- 9: DW relatív típusú operandussal,
- A: DS direktíva (mindig abszolút),
- B: tárgyprogramrekord vége,
- C: a tárgyprogram vége.

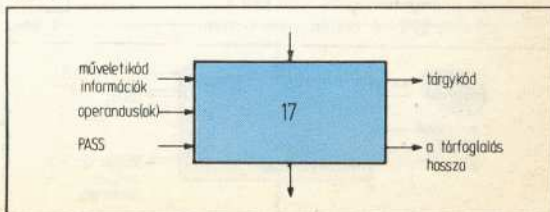
### 19. operátor: direktíva-feldolgozó (4. ábra)

Feladata: a fordítóprogramnak szóló utasítások (direktívák) végrehajtása. Így például az operátor működése során sor kerülhet bizonyos rendszerváltozók értékének megváltozására (IF,ENDIF,MACRO,ENDM,END direktívák esetén), a címke táblában készített egy bejegyzés módosítására (EQU,SET), a beültetési számlálónak történő értékadásra (ORG), a kívánt térfoglalás hosszának és esetleg tartalmának meghatározására (DB,DW,DS), a forrásprogram végének érzékelésére (END), a

1. ábra



2. ábra



tárgyprogramba történő bejegyzés elkészítésére (ORG,END,DB,DW,DS), s bizonyos szabályozó-vezérlő indikátoroknak és hibaindikátoroknak a beállítására például a 18. operátor számára (valamennyi direktívánál).

Az elvégzendő műveletek nemcsak a direktíva típusától, hanem a fordítási menettől is függenek.

Minden direktíva feldolgozását végző programrészlet (ehhez mutat a műveletkód-információban elhelyezett ugrási cím) önálló, s működésében egymást kizáró operátoroké fogható fel. Az ábra egyszerűsítése érdekében vontuk őket egyetlen operátorba össze.

Az egyes direktíváknál elvégzendő műveletek, azaz az egyes aoperátorok működése rendre:

DS: a tárfoglalás hosszának megállapítása (mindkét menetben); a tárgyprogrambejegyzés elkészítése (a második menetben),

DB,DW: a tárfoglalás hosszának kiszámítása (mindkét menetben); a tárfoglalás tartalmának és a tárgyprogrambejegyzésnek a meghatározása (a második menetben),

ORG: értékadás a beültetési számlálóknak (LOC) (mindkét menetben); tárgyprogrambejegyzés készítése (a második menetben),

END: a forrásprogram befejeződését jelző indikátor (IEND) beállítása; tárgyprogrambejegyzés készítése (a második menetben),

EQU: ha az adott forrásnyelvi sor címke-mezejében szereplő azonosítóhoz tartozó címke-tábla-bejegyzés szerint nem többszörös címke-definícióról van szó, akkor az operandusmező értékével írja felül a címkeértéket a címke-táblában, s beállítja a címkeflag végleges értékét (1 vagy 2),

SET: elvégzi az EQU direktívánál leírt műveleteket, továbbá ha a rendszer makrókifejtési állapotban van, megvizsgálja a modulcímke-táblát is, van-e ugyanolyan nevű SET direktívával definiált azonosító, s ha ilyet talált, akkor az ehhez tartozó bejegyzésben is módosította a címkeértéket és címkeflagot,

IF: vagy az IF1 vagy az IF2 rendszerválasztó értékét növeli eggyel. IF2 ≠ 0 esetén a beskatulyozott IF direktívák operandusmezőit nem kell kiértékelni, IF2 = 0 esetén viszont min-

den beskatulyozott IF direktíva operandusa számít, függetlenül IF1 értékétől,

ENDIF: IF1 vagy az IF2 rendszerválasztó értékét csökkenti eggyel; az előbbi akkor, ha IF2 aktuális értéke nulla, különben IF2-t,

MACRO: ha a rendszer az A1 állapotban van: bejegyzést készít a makró-definíciós táblába, s a DEF rendszerválasztóznak 1-es értéket ad, ha a rendszer az A2 állapotban van: a DEF rendszerválasztóznak 1-es értéket ad, ha a rendszer az MK1i állapotok egyikében van: az operandusmező tartalmát (ha nem üres) átviszi a megszakítási hely és a makróparaméterek verem-tárbá és lezárja a verembejegyzést.

A második menetben a MACRO direktíva sorára nincs szükség, ezért egy különleges, „az operátor kész a kifejezést indító MACRO feldolgozása után” elnevezésű kimenő jel segítségével megakadályozzuk, hogy a sor felkerüljön a háttértra.

MEND: ha a rendszer a DEF állapotban van: a DEF rendszerválasztóznak 0 értéket ad (visszatérés az A1 vagy az A2 állapotba), ha a rendszer az MK1i állapotban van: a kifejtési verembejegyzés alapján beállítja a visszatérési paramétereket és veremvisszaállítást hajt végre, azaz:

- a) ha az előző, megszakított szinten lemezről történt a forrásprogram olvasása, akkor a megszakítási helyre vonatkozó lemezcímről visszaolvassa SPB1-re az előző szint következő sorát,

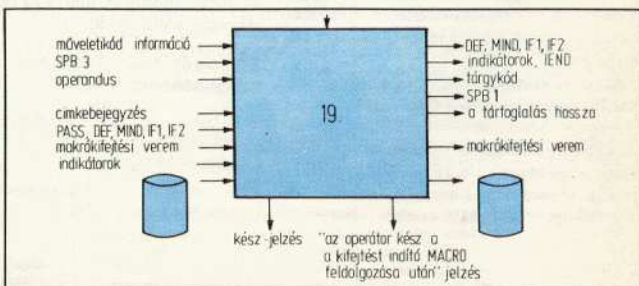
- b) ha a visszatérés a nulladik szintre fog történni és a bemeneti periféria nem mágneslemez, akkor SPB3-ról visszatölti a következő sort SPB1-re,
- c) lezárja az adott szinthez tartozó makrócímke-táblát és kiviszi a lemezre, a makrócímke-tábla-verem megfelelő bejegyzése által meghatározott címre,
- d) a megszakítási hely és makróparaméterek, valamint a makrócímke-tábla verem-tárbakban újból az előző szint bejegyzéseit érvényesíti,
- e) a MIND rendszerválasztó értékét eggyel csökkenti,
- f) ha MIND új értéke nem 0, akkor a memóriacímke-tábla verem-tárbé bejegyzése alapján visszahívja a DFB 4 puffere a lemezről a korábban megszakított és most visszatért szinthez tartozó makrócímke-táblát (illetve K > 1 esetén ennek utolsó rekordját), ha a rendszer az MK2i állapotban van: az előző lépések közül csak d)-t, de csak a makrócímke-tábla verem-tárbá, e)-t és f)-et hajtja végre.

## 20. operátor: forrásprogramot mágneslemezre író (5. ábra)

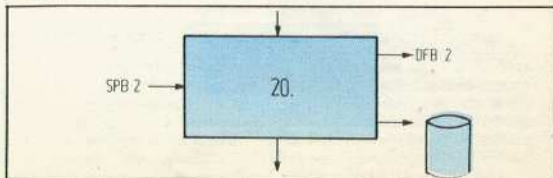
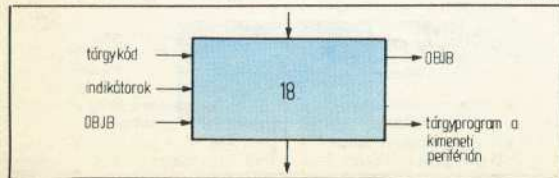
Feladata: az SPB2-n tárolt és az első menetben már feldolgozott forrásnyelvi sor háttértra való felírását végzi el.

A sort az SPB2 területről átviszi a DFB2 puffertérületre, s megvizsgálja, hogy az betelt-e. Csak betelés esetén kezdeményezi ténylegesen a lemezre írási műveletet.

4. ábra



3. ábra 5. ábra





## 21. operátor: sorfeldolgozási ciklust lezáró (6. ábra)

Feladata: lezárja az adott SPB2-n tárolt sor feldolgozási folyamatát, és felkészíti a rendszert a következő, pillanatnyilag még az SPB1-en tárolt sor feldolgozására.

Valamennyi indikátort (köztük a hibaindikátorokat) alapállapotba állítja vissza, ha szükséges, akkor a tárfoglalás hosszának megfelelő értékkel növeli a beültetési számláló értékét, s a második menetben szököz karakterekkel tölti fel a nyomtatópuffert (LPB).

## 22. operátor: a fordítás első menetét lezáró (7. ábra)

Feladata: az END direktíva beérkezése után lezárja a fordítás első menetét, és felkészíti a rendszert a második menet indítására.

Az operátor a következő műveleteket hajtja végre: a modulcímke-tábla utolsó (csonka) rekordját (MLB) lezárja, és kiviszi háttértárra — hacsak MLB nem üres; az összes indikátort alapállapotba állítja; a puffertérületek mutatóit, a lemez-munkaterületek címmutatóit a területek elejére állítja be; LMCT-t újra a makrócímke-táblák számára fenntartott terület kezdőcímevel tölti fel; a rendszerváltozóknak az induló A2 állapotnak megfelelő értékeket adja.

## 23. operátor: hibaüzenet-beállító (8. ábra)

Feladata: a forrásprogram címke-, műveletikód- és operandusmezőit feldolgozó operátorok által beállított hibaindikátorok alapján elhelyezi az esetleges hibaüzenetet a nyomtatópuffer (LPB) kijelölt területére.

Több hiba esetén az üzenet tartalmazhatja valamennyi hibát, vagy csak egy tetszőlegesen kiválasztottat. Magának az üzenetnek az összeállítására egy kódtábla segítségével

vel történik, amely az egyes hibaindikátorokhoz tartozó szöveges vagy numerikus üzenetkódot tartalmazza.

## 24. operátor: fordítási listát készítő (9. ábra)

Feladata: a feldolgozott forrásprogram-sort, a kísérő információkkal kiegészítve, kilistázza.

A fordítási lista tartalmát előre, a fordítóprogram tervezésekor kell rögzíteni. Egy lehetséges megoldásban soronként a következő mezők szerepelhetnek a listán:

- hibaüzenet (a 23. operátor tölti ki ezt a mezőt),
- sorszám: a forrásprogramban levő sorok sorszám, makrókifejtéskor a sorszám nem növekszik,
- a beültetési számláló értéke: a tárgykód — ha van — első bájtnak fordítási kezdőcíme,
- a generált tárgykód — ha van,
- az operandus értéke — ha van,
- referenciamező: ha az utasításnak vagy direktívának volt operandusa és ebben egy címke-re történt hivatkozás, akkor a hivatkozás helyét kapjuk meg ebben a mezőben,
- a forrásprogramsor tükörlistája (SPB2 tartalma).

Látható, hogy nem minden sornál lehet vagy kell valamennyi mezőt kitölteni, ennek szabályozását az operátor végzi el különféle, a korábban működött operátorok (16., 17., 19.) által beállított indikátorok alapján.

A 10. és 11. ábrán a most definiált 24 operátor segítségével összeállított kapcsolattábrát adjuk meg, amelyen a rendszer — a fordítóprogram — működése nyomon követhető. Mivel az egyes operátorokat jelző négyzetekbe csak a operátorok sorszámát jegyeztük be, az alábbi listában foglaljuk össze őket:

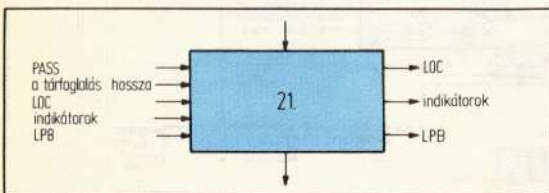
1. Paraméterlekérdező és kezdetiérték-beállító operátor

2. Forrásprogram-olvasó operátor nem lemez bemeneti perifériáról
3. Forrásprogram-olvasó operátor mágneslemezzel
4. Karakterlánc-mozgató operátor
5. Egy forrásprogramot mezőkre bontó operátor
6. Több-bites párhuzamos komparátor
7. Formális-aktuális paraméterscserét végző operátor
8. Címke-táblában kereső és címkeinformációt szolgáltató operátor
9. Címkeinformációt a címke-táblában elhelyező operátor
10. Többszörös címke-definiálást kezelő operátor
11. Címke-táblát lemezre író operátor
12. Makróhívást felismerő operátor
13. Makrókifejtést indító operátor
14. Műveletikód-kereső és műveletkód-információkat szolgáltató operátor
15. Direktíva-felismerő operátor
16. Operandusmező-kiértékelő operátor
17. Tárgykód-meghatározó operátor
18. Tárgyprogramkészítő operátor
19. Direktíva-feldolgozó operátor
20. Forrásprogramot mágneslemezre író operátor
21. Sorfeldolgozási ciklust lezáró operátor
22. A fordítás első menetét lezáró operátor
23. Hibaüzenet-beállító operátor
24. Fordítási listát készítő operátor

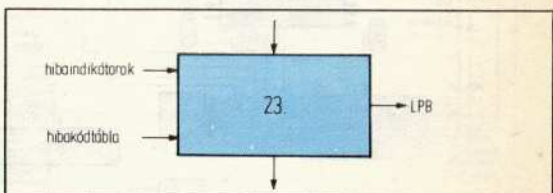
Az ábra áttekinthetősége érdekében egyes operátoroknál azokat a bemeneti és kimeneti változókat nem tüntettük fel, amelyeknek az adott menetben nincs szerepük.

Ahol egy operátor bemeneti vagy kimeneti változói között egy periféria szimbóluma jelenik meg, ott utalunk e periféria szerepére is. A szimbólumokba beírt betűk jelentése:

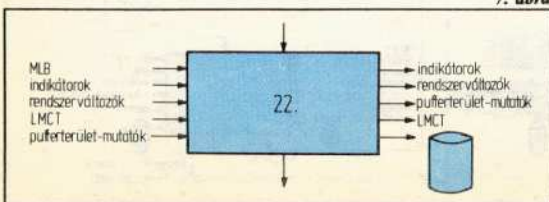
- a: az eredeti vagy a kifejtett forrásprogram adatállománya mágneslemez,
- b: makródefiniáció adatállománya mágneslemez,



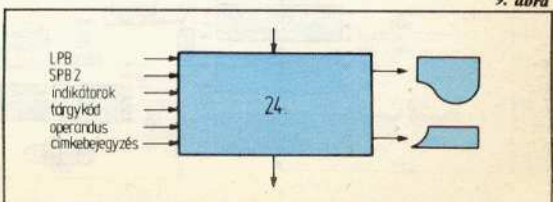
6. ábra



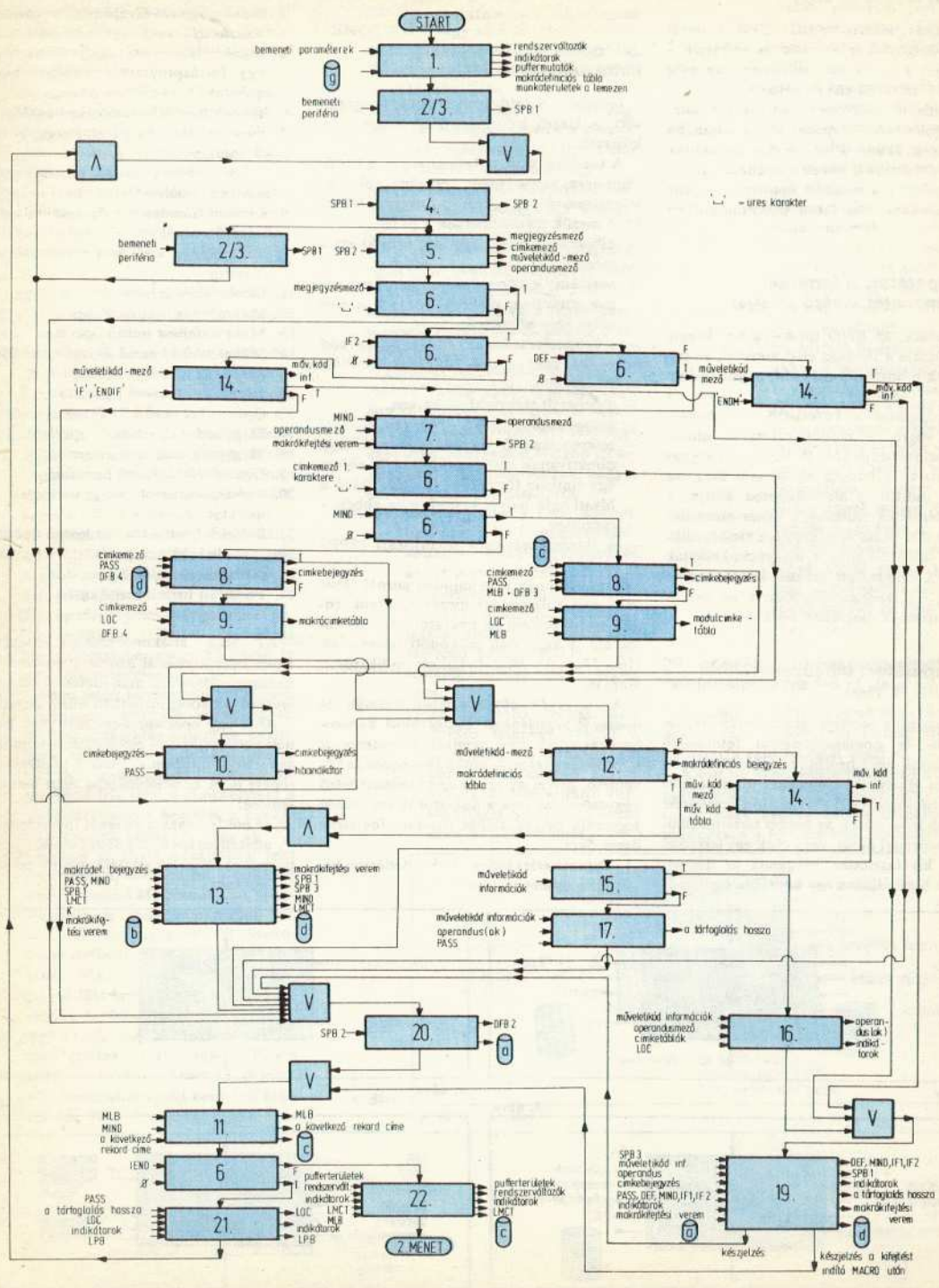
8. ábra

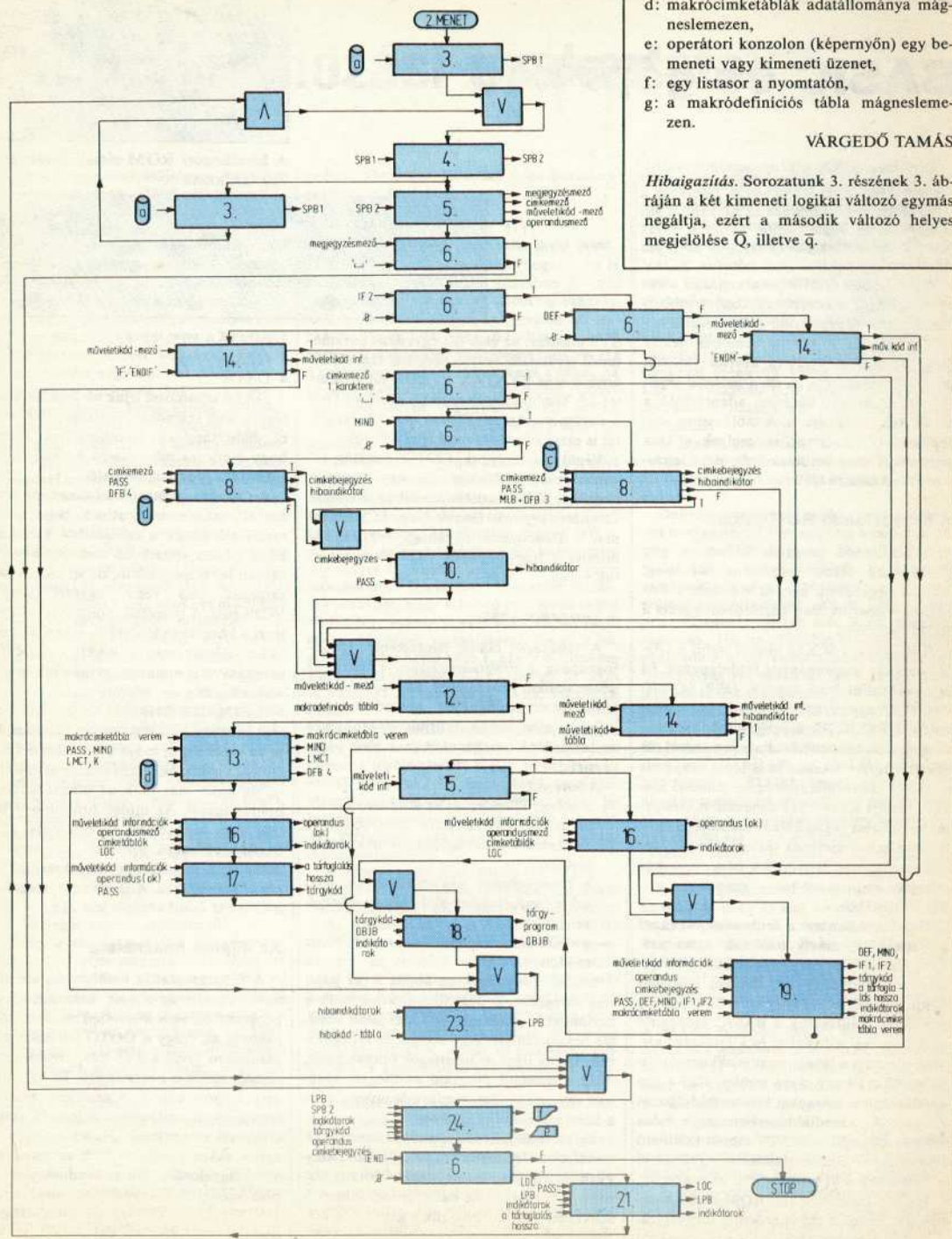


7. ábra



9. ábra





## BASIC és gépi kód

Utóljára a 65XX mikroprocesszor család regisztereiről és az áprilisi számban megjelent segédlet használatáról volt szó, és megismerkedtünk néhány vezérlésadáó utasítással is. Most ezek gyakorlati alkalmazásaira látunk egy példát.

A kiszámított GOTO programjának elkészítését fogjuk a korábbinál részletesebben megismerni. A gépi kódú rutin BASIC betöltőprogramja a júniusi számban jelent meg. Itt csak a VC20-ra készült változat programozását írom le, de a táblázat segítségével az eljárás könnyen adaptálható a másik két géptípusra is. A táblázatban szerepelnek olyan címek is, melyek a kész programba nem kerülnek bele, de a leírásban hivatkozom rájuk.

### A használandó ROM-rutinok

A készítő program három, a gép ROM-jában tárolt szubrutint hív meg, majd a negyedikre ugorva visszatér a BASIC-be. Ismerjük meg közelebbről ezeket a ROM-rutinokat.

A VC20-on a \$C8A0 címről indul a GOTO utasítás hagyományos feldolgozása. Itt egy szubrutint hívó utasítás (JSR \$C96B) van, mely egy olyan alprogramot hív meg, amely a BASIC forrásprogramban található egész számot beolvassa a programból (és egyúttal természetesen át is lépi), majd ezt a számot címfórmátumú egész számmá alakítva tárolja a \$14–\$15 címeken. A címfórmátumú egész olyan kétbájtos, előjel nélküli egész szám, melynek tárbeli elhelyezésétől az alacsony helyértékű bájttól — a már többször említett módon — megelőzi a magas helyértékűt.

Eljárásunkban ezt a szubrutint olyannal helyettesítjük, amely nemcsak egész számot, hanem tetszőleges aritmetikai kifejezést is ki tud értékelni. Ezt két lépésben hajthatjuk végre. Az elsőben a JSR \$CD8A utasítással beolvassuk a BASIC szövegéből az aritmetikai kifejezést, és kiértékelve azt, az eredményt a lebegőpontos akkumulátorba tesszük. Ez egy olyan terület, ahol a gép a valós típusú számokat tárolja feldolgozás közben. A második lépésben ezt a valós számot alakítjuk a \$D7F7 címen található szubrutinnal címfórmátumúra, egyúttal el is helyezzük a \$14–\$15 című bájtokon.

Ezeket kívül még egy ROM-rutint használunk. Ennek a címét érdemes megjegyezni, mert a paraméteres SYS utasításoknál gyakran találkozunk vele: \$CEFD. Ez a

VC20	C64	C16
\$C8A0	\$A8A0	\$8D4D
\$C8A3	\$A8A3	\$8D50
\$C96B	\$A96B	\$8E3E
\$CD8A	\$AD8A	\$9714
\$CEFD	\$AEFD	\$9491
\$D7F7	\$B7F7	\$9DE4

A hivatkozott ROM-címek összehasonlító táblázata

\$033C	JSR \$CEFD	20 FD CE	828: 32 253 206
	JSR \$CD8A	20 8A CD	32 138 205
	JSR \$D7F7	20 F7 D7	32 247 215
	JMP \$C8A3	4C A3 C8	76 163 200
	1	2	3

SYS utasítást az első (itt egyetlen) paramétertől elválasztó vessző meglétét figyeli, hiánya esetén SYNTAX ERROR hibüzenetet ad. Többparaméteres SYS utasításoknál a paramétereket elválasztó vesszők meglétéit is ezzel a szubrutinnal ellenőrizzük.

Végül visszatérünk GOTO utasítás további feldolgozásához. Ez úgy történik, hogy egy JMP utasítással a kihagyott utáni utasításra ugunk. Ennek címe: \$C8A3. Innen a feldolgozás ugyanúgy folytatódik, mintha a hagyományos GOTO-t használtuk volna.

### A programozás

A szükséges címek ismeretében hozzáfoghatunk a programozáshoz. Először az ábrán látható programnak az 1-es számmal jelölt részét készítjük el. Az indulási cím felírása után sorba felírjuk a szükséges szubrutinhívó utasításokat és a záró ugróutasítást.

A sorrend:

- a vessző ellenőrzése és átlépése,
- az aritmetikai kifejezés kiértékelése és elhelyezése a lebegőpontos akkumulátorban,
- a lebegőpontos akkumulátorban lévő szám címfórmátúra alakítása és elhelyezése,
- a GOTO feldolgozása a hagyományos módon.

Ezután következik az ábrán 2-vel jelölt rész elvégzése. A segédletből kikeressük a hárombetűs emlékeztetőköddal jelölt utasítás hexadecimális gépi kódját, és feljegyezzük. Utána írjuk az utasítások operandusát, mely a mostani program mindegyik sorában egy-egy tárcím, címfórmátumban, azaz a bájtok sorrendjét felcserélve.

Befejezéstül a hexadecimális számokat kicseréljük a decimális megfelelőjükkel, ahogyan az ábra harmadik részén látható. Ezután egy rögtönzött betöltőprogrammal a szükséges helyre bevisszük a gépi kódú utasításokat:

1 FOR I=828 TO 828+11

2 READ A : POKE I,A

3 NEXT

4 DATA ...

A DATA utasításba írjuk az ábra harmadik részén lévő számokat. Futtatás előtt célszerű háttértárolóra kimenteni az egészet, hogy esetleges hiba esetén ne kelljen újra kezdeni a program bevitelét.

A C64-en ugyanezt a módszert alkalmazhatjuk, csak a szubrutinok címe más, a megfelelő címek a táblázatból kereshetők ki. A C16-ra vonatkozó címeket is a táblázatban lehet megtalálni, de itt csak a programozás első része egyezik meg a VC20-éval. A beépített monitor feleslegessé teszi a következő lépéseket, a kézi assemblálást. Hívjuk meg a BASIC MONITOR utasításával a monitort, majd írjuk be a következő sort:

A \$33C JSR \$9491

A monitor lefordítja ezt az utasítást, kiírja az eredményt, majd a következő szabad címet. Vigyük be a következő utasításokat is, de most már csak az utasításkódot az operandussal. Az utolsó programsor bevitelén kiírt címre válaszuljunk a RETURN billentyű azonnali lenyomásával, aztán az X monitorparanccsal térjünk vissza a BASIC-hez. A program kimentéséről a C16-nál majd később lesz szó.

### Az eljárás használata

A futtatás után a betöltőprogram törölhető. Ezután az eljárás használható akár programból, akár közvetlen módban. A különbség az, hogy a GOTO sor-szám alakú utasításon kívül a SYS 828, aritmetikai kifejezés alakú is használható. Ez az utasítás arra a sorra adja át a vezérlést, amelynek sorszáma az aritmetikai kifejezés aktuális értékével megegyezik. Ha ez az érték nem egész, akkor az eljárás csak az egész részét veszi figyelembe. Ha az eredményként kapott számnak megfelelő sor nem létezik, UNDEF'D STATEMENT hibajelzést kapunk.

BARNA LÁSZLÓ

# Z80 programozási gyakorlatok

A gépben számokat előjelesen és előjel nélkül ábrázolhatunk. Előjel nélkül rendkívül egyszerűen, a szokásos kettes számrendszerbeli alaknak megfelelően; előjelesen ún. kettes komplementes ábrázolási móddal. Ha  $n$  biten kettes komplementes ábrázolással akarunk felírni  $m$  számot, akkor ha az nem negatív, a kettes komplementes alak megegyezik az előjel nélküli ábrázolásmóddal; ha  $m$  negatív, akkor az  $m+2^n$  szám kettes számrendszerbeli alakjával. Ennek lényegét mutatja négy bitre az 1. ábra.

Természetesen lehet öt, hat... stb. biten is ábrázolni kettes komplementes kódban számokat. Célszerű a központi egység (CPU) belső felépítése és utasításkészlete miatt 8 vagy 16 biten ábrázolni az egészet, és mivel nyolc bit csupán a  $(-128, 127)$  intervallumba eső számokat enged meg, ezért a legtöbb interpreter és fordító az egész számokat 16 biten ábrázolja.

Sorozatum e cikkében azt fogjuk megnevezni, hogy hogyan lehet az ilyen alakban felírni számokkal számolni. Ha az ADD vagy SUB utasítást használjuk, és nincs túlsordulás, azaz az eredmény a  $(-2^{18}, 2^{18}-1)$  intervallumba esik,  $n=8$  vagy  $n=16$ , akkor helyes eredményt kapunk az operandusok előjelétől függetlenül. Egy szám előjelét megállapíthatjuk csak a legfelső bit vizsgálatával, ugyanis ha ez egy, akkor a szám negatív, ha nulla, akkor a szám pozitív vagy nulla.

Érdekes azt is megfigyelni, hogy ha egy szám minden bitjét az ellenkezőjére váltjuk, majd a számhoz hozzáadunk egyet, akkor eredményül pontosan a szám mínusz egyszerűsített kapjuk (2. ábra), egy esetet kivéve. Ez az eset a  $-2^{(n-1)}$  ( $n$  a bitek száma), ugyanis a pozitív számokat csak  $2^{(n-1)}-1$ -ig tudjuk ábrázolni. Ha  $-2^{(n-1)}$  számról végezzük el a műveletet, akkor önmagát kapjuk eredményül.

Néhány gyakorlat a kettes komplementes kódábrázolás teljes megértéséhez. Írjuk fel négy- és nyolcbites alakban  $-1$ -et és  $-8$ -at. Számoljuk ki a megadott eljárással az ellentettjüket. Annak, aki most találkozik először ezzel az ábrázolással, azt ajánlom, hogy próbálja meg előbbi állításaimat bizonyítani, és gondolja végig, hogy IX vagy IY regiszterrel való címzésénél hogyan adódik ki a végleges cím, illetve hogy a relatív ugrásoknál hogyan kell kiszámítani az ugrás hosszát. Most pedig nézzük meg a példákat.

