



**Kedves BIT-LET híveink!**  
**Van egy jó hírünk, meg egy rossz!**  
**Kezdjük a jóval: elkészült a nagy dírral**  
**behangozott LOGO program. Szekfű**  
**András farsangi ajándékát innentől a**  
**lap közepéig találják. (Szép hosszú**  
**program, értelmes kommentekkel!) A**  
**rossz hír pedig, hogy mindez egyelőre**  
**csak Spectrumon futtatható!**  
**Amikor a LOGO terjedelmét ijesztőnek**  
**találják, gondoljanak gyerekeikre, és**  
**vessék bele magukat!**

A BIT-LET 1983. októberi számában azt írtuk, hogy a LOGO számítógépnyelv elemeit szimulálni lehet bármely ismert mikroszámítógépen. A következő leírás alapján, akinek kedve van, barkácsolhat magának Sinclair ZX Spectrum számítógéphez afféle CSM-Logo-t (CSM: csináld magad!) saját és gyerekei örömére. A CSM-Logo nem versenyezhet a nyelv profi változataival, de az nem is célja. Mit tud és mit nem tud? A LOGO sajátosságai közül tudja a teknőc-grafikát, tudja a teknőcöt közvetlen parancsokkal irányítani, vagy eljárás-programmal, ismételtető módon vezényelni. A programozás módja az októberi cikkünkben ismertetett eljárásokra (procedúrák) épül. Kezelési rendkívül egyszerű, soronként írhatjuk be magyar nyelvű parancsainkat. (Pl.: ELORE 30) nem felejtkezve el a szó és a szám közötti egy szóközzel (Space). Utána nyomjuk be az ENTER gombot. Ha parancsaink végéhez értünk, írjuk be: VEGE. Erre a gép „elindítja” a teknőcot, amely „bejárja” a parancsok által előírt utat.

A BIT-LOGO-nk egyelőre a következő parancsokat ismeri:

**ELORE** – lépésszám

**HATRA** – lépésszám

**JOBBRA** – fok

**BALRA** – fok

**ISM** – hányszor

**ISMVEGE**

**TOLL**

**TOLLNE**

**KOZEP**

**SZIVACS**

**FELEJT**

ELORE HATRA esetén egy-egy lépés egy-egy képpontnyi távolságnak felel meg a képernyőn.  
 JOBBRA BALRA esetén a fordulat mértékét fokban kell megadnunk.  
 ISM ISMVEGE a két parancs közötti programrészt annyszor ismétli teknőcünk, ahányas számot az ISM után írunk (egy szóközzel).  
 TOLL TOLLNE indulásnál a TOLL parancs van érvényben, azaz a teknőc

vonalat húz maga után a képernyőn. Amint a TOLLNE parancsot megkapja, „felemeli tollát” és vonalhúzás nélkül halad. Újabb TOLL utasítás után megint nyomot hagy.

KOZEP erre az utasításra a teknőc vonalhúzás nélkül a képernyő közepére ugrik, anélkül, hogy addigi rajzunkat letörölné.

FELEJT parancsunkra nemcsak rajzunkat töröljük le, de a teknőc addig betáplált összes tudományát elfelejti. Viszszaugrik a képernyő közepére, s kezdhetünk mindent elölről. (Lásd majd: Hidegindítás.)

Ha azt akarjuk, hogy utasításainkat a teknőc megjegyezze és később is végrehajtsa, írhatunk eljárásokat. Minden eljárás a LEGYEN (vagy rövidítve LE) szóval kezdődik, melyet szóköz után az eljárás általunk szabadon választott

neve követ. A HAZ nevű eljárás meghatározását például (lásd BIT-LET 1., október) úgy indítottuk, hogy LEGYEN HAZ. Az eljárás végén be kell írunk a VEGE szót. De a gép az eljárást ekkor még csak „megjegyzi”. Az eljárás végrehajtása úgy történik, hogy „parancsba adjuk” az eljárás nevét, majd beütjük a VEGE szót. Ekkor a teknőc az eljárást végrehajtja. Mindaddig, amíg a FELEJT parancsot nem használjuk (vagy ki nem kapcsoljuk a gépet...), az egyszerű beadott eljárások nevükön szólíthatók és végrehajthatók.

A CSM-Logo rekurzív, azaz egy-egy eljárás újra „meghívhatja” saját magát. CSM-Logonk nagy hátránya a profi változatokhoz képest, hogy lassú, mivel BASIC nyelven írjuk, nem gépi kódban. E hátrány azonban két előnyt is jelent: a lassú teknőc mozgását könnyen követhetjük a képernyőn, és a BASIC-ben írt programot minden Spectrum-tulajdonos olvasónk megértheti, mert nemcsak a listát közöljük, hanem azt is, hogyan működik. A most közölt CSM-Logo változatnak még további hiányosságai is vannak. Nincs benne listafeldolgozás, ami az „igazi” LOGO-ban alapvető. De egyelőre nincsen benne elágazás sem (lásd a BASIC-ben: IF-THEN), nincsenek változók és nem végez számtani-logikai műveleteket sem. Miért írjuk, hogy „egyelőre”? Azért, mert CSM-Logonk építőköveken épül (bikkfanyelven: moduláris szerkezetű), azaz minden olvasónk saját ízlése és szükségletei szerint bővítheti, tökéletesítheti. A most közölt első változat tehát csak az alap. Ez a változat belefér a 16 K-s Spectrum tárolójába is. A bővített változatokhoz már a 48 K-s memóriára lesz szükség. Első változa-

(folytatás a 20. oldalon)

**BELÜLRŐL**

- 18 **Hirodal** – kiderült, hogy nem csalás, nem átlom ... Gigantikus új hírek a gigás lemezről
- 19 27 30 **Posta** – egy levélírónk rajtunk kéri számon olvasótársa levelét
- 20 **CSM-LOGO** – Szekfű András a BIT-LET történetének eddigi legizgalmasabb szoftvertermékét adja közre
- 24 **Mi mennyi?** – egy régi vicc új poénnal, de az igazi poén képes társasjátékunk, amelyben a start a határ, a cél a biományi
- 26 **Vallató-cska** – átlagosztályzat nincs, s ráadásul nem is számítógépet vizsgáltak inkvizitoraink, hanem egy gmk új berendezését
- 29 **Néhány jó tanács** – azoknak, akik unják már a ZX-81 megbízhatatlanságát, de nincs pénzük a Vallatócskában megismert készülékre
- 31 **Sorvezető** – végre megszólal egy diák is, s amit közöl, nem is akármilyen
- 32 **Félgép-nyerő** – lehet, hogy az olvasók unnak bennünket, de mi még mindig és újra lámpáscázunk ...

