

..... It's a hard life !

TV-VIDEOTON Computer

TVC-sek havonta megjelenő, belső terjesztésű lapja. 92. augusztus
Alapította: TVC-Központ Szerkesztő: Vindics István 19. szám

Ismét az "utószóval" kezdem az újságot, amely egyben kicsit elköszönés is lett. Végre tisztázódtak egyelőre az újság dolgai (remélem nem kiabálom el). A következő számokat ígérete szerint legalább az év végéig - Dunder Krisztián fogja megcsinálni, a megrendelésekkel őt keréssétek. (Aki még nem fizette ki ezt a számot, az legyen szives nekem elküldeni. Köszönöm!)

Ezúton is szeretném megköszönni a sok biztatást és segítséget azoknak, akik támogattak e két szám kiadásában. Azt hiszem kell egy-két szót szólnom arról, hogy mindezek ellenére miért adom át az újságot. Talán a legfontosabb az idő. A munkám is elég sok elfoglaltságot jelent, emellett már a programírára is alig tudok időt szakítani. Egy ilyen újság összeállítása pedig tisztán 3-4 napot kíván, ezt a "luxust" nemigen engedhetem meg magamnak.

Persze, ha más lehetőség nem lett volna, csinálom tovább. Krisztián azonban nagy lelkesedéssel állt neki a szerkesztésnek, és úgy néz ki, jól tudja majd továbbvinni a lapot. Remélem nem kell csalódnunk.

Annyit megjegyzek, nem lesz könnyű dolga, sajnos nagyon nehéz ma TVC-snek lenni. A helyzet illusztrálására néhány apróság. A 18. szám mellékletében leírt programokra nagyon kevés vevő jelentkezett eddig (az ujjaimon meg tudom számolni őket, és még a lábujj sem kell), annak ellenére, hogy gyakorlatilag másolási áron adom. Lehet persze azt mondani, hogy ez az én személyes problémám, de ajánlom figyelmetekbe a levelezési rovatban lévő felhívást. Sajnos még talán kicsit rózsaszindre festi az író a helyzetet, amikor azt mondja, hogy a programírás nem hoz hasznot. Ha összességében nézem, nagyon ráfizetéses ez üzletnek. Ha összeszámolom, mennyi lemezt, papírt, festékszalogot, áramot, levelezési költséget emészt fel az, hogy programokat szeretnék csinálni a kollégáknak, igencsak mínuszra vált az egyenleg. S emellett megtörténik a következő: kapok egy levelet, amelyben egy géptárs megírja a véleményét az újságról, programokról, egyebekről és rendszer módon még válaszborítékot is küld. Viszont nem vásárol programot (úgy tőnik, valahonnan már megszerezte), nem kér újságot (elég neki, ha kölcsönkapja a barátjátét), nem kérdez semmit, amire válaszolni kellene. Szóval nehéz így az élet ...

Ennyit a gondokról, aki még tud vidám lenni, az legyen vidám, nem akarom mindenképpen elrontani a hangulatát. Persze nem dobom ki a gépet, és azt hiszem sokáig számíthat még rám, aki programokat szeretne. Az újságban találsz némi leírást az ANTIRIAD-ról, ez a játék lesz az első Spectrum átiratom, de bízom benne, hogy lesz több is. Ha simán megy minden, szeptember végére kész lehet, de biztos ami biztos, csak az októberi számban tudom jelezni az elkészültét. Számíthatok arra is, hogy néhány cikkemmel is találkoztok majd a következő számokban.

Tehát az élet megy tovább - még ha dőcögösen is.

Tartalom:

| | |
|--|----------|
| R.I.P | 6. oldal |
| Levelezési rovat | 2. oldal |
| /most már igazi hírekkel. | |
| Gépi kód abszolút kezdőknek | 3. oldal |
| /néhány trükk a képernyőn. | |
| Egy színezési algoritmus | 4. oldal |
| /basicben és gépi kódban - demoval. | |
| Antiriad bemutatkozik | 5. oldal |
| /kis izelítő egy készülő játékról. | |
| TVCM oldal | 6. oldal |
| /benne gépi kód, trükkök, programleírás és egy szomordó hír. | |
| UPM lemezeghajtó (3) | 7. oldal |
| /a sorozat befejező darabja. | |
| Oktatóprogramok | 7. oldal |
| /Góra Rajmund írása. | |
| Griff - 3 dimenziós grafika | 8. oldal |
| /egy kis kedvcsináló egy remek programhoz. | |
| TVC 1.2.3 | 8. oldal |
| /egy táblázatkezelő - Góra Rajmund ismertetésében. | |
| Apróságok | 8. oldal |
| /néhány rajz és programocska. | |

Mi lesz a következő számban (20)?