1. Írjunk szubrutint, amely HL-ben adott előzetes számnak kiszámítja az abszolút értéket! Carry jelezze a hibát!

A HL regiszterpárban, 16 biten ( $-2^{15}, 2^{15}-1$ ) intervallumba eső egész számokat

tudunk ábrázolni. Ez a két szám hexadecimálisan és binárisan

$$+2^{15}-1 = 7FFFh = /$$

$$0111\ 1111\ 1111\ 1111$$

$$-2^{15} = 8000h = 1000\ 0000\ 0000\ 0000$$

A HL-ben levő szám akkor negatív, ha H hetes bite 1. A bitek általában jobbról balra nullától hétig vagy tizenötig számozzuk, így a nullás bit  $2^0$ , az egyes bit  $2^1$  stb. helyiértéknek felel meg. Az abszolút érték fogalmát definiáljuk a szokásos módon. Ha a szám nem negatív, akkor maga a szám, ha negatív, akkor a mínusz egyszerűsége. Gondban csak akkor vagyunk, ha HL értéke  $-2^{(n-1)}$ . Ebben az esetben kell hibát jelezniünk. (1. program)

Ha HL nem negatív, akkor a carry bitet töröljük és visszatérés. Ha negatív, akkor átváltjuk az ellentettjére. Ezt két lépésben tehetjük meg, külön az alsó, majd a felső bájtot negálva, és növelve eggyel HL-t. A  $-2^{15}$  számot kivéve az eredmény mindig helyes és pozitív.  $-2^{15}$ -re a szubrutin lefutása után HL értéke változatlan marad, így tehát azt, hogy HL-ben ez a szám volt-e eredetileg, könnyen elődönthetjük az eredmény előjelének vizsgálatával. Ha ez a bit, a H regiszter hetes bite 1, akkor a szám negatív, pontosabban  $-2^{15}$ , és így az SLA A hatására a carry bit egy lesz, különben nulla.

Az előző részben arról is volt szó, hogy a carry bitet az OR A vagy az AND A utasítással is törölni lehet. Vajon itt miért nem ezt alkalmaztuk? Ha felcseréljük az előjelvizsgálatot és a carry bit törlését, akkor lehet-e ezen utasítások valamelyikét használni? Próbáljuk meg elkészíteni a program dokumentációját!

2. Írjunk szubrutint, amely HL és DE regiszterpárokban adott előjeles számokat összeadja! Carry bit jelezze a túlsordulást!

Az összeadásra lehet használni az ADD utasítást, csupán a túlsordulást kell figyelni. Ezt az előjelek vizsgálatával tehetjük meg. Ha a két szám előjele különböző, akkor nem lehet túlsordulás, mert az eredmény nagyobb mint a negatív, és kisebb mint a nem negatív operandus. Ha a két előjel megegyezik, akkor az eredmény előjele is ugyanannak kell lennie. (2. program)

A kizáró VAGY-ok alkalmazásakor koncentrálni csak a hetes, azaz előjelbitre. Ha a két előjel egyezik, vagyis H és D regiszterek előjelbitje azonos, akkor a kizáró vagy művelet hatására az akkumulátor hetes bite nulla, különben egy. Ez a bit kerül a carry bitbe. Ha különböző a két előjel, akkor összeadjuk a két számot, és mivel bizonyos, hogy nem volt hiba, majd a vissza-

térés előtt töröljük a carry bitet. Ha egyezik a két előjel, akkor az LB címkénél folytatódik a futás. Itt az akkumulátor hetes bitejébe kerül a közös előjel, majd az összeadás után, az eredmény előjelével vett kizáró VAGY művelet hatására az akkumulátor hetes bitejé egy lesz, ha a két előjel különböző, nulla, ha nem változott az előjel, azaz ha nincs túlsordulás, és a shiftelő utasítás ezt a bitet viszi a carry bitbe. A rutin megváltoztatja az A regiszter tartalmát. Az LB címkével megjelölt sorba írhattunk volna LD A,D utasítást is, a későbbiek folyamán azonban felhasználjuk, hogy nem ezt választottuk.

3. Írjunk szubrutint, amely HL-ből kivonja DE-t! Tekintsük HL-t és DE-t előjeles számoknak! Carry bit jelezze a túlsordulást!

Definiáljuk úgy a kivonást, mint az ellentett hozzáadását! Egy esetben nem tehetjük meg ezt, ha DE értéke  $-2^{(n-1)}$ . Ezt az esetet külön kell kezelni. Szükségünk van tehát egy olyan szubrutinra, amely egy számot az ellentettjére vált, és jelzi, ha éppen  $-2^{(n-1)}$ -ről van szó. Ilyen rutinunk már van. Ha az abszolút értéket számító programba a MINUSZ címkével jelölt helyen lépünk be, akkor az HL-t az ellentettjére váltja. Ha ezután carry = 0, akkor használjuk a PLUSZ rutint. Ha carry = 1, akkor is össze kell adni HL-t és DE-t, de DE előjelét pozitívnak kell tekintenünk annak ellenére, hogy D előjelbitje egy. Ha az összeadott rutinba a PLUS1 címmel lépünk be, és az akkumulátorba H regiszter értékét töltjük, akkor a szubrutin mindenütt pozitívnak tekintti DE-t. (3. program)

Mivel MINUSZ rutint HL-t változtatja, nekünk viszont DE mínusz egyszerűsége van szükségünk, ezért a szubrutint meghívása előtt és után ki kell cserélni HL és DE regiszterpárokat. A cseré nem változtatja meg carry bit értékét, és ha éppen  $-2^{15}$  a kivonandó, akkor az akkumulátor hetes bitejébe H előjelét tesszük, ami ha nulla, akkor egyezik, ha egy, akkor különbözik DE előjelétől, mivel DE-t pozitívnak tekintjük. A PLUS1 belépési címmel ennek a bitek ugyanaz a jelentése, amikor az összeadott rutin fut. A JP PLUS1 utasítás ugyanazt teszi, mint az egymást követő CALL PLUS1, RET utasítások. A program megváltoztatja az A regiszter tartalmát.

4. Írjunk szubrutint, amely HL értékét megszorozza tízzel. Carry jelezze a túlsordulást! Tekintsük HL-t előjel nélkülinek!

A legegyszerűbb megoldás, ha HL értéket áttöltjük BC-be vagy DE-be, és kilencszer hozzáadjuk HL-hez. Ez a megoldás azonban lassú. Ha HL-hez önmagát adjuk,

azzal kettővel szoroztuk. Ha ismét hozzáadjuk önmagához, akkor az eredeti érték négyeszeresét, ha ismét hozzáadjuk önmagához, akkor nyolcszorosat stb. kapjuk. Ilyen módszerrel ki tudjuk számítani HL bármely kettőhatványosorozását. A tíz nem kettőhatvány, de mint bármely szám, felírható kettőhatványok összegeként, azaz felírható kettes számrendszerben. A  $10 = 2^{13} + 2^{11}$  összefüggést felhasználva gyorsá válik a tízzel való szorzás. HL-t szorozzuk kettővel, azaz hozzáadjuk önmagához, ezt az értéket elteszük BC-be, majd még kétszer szorozzuk kettővel, és hozzáadjuk a BC-be eltett értéket. Ha bárhol túlszordulás van, akkor HL tiszterese nem fér el 16 biten. Bármelyik RET utasításnál is ér véget a program, a carry bit jelzi a túlszordulást. (4. program)

A szubrutin felhasználja BC regiszterpárt is.

A most következő program előtt meg kell hogy álljak egy pillanatra. Az első részben gépfüggetlen példákat ígértünk. A továbbiakban azonban szükségessé válik valamilyen gépi reprezentációhoz való kötődés. Így a következők példában feltételezzük egy PRNA nevű szubrutin létezését, amely a regiszterek értékének megváltoztatása nélkül az akkumulátorban lévő ASCII karaktert kírja például a képernyőre. Ez a szubrutin gépfüggetlen. A gép leírásában többnyire megtalálható valamilyen formában, hogy hogyan kell egy ilyen rutint megírni.

5. Írjunk szubrutint, amely HL-t mint előjel nélküli egész számot kírja a standard kimenetre decimálisan!

A szám maximum 2115 - 1 lehet. Ez decimálisan 65 535, tehát kisebb, mint százezer. Az algoritmus az első közelítésben a következő: addig vonunk le HL-ből 10000-et, hogy az még éppen negatívá ne váljon. Ahányszor ezt megtettük, az a tízezres helyiértéken álló számjegy. Ezt kírjuk. HL-ben ezután az eredeti érték 10000-rel való osztási maradéka van. Ezt a műveletet megismétljük ezer, száz, tíz levonogatásával is, majd a maradékok, ami az egyesek helyén álló számjegy, is kírjuk. Mit ad eredményül ez az algoritmus, ha HL-ben 32 van? A standard kimeneten 00032 fog megjelenni.

Valahogy el kellene kerülni, hogy a bevezető nullák kiíródjának. Az nem lehet megoldás, hogy ha a kurrens digit nulla, akkor nem iratjuk ki, mert így 102 helyett 12 lenne meg. Akkor nem kell csak kírni a nulla karaktert, ha az bevezető nulla, azaz még nem iratunk ki egy számjegyet sem, de ettől függetlenül az utolsó számjegyet mindenképpen ki kell iratnunk. Így HL=0 esetén is helyesen fog működni a program. A C regiszterben tároljuk, hogy volt-e már számjegykírás. A program elején C-t nul-

lázuk, ez jelzi, hogy még nem irattunk ki egy karaktert sem. Minden számjegy kiírása után C-t növeljük eggyel, így a szubrutin lefutása után C a kiírt jegyek számát mutatja. Mivel ez maximum öt lehet, így nem kell attól félni, hogy a növelés következtében C nullává válik. Az akkumulátort azért nullázzuk minden kiíratás előtt, mert ebben a regiszterben számoljuk, hogy DE-t hány-szor tudjuk levonni HL-ből. (5. program)

A számláláshoz felhasználtuk a már megírt összehasonlító rutint. Ennek visszatértekor carry=0, ha HL nagyobb vagy egyenlő mint DE; ekkor le kell vonni HL-ből DE-t. Mivel carry=0, ezért nem kell törölni. Ha carry=1, azaz nem lehet többször levonni, akkor meg kell vizsgálnunk, hogy nulla bájtról van-e szó. Ha nem, akkor A-t átalakítjuk ASCII kódra, kírjuk a karaktert, és növeljük C-t. C-ben mindig a már kiírt karakterek száma van. Ha nulla bájtról van szó, akkor a NULL címkenél folytatódik a program. Ha még nem iratunk ki egy jegyet sem, akkor most a nullát sem írjuk ki, ha már volt, akkor kírjuk. Az utolsó jegy mindenképpen kiíródik, mert itt nincs szükség számlálásra, és így a vizsgálatot is kihagyhatjuk. Tízezret, ezret, százat a DE regiszterpárba töltöttünk, tízet elég az E-be, mert ekkor D-ben már nulla van. Felhasználtuk azt is, hogy a feltételezett PRNA rutint nem változtatja meg a regiszterek értékeit. A program az A, C, D, E, H és L regisztereket használja fel. Meg lehet írni valamilyen rövidebben is a programot, az összehasonlító rutin hívása nélkül is, így valamilyen gyorsabb.

6. Írjunk szubrutint, amely HL-t mint előjeles egész számot kírítja a standard kimenetre decimálisan!

Az előző program megírása után ezt a problémát már nem nehéz megoldani. Ki kell iratni HL előjelbitjének megfelelően egy "+" vagy "-" jelet, esetleg csak szököt, meghívni az abszolút értéket számító szubrutint, nem kell törölni a túlszordulással sem, és kírni HL-t mint előjel nélküli egész számot. (6. program)

A program A, C, D, E, H és L regisztereket használja fel.

Assembly programozás során, de főleg gépi kódú programozásnál gyakran találkozunk a tizenhatos vagy más néven hexadecimális számrendszerrel. A kódtáblázatok ilyen alakban adják meg a kódokat, és a legtöbb monitorprogram is ezt a számrendszert használja. A következő két feladat ehhez a számrendszerhez kapcsolódik.

7. Írjunk szubrutint, amely a HL címen lévő kétbájtos számot kírja a standard kimenetre hexadecimális alakban.

```

1 Program
RES BIT 7,H :az előjel vizsgálata
SCF :carry
CFP :törlese
RET Z :ha Pozitív volt a szám
MINUSZ LD A,L :alsó bájtt
CPL :megállása
LD L,A :alsó bájtt
LD A,H :megállása
CPL :felső bájtt
LD H,A :megállása
INC HL :ha felső bájtt
LD A,H :a legfelső bájtt
SLA A :carrybe tolikuk
RET
    
```

```

2 Program
PLUS LD A,D :az előjelek
XOR H,A :vizsgálata
PLUS1 SLA A :törlesek az előjelek
JR NC,LC :megvesznek az előjelek
ADD HL,DE :carry törlesek
AND A :carry törlesek
RET
LB LD A,H :a közös előjel
ADD HL,DE :az eredmény előjele
XOR H,A :az eredmény előjele
SLA A :carry flasbe
RET
    
```

```

3 Program
KIV EX DE,HL :ellentettre
CALL MINUSZ :váltás
EX DE,HL :váltás
JP NC,PLUS :há összeadás
LD A,H :előjel bit
JP PLUS1
    
```

```

4 Program
TEN ADD HL,HL :szorzás kettővel
RET C :carryra szorzandóulka
LD B,H :BC-be az érték
LD C,L :kétazérés
ADD HL,HL :négyeszeres
RET C
ADD HL,HL :nyolcszoros
RET C
ADD HL,BC :tízszerez
RET
    
```

```

5 Program
OUTNUM XOR A,A :C és A
LD C,A :törlesek
LD L,DE :10000
CALL OUTDIO :tízezresek kiírattása
XOR A
LD L,DE :1000
CALL OUTDIO :ezresek kiírattása
XOR A
LD L,DE :100
CALL OUTDIO :százszok kiírattása
XOR A
LD L,DE :10
CALL OUTDIO :tízszok kiírattása
LD A,0
JR LAST :egyesek kiírattása
SZAML SBC HL,DE :carry=0
INC A :számlálás
OUTDIO CALL COMP :hasonlítás
JR NC,SZAML :nulla számjegy ?
OR A
JR Z, NULL :igen
LD A,0 :komerzid ASCII-ra
CALL OUTDIO :kírattás
INC C :számjegyek számlálás
RET :kötvetkez
LD A,C :ki kell-e
OR A :irni?
RET Z :zeren? nem
XOR A :A-ba 0-ra nulla
JR LAST
    
```

# A CAD alapelemei Commodore mikroszámítógépeken A három- dimenziós ábrázolás

1000	=	-8
1001	=	-7
1010	=	-6
1011	=	-5
1100	=	-4
1101	=	-3
1110	=	-2
1111	=	-1
0000	=	+0
0001	=	+1
0010	=	+2
0011	=	+3
0100	=	+4
0101	=	+5
0110	=	+6
0111	=	+7

1. ábra.

$$\begin{array}{r} 01100111 \\ + 10011000 \\ \hline 11111111 \end{array}$$

$$\begin{array}{l} A + \bar{A} = 2^n - 1 \\ A + 1 = 2^n - A \\ \bar{A} + 1 = -A \end{array}$$

2. ábra

## 6. Program

```
OUTDEC BIT 7,H ;előjelvizsgálat
JR Z,POZ
LD A,'-' ;minusz előjel
CALL PRNA ;előjel kiírás
CALL ABS ;abszolútérték
JP OUTNUM ;szám kiírása
POZ LD A,'+' ;lehet szóköz is
JR IR
```

A feladat megoldásához először próbáljunk meg egy segédprogramot írni, amelynek bemenő adata egy "0" és "F" közé eső egész szám az akkumulátorban, és amely ezt a számot kiírja hexadecimálisan. Ehhez át kell alakítani a számot a megfelelő ASCII kódra. Ha a szám kisebb mint tíz, akkor "0"-t kell hozzáadni, ha nagyobb mint kilenc, akkor „A” – 10-et. Ezt csinálja a program HEX címktől kezdődő része. Ezek után nincs más dolgunk, mint az akkumulátor alsó négy bitjébe belevárazsolni egymás után először a felső, majd az alsó bájti rendre felső, majd alsó félbájtiát, és mindig meghívni a HEX címktől kezdődő részt. Eltolási műveletre legalkalmasabb az RLD utasítás. Általában, ha azt külön ki nem hangsúlyozzák, akkor egy Z80-as rendszerben mindig az alacsonyabb címen van az alacsonyabb, a magasabb címen a magasabb helyiértékű bájti. Ez csupán szokás, de ezt segítik a központi egység 16 bites adatmozgató utasításai is, érdemes ehhez a szokáshoz ragaszkodni. Ha a HL címen levő számról beszélünk, akkor HL az alacsonyabbik értékű bájtra mutat, ezért kell HL-t növelni a szubrutin elején. Az AND műveletre azért van szükség, mert nem tudjuk, hogy mi volt az akkumulátor eredeti tartalma, és nekünk az kell, hogy a felső négy bit nulla legyen. Az AND n utasítás eredménye, ha n valamely kettőhatvány minusz egy, mindig az akkumulátor eredeti tartalma n + 1-gyel való osztási maradéka. (7. program)

A program használja A,C,H és L regisztereket, de csak C értékét írja felül. Próbáljuk meg ezzel a szubrutinnal kilisztáztatni a ROM egy darabját! Miért nem lehet?

8. Írjunk szubrutint, amely a HL regiszterpárban lévő számot kiírja a standard kimenetre hexadecimális alakban! Használjuk fel az előző szubrutint!

Ha fel akarjuk használni az előző szubrutint, akkor ahhoz HL értékét le kell tenni valamilyen memóriahelyre, és a címet tölteni HL-be. Legyen ez a memóriahely, ahova HL-t tesszük, a verem (stack). A címe benne van a veremmutatóban. Valahogyan át kell juttatni HL-be. Mivel erre nincs utasítás, ezért két lépésben valósítjuk meg. (8. program)

Érdeemes a program kapcsán megjegyezni, hogy a veremmutató mindig az utoljára eltett bájtra mutat. HL-t nem csupán azért kell kivenni a veremből, hogy megmaradjon az eredeti értéke, hanem azért is, hogy a RET utasítás a megfelelő értéket találja a veremben.

A CAD (Computer Aided Design) rendszerek fogalmán tágabban értelmezésben a számítógéppel segített tervezést értjük, melybe a rajzi megjelenítés, a térbeli ábrázolás, a formatervezés építvány beletartozik, mint a statikai számítások, költségvetés készítése vagy az anyagmennyiségek meghatározása. Szűkebb értelmezésben a CAD fogalmán a rajzi megjelenítés értjük: 2 dimenziós alaprajzok, nézetrajzok, metszetek (2D-rendszerek), formatervek, 3 dimenziós tömegvázlatok (3D-rendszerek) számítógépes előállítását.

Annak ellenére, hogy a CAD rendszerek többsége nagy központi egységgel, 16 vagy 32 bites processzorral, nagy kapacitású gyors háttértárolókkal, nagy felbontású grafikával (1024×1024 pont) rendelkező számítógépekre készült, több 8 bites mikro-gepre írt, sikeres CAD rendszer is bizonyítja, hogy mikroszámítógépekre is lehet professzionális színvonalú programokat készíteni (SCRIBE—Apple II/e).

A háromdimenziós tömegek és terek két-dimenziós síkon való ábrázolásának leg-szemléletesebb, a műszaki rajzok és tervek világában nem járatos felhasználók számára is közerthető módja az axonometria és a térbeli képi látvánnyal megegyező perspektíva.

Évszázados út volt, míg a perspektív ábrázolás törvényeit felismerték, és az összefüggések leírhatóvá váltak. A XV. században jelentek meg az ezzel kapcsolatos első munkák, melyeket a mai napig igen sok további tanulmány, könyv követett.

A pontos perspektív rajzok előállításának eszköze a szerkesztés volt, bár történetek kísérletek mechanikus szerkesztőgépek kifejlesztésére is. Annak ellenére, hogy a perspektív ábrázolás matematikai összefüggéseit már régen megfogalmazták, a számítás, mint perspektívaszerkesztési eljárás csak a számítógépek megjelenésével kapható létjogosultságot, általános gyakorlati felhasználása pedig a grafikus printerek és a plotterek elterjedésével vált lehetővé.

A 3 dimenziós programrendszerek bizonyultsági foka különböző. Az igényesebb rendszerekben a takart élek törése, vetett

## 7. Program

```
OUTHEX INC HL ;két bjt.
LD B,2 ;akkumulátorba
RLD ;felső 4bít
CALL HEX ;akkumulátorba
RLD ;alsó 4bít
DEC HL ;alsó bájtra visszahelyezés
BITN BF
INC HL
RET
HEX LD C,A ;A elemeztése
AND BFH ;maszkolás
CP 10
JR C,SKIP ;ha kisebb mint 10
ADD A,A,'0'-10
SKIP ADD A,'0'
CALL PRNA ;kiírás
LD B,C ;eredeti tartalom
RET
```

## 8. Program

```
OUTHLX PUSH HL ;a stack tetejére
LD HL,0 ;HL-be
ADD HL,SP ;SP értéke
CALL OUTHEX
POP HL ;HL vissza
RET
```



A. Dürer: „Der Zeichner des liegenden Mannes.“ U.D.M. 1538: 21,5-757

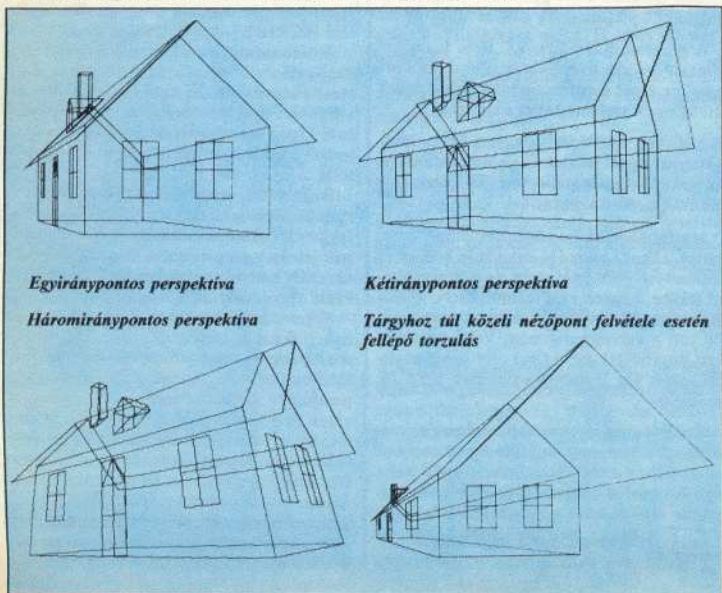
1. ábra. A. Dürer 1538-ban készített metszete. Az ábrázolt perspektívaszerkesztési eljárás megegyezik a program számítási elvvel

és önárnyék szerkesztése és tónusozása, az ábrázolandó tárgyban belüli nézőpont felvételének lehetősége egyaránt megoldott. A rendszerek értékét nagyban meghatározza a bennük alkalmazott adatbeviteli mód. Egyes rendszerekben a csúcspontok koordinátáit és azok összekötési sorrendjét kell megadni, máshol az ábrázolandó testet éllel, határoló lapjaival vagy egyszerű tömegek összegezésével definiálhatjuk.

A programlista egy egyszerű 3 dimenziós program szervezését és alapvető rutinjait tartalmazza. A program megírása döntő részben a Commodore 64 BASIC 2.0 verziójával történt, de e nyelv fogyatékoságai miatt a közismert Simon's BASIC bővítést is felhasználtuk.

A program modulokból áll, melyek módosíthatók, cserélhetők. A korlátozott terjedelem miatt az egyszerűbb rutinokat nem közöljük, megírásuk azonban nem nehéz feladat. A program alkotóegységei a következők.

2. ábra. Lakóépület különböző nézőpontokból felvett perspektív rajzai



## 1. Menü

A programot szervező egység. Az aktuális modul inverz alakban jelenik meg, ennek változtatása a kurzormozgató, hívása a RETURN billentyűvel történik.

## 2. Alapadatok beírása

A néző és a főpont koordinátáinak megadását szolgálja. Ezen adatok és az ábrázolandó pontok megadása ugyanabban a tészőleges koordináta-rendszerben történik.

A néző koordinátáinak megadása:

```
10 REM KURZORVEGZARLO TELFK
20 REM CLR
30 REM HOME
40 REM KURZOR JOBBRA
50 REM KURZOR BALRA
60 REM KURZOR LE
70 REM KURZOR FEL
80 REM TÁJÉKZÁS
90 REM TÁJÉKZÁS
```

```
100 PH=200: DIM X(PI), Y(PI), Z(PI)
110 DIM X(PI), Y(PI), Z(PI)
120 EH=200: DIME(PI)
130 ME(0)= 0: ME(1)=0
140 DIM#(0)
```

```
150
1600 REM --MENU-----
1610 PI=0:GOSUB1200
1620 ME(0)= "MENU"
1630 ME(1)= "ALAPADATOK"
1640 ME(2)= "PONTOK KOORDINÁTÁI"
1650 ME(3)= "ÉLEK BEÍRÁSA"
1660 ME(4)= "SÁJZOLÁS"
1670 ME(5)= "ADATOK TÁJÉKOZTATÁSA"
1680 ME(6)= "KILÉPÉS A PROGRAMBÓL"
1690 FOR I=0TO6:PRINT@0.11+I:ME(I):NEXT I
1700 PRINT@0.11+I: "S' ME(PI)"
1710 BETA=
1720 IF K=0 THEN I=0
1730 IF I=0 THEN I=1
1740 PRINT@0.11+I:ME(PI)
1750 MI=MI+1:PRINT "S' ME(PI)"
1760 IF K=1 THEN I=0
1770 IF MI=6 THEN I=0
1780 MI=MI-1:PRINT@0.11+I: "S' ME(PI)"
1790 PRINT@0.11+I:ME(PI)
1800 IF K=2 THEN I=0
1810 GOTO1110
1820 GOSUB1200
1830 GOTO1110
1840
```

```
1820 ONI+18TO110,200,300,400,500,600,700
1830 GOTO1110
1840
```

```
1850 PRINT " "
1860 PRINT " "
1870 PRINT " "
1880 PRINT@0.22: " "
1890 PRINT@0.23: " "
1900 PRINT@0.24: " "
1910 PRINT@0.25: " "
1920 PRINT@0.26: " "
1930 RETURN
1940
```

```
2000 REM --3-ALAPADATOK-----
2010 PI=0:GOSUB1200
2020 N(0)=0: ME(0)=0
2030 N(1)=0: ME(1)=0
2040 N(2)=0: ME(2)=0
2050 N(3)=0: ME(3)=0
2060 N(4)=0: ME(4)=0
2070 N(5)=0: ME(5)=0
2080 FOR K=1TO6
2090 PRINT@0.14+K:ME(K): " "
2100 NEXT K
```

```
2110 K=1:PRINT@0.14+K: "S' N(0)"
2120 AL=0:FOR I=0TO5:PRINT@0.14+K: "S' N(0)"
2130 BETA=0: IF K=1 THEN I=0
2140 IF K=1 THEN I=0
2150 IF K=1 THEN I=0
2160 IF K=1 THEN I=0
2170 IF K=1 THEN I=0
2180 GOTO2130
2190
```

```
2200 IF K=2 THEN RETURN
2210 PRINT@0.14+K:ME(K)
2220 K=K+1:PRINT@0.14+K: "S' N(0)"
2230 RETURN
2240
```

```
2250 IF K=2 THEN RETURN
2260 PRINT@0.14+K:ME(K)
2270 K=K+1:PRINT@0.14+K: "S' N(0)"
2280 RETURN
2290
```

```
2300 REM --5-----
2310 PRINT@25.14+K: "S' "
2320 BETA=
2330 IF K=1 THEN I=0
2340 IF K=1 THEN I=0
2350 IF K=1 THEN I=0
2360 IF K=1 THEN I=0
2370 IF K=1 THEN I=0
2380 PRINT@20.14+K: "S' "
2390 GOTO2330
2400 AL=0: FOR I=0TO5
2410 PRINT AT(25.14+K): " "
2420 GOSUB2200
2430 RETURN
2440
```

```
3000 REM --3-PONTOK KOORDINÁTÁI-----
3010 PI=0:GOSUB1200
3020 I=1:RESPT800
3030 READX(1): IF X(1)=999 THEN I=3070
3040 READY(1): IF Y(1)=999 THEN I=3070
3050 READZ(1): IF Z(1)=999 THEN I=3070
3060 I=I+1: GOTO3030
3070 I=I-1:GOTO3010
3080
4000 REM --4-ÉLEK BEÍRÁSA-----
4010 GOSUB2200
4020 I=1:RESPT 3000
4030 READE(I): IF E(I)=999 THEN I=4050
4040 I=I+1: GOTO4030
4050 I=I-1:GOTO3010
4060
```