# HÍRADAL

**Kedves olvasónk!**

*Az újságyszerkesztés és -összeállítás nehéz munka. Fárasztó. Ebben a kemény munkában jól jön egy kis humor. Ne haragudjon tehát ránk komolytalanságainkért. Mindaz, amit ezen az oldalon (zárójelben) talál, ne vegye komolyan (sic!), azaz bocsánat (vicc!)*

## HÍREK a „gigás” lemezről

A lézer-videó technika továbbfejlesztésével készült új Shugart lemezről már hírt adtunk a BIT-LET 1. számában. Akkor némi kétke-  
dést tanúsítottunk a dologgal kapcsolatban. Most újabb adatokat találtunk a Practical Computing januári számában. Íme: Kapacitás: 1 Gbyte (25 Kbyte/sáv); elérési idő: 1 msec (sávról/sávra), 100 msec (átlag), 200 msec (maximum); átviteli sebesség: 5 Mbit/sec. Mérete 177x477x610 mm; súlya 22,7 kg; ára 5000 font meghajtó, 200 font lemez. (Ez még mindig Gigantikus!)

**ÚJ!**



## IBM itt-ott, mindenütt

Ahogy arról már többször is hírt adtunk, a régóta várt PC<sub>jr</sub> otthoni számítógépet az IBM bejelentette. Ez a modell a mikrogéppiac alsó végét célozza 669, illetve 1269 \$-os áron. Vezeték nélküli billentyűzet-képernyő csatlakozójával, színes grafikus lehetőségeivel, a fényceruza illeszthetőségével és az IBM nagyszámítógépekhez való kommunikációs szoftverjével ez a „junior” gép lehet olyan erőteljes, mint a piacon kapható professzionális mikrogépek kisebb képviselői.

●A mikrogéppiacon található termékszála felső végét célozza az IBM újabb bejelentése. A termék neve: IBM 3270 PC. Ezt stílszerűen

PC<sub>sr</sub>-nek (PC senior) lehetne nevezni. Ezt a mikrogépet egyidejűleg több nagyszámítógéphez lehet csatolni. Képernyőjén hét külön ablak jelenhet meg egyidejűleg. Ezek közül négy ablak „közvetíti” a nagygépek eseményeit. Két ablakot „jegyzetelésre” használhatunk. A hetedik ablak szolgál tulajdonképpen a gép saját eredményeinek megjelenítésére. Így ez a mikrogép a normál 3270-es terminálok helyettesítésére és önálló mikrogép használatára is szolgál. Felfelé követi ezt a terméket a PX XT/370. Ez a „hard” lemezzel ellátott PC továbbfejlesztése, és mindössze 3800 \$-ért az IBM 370-es nagyszámítógépek utasításkészletét tudja.

Figyelembe véve azt a fantasztikus nagy piaci részarányt, amit az IBM 370-es számítógépek és az azon működő szoftver jelent, azt mondhatjuk, hogy az IBM ezzel a termékkel befejezte a mikrogéppiac átkarolását. A becslések szerint 1984-ben az IBM megszerzi a teljes mikrogéppiac 35, a teljes számítógéppiac 50 százalékát – jósolja a Practical Computing. (J-áj-bí-em – mondja az Apple, a Sinclair és az Atari.)

●Az IBM újdonságainak egyike a Byte szerint az IBM PC Color Printer. Az új sornyomatató négy nyomtatási módban dolgozik: adatfeldolgozás 200 karakter/sec szöveg 110–150 karakter/sec „majdnem” levél minőség 30–40 karakter/sec

Négyszínű szalagjával nyolc színt produkál és az IBM grafikus nyomtatató lehetőségei is alkalmazhatók rajta.

●Ugyancsak a Byte-ban olvastuk, hogy a Herculac Computer Technology grafikus alkalmazást lehetővé tevő kártyát fejlesztett ki. Ez az IBM monokrom képernyőkártyát helyettesíti azzal, hogy az IBM PC monitorjához csatolva egy 720x348 pont felbontású grafikus képernyőt biztosít.

## Buborékmemória az IBM-hez

A fél megabites buborékmemória egy mini-winchester lemezként viselkedik az IBM mikroszámítógéphez. Az új terméket a San Diego-i Helix Laboratories tervezte, s a Byte-ban olvastunk róla. Az eszköz engedelmese-  
kedik az MS/DOS 2.0 és a CP/M-86 lemezkezelő utasításainak. A tár „elérési ideje” 40 millisecond, átviteli sebessége 400 ezer bit másodpercenként. Ára: 1495 dollár.

## Komplett csomag

A személyiszámítógép-piac újdonságai az úgynevezett „értéknövelő” üzletek. Ezek a vállalkozások speciális feladatokra, igényekre komplett csomagot állítanak össze a gyártótól beszerzett hardverből és a fejlesztett vagy vásárolt szoftverből és így ajánlják a számítástechnikai ismeretekkel nem rendelkező felhasználóknak. (A csomagolást nemcsak a kö-zértben kell megfizetni.)

## A képernyő előtt

A képernyős terminálokkal dolgozók gyakran panaszkodnak fejfájásra, szemfájásra, átmeneti látásgyengülésre. E panaszok csökkentésére a hardvergyártók fejlesztési munkájának eredményeként csillogásmentes, éles, kontrasztos képernyők készülnek. Ugyanakkor szükséges, hogy a képernyők előtt ülők leg-  
alább háromóránként szünetet tartsanak munkájukban. (Előbb azonban meg kell írniuk a munkahelyi testnevelési programot.)