- CLI/CPM floppy kapcsolási rajz.
- beépített botkormány betétcsere.
- a kazettás szövegszerkesztő leírása.
- nyomatató első látásra.
- hirdetés, meglepetések.
- új programok leírásai.
- Griff példaprogramok.
- kazettakalauz.

(ez volt Dunder Krisztián ajánlata, amit meg tudok toldani egy gépi kódú cikkel, gondolom ez is be fog kerülni.)

Az újság megrendelhető a következő címen:

Dunder Krisztián, 3800 Szikszó, Babits M. u. 3.

(Tudomásom szerint az újság ára változatlan marad)

Levelezési rovat

Órómmal vettem, hogy sokan felkerestek levélben hirdetésekkel, észrevételeikkel, most ezekből következik egy csokor.

Kisfürjesi Zsolt (8400 Ajka, Tószőlő u 3.)

-levelemben emliti, hogy higanygáz lámpából is lehet EPROM-törölt csinálni. (Az én megjegyzésem, hogy valószínűleg a cikkben leírt módon.)

-van néhány eladó kiegészítője TVC-hez:

>több férőhelyes, cserélhető epromos programmodulok 64-128k

>különféle játék, felhasználói epromok

>SID zenekártya

>december folyamán EPROM-égető

Sándor Csabánál (7300 Komló, Cserna út 9. Fsz. 3) vásárolható

>TVC 64k+ két operációs rendszerrel, 3-400 programmal 8000 Ft.

>720k dupla floppy 18000 Ft (esetleg cserélhető egylemezre)

>DOS oldalkártya 1200 Ft

Barca Balázs (8000 Székesfehérvár, Kőszegi F. u. 43)

-levelezne SID kártyával rendelkező TVC-ssekkel.

-keresi a HEARTLAND 64k+ -os változatát reális áron.

Szalai János értesíti mindazokat, akik nála jelentkeztek a Spectrum hardver ügyében, hogy részéről lezártnak tekinti a dolgot, ezen az úton biztosan nem lesz semmi belőle.

Király Ernő (7192 Szakály, Tanácsköztársaság u 116) közleménye
Ha valaki TVC-t vagy egyéb TVC-s dolgot szeretne vásárolni, érdeklődjön nála.

Sebők Ferenc (5742 Elek, Vécsey sor 11/A)

-jelentkezőket keres a SIMCITY program folytatására. Egy barátja elkezdte, valakinek be kellene fejeznie.

-kérte a következők közzétételét:

Kedves Erintettek!

A szerzői jogról szeretnék benneteket untatni. A 12. számban már feltették előttem a kérdést: NEM AKARTOK TOBB PROGRAMOT TVC-RE?.

Tényleg, most már ott tartunk, hogy alig jelennek meg programok. Ennek az oka közismertek: sok az engedély nélküli programmásolás, programcsere. Az eredmény köztudott: rossz minőség, és a programozók szép lassan abbahagyják a program írását, mivel nincs semmi hasznuk.

Ezt a másolgatást és programeladást kellene abbahagyni!

Arra a kérdésre, hogy kitől vedd meg a jó minőségű programokat, azt a választ tudom adni, hogy a program szerzőjétől, vagy a programozó cégtől (Comread bt, GATO bt - ez egy kis reklám volt). Higgyétek el, nálam is volt olyan időszak, mikor nem a programozói cégtől vettem a programokat, de most már áttértem a helyes vásárlási módra.

Kérem, ti is tegyétek ezt meg, ez mindannyiunk érdeke!

Szerk. megjegyzés: Mint programíró, remélem sokan megszívlelik e sorokat. Fontos tudni, hogy a Comread Bt "eltűnt", sok helyről hallottam, hogy nem válaszol a levelekre. A GATO BT-nél profi, elsősorban lemezes programokat lehet beszerezni, ennek megfelelően az árak is magasabbak.

Az ASIS (Amatőr Számítógépes Információs Szolgálat) tájékoztatója:

Kedves leendő klubtagunk!