A néző nem lehet az ábrázolandó tárgyon belül.  
A tárgyhöz túl közeli nézőpont felvétele esetében fellépő erős torzulások miatt (fényképezésnél nagy látószög objektív) célszerű a nézőpontot a 60 fokban kisebb látókép figyelembevételével megadni.



```

8000 REM -5-PERSPEKTIV SZANTAGA-----
8010 GOTO8250
8020 PRINTAT(5,10):1 -- MONITOR UAGY TU
8030 PRINTAT(5,12):2 -- PRINTER (MPS-801)
8040 PRINTAT(5,14):3 -- PLOTTER (1520)
8050 GOTO14
8060 IFPL<0 THENGOSUBS10
8070 IFPL<=1 THENGOSUBS10
8080 IFPL<=2 THENGOSUBS200
8090 IFPL<=3 THENGOSUBS240
8100 IFPL<=4 THENGOSUBS270
8110 GOTO8950
8120 PRINTAT(2,23) TURELEM
8130 IFAL(5)≠AL(2) THENAL(5)=AL(5)+AL(5)+1000000
8140 IFAL(5)≠AL(2) THENAL(5)=AL(5)+1000000
8150 GA=ATN(TAL(0)-AL(3))/AL(5)-AL(2)/33
8160 FI=ATN(TAL(4)-AL(1))/AL(5)-AL(2)/33
8170 G1=SIGN(FI)*31+GOSUBS1
8180 G2=SIGN(G1)*31+GOSUBS1
8190 FORI=1TO10
8200 XI=YY(I)-AL(2)
8210 VI=YY(I)+AL(2)
8220 XX(I)=XI*XI+VI*VI
8230 YY(I)=XI*XI+VI*VI
8240 NEXTI
8250 FORI=1TO10
8260 VI=YY(I)
8270 ZI=Z(I)-AL(3)
8280 YY(I)=VI*XI+ZI*XI
8290 ZZ(I)=VI*ZI+ZI*ZI
8300 NEXTI
8310 NA=1+J*H+J*H
8320 FORI=1TO10
8330 IFYY(I)≠0 THENXX(I)=XX(I)+YY(I)-ZZ(I)+YY(I)
8340 XX(I)=XX(I)+NA*YY(I)+NA*XX(I)+NA*YY(I)
8350 FORI=1TO10
8360 IFXX(I)≠0 THENHENON=XX(I)
8370 IFYY(I)≠0 THENHENON=YY(I)
8380 IFXX(I)≠0 THENHON=XX(I)
8390 IFXX(I)≠0 THENHON=XX(I)
8400 IFYY(I)≠0 THENHON=YY(I)
8410 NEXTI
8420 IFHON<>0 THENCYKTHENSZ0
8430 PRINTAT(02,23) HUBBS ADATBEADAS $RETURN
8440 GOTO4
8450 IFHON<>0 THENGOSUBS13 THENS440
8460 GOTO1010
8470 NA=310+J*H+J*H *-1
8480 NI=150+J*H+J*H
8490 IENI=0 THENHON=NI
8500 JI=NA*XX(I)+310+NA*HON-XX(I)/2
8510 JI=NA*XX(I)-150+NA*HON-XX(I)/2
8520 FORI=1TO10
8530 XI=XX(I)+NA*JI
8540 XX(I)=XI+XI-XI
8550 VI=YY(I)+NA*JI
8560 IFAL(5)≠AL(2) THENFI(1)+199-VI ELSE:YY(I)=VI
8570 NEXTI
8580 PRINTAT(2,23) KIJUTELI FASZ027
8590 RETURN
8600
8610 HIREM(1)=FI+1
8620 FORI=1TO10
8630 IFAL(1)+1=0 THENI(1)=0 THENS650
8640 L=INEX(XX(I)),YY(I),XI,XX(EE(1)+1),YY(EE(1)+1),XI
8650 NEXTI
8660 PAUSE1000:REM
8670 RETURN
8680
8690 IFFF=0 THENGOSUBS10
8700 IFFF=0 THENGOSUBS10
8710 CSFT2:COPY:REM
8720 RETURN
8730
8740 OPEN1,0,1
8750 IFFF=0 THENGOSUBS10
8760 CSFT2
8770 FORI=1TO10
8780 IF(EE(1)-1)≠0 AND(EE(1)+1)≠0 THENS800
8790 PRINTAT(1,1) STR$(S20+XX(EE(1))),STR$(YY(EE(1))),GOTO1010
8800 PRINTAT(1,0) STR$(S20+XX(EE(1))),STR$(YY(EE(1)))
8810 NEXTI
8820 CLS:FI=0:REM
8830 RETURN
8840
8850 REM -6-ADATTAROLAS-----
8860 REM RUTIN HELYE
8870 GOTO1010
8880
8890 REM -7-KILEPES A PROGRAMBOL
8900 PRINT"0 VESE" FND
8910
8920 REM -8-PONTOK KOORDINATAI
8930
8940 DATA 0, 0, 0
8950 DATA 5, 0, 0
8960 DATA 5, 0, 0
8970 DATA 0, 0, 3
8980 DATA 5, 0, 3
8990 DATA 5, 0, 3
9000 DATA 0, 0, 3
9010 DATA 4, 1, 2, 5
9020 DATA 5, 5, 1, 2, 5
9030 DATA 5, 5, 10, 4, 5
9040 DATA 5, 10, 5, 10, 5
9050 DATA 2, 5, 1, 5, 5
9060 DATA 2, 5, 10, 5, 5
9070 DATA 2, 5, 0, 5, 5
9080 DATA 2, 5, 0, 5, 5
9090 DATA 2, 5, 0, 5, 5
9100 DATA 2, 5, 0, 5, 5
9110 DATA 2, 5, 0, 5, 5
9120 DATA 2, 5, 0, 5, 5
9130 DATA 2, 5, 0, 5, 5
9140 DATA 2, 5, 0, 5, 5
9150 DATA 2, 5, 0, 5, 5
9160 DATA 2, 5, 0, 5, 5
9170 DATA 2, 5, 0, 5, 5
9180 DATA 2, 5, 0, 5, 5
9190 DATA 2, 5, 0, 5, 5
9200 DATA 2, 5, 0, 5, 5
9210 DATA 2, 5, 0, 5, 5
9220 DATA 2, 5, 0, 5, 5
9230 DATA 2, 5, 0, 5, 5
9240 DATA 2, 5, 0, 5, 5
9250 DATA 2, 5, 0, 5, 5
9260 DATA 2, 5, 0, 5, 5
9270 DATA 2, 5, 0, 5, 5
9280 DATA 2, 5, 0, 5, 5
9290 DATA 2, 5, 0, 5, 5
9300 DATA 2, 5, 0, 5, 5
9310 DATA 2, 5, 0, 5, 5
9320 DATA 2, 5, 0, 5, 5
9330 DATA 2, 5, 0, 5, 5
9340 DATA 2, 5, 0, 5, 5
9350 DATA 2, 5, 0, 5, 5
9360 DATA 2, 5, 0, 5, 5
9370 DATA 2, 5, 0, 5, 5
9380 DATA 2, 5, 0, 5, 5
9390 DATA 2, 5, 0, 5, 5
9400 DATA 2, 5, 0, 5, 5
9410 DATA 2, 5, 0, 5, 5
9420 DATA 2, 5, 0, 5, 5
9430 DATA 2, 5, 0, 5, 5
9440 DATA 2, 5, 0, 5, 5
9450 DATA 2, 5, 0, 5, 5
9460 DATA 2, 5, 0, 5, 5
9470 DATA 2, 5, 0, 5, 5
9480 DATA 2, 5, 0, 5, 5
9490 DATA 2, 5, 0, 5, 5
9500 DATA 2, 5, 0, 5, 5
9510 DATA 2, 5, 0, 5, 5
9520 DATA 2, 5, 0, 5, 5
9530 DATA 2, 5, 0, 5, 5
9540 DATA 2, 5, 0, 5, 5
9550 DATA 2, 5, 0, 5, 5
9560 DATA 2, 5, 0, 5, 5
9570 DATA 2, 5, 0, 5, 5
9580 DATA 2, 5, 0, 5, 5
9590 DATA 2, 5, 0, 5, 5
9600 DATA 2, 5, 0, 5, 5
9610 DATA 2, 5, 0, 5, 5
9620 DATA 2, 5, 0, 5, 5
9630 DATA 2, 5, 0, 5, 5
9640 DATA 2, 5, 0, 5, 5
9650 DATA 2, 5, 0, 5, 5
9660 DATA 2, 5, 0, 5, 5
9670 DATA 2, 5, 0, 5, 5
9680 DATA 2, 5, 0, 5, 5
9690 DATA 2, 5, 0, 5, 5
9700 DATA 2, 5, 0, 5, 5
9710 DATA 2, 5, 0, 5, 5
9720 DATA 2, 5, 0, 5, 5
9730 DATA 2, 5, 0, 5, 5
9740 DATA 2, 5, 0, 5, 5
9750 DATA 2, 5, 0, 5, 5
9760 DATA 2, 5, 0, 5, 5
9770 DATA 2, 5, 0, 5, 5
9780 DATA 2, 5, 0, 5, 5
9790 DATA 2, 5, 0, 5, 5
9800 DATA 2, 5, 0, 5, 5
9810 DATA 2, 5, 0, 5, 5
9820 DATA 2, 5, 0, 5, 5
9830 DATA 2, 5, 0, 5, 5
9840 DATA 2, 5, 0, 5, 5
9850 DATA 2, 5, 0, 5, 5
9860 DATA 2, 5, 0, 5, 5
9870 DATA 2, 5, 0, 5, 5
9880 DATA 2, 5, 0, 5, 5
9890 DATA 2, 5, 0, 5, 5
9900 DATA 2, 5, 0, 5, 5
9910 DATA 2, 5, 0, 5, 5
9920 DATA 2, 5, 0, 5, 5
9930 DATA 2, 5, 0, 5, 5
9940 DATA 2, 5, 0, 5, 5
9950 DATA 2, 5, 0, 5, 5
9960 DATA 2, 5, 0, 5, 5
9970 DATA 2, 5, 0, 5, 5
9980 DATA 2, 5, 0, 5, 5
9990 DATA 2, 5, 0, 5, 5

```

adjuk meg. A valóságú ábrázolás érdekben célszerű, a főpont a tárgy geometriai középpontjához közel esik.

Amennyiben a néző és a főpont megfelelő koordinátái különböznek, úgy a számított ábra háromirányponthos, ha egy koordináta azonos, úgy kétirányponthos, ha két koordináta azonos, akkor egyirányponthos a perspektív ábra.

### 3. A pontok koordinátáinak megadása

tetszőleges, célszerűen úgy történik, hogy az ábrázolandó tárgy méretei könnyen leolvashatók legyenek.

A korlátozott terjedelem miatt a pontok koordinátáit beíró programrészt egy igen rövid rutin helyettesíti, melyben a pontok koordinátáit DATA sorokból READ ciklus olvassa be.

A rutin átírható, a pontok x, y és z koordinátáit kell X(I), Y(I), Z(I) tömbökbe betöltenünk.

### 4. Élek megadása

A pontok összekötési sorrendjének megadását szolgálja. 0 számú pont megadása esetén az el nem rajzolódik meg (tollemlés). A programrészt itt is rövid DATA-READ ciklus helyettesíti.

A modul átírása esetén az összekötött pontok sorszámát kell megfelelő sorrendben az E(J) tömbbe betöltenünk.

### 5. Perspektív ábra számítása és rajzolása

A rutin a térbeli koordinátatranszformá-

ció alapadatait, a képernyő méretének megfelelő nagytítás, vízszintes és függőleges eltolás nagyságát számítja, majd az egyes koordináták transzformációja után a képernyőre vagy plotterre történő rajzolás következik.

### 6. Adatkivétel és adatbevitel

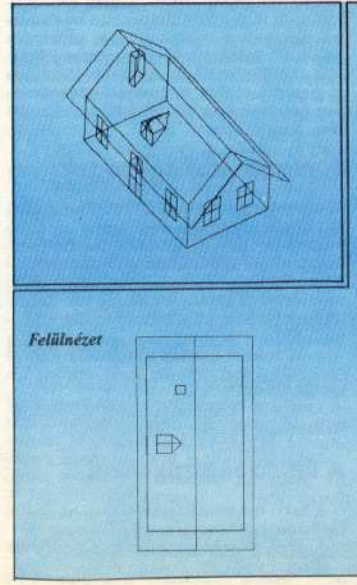
A pontok koordinátáit, a pontok összekötési sorrendjét, esetleg a perspektív alapadatokat, perspektív koordinátákat, perspektív ábrát háttértárolóra mentő és betöltő modul helye.

A program átírható és továbbfejleszthető. Működésének megértését és továbbfejlesztését az alábbi szakirodalom tanulmányozása segítheti:

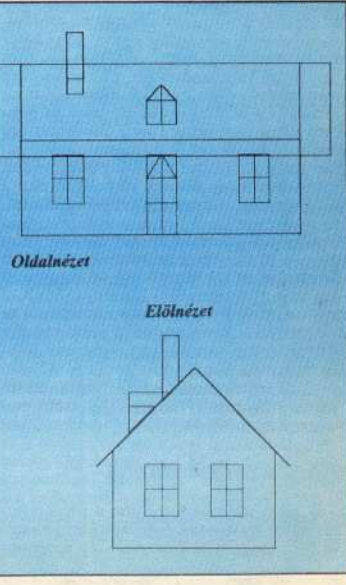
1. W. M. Newman—R. F. Sproull: Interaktív számítógépes grafika. Műszaki Könyvkiadó, 1985.
2. W. Heft: Einführung in CAD mit dem COMMODORE 64. Data Becker Gmbh, 1985.
3. M. Weber: 3—D—Grafik in Theorie und Praxis. IWT—Verlag, 1984.
4. J. Plate: Abbildung dreidimensionaler Objekte. MC 1986/2.
5. G. Pomaska: Alles darf man nicht zeigen. Chip, 1984/3.
6. G. Pezzi: Studio di Prospettive. MCmicrocomputer n. 31 giugno, 1984.
7. F. Danielowski—A. Pretzsch: Architekturperspektive. VEB Verlag, Berlin, 1969.
8. E. Gull: Perspektivlehre. Verlag für Architektur, Erlenbach—Zürich, 1946.

FEKETE ZOLTÁN

3. ábra. Távoli nézőpont felvételével eredményül kapott axonometrikus rajz



4. ábra. Nézetrajzok előállítására egyirányponthos perspektív rendszer és távoli nézőpont felvételével



— Távoli nézőpont esetében a perspektív torzulás egyre kisebb (fényképezéssel teobjektív), igen távoli nézőpont esetében nézetrajzokat, illetve axonometrikus rajzokat kapunk.

A főpont koordinátáinak megadása:

— A főpont megadásával a nézési irány

Az alkalmazásokban először a központi telepítésű számítógépekkel közeli vagy távoli — például postai távbeszélő-hálózaton keresztül — vonalkapcsolatban álló, többé vagy kevésbé intelligens terminálok rendszere terjedt el. Ebben a terminálhálózatban az egyes munkahelyekről a felhasználók közösen használták a központi telepítésű, nagy teljesítményű feldolgozó számítógép erőforrásait, tárolt adatállományait, programjait, feldolgozó teljesítményét stb. Az ilyen rendszerekben a terminálok és a központi gép telepítési távolsága több tíz kilométer is lehet, a köztük levő adatátvitel sebességét részben a vonalkapcsolat „minősége” (fizikai jellemzői), részben az adatátviteli készülékek (modemek) paraméterei határozták meg; általában 300–9600 bit/s. Ez a megoldás még nem eredményezett „valódi” számítógépes hálózatot.

A szoros értelmében vett számítógépes hálózatban a hálózat egyes gépei, a csomóponti számítógépek — lehetnek azonos vagy különböző típusúak — önálló erőforráskészlettel és intelligenciával rendelkeznek (memória, tárolt információ, önálló feldolgozó teljesítmény stb.), egymás között önállóan fogalmazhatnak (programok távolról történő futtatása, adatsere stb.), és hozzáférhetnek egymás erőforrásaihoz is.

Az ipar, a tudomány és a gazdaság tevékenységi körében számos esetben van szükség olyan rendszerre, amely viszonylag szűk földrajzi környezetben, például egyetlen vállalatban belül teremt információs kapcsolatot és/vagy végez meghatározott célú feldolgozásokat. Ezek az adottságok, azaz a kis távolság, a jól körülhatárolt intézményi feladatok, valamint a speciális igények: a nagy adatátviteli sebesség és a megbízhatóság sajátos műszaki megoldásokat kívánnak, amelyeket az ún. helyi hálózatok valósítanak meg.

A helyi hálózatokról kb. 15 éve jelentek meg az első publikációk. A „helyi” jelző elsősorban a hálózat egyes gépeinek és perifériájának viszonylag közeli, általában egy épületen belüli telepítésére utal. A megoldás főleg a rendszer gazdaságossága miatt vált népszerűvé, aminek elsődleges oka a hálózateltérés ötletes műszaki megvalósítása.

A helyi hálózatok fő alkotóelemei:

- a 10 km-nél általában nem nagyobb távolságot átfogó adatátviteli közeg (koaxiális vagy sodrott érpárú kábel),
- a számítógépes munkaállomásokat és a központi, nagy kapacitású erőforrásokat (lemezegység, nyomtató) rendszerbe kapcsoló serverállomás,
- az egyes munka- és serverállomásokat az adatátviteli közegre rákapcsoló csatlóegység (LAN board).

## Rendszerjellemzők

A távoli erőforráskapcsolat a helyi hálózat egyik legfontosabb jellemzője. Lehetővé teszi, hogy a hálózat adat- és program-

fájljait a hálózat központi telepítésű lemezegységen tárolják (fájl server funkció), amihez a hálózat minden munkaállomása a hálózaton keresztül hozzáférhet. A megoldás növeli a tárolt adatok védelmének hatékonyságát is. Hasonló erőforrás-megosztást tesz lehetővé a „print server” funkció is, melynél a hálózat munkaállomásai közösen használhatják a helyi hálózat központi nyomtatóját.

A nagy sebességű adatkapcsolat a helyi hálózat állomásai számára lehetővé teszi a hatékony és gyors adatátvitelt a nagy sebességű adatátviteli csatornában, például koaxiális kábelben keresztül (1–10 Mbps).

A hálózat egyes gépeire telepített szolgáltatások hozzáférhetővé tehetők a hálózat többi gépe számára is. Egyes feldolgozások során szükség lehet például a feldolgozandó adatoknak a hálózaton belüli megkeresésére és a feldolgozást végző gépre való átvitelére, vagy a feldolgozás eredményének a hálózati nyomtatón való kinyomtatására.

A terhelések hálózaton belüli átcsoportosíthatósága lehetővé teszi a pillanatnyilag túlterhelődött hálózati számítógép időszakos tehermentesítését, egyes funkciók átirányítását a hálózat más gépeire. Ennek vizsgálatánál természetesen figyelembe kell venni olyan szempontokat is, hogy egy feladatátcsopontosítás mennyire terheli túl a többi funkciót „átvállaló” számítógépet. Az adatátviteli csatorna terhelése például egy meghatározott terhelésen túl már nem növelhető az ezzel együttjáró gyakori adatsomag-ütközések miatt, így adott esetben fontolóra kell venni a szóban forgó funkcióknak ideiglenes letiltását.

A feldolgozó teljesítmény megosztására olyan esetekben kerül sor, amikor egy nagy feldolgozó teljesítményt igénylő feldolgozást csak több részfeldolgozásra bontva lehet végrehajtani a hálózat egyes gépein.

A helyi hálózatok további fontos jellemzője az állomások és az átviteli hálózat igen jó megbízhatósága, melyet jelentős mértékben befolyásol az átviteli hálózat elrendezése, a topológia.

## Funkcionális egységek

A tulajdonképpeni feldolgozásokat végző számítógépek: fájl server állomás (osztott hozzáférést biztosító lemezelrendszert); printer server állomás (osztott hozzáférést biztosító nyomtató rendszer); gateway server (a lokális hálózatot más számítógépes hálózatokkal összekapcsoló számítógép).

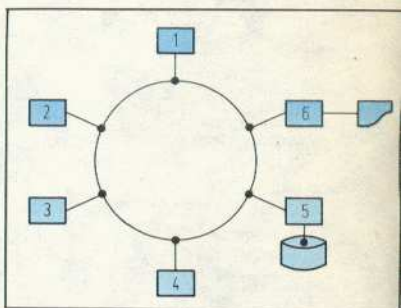
A lokális hálózat adatátviteli közege (koaxiális kábel, sodrott érpárú, optikai kábel stb.).

Egyéb kommunikációs hálózatkapcsolat (telefon, telex, videotex stb.).

## A fejlődés szakaszai

A helyi hálózatok fejlődésében az alábbi funkcionális fázisok figyelhetők meg:

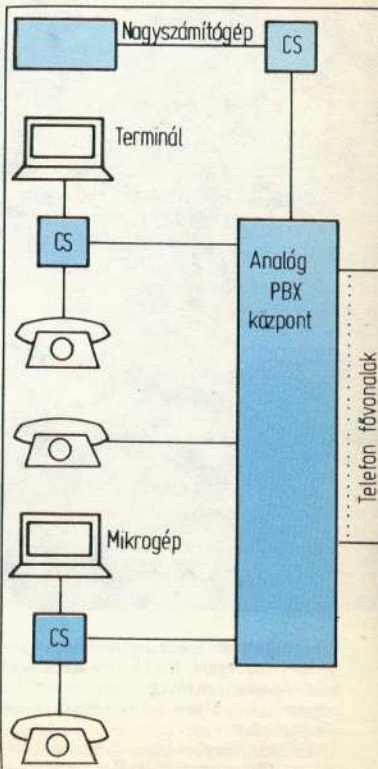
1. fázis. A számítógépek és a perifériák



1. ábra. Az általános célú helyi hálózat elvi vázlata

- 1–4: Munkaállomások (lehet terminál, PC stb.)  
 5: Osztott hozzáférést biztosító lemezelrendszert  
 6: Osztott hozzáférést biztosító nyomtató-alrendszer

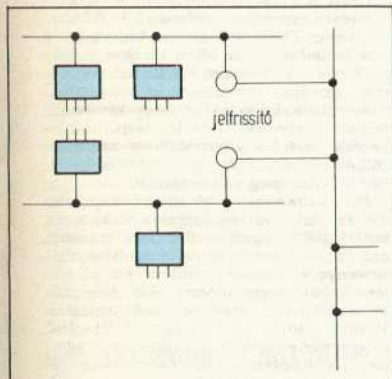
2. ábra. Analóg üzemmódú PBX helyi hálózat elvi vázlata



közi adatátviteli teljesítmény növelése — például koaxiális kábel rendszer alkalmazása, szemben a hagyományos rendszerben alkalmazott modem átvitellel. Ennek eredményeként a csatorna átviteli sebesség felső határa 9600 bps-ről 1–10 Mbps-ra növekedett.

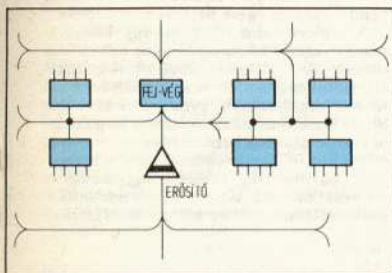
2. fázis. Az adatátviteli hálózatra csatlakoztatható egységek (PC-k) választékának rugalmas bővíthetősége: újabb gépek bekapcsolása a hálózatba, viszonylag egyszerű módon.

3. fázis. Funkciómegosztás a hálózati struktúrán belül, hálózatok közötti kommunikáció (gateway) megvalósítása.



3. ábra. Alapsávú helyi hálózat elvi vázlata

4. ábra. Széles sávú helyi hálózat elvi vázlata



4. fázis. Az adatátvitellel párhuzamosan megvalósított kép- és hangátvitel (széles sávú rendszerek). Komplex szolgáltatást nyújtó integrált információs rendszerek létrehozása.

## Adatátviteli közeg

A helyi hálózatok adatátviteli közege (kábel) lehet sodrott érpáru, mellyel általában néhányszor tíz állomás kapcsolható

össze maximum 1–2 Mbps sebességű rendszerre. Ez a közeg érzékeny a zajra, áthallásra, beruházási költsége viszont az összes megoldás között a legalacsonyabbak.

Koaxiális kábellel már akár több száz állomás is összekapcsolható, maximum 10 Mbps-os sebességű rendszerre. Koaxiális kábellel működő helyi hálózat lehet alapsávú (baseband), mellyel egyidőben csak egyetlen csatornán létesíthető átvitel, vagy széles sávú (broadband), mellyel egyidőben több csatornán keresztül is létesíthető átvitel. Az adatátvitellel egyidőben hang- és képátvitelre is van mód.

A jövő átviteli közege az optikai kábel, melynek fő előnye a kis súly és huzalvezetési tulajdonságai, a kiváló zaj- és áthallásvédelem, a nagy átviteli sávszélesség. Ilyen kábellel jelenleg csak néhányszor tíz állomás kapcsolható össze, maximum 10 Mbps-os rendszerre, kb. 1 km-es távolságon belül. A megoldás szélesebb körben való elterjedését is meg gátolja az optikai kábel meglehetősen magas ára.

## Hálózattípusok

Az általános célú helyi hálózatok (1. ábra) a legerjedtebbek. Nagy központi számítógépeket, mini- és mikrogepeket, képernyős terminálokat és egyéb perifériákat kapcsolnak össze velük, rendszerint sínyvagy fastruktúrájú hálózatba, koaxiális vagy sodrott érpáru kábel segítségével. Átviteli sebességük 1–10 Mbps körüli. A hálózatba kapcsolható készülékek száma több száz lehet. Adat, hang és képi információ is továbbítható.

A nagy sebességű helyi hálózatot rendszerint nagy számítógéppontokban alkalmazják a központi számítógép és a nagy sebességű perifériák közötti minél gyorsabb adatátvitel megvalósítására. Az ilyen rendszerek átviteli sebessége akár 50 Mbps is lehet, az összekapcsolható eszközök száma csekély.

A fenti két hálózat csomagkapcsolt típusú.

A telefonhálózaton keresztül vonalkapcsolásos választással működő helyi hálózatokat a szakemberek PBX (Public Branch Exchange) néven ismerik. A hálózat az intézményeknél már meglévő helyi telefonhálózat és telefonközpontokra épül, rajta beszéd és adat egyaránt továbbítható.

## PBX hálózatok

1983-ban az USA-ban az ilyen elven működő rendszereknek több mint 300-féle változatát tartották nyilván. A változatok nagy számát az magyarázza, hogy ezek a rendszerek a már meglévő és igen sokféle megoldás továbbfejlesztésekként alakultak ki. A leggyakoribb típusokat néhány szóval bemutatjuk.

Az analóg hálózatokban (2. ábra) az átvitel modemmel valósítják meg, a telefonhá-

lózathoz olyan egységen keresztül csatlakoztatják a munkahelyeket, amely biztosítja a telefonkészülék csatlakoztatását is. A csatlakoztató egységen kapcsolóval választható a beszéd (telefon) vagy az adatátviteli (terminál) üzemmód, egyidőben ugyanis vagy csak beszéd, vagy csak adatátvitel hajtható végre.

A hibrid hálózatokban a beszéd- és az adatkapcsolatot egymástól független alközpontok vezérlik. A készülékeknek a hálózathoz való csatlakoztatását itt is egy csatlakoztató egység biztosítja, mely a beszédátvitelt alapsávon (300 Hz–4 kHz), az adatátvitelt a 36–40, illetve a 72–80 kHz sáv tartományban valósítja meg. Annak ellenére, hogy az adat- és a beszédátvitel közös érpáron történik, a kétféle átvitel független egymástól, tehát egyidőben is megvalósítható.

Az analóg vonallal digitális központhoz kapcsolódó hálózatokban a központ maga digitális elven működik, vonalkapcsolatait analóg-digitális átalakítók biztosítják. Ez a rendszer az analógnál megbízhatóbb, kapcsolási sebessége nagyobb, mérete kisebbek, költségei alacsonyabbak.

A digitális vonallal digitális központhoz kapcsolódó hálózatok a telefonkészülékben digitalizálják (ködölik) a beszédjeleket, majd ezt viszik át a hálózaton keresztül a digitális központra.

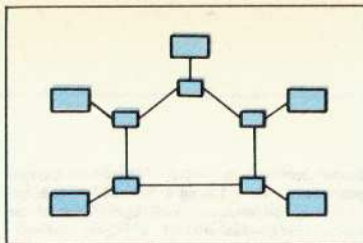
Az egyedi PBX hálózatok kapcsolóközpontja két szektorra oszlik, s ezek vagy csak telefon-, vagy csak adatkapcsolatot valósítanak meg. Vonalai közül egy a telefonösszeköttetésre, kettő az adatösszeköttetésre szolgál. A telefon és az adatkapcsolat egymástól függetlenül működik. Ez a típus a leglassabb az összes között, ezért csupán szerényebb igények kielégítésére alkalmas. Az átvitt üzenetek hossza tetszőleges, a kapcsolatfelépítés lassú, ami egyúttal meghatározza alkalmazási körét is. Beruházási költségigénye — tekintettel a már minden intézménynél meglévő helyi telefonhálózatra — viszonylag alacsony.

## Működési frekvencia

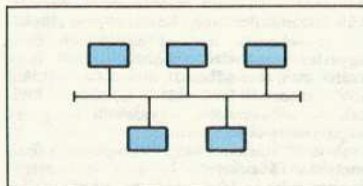
A helyi hálózatok működésük frekvenciatartománya szerint lehetnek alapsávbán működő (baseband) vagy frekvenciaátbelyezéssel működő, ún. széles sávú (broadband) rendszerek.

Az alapsávú rendszerekkel (3. ábra) 1 km körüli távolság hidalható át. Az átviteli közeg 50 ohmos koaxiális kábel, amely kevésbé érzékeny a csatlakozások okozta reflexiókra és a zajra. Az átvitel sebessége 10 Mbps, a csatlakoztató munkahelyek száma kb. 100. Alacsonyabb sebességű, olcsóbb hálózatokban átviteli közeget használható a sodrott érpáru kábel is. Az átviteli sebesség ekkor csak kb. 1 Mbps, a rendszerbe kapcsolható állomások száma néhányszor tíz, zajérzékenysége viszont alig rosszabb a koaxiális kábellel használt rendszerekénél.

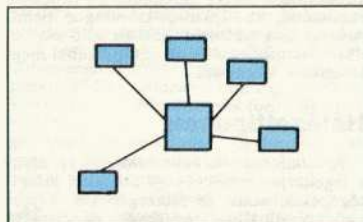
A széles sávú hálózatok (4. ábra) analóg jelek frekvenciaosztásos átvitelével működ-



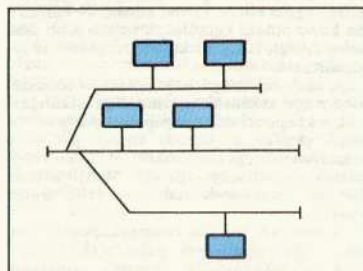
5. ábra. Gyűrűs elrendezésű helyi hálózat elvi vázlatja



6. ábra. Sín elrendezésű helyi hálózat elvi vázlatja



7. ábra. Csillag elrendezésű helyi hálózat elvi vázlatja



8. ábra. Fa elrendezésű helyi hálózat elvi vázlatja

nek, rajtuk az adatátvitellel egyidőben kép és hang is továbbítható. Rendszerint sínvagy fastruktúrájúak, az átviteli egy- vagy kétirányú, az áthidalható távolság több tíz km, a rendszerbe kapcsolható állomások száma akár több száz is lehet.

Kétirányú átvitelnél az egyik megoldás szerint egy adási és egy vételi adatútra van szükség, melyek a hálózat egy pontján, az ún. „fejvég”-en (head-end) csatlakoznak egymáshoz. A másik megoldásnál csak egy adatút van, itt viszont az adási, illetve vételi irányú vivőfrekvenciák különböznek egymástól. Az ilyen megoldás „fejvég”-e frekvenciaváltaként működik. A legismertebb rendszerekben az adási frekvenciasáv 5–116 MHz, a vételi 168–300 MHz tartományba esik.

Ha egy széles sávú rendszert egynél több csatornára készítenek, a többi csatornán az adatátvitellel párhuzamosan kép- (tv) és hangátvitel is megvalósítható.

## Hálózati elrendezések

Hálózati elrendezésen a hálózat állomásainak egymással való összekapcsolási sémáját értik. A hálózati elrendezés befolyással van a hálózat paramétereire, például a megbízhatóságra.

A gyűrűs hálózat (5. ábra) igen érzékeny az átviteli közeg hibájára, mivel rajta az állomások egymás után, sorosan vannak fel-fűzve, ezért ha valahol a kábel megsérül — megszakad vagy zárlatos lesz —, a teljes hálózat működésképtelenné válik. Az ilyen típusú hálózat úgy működik, hogy az üzenetküldő állomás ráteszi a gyűrűre forgalmazandó üzenetét, a sorban öt követő állomás megvizsgálja, hogy az üzenet neki szól-e, és ha igen, beolvassa annak tartalmát. Ha nem neki szól, továbbítja az üzenetet a következő állomásnak. Az üzenet adott irányban halad végig a gyűrűn mindaddig, amíg meg nem találja a címzett állomást, amely kiolvassa az üzenet tartalmát, és a vételt nyugtázza beállítja és visszaküldi a „sikeresen kiolvastva” jelzést a forgalmazó állomásnak. Léteznek olyan rendszerek is, melyekben a gyűrűben egyszerre több üzenet is keringhet.

A sín (busz) hálózatban (6. ábra) egyszer-

re csak egy üzenet továbbítható. A hálózat működtethető központi vezérléssel is, ilyenkor a forrásállomás által küldött adatcsomag először a központba továbbítódik, majd a fogadóállomáshoz. Az egyidőben forgalmazni kívánó forrásállomások „versenyeznek” a hálózathasználat jogáért.

A hálózathasználati jogát többféleképpen lehet biztosítani: központilag, amikor a központi állomás egy-egy időtartamra egymás utáni adási joggal ruházza fel a hálózat állomásait (polling technika), és ha az adásra feljogosított állomás forgalmazni akar, a számára kijelölt időtartamra adásra lefoglalhatja a sávot; időosztásos módszerrel (TDMA = Time Division Multiple Access) és frekvenciaosztásos módszerrel (FDMA = Frequency Division Multiple Access).