## Commodore 64 kezdőknek...

Egy amerikai szoftverház Code Pro néven hozta forgalomba a kezdő Commodore-felhasználókat oktató programcsomagját – írja az Info World. A programcsomag ára 60 dollár. A mintaprogramokkal, kézikönyvvel támogatott oktatóprogram programozási feladatmegoldást, Basic nyelvet és a Commodore kitérő színes grafikus és zenei lehetőségeinek alkalmazását tanítja meg a képernyő előtt ülő felhasználónak interaktív módon. (Jobb ma egy oktatóprogram hatvanért, mint holnap egy Syntax error három napi munkáért!)

## ...és haladóknak?

A Commodore cég a Las Vegas-i téli vásáron új mikroszámítógépeket jelentett be. Az új gépek neve Commodore 264, illetve 364. A 264-es típust áprilistól lehet kapni kb. 500 dollárért. A gépek alapja egy 7501 mikroprocesszor, amely ugyanazzal az utasításkészlettel rendelkezik mint a Commodore 64-es 6510-es chipje. Az új chip csak abban különbözik elődjétől, hogy VLSI technológiával készült. A 364-es abban különbözik a 264-től, hogy külön numerikus billentyűzete és beépített beszédgenerátora van. Az új gépek BASIC interpreter mellett hasznos operatív tár mérete 60 k bit a 64-es 38 K bitjével szemben.

Meglepetést okozott, hogy az új gépek nem teljesen kompatibilisak a 64-es típusal (tár-cím-változások miatt) és grafikus-zenei lehetőségei szegényesebbek.

Újdonság, hogy a gépek ROM-ba égetett alkalmazási programokkal vannak ellátva. Ezek a szövegfeldolgozási, számviteli, illetve adatbázis kezelő programcsomagok a mikrogéppiac különböző szegmensei számára készültek.

A mikrogép BASIC-je elődjének továbbfejlesztett változata, s rendelkezik a manapság nagyon divatos „ablak” definiáló utasításokkal is. (Kíváncsiak vagyunk, mikor törik be valahol egy ilyen ablak.)



## POSTA

Kovács László amint írja, tavaly érettségizett a budapesti Apáczai Csere János Gimnáziumban, pillanatnyilag éppen katonai szolgálatát tölti.

„Kérésem – írja – a HT 1080 Z géppel lenne kapcsolatos. Hallottam róla, hogy ehhez a géphez létezik egy assembler-disassembler program, amely az assemblerben beírt programot lefordítja gépi kódra, illetve ugyanezt visszafelé is tudja. Egy ilyen programot már láttam ABC 80-on futni, és az, akié volt, azt mondta, hogy ő ezt egy ugyanilyen HT-re készült programról ültette át. Ha nem hívna be katonának, talán már sikerült volna megszereznem. Mostani helyzetemben úgy gondolom, hogy legjobban, ha a szerkesztőséghez fordulok. Kérdésem a következő lenne: tudnak-e egy ilyen assembler-disassembler fordító program létezéséről, amely ehhez a géphez (HT 1080 Z) készült? Ha van, hogyan lehet hozzájutni, hol lehet megvásárolni? Mivel jelenleg katona vagyok, kérhetném-e a Szerkesztőséget arra, hogy ezt a programot egy bővebb használati tájékoztatóval, kazettára kimentve elküldene címemre. Lakcímem: Kovács László, Kecskemét, Zalka M. u. 1. III/12. 6000”

Sajnos szerkesztőségünk nem vállalkozik programok másolására, megküldésére. Kovács László is csak abban reménykedhet, hogy valamelyik olvasónk segít rajta. Viszont Kovács Lászlónak is, másoknak is rendelkezésére áll a BIT-LET szoftverbörze rovata.

„Két témához szeretnék hozzászólni. Az első – a szerintem nagyon fontos – iskolaszámítógép-programmal kapcsolatos. Nagy eredmény az, hogy a középiskolák számítógépet kaptak, de ez „csak” az első lépés. Most nagyon sok múlik az igazgatókon, tanárokon. Mindenképpen azt kellene elérni, hogy ők ne érezzék felesleges tehernek az egész programot. Sajnos nekem negatív tapasztalataim is vannak ezen a téren.

Ehhez a témához tartozik az iskolák programokkal történő ellátása. A cél az lenne, hogy minél több, az oktatásban jól felhasználható programot kapjanak az iskolák. Sajnos a TII-nek a pályázata nem segíti ezt elő. Nem akartam hínni a szememnek, amikor átlapoztam a kiadott pályázati feltételeket. Csak csodálni tudom azokat, akik ilyen feltételek mellett küldenek be programokat. Az illetékeseknek nem ártana felülvizsgálni a programdokumentálás előírásait. Valószínű, hogy így már sokkal többen vállalkoznának, és ez mindenkinek hasznos lenne.

A második téma amihez hozzá szeretnék szólni, az a „Vallató”. A kinyilatkoztatott elvek ellenére néhány osztályzat igencsak megkérdőjelezhető, bár azt hiszem olyan teszt nem lesz, amivel mindenki egyetért. Valószínűleg a szubjektivitás (esetleg kevés gép ismerete) volt az oka néhány mosolygásra készített osztályzatnak, főleg a Sinclair gépeknél. Megemelem képzeletbeli kalapomat, de elsősorban nem a ZX-nek, hanem tervezőik előtt.”  
Világi Gábor

Vallatónkkal sokan vitatkoznak. Ennek ellenére nem áll szándékunkban megszüntetni. Úgy gondoljuk, nem szabad túlságosan komolyan venni az inkvizitorok által adott osztályzatokat. Hiszen az valóban szubjektív véleményük. Ami a TII pályázatával kapcsolatos megjegyzését illeti, az illetékesek, ahogy Ön írja, „felülvizsgálták” bizonyos kikötéseit a pályázatnak. Erről rövidesen informáljuk olvasóinkat Sorvezető rovatunkban.