Az ASIS az egész országra kiterjedő levelező hálózat, melyet két programozó alapított. Célunk az, hogy a legolcsóbban juthass szoftverekhez, hardverekhez, leírásokhoz és más információkhoz. Tőlünk kérhatsz tanácsot, cserepartnert bármilyen gépre, kapcsolási rajzot, hasonló érdeklődésű személyek címeit, ajánlatokat. Kereshetsz egy adott szoftvert, hardvert, leírást. Hirdetéseidet, közleményeidet ingyen lehozzuk tájékoztatónkban, melyet a klubtagok részére automatikusan megküldünk. Őj ismereteinket vírusokról, alkalmazási veszélyekről átadjuk neked. Rendezvényeinken 50 %-os kedvezményben részesülsz. Ha bármilyen kívánságod van, csak egy leveledbe kerül - adatállományaink biztosan tartalmazzák a szükséges információt. Ha mégse, közzétesszük vagy megkapod a szakértő címét. Szoftvercentrumokkal állunk kapcsolatban, melyek címeit és híreit te is megkapod. Hálózatunkon keresztül bármit vehetsz és eladhatsz. Megírjuk neked, hogy a keresett áru hol a legolcsóbb.

Címünk: 1425 Budapest 70. Postafiók 729.

(Express telefon, csak sürgős esetekben: 142-8075. Alexa Miklós, Kliment Emilia, Sárközi Sándor)

A belépés feltételei:

Írj egy levelet, pontosan feltüntetve neved, címed, géped típusát, érdeklődési körödet, kérésedet és/vagy ajánlatodat. Fizess be címünkre 500 Ft-ot, mely egy évi tagdíjad. Evkezdet az utalvány beérkezésének ideje. Célszerű, ha a visszakapott szelvényt vagy annak fénymásolatát is elküldöd leveledben. Ekkor azonnal válaszolunk, nem kell az utalványt megvárni. Az ASIS működését egy programrendszer támogatja, így leveleidre késedelem nélkül tudunk válaszolni. Ez a rendszer nyilvántartja a te adataidat, hardverek és szoftverek adatait, cégek adatait, árakat. Címünket megtalálod az Alaplap és a Mikrovilág Magazin hirdetési rovatában is.

Minta a jelentkezéshez:

Név: Irányítószám:

Város: ker ... utca

Géptípus:

Érdeklődési kör, kérés, hirdetés, ajánlat:

Góra Rajmund címe: 4400 Nyíregyháza, Szarvas u. 111

(lásd az oktatóprogramok c. cikket)

Gépi kód - abszolút kezdőknek

Az előző részben megtanultuk, hogy a processzor a memóriában található értékek alapján dolgozik, ezek a számok a megfelelő utasítás végrehajtására "ösztönzik" a Z80-at. Megtanultuk azt is, hogy a processzornak vannak belső tárolói, az úgynevezett regiszterek, ezek közül a HL regiszterpárt említettük. Szó volt három utasításról. A 35-ös utasítás a HL tartalmát eggyel növeli, a 41-es pedig kétszeresíti, a 201 pedig visszatér a hívás helyére.

Amikor bekapcsoljuk a gépet, akkor a beégetett ROM-ok veszik át a vezérlést. Ez a program két fő részből áll. Az egyiket operációs rendszernek szokás nevezni, ez kezeli a perifériákat: képernyőt, billentyűzetet, mágnot, nyomtatót, lemezegységet stb. A másik része a BASIC interpreter, ez az operációs rendszeren keresztül kapott adatokat (pl. billentyűzetről) a BASIC nyelv szabályai szerint értelmezi és feldolgozza. Azt ígértem, hogy igyekszem párhuzamokat mutatni a gépi kód és a BASIC között. A regiszterek kb. a változóknak felelnek meg, persze van elég sok különbség is a kettő között. Nagyon fontos, hogy a változókkal ellentétben csak nagyon kevés regisztert használhatunk.

Egyelőre a következő regisztereket használjuk: A, B, C, D, E, H, L. A bennük tárolt érték 0 és 255 között lehet. Ha ezekkel műveleteket végzünk, és az eredmény tdi nagy lesz, a felső részét elveszítjük. Például A=200, B=60 értékek esetén végrehajtva az A=A+B utasítást, a BASIC-ben 260 lenne az eredmény, itt viszont A értéke 260-256=4 lesz, a 256-ot elveszítettük. Persze nem teljesen, mert ilyenkor megjelenik egy jelzés (az dn. C flag-ben), ami jelzi, hogy "baj" történt.