A sín megszerzése iránti kérelem indítása, illetve a megszerzett sín használatba vétele történhet tetszőleges időpillanatban (random) is. Ez azzal jár, hogy az egyidőben kezdeményezett forgalmazások „ütközhetnek”, aminek következtében az „ütköző” üzenet elvesz, és egy későbbi időpontban az adást meg kell ismétlni.

Ha a sínrre nagyon sok állomás kapcsolódik és ezek gyakran forgalmaznak, a sín túlterhelődik, ugrásszerűen megszaporodnak az „ütközések”, emiatt a rendszer teljesítménye erőteljesen csökken. Ezt elkerülendő, különleges módszereket dolgoztak ki az ütközések számának csökkentésére. Ilyenek a sínforgalomba való „befigyelés” (forgalmaz-e valaki már a sínen) és az adásismétlés időzítésének adott algoritmus szerinti vezérlése (például ütközés esetén az első ismétlés  $t$  idő múlva, a másodikra  $1,5 t$  múlva stb. kerül sor).

A sínelrendezésű hálózat egyszerűen bővíthető, és az egyes állomások hibája nem bontja meg az egész hálózat működését.

A csillaghálózat (7. ábra) egy központi hálózatvezérlő állomásból és az ennek alárendelt munkaállomásokból áll. A munkaállomások a központi vezérlőállomáson keresztül forgalmaznak egymással, a hívó és a hívott közötti adatkapcsolatot a forgalmazni kívánó állomás kezdeményezésére a központ építi fel. Az elrendezés előnye az egyszerű bővíthetőség, továbbá hogy az egyes munkaállomások kiesése nem teszi működésképtelenné a teljes hálózatot. Hátránya

1. táblázat

Jellemző	Rugalmasság	Bővíthetőség	Megbízhatóság	Interfész bonyolultsága	Költsége
Sín	nagy	nagy	nagy	közepes	alacsony
Gyűrű	közepes	közepes	nagy	alacsony	közepes
Csillag	alacsony	nagy	közepes	alacsony	nagy
Fa	nagy	nagy	nagy	közepes	alacsony

2. táblázat

Jellemző	Csatornaszám	Karbantartás	Állomásszám	Kábeltávolság	Zavarvédelem
Sodrott érpár	egyetlen (adat/hang)	nehézes	kevés	néhány mérföld	gyenge
Koaxiális alapsáv	egyetlen (adat)	igen egyszerű	sok	néhány mérföld	közepes
széles sáv	több száz (adat, hang, kép)	könnyű	nagyon sok	40 mérföld	kiváló
Optikai kábel	több száz	nagyon nehéz	kevés	korlátlan	kiváló

## COMMODORE 64

# Sztringváltozóban elhelyezett összefüggést kiértékelő szubrutin

A program a kiértékelendő kifejezést tagokra, a tagokat tényezőkre bontja. Kiértékeli a tényezőket, elvégzi a szorzási, osztási műveleteket — azaz kiértékeli a tagokat —, és azokat az E változóba összegzi.

A tag és tényező fogalmakon a következők értendők. *Tagok* a kifejezés azon részei, melyek "+" illetve "-" műveleti jelek között helyezkednek el. A tagok "+" és "/" műveleti jeleket, általunk definiált függvényjeleket és számkonstansokat tartalmazhatnak.

*Tényezők* a tagok azon részei, melyek "\*" illetve "/" műveleti jelek között találhatók. A tényezők számkonstansokat és általunk definiált függvényjeleket tartalmazhatják.

### Fő részei

- 100—191 Magyarózó feliratok képernyőre
- 200 Értékelés D\$ sztringnek, majd sztring lezárása "+" karakterrel
- 216 Pointerek, flagek lenullázása, D\$ hosszának vizsgálata, tárolása H változóban
- 220 Ha a H=1 feltétel igaz, akkor csak a RETURN billentyű kódját — chr\$(13) — tartalmazza a D\$ sztring. Az eredmény definíciószerűen =0. Az 530-as sornál kezdődő programrész ezt kijelzi
- 230—520 A D\$ sztringben tárolt kifejezés tagokra, tényezőkre, függvényekre bontása. A kifejezés kiértékelése
- 530—550 Eredmény kijelzése
- 560—570 Várakozás. RUN-nal változók törlése, program újraindítása
- 600—660 Hibáüzenet megjelenítése a képernyő alon részén.

A kiértékelendő kifejezés a D\$ sztring-

viszont, hogy a hálózatvezérlő állomás hibája az egész hálózat működését megbénítja.

A *fa-hálózat* csillagelrendezéseket kapcsol össze hierarchikus felépítésű és vezérlésű hálózat (8. ábra). A struktúra előnye egyszerű bővíthetősége, hátránya viszont, hogy a magasabb hierarchiaszintekhez tartozó útvonalak kiesésével önálló hálózatokká esik szét, továbbá, hogy egyes adatutak túlterheltsége (bedugulás) esetén a hozzá tartozó hálózati részekről elküldött üzenetek késleltetése megnő.

A leggyakrabban használt hálózati elrendezések összehasonlítása az 1. táblázaton látható.

A felsoroltakon kívül léteznek további hálózati elrendezések is — főleg kombinált megoldások —, ezek azonban különböző okok miatt kevésbé terjedtek el.

Fel kell hívni a figyelmet arra, hogy a hálózat topológiája és az alkalmazott kábeltípus nem választható meg egymástól függetlenül. A sintopológiához például sodrott érpáru és koaxiális kábel is használható, míg a fastruktúrájú hálózatokhoz inkább szelvényes sávú koaxiális kábelt, a gyűrűs hálózatokhoz sodrott érpáru, alapsávú koaxiális vagy optikai kábelt, csillaghálózatokhoz leginkább sodrott érpárt alkalmaznak.

A különböző adatátviteli közegek hatását a velük összekapcsolt rendszerekre a 2. táblázat szemlélteti.

## Hálózatelérési módszerek

A helyi hálózatok másik lényeges jellemzője a hálózatok való hozzáférés elve, azaz, hogy milyen módszer szerint érke el az egyes munkaállomások a csatornát, és hogyan forgalmazzanak azon.

A hálózat-hozzáférést befolyásolja a csatornakapacitás (Mbps), a közösen használt átviteli kábel, a címfigyelés (a hálózati állomások érzékelik a csatorna forgalmát, de csak a címzett állomás olvassa le róla az üzenetet) és a hálózati elrendezés (topológia).

Centralizált vezérlésű gyűrűs hálózat esetén a következő hálózat-hozzáférési módszerek terjedtek el: vezérelt jelzéses (token passing), réselj gyűrűs (slotted ring), pufferveszűrős (puffer insertion).

A *vezérelt jelzéses hálózatban* az egyes állomások egymás után sorosan kapcsolódnak a hálózatra, ily módon alakítva ki a gyűrűs hálózati struktúrát. A hálózati kábel lehet sodrott érpár vagy koaxiális. A hálózatban a jelzés (token) állomásról állomásra körbejár.

Ha egyik állomás sem kíván forgalmazni, a hálózat nyugalomban van, és a szabad vezérlőjelbe (token) egyik állomás se tesz bele csomagot (üzenetet). Ha valamely állomás forgalmazni kíván, megvárja, amíg a szabad vezérlőjel hozzá ér, majd fel foglálja (foglaltsági jelzés beállítása), beleszítja üzenetét (adat, feladó és címzett címe), és a hálózatra teszi. A jel haladási iránya szerint

következő állomás érzékeli a „foglalt” jelzést és benne az üzenetet, leolvassa annak címzését és összehasonlítja a sajátjával, ha megegyezik, kiolvassa az adatrészt is, és beállítja a „sikeres forgalmazás”-t jelző nyugta bitet, és továbbengedi azt a gyűrűben. Ezt a nyugta üzenetet a forrásállomás „veszti”, és a „sikeres átvitel” jelzésből tudja, hogy forgalmazása sikeres volt, így azt nem kell megismételni. Mindig a forrásállomás távolítja el a hálózatból a feleslegessé vált nyugta üzenetét és tesz helyette egy szabad jelzést a hálózatba, mellyel elsőként a gyűrűben rá következő állomás forgalmazhat, ha kíván. Ha nem, úgy a szabad jelzés — és ezzel a forgalmazás joga — az őt követő állomásra száll át.

A vezérelt jelzéses gyűrűs hálózatban megvalósítható a többszintű prioritás, azaz a magasabb prioritású állomások felfüggeszthetik az alacsony prioritású állomások adáshoz jutási jogát.

Az állomás addig nem generál új adási jogot, ameddig saját csomagját nyugta formájában vissza nem kapta. Ezt a rendszert nevezik egyetlen adási jogú rendszernek, és e megoldással a rendszer megbízhatósága nagymértékben növelhető, ha a forgalmazni szándékozó állomás előbb meggyőződik a hálózat hibátlanágáról, és csak azután kezd el adni.

A *réselj gyűrűs hálózatelérés* esetében a gyűrűben előírt hosszúságú üzenet befogadására szolgáló „rések” keringenek, amelyek lehetnek vagy „üres” vagy „tele” rések; ezt egy állapotbit jelzi a résen belül. Az az állomás, amely forgalmazni kíván, megvárja, hogy egy rés hozzá érkezen, majd megvizsgálja, hogy üres-e a rés, és ha igen, „beleteszi” adott hosszúságúra tördelt üzenetét. A most már üzenettel feltöltött rés címzését az állomások sorban megvizsgálják, nekik szól-e az üzenet. Ha igen, kiolvassák annak tartalmát, beállítják a sikeres vételt nyugtázó bitet, és továbbengedik a kiolvastott, de még nem üres részt a csomagot feladó állomás felé. A rés kiürítése (szabaddá tétele) a csomagot feladó állomás feladata. A kiürített részt ezt követően a sorrendben következő állomásnak kell felajánlani, ezzel megakadályozva, hogy egy állomás monopolizálja az adási jogot, kizárva más állomásokat a forgalmazás lehetőségéből. A hálózatban egyidőben keringetett részek számát a hálózat saját késleltetése korlátozza.

A *pufferveszűrős hálózatelérés* változó hosszúságú üzenetek átvitelét teszi lehetővé. Az üzeneteket átmeneti tárolóban tárolják. Az egyes munkaállomások interfészei az üzenetek átmeneti tárolására szolgáló puffereket tartalmaznak. Az adó állomás a gyűrűn keringő két üzenet közé szúrja be a sajátját, erre az időre a második üzenet a puffereben tárolódik. Az eljárás segítségével az átlagos üzenetátviteli késleltetés csökkenthető. A sikeresen vett üzeneteket a vévő oldal „távolítja” el a hálózatból.

CSEH KÁLMÁN

ben található. D\$ stringnek a 200-as sorban szereplő INPUT utasítás ad értéket. Az értékdás után D\$ stringhez hozzáadunk egy "+" karaktert, mely az utolsó tag végét jelzi a 260-as sor számára.

A 230—520-as sorok a működési elvnek megfelelően épülnek fel. A 230—300 szubrutin végzi a tagok leválasztását. Megkeresi M-től H (D\$ hossza)-ig a +, illetve a - előjeleket, ezt s2-ben, majd s1-ben tárolja. A keresés úgy történik, hogy D\$ stringet karakterenként vizsgálja.

A tagot a D1\$ stringben tárolja, majd hívja a 320—440-es sorokban található szubrutint, amely a D1\$-ben tárolt tag kiértékelését végzi. H2 a D1\$ hossza. A D1\$-ben tárolt tagot ez a szubrutin nem karakterenként vizsgálja, hanem kihasználja a val() függvény tulajdonságát, vagyis azt, hogy a string számkarakternek megfelelő részét adja át valós számnak E1-nek.

Ha E1 = val(D1\$) értékdás után E1 = 0 és h2 < > 0, akkor D1\$ kifejezés nem számkarakterrel kezdődik, tehát meg kell vizsgálni, hogy a kezdő karakter egyezik-e valamelyik általunk definiált függvényjellel.

Ezt a 460—501-es sorokban található szubrutin végzi, de hívása előtt Q\$ string-változóba helyezük a vizsgálandó karaktert, L3 és L2 pedig azt tárolja, hogy a D1\$ stringben mi a következő kiértékelendő tényező pozíciója.

A függvénykiértékelést a 460—520-as programrész végzi. Itt az általunk definiált függvények argumentuma csak számkonstans lehet, mivel nincs a programban olyan szubrutin, amely elvonzná a függvényargumentumok kiértékelését.

## Példák

### 1. A kifejezés:

- 3.3 \* s12 + 23.5 \* s25 \* t30 - 29.34

A 200-as sorban szereplő INPUT utasítás a fenti értéket adja D\$ stringnek. Ehhez (D\$ = D\$ + " + ") egy karaktert hozzáadunk. Így az általunk begépelte kifejezés alakja D\$-ben:

- 3.3 \* s12 + 23.5 \* s25 \* t30 - 29.34 +

1. 2. 3.

A kifejezés 3 tagból áll. Az első tag 2 tényezőt tartalmaz: egy konstans és egy kiértékelendő függvényt (s12 = > sin(12)). A második tag 3 tényezőt tartalmaz: egy konstans és két kiértékelendő függvényt s25 = > sin(25), t30 = > tan(30). A harmadik tag egy számkonstans.

A programlistából kitűnik, hogy a szögfüggvények definiálása oly módon történt, hogy a kifejezés megadásakor beirt szögfüggvények argumentumai szögfokokat jelentenek. Mivel a szögfüggvények definícióhoz az alap BASIC szögfüggvényeit használtam, és mint ismeretes, e függvényargumentumokat radiánként értelmezi az interpreter, a fokértékeket radiánra kellett átszámítani. Ezt a 465-ös sorban a P2 változó tartalmazza.

### 2. A kifejezés: 2\*3\*6.2

A 2\*3\*6.2 kifejezés D\$-be kerül, majd a "+" jel hozzáadása után a 2\*3\*6.2+ alakot kapja. Itt a 2\*3\*6.2 kiértékelendő kifejezés egyetlen tagból áll, a tagnak 3 db konstans tényezője van.

A program a 470—500-as sorok megváltoztatásával alakítható át egyedi igények esetén. Természetesen más, az itt bemutatottal összetettebb függvények is definiálhatók.

## Új függvénydefiniálás szabályai

1. A definiált függvények jele egy karakter hosszúságú lehet, és nem használható numerikus karakter.

2. Egyetlen definiálandó függvényt sem jelölhetünk már felhasznált függvényjellel.

3. Ha a definiált függvény argumentumát szögfokokban kívánjuk megadni, és ez az argumentum a kiértékelés során egyben szögfüggvény argumentuma is, akkor a szögfüggvénynek P2-vel kell számolnia, egyéb függvényargumentumoknak P3-mal. Definiálható olyan összetett függvény is, amely P2-vel és P3-mal egyaránt számol.

## Példa függvénydefiniálásra

501 ifq\$="A" then p2 = p312 \* P1 / 4

A definiált függvényre "A30" formában hivatkozhatunk, és egy 30 egység átmérőjű kör területét számítja ki.

Ha az 501-es sort például a következőképpen definiáljuk:

501 ifq\$="A" then p2 = p312 \* P1 / 4 \* 150, akkor az "A30" hivatkozás egy 150 egység magasságú, 30 egység átmérőjű henger térfogatát számítja ki.

Végül megemlítjük az összefüggés-kiértékelő szubrutin hiányosságait: a definiált függvények argumentuma csak számkonstans lehet; a kifejezésnek megadásakor zárójelek nem használhatók; "kiakadás" elleni védelme nem százszázalékos. Természetesen további programrészletek megírásával e hiányosságok megszüntethetők.

SHELL FERENC

```
100 V=53288-POKEV,6:POKEV+1,6:PRINTCHR$(142)CHR$(8)
110 PRINT "A30 HAZSHALHARTO CHR$(13) AJELEK."
120 PRINT "XXXXXXXXXXXXXXXXXXXX => OSSZEADAS"
130 PRINT "XXXXXXXXXXXXXXXXXXXX => KI VOHNAS"
140 PRINT " * => SZORZAS"
150 PRINT " / => OSZTAS"
160 PRINT " S => SIN PL: S25=>SIN(25)"
170 PRINT " C => COS PL: C32=>COS(32)"
180 PRINT " T => TAN PL: T67=>TAN(67)"
190 PRINT " L => LN PL: L76=>LN(76)"
191 REM *** BOVITES ***
192
200 INPUT "ADJA OSSZEFUGGES" : D$ = D$ + "+"
215
220 H=LEN(D$) : L1=0 : L2=0 : L3=0 : H1=0 : H2=0 : H3=0
225 IF H=1 THEN D$="" : GOT0530
230 E=0 : S1=1 : M=1 : L1=1 : ILEFT$(D$,1)="" : THEN S1=-1 : L1=2 : M=2
235 ILEFT$(D$,1)="" : THEN L1=2 : M=2
240 FOR I=1 TO H : A$=MID$(D$,I,1)
250 IF A$="" : THEN S2=-1 : GOT0280
260 IF A$="" : THEN S2=1 : GOT0280
270 GOT0900
275
280 M1=1 : D1$=MID$(D$,L1,M1-L1) : GOSUBS20
290 E=E+S1*E1 : S1=S2 : L1=M1+1
295
300 NEXT
305
310 GOT0530
315
320 H2=LEN(D1$) : E1=VAL(D1$) : L2=LEN(STR$(E1)) : L3=L2 : Z=0
330 IF H2=0 THEN G600 : REM * HIBA UZENET *
340 IF E1=0 THEN G5=LEFT$(D1$,L3) : L3=L2 : L2=1 : GOSUB460 : E1=P2 : L3=L2
350 Z=1
360 IFL2=1=H2 THEN RETURN
365 IFL3=H2 THEN G600
370 P2=VAL(RIGHT$(D1$,H2-L3))
380 IFF2=0 THEN G5=MID$(D1$,L3+Z,1) : GOSUB460 : Z=1 : GOT0400
390 L2=LEN(STR$(P2))
400 A$=MID$(D1$,L3,1)
410 IFA$="" : THEN RND$(0) / " THEN G600 : REM * HIBA UZENET *
420 IFA$="" : THEN E1=E1#P2 : GOT0440
430 E1=E1/P2
440 L3=L3+L2 : IFL3=1=H2 THEN RETURN
445 GOT0370
450 GOT0600
455
460 IF H2=L3-Z THEN G600
465 P2=VAL(RIGHT$(D1$,H2-L3-Z)) : L2=LEN(STR$(P2))+1 : P3=P2 : P2=P2#r/100
470 IF0$="S" THEN P2=SIN(P2) : GOT0520
480 IF0$="C" THEN P2=COS(P2) : GOT0520
490 IF0$="T" THEN P2=TAN(P2) : GOT0520
500 IF0$="L" THEN P2=LOG(P2) : GOT0520
501 REM *** BOVITES ***
502
510 GOT0600 : REM * HIBA UZENET *
520 RETURN
525
530 IFE=0 THEN H2$=RIGHT$(STR$(E) : LEN(STR$(E))-1) : GOT0550
540 D$=STR$(E)
550 PRINT "ADJA EREDMENY : D$"
560 GETA$ : IFA$="" : THEN H560
570 RUN
600 REM ** HIBA **
610 H$="** HIBAS OSSZEFUGGES ! " : GOSUB580
620 GETA$ : IFA$="" : THEN H620
630 RUN
650 POKE214,22:POKE211,1 : SYS58732
660 PRINTLEFT$(H$+" ,39) RETURN
```

# Egyszerűbben assemblerül

Lebecsülte Lindeisz László a Floppykezelés Kernal-rutinok segítségével c. hozzáférése alapján a HELP+ és a PROFIMAT fordító lehetőségeit (1986/3. szám). A Frey Judit azonos című cikkéhez (1985/4. szám) kapcsolódó írásában megkérdőjelezi a címcímek alsó és felső bájttát értelmező „használatos tevé” operandus létezését. Ezt a többi között azzal támasztja alá, hogy csupán az Advanced Machine Code Programming for the Commodore 64 (Granada Technical Books) című könyvben talál utalást a MIKRO—64 assemblerre, amely szerinte csak *talán* ismeri a fenti operandust.

En szintén Frey Judit cikke kapcsán rajzoltam el a problémához. A HELP+ ról szóló leírást Erdős Iván: Commodore 64 assemblé című könyvében találtam meg, a PROFIMAT-féle alkalmazást pedig próbálgatással okoskodtam ki, ahogyan az alábbi két példa is mutatja.

Mindkét példaprogram nyolc csillagot rajzol a képernyőre. Az LI szubrutin használatához az UZ címe alsó bájttát az A regiszterbe, a felső bájttát pedig az Y regiszterbe kell tenni. A .BYTE direktíva decimális számot épít be a tárgyprogramba, a .TEXT pedig a karakterek ASCII kódjait. A szöveg végét zérus bájttal jelenti. Az első példaprogram a PROFIMAT, a második a HELP+ fordítóját használja. Ily módon nem okozhat gondot a SETNAM rutin használata sem, mivel mindkét assembler megengedi a címkék értékének (címenek) kivonását, sőt összeadását is. Ezért nagyon egyszerűen kiszámíthatjuk például a fájlnev hosszát.

Igy használhatók eredeti formájukban — kis változtatással — Frey Judit részprogramjai, és a programunk is relokálható lesz.

DOSZTÁNYI ZSOLT

```

C000          100      ##C000
C000 4C 0C 03 110    JMP KEZU
MB1E         120 LI   =MB1E
C003 2H 2H 2H 130 UZ .TEXT "####"
C007 2H 2H 2H 140   .TEXT "####"
C00B 00         120   .BYTE 00
C00C 03 03     160 KEZU LDM #UC
C00E 00 00     170   LDY #UZ
C010 20 1E 0E 180   JSR LI
C013 00         190   RTS
C014          200     .END

ZEILEN:11  SYMBOLE:3  FEHLEK:0
KEZU #C00C LI #MB1E UZ #C003
    
```

1. példa

2. példa

```

MELUP#H
2 240 C000          .UPT #3,U0
120 C000          ## #C000
130 C000 4C 0C 03 JMP KEZU
140 C003          = #MB1E
150 C003 2H 2H 2H .BYT42,42
160 C003 2H 2H 2H .BYT42,42
170 C007 2H 2H 2H .BYT42,42
180 C009 2H 2H 2H .BYT42,42
190 C00B 00       .BYT00
200 C00E 03 03    LDM #C0Z
210 C00E 00 00    LDY #C0Z
220 C010 20 1E 0E JSR LI
230 C013 00       RTS
    
```

```

100 : INPUT+ UTASITRS C-64-REI
100 : BY KEINDL LASZLO
100 : (K) 1986.03.06. TATABANYAI
100 :
150 ##C000
160 BASVEC=#0300 :776 RENDSZERMUTAO
170 CHROET=#F73 :CHRGOT RUTIN (115)
180 CHRROT=#F79 :CHRGOT RUTIN
190 RINPUT=#ABBF :REOI INPUT RUTIN
200 RINTER=#A7E7 :INTERPRETER
210 PLUSZ=#AA :(> JEL KOOJA
220 INTER=#A7AE :INTERPRETER
230 INPUT=#85 :INPUT TOKES
240 LDR #VIZSGO :CIBETOLTES
250 LDY #VIZSGO
260 STA BASVEC :A RENDSZER-
270 STA BASVEC+1 :MUTATOER
280 RTS
290 NOP
300 NOP
310 NOP
320 VIZSGO JSR CHRGOT
330 CMP #INPUT
340 BEQ TALAL
350 JSR CHRGOT
360 JMP RINTER
370 TALAL JSR CHRGOT
380 CMP #PLUSZ
390 BEQ RUTIN
400 JSR RINPUT
410 JMP INTER
420 RUTIN JSR #B79B
430 CPX #F19
440 BCC FOLYT
450 HIBA JMP #B248
460 FOLYT STX #FB
470 JSR #E200
480 CPX #F2B
490 BCS HIBA
500 TXA
510 TAY
520 LDX #FB
530 JSR #E50A
540 JSR CHRGOT
550 JSR RINPUT
560 JMP INTER
570 .END

ZEILEN:48  SYMBOLE:13  FEHLEK:0
BASVEC=#0300  CHROET=#073  CHRGOT=#079
FOLYT=#C032  HIBA=#C02F  INPUT=#085
INTER=#A7AE  PLUSZ=#00AA  RINPUT=#ABBF
RINTER=#A7E7  RUTIN=#C028  TALAL=#C01B
VIZSGO=#C00E
    
```

```

BASVEC=#0300  CHROET=#073  CHRGOT=#079
FOLYT=#C032  HIBA=#C02F  INPUT=#085
INTER=#A7AE  PLUSZ=#00AA  RINPUT=#ABBF
RINTER=#A7E7  RUTIN=#C028  TALAL=#C01B
VIZSGO=#C00E
    
```

1. lista

# Egy új BASIC utasítás

Ez a gépi szubrutin az adatbevitelt könnyíti meg a képernyőszervezésnél. A C64-ből hiányzik az INPUT utasításnak az a fajta megoldása, hogy a képernyő tetszőleges sorába és oszlopába vihetők be az adatok. Ez csak a fénypontvezérlő karakterekkel lehetséges, vagy pedig bonyolult módon összePOKE-olva a gépet. Ezen a problémán segít a program, amely az INPUT+ utasítás használatát teszi lehetővé. (1. lista)

Az utasítás szintaxisa:  
INPUT+ sor, oszlop, "szöveg"; A ahol a szöveg jelentése ugyanaz, mint az eredeti INPUT utasításnál, és "A" lehet sztring vagy numerikus adat. Az utasítás aktivizálása a program beírása után: SYS 49152. Ezek után az utasítás mindaddig érvényben marad, amíg a gépet ki nem kapcsoljuk.

Természetesen az INPUT+ mellett a régi INPUT is használható. Azok számára, akiknek nincs monitoruk, közlöm a 2. listát, a BASIC betöltőprogramot.

KEINDL LÁSZLÓ

```

10 FOR I=49152TO49226
20 REPOJ $=S+J
30 POKE I, J
40 NEXT I
50 IFS=0:8937THEN65
60 SYS49152:NEW
65 PRINT"HIRA AZ ADATOKBAN!" :END
70 DATA 169, 14, 160, 192, 141, 8, 3
71 DATA 140, 9, 3, 96, 234, 234, 234
72 DATA 32, 115, 0, 201, 133, 240, 6
73 DATA 32, 121, 0, 76, 231, 167, 32
74 DATA 115, 0, 201, 170, 240, 6, 32
75 DATA 191, 171, 76, 174, 167, 32, 155
76 DATA 183, 224, 25, 144, 3, 76, 72
77 DATA 179, 134, 251, 32, 0, 226, 224
78 DATA 48, 176, 244, 138, 168, 166, 251
79 DATA 32, 10, 229, 32, 115, 0, 32
80 DATA 191, 171, 76, 174, 167
    
```

2. lista

## COMMODORE 64

# GET# helyett INPUT#

Ismeretes, hogy a Commodore 64 INPUT# utasítása maximálisan 88 (más forrás szerint 80) karaktert tud egyszerre beolvasni. Ennél hosszabb rekord beolvasására a szakirodalom a GET# utasítás vagy gépi kódú szubrutin használatát javasolja.

A GET# utasítással történő beolvasás legnagyobb hátránya, hogy rendkívül időigényes, és így az adatfeldolgozást jelentős mértékben lelassítja. Ennek kiküszöbölésére egyfajta olyan, BASIC nyelven írt programrész mutatunk be, amely 88 karakternél hosszabb, példánkban a maximális 254 (ebből „hasznos” 251) karakterből álló rekordot INPUT# utasítással olvas be. A példát egy, a legkényelmesebben alkalmazható relatív fájlra alakítottuk ki.

Az egyszerűség kedvéért változóként egy 10 elemű, 251 karakterből, 0–9-ig terjedő számokból álló tömböt alkalmaztunk (64 \$ (1)).

A programrész működésének lényege,

hogy az egyes tömbelemeket három változóra (V1\$, V2\$, V3\$) bontjuk, ezeket rögzítjük, majd az adatokat ugyanilyen formában olvassuk vissza. A rögzítéskor és a beolvasáskor a relatív fájl rekordjain belül a pointer szintén azonos értékekre állítjuk be. Megjegyezzük, hogy már az adatbevitelkor célszerű a „:” és a „;”-t leltítani, mert ezeket a karaktereket a gép utasítás-ként értelmezi. Ennek megfelelően ajánlatos a mezőket más jellel elválasztani.

A program 450-es sorában található IF utasítás azt a célt szolgálja, hogy üres sztring esetén a gép ne adjon ki STRING TOO LONG hibajelzést, ezért már a bevitelnél minden egyes rekordba egy „0” jelet tettünk be.

Ugyanazt a rekordot GET#, illetve INPUT# utasítással beolvassuk, az első esetben a művelet 14, a másodikban 0,6 másodpercig tart; ez már jelentős mértékű megtakarítás. Megjegyezzük, hogy célszerű új

vagy újrafarmázzott adatmezeivel dolgozni, mert a lemez és a meghajtó kapacitásának végső határán vagyunk. Más különben STRING TOO LONG hibajelzést kapunk.