Molnár Levente Debrecenből egy sor kérdést tett föl. Elsősorban a Commodore és a Spectrum közti ingadozásában szeretne döntő lökést kapni tőlünk. Melyek az egyik s melyek a másik előnyei? Melyiket érdemesebb megvenni? Sajnos a döntést a az összevetést vissza kell hogy hárítsuk Önre. Különösen könnyíti a helyzetét, hogy előző számunkban a VC 64-es, a decemberiben pedig a Spectrum Vallatását olvashatta. Annyi bizonyos, hogy a SPECTRUM jobb tanuló gépnek, a Commodore pedig jobb komolyabb feladatokhoz!

Ami a Z80 és a MOS 6510 mikroprocesszorok vonatkozó kérdéséről – hogy tudniillik melyik a gyorsabb, pontosabb? – Nos, a dolog nem ilyen egyszerű. Ugyanis jó és rossz gépek egyaránt készülnek ugyanazzal a processzorral. A gépek sebessége, pontossága ugyanis a processzoron kívül alapvetően függ az alkalmazott szoftvertől és a kialakított hardvertől is! Mindezek leronthatják vagy javíthatják az ugyanazzal a processzorral épített gép gyorsaságát, pontosságát.

Ami a SPECTRUM képgenerálása és a lustulása közti összefüggést illeti, elég bonyolult a helyzet. Röviden azt lehet válaszolni, hogy némileg igen, de a SPECTRUM képgenerálása nem kikapcsolható, így sok értelme nincs is a dologgal foglalkozni.

Egyik legérdekesebb kérdés, amit feltett ez: „A COMMODORE 64 prospektusában megütötte egy mondat a fületem. Mégpedig: 8 független sprites egyenként 21x24 ponttal. Mi az a sprite?” A Sprite-ot mi – nem biztos, hogy a legáltalában – szellemnek kereszteltük, magyarítottuk. Ez egy olyan alakzat, amely önállóan mozgatható a képernyőn és kétszeresére nagyítható. Összesen egyszerre nyolc alakzat mozgatható egyszerre, egymástól függetlenül a képernyőn. Lehet ez például nyolc autó versenyenye, felhők mozgása stb. BASIC-ből egyébként elég nehézkesen használhatók ezek a spritek.



### Úgy hírlik...

● Az NDK-beli ROBOTRON cég elkészítette az első szocialista gyártmányú intelligens író-gépet. A margaréta és az ún. proporcionális kerek íróműves készülék U 880 típusú mikroprocesszorral épül. (Jó lenne már egy olyan intelligens írógép, amely magától, megírja a BIT-LET-et!)

● A személyi számítógépek fontos tartozékainak a mikroperifériáknak egyik legmodernebbike az úgynevezett Winchester lemez, melynek hazai megvalósítása céljából a Magyar Optikai Művekben folytak eredményes kutató-fejlesztő munkák. Terveik szerint a gyártás előkészítésére már ebben az évben kerül. (1-2 év előkészítés, azután jöhet a következő előkészítés.)

● Mikroszámítógépes logikai klub nyílt Kecskeméten a Szalvai Mihály Úttörő és Ifjúsági Házban. A közel hárommillió forintért létrehozott klubban videojátékok és iskola-számítógépek állnak a látogatók, klubtagok rendelkezésére. A mikrogeprajongó klubtagok számos játékprogrammal szórakoztathatják magukat, illetve maguk is készíthetnek játékos programokat. (A klubtagdíj havonta 10 bit.)

### Újságot olvasó gép

Egy spanyol számítástechnikusokból és nyelvészekből álló csoportnak számítógép felhasználásával sikerült elkészíteni az első újságot olvasó berendezést. A nagy sebességgel olvasó gépet úgy kívánják továbbfejleszteni, hogy képes legyen az újságok cikkeiről tematikus tárgymutatókat szerkeszteni. Mint tudjuk, eddig a számítógéppel írni, tördelni, nyomtatni lehetett az újságot, most pedig olvasni is. (Talán jobb is ha a gép olvassa, mint írt.)

### Borkóstoló

A múlt évben Kecskeméten megrendezett borkóstoló versenyen közel harminc bortermelő gazdaság termékeit kóstolták végig a zsűri tagjai. A csaknem hétszázféle italt vizsgáló öt bizottság mellett számítógép is segítette a legjobbak kiválasztását. (Utána 3 napig üzemképtelen volt.)

### 16 bites

A Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete és a Zagyvarékesi Béke termelőszövetkezet Számítástechnikai Kutató-Fejlesztési Társaságot hozott létre AGROSYS néven. Az új társaság fejlesztési eredményei között szerepel egy 16 bites, helyi hálózatok kialakítására is alkalmas mikrogep-család. (Zagyvarékesi tsz-tag dilemmája – ma hagymát szedjek vagy chiptet?)

tunk tehát kötött paraméteres teknőc-grafikát tud, de hogy ezzel milyen szép és tanulságos ábrákat lehet igen egyszerűen rajzoltatni, arra példa lehet az októberi szám 31. oldalának képernyő-felvétele, mely ezzel a CSM-Logoval készült. Végül egy szó a más típusú gépek használói számára: a CSM-Logo alapelvei igen egyszerűek, de felhasznál néhányat a Spectrum grafikai lehetőségei közül. ZX 81-re igen egyszerűen átalakítható, de az ábrák durvábbak lesznek a kisebb felbontóképesség (1. pixelek) miatt. Más gépeknél a BASIC is eltérhet, ettől azonban még a program átültethető, csak kicsit több utánagondolással.