Ha nagyobb értékekre van szükségünk, kezelhetünk együtt regiszterpárokat. Három ilyen tudunk csinálni: a BC, a DE és a HL párt. Ezek értéke a következő módon képződik: HL=H*256+L. Így egy regiszterpárban 0-65535 közötti érték lehet. Ha ennél is nagyobb számokra van szükség, a memóriában kell tárolnunk.

Fontos tudnivaló, hogy regiszterpárokkal végzett műveletek esetén nem mindig kapjuk meg az átvitel jelzését a flagekben. Miért a többszám? A C flag az átvitelt, a Z az eredmény 0 voltát jelzi, egyelőre csak ezeket használjuk.

Nézzük, mi kell ahhoz, hogy basicben egyszerű programot csináljunk? Kell az értékadó utasítás a változóhoz, néhány művelet, kell feltételes elágazási lehetőség, és valamilyen módszer, amivel kezelni tudunk legalább egy perifériát. Legyen ez a mi esetünkben a képernyő. Most összefoglalom azokat a kódokat, amelyek ezt végzik, és leírom a BASIC megfelelőiket.

Egy regiszterre vonatkozó utasítások (a kódok A,...,L sorrendben, az R betű helyére bármelyik helyettesíthető).

Érték beállítása (pl. R=12). Kódok: 62, 6, 14, 22, 30, 38, 46. A kód utáni címre kell elhelyezni a betöltendő értéket.

Regiszter növelése (R=R+1). Kódok: 60, 4, 12, 20, 28, 36, 44.

Regiszter csökkentése (R=R-1). Kódok: 61, 5, 13, 21, 29, 37, 45.

Hozzáadás A-hoz (A=A+R). Kód: 135, 128, 129, 130, 131, 132, 133.

Betöltés A-ba (A=R). Kód: 127, 120, 121, 122, 123, 124, 125.

Töltés A-ból (R=A). Kód: 127, 71, 79, 87, 95, 103, 111.

Regiszterpárookra vonatkozó utasítások (BC, DE, HL).

Növelés (RR=RR+1). Kód: 3, 19, 35.

Csökkentés (RR=RR-1). Kód: 11, 27, 43.

Hozzáadás HL-hez (HL=HL+RR). Kód: 9, 25, 41.

Ugró utasítások (kb. GOTO). Az utasítás kódja után meg kell adni az ugrás nagyságát bajtokban. Ha 0 a nagyság, akkor a következő utasításra lép. Ha pl. 5, akkor 5-tel nagyobb címen folytatja. Ha 128 vagy nagyobb a lépés, akkor ennyit

előre, és 256-ot vissza. Vagyis a 250 a következő utasításhoz képest 6-tal kisebb helyre irányítja a processzort.

Feltétel nélküli ugrás: 24.

Feltételes ugrások. Ugrás, ha a műveletnél: eredmény nulla: 40. eredmény nem nulla: 32.

van átvitel: 56. nincs átvitel: 48.

Néhány egyéb utasítás. Olvasás a HL által mutatott címről az A regiszterbe (A=PEEK(HL)). Kód: 126.

Írás a HL címre (POKE HL,A). Kód: 119.

Műveletek az A regiszter és egy adat között. Az adat az utasításbajt utáni címen van. (pl A=A OR 128). Kódok: OR-246 AND-230 XOR-238.

Már csak egyetlen dologra van szükségünk: a képernyő elérésére. Gépi kódban csak akkor dolgozhatunk a képernyőn, ha be van lapozva. Ezt az egyszerűség kedvéért BASIC-ben tesszük meg. A POKE 3,80 utasítás után a következő megszakításban belapozódik a képernyő a 32768-48127 címtartományra. Hogy ne okozzon ez gondot a BASIC-nek, a POKE 2842,127 utasítással letiltjuk számára ezt a területet. Tehát illesszük a midtkori programba az 5 POKE 2842,127;POKE 3,80 sort.