A felhasználás során természetesen a változókban belül tetszőleges hosszúságú mezők alakíthatók ki. 251-nél rövidebb rekord esetén pedig annak csak első és/vagy második részét kell beolvasni, illetve rögzíteni. Sőt a példában alkalmazott egyetlen változót (GY \$ (I)) helyett lehet több változót is használni. Ilyenkor csak arra kell vigyázni, hogy rögzítéskor a változókat CHR\$

(13)-mal válasszuk el, még a pointer újrabéállítását elől is. Például:

```
300 P=1: GOSUB 600: REM REKORD
ÉS POINTER ÁLLÍTÁS
310 PRINT# 5, V I $; CHR$ (13); ...; V
4 $; CHR$ (13);
320 GOSUB 540: P=88: GOSUB 600:
REM REKORD ÉS POINTER ÁLLÍ-
TÁS
330 PRINT# 5, V 5 $; CHR$ (13); ...; V
9 $
340 stb. ...
```

DR. KIRILLY ANDRÁS

## COMMODORE 64

### Beviteli rutin

```
0 REM * BEVITELI RUTIN BEVITELES *
1 IFFEK(49153)=2327EN10
2 0=49151 FORI=1 TO 2: READ: POKE6+1,X:V=V+X
3 NEXT I: V=C%3618+THEMPRINT "KOMODORA HIBA"
4 251 DDATA 251,174,32,156,180,150,70,32,252
5 DDATA 174,32,159,180,168,24,32,240,255
6 DATA 253,174,32,164,179,56
9 REM * BEVITELI PROGRAM *
10 PRINT "KOMODORA": POKE53260,15: POKE53261,15
15 PRINT "BEVITELI RUTIN BEVITELŐ"
16 PRINT "*****"
20 INPUT "HANGVÁLLÁS: SZER: " : A$
25 INPUT "OSZLOP: " : B$
30 INPUT "KÉZKÖR. BEVITELI": S$
31 IFS=C% THEN S35
32 INPUT "TÍZHEZESK: " : T$
33 INPUT "KÉZTŐL KARR. " : J$
35 INPUT "ÁRBEVEZÉS, BEVITELI " : R$
40 INPUT "NEGYVEZES " : N$
45 INPUT "MAX. KARBETTER " : M$
50 PRINT "C"
70 GOSUB 6000
200 REK = FELDOLGOZD PROGRAM *
205 PRINT "KOMODORADATBANK" : B$
210 IFS=C% THEN C%
220 PRINT "KOMODORADAT BEVITELI SZÖVEGES VALTOZO" : GOTO500
250 B=VAL(B$) : N=VAL(N$) : PRINT "KOMODORADAT" : N$
260 PRINT "KOMODORADAT SZER NEVZESE"
300 END
5000 REM * INPUT RUTIN *
6000 I: H=LEN(H$) : C%00=0
6002 IF D%280C0R=C%+2+H%400R=N: THEN C%00=0
6003 SVS=64: B=1824, C%0, D, H$ : "I" : POKE5325,25
6004 SVS=640: C%0, C% : "I" : POKE5325,25
6005 POKE214, D : POKE211, C% : SVS=640: POKE204, 0
6006 IFR=(C%00>>)>H% THEN C%00=0
6007 IFFEK(207) THEN C%00=15
6008 POKE204, 1
6009 H=H%00+C%00
6025 IFR=13AND5=" " THEN GOSUB 60000
6030 IFR=13 THEN GOSUB 60000
6040 IFR=2 THEN GOSUB 60001
6042 IFR="S" AND VAL(H%0) = 0 AND C%0=46:
6043 AND(C%0="0" AND C%0="") THEN B$=" " : GOTO 60000
6044 IFR=(C%00>>)>H% THEN C%00=0
6045 IFR="R" THEN GOTO 60001
6046 IFR="R" THEN C%0=15
6047 IFR="R" THEN C%0=15
6048 IFR="R" THEN C%0=15
6049 IFR="R" THEN C%0=15
6050 POKE214, D : POKE211, C% : SVS=640: POKE204, 0
6052 GOTO 60000
6050 IFFEK(207) THEN C%00=0
6055 POKE204, 1: FORI=1 TO 2000: NEXT I
6056 POKE5325, 68: RETURN
6060 PRINT "HIBA: PARAMETERESEN KÖR"
60610 IFR="R" THEN C%0=0
6062 IFR="R" THEN C%0=0
6063 IFR="R" THEN C%0=0
6064 PRINT "KOMODORA HIBA" : RETURN
6065 IFR="R" THEN C%0=0
6066 IFR="R" THEN C%0=0
6067 IFR="R" THEN C%0=0
6068 IFR="R" THEN C%0=0
6069 IFR="R" THEN C%0=0
6070 POKE5325, 68: END
6080 IFR="R" THEN C%0=0
6085 TU=10: I$=""
6090 B=VAL(B$) : C%0=INT(TU/8+5)+TU
6091 B=INT(TU/8+5)+TU
6092 B=INT(TU/8+5)+TU
6093 B=INT(TU/8+5)+TU
6094 IFR="R" THEN C%0=0
6095 IFR="R" THEN C%0=0
6096 IFR="R" THEN C%0=0
6097 IFR="R" THEN C%0=0
6098 IFR="R" THEN C%0=0
6099 IFR="R" THEN C%0=0
6100 IFR="R" THEN C%0=0
```

Az UNIN rutint olvasva (1985/6. szám), előtöltségelviselése, bonyolultsága és változóparazálisa miatt kezdem hozzá egy beviteli rutin megírásához. A program kurzorkezelő, bevitel ellenőrzést és formázott kiíratást is tartalmaz. Egy jól ismert gépi rutint használok, amely megtalálható Lángos István: A Commodore 64 mikrogepi kezelése és programozása című könyvében, és aminek talán új oldalát sikerült megjelentetnem. A gépi rutin bevitelére a program egyetlen DATA listát adok; haladóbbrak az assembly lista alapján készíthetnek maguknak programfájlt. Talán meglepő, hogy a nem megfelelő beviteli török; ez a biztonságra törekvésnek köszönhető. Az esetleg elrontott billentyűzést a "font" billentyűvel lehet javítani.

Természetesen a program továbbfejleszthető, ezért is adom közre. Például az elkészült oldalak ellenőrizhető a rutin kisméretű módosításával: a kurzor ugorjon mindig a következő adatra.

A használt numerikus változók: A,B,C,C0,D,HU,M,MK,T,TU,X,Y), a szöveges változók: A\$,B\$,BU\$,K\$,M\$,R\$,S\$.

Felhívom a figyelmet arra, hogy a 60042 és a 60043 sor egy sor alkot, csak nyomdatechnikai okokból vannak különválasztva.

MOCSÁRI ATTILA



# COMMODORE 64

## UNIN példaprogram

A mintapélda remélhetőleg bebizonyítja, hogy milyen hatékony, gyors programozást tesz lehetővé a rutin használata (1985/6. szám). A példa ugyan töredéket mutatja be a rutin adta lehetőségekről, de így is jól érzékelteti a paraméterezhetőség előnyeit.

A minta max. 10 ember adatait képes felvenni. Egy ember adatait egy képernyőn vihetjük fel vagy módosíthatjuk, s lapozással juthatunk egy másik egyén kartonjára. Bármelyik kartonon bármit módosíthatunk. A hibázenet kezelésére is láthatunk példát.

```

10 REM*****
20 REM****
30 REM**** UNIVERSALIS ****
40 REM**** INPUT ****
50 REM**** PELDHA ****
60 REM****
70 REM**** KESZITETTE BORBELY ****
80 REM**** JOZSEF ****
90 REM****
95 REM****
98 REM*****
99 GOSUB100:GOTO1000
100 REM-----
110 REM-----
120 REM---- 1 0 0 - 5 4 5 - SUROK ----
130 REM-----
140 REM---- INPUT ----
150 REM---- RUTIN ----
160 REM-----
170 REM-----
545 REM-----
1000 P*(1)="00021220":P*(2)="09041611"
1005 P*(3)="00061620":P*(4)="07081805"
1010 P*(5)="00101620":P*(6)="00121615"
1015 P*(7)="48142696"
1020 PRINT"JTBTA(13)"PELDA PROGRAMM"
1023 PRINT 1. NEV " "
1027 PRINT 2. SZEM.SZAM " "
1030 PRINT 3. SZUL.HELY " "
1035 PRINT 4. SULYVALTOZMS : KG00"
1037 PRINT 5. MUNKAHELY " "
1040 PRINT 6. BESZTASA " "
1045 PRINT 7. UTOLS MERES DAIUMA :000"
1050 PRINTTAB(6)"MODOSIT7":
1052 PRINTTAB(20)"MELYIK SORT?0":
1055 PRINTTAB(8)"LAPOZ? + - NUM"
1057 PRINTTAB(9)"VEGE?":
1060 CO=7: B6=7: B8=0: B0=0: B7=0
1065 FOR J=1 TO 7:GOSUB1100:NEXT
1070 L=17: C=15: B1=0:GOSUB535: J6="Y"
1075 GOSUB1200: IFR#< "Y" THEN1100
1080 L=17: C=33: B1=9: J6="1":GOSUB1200: J=R
1085 IFR<10R0:7 THENGOSUB1210:GOTO1000
1090 J6=I#(CLAP, J):GOSUB540: A=1
1095 GOSUB1100:GOTO1070
1100 L=19: C=15: J6="1":GOSUB1200
1110 IFR#< "1" ANDLAP<9 THENLAP=LAP+1
1120 IFR#< "1" ANDLAP>9 THENLAP=LAP-1
1130 IFR#< "N" THEN1160
1135 IFR#< "+" ANDR#< "-" THEN1100
1140 IFR#(CLAP, J)="1" THEN1020
1150 PRINT"000":FOR J=1 TO 7
1155 PRINTTABVAL(MID$(P*(J), 5, 2));
1157 PRINTTAB$(CLAP, J)"0":NEXT:GOTO1100
1160 L=21: J6="N":GOSUB1200
1165 IFR#< "1" THEN1070
1170 END
1180 B1=VAL(LEFT$(P*(J), 2))
1185 L=VAL(MID$(P*(J), 3, 2))
1190 B=VAL(MID$(P*(J), 5, 2))
1195 B=VAL(RIGHT$(P*(J), 2)):GOSUB100
1195 TAB$(CLAP, J)=A#:RETURN
1200 B7=7: A=1:GOSUB540:GOTO1000
1210 L=24: C=0: B7=0: B=0:GOSUB530
1215 PRINT "HIBAS VALASZ 1":GOSUB100
1217 PRINT " " :RETURN
1230 REM
1300 REM ALTLALBANI : X# MODOSITASA :
1310 REM
1320 REM B7=7 : GOSUB 535 : J6=X#
1325 REM GOSUB 545 : GOSUB 100 : X6=X#

```

A programokorok magyarázata:

- 99 Kezdeti feltételek beállítása, ugrás a főprogramra.
- 1000-1015 A bemeneti mezők egyedi paramétereit P\*( ) sztring tömbként vannak megadva, egy tömbelem egy mezőt határoz meg. Egy-egy tömbelemben 2 karakterenként egy paraméter értéke helyezkedik el. Így egy mezőhöz összesen 4-fajta egyéni tulajdonság kapcsolódhat, sorrendben: a mező típusa, képernyősor, oszlop, mezőhossz.
- 1020-1057 Egy karton állandó részeinek kiírása a képernyőre.
- 1060 Állandó paraméterek értékadása.
- 1065 Új karton mezőinek kitöltése.

1070

Módosítás? Ha nem, ugrás 1100-ra.

A módosítandó sor sorszámának bekérése és ellenőrzése. A megadott mező módosítása. A módosítandó sztring J6-on keresztül kerül A6-ba az 535-os és 540-es rutinok segítségével. Ugrás 1070-ra.

1100-1135 Lapoz előre (+), hátra (-) vagy ugrás 1100-re (N).

1140 Üres lap esetén új lap kitöltése: ugrás 1020-ra.

1150 Új lap kiírása, ugrás 1100-ra.

1160 Vége? Ha nem, ugrás 1070-re.

1170 Vége.

1180-1195 A módosítandó mező paramétereinek beállítása, módosítás. Hibakiöröz szubrutin.

BORBÉLY JÓZSEF

## ADOK - VESZEK - CSERÉLEK

Ebben a rovatunkban 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 címeire várjuk.

■ TRS 80 MC—10 sz. géphez 16 k-s bővítő, interfész-kábelek, könyvek, programok és 1 db COMPUTEK botkormány eladó vagy elcserélhető. Képző Róbert, Bp., Imre u. 5. 1093. Telefon: 378-075.

■ Sinclair Spectrum megkopott fedőlappját (fém) olcsón és gyorsan újjavarázolom. Képző Róbert, Bp., Imre u. 5. 1093. Telefon: 378-075.

■ Commodore 64 gépre eredeti OXFORD—PASCAL programnyelv doku-

mentációval eladó: Leskóné Puskás Judit, Bp., Aszú u. 25. 1028. Telefon: 830-443 vagy 298-750.

■ Commodore VC20 játék- és egyéb programokat cserélnék. Keresem a Sakk, Super Expander és bővítő leírást másolásra. Rába Kálmán, Szombathely, Bolyai János u. 13. F. IV. 19. 9700.

■ C64-hoz hard disket, Winchestert vennék. Programírást és oktatást vállalkoz minden szinten (BASIC, FORTH). Telefon: 154-352.

■ A Mikroszámítógép Magazin első 13 száma — csak egyben — eladó. Ára 400 forint. Postán utánvétellel elküldöm. Kovács László, Karcag, Zádor u. 8. 5300.

■ VC20 számítógép magnóval, botkormánnyal, jelentős mennyiségű programmal és dokumentációval együtt 19 000 forintért eladó. Setény János, Békéssámsón, Vöröskatonák u. 97. 5946.

**Fejlesztési Intézetünkbe  
felvételre keressük  
korszerű mikroprocesszoros terminálok  
és berendezések HARDWARE—SOFTWARE-  
fejlesztéséhez villamosmérnököket,  
készülék konstrukciós kialakításához  
felsőfokú végzettségű szakembereket.**

**RUGALMAS MUNKAI DŐ  
KIEMELT FIZETÉS  
MAGAS JÖVEDELEM**

Jelentkezés:

**TELEFONYÁR**

1143 Bp. Hungária krt. 126-132.

Tel.: 634-330.



(Móttó: A szabad értelem pró-  
bája nem az értékek tagadásá-  
ban rejlik. Hanem abban, hogy  
bölcsen tud választani az értékek  
közül. — A. L. Sacher)

# Barátunk, a számítógép

## Vélt és valós tendenciák

Ülök a televízió előtt. 12 éves, mosolygós, szemérvényes fücska áll valamilyen szerkesztő mellett. A riportert biztató kérdése: — Ugye, Pistike, te már valóságos zsonglőr vagy a programozásban? Pistike határozottan bólint. A riportert tovább évődik: — És ugye kisfiam, ugye apukád, ugye nem kezeli a számítógépet? Mire a gyerek kihúzza magát, és jön a meglepő válasz: — Nem, apuka nem tudja kezelni a számítógépet.

Döbbenet, hogy a józan hittel szemben milyen gyorsan terjednek a hamis mítoszok, amelyeknek egyike a boldogulni. Szomorú, hogy mennyire nem tudják látni a barátunk népszerűsítésére szolgáló műsorokat. Holott kitudott, hogy az újjal szembeni természetes ellenérzés csak igen nagy tapintattal lehet leküzdött.

Emiatt a félrekezelés miatt ragadtam írógépet. Szeretnék hitet adni a felnőtteknek, barátjukká tenni a számítógépet. A fiatalokat pedig realitásra kívánom inteni, mint amolyan középkorú. Vég-eredményben a vélt és valós tendenciákról lesz szó ebben a cikkben. Remélem, hogy sikerül majd tisztáznom egy alapkérdést: az eszközök szerepét életünkben. Nem fogok laposkáról, programokról, billentyűkről beszélni; majd eljön annak is az ideje. Ha már látjuk a lényegét, amit nem takarnak el a részletek.

több találmány szerzője. Nem az a legfontosabb ismérve, hogy nem tud számítógépet kezelni.

Más családban anya keze alatt egyszerre 15-20 hatalmas szövőgépre zakatol. A második műszak után kicsit fáradt ahhoz, hogy számítógépet tanuljon. Ahhoz különben is matematika kell, ami anyu-  
nak — a kosztépénz kivéve — nem erős oldala.

Tájékoztatónk félíg átgondolt. Jóságunk, de ügyetlen. Nem épít a réteggazdasági titkaira, amiket még középkorú szomszéd-  
nők is átérz. Így a kommunikáció az amúgy is értékes szó, és nem ér el másokhoz, hogy bővítsé barátunk táborát.

Kedves apukák és anyukák! Apa igen kis erőfeszítéssel saját maga kérhetné le az aktuális trendet, a fedezetszámítást, az áralkulációt. Úgy, hogy saját maga kezelne a gépet, de nem furcsán irt vezényszavakkal. Anya pedig mindig csodálatos érzékkel rendelkezett a színek-minták iránt. Géppel tervezte csodálatos szöveteket, amire mindig vágyott. Nem kellett sóhajtozva — de büszkén — le-  
gyintenie: majd a kisfiam...!

Higgyék el a kedves felnőttek, hogy barátunk kezelése nem nagyobb ügy — kellő felkészítés esetén! —, mint mondjuk az autóvezetés. Remélem, hogy e cikksorozat végére ezt így fogják látni. Valóban barátunk a számítógép!

helyen hangoztatják. Kicsit nehézkes, kicsit unalmas. Szóval: van más is. Remélem, hogy a cikksorozat végére belátod azt. Addig is megérezhetitek abban, hogy még e században is irdatlan sokat kell tanulnod ahhoz, hogy kis barátod-  
ból, a számítógépből kihozd, ami benne van.

### Gépekről, komolyan

Számítógép és számítógép között akkor a különbség van, mint jármű és jármű között. Nem megyek a szomszéd közértbe autóval, nem biciklizem Párizsig, és az óceán fölött repülni fogok. Be kellene már végre látni, hogy egy-egy szervezet komoly információs feladataira csak meghatározott képességű számítógépek adhatnak megoldást.

Ne csináljunk erényt a kényszerből! Ha nem telik repülőre, akkor autóval nem fogok áthatolni Amerikába! A személyi számítógépek adott célokra valók. Igaz, e fránya kis masinák egyre többet és többet tudnak. Már nem csak „személyi-  
ek”. De még nem pótolják a nagybácsit, az „igazi” számítógépet.

Utóbbiak kezelése nem mindennapos feladat, hanem hivatás. Pistike, aki ma kitűnően kezeli a számítógépet, holnap már alkalmas lesz arra a feladatra, ha kitart. Örülünk kell tehát minden végére és egyéb oktatásnak, amiben a mi Pistink részül. Bár a felnőttek is vannak olyan komolyan barátunkat, mint Pistike teszi!

Nem úgy, mint egyes hozzászólók. Felnőttek, akik semmit sem értenek. Idezék:

„Vettünk 12 Commodore 64-es gépet. Ezzel nálunk a elektronizáció befejeződött.” És az élet? Azóta a nyilatkozó karrierje fejeződött be. Mert a gép használatához rendszerekre, programokra van szükség — és sok-sok pénzre.

„A készletrendszert számítógépre ültetjük.” Annál a nagyvállalatnál annyiféle tétel mellett aligha lesz elegendő az a kis IBM PC.

### A fiatalokhoz

Kedves, ifjú szomszédom komoly számítógépet kaptam. Nem jártékot, amin két cowboy ülőz három bölényt, miközben őket vipe-  
rők kergetik. Édesanyja elenizti a durvaságot terjesztő „játékokat”. A kis szomszéd lelkesen készíti programokat BASIC nyelven, és szentül hisz a kiadott jelszóban: a számítógép fejleszti a fiatalok feladatmegoldási képességét és gondolkodási készségét.

Nem dicserlek szembe, csak hátad mögött mondom: az igazság pont fordított. Mert van akarat-  
erő, van feladatmegoldási készség és képesség, még ma is ki-  
tartasz újabb barátod mellett. Nem úgy, mint Feri, akinél a gép státuszszimbólum, és aki az előregyár-  
tott program ötdik futása után unatkozott. Apja nem vett újabbat, mert nem is olyan olcsó. Ő meg nem irt programot, mert az fárad-  
ságos. A gép porosodik: nem lett barát.

Ifjú szomszédom! Örülök siker-  
reidnek, de tanulj meg egyet: nem jelenti képességeid csúcsát, hogy tudsz BASIC-ben programozni. A BASIC nem a jövő század kom-  
munikációs nyelve, mint számos

Nem kívánok itt ellenpropagandát kifejteni. Csak a felhangozt akaro-  
lom letérni. Apa, az ő fizetésével csak ne programozz! Anya pedig ne tanuljon matematikát, ha már eddig nem tette!

A személyi számítógépek forgal-  
mazásának hatalmas lendülete az utóbbi két évben megtört. Elsősor-  
ban azért, mert egyre többen ismer-  
ték fel, hogy ez a gép nem olcsó játékszer. A hivatástudó fiatalok számai maradnak meg hűsé-  
gen a barát mellett, de nem minden-  
kiből lesz programozó, aki számít-  
ógépet látott-kezelt. Mi tehát a helyes út?

Közéleti anyka születésnapja. Pisti újdonsággal akarja őt meglepni. Bár már érmet is nyert programozási versenyen, most valami nem megy, nem megy... És Pisti ekkor apuhoz megy és kérdez: — Beszélj magyarul, fiam! — inti apja. Nem utasít el, nem mondja: neked vettem. Apa figyel és meg-  
ráz. És a magyarázatok végére Pistikéből István lesz: megérti, hogy nem elágazás, ciklus, RUN és CASE a lényeg. A feladat a kemény dió, amit apu játszva feltört, pedig nem is tudja kezelni a számítógépet. Pistike megtanulta, hogy az új seprő jól seper, de a régi tudja, hogy hol van a piszok. Pistiből most már bizonyosan jó szakember lesz.

Anya a konyhában matat: hová is tettem a bejgli receptjét? Pisti büszkén mondja: kidobtam. Nem, ne haragudj. Itt van, csak kezdél a számítógépet! — No de fiam, ne bolondozz már! — néz vissza ag-  
gódvá anya a konyha felé. — Anya, csak ezt a két gombot nyomd le! Úgy! Most a kérdőjel után irt ad, hogy bejgli! — utasítja Pisti. Anya vonakodva megteszi, amit okos fia mond. Hátrahőköl, amikor felperog a nyomtató, és du-  
rusuló hangon kezdi: A mákos-bejgli receptje. Végy...

Szép volt a születésnap ajándék. Emberként is és programozó-  
ként is éretté vált Pisti. Olyanná, aki képes fel fogni mások valódi in-  
formációs igényét. Aki barátot akar szerezni az ő barátjának az-  
zal, hogy nem kényszeríti szokatlan trükkökre. Hanem úgy készíti fel a találkozást, hogy az mindkét félben, de főleg az emberben ked-  
ves és maradóan — hasznos? — emlék legyen. — Lám, anya mate-  
matika nélkül is tudja kezelni a gépet! — nézett össze apa és fia.

Az egész ilyen egyszerű. Kell, aki feladatot lát. Kell, aki megoldást kínál. Már rutin a megvalósítás. Kell sok boldog, géparát anyuka, és szükség van az igényt megvalósító tehetséges ifjakra. Kezdetnek csak ennyit akartam mondani. DR. HALASSY BÉLA

### A felnőttekhez

APA a hosszú tárgyalások után fáradtan ül a tévé előtt. Ha a most jó krimi kezdődne, akkor apa (lelkis-  
merete) kiszólna a konyhába anyához: — Fiam, hagyd azt most! Jó a műsor!

De most a tévében éppen arról van szó, hogy a bal felső sarokban írjunk egy furcsa angol szót. Felki-  
áltójel, pontosvessző, F6 billen-  
tyű... Apa dörmög valamit, és át-  
kapcsol a focimeccsre.



APA vállalati vezető. Naponta dönt emberekről, gépekről, milliókról. Aranygyűrűs diplomás,



### A kulcskérdés

Van-e szüksége egy ekkoraka országnak arra, hogy tízezrek tudjanak BASIC-ben programozni?

# Az Építsünk számítógépet! sorozat egy hibájáról

Az 1985/1. és 2. számban közölt video-NYÁK rajzában „hibákat” találtam, ezeket foglalkozom.

Az EPROM (IC 41) kiválasztása a PIA (IC 33) PB0 adatvonalával történik. Ha az IC 23A 7. lábán „1” szint van, akkor az EPROM van kiválasztva. Nos, ez így nem igaz, mert az IC 23A 7. lába le van kötve a negatívra. Ugyanakkor ez a vezeték egy, de ha jobban megfigyeljük, két 3K ellenállással a pozitívra van kötve. Miért?

Az IC 41 alatt található a grafika kiválasztása. Nem értem, hogy mi az, ami az IC 30, 29-es bemeneteire kapcsolódik?

Az IC 46 3. lába (bemenet) az IC 17 9. lábára (bemenet) van kötve. Miért? Ugyanakkor az IC 17 kimenetei nincsenek felhasználva!

Számomra még mindig érthetetlen, hogy az IC 46-tal miért végezteti el még egyszer azokat a feladatokat, amiket az IC 22 már elvégez, és ezen IC-k 7, 12, 9-es lábait összeköti. Miért?

A video RAM-ok címzésénél két eset lehetséges: 1. a 6845 címezi a RAM-ot; 2. a processzor címezi. Szerintem a 6845 nem képes az 1 k tárterületeket kiválasztani, mivel az IC 20 bemeneteire közvetlenül a processzor A0, A1 címvonalai kerül. Ezzel szemben a leírásban VA10, 11-es vonalak szerepelnek. Így a 6845 már képes a RAM-csoportok kiválasztására, mivel az IC 26 bemenetei pozitívan vannak. Ebből követ-

kezik, hogy az M2-es vezeték logikai „0” állapotban van, így az MX-K IC 21, 22, 23 a 6845 címvonalait kapcsolják a memóriákra. Itt megint felmerül egy probléma. Most a processzor nem képes a RAM-ok címzésére! Ennek ui. az a feltétele, hogy az M2-es vonal „1” szinten legyen, ez pedig a jelenlegi kapcsolásban nem jöhet létre tartósan. Az IC 26 valamelyik bemenetének „0” állapotba kell kerülnie. Ez egy pillanatra meg is történik, amikor a 6845 A10, 11, 12 címvonalai „1” állapotban vannak. Ekkor az M2-es vonal „1” szintű. Az IC 46 3-as bemenetét az IC 20 3-as bemenetére kapcsolja. Amennyiben az IC 46 3-as bemenete „0”, akkor az M2-es vonal ismét „0” szintű, azaz a 6845 címezi a memóriát. Talán ez a pillanatnyi állapot elegendő arra, hogy a processzor a megfelelő memóriarekeszt átírja? Ha igen, az IC 46 3-as lábát a címbusz A12-es vezetékére kell kötni, hogy a processzor is a teljes tárterülettel tudjon dolgozni!

Kérem, amennyiben a hibák igaznak bizonyulnak, közöljék velem. Ha lehet, a Magazinban. Javasolnám a komplett kapcsolási rajz közlését.

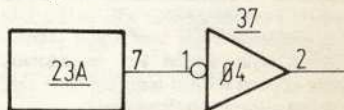
CSÁNYI CSABA

## A szerkesztő válaszol

A videokártya egy nem grafikus kártya módosításával készült. A módosítás miatt átkötések, egymásra építések és visszamaradt

— már felesleges — vezetékek is találhatóak. A lap 1985/2. számában például a 2. ábrán D7 az IC 29 14. bemenetén kívül egy szám nélküli volt alkatrész 20. lábára — pontosabban a foglalatra — is rá van csatlakoztatva. Ez után az általános megjegyzés után konkrétan válaszolok a kérdésekre.

1. EPROM-kiválasztás. Az 1985/2. számban a 2. ábrának hiányzik az IC 23A 7. láb után egy részlete:



2. Grafika kiválasztás. Erre az általános megjegyzés a magyarázat.

3. Memóriacímzés. Az 1985/1. szám 4. ábrájáról lemaradt: az IC 46 3. lábára és az IC 17 9. lábára (bemenetek) az A12 jel kerül. Az általános megjegyzés itt is érvényes.

4. Az IC 46 feladata. A módosítás többlet-RAM beépítését és ennek címzését jelentette.

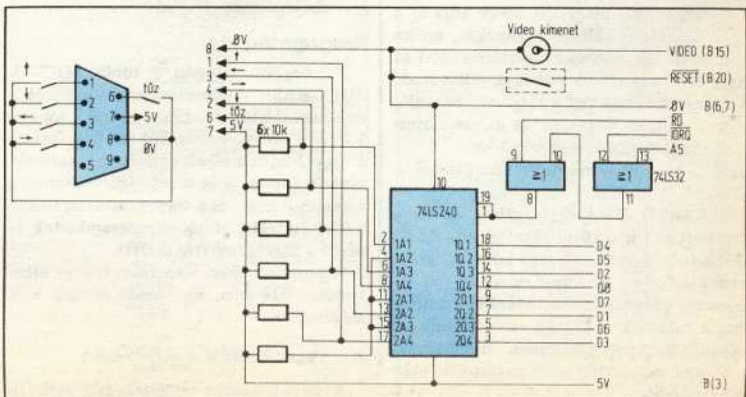
5. A video-RAM sávkiválasztása. Az 1985/1. szám 4. ábrájáról lemaradt: az IC 46 4. lábára és az IC 20 3. lábára (bemenetek) a VA12 jel kerül. Nem a rajz, hanem a szöveg a helyes. Így az IC 20 1. és 2. lábaira (bemenet) nem A0 és A1, hanem VA10 és VA11 kerül.

6. A teljes rajz közlésére jelenleg nincs mód.

## Lehet egyszerűbben is

Az 1986. március havi szám 28. oldalán megjelent ZX-Spectrum Kempstone botkormányillesztő kapcsolási rajzához szeretnék kiegészítést fűzni. Kétségtelen, hogy az áramkör működik, de szükségtelem bonyolult. Én egy olyat mutatok be, amely csak két IC-t (74LS240 és 74LS32) tartalmaz, szemben a három IC-t tartalmazó eredetivel, viszont ugyanazokat a funkciókat látja el.

PINTÉR GÁBOR  
Kiskunhalas



# PTA-4000 BASIC

Sorozatunkkal egyidőben a Rádiótechnika c. lap bemutatja a gép alap BASIC-jét. Itt ennek csak néhány, a gépi kódú programozásnál használt utasítását ismertetem.

## Néhány utasítás

Az utasítások formátuma és jelentése a következő.

### NEW kifejezés

Ezt a parancsot, mely programban is használható, a használati utasítás is ismereti, ha a kifejezés egy állandó, éspedig 0. Más érték is adható azonban. Ebben az esetben a kifejezés értéke adja meg a BASIC tárolóterületének kezdőcímét. Természetesen olyan értéket kell megadnunk, amely egyrészt a RAM-területen belül van, másrészt a foglalt területen kívül.

### STATUS kifejezés

A kifejezés értéke itt is korlátozott. Lehetőségei értékei: 0,1,2,3,4. A 0 érték esetén a szabad memória és az adatterület összegét adja vissza. Az 1 érték megadja a BASIC programterület nagyságát. A 2 a BASIC programterület végcímé utáni első címet, a 3 az adatterület kezdőcímét jelzi.

### PEEK\* kifejezés

Hasonlít az ismert függvényhez, de az ME1 memóriaterületen levő cím tartalmát szolgáltatja.

**POKE 1, 2, 3, 4** kifejezés, 2. kifejezés, 3. kifejezés...

Az első kifejezés a memóriacímre adja, a következő az oda tárolandó adatot, az ez után következő a következő memóriacímre tárolandót stb.

### CALL kifejezés, változó

Változó és az azt megelőző vessző nélkül a kifejezés által megadott címre adja át a vezérlést. Ha a változó numerikus, értéke átkerül az Xreg-be, és a visszatérés előtt az Xreg értéke átadódik ennek a változónak. Karakteres változónál a folyamat hasonló, csak a változó nagysága az akkumulátorban, kezdőcíme az Xreg-ben lesz.

**CSAVE M-1 „állománynev”;** 1. kifejezés, 2. kifejezés, 3. kifejezés

A CSAVE M-1 egymagában átadja a vezérlést az 1 jelű külső készüléknek. Az állománynev a védendő gépi kódú program nem kötelezően kiadandó neve; az első kifejezés a védendő memóriaterület kezdőcíme, a második a végcím, a harmadik az ugyancsak nem kötelezően használható programindítási cím, mely visszatöltés után aktivizálódik.

### CLOAD M-1 „állománynev”;

kifejezés  
A CLOAD M magát a beépített kazettás-egység-illesztőt használva tölt be egy programot, mégpedig a szalagon levő következő gépi kódút; az állománynev használata esetén pedig a megnevezettet. A-1 használata ugyanazt jelenti, mint a CSAVE esetén. A kifejezés azt a címet adja meg, ahová a program töltődni kezd. Ha a CSAVE utasításnál használtuk a harmadik kifejezést, akkor a program automatikusan indul.

## Kulcsszavak

A gép BASIC fordítóprogramját a MICROSOFT cég készítette, így a BASIC által használt szavak, az ún. kulcsszavak felismerése egy kódtáblázat tartalmával való összehasonlítással történik. Először a legtöbb MICROSOFT-BASIC-től, itt minden kód 2 bájttal hosszúságú. A használt kulcsszavakat és kódjukat az 1. táblázat foglalja össze.

## Adattárolás

Az adatokat 8-8 bájtól tárolja a fordító, függetlenül az adat típusától. Mégis megkülönböztet háromféle adatot. Ezek közül csak a decimális szám tárolásánál használja ki a fenntartott 8 bájt.