#### A CSM-LOGO szerkezete

Ha végiggondoljuk, mit kell „tudnia” nyelvprogramunknak, többé-kevésbé világosan előttünk áll kívánatos szerkezete is. Kell egy bemeneti rész, ahol parancsainkat (azonnali végrehajtás) vagy eljárásainkat (későbbi végrehajtás külön parancsra) beadhatjuk a gépbe. Az eljárásokat a gépnek tárolnia is kell, hiszen többször akarjuk használni őket. A CSM-Logo az eljárásokat a P\$ nevű string-változóban tárolja. De tárolnunk kell az eljárások neveinek „tartalomjegyzékét” is, tehát hogy melyik eljárás hol található a P\$ stringen belül. (Nagyon lassú lenne minden alkalommal végigkeresgetni a P\$ stringet, amikor egy-egy eljárásra szükségünk lesz.) A bemeneti részt bohókás nagyravágással a komoly gépekhez hasonlóan mi is SZERKESZTŐ-nek (magyarul editor...) fogjuk nevezni.

Parancs-programjainkat és eljárás-programjainkat a gépnek le kell fordítania a maga számára. Ha azt parancsoljuk: ELORE 60, a gépnek „értelmeznie” kell, hogy az előre parancs nyomán rajzolnia kell egy egyenes vonalat, és a „60” azt jelenti, hogy ez 60 egység hosszú lesz. Lesz tehát egy FORDÍTÓ programrészünk (interpreter). Ez három al-részből épül fel: a BEHATÁROLÓ megkeresi a programban a szavak-számok határait, majd ha talált egy szót, akkor továbbadja azt az AZONOSÍTÓ-nak. Az AZONOSÍTÓ megkeresi, hogy hanyadik utasításról vagy eljárásról van szó. (Ha nem találja, közli a használóval, hogy nem ismeri a ... parancsot). Végül a költői nevű VÉGREHAJTÓ rész elvégzi, amit a parancs kíván, és ezután visszalép a BEHATÁROLÓ részhez a következő végrehajtandó parancsért. Ha ilyen nincs (azaz VÉGE volt), akkor visszatérünk a SZERKESZTŐ részbe, várva az új tennivalókat.

A LOGO (a CSM-Logo pedig különösen, hiszen mást nem is tud) grafikai célú számítógépnyelv. Kell tehát bele egy olyan rész, mely könnyen és egyszerűen rajzol. Az októberi számban azzal cukkoltuk BASIC-ben dolgozó kollégáinkat, hogy próbálják csak meg az ott közölt igen egyszerű „ház” programot megírni gépük BASIC-jében! Most leleplezzük önmagunkat: a CSM-Logóban is a Sinclair BASIC utasítások fogják megrajzolni a megrajzolni valót, „csak” éppen ezeket előre beépítjük a CSM-Logo nyelvbe, tehát a használónak a részletekkel már nem kell törődnie. Lesz tehát programunknak egy RAJZOLÓ része is.

A RAJZOLÓ rész lényege a VONAL-HÚZÓ. Ehhez azonban csatlakozik két al-rész is: a NYÍLRAJZOLÓ és a NYÍLTÖRLŐ. Nem kötelező ugyanis, de jó, ha van LOGO-teknőcünknek orra. Ebből látjuk, hol van a teknőc és merre néz. Ha teknőcünket elindítjuk valamerre, akkor ezt az „orrot” (kis nyílhegyet) le kell törölnünk a képernyőről, és megérkezés után az új helyen és esetleg új irányban ki is kell újra rajzolnunk. Jó, ha van a programban valamiféle védelem arra az esetre, ha teknőcünket tévedésből a képernyő határán túlra küldjük. Védelem nélkül a program ilyenkor hibajellel leállna.

Végül általában bármiféle programnak kell egy INDÍTÓ rész, ahol megtörténik a változók értékekkel való ellátása, a tömbváltozók kijelölése (deklarációja), stb. A CSM-Logóban kétféle indításunk lesz: a HIDEINDÍTÓ-val nulláról indulunk, minden tárunk üres lesz. A MELEGINDÍTÓ arra szolgál, hogy az addig írott eljárásokat megtartva indulunk újra, azaz nem nulláról. A MELEGINDÍTÓ a HIDEINDÍTÓ-n belül egy rész. Van a SINCLAIR-BASIC (de másoknak is) egy olyan sajátja, hogy minél hátrább van egy rész a teljes BASIC-programban, annál több időbe telik a gépnek, míg megtalálja. Századmásodpercekről van csak szó, de sok kicsi sokra megy. Érdemes azokat a részeket előre venni a programban, melyeket gyakran fogunk használni, és amelyeket ritkán, azokat hátra. Ezért (mint legtöbb programban) az INDÍTÓ rész kerül a program végére (hiszen azt használják legritkábban), a RAJZOLÓ részek pedig a program elejére, hiszen ezeket igen gyakran fogjuk működtetni.

#### A CSM-Logo felépítése:

RAJZOLÓ:

NYÍLTÖRLŐ

VONALHÚZÓ

NYÍLRAJZOLÓ

FORDÍTÓ:

BEHATÁROLÓ

AZONOSÍTÓ

FENNTARTOTT SZAVAK

ELJÁRÁSOK

VÉGREHAJTÓ

SZERKESZTŐ:

INDÍTÓ:

HIDEINDÍTÓ

MELEGINDÍTÓ

(Ha a CSM-Logot már LOGO-ban írhattuk volna, ezek lennének az eljárások...)

Lusták kedvéért:

Akinek több a pénze, mint az ideje és türelme, a kész LOGO kazettáját megrendelheti az Integral GMK-tól. (Budapest 1368 Pf. 192.) Ára 250 Ft + utánvét.

**A szerkesztő azért van,**

**hogy a lap olyan legyen,**

**mint amilyenek az olvasói**

E sorok írja meglehetősen utálja a programsorok mechanikus „bepötyögését” a számítógépbe. Talán más is így van ezzel. Ezért a CSM-LOGO programot nem egyfolytában közöljük, hanem úgy, hogy részenként mehessünk rajta végig, és lehetőleg a kezdettől fogva csináljon már valamit a program. Kivétel az INDITÓ programrész, amely minden többi részhöz épül bele, és az INDITÓ egyre több részegységhez ad indulóadatokat, deklarációkat. Amint majd látjuk megjegyzéseink egy részét szabályos REM-ekbe írjuk. A kedves olvasó ezekből csak annyit írjon be gépébe, amennyit szükségesnek tart, és amennyit a rendelkezésre álló memória enged.