Végre elérkeztünk a példaprogramokhoz. Legyen az első a képernyő törlése. Legyen HL-ben a képernyő kezdőcíme, ekkor a BASIC program FOR ciklussal: A=0:FOR B=0 TO 239:FOR C=0 TO 63: POKE HL,A: HL=HL+1: NEXT C: NEXT B (Ugye kitaláltad? A B fut végig a sorokon, a C az oszlopokon). Ezt nem tudjuk átírni gépi kódba, ezért próbáljuk FOR nélkül: (nem kell beírni, de aki akarja, kipróbálhatja.)

10 H=128:L=0:A=0:B=239 38,128,46,0,62,0,6,239

20 C=63 14,63

30 POKE HL,A: HL=HL+1 119,35

40 C=C-1:IF C<>0 THEN 30 13,32,[251]

50 B=B-1:IF B<>0 THEN 20 5,32,[246]

A programsorok mellett a megfelelő gépi kódú utasítások állnak, ezeket kell a DATA sorába írni, vagyis:

DATA38,128,46,0,62,0,6,239,14,63,119,35,13,32,251,5,32,246,201

Az ugrások nagyságának kiszámításánál a következő módszert szoktam alkalmazni: felírom az utasítások kódjait sorrendben, az ugrás nagyságának a helyét kihagyom, pontosabban pl. zárójellel jelölöm. Ha erre a helyre akarnék ugrani, 255-öt kéne ideírnom. Mivel vissza akarok ugrani, így addig számolok - és lépegetek a bajtokon - visszafelé, ameddig a kívánt utasítás kódját el nem érem. Az ekkor aktuális számot kell a zárójel helyére írni.

Az A regiszterbe nem fontos 0-t töltenünk, tehetünk mást is. Pl. A=255 esetén 3-as, A=240 esetén 1-es, A=15 esetén 2-es színre színezzük a képernyőt. Megcsinálhatjuk azt is, hogy az USB függvényben adjuk meg a feltöltéshez szükséges értéket; ekkor az A=0 helyett A=L-t kell írunk. Ez a DATA-nál a 62,0 helyett 125-öt kíván.

Kis változtatással inverz képet kaphatunk: a POKE HL,A helyett A=PEEK(HL):A=A XOR 255:POKE HL,A kell. Ekkor változnak az ugrások is, az A=0 utasítás pedig kihagyható. Nézzük a DATA-t: 38,128,46,0,6,239,14,63,126,238,255,119,35,13,32,(248),5,32,(243),201

Érdeemes kipróbálni a 255 helyett 240-nel és 15-tel is.

Ugyancsak érdemes leírni egy-egy rutin teljes basic változatát is, amire itt nincs helyünk. Végül legyen házi feladat a következő DATA sor kibogarástása. (Híváskor az USB függvényben második paraméterként legyen egy 0-63 közti szám!) 38,128,22,0,30,64,14,63,62,255,119,35,13,32,(251),201.

Egy színezési algoritmus

Amikor először láttam a Póklakoma, később a Varázsecset előképét, egy pillanatra elcsodálkoztam. Meglepően sok szín került a négyzind képernyőre. A magyarázat persze nagyon egyszerű, kicsit gondosabb odafigyeléssel megállapítható, hogy a narancssárga színt piros és sárga képernyőpontok "pepita" elrendezésével kapták. Mivel a TVC felbontása viszonylag jónak mondható, nem veszélyes a pontok elkülönülése, nagyjából egyenletes színt kapunk. Az egyetlen probléma, hogy a TV képernyője "allergiás" néhány színkombinációra - legalábbis az enyém -, így időnként átlósan futó sötétebb csíkok jelennek meg.

Nézzünk egy kis kombinatorikát! Hány színt lehet e módszerrel a képernyőre varázsolni egy négyzind palettából? A keverésnél az első színre 4, a másodikra - mivel nem akarom egyformára - 3 választásom van. Ez összesen $4 \times 3 = 12$ eset. Azonban a piros-sárga ill. sárga-piros pepitázás ugyanaz, a 12 esetből csak 6 különböző. Ehhez hozzávesszük az eredeti 4 színt, akkor a végeredmény 10 lesz.

Hogyan csináljunk ilyen színeket?

Ha rajzprogramot használunk, akkor a szolgáltatásai meghatározzák a lehetőségeinket. A Varázsecsetnél csak a nagyítás funkcióban, pötyyönként tudjuk pepitára színezni az alakzatot, ami nagyobb terület esetén nagyon fárasztó lehet. A Pictures esetében már könnyebb a dolgunk, hiszen tud megadott mintával is területet kitölteni. Ezzel viszont az a gond, hogy később már nem tudjuk kedvünk szerint átszínezni.