Decimális számok tárolásánál az első bájtól a kitevő, a következő az előjel, a következő 5 bájtól a mantissa, az utolsón mindig 00H található. Néhány változatot bemutatok a 2. táblázatban. Ugyanitt található példa a bináris szám tárolására is. Ezeknél az első négy bájtól a fordító nem veszi figyelembe, a következő mindig B2H, az ezután következő két bájtól található a binárisan tárolt szám, az utolsót ugyancsak nem értékeli a fordító.

Karakter sorozat tárolásánál (3. táblázat) az első négy bájt szintén közömbös, a következő értéke mindig D0H, az ezután következő két bájt a karakter sorozat tárolásának kezdőcíme, az utolsó pedig a sorozat hossza, binárisan tárolva.

## Programtárolás

A program tárolása a többi MICROSOFT-BASIC-hez hasonlóan a kulcsszavak használatával történik. Minden sor egy 2 bájtól tárolt sorszámából (bináris forma) és egy 1 bájtól tárolt sorhosszból (szintén bináris forma, a sorhossz számításánál a sorszámot nem, de a sorvégkaraktert beszámítja) áll, ezt követik a programkódok és végül a sorvégkarakter (0DH).

A program végét a sorszám helyén elhelyezett FFH jelzi. Egy példa látható a 4. táblázatban.

## Kulcsszavak a ROM-ban

A ROM három területén található a

kulcsszavak kódok: B000H-B0E9H, B800H-B887H, C054H-C34EH.

Az 5. táblázat az 1. táblázat kulcsszavait tartalmazza az ezen a ROM-területen belüli címekkel együtt. Néhány helyen a cím hiányzik. Ezek ott nem találhatók, de nem találhatók a ROM más részein sem. Ezt a táblázatot azért állítottam össze, hogy megkönnyítsem a ROM-ban keresést azoknak, akik nem kívánják kivérni a sorozat végét.

## Üzenetek

A rendszer „bejelentkezésénél” használt üzenetek a következők címtérületen találhatóak (6. táblázat): C34FH-C36AH.

DR. SIMONYI ENDRE

### 1. táblázat. Kulcsszavak kódok

ABS	F170H	DTE	E884H
ACS	F174H	END	F18EH
AND	F150H	ERL	F053H
AREAD	F180H	ERN	F052H
ARUN	F181H	ERROR	F1B4H
ASC	F160H	EXP	F178H
ASN	F173H	FEED	F0B0H
ATN	F175H	FOR	F1A5H
BEEP	F182H	GCURSOR	F093H
BREAK	F0B5H	GLCURSOR	E682H
CALL	F18AH	GOSUB	F194H
CHAIN	F0B2H	GOTO	F192H
CHRS	F163H	GPRINT	F09FH
CLEAR	F187H	GRAD	F186H
CLOAD	F089H	GRAPH	E681H
CLS	F088H	IF	F196H
COM\$	E858H	INKEY\$	F15CH
CONSOLE	F0B1H	INPUT	F091H
CONT	F183H	INSTAT	E859H
COLOR	F0B5H	INT	F171H
COS	F17EH	LCURSOR	E683H
CSAVE	F095H	LEFTS	F17AH
CSIZE	E680H	LEN	F164H
CURSOR	F084H	LET	F198H
DATA	F18DH	LF	F0B6H
DEG	F165H	LINE	F0B7H
DEGREE	F18CH	LIST	F090H
DEV\$	E857H	LLIST	F0B8H
DIM	F18BH	LN	F176H
DMS	F166H	LOCK	F1B5H
		LOG	F177H
LPRINT	F0B9H	ROTATE	E685H
MEM	F185H	RUN	F1A4H
MERGE	F08FH	SETCOM	E882H
MIND\$	F17BH	SETDEV	E886H
NEW	F19BH	SGN	F179H
NEXT	F19AH	SIN	F17DH
NOT	F16DH	SORGN	E684H
OFF	F19EH	SPACES	F061H
ON	F19CH	SQR	F16BH
OPN	F19DH	STATUS	F167H
OR	F151H	STEP	F1ADH
OUTSTAT	E880H	STOP	F1ACH
PAUSE	F1A2H	STR\$	F161H
PEEK	F16FH	TAB	F0BBH
PEEK*	F16EH	TAN	F17FH
PI	F15DH	TERMINAL	E883H
POINT	F168H	TEST	F0BCH

POKE	F1A1H	TEXT	E686H
POKE #	F1A0H	THEN	F1AEH
PRINT	F097H	TIME	F15BH
RADIAN	F1AAH	TO	F1B1H
RANDOM	F1A8H	TRANSMIT	E885H
READ	F1A6H	TROFF	F1B0H
REM	F1ABH	TRON	F1AFH
RESTORE	F1A7H	UNLOCK	F1AFH
RETURN	F199H	USING	F085H
RIGHTS	F172H	VAL	F162H
RINKEYS	E85AH	WAIT	F1B3H
RLINE	F0BAH	ZONE	F0B4H
RMT	E7A9H		
RND	F17CH		

34H	= "
42H	= B
34H	= "
0DH	= sor vége
00H 1EH	= 30
03H	= a sor karaktereinek száma
FIH ACH	= STOP
0DH	= sor vége
FFH	= program vége

DIM	C0D2H	LN	C197H
DMS	C0ECH	LOCK	C188H
		LOG	C18FH
LPRINT	B0A6H	ROTATE	B0BBH
MEM	C1C1H	RUN	C25EH
MERGE	B873H	SETCOM	—
MID\$	C1C9H	SETDEV	—
NEW	C1E3H	SGN	C2D0H
NEXT	C1D2H	SIN	C2C8H
NOT	C1DBH	SORGN	B0C6H
OFF	C201H	SPACES	—
ON	C1EBH	SOR	C2C0H
OPN	C1F9H	STATUS	C2E1H
OR	C1F2H	STEP	C2E8H
OUTSTAT	—	STOP	C2B7H
PAUSE	C24AH	STR\$	C2D8H
PEEK	C224H	TAB	B0D0H
PEEK #	C21AH	TAN	C2FEH
PI	C213H	TERMINAL	—
POINT	C240H	TEST	B0D8H
POKE	C237H	TEXT	B0E1H
POKE #	C22DH	THEN	C2F5H
PRINT	C209H	TIME	C306H
RADIAN	C2A4H	TO	C322H
RANDOM	C2E8H	TRANSMIT	—
READ	C271H	TROFF	C318H
REM	C2AFH	TRON	C30FH
RESTORE	C27CH	UNLOCK	C333H
RETURN	C266H	USING	C329H
RIGHTS	C299H	VAL	C33EH
RINKEYS	—	WAIT	C346H
RLINE	B0B1H	ZONE	—
RMT	B87DH		
RND	C286H		

2. táblázat. Számítárolás

Decimális  
 04H 00H 12H 34H 00H 00H 00H = 1234  
 00H 00H 12H 34H 00H 00H 00H = 1.234  
 FDH 00H 12H 34H 00H 00H 00H = 0.001234  
 03H 30H 12H 34H 00H 00H 00H = -1234  
 Bináris  
 XXH XXH XXH XXH B2H 7FH FFH XXH = 32767  
 XXH XXH XXH XXH B2H 80H 00H XXH = -32768

3. táblázat. Karakteroszorzat-tárolás

XXH XXH XXH XXH 00H 00H 7BH 22H 11H = A  
 karakteroszorzat a 7B22H címtől kezdve tárolódik, és 17 bájttal hosszú

4. táblázat. Programtárolás

00H 14H = 20  
 06H = a sor karaktereinek száma (6, mert a sorvég is számít)  
 F0H 97H = PRINT

5. táblázat. A kulcsszókódok címei

ABS	C067H	DTE	—
ACS	C07FH	END	C10DH
AND	C05FH	ERL	—
AREAD	C055H	ERN	—
ARUN	C08FH	ERROR	C11DH
ASC	C087H	EXP	C115H
ASN	C077H	FEED	—
ATN	C06FH	FOR	C127H
BEEP	C098H	GCURSOR	C14DH
BREAK	C35CH	GLCURSOR	B073H
CALL	C0D8H	GOSUB	C138H
CHAIN	B855H	GOTO	C12FH
CHR\$	C0CFH	GPRINT	C142H
CLEAR	C0B5H	GRAD	C159H
CLOAD	B05FH	GRAPH	B069H
CLS	C0BFH	IF	C16CH
COMS	—	INKEYS	C17BH
CONSOLE	—	INPUT	C162H
CONT	C0A1H	INSTAT	—
COLOR	B055H	INT	C173H
COS	C0C7H	LCURSOR	B080H
CSAVE	B069H	LEFTS	C1AEH
CSIZE	B05FH	LEN	C1A6H
CURSOR	C0AAH	LET	C19EH
DATA	C104H	LF	B08CH
DEG	C0F4H	LINE	B093H
DEGREE	C0E9H	LIST	C186H
DEV\$	—	LLIST	B09CH

6. táblázat. Üzenetek

NEWO?	C34FH
:CHECK	C355H
BREAK	C35CH
IN	C362H
ERROR	C36CH

PTA—4000

Forgástestek és felületek

A program a síkbeli koordinárendszerben megadott paraméteres függvényt forgatja meg az Y tengely körül, és ábrázolja a megadott rálátással.

A program felépítése

- 10—16 kezdőértékek beállítása
- R: nagytás
- N: szeletek száma
- PHI: a rálátás szöge
- COLOR 1, 2: színek (két szín is lehet)
- VÁLTÁS: két szín esetén X() és Y() vektorok kiszámítása, X szélsőérték-kikeresés
- X() vektor transformációja, Y() szélsőértékek kikeresése
- 30—60 forgatás és ábrázolás: külső ciklus: szeletekre bontás
- belső ciklus: egy szelet ábrázolása
- 500—520 A paraméteres függvény Q segítségével, melynek értéke  $\theta$ -tól  $2\pi$ -ig fut.

Az 1. és 2. ábrán példaként néhány trigonometrikus függvényt ábrázoltunk.

MÉSZÁROS CSABA

1. ábra

```

10: CLEAR : RADIAN : GRAPH : INPUT "R="
11: "N:" N : PHI = 0
15: DIM X(60), Y(60) : INPUT "COLOR 1="
16: "2:" COLOR 2 = 1 : UNLESS (1:1 B:
17: ) 1
18: C=COS F: S=SIN F: Y=99: X=Y: HAXIN
19: =H: (M: 2: ) INT (M: 2: )
20: FOR I=0 TO 60: G=0: 360: I: GOSUB 300:
21: NEXT I: FOR I=0 TO 60: Y(I)=Y: I: Y:
22: IF X(I) <= 0 THEN X(I)=
23: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
24: Y(I)
25: IF X(I) <= 0 THEN X(I)=
26: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
27: Y(I)
28: FOR I=0 TO 60: G=0: 360: I: GOSUB 300:
29: NEXT I: FOR I=0 TO 60: Y(I)=Y: I: Y:
30: IF X(I) <= 0 THEN X(I)=
31: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
32: Y(I)
33: IF I=0 THEN X(I)=
34: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
35: Y(I)
36: IF I=0 THEN X(I)=
37: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
38: Y(I)
39: IF I=0 THEN X(I)=
40: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
41: Y(I)
42: IF I=0 THEN X(I)=
43: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
44: Y(I)
45: IF I=0 THEN X(I)=
46: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
47: Y(I)
48: UNY(Y(I)-X(I): ) Y: I: Y:
49: IF I=0 THEN X(I)=
50: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
51: Y(I)
52: IF I=0 THEN X(I)=
53: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
54: Y(I)
55: IF I=0 THEN X(I)=
56: =X(I): 90: UNY(Y(I)-X(I): ) Y: I: Y:
57: Y(I)
58: LINE (I: 0, Y: 15): NEXT I:
59: GLCURSOR (I: 0, Y: 15):
60: RETURN

```

2. ábra

```

500: Y(I)=Y: I: Y:
510: X(I)=X: I: X:
520: RETURN

```

# MEGRENDELHETŐ!

<b>1. MÓDSZERTANI SEGÉDANYAG A MUNKÁLTATÓK SZOCIÁLIS TERVEZÉSÉHEZ</b>	<b>300,— Ft</b>
<b>2. LÍZING ALKALMAZÁSÁNAK GYAKORLATA</b>	<b>1200,— Ft</b>
<b>3. BELKERESKEDELMI DOLGOZÓK BÉRBESOROLÁSA</b>	<b>900,— Ft</b>
<b>4. VAGYONBIZTONSÁG A KERESKEDELEMBEN</b>	<b>500,— Ft</b>
<b>5. KÖZÜTI ÁRUFUVAROZÁS I—II—III.</b>	<b>800,— Ft</b>
<b>6. JÖVEDELEMÉRDEKELTSÉG A BOLTI KISKERESKEDELEMBEN</b>	<b>200,— Ft</b>
<b>7. SZERZŐDÉSES ÜZLETEK NYILVÁNTARTÁSI RENDSZERE</b>	<b>300,— Ft</b>
<b>8. KERESKEDELMI VÁLLALATOK KÖZÉPTÁVÚ TERVEZÉSE</b>	<b>930,— Ft</b>
<b>9. SZERZŐDÉSES ÜZLETI VÁLLALKOZÁS A VENDÉGLÁTÁSBAN</b>	<b>300,— Ft</b>
<b>10. FIZIKAI DOLGOZÓK BÉRBESOROLÁSA</b>	<b>600,— Ft</b>
<b>11. ÜZLETI MUNKAREND MINTASZABÁLYZATA</b>	<b>300,— Ft</b>
<b>12. A KERESKEDELMI VÁLTÓ ÉS HITEL GYAKORLATA</b>	<b>1500,— Ft</b>
<b>13. A PÉNZTÁRI MUNKAFELTÉTELEK FEJLESZTÉSI IGÉNYEI ÉS LEHETŐSÉGEI</b>	<b>85,— Ft</b>

## **KERESKEDELMI SZERVEZÉSI INTÉZET**

Budapest XIII., Dózsa György út 150. 1134

Marketing osztály      Telefon: 202-650, 202-670



## szisztémái — mikroszámítógépre

A felhasználói programok közül a SYSTEM Szervezési Vállalat HSR—MICRO programcsomagját feltűnően sok, több mint 100, a KFR—MICRO és PEGA—MICRO programcsomagját 10-10 vállalat vásárolta meg.

A közelmúltban a HSR—MICRO programcsomagért a vállalat az SZKI—SCITEL partnertalálkozón I., a Software '86 kiállításon és vásáron II. díjat kapott. Feltehető a kérdés, hogy ez a köztudottan nagy számítógépet üzemeltető szervezési vállalat a mikroszámítógépek alkalmazása és felhasználása területén is miért tudott ilyen jelentős eredményt elérni, milyen innovációs képességekkel rendelkezik. A kérdésre a következőkben adunk választ.

A SYSTEM közel húsz éve foglalkozik beruházásszervezéssel. Állíthatjuk, hogy ez idő alatt a SYSTEM — beruházásszervezési tapasztalatai alapján — a beruházások szervezésének szakértőjévé fejlődött. Kezdetben a vállalat beruházásszervezési munkáját — érthetően — saját nagy számítógépével támogatta. Később a lehetőségek felismerésével, valamint az alkalmazók és felhasználók igényére, áttért a sok tekintetben nagyobb rugalmasságot biztosító mikroszámítógépek alkalmazására, és egymás után dolgozta ki a HSR—MICRO, hálószerkesztés és rajzoló sornyomtatóra, kombinált MPM/CPM módszer, vonalas ütemterv változatható időtengellyel, erőforrások, aggregáció-histogram; a KFR—MICRO, a fejlesztések, a beruházások, a fővállalkozások sikeres irányítását elősegítő költségfigyelési; a PEGA—MICRO, a beruházások pénzügyi és gazdasági elemzését szolgáló stb. programcsomagokat.

A SYSTEM a H. B. Maynard and Ltd. angol szervező cégtől megvásárolta az UNIVERSAL MAINTENANCE STANDARDS néven ismert UMS munkamérésen alapuló karbantartási-irányítási rendszert. A módszer segítséget nyújt a karbantartás valamennyi területének, a munkamegrendelési és elszámolási tevékenységnek, a megelőző karbantartásnak, az ösztönzésnek, az ellenőrzésnek a hatékony szervezésére. A SYSTEM a vállalati alkalmazás manuális munkaigényességének csökkentésére kifejlesztette a mikroszámítógépes MICRO—UMS karbantartási munkalapkezelő és vezetői információt szolgáló rendszert. A rendszer bevezetése után a karbantartási munka termelékenysége minimum 40%-kal nőtt.

A munkatanulmányozás, a munkaszervezés, a munkatervezés alkalmazása, de kevésbé használt módszere az MTM. A módszer alkalmazásával a munka megkezdése előtt megállapítható a munka minden elemének kivitelezési módja, az ehhez tartozó idő és ennek megfelelően az optimális munkamódszer. Az előbbihez képest tömeges, nagyméretű, idő- és munkaigényes feladatok (termelői folyamatok) széles körű racionalizálását egy továbbfejlesztett változat, a H. B. Maynard cég által kidolgozott MOST rendszer teszi lehetővé. A mikroszámítógépre kifejlesztett VIA—SYSTEM varrodai időadatrendszer segítségével — az elemzések alapján — gyors és megbízható normák képezhetők és előalkulációk készíthetők az igény szerinti konfekcionálási tevékenységre.

A SYSTEM számítógépes anyaggyártórendszere a mikrogépes felhasználók szinte minden nyilvántartási igényét kielégíti, mivel rendkívül rugalmasan képes alkalmazkodni a helyi sajátosságokhoz, igényekhez.

A SYSTEM állóeszköz-nyilvántartási és -elszámolási rendszere

nemcsak a komplex állóeszköz-gazdálkodási rendszernek, de az üzemfenntartásnak is bázisát képezi. Az állóeszköz-állományban és -értékesítésben bekövetkező változások mikrogépes kimutatása, az értékcsökkenési leírások kiszámítása és elszámolása mellett ugyanis az állóeszközök műszaki adatainak nyilvántartását is biztosítja.

A SYSTEM mikrogépes munkaügyi nyilvántartási rendszere magában foglalja a személyzeti, oktatási, létszámgyártó feladatköröket. Az operatív irányítás feladatainak megoldásához szükséges mikrogépes információk biztosítása elősegíti a gazdálkodás által megkívánt élőmunka-hatékonysági szint elérését.

A SYSTEM mikroszámítógépes menetlevél-feldolgozó rendszere biztosítja a gépjárművek teljesítményadatainak gyűjtését, az üzemanyag-felhasználási elszámolásokat, fuvarköltésegek figyelését.

A SYSTEM személyi számítógépre kidolgozott vezetői rendszerével a vezetői döntési alternatívák komplex hatékonysági és rangsorát lehet megállapítani.

Az SCR 11 MSZR számítógépre készült szubrutin-csomag a FORTRAN, a BASIC és C nyelvi interaktív program készítését segíti, több terminálkezelési lehetőséggel és minimális memóriáigénnyel. A kidolgozott QUICK—SYS programfejlesztő rendszerrel a mikroszámítógépes programok futási sebessége jelentősen növelhető.

A mikroszámítógépek vállalati alkalmazása — többek között — az elektronika befogadására alkalmas vállalati szervezetet is igényel. A SYSTEM foglalkozik az erre alkalmas korszerű vállalati szervezet és vállalatirányítási rendszerek kidolgozásával, bevezetésével, az iparvállalatok működési és szervezeti rendszerének komplex átalakításával, a vállalati érdekeltségi rendszerek kidolgozásával, kialakításával, a vezetési, irányítási információk mikroszámítógépes feldolgozásának előkészítésével. A számítástechnikában rejlő lehetőségek kényelmes és szakszerű kihasználását szolgáltatja a vállalatok számára a SYSTEM számítógéppontja. IBM kompatibilis nagy számítógépe — amelyhez bérletszerű formában kihelyezett terminálokat is a megbízók rendelkezésére bocsát — minden igényt kielégít. A SYSTEM számítógéppontja megbízói részére komplex kiszolgálást biztosít, amely magába foglalja a konkrét gépi futtatások elvégzésén kívül a megfelelő számítógépes munka előkészítést, valamint — az üzemeltetési dokumentációban rögzített szintig — az eredménytáblák ellenőrzését, esetenként a kész eredménytáblák kiszállítását is.

A SYSTEM számítógéppontja vállalja TAF és osztott adatfeldolgozási rendszerek tervezését, fejlesztését, üzemeltetését.

A SYSTEM a FERROGLOBUS, a METALOGLOBUS, a VEGYTEK és a VILLÉRT TEK vállalatokkal létrehozta az UNGLOB Készletinformációs Értékesítő Közös Vállalatot, melynek célja a vállalati felesleges és lassan mozgó készletek mobilizálása, az anyaghiány miatti termelés kiesések kiküszöbölése, korszerű számítástechnikai eszközök segítségével.

Rövid ismertetőnkben természetesen lehetetlen a SYSTEM valamennyi módszerét, eljárását, szolgáltatását bemutatni. Abban viszont biztosak vagyunk, ha felkeresi a vállalatot, problémáira megtalálja a legjobb megoldást.

# TURBO interfész

## ZX-Spectrumhoz

EPROM program

```
0020 CB C9 CA CB C3 C1 C2 C0
0028 8B 89 8A 8B 83 81 82 80
0030 4B 49 4A 4B 43 41 42 40
0038 0B 09 0A 0B 03 01 02 00
```

```
0120 CB C9 CA CB C3 C1 C2 C0
0128 8B 89 8A 8B 83 81 82 80
0130 4B 49 4A 4B 43 41 42 40
0138 0B 09 0A 0B 03 01 02 00
```

```
0140 7F 7F FF FF 7F 7F FF FF
0148 7F 7F FF FF 7F 7F FF FF
0150 7F 7F FF FF 7F 7F FF FF
0158 7F 7F FF FF 7F 7F FF FF
```

```
01C0 34 35 36 37 3C 3D 3E 3F
01C8 74 75 76 77 7C 7D 7E 7F
01D0 B4 B5 B6 B7 BC BD BE BF
01DB F4 F5 F6 F7 FC FD FE FF
```

```
0220 CB C9 CA CB C3 C1 C2 C0
0228 8B 89 8A 8B 83 81 82 80
0230 4B 49 4A 4B 43 41 42 40
0238 0B 09 0A 0B 03 01 02 00
```

```
0240 35 3D 35 3D B5 BD B5 BD
0248 75 7D 75 7D F5 FD F5 FD
0250 37 3F 37 3F B7 BF B7 BF
0258 77 7F 77 7F F7 FF F7 FF
```

```
02C0 34 74 B4 F4 3C 7C BC FC
02C8 35 75 B5 F5 3D 7D BD FD
02D0 36 76 B6 F6 3E 7E BE FE
02DB 37 77 B7 F7 3F 7F BF FF
```

```
0320 CB C9 CA CB C3 C1 C2 C0
0328 8B 89 8A 8B 83 81 82 80
0330 4B 49 4A 4B 43 41 42 40
0338 0B 09 0A 0B 03 01 02 00
```

ján, megfelelő felszereléssel az interfész könnyen elkészíthető. Ha valakinek mégis sincs lehetősége a NYÁK elkészítésére vagy az EPROM beégetésére, annak készséggel segítségére leszek a sorok írója.

RUSZNYÁK GÁBOR

### Ki ad magyarázatot?

Szieberth András olvasónk SPECTRUM gépén tapasztalta a következőket:

Ha a DRAW utasítás után harddisk számként 300-1000

közi számot adunk meg, akkor egyes esetekben hibázenetet kapunk, más esetekben érdekes alakzatok jelennek meg a képernyőn.

Legyen pl.

PLOT 130,80:DRAW 100,30,1400

Az eredmény egy háromszög, amely forog. Az utolsó szám helyett 1900 értéket adva ötszöget kapunk.

A jelenség magyarázatát szakkönyvekben sem találta meg.

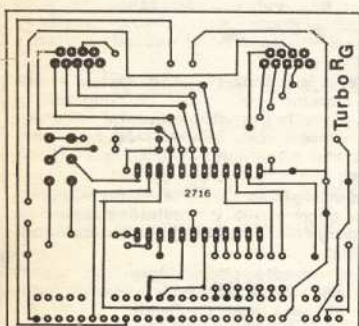
Nemrég mutattak nekem egy gyári készítésű TURBO interfészt. Amikor elmondták, mit tud, elhatároztam, hogy én is készítek egy hasonlót. Nosza, szétszedtem, de a sejtésem beigazolódt: az áramkörben egy „Taiwan” felirátú, speciális integrált áramkör volt. Ettől egy kissé elment a kedvem. Később mégis gondolkoztam, hogy mit lehetne tenni, és akkor jutott az eszembe a következő megoldás.

Szükség van egy IC-re, aminek 10-11 be-

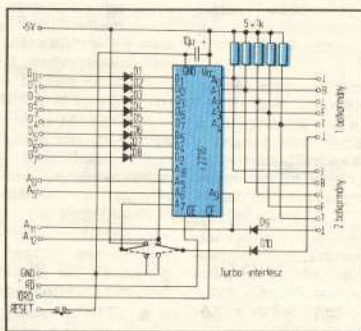
— A Protek a felső billentyűzetsort olvassa le, az irányítást a nyíllal jelzett billentyűk, a lövést a 0 jelenti.

— Az Interface II szintén a felső sort olvassa le, de ez két botkormányt kezel úgy, hogy egy fél sor jelent egy botkormányt.

Ebből is látszik, hogy a Protek és az Interface II fedik egymást. Ezt a problémát egy kapcsoló beiktatásával oldottam meg. Ez a kapcsoló egyrészt 0 vagy 1 állapotba hozza az EPROM egyik bemenetét, más-



1. ábra



3. ábra

részt az 1. számú (jobb oldali) botkormány közös pontját kapcsolja a földre vagy az A12 vezetékre.

A kapcsoló felső állásában az 1. számú botkormány Kempston és Protek üzemmódban működik, a kapcsoló másik állásában a két botkormány Interface II módon illeszti az áramkör.

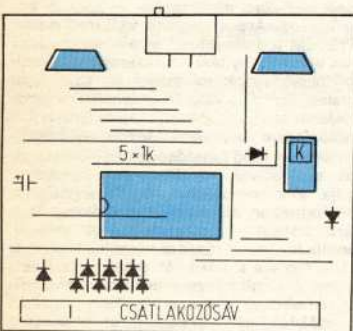
Ezenkívül a bemenetekre csatlakozik még az A0, ami a billentyűzetet cimezi, az A5, ami a Kempstont cimezi, az A11 és az A12, ami a billentyűzetet a két felső fél sort választja ki. A CE bemenetre kapcsolt IORQ és az OE-re kapcsolt RD gondoskodik arról, hogy az EPROM csak az IN műveletekre aktivizálódjon.

A 8 db dióda (D1-D8) biztosítja, hogy az áramkör ne tiltsa le a billentyűzetet. A diódák típusa 1N4148 vagy 1N914.

Az EPROM-ot érdemes foglalalba tenni, mert egyébként a beforrasztások tönkremehet. A botkormány-csatlakozókat kb. 10 mm hosszú ónozott rézhuzalok beforrasztásával lehet helyettesíteni, melyeknek az átmérője 1 mm. A kapcsolás RESET gombjával a NYÁK tetejére szerelt mikrokapcsoló szolgál.

A kapcsolás helyes működéséhez az EPROM-ba a lista szerint közölt programot kell beégetni. Mivel csak 1 kb-ajt memóriát használunk fel, az EPROM A10 bemenetét földre kell kötni, így csak az első 1 kb-ajtot kell beégetni. A listában nem közölt bajtókba FF (255) tartalmat kell beégetni.

Az utánépítők munkájának megkönnyítésére közlöm a főlírajzot (1. ábra), a beültetési rajzot (2. ábra) és a kapcsolási rajzot (3. ábra). A NYÁK elkészíthető egyszálalású kivitelen átkötésekkel, illetve kétoldalas kivitelen. A közölt dokumentációk alap-



2. ábra

menete és 8 kimenete van. A kimenetek állapota mindig a bemenetek állapota szerint változik. Ezt a funkciót egy kellően beprogramozott EPROM is ellátja. A jól bevált i2716 típus mellett döntöttem, mivel ezt könnyen be lehet szerezni, és az ára is elfogadható.

A TURBO ismeri a háromféle botkormány-illesztési módot, a Kempston, a Protek és a Sinclair Interface II üzemmódot.

— A Kempston az IN 31 utasítással olvassa be a botkormány biteit, és így azonosítja az irányítást.



Az értékelőfüggvény egyik nagyon fontos komponense a királytámadás és a királyvédelem értékének a meghatározása, hiszen tulajdonképpen az egész sakkjáték ezen alapszik. Az előbbieken általános sakkstratégiai elemeket ismertettünk, most pedig a kiüntetett szereppel bíró figurák, a király támadásának és védelmének kvantitatív meghatározását tárgyaljuk. Az előbbi cikkben taglaltuk, hogy a közép-rekedt király mennyire hátrányos, mert kettéosztja saját haderejét, megakadályozza annak ésszerű elhelyezését és átcsoportosítását. A közép-rekedt királyra az ellenfél könnyen csapást mérhet, mert miután a megnyitásban a centrum-nyalokkal kiléptünk helyükről, a középben maradt királyt a gyalogok nem védik, és a tábla közepén megnyíló vonalakon, átlókon keresztül hamar az ellenséges figurák célpontjává válik. Ezért szinte nélkülözhetetlen a sakkjátékban a mielőbbi sáncolás.

Steinitz még gyönyödlött azokon, akik királyukat szerinte gyáván sáncolással a gyalogok mögé rejtették. Az ő király — mondotta — bátran rohán a harcba. De már a kortársai között is akadtak, akik a király biztonságáról való gondoskodást — a sáncolást — nem a gyávaaság, hanem a körültekintés és az óvatosság megnyilvánulásának tartották. A romantikus sakkiskola hívei is kevesebb figyelmet szenteltek a sáncolásnak. Az akkor divatos megnyitások — királycsel, Evans-csel — gyakran jártak a sáncolás korai elvesztésével.

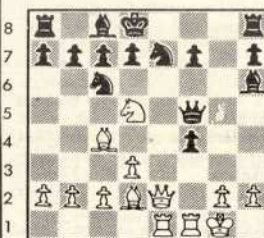
A mai gyakorlatiasságra hajló korban azonban, amikor a versenyszerű sakkozás térhódításával az eredményesség szempontja mindenek feletté vált, sokan nem gondolnak aktív hadműveletre addig, amíg saját királyuk biztonságát sáncolás útján meg nem teremteték. Az egészséges játszmafeleltetés elengedhetetlen feltételnek tekintik, hogy a király és a bástya közti könnyű tisztetek mielőbb kifejesszék, hogy azok ne akadályozzák a sáncolást.

Akadnak kivételek, amikor a sáncolás előtt hasznosabb lépéseket is tehetünk. Vannak megnyitási változatok, amelyekben csak a tizenötödik és huszadik lépés között kerül sor a sáncolásra. A sáncolás lehetőségét azonban, még ha nem is élünk vele, azonnal célszerű mindig készenlétben tartani.

A játszma többiségében ró-

## BITEK ÉS FIGURÁK ÁLLÁSÉRTÉKELÉS IV.

### Királytámadás és védelem I.



a b c d e f g h

1. ábra

8	1	1	1	1	1	1	1	1
7	2	2	2	2	2	2	2	2
6	2	3	3	3	3	3	3	3
5	2	3	4	4	4	4	4	4
4	2	3	4	5	5	5	5	5
3	2	3	4	5	6	6	6	6
2	2	3	4	5	6	7	7	7
1	2	3	4	5	6	7	7	7

a b c d e f g h

3. ábra

vid sáncot alkalmaznak. Ehhez csak két bábuval, a huszárval és a futóval kell lépni, míg a hosszú sánc esetén a vezérnek is el kell hagynia alapállásbeli mezéjét. A futó és a huszár korán fejleszhető figurák, a vezért viszont később kell fejleszteni, mert ellenkező esetben a könnyű tisztetek célpontjává válhat, és tempóvesztésekkel kell visszavonulnia.