**A SZERKESZTŐ rész**  
A CSM-LOGO-ban megrajzoltunk valamit, utána megint rajzolni akarunk. A program bemeneti része tehát végtelen hurkot alkot: ha elvégezte a dolgot, visszatér a kezdethez, új feladatot vár, nem áll le. Tehát:

```
1000 REM SZERKESZTO
1190 GO TO SZERKESZTO: REM 1000
```

A SZERKESZTŐ részben is megadunk változókat. Az AS stringbe gyűjtjük először a beadott programsorokat, és SORSZ névvel jelöljük a képernyőn azt a SORSZÁMOT, ahová a következő sort kifratjuk.

```
1010 LET SORSZ=0
1020 LET A$=""
1100 INPUT "LOGO: ";B$
1105 IF B$="S" THEN STOP
1110 IF B$="" THEN GO TO 1100
1120 IF B$(LEN B$)<>" " THEN LET
  B$=B$+" "
1130 LET A$=A$+B$
1135 PRINT AT SORSZ,0;"
1137 PRINT AT SORSZ+1,0;"
1140 PRINT AT SORSZ,0;B$: LET SORSZ=
  SORSZ+1: IF B$(1 TO 2)<>"UE"
  THEN GO TO 1100
1150 IF A$(1 TO 2)="LE" THEN GO
  TO 1020: REM ELJARAS=1700
1160 GO SUB PARANC: REM PARANC=
  1300
1305 LET RSP=0: REM ISM.VEREM MU
  TATO
1310 LET PRUEGE=LEN P$: REM ELTE
  SZI A PROGRAM-VEGE CIMET
```

A Sinclair BASIC-ban az első sor a 0-dik számú.  
Az éppen beírt programsort a B\$ string tárolja.  
Ha mégis ki akarunk lépni a végtelen LOGO hurkából a BASIC-be, akkor S-et adunk be.  
Ha üres stringet adunk be, azaz csak megnyomtuk az ENTER gombot, semmi baj nem történik, de ha ez a sor nem lenne, úgy az ENTER gomb megnyomását félreérthetné a gép.  
Szóközt tesz utána.  
Az éppen beadott sort a többihez adja.  
(Az idézőjelek közé minimum tizenkét szóközt – space-t – kell tenni.) Letörli az előzőből ott maradt programsorok helyét. Azért nem CLS-sel törölünk, hogy ha valamit éppen rajzolunk a képernyőre, az maradjon meg.  
Kinyomtatja a programsort, egyvel emeli a sorszámot. Ha a VEGE parancsról van szó, akkor továbbbengedi, de ha nincs még vége, akkor visszaküldi a vezérlést az 1100-as sorba, azaz új programsort (utasítást) vár. Azért csak a parancs első két betűjét vizsgáljuk, hogy később rövidített (két betűs) parancsokkal is dolgozhassunk.  
Ha a beírt program (nemcsak az utolsó sor) LEgyen-nel kezdődik, ELJARAS-ról van szó. Az eljárást a Szerkesztő az 1700-as sornál dolgozza fel. Ha feldolgozás, visszatér 1100-ra, azaz folytatást vár.  
Ha nem LEgyen-nel kezdődött, csak parancsról lehet szó. Ezt az 1300-on dolgozza fel a Szerkesztő.  
A végrehajtott parancs után a Szerkesztő rész legelejeire térünk vissza.  
A PARANCOK feldolgozása  
Eltesszi a program vége címet későbbre.

Vigyázat! A PS-t eddig nem deklarártuk. Ez lesz tehát az INDITÓ rész első feladata, és pedig a HIDEINDITÁS-ban:

```
9000 REM INDITAS
9005 POKE 23558,24: REM NAGYBETU
9010 REM HIDEINDITAS
9030 LET P$="" : REM URESEN INDUL
```

Folytatjuk a Szerkesztő program parancs részét:

```
1320 LET VM=1: LET V(VM)=PRUEGE+
  1: REM A VEREMBE ELTESZI A PARAN
  CS CIMET, AZ 1. HELYRE
1330 LET P$=P$+A$
1340 GO SUB 100: REM FORDITO
1350 LET P$=P$(1 TO PRUEGE): REM
  LEVAGJA A PARANCOT, MERT AZT NEM
  TAROLJUK
1360 RETURN : REM VISSZA A SZERK
  ESZTOBE
```

A „verembe” teszi a parancs kezdő címét a P\$-on belül, és pedig a VM (veremmutató) változó által megadott első helyre.  
Hozzáteszi a parancsot (A\$) az eddigi program (P\$) végéhez. Ezért lesz a parancs címe (azaz első betűjének helye a P\$ változóban) a P\$ korábbi hosszánál egyvel nagyobb, azaz PRUEGE+1. (lásd az 1320 sort.)  
Itt végzi el, amit kell.  
Levágjuk a program végétől a már végrehajtott parancsot, mert azt nem tároljuk tovább.  
Vissza a Szerkesztőbe, ahonnan GOSUB-bal indult.

Újra vigyázat: az 1320 sorban deklaráció nélkül használtuk a V( ) tömbváltozót. És egyáltalán mi az a verem és hogyan működik? Előbb deklaráljuk a V-1 a HIDEINDITÁS-ban: 50-féle értéket tárolhatunk a veremváltozóban, 16 K-gép esetén, ha szűk lett a memória, DIM V(10) is többnyire megteszi. 1200 byte-t nyer vele.