Az a lehetőség, amit most felajánlok szintén nem tökéletes, de van néhány előnye. Az egyik, hogy basicben is használható, és egyszerűen kezelhető. Be lehet állítani például SETINK-kel azt, hogy milyen színnel (pepita!) színezzen. A hátránya, hogy a festendő területet csak 3-as szín határolhatja, a pepita összetevői pedig a 0,1,2 színek közül kerülhetnek ki - legalábbis, ha az átszínezhetőség fontos. Véleményem szerint azonban így is jól használható pl. játékok készítésénél.

A színezésnél még egy korlátozást vezettem be, hogy a program egyszerű és viszonylag gyors legyen. A színezendő alakzat "majdnem konvex" kell legyen. Pontosabban fogalmazva: két pontja között húzott vonal, ha vízszintes, nem szabad, hogy kilépjen az alakzathoz (lásd az ábrán).

Az algoritmus lényege a következő. Először megkeressük a terület legfelső pontját: Ha tudunk felfele lépni, akkor ezt megtesszük, és ismét megpróbáljuk. Elmegyünk balra, amíg lehet. Ezután jobbra lépegetünk, és megnézzük, lehet-e felfele menni. Ha sikerül feljebb mennünk, akkor újra kezdjük a kereső algoritmust. Ha nem megy se fel, se jobbra, akkor megtaláltuk a legfelső pontot. A mellékelt program középső része éppen ezt a keresést valósítja meg, ezért esetleg érdemes több alakzat esetén megnézni, mit csinál.

A program megértéséhez tudnunk kell még a képernyő szerkezetét. A basic esetében a képernyő a legfelső 16 kilobájton érhető el, vagyis a 49152-es címtől a 64511-es címig. Négyzind üzemmódban egy memóriacím 4 pontot fed le. Ha a tárolt értéket kettes számrendszerben írjuk, a számjegyek így határozzák meg a négy pontot: 12341234. Ez azt jelenti, hogy ha a képernyő bal felső sarkában 2 hármas színű pont között két alapszínű pontot szeretnénk, akkor a pontok bitjei: 1:11 2:00 3:00 4:11. A beírandó bájt kettes számrendszerben 10011001, a 10-es alakja $1 \times 128 + 0 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$, vagyis 153. Tehát a POKE49152,153 utasítás hatására fog

megjelenni a kívánt pontsorozat. Ha pedig 4 különböző pontot szeretnénk, akkor 1:11 2:01 3:10 4:00, az érték 10101100-172. Azt hiszem érdemes néhány esetet önállóan is kipróbálni, hogy teljesen világos legyen a dolog. (Egy megjegyzés: a bitek fordított sorrendben vannak, tehát 10 az egyes, 01 a kettes palettaszínt jelenti.)

Most pedig nézzük a basic programot. A 100-135 sorok az előkészületek. Rajzolunk benne egy alakzatot, és kiválasztjuk egy pontját a memóriacím és a maszk segítségével. (A maszk értékei balról jobbra: 136,68,34,17 - ez következik a képernyő szerkezeténél leírtakból.) Lehet kísérletezni más alakzattal is, de figyelj arra, hogy 3-as színnel rajzolt zárt alakzat legyen, a kiválasztott pont pedig a belsejében maradjon. Ha ugyanis a színezés "elszabadul", komoly kárt tehet a basic rendszerváltozóiban - mivel nem figyeli a képernyő szélét -, ezért elszáll, és csak a reset segíthet, ami legtöbbször a program törlésével jár együtt.

A 200-245 a felső pont keresése. A 200. sor a felfele lépés, ameddig lehet. Az utolsó lépést vissza kell csinálni, erre kell a 205-ös sor. A GOSUB 290 utasítások végzik a keresési dtvonal kirajzolását. Ezeket ki lehet hagyni, mert lassítják a futást. A 215 a balra lépés, a maszk kétszerezésével lép balra; ha túl nagy, akkor a szomszédos cím szélére áll. 225: lépés jobbra. Ha nem megy, készen vagyunk. 235-245: lépés felfelé. Ha sikerül, kezdjük elölről, ha nem, megint jobbra.