A rövid sáncot más szempontból is biztonságosabbnak tartják a hosszú sáncolásnál. Rövid sánc esetében a királyt az f2, g2, h2 gyalogok védik, míg hosszú sánc esetében, amikor a király c1-re kerül, annak a d2, e2, b2 gyalogok felelnek meg. De a vezérfutó lépése előtt a d és b gyalogok közül legalább az egyikkel el kellett lépünk, hogy így lehetővé vál-

8	5	6	7	7	7	6	5	4
7	5	6	7	7	7	6	5	4
6	5	6	6	6	6	5	4	
5	5	5	5	5	5	5	5	4
4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1

a b c d e f g h

2. ábra

8	2	4	8	8	8	4	2	1
7	2	4	8	8	8	4	2	1
6	2	4	4	4	4	4	2	1
5	2	2	2	2	2	2	2	1
4	1	1	1	1	1	1	1	1
3	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0

a b c d e f g h

4. ábra

jék a hosszú sáncolás. A királyt védő gyalogosok közül tehát egyiknek el kellett lépnie a helyéről, s így a királyállás eleve meggyengült. Ezenkívül a védetlen a-vonal és az a2 (a7) gyalog hosszú sánc esetén további támadható pontot jelent az ellenfélnek. Ezért a királyállás biztonságosabbá tétele érdekében szükség van még a Kc1-b1 lépés megtételére is. Ennek azonban hátránya, hogy a király már a megnyitásban kétszer lép, és ez tempóvesztést jelent.

Rövid sánc elleni támadásnál két eset között kell különbséget tennünk.

1. Támadás meggyengített királyállás ellen, amelynél a királyállást védő három gyalog egyike már lépett vagy eltűnt a tábláról.

2. Támadás intakt, érintetlen királyállás ellen, ahol a gyalogok még eredeti helyükön állnak.

A két eset egyébként elég gyakran átmege egymásba, vagyis az intakt királyállást a védekező az ellenfél különböző fenyegetéseinek hatása alatt kénytelen valamely gyaloglépéssel meggyengíteni. Ebből a szempontból legfontosabb az f3 (f6) huszár szerepe, amelyet a rövid sánc legfontosabb védőjének tekinthetünk. Ha ezt sikerül valamely módon, csere, gyaloggal való támadás, esetleg áldozat útján eltávolítanunk, a h7 pontra gyakorolt nyomással vagy más fenyegetésekkel rendszerint sikerül a g vagy h gyalog lépését kikényszeríteni. Mennyiben bomlik meg az állás egyensúlya e lépések által, amelyek látszólag csak minimálisan módosítják a gyalogállás szerkezetét?

Valójában többféle okból is lényegesen gyengíthetik a hadállást. Ezek a gyaloglépések megkönythetik a vonalynyt a királlyal szemben, akár oly módon, hogy az ellenfél előrenyomuló gyalogjai cserélődnek e gyalogok ellenében, ami a g- vagy h-vonal megnyitására vezet, akár esetleg tisztáldozat árán is. Másrészt a gyalogok által védetlenül hagyott mezőknél az ellenfél tisztjei a király közvetlen közelébe kerülhetnek. Különösen világos ez például g7-g6 (g2-g3) lépés esetén. A védetlenül maradt h6 (h3) mezőn, sőt többnyire az f6 mezőn is ellenséges tisztetek jelenhetnek meg, és fokozhatják a támadást.

Ezeket a szempontokat a sakkprogramnak is figyelembe kell vennie. De mielőtt ezzel konkrétan foglalkoznánk, tanulmányozzuk a két fél elentetés oldalra történő sáncolásának esetét!

A támadás legszebb példáival találkozhatunk az ilyen játszmákban. Ez érthető, hiszen az egyensúly — a két király elhelyezkedésének szempontjából — erősen megbomlott. Az azonos oldalra történő sáncolásnál a királyok szimmetrikus elhelyezkedése sokáig egyensúlyban tarthatja az állást. Az ellentétes oldalra történő sáncolásnál viszont nyugodtan támadhatunk minden eszközzel az ellenfél királyának térségében, mert ezek az akciók nem járnak egyidejűleg a saját királyállásunk veszélyeztetésével. Ezért itt a támadás egyik leggyakoribb módszere a legyoprogram. Ennek célja itt is a vonalynyt az el-

lenfél királyával szemben. Sok esetben a gyalogrohamot nem is lehet passzív védekezéssel elhárítani. Ezért szokták mondani, hogy ellentétes oldalú sáncolás esetén a legjobb védekezés az ellentámadás. A küzdelem a végősig kiéleződik, igen sok szerephez jut az idő, minden tempónak döntő jelentősége lehet. Ezért a merész áldozatok, meglepő taktikai fordulatok egész tárházát alkalmazhatja a támadó, mert ha a számára fontos hadszínterén kierőszakolja a döntést, nincs jelentősége annak, hogy a másik hadszínterén esetleg anyagi vagy pozíciós hátrányban maradjon. Itt valóban érvényes az a tétel, hogy a döntő időben a döntő hadszínterén kell előnyre szert tenni.

Mivel ilyen jellegű küzdelem során az anyagi egyensúly is gyakran felborul, konkrét, változatokénti számítás szükséges, aminek a számítógép gyorsan és pontosan eleget tud tenni.

Aljechin a következőket mondta: „A két szárnyon egyszerre folyó hadművelet az én kedvelt stratégiám. Az ilyen játszmám fő jellegzetessége a váratlanul és gyors döntést hozó támadás. A támadásokat nem a célpont közelében készítsem elő, hanem valamennyi előkészítő manővert, melynek célja a királyállást védő figurák eltávolítása volt, a centrumban vagy az ellenkező szárnyon bontakozott ki. Igen érdekes az a körülmény is, hogy a döntő kombináció, melyet porlyócsapáshoz lehet hasonlítani, a futó támogatásával és mindig áldozattal történt.”

Oszvath András mesteredző így foglalt állást: „Az aszimmetrikus királyállások a játék különböző fázisaiban éreztetik hatásukat, és ezzel hozzájárulnak a hadállások kiélezéséhez. Ezért a különböző oldalú sáncolás, mint az állás egyensúlyának megbontására irányuló törekvés, szinte kísérő jelensége a modern megnyitásképezésnek is. Aki győzelemre tör, az kockáztat, felborítja a szimmetriát, megbontja az egyensúlyt, és ellenkező oldalra sáncol.”

Ezeket az elveket figyelembe véve célszerű sakkprogramunkat is elkészíteni. Tudvalevő, hogy a programoknak a végjáték a gyengéje, ezért úgy kell vezetnünk a játékok, hogy lehetőleg már a középjátékban eldőljön a játék kimenetele. Ezért David Levynek az előző számban említett példájával szemben az ellentétes oldalra történő sáncolás felpontozását

tartom jónak, annak ellenére, hogy ezekben az esetekben legtovább a programnak kell a vezérszárnyra sáncolnia. A programnak fel kell mérnie a király megtámadásának mértékét, aminek kvantitatív meghatározása nagyon nehéz feladat. Bemutatók egyelőre jónak bizonyuló közelítést.

Az ellenséges király elleni támadás erősségének meghatározására egyszerű módszer a király körüli mezők súlyozása. Hasonló, mint ahogy az előző cikkeinkben láttuk a centrum ellenőrzésekor.

A király melletti mezőkhöz sokkal nagyobb pontértéket rendelünk, mint a távolabbiakhoz. Így a király körül a királytól távolodva koncentrikus körökben egyre kisebb lesz a táblamező súlyozásának mértéke. A program támadás esetén össze tudja hasonlítani az értékeket, és ebből kiderül hogy mekkora aktivitást fejt ki.

David Levy egy egyszerű terv alapján 7-tel számolta a király által elfoglalt és azzal határos mezőket, 6-tal az ezeket határoló mezőket, 5-tel az ezután következőket és így tovább. Az 1. ábrán bemutatott hadállás királytámadási táblája a fentiek alapján világos szempontjából a 2. ábrán, sötét szempontjából a 3. ábrán látható módon alakul.

A összeszámoljuk az egyes figurákra vonatkozó pontértékeket, a következő eredményre jutunk.

A világos figurákra:

- Be1: 1+1+1+1+1+2=7
- Bf1: 1+1+2+3+4=11
- Kg1: 1+1+2+2+2=8
- a2: 3
- b2: 3+3=6
- c2: 3+3=6
- Fd2: 1+1+3+4+5+3+4=21
- Ve2: 1+1+1+2+2+3+4+5+6+7+3+4+4=43
- g2: 3+3=6
- h2: 3
- d3: 4+4=8
- Fc4: 2+3+3+5+6+5=24
- Hd5: 3+3+4+4+6+6+7+7=40

A sötét figurákra:

- Ba8: 1+2=3
- Fc8: 2+2=4
- Kd8: 1+1+2+2+2=8
- Bh8: 1+1+1+1+2=6
- a7: 3
- b7: 2+3=5
- c7: 3+3=6
- d7: 3+3=6
- He7: 1+1+3+3+4+4=16
- f7: 3+3=6
- h7: 3
- Hc6: 1+1+2+2+2+4+3+5=20

- Fh6: 1+2+4+5=12
- Vf5: 2+3+2+3+2+3+4+4+4+4+5+5+5+5+6=57
- f4: 6+6=12

Így világos figurái számára a királytámadási érték 186, a sötétre pedig 167.

Láthatjuk, hogy a különbség világos figuráinak javára mutatkozik, de nem egészen túlrözi a támadás erősségét, mely a sötét királyra sokkal veszélyesebb, mint amit a 186 és 167 pontérték közötti különbség mutat. Ennek oka, hogy itt bizonyos mezők pontértékeit még nem vettük figyelembe.

Ez nem kétséget kelt, hogy vannak figurák, melyek egy másikon keresztül is kifejthet hatást. Ilyen például az el bástya. Ez az e2 mezőn álló vezéren túl hatva fejt ki hatását a sötét térfelére. Ha ezeket a mezőket is hozzáadjuk a támadás mértékéhez — ami még +27 pontot jelent világosnak —, akkor az eredmény még feltűnően világos számára kedvez. Ezek közül az el bástya a gl mezőre is hat fl-en keresztül, és ugyanígy korrigálható az fl bástya királytámadási értéke is. Ezek után a királytámadás valódi értéke: 221 világos és 167 sötét számára. Ez hibében tükrözi a két fél támadási értéket.

Még jobb közelítést adhatunk az értékek a megállapítására, ha a királytól túl távoli mezőknek 0 pontértéket adunk — mivel ezek a mezők a királytámadás szempontjából kevésbé játszanak szerepet —, és az így súlyozott táblát veszszük figyelembe. Ugyanerre az állásra a sötét király szempontjából ez a 4. ábrán látható módon alakul.

A királytámadás szempontjából még sok más is figyelembe kell venniük, de ennek ismertetésével a következő folytatásban foglalkozunk.

KOVÁCS P. ATTILA



```
480 PM=PM+1:1FPM=5THEN PM=1
```

### 28. lista

```
690 IF X>M<-2&AND X<M<+6&AND Y<M<+13&AND Y<M<+18THEN M#&:LINE (M<,Y<)-<M<+5,M<+11>:PSET:BF:SC=SC+18:PLAY"1250431GFG0ED":FU=FU+10:M<CH<+1
```

### 29. lista

```
780 IF M<=>5THEN LV=LV+1:MS=MS+1:GOTO2248
```

### 30. lista

```
2248 BONUS=(LV-1)*10:SC=SC+BONUS<
2250 CLS:PRINT 2128,"WELL DONE,Y<
DU WAWE FINISHED"
2260 PRINT 234,"BONUS"
2270 TUNE#="02L4G:1L2GL4BBL2BGL<
46B03L2D0L4C02BL1AL4A03L2C02L4<
BAL2BGL4BBL2ADL4FNA1G:"
2280 ZT#="TUNE#*TUNE#":PLAY"7"&Z<
2290 GOTO2168
```

### 31. lista

```
710 PSET (RND(256),RND(78)+22)
720 GOSUB920
```

### 32. lista

```
920 IF U<190THEN FU=190
930 FU=FU-.75
940 LINE (FU,3)-(208,18),PSET,B<
F1:LINE (FU,1,4)-(38,9),PSET,BF<
950 IF U<31THEN I=278
960 IF U<98THEN SOUND150,1:PUT<1<
48,3)-(178,11),DR,PSET
```

### 33. lista

```
1278 PUT(X-3,Y-2)-(X+28,Y+14),CR<
,PSET
1280 SOUND280-Y,1
1298 Y=Y+2
1300 PUT(X,Y-2)-(X+23,Y+14),CH,P<
SET
1318 SOUND280-Y,1
1328 PUT(X,Y-2)-(X+23,Y+14),CL,P<
SET
1338 SOUND280-Y,1
1348 Y=Y+2
1345 IF Y<150THEN I=270
1358 PUT(X,Y)-(X+28,Y+11),CR,PSE<
T1:FORT=31:TO18STEP-2:PLAY"U"&STR<
(T*#)1255;03;DEAD#:NEXT:FOR=1T<
0500:NEXTT
```

### 34. lista

#### 35/a lista

```
970 GOSUB490
980 RETURN
490 IF I=1THEN 558
500 X1=280
510 ON PC GOSUB2100,2120,2140
520 IF I=2THEN 558
538 X1=227
540 ON PC GOSUB2100,2120,2140
558 PC=PC+1:IF PC=4THEN PC=1
560 RETURN
```

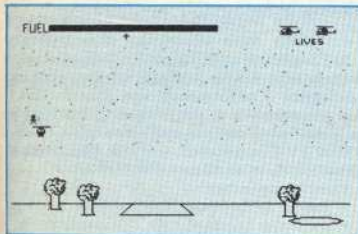
# A játékprogramozás technikája

## A FŐCIKLUS

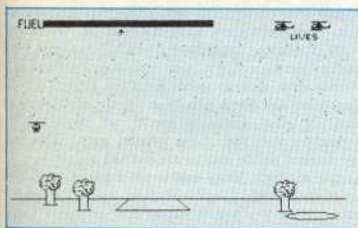
```

2100 PUT(X1,1)-(X1+23,14),CR,PSE
T
2110 RETURN
2120 PUT(X1,1)-(X1+23,14),CL,PSE
T
2130 RETURN
2140 PUT(X1,1)-(X1+23,14),CH,PSE
T
2150 RETURN
    
```

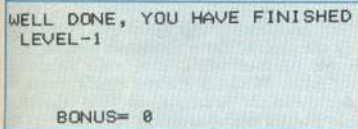
### 35/b lista



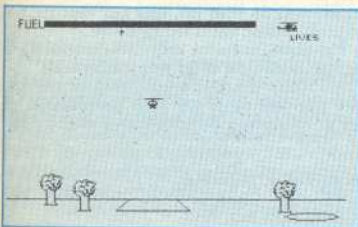
27. ábra



28. ábra

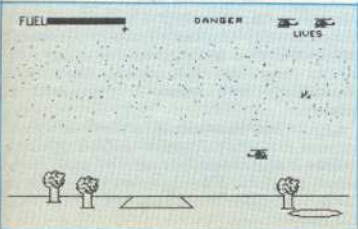


29. ábra



30. ábra

31. ábra



### Az emberkék zuhanása

A 28. listában a játék egyik fő része látható, melyben ciklikusan változtatjuk a zuhanó emberke pozícióját. Az emberke zuhanás közben hol áll, hol fekszik, hol pedig fejjel lefelé helyzetben van, és eközben zuhan.

Emlékeztetőül: a fejjel lefelé helyzetnek az M változó (PM=1), az álló helyzetű emberkének az M2 változó (PM=2;4), a fekvő helyzetű emberkének az M1 változó (PM=3) felel meg.

### A pontszerzés lehetősége

Ha sikerült időben elkapnunk a zuhanó emberkét, pontot szerezhünk, amennyiben mind az ötöt elkapjuk, úgy még jutalompont is jár. Fontos része tehát a programnak annak vizsgálata, hogy sikerült-e elkapnunk? Ez a 29. listában leírt programrészlettel történik. Ha igen, az emberke „eltűnik”, vagyis a helyét egy üres négyszöggel töltjük fel; kapunk 10 pontot, zenét, többletenergíát (FU); nő az elkapott emberkék száma (MC).

Megjegyzendő, hogy az „elkapáshoz” nem szükséges a helikopter és az emberke tényleges találkozása, elég, ha erősen megközelítjük. Ez látható két pozícióban, a 27. és a 28. ábrán. Az ezek előállításához szükséges programmodosítás:

```

690 ... THEN STOP: ...
MC=MC+1:STOP
    
```

A futtatás először

RUN

majd

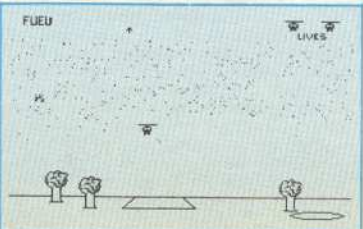
CONT

A dallam egy rövid, gyors futam, amelynek hangja olyan, mint amikor valaki csücsörített szájjal beszívja a levegőt.

### Mindent elraktuk?

A 30. listában megvizsgáljuk, hogy mind

32. ábra



az 5 emberkét sikerült-e elkapnunk. Ha igen, nő a szintszám (LV), a bonyolultsági mutató (MS), és rátérünk egy mellékágra.

### A „siker” mellékága

Ha mind az 5 emberkét elkaptuk, akkor egy mellékágban különböző jutalmazásokat teszünk. Ez úgy történik, hogy először kiszámítja a programrészlet az elért pontszámot, törli a képernyőt, kiír egy dicséretet, jelzi, hogy melyik szinten sikerült túljutnunk, majd eljátszik egy jutalomzenét. A kijelzett dicséretet a 29. ábra mutatja. Mindezt a 31. listában látható programrészlet valósítja meg.

A programmodosítások:

690 visszamodosítani

2285 STOP

A futtatás:

RUN 2240

A mellékág folytatása az eredményjelzésen keresztül történik. Az eljátszott jutalomzene az „Öh, te drága Clementina” című dalocska.

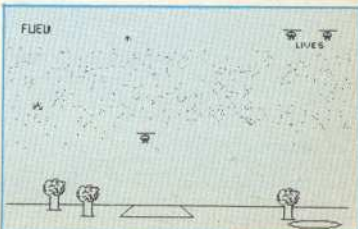
### A közös főág

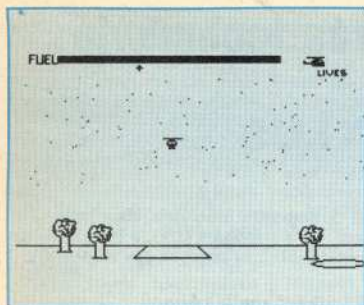
A 32. listában a 640-es sortól ugrás és az előző főág közös folytatása látható.

### Üzemanyag-vizsgálat

A játék során az üzemanyag fogy, töltéskor pedig egy bizonyos szintig növekszik. Az üzemanyag mennyiségével kapcsolatosakat a 33. listában lévő programrészlet intézi. A 920-as sor megakadályozza a „túl-töltődést”. A következő sor fogyasztja az üzemanyagot, és egyúttal ábrázolja annak mennyiségét. Ez a 30. ábrán látható. A 950-es sor vizsgálja, hogy elfogyott-e az üzemanyagunk. Ha igen, rátér egy mellékágra, ha még nem, de már veszélyesen kevés, akkor vészjelzést ad (960-as sor és 31. ábra).

33. ábra





34. ábra

A szükséges módosítások:  
2285 nincs  
945 IF FU < 110 THEN STOP  
965 STOP  
A futtatás:  
RUN  
majd  
CONT

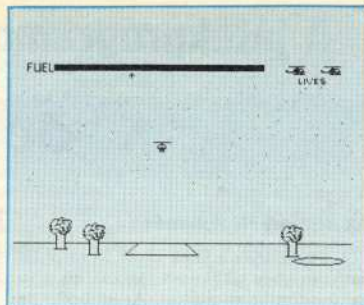
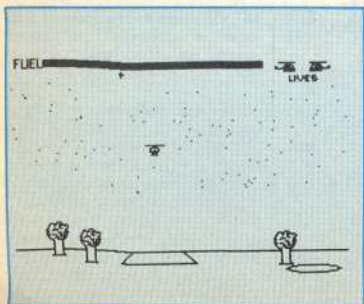
### Az üzemanyag-kifogyás mellékága

Ha elfogyott az üzemanyagunk, helikopterünk lezuhan. Ezt a 34. lista valósítja meg. Ezen belül váltakozva felrajzolódik a különböző irányba néző helikopter, egyre ejtőbb zuhanva, egyre mélyülő hangú zörrejjel. Amíg nem ér földet, ez ciklikusan ismétlődik. A földet érést gyorsan rezgő rövid dallam jelzi. Ha már lezuhant a helikopter, az 1480-as sorban ismertetettel azonos hanghatás, és az 1370-es sorral történő folytatás következik. A zuhanó helikoptert a 32. és a 33. ábrán láthatjuk.

A szükséges programmódosítások:

945, 965 nincs  
1275 STOP  
1305 STOP  
1365 STOP  
A futtatás:  
RUN  
majd  
CONT

36. ábra



35. ábra

és ismét  
CONT

### A főciklus folytatódik

A 35. listában látható a főciklus ezen szubrutinjának még hátralevő része. Ebben az életszámától függő helikopterhelyzet-rajzolás és néhány paraméter beállítása történik. Az ehhez tartozó rajzot a 34. és 35. ábra mutatja.

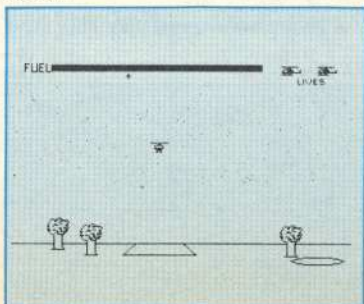
A futtatáshoz szükséges módosítások:

1275, 1305, 1365 nincs  
515 STOP  
545 STOP  
A futtatás:  
RUN  
majd  
CONT  
PC = 1  
eset volt beadva  
PC = 2  
értéket és  
RUN 490  
majd  
CONT  
után adódik a 36. és 37. ábra, melyeken látható, hogy a helikopterek a játék közben ide-oda forognak.

Záró cikkünkben a program még hátralevő részeit ismertetjük, és néhány tanácsot adunk az érdeklődőknek.

GALINA FERENC

37. ábra



Van egy MSX számítógépem, ami egy egyszerű gép, használhatatlan gépkönyvvel. Szívesen leveleznék vagy személyesen találkoznék MSX típusú számítógépek tulajdonosaival. Főleg programokat és dokumentációt cserélnék.

Az MSX nem nagyon gyakori gép az országban, azért remélem, hogy találni fog egy-két hasonló problémával küszködő társat.

Láposi Gábor, Nyíregyháza,

Széchenyi út 24.

Rendszeres olvasója vagyok az önök által szerkesztett lapnak. Az eddig megjelent számok mindegyike a birtokomban van. (Szerencsére.) Már két számban is megjelent a „Micsoda kiszolgálás” című cikk. Az utóbbiban épp a VIDEOTON TV-Computeréről volt szó. Egyetértés a cikk írójával, ez az állapot SIRALMAS.

Ezzel kapcsolatban:

A minap beugrottam Nyíregyházán a Centrum áruházba. Legalább 3 db TV-Computert látam a polcokon, illetve még egyet. És ez még nem minden. Ez az egy nem volt a pult mögött elbarikádozva, hanem egy külön kis asztalkán volt. Nem dobozban, hanem egy színes VIDEOTON tv teten! Ott pihent (működött) a gép, bárátságosan kacsingató cursorról. Kérdéseimre jóformán csak az árát tudták közölni. Ez egy kicsit elszomorított. Bár a gép csak aznap jelent meg a vásárlók előtt. Érzésem szerint az elkövetkező napokban sem fog sokat javulni ez az állapot. Így viszont nehéz lesz az eladóknak meggyőzni a tisztelt vásárlót, hogy csakis ezt a gépet érdemes most megvenni. Megkérdeztem, hogy kipróbálhatom-e ezt a gépet. A válasz csupán ennyi volt: TERMÉSZETESEN. Egy egyszerű kis programot írtam, ami távozásom után továbbra is futott.

Íde tartozik még, hogy aznap igen meleg volt. Egy szál ingben, illetve nadrágban mentem be a boltba. Egyértelműen látszott, nincs nálam 20 ezer forint. Mindezek ellenére még a mellékelt „dokumentációt” is megkaptam. Nem tagadtam, hogy jelenleg rendelkezem egy Spectrum géppel. A közeljövőben nem akarok vásárolni sem ilyen, sem más gépet. Érdeklődve nézték, hogy mit csinállok. Hisz ha vevő jön, meg kell vagy inkább kellene mutatni, mit is tud ez a gép!

Egy biztos, az eladók kedvessége már súrolja a nyugati szintet. De ez így még mindig kevés.

A VIDEOTON TV-Computer-ének az ára közben megváltozott, 12 000 Ft-ra csökkentették, és így az elfogadható árszintet nagyon megközelítette.

Bogyó Gábor, Hajmáskér,

Kossuth Lajos út 1. 8192

Véleményem szerint Magyarországon nagyon hiányzik egy összefoglaló prospektus, amely tartalmazná a számítástechnikai, műszaki és egyéb cikkek aktuális vámtarifáját, amely a lakosság széles köreinek is hozzáférhető lenne. Egy ilyen sokrétű (megújuló) tájékoztatóval szerintem a vámszabálysértések száma is nagymértékben csökkenne.

Legjobb tudomásom szerint a Vámhivatal folyamatosan aktualizált díjalappal dolgozik. A hivatal listáján a Magyarországon népszerű gépek szerepelnek. Ha valaki a listába fel nem vett gépet kíván behozni, akkor ennek vámértékét szakértővel állapítatják meg.

mégis azonos lehet léerni, hogy hányan olvasásuk! Talán egy kisebb közvélemény-kutatással el lehetne érni, hogy beállítható legyen a lap hasznos tartalma és a reklámok közötti kompromisszum. És ha már a bírálatnál tartok, hadd kérjek öntöktől egy kicsit gondosabb tordelést, ill. „lektorálást”, ugyanis elég sokszor találkozom olyannal, hogy az adott cikkhez tartozó gépi kódú lista elkeveredett, vagy egyik (talán a 85/5.) számukban ismét megjelent a HT 1080 Magyar hibaüzenet c. cikk, de a gépi kódú listán egészen más program volt. Ilyesmi könnyen elkerülhető lenne egy hozzáértő által elvlasott „kefe”-levonattal. (Ha ugyan még van olyan.) Csak egy csöpp figyelem kellene, és valóban minőségű újságról beszélne majd az olvasók, ha szóba kerül a  $\mu$  M!

*Köszönjük a dicséretet és a kritikát is. Amíg nem kerültem egy újsághoz közel, addig én is azt hittem, hogy lehet hibátlan lapot készíteni. Szerkesztőségünk minden egyes alkalommal megpróbálja — sajnos sokszor sikertelenül. Ami a reklámokat illeti, ezek tartják el a lapot, így az arány változása egyelőre nem várható.*

Toth László, Sopron,

Baross út 8/d 9400

A Mikroszámítógép Magazin című kiadvány alkalomszerű olvasója vagyok. Rendszeresen olvasom viszont a Markt&Technik Kiadó (NSZK) 64'er és a Happy Computer című újságokat.

Véleményem szerint mindkét, NSZK-ban megjelenő kiadvány tartalmaz olyan írásokat, amelyek a magyar mikroszámítógép-használók érdeklődésére számot tarthatnának.

Ezúton felajánlom önöknek közreműködésüket egyes német nyelvű írások magyarrá fordításához.

*Küldjön lehetőleg rövid és figyelemfelkeltő írásokat, a külföldi lapban megjelent cikkek kivonatát. Pontosan nevezze meg a forrást, ha új és érdekes írásokat válogat, megjelentetjük.*

Sakai Zoltán,

Szolnok 5000

Szerintem — a lap hangvételével ellentétben — egy számítógép nemcsak arra való, hogy programozzunk, kiismerjem a memóriafelosztást stb., hanem arra is (és ezt érzem a fontosságban), hogy felhasználjam; azaz bekapcsolom, és a bőveiben lévő felhasználói programok közül a számomra legmegfelelőbbet kiválasztva, a program által értelmesen alkalmazható menü és kommunikációs rendszer segítségével alkalmazom a programot. (A tv-t sem csak úgy tudom élvezni, ha ismerem az áramköröit, hanem úgy is, hogy bekapcsolom és nézem. Bár kétségtelen, ha „belepiskálalni” — javítani, feljesztani akarom, akkor ennyi nem elég.) Mindez persze akkor igaz, ha bőveiben válogathatok a jobbnál jobb programok között.

Szivemből szól.

Oláh Gyula, Debrecen,

Ispotály u. 15. III. 15. 4025

Örömmel értesültem arról, hogy megjelent az üzletben a TV-COMPUTER. Felkeltette az érdeklődésemet ez a gép, és szeretnék Önökötől néhány kérdésemre választ kapni. Eddig SPECTRUM-ot szerettem volna ven-

str. Aluminei Nr. 86. BL Sc A op. 18. Románia

A kolozsvári műegyetem gépgyártás-technológia szakos hallgatója vagyok. Nagyon érdekel a számítástechnika és a szoftver. Magam is rendelkezem egy Sinclair ZX Spectrum személyi számítógéppel.

Szeretnék levelezni és programokat cserélni magyarországi Spectrum-tulajdonosokkal.

*Remélem, lesznek levelező partnerei.*

Juhász József, Alsózsoltca,

Ságvári E. út 5. 3571

A Mikroszámítógép Magazin 1986. áprilisi számában Szilágyi Róbert andormaktályai olvasótársamnak a következőket válaszolta:

— A Magazin első száma már aranyéremt sem kapható!

Nos, én az „aranyat” érő első számot elküldeném cserébe annak, aki küld címre egy 1984/5-ös számot. Nekem ugyanis csak az hiányzik a sorozatból, az 1983-as beköszöntő számból kettőt is sikerült utólag keríteni, mert nincs mindenki gyűjtőszennvedéllyel megáldva.

Ezért kérem, az olvasói rovatban levelemet szíveskedjenek közzétenni.

*Közzététük.*

Lakatos Péter, Miskolc,

Klapka Gy. u. 36. 3524

A Mikromagazin régi olvasója vagyok. Néhány kivételével minden szám megvan nekem. Ezek az 1983-as, 84/1, 84/3, 85/5.

Egyre több a színvonalas cikk, de még mindig kevés a hardver és a szoftver. A „Diákrovat” nagyon tetszik, sok újdonság található benne. Írták, a diákyszerkesztőség számára tagokat keresnek. Én szívesen lennék „levelező” tagja.

A Mikromagazinnak is lehetne néhány különkiadása, az ÖTLET BIT-LET-jéhez hasonlóan, amiben a válogatott cikkeket, programok, hardverötleteket és az „Építünk számítógépet” sorozat jelenne meg!

*Nagyon sok tervünk van, pl. ilyen a  $\mu$  Könyvtár is, ezeknek a megvalósításához nem könnyű az anyagiakat előteremteni. A Diákyszerkesztőség vezetője: KARDOS ZSUZSA. Tel: 154-250/574.*

Gácsik Attila, Miskolc,

Klapka Gy. u. 56. IV. 3. 3524

1984 januárjától vagyok a  $\mu$ M törzsolvása, és szívesen olvasom lapjukat. Bár csak hobbi-ból foglalkozom a számítástechnikával és tudom, hogy ebben a korban (33 évesen) nem annyira fogékony, ill. képlékeny a felfogásom a programozás iránt, mint a most felövő nemzedéknek, mégis igyekszem minél jobban elsajátítani a programozás fogásait.