A CSM-LOGO verem hasonlóan működik, mint a SINCLAIR BASIC GO SUB utasításé. Ezért érdemes elolvasni a gépkönyv 5. fejezetét! Röviden arról van szó, hogy a verem „legtetején”, azaz az éppen legnagyobb sorszámú változóban a V( ) tömbváltozóban belül, ott őrizi a gép a következő elvégzendő feladatának címet (esetünkben a helyét a PS programtárolón belül). A VM változó (a verem-mutató) mondja meg, hogy éppen melyik sorszámú az őrverseny cím. Ha a gép elvégzett egy feladatot, akkor mintegy kipipálja, azaz leveszi a verem tetejéről. Ilyenkor a VM értéke egyvel csökken. Ha viszont új feladatba fog, mielőtt a régi betűjezte volna, ráteszi a verem tetejére az új címet, és egyvel növeli a VM értéket. Világos? (Nem szégyen bevallani, ha még nem világos. A FORDITÓ működésének során reméljük, az is érthetővé válik.)

```
100 REM FORDITO
110 IF VM=0 THEN RETURN
200 REM BEHATAROLO
210 LET U$="NEVEGE"
220 FOR I=V(VM) TO LEN P$
240 NEXT I
```

Ha nincs feladatcím a veremben, akkor nincs mit csinálni, vissza kell térni a Szerkesztőbe.  
Itt keressük meg az utasítások végét a programban.  
A Vége-utasítás alapvetően különbözik a többitől: utána már nincs mit tenni. Ezért a V\$ lesz a „VÉGE-zászló”, itt „felcsapjuk”. Mint látni fogjuk, a 230 sor mindig megvizsgálja „le van-e csapva”.

végig (amit a LEN PS mutat). Végig megvizsgálja a PS tartalmát. Szóközt keres, mert az a tagoló elemünk. Ezért olyan fontos a CSM-LOGO nyelvterületében, hogy minden utasítás vagy számjegy (paraméter) után tegyük ki a szóközt. A FOR-NEXT hurokban a 230 sor tehát addig „keres”, amíg szóközre nem lel. Ha talált szóközt, akkor már tudja: ami az indulás és a szóköz között van, az a keresett szó. Ezt az íppog magolt szót beteszi az IS stringváltozóba.

```
230 IF P$(I)="" THEN LET I$=P$(
  V(VM) TO I-1): LET V(VM)=I+1: G
  O SUB AZONOSITO: IF U$="VEGE" TH
  EN GO TO FORDITO
```

A 230 sor megértéséhez tegyük fel, hogy a következő CSM-LOGO programot adtuk be a P\$-ba:

```
ELORE GO VEGE
12345678901234
```

Amint látjuk, ez a PS a végén lévő kötelező szóközzel együtt 14 betűnyi hosszú. Ezen a veremmutató VM=1 és az általa jelölt első verem-helyen V(1)=1, mivel a feladat a legelső betűnél kezdődik, az E-nél. A 220-as FOR-NEXT tehát I=1-nél kezd, és I=6-nál találja meg az első szóközt. Most működésbe lép a 230-as sor, és az IS értéke a P\$ 1 és 1-1 közötti, azaz 1 és 5 közötti részre lesz. Azaz IS egyenlő lesz az „ELORE” stringgel. A verem tetején az értéket most lecserejlük I=1-re, azaz 7-re, mivel a következő feladat a 7-nél kezdődő „valami” értelmezése lesz. A vezérlés most átkerül az AZONOSITÓ részbe, és ha onnan visszajövet a VS tartalma az lesz, hogy „VEGE”, akkor visszamegy a FORDITÓ elejére. Ott kiderül, hogy van-e még feladatcím a veremben. Ha nincs, azaz ha VM=0, akkor vissza lehet térni a Szerkesztőbe az új feladatért.

```
250 STOP
300 REM AZONOSITO
310 LET C$=""
320 IF LEN P$<I THEN RETURN
330 IF LEN P$=I THEN GO TO 550
340 IF (P$(I+1)<"A" AND P$(I+1)
  >" ") THEN LET C$=C$+P$(I+1): LE
  T I=I+1: GO TO 330: REM NUMBER(IP
  PARAMETER)
340 IF C$<>" " THEN LET PAR=VAL
  C$: LET V(VM)=I+2: LET I=I+1: RE
  M PARAMETER ERTEKE
350 LET I$=I$+" " : LET I
  $=I$(1 TO 7)
360 FOR A=1 TO 16: IF I$=P$(A)
  THEN GO SUB 700+A: RETURN : REM
  FENNTARTOTT SZO
365 NEXT A
395 PRINT AT 20,0;"NEM ISMEREM
  A ";I$;" PARANCOT!"; LET U$="VE
  GE": LET VM=0: RETURN
```

Ha minden rendben lesz, erre a sorra sose lesz szükség. De ha mégsem, akkor a hibás program nem fut bele valami másba, hanem itt megáll. Hogy a rák a vetésbe ne menjen...  
Ideolvassuk majd a paramétert (szám).  
Ha túl vagyunk a program végén.  
Ha a P\$ I-adik helyén az „A” és a szóköz közé eső karaktert (ezek főleg a számok) talál, azt hozzáadja a C\$-hoz, és az I számlálót egyvel tovább lépteti. Mindezt addig teszi, amíg nem jut betűhöz, vagy szóközökhöz.  
Ha a C\$ már nem üres string, akkor talál paramétert.  
IS hosszát 7-re szabjuk, hogy összehasonlítható legyen a 7 betűs hosszúságú fenn-tartott szavainkkal.  
Ha teljesül, akkor fenntartott szavát talál és elindul a VÉGREHAJTÓ részbe.  
Ha a 13. értéknél sem talál, akkor tovább keres az ELJARAS-ok között.  
Ha nem találta meg sehol, akkor ezt küzi.

**Az utasítás-VEGREHAJTÓ**

E rész feladata, hogy a fentiekben azonosított elemi CSM-LOGO utasításokat BASIC-ben végrehajtsa. Az 501 sor küldi a megfelelő helyre a vezérlést.

```

700 REM VEGREHAJTÓ
701 LET TL=PAR: GO SUB 10: GO 5
UB 20: GO SUB 45: RETURN: REM E
LORE
702 LET TL=-PAR: GO SUB 10: GO
SUB 20: GO SUB 45: RETURN: REM
HATRA
703 GO SUB 10: LET DG=DG-PAR: G
O SUB 45: RETURN: REM JOBBRA
704 GO SUB 10: LET DG=DG+PAR: G
O SUB 45: RETURN: REM BALRA
    
```

A TL változó a teknőc lépések számát tárolja.  
A DG (degree-ok) változó a teknőc irányát tárolja.