A 300-as sortól kezdve jön a színezés. Beállítjuk a kódot és a színt, és korrigáljuk az utolsó - már érvénytelen - jobbra lépést. A színértéket úgy kell megválasztani, mintha az egész bájtot betöltenénk ezzel a színnel. Tehát az értékek a színsorszámok sorrendjében: 0,240,15,255 (a 255-öt nem érdemes választani, ha később újra színezni akarjuk a területet). A kód a kiválasztott két színérték összege legyen.

A 305-ös sorban elraktározzuk az aktuális pont adatait. 310: színt váltunk, és lépünk balra. Ha színezhető, akkor beszínezzük, és újra próbálkozunk. Ha nem színezhető, a 330-as sorban visszahozzuk az adatokat. Most jobbra megyünk, amíg lehet. A 355-ös sorban visszacsináljuk az utolsó - már érvénytelen - jobbra lépést. Ha most is rossz helyen állunk, vége a színezésnek, ha nem, lépünk lefelé, és megyünk a színezés elejére.

Ezzel végére is értünk a programnak, következik két lista. Az egyik a basic program, a másik a gépi kód változat. A gépi kód esetében a SET INK-el beállított szín alapján színez. A kódok a betöltő program listájában találhatóak.

```

100 GRAPHICS 4
105 REM be: HL=mem.cim C=maszk
110 HL=61397:C=68:CLS:SETINK3
115 SET PALETTE 0,68,84,80
120 PLOT200,100;600,250;280,410;350,600;200,400;400,300;
    200,100
125 SET INK 1
200 HL=HL-64:A=C AND PEEK(HL):IF A<>C THEN GOSUB290:GOTO200
205 HL=HL+64
210 GOSUB290
215 C=C*2: IF C>255 THEN C=17:HL=HL-1
220 A=C AND PEEK(HL): IF A<>C THEN 210
225 C=C/2: IF C<17 THEN C=136:HL=HL+1
230 A=C AND PEEK(HL): IF A=C THEN 300
235 HL=HL-64:A=C AND PEEK(HL):IF A<>C THEN GOSUB290:GOTO200

```

```

240 HL=HL+64
245 GOTO 225
290 POKE HL, (PEEK(HL)AND(255-C))OR(C AND 15):RETURN
300 SZIN=15:KOD=255:C=C*2: IF C>255 THEN C=17:HL=HL-1
305 PUSH1=HL:PUSH2=C:PUSH3=SZIN
310 SZIN=KOD-SZIN:C=C*2: IF C>255 THEN C=17:HL=HL-1
315 A=C AND PEEK(HL): IF A=C THEN 330
320 POKE HL, (PEEK(HL)AND(255-C))OR(C AND SZIN)
325 GOTO 310
330 SZIN=PUSH3:C=PUSH2:HL=PUSH1
335 POKE HL, (PEEK(HL)AND(255-C))OR(C AND SZIN)
340 SZIN=KOD-SZIN:C=C/2:IFC<9 THEN C=136:HL=HL+1
345 A=C AND PEEK(HL): IF A=C THEN 355
350 GOTO 335
355 SZIN=KOD-SZIN:C=C*2:IF C>255 THEN C=17:HL=HL-1
360 A=C AND PEEK(HL):IF A=C THEN 400
365 SZIN=KOD-SZIN:HL=HL+64:A=C AND PEEK(HL):IF A<>C THEN 305
370 HL=HL-64:SZIN=KOD-SZIN