Visszatérve az újságra, természetesen adok hangot, ti. a havonkénti megjelenés előnyére válik, például a sorozatok valóban sorozatok lesznek, nem pedig szétszabdalt részcikkek. A lap tetszetős külalakú, a rovatszerkezetek és a cikkek tartalmát illetően is elégedett vagyok. Egyetlen fájó pont (gondolom nem csak nekem), ha túl sok, szinte semmit nem mondó, de annál tetemesebb helyet foglaló reklám található a lapban. Jó tudom, hogy az hozza a pénzt, de az olvasókat aligha! Egy lap sikerét

Olvasóink ismét nagyon sok jó gondolatot adtak. Köszönjük. Valamennyi olvasói levelet nyilvánvalóan nem tudjuk közölni, így a legérdekesebbekből válogattunk. Nagyon sokan kérnek — még külföldiről is — gépkönyveket, forrásnyelvi programokat, ROM leírásokat, különféle perifériák főleg magyar nyelvű dokumentációit. Ezeket — sajnos — nem tudjuk megszerezni, ezért elküldeni sem. Ha közérdekű kétről van szó, akkor a levelet közöljük, hátha valaki olvasóink közül segíteni tud.

Scabó László, Békés

Teleky u. 105. 5630

Amiért önhöz fordulok, az a következő: már sok szoftverleírást olvastam a lapban, s én szeretném, ha ez a szokás megmaradna. Két programról küldenek leírást. Nem akarom, hogy mások is úgy járjanak, mint én. Két hetembe került a programok megismerése. A programok a következők: Koalopainter és a Doodle. Mindkettőt Commodore 64-re készült, és grafikai segédlet. Esetleg, ha valaki meg akarná szerezni őket, szívesen állok rendelkezésére.

*Az utolsó mondat, remélem, azt jelenti, hogy ha valaki boltban megvásárolja a fenti programokat, akkor olvasóink szívesen ad tanácsot a programok megismeréséhez.*

Halász Péter, Budapest,

Irinyi u. 42. 1117

Az 1986. márciusi számban a  $\mu$ klub rovaton belül megjelent egy cikk „KOMPUTER HÍREK” címmel. Ebben egyebek között egy, a HCC Commodore szekciójára vonatkozó utalás is van, miszerint az egyetemi klub nem a HCC szekciója. A lapban többször megjelent, hogy a HCC különböző szekciói hol találhatók meg, de az információik tévesnek bizonyultak.

Kérem Önöket, hogy írják meg végre egyértelműen, hogy a különböző klubokat hol lehet megtalálni, és ki a vezetőjük.

*A HCC szekciói részére nagyon nehezen tudunk állandó klubhelyiséget találni. A HCC vezetése kéri mindazokat, akik 100—150 fős helyiséget tudnak ajánlani, asztalokkal és néhány zárható szekrénnyel, esetleg térítési nélkül, keressék meg a HCC elnökét (Dr. Simonyi Endre, telefon: 556-245 illetve titkárát (Diebel Dietrich, telefon: 296-600/166), vagy bármelyik szekció vezetőjét. Az egyes szekciók — most — az alábbi helyeken tartják összejöveteleiket:*

Aircomp Homelab: Barabás Dezső, telefon: 176-487/22 mellék, Belvárosi Művelődési Ház, V. Molnár u. 9., minden kedden 6—8 óráig  
Apple: Diebel Dietrich, 296-600/166  
Báthori u. 16.: a hónap 1. és 3. keddjén, 3 óra-  
kor

Commodore: Dr. Simonyi Endre, 556-245, helyszín változó. Simonyi felhívni.  
M 6800: Dr. Simonyi Endre, 556-245, helyszín változó. Simonyi felhívni.  
HT: Gyöző Miklós, telefon: 692-800/név  
TRS—80 változó. Gyöző Miklós felhívni.  
SINCLAIR: Hivessy Ferenc, 403-990, Szellőző Művek, XI., Építész u., hétfő 18—20 óráig

ni; esetleg C 16-ost. Még csak néhány hónapja ismerkedem a számítástechnikával, s ezért elsősorban olyan gépet szeretnék venni, amelyhez játékprogramok is kaphatók, hisz egy magamfajta számítógép-kezelőnek fontos az, hogy amíg elfogadható programokat nem tud írni, addig is örömmel használhassa a gépet. (Sokan az első néhány kudarcot elfordulnak a gépektől.) Szóval a lényegre térve: vannak-e, vagy inkább várhatók-e a közeljövőben elfogadható programok a géphez? Kaphatók-e perifériák a géphez, adnak-e ki e tipushoz részletes ismertető könyvet? (A gépkönyvet az üzletben átfutottam, de eléggé felületesnek tűnik.) Elterjed-e önök szerint ez a gép nálunk? Nem kell-e attól tartanunk, hogy az első széria esetleg gyengébb minőségű? (Sokaknak okozott ez kellemetlenségeket az AIRCOMP 16-osnál.)

Én beszéltem a Videoton illetékes vezetőivel, akik azt mondták, hogy viszonylag rövid idő alatt a TV-Computer-t minden szempontból (ár, programellátás, vevőszolgálat stb.) versenyképessé teszik a nálunk kedvelt külföldi gépekkel. Levélét elküldtem a VT kereskedelmi igazgatójának.

Honti József, Csákvár,

Május 1. u. 11. 8083

Mint már egy korábbi levelemben beszámoltam róla, közsegünkben is megalakult a Mikroklub. A klub még nincs egy éves, de már bizonyítja életképességét. Különösen a fiatalok körében nagy az érdeklődés a klubmunka iránt.

Örülök a klub megalakulásának, a küldött programot átadtam az illetékes rovatvezetőnek.

Köszönöm a leveleket, valamennyi olvasónak kellemes nyaralást kívánok

KOVÁCS GYÖZÖ

## Olvasónk javaslója

A legnépszerűbb mikrogépeknek (jelenleg a Commodore, a Spectrum, a HT és a Primo) nyisanak egy-egy dupla oldalt, ami programoktól kezdve híreik mindent tartalmazhatna, de csak azzal a számítógéppel kapcsolatban. Szerintem ugyanis lapjuk vásárlóinak zömét éppen ezek programozói alkotják.

Ennek a két oldalnak a kezelését rá lehetne bízni egy-egy működő klubra. Ők felhívásokat intézhetnének az adott gép felhasználóihoz, hogy küldjenek be saját készítésű, általuk érdekesnek tartott programokat. Ezeket ők maguk tesztelhetnék, a szerzője engedélyével módosíthatnák, esetleg bővíthetnék. Sokakat a cikk megírása riaszt el a publikálástól. Helyettük vállalhatnák ezt a munkát is. Ez a megoldás tehermentesítene a lap dolgozóit, és lehetővé tenné több program megjelenését, ami — ha szabad környezetből kiindulva általánosan — az olvasótábor kívánsága.

PIVARNYIK ATTILA

Az ötletet jónak tartjuk, várjuk a vállalkozó klubok jelentkezését. A szerkesztőség.

Legutóbbi levelemből a 86/1. számban a reméltnél többet, de kevesebbet is közöltek. Ugyanis az ÁRAK miatti háborgásom Kovács Győző kedves, csitító szavai ellenére nem szűnt meg, de azt hiszem, öböne sem.

Idézet a 86/2. szám 86 kiállítás bevezetőjéből: "... a számítástechnika tömeges elterjedésének még ma is az egyik legnagyobb akadályja a magas ár."

Hozzátenném, a **büvészkulcsokkal** kalkuláltai hazai ár:

— Az Ötletben Angyalosi László még ma is várja tán, hogy jelentkezzenek azok a cégek, amelyek neki — és az olvasóknak — meg tudják magyarázni, miként lehet az ÁPISZ-nál 8000 Ft alatt haszonnal értékesíteni a C-16-os gépeket, másoknál viszont csak 10-12 ezer Ft-ért.

— Az SFD 1001 (1,05 Mb-os) lemez meghajtót a Centrum Nagycser. V. a kisker. egységeknek egy C-610-es alappépre épülő konfiguráció tagjaként 49 500 Ft-ért árulja, ugyanakkor a Vámhivatal az ajándék- és utasváltárifa alapján 60 000 Ft vámalappal számol, és — most jön a büvészkulcs — a Novotrade 2C áruház telefoninformációja szerint ők 150 000 Ft-ért árúsítják! A nyugati árat nem ismerem, csak sejtem, hogy a C-128 mellett árult hasonló gép 7-8000 osztrák schillingjehez nem állhat messze. (A Herlango téli katalógusában 7990,- Sch)

Kovács Győző azzal biztatott, hogy ezek reálisok! Ahogy vesszük.

Egy közületnek nem jelent akkora különbséget, hogy valami, amit meg akar valósítani az 100 000 vagy 150 000 Ft-jába kerül, így mint piaci realitást könnyen elfogadja az így képzett árat. Ennyiben tehát reális.

Az ilyen piacon lezajló események azonban igen messze állnak a számítástechnika valódi tömeges elterjesztésétől. Legfeljebb akkor, ha már mint leirt eszközök kerülnek majd mondjuk ajándékozás útján a klubokhoz (5-10 év múlva).

A sajtó szavával viszont igen eredményesen lehet — s szerintem kell is — küzdeni a „büvészkulcsos” árképzés ellen, például az ilyen árképzés nyilvánosságára hozatalával! Tán várjuk meg a sajtótörvényt!

A sajtónak az ilyen magatartása már eredményesen megmutakozott — a kormányserveket is döntésre készítette —, amikor pl. a vámkulcsokat a mikrók és videók körében jelentősen, de sokak szerint még nem elégségesen lecsökkentették.

A Vámörség parancsnoka a HVG-ben nyilatkozott úgy, hogy nem a vámok verik fel az árat. Hát ez igaz is, meg nem is. A kialakuló — bizonytalan — árákban még ma is jelentős tétellel szerepel a vám. A vámhivataloknál béso használatra kiadott lista, mely a belföldi forgalmi érték

— vámalap — meghatározásait a legismertebb árukra tartalmazza, egyrészt nem vagy igen ritkán kerülnek nyilvánosságra, ugyanakkor ritkán kerülnek változtatásra. Ezeket a változtatásokat kapd a nyugati kiskereskedelmi árákban esik — lehetne lehet tapasztalni, de egy-egy konkrét géptípus esetén felvéteknél valószínűs „felvétezi idő” valószínű meg. A Sinclair Spectrum például közel egy árbán volt a C-64-gyel még egy évvel ezelőtt — cca. 4000 schilling —, most 1500-1700 Sch. A vámalap viszont még tudomás szerint most kb. 21 000 Ft. (kiskereskedelemben 14-16 000 Ft-ért is látni).

Az összefüggés a belföldi árákkal pedig nyilvánvaló. Ha a vám 21 000 Ft után kerül kiszabásra, az utasforgalomban behozott gépet — állami szervezett import alja van — a tulajdonos nem fogja, ill. nem akarja ennél olcsóbban eladni.

Téhat ha a vámszervek által használt lista nem követi folyamatosan a külföldi árcsökkenéseket, nem fogja engedni ennek jöteknő begyűrésését sem! Így már nem igaz a Vámörség Országos Parancsnoksága állítása sem, a vámok áralakító hatásáról.

Nem bonyolodom bele a valutaárfolyamon történő számítások kérdésébe, mert az igaz, hogy esetenként az exportban nem lehet az árfolyamoktól érvényesíteni, viszont a valutáinség miatt az így létrejövő üzletkötésekre is szükség lehet.

A vámot, illetve a devizagazdálkodást érinti az a kérdés is, hogy az utas 25 000 Ft belföldi forgalmi értékhatárig hozhat árut. Ez sem követi egyébként a VI. ötéves tervben megvalósult 4,3-8,8%-os inflációt, de a fentebb írt lassú „listaváltozás” folytán előállt a múlt évben egy olyan helyzet, hogy a jogilag nem kifogásolható, de a közvélemény megítélése szerint annál inkább kifogásolt módon a C-64-es gépet behozók — a vámalap akkor 27 000 Ft volt — nem kapták meg az utólagos behozatali engedélyt, s gépek jönszereladásra került, mégcsak előveteli kény sem biztosítva részükre. (A Heti Világáadás olvasói rovata is írt ilyenekről.) Ezt kivételesen megítéltek azok az „ajándékozó” okiratok, melyek leegyszerűsített eljárásban és magasabb értékhatárral vitték a jogon kívüli útra az embekeket. Embereken nem a spekulánsokat, hanem azokat értem, akik saját fejlődésük, rajongásuk érdekében döntöttek az utazásuk alatt a gépvásárlás mellett.

A spekulációk visszafogása véleményem szerint sokkal inkább ez utóbbi módszernél kellene, semmint a 25 000 Ft-os értékhatár változtatlanul hagyásával. A fent már írt listamérvesség folytán könnyen elképzelhető, hogy a szabályosan szerzett valuta felhasználásával — illetve annak kb. 50%-a ami vásárlásra fordítható — ennél magasabb belföldi forgalmi értékű árut tudunk beszerezni.

LUKÁCS ATTILA

## Keresünk megvételre jó állapotban lévő

**VIDEOTON gyártmányú  
VDT 52103, VPC, VPPC  
display-eket.**

Villamosenergiaipari  
Kutató Intézet

Ügyintéző: Papp György osztályvezető  
Telefon: 178-698

Newman, W. M. —

Sproull, R. F.:

**Interaktív számítógépes grafika (Budapest, 1985. Műszaki Könyvkiadó, 492 oldal. Ára: 165,— Ft.)**

A számítógépes grafika kapcsolatot teremt a számítógép és az ember között: a számítógép egy grafikai képet állít elő sornyomatóval papíron vagy a megjelenítő képernyőjén az ember számára, amelyet az vizuálisan észlel. A nemzetközileg is elismert szerzőpáros kiváló szakkönyve részletesen tárgyalja a számítógépes grafika hardver- és szoftvereszközait, a használatos számítástechnikai és matematikai módszereket, és foglalkozik az ilyen rendszerek fejlesztésével és alkalmazási lehetőségeivel.

Vitray Péter:

**Hetedhét Atari 800LX (Budapest, 1985. Novotrade RT. 151 oldal. Ára: 92,— Ft.)**

A Hetedhét... című sorozat célja, hogy a különböző számítógéptípusokat a teljesen kezdő felhasználók számára érthető nyelven, szórakoztató formában, szellemes, egyszerű programok segítségével mutassa be — mindössze hét nap alatt. E célját a könyv sikerrel teljesíti, amikor a gép üzembe helye-

zésétől kezdve, a programírás alapismeretein keresztül, a BASIC nyelvű utasítások gyakorlásával a felhasználókat egyszerű játékögramok írására is képessé teszi. A kötetet a gép használatát és a programírást segítő táblázatok egészítik ki.

Pomper János:

**Példák a mikroszámítógépek vállalati alkalmazására (Budapest, 1985. COMPORGAN, 375 oldal. Ára: 370,— Ft.)**

A kötet a vállalatok vezetési, szervezési, irányítási feladatainak mikroszámítógépes megoldásaival foglalkozik. 27 oktatósi célú programot tartalmaz, melyek egy 60 modulból álló, bármely mini- vagy mikroszámítógépre implementálható, ipari alkalmazásra készült programcsomag erősen leegyszerűsített változatai.

A szerző a világ 500 legnagyobb vállalatánál végzett nemzetközi felmérés eredményét hasznosította annak megállapítására, hogy milyen vezetési problémák megoldásában használnak leggyakrabban számítógépet, és milyen módszereket alkalmaznak a problémák megoldásában. Az egyes fejezetek a megoldások módszereit tárgyalják.

Végül a szerző a modellalkotás legfontosabb kérdéseit foglalja össze, és táblázatot közöl a programok méreteiről is.

# A NOVOTRADE RT

## az alábbi szoftverek

### bemutatóját tartja

## a Hotel Rege

„KÁRTYA” termében  
(Bp. II. ker., Pálos u. 2.)  
de. 10 órától,  
melyre  
minden érdeklődőt  
szeretettel várunk!

### IX. 1.

Főkönyv- és folyószámla-  
könyvelési rendszer  
bemutatója

### IX. 2.

Készletgazdálkodási  
rendszer

### IX. 3.

Bér- és munkaügyi rendszer  
Személyzeti rendszer

### IX. 4.

Alap-, illetve fejlesztői  
rendszerek  
és lokális hálózatok  
IBM kompatibilis  
gépekből kialakítva

### IX. 5.

Állóeszköz-gazdálkodás

Személyszámítógép-javítás, karbantartás  
közületeknek, magánszemélyeknek.  
Egyedi megrendelés alapján  
kiegészítő berendezések gyártása.  
Pl. Sinclair fényceruza, Joystick Interface,  
oktatási intézménynek kabinet kialakítása.

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

**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

# NOVOTRADE

## Kínai szénbányákban

Négyezer darab professzionális mikroszámítógépet vásárolt Kína az amerikai Intel cégtől. Ezeknek az IBM PC-vel kompatibilis gépeknek csupán az egynegyede érkezik összeszerelt állapotban. A háromezer komplett alkatrészkészletet a Hongkong mellett fekvő sen-csieni különleges gazdasági övezet egyik gyárában szerelik össze. A gépeket elsősorban a bányászati ágazatban, ezen belül is a szénbányászatban kívánják alkalmazni.

## 10 Mbájtos tároló áramkör?

A Japán Tudományos és Technológiai Ügynökség három éves kutatási tervet indított, hogy kifejlesszenek egy olyan nagy sebességű memóriamorzst, amely helyettesítheti a hagyományos mágnesszalagot és mágneslemezt. A 10 Mbájtos tárolási kapacitású SRAM integrált áramkör a Citizen órágyár fejlesztési japán kormány 2,5 millió dolláros hozzájárulásával. Az első kereskedelmi példányok megjelenése 1990-re várható, és körülbelül 100 dollárba fognak kerülni.

A szakemberek szerint az új tároló forradalmasíthatja a hordozható mikroszámítógépek piacát, ahol a lemezes gépek nem tudnak igazán meggyökeresedni.

## Szimat-robot

A fejlett ipari robotok már „látnak” és „hallanak”. Az oxfordi Austin autógyárban viszont már olyat is kifejlesztettek, amely „szaglászik”, s jelzi is az észlelt szag helyét. A robot feladata annak megállapítása, hogy jól zárnak-e az új kocsik. Először héliumgázt fecskendeznek a kocsikba, és ahogy azok a futószalagon tovaagódnak, a roboton lévő gázérzékelő „végigszagolja” a karosszériát, elsősorban az ajtók és az ablakok mentén.

## Iskolagépek a Szovjetunióban

A reál tagozatos szovjet tiszosztályos iskolákban már 5-10 db iskolaszámítógép működik. Ezek részben szovjet gyártmányú Agatok, részben japán Yamaha háziszámítógépek.

Mint ismeretes, az Agat gépet 1983 novemberében jelentették be a Szovjetunióban; ez az Apple II funkcionális megfelelője. 1985 végére már megtörtént a gyártásba vétel, de sajnos mágneses háttértár nélkül. Így ezeken a gépeken egyelőre csak „pötyögni” lehet.

Az Elektronorgtechnika tavaly hirdetett nemzetközi versenydíjat az iskolaszámítógépek importjára, amelyen nagy esélyesként részt vett a Sinclair ZX-Spectrum gép is. A győztes — óriási meglepetésre — a Yamaha nevű japán háziszámítógép lett. Az értesü-

lések szerint a 8 bites gépet képernyővel, hájlékonylemez tárolóval és nyomtatóval 314 USA-dollárért szállították a japánok. Az üzlet érdekességét növelte, hogy az amerikai Microsoft cég által kidolgozott, a különböző típusú számítógépek közötti teljes kompatibilitást megteremtő MSX (Microsoft Extended Basic is) operációs rendszer fut a gépeken. A jelentések szerint az idén összesen 4000 db Yamaha-típusú gépet szállít a Nippon Gakki cég a Szovjetunióba.

## A Proper 8 és 16 után 32

Már a 16 bites IBM PC-vel kompatibilis gépeket „kinövő” felhasználókat készíti fel a 32 bites gépre való áttállásra az SZKI új „kétarcú”. Proper 16/32/MT-típusú személyi számítógépe. A 16 bites mikroprocesszor biztosítja a gépnek a Proper 16-tal való kompatibilitást, hiszen fut rajta a Propos operációs rendszer 3.2 verziója, a 32 bites mikroprocesszor pedig a Unix operációs rendszert használják. Ez utóbbinak 16 Mbájtos címezhető memóriaterülete van, a valós tár is 0,5-2 Mbájttal változik. A gép műveleti sebessége mintegy háromszorosa a Proper 16-énak.

Az új géptípus jól hasznosítható a többfelhasználós ipari és mezőgazdasági alkalmazásoknál, a mérnöki tervezői munkahelyeken és az irodai automatizálási feladatoknál.

## Morze adó-vevő

ZX-Spectrum gépre dolgozta ki a HG5VQ hívójelű Csapó István a morze adó-vevő programját. Az adás sebessége tetszőlegesen beállítható 30-150 karakter/perc intervallumban. A program a begépet és a képernyőn lévő karaktereket morzejelekké alakítja. Az adni kívánt szöveget folyamatosan gépelléjük. Vétel esetén a program folyamatosan méri az adóállomás adási ütemét, és ennek megfelelően módosítja a vételhez szükséges időadatokat. Veszélyre hamar rááll a megfelelő sebességre, és ott megbízhatóan dolgozik.

## Telexkapcsolat a számítógéppel

Sok vállalat a telexgépet nemcsak üzenetváltásra használja, hanem feldolgozandó adatot is küld át. Az eddigi gyakorlat az, hogy a fogadó cégnél valaki számológéppel dolgozza fel ezeket az adatokat.

A Háziszámítógép-építők Klubja erre a feladatra elkészítette a ZX-Spectrum és a Commodore 64 gépre a megfelelő programot. Első része az úgynevezett „lefűlő” szoftver, ami a telexgépen érkező adatokat automatikusan, párhuzamosan a számítógépbe küldi. Ezen az üzenetváltás befejezésekor az eredmény rögtön megjelenik. Emellett lehetőség van arra is, hogy a telexgépről lejövő lyukszalagot dolgozza fel a gép.

## MEGJEGYEZZÜK...

A lap ez évi 7. számában „Jelentkezem” címen Baky Miklós tollából egy rövid humoros írást közöltünk. Az olvasók leveleiből úgy tűnik, hogy az írást sokan félreértették, és úgy hitték, hogy ezzel a dr. Simonyi Endre által kiírt pályázatot akarjuk lejártni.

A szerkesztőség kijelenti, hogy ez félreértés. A lap az amatőrök szellemi termékeinek értékesítését elősegíteni kívánó pályázatot nem akarta támadni, sőt sajnálatosnak tartja, hogy az eddig mindössze 8 pályázat érkezett, amelyből kettő látszik olyanoknak, amiből „talán valami lesz” (dr. Simonyi közlése). Úgy véljük, hogy az ilyen pályázatokhoz még nem szoktunk hozzá, de hát minden kezdet nehéz.

A szerkesztőség

## COMMODORE-tulajdonosok figyelmébe!

- C16, C116 RAM memóriájának bővítése 64 k-ra beépített bővítővel: BASIC munkaterület 60671 bájtra nő
- videoadapter beépítése C16-ba
- magyar betűkészlet beépítése C16, C116 és PLUS/4-be.

### Iskoláknak engedmény!

Vidékieknek egy utazással!

Kérje részletes tájékoztatónkat!

EL—KA GM. 346-933 délután.

Levél cím: 1141 Budapest, Lipótvár u. 59.

EL—KA Elektromos Kapuszerviz és Automatika

Tervező, Kivitelező Gazdasági Munkaközösség

1141 Budapest, Lipótvár u. 59. Németh Tibor, 574-811/192.



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

- főkönyvi könyvelés,
- költségfelosztás,
- állóeszköz-nyilvántartás,
- rendelés-nyilvántartás,
- bérelszámolás,
- adóelszámolás.

1027 Budapest, Fő u. 68.  
Telefon: 154-090

*Bővebb  
felvilágosítással  
készséggel állunk  
rendelkezésükre!*

## 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

Egészségügyi Elektronikai Gm  
1045 Bp., Erzsébet u. 14. X. 59.

Telefonügyelet: Grósz Andor, 632-720 (9—17 óráig)

**ORTEKON**



**ORTEKON TERVEZÉSI ÉS FEJLESZTÉSI gazdasági munkaközösség**  
1036 Bp., Árpád fejedelem útja 69.

Telefon: 154-250, 887-861

Bányai

Vállaljuk:

\_\_\_\_\_

\_\_\_\_\_

- elektronikus (digitális) vezérlőegységek,  
szabályozókészülékek és -rendszerek ter-  
vezését és gyártását;

\_\_\_\_\_

\_\_\_\_\_

- CNC-vezérlés komplett felújítását;

\_\_\_\_\_

\_\_\_\_\_

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

\_\_\_\_\_

\_\_\_\_\_

- a gyártás-előkészítés és elszámolás irányí-  
tási és adatfeldolgozási feladatainak meg-  
valósítását több munkahelyes számító-  
géprendszer alkalmazásával.

**A megfelelő referenciákról személyesen adunk tájékoztatást.**

# DÖNTÉSÉT MEGKÖNNYÍTJÜK!

**Széles választék, kedvező ár,  
sokoldalú szolgáltatás!**

## 8 bites ALAPKONFIGURÁCIÓK

PROPOS—8 (CP/M vagy ezzel kompatibilis) operációs rendszerrel

**M08X  
PROPER—8**

**250 000,— Ft  
250 000,— Ft**

A konfigurációk érdekeltségi alap nélkül megvásárolhatók

## 16 bites ALAPKONFIGURÁCIÓK

PROPOS—16 (IBM PC XT vagy ezzel kompatibilis) operációs rendszerrel

**PROPER—16/m**

**266 000,— Ft**

**PROPER—16/W1**

**439 000,— Ft**

**PROPER—16/W2**

**549 000,— Ft**

**PROPER—16/MEGA**

**699 000,— Ft**

**MÁTRIXNYOMTATÓ (MP—80,  
kvázigrafikus jelkészlettel)**

**30 000,— Ft**

## RENDELÉSEKET VESZÜNK FEL 1986. II. FÉLÉVRE

**PROPER—16/MT (IBM PC AT kompatibilis operációs rendszer)**

**850 000,— Ft**

**PROPER—16/G (nagy teljesítményű grafikus munkahely)**

**1 500 000,— Ft**

## SZOLGÁLTATÁSOK:

12 hónap garancia, további 12 hónapra a hardver árának 6%-áért  
hardver-szervizszolgálat  
Világ színvonalú perifériák  
Széles körű szoftverválaszték  
Kulcsrakész alkalmazói rendszerek  
Hálózatba integrálási lehetőség  
Országos szervizhálózat

## KÉRJE RÉSZLETES ÁRJEGYZÉKÜNKET!



Számítástechnikai Kutató  
Intézet és Innovációs  
Központ  
Budapest 1251 Pf. 19.

### Információ:

SCI—L Számítástechnikai  
Informatikai Fejlesztő Leányvállalat  
Budapest, Iskola u. 10. 1011  
Telefon: 153-204

## AZ SZKI STABIL PARTNER!

SCITEL Számítástechnikai  
Fejlesztő Leasing Leányvállalat  
Budapest, Donáti u. 35—45. 1015  
Telefon: 350-180/142



Várjuk Önöket az ŐSZI BNV F 2-es pavilonjában!  
Keresse a FOTOELEKTRONIK EMBLÉMÁT!



```

10 REM * * * * * EZ C64 PROGRAM * * * * *
20 REM * * * * *
30 B$ = " SYSTEM SZERVEZÉSI VÁLLALAT
40 C$ = " 1055 BUDAPEST, SZENT ISTVÁN KRT. 11.
50 REM * * * * *
60 PRINT " ♥ " PRINT TAB(1/); " MUNKAHELYE DOLGOZHATNA SZERVEZETTEBBEN?
70 REM * * * * *
80 PRINT: TAB(5/); " I (IGEN), N (NEM) A BEÍRANDÓ!
90 REM * * * * *
100 GET A$: IF A$ = "N" THEN 140
110 IF A$ = "I" THEN 180
120 GOTO 100
130 REM * * * * * HA N (NEM) * * * * *
140 PRINT " ♥ " PRINT TAB(5/); " REMEKI EZ A LEGJOBB MUNKAHELYI
150 PRINT TAB(4/); " HÍVJANAK MEG TAPASZTALATCSERÉRE!
160 PRINT B$; C$; FOR I=1 TO 1E3 NEXT I: GOTO 140
170 REM * * * * * HA J (IGEN) * * * * *
180 PRINT " ♥ " PRINT TAB(14/); " AKKOR AJÁNLUK
190 PRINT TAB(5/); " MIKROSZÁMÍTÓGÉPPAL TÁMOGATOTT
200 PRINT TAB(7/); " MUNKA-, ÜZEM-, VÁLLALAT- ÉS
210 PRINT TAB(6/); " INFORMÁCIÓ RENDSZERSZERVEZÉSI
220 PRINT TAB(3/); " -VEZETÉSI (DÖNTÉSELŐKÉSZÍTÉS STB.)
230 PRINT TAB(4/); " -GAZDÁLKODÁSI (ANYAG-, MUNKAÜGY-,
240 PRINT TAB(0/); " ÁLLÓESZKÖZ-, MENETLEVÉL-FELDOLGOZÁSI STB.)
250 PRINT TAB(15/); " -TERMELESI
260 PRINT TAB(4/); " (BERUHÁZÁSI, KARBANTARTÁSI STB.)
270 PRINT TAB(1/); " MÓDSZEREINKET, RENDSZEREINKET, PROGRAM-
280 PRINT TAB(6/); " CSOMAGJAINKAT ÉS MIKROGÉPES
290 PRINT TAB(5/); " PROGRAMOZÁSI SEGÉDESZKÖZEINKET,
300 PRINT TAB(2/); " OSZTOTT ADATFELDOLGOZÁSI RENDSZEREK
310 PRINT TAB(6/); " TERVEZÉSÉT, FEJLESZTÉSÉT ÉS
320 PRINT TAB(6/); " ÜZEMELTETÉSÉT IS VÁLLALJUK,
330 PRINT TAB(6/); " REFERENCIAKKAL RENDELKEZÜNK!
340 PRINT TAB(0/); " RÉSZLETESEBB ISMERTETÉS E LAP OLDALÁN
350 PRINT TAB(2/); " TALÁLHATÓ. AZ ISMERTETETT MÓDSZEREK,
360 PRINT TAB(5/); " RENDSZEREK, PROGRAMCSOMAGOK ÉS
370 PRINT TAB(7/); " ESZKÖZÖK MEGTEKINTHETŐK AZ
380 PRINT TAB(4/); " ORGTECHNIK HUNGÁRIA BUDAPEST '86
390 PRINT TAB(3/); " SZAKKIÁLLÍTÁSON, AHOZ SZAKEMBEREINK
400 PRINT TAB(2/); " AZ ÉRDEKLŐDŐK RENDELKEZÉSÉRE ÁLLNAK
410 REM * * * * *
420 FOR I=1 TO 5E4 NEXT I: PRINT " PRINT B$; C$; FOR I=1 TO 1E4; NEXT I: GOTO 140
430 REM
440 REM

```