A 10, 20, 45 sorok BASIC szubrutinjait (nyíltörő, vonalhúzó, nyílráajzó) alább közöljük. Mivel elvi jelentőségük sekély helyhiány miatt különösebben nem kommentáljuk.

```

10 REM NYILTORLO
11 IF ORR=0 THEN RETURN
12 LET XP=COS (OLDRAD-2.5): LE
T YP=SIN (OLDRAD-2.5)
13 PLOT X,Y
14 DRAW INVERSE 1;XP*5,YP*5
15 PLOT X,Y
17 LET XP=COS (OLDRAD+2.5): LE
T YP=SIN (OLDRAD+2.5)
18 DRAW INVERSE 1;XP*5,YP*5
19 RETURN
200 REM VONALHUZO
21 LET RAD=DG*PI/180
22 LET XP=COS RAD: LET YP=SIN
RAD: REM XPLUSZ YPLUSZ
23 IF PP=1 THEN PLOT X,Y
27 IF PP=0 THEN PLOT INVERSE 1
;X,Y
28 IF X+XP*TL>252 OR X+XP*TL<3
THEN LET TL=TL-1: BORDER 0: BOR
DER 7: GO TO 23
29 IF Y+YP*TL>172 OR Y+YP*TL<3
THEN LET TL=TL-1: BORDER 0: BOR
DER 7: GO TO 23
30 IF PP=1 THEN DRAW XP*(TL-1)
,YP*(TL-1)
40 LET X=X+XP*TL: LET Y=Y+YP*TL
L
42 RETURN
45 IF ORR=0 THEN RETURN: REM
NYILRAJZOLO
46 LET RAD=DG*PI/180
47 PLOT X,Y
50 LET OLDRAD=RAD
52 LET XP=COS (OLDRAD-2.5): LE
T YP=SIN (OLDRAD-2.5)
54 DRAW XP*5,YP*5
56 PLOT X,Y
58 LET XP=COS (OLDRAD+2.5): LE
T YP=SIN (OLDRAD+2.5)
60 DRAW XP*5,YP*5
70 RETURN
    
```

PP a TOLLhelyzet változója.  
Itt iktatjuk be azt a védelmet, amely megakadályozza, hogy a teknőc „femenjen a tértárho!”

A 26 sorban a PP a TOLLhelyzet változója, de még nem deklaráltuk. Ezért:

```
9070 LET PP=1: REM TOLL IGEN
```

Azaz húzzon vonalat a teknőc.

Folytasuk az utasítás végrehajtó további soraival:

```

705 LET RSP=RSP+1: LET R(RSP)=I
+1: LET W(RSP)=PAR-1: RETURN: R
EM ISMETLES
706 IF W(RSP)>0 THEN LET I=R(RS
P): LET U(UM)=I: LET W(RSP)=W(RS
P)-1: RETURN: REM ISMUEGE
707 LET RSP=RSP-1: RETURN: REM
LORELES BEFEJEZVE
708 LET US="VEGE": LET U(UM)=0:
LET UM=UM-1: RETURN: REM VEGE
709 LET PP=1: RETURN: REM TOLL
710 LET PP=0: RETURN: REM TOLL
NE
712 GO SUB 10: LET X=126: LET Y
=77: PLOT X,Y: LET DG=90: GO SUB
45: RETURN: REM KÖZEP
713 CLS: GO SUB 45: RETURN: R
EM SZIVACS
714 GO SUB 9000: RETURN: REM F
ELEJT
715 LET ORR=1: GO SUB 45: RETUR
N: REM ORRAT KIDUGJA
716 GO SUB 10: LET ORR=0: RETUR
N: REM ORRAT BEHUZZA
999 STOP
    
```

ISMétlés, hasonló veremszerkezettel, mint korábban láttuk.  
Az utolsó ismétlés után a 707 sor fejezi be:  
ismétlés befejezve.

Az utasítás végrehajtót még igen sok mindenre meg fogjuk tanítani. Az ISMétlés utasításokhoz azonban még néhány változót deklarálnunk kell az INDITÓ-ban. Ezt azonban össze-  
költjük a teljes INDITÓ áttekintésével és végleges kialakításával. Az alábbi listában a teljesség kedvéért a már megadott sorok is szerepelnek, csak sorszámukat aláhúztuk. (az ismét-  
lésre vonatkozó sorok: 9080, 9090, 9100).

**A teljes INDITÓ**

```

9000 REM INDITAS
9005 POKE 23658,24: REM NAGYBETU
9010 REM HIDEGINDITAS
9030 DIM U(50): REM UEREM
9040 DIM F$(16,7): REM FENNTARTO
TT SZAVAK
9050 FOR A=1 TO 16: READ F$(A):
NEXT A
9060 DATA "ELORE", "HATRA", "JOBBR
A", "BALRA", "ISM", "ISMUEGE", "U
EGE", "TOLL", "TOLLNE", "KÖZEP",
"SZIVACS", "FELEJT", "ORR", "ORRNE"
9065 RESTORE
9070 LET PP=1: REM TOLL IGEN
9080 LET P$="": REM URESEN INDUL
    
```

Fenntartott szavak tárolója  
Beolvasse a fenntartott szavakat az FS tárolóba. Egyelőre csak 14 szavunk van.  
A szavak az előző sorban, ez pedig visszaállítja a beolvasás mutatóját az elejére.  
A TOLL változója, induláskor van TOLL.  
Hideginításkor üres a tároló.

**Beszéls címkék:**

```

90000 LET NYILTORLO=10
90010 LET VONALHUZO=20
90020 LET ORR=1
90030 LET NYILRAJZOLO=45
90040 LET FORDITO=100
90050 LET AZONOSITO=500
    
```