```

```

375 GOTO 355
400 GET:GRAPHICS2:LIST

```

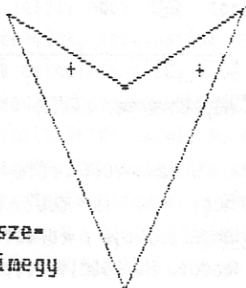
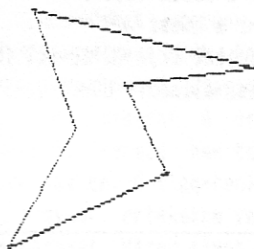
A következő program a gépi kód rutin betöltője, egy kis demoval kiegészítve. A betöltő lefutása után törlődik, a demo részt RUN-nal kell indítani. A megjelenő 6 sáv mellett hetedikként a határoló szín - most fekete - használható. Erdemes kipróbálni a más palettaszíneket is.

```

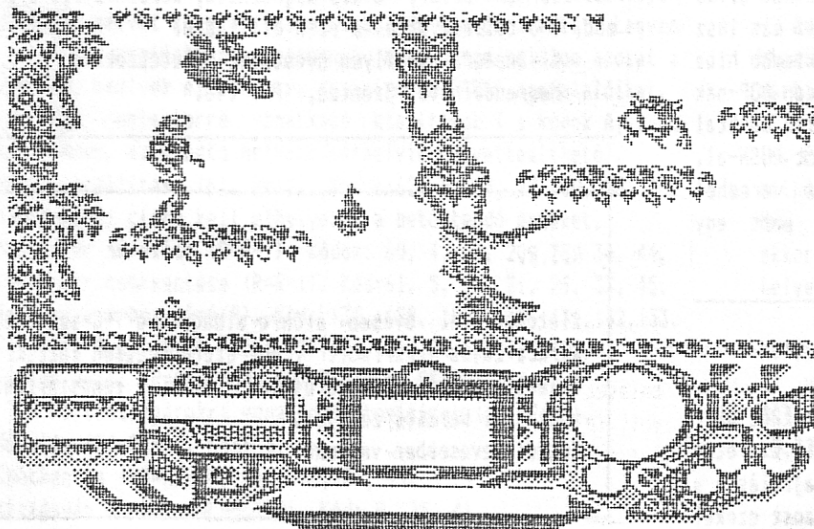
1 LOWEMPEEK(5922)+PEEK(5923)*256+270
2 CIM=PEEK(5922)+PEEK(5923)*256-270
3 FORI=0TO261:READA:POKECIM+I,A:NEXT
4 DATA62,80,50,3,0,211,2,84,93,205,20,64,62,112,50,3,0,211,
2,201,62,239,147,46,0,31,203,29,31,203,29,246,128,103,122,6,0,
31,203,24,31,203,24,133,111,120,7,7,71,4,14,17,203,9,16,252,58
,77,11,6,0,183,40,8,6,240,230,17,32,2,6,15,120,8,58,77,11,6,0,
230
5 DATA240,40,19,6,255,230,48,32,13,58,77,11,230,15,6,240,25
4,1,40,2,6,15,17,64,0,121,166,185,200,183,237,82,121,166,185,3
2,248,25,203,33,48,3,14,17,43,121,166,185,32,244,203,57,48,3,1
4,136,35,121,166,185,40,11,183,237,82,121,166,185,32,215,25,24
6 DATA233,203,33,48,3,14,17,43,229,197,8,245,8;8,168,8,203,
33,48,3,14,17,43,121,166,185,40,16,8,197,71,121,47,166,119,121
,160,182,119,120,193,8,24,225,8,241,8,193,225,8,197,71,121,47,
166,119,121,160,182,119,120,193,168,8,203,57,48,3,14,136,35,12
1
7 DATA166,185,32,229,8,168,8,203,33,48,3,14,17,43,121,166,1
85,40,17,8,168,8,25,121,166,185,32,164,183,237,82,8,168,8,24,2
24,8,201
8 DELETE1-8:REM megint RUN kell!
9 CIM=PEEK(5922)+PEEK(5923)*256-270:RH=INT((CIM+20)/256):RL
=CIM+20-RH*256:POKECIM+10,RH,RL:POKECIM+11,RH
10 GRAPHICS4
11 SET PALETTE65,84,68,0
12 SETINK3:PLOT0,0;1023,0;1023,959;0,959;0,0:FORI=1TO5:PLOT0
,160*I;1023,160*I:NEXT
13 FORI=1TO6:READS,X,Y:KOR=X*256+Y:IFKOR>=32*1024 THEN
KOR=KOR-64*1024
14 SETINK S:A=USR(CIM,KOR):GET:NEXT
15 DATA32,10,10,66,10,50,65,10,90,2,10,130,1,10,170,0,10,210

```

színezhető alakzat



nem színezhető, mert a két megjelölt pontot összekötő vízszintes vonal kimegy az idomból.



Ez a kép az ANTIRIAD című játékból származik. A program létezik C64-en, Spectrumon, és hamarosan elkészül TVC-re is. A szeptember végére várható STV 92/3 programcsomag fénypontja lesz. A C64-es leírásból olvashatsz néhány sort >>>

"... az ANTIRIAD a távoli jövőbe visz minket. Az atomháború elpusztította az emberiséget és a civilizációt - csak néhány egyszerű bennszülött élte túl az apokalipszist, akiket a dzsungel megvédett a sugárzástól. A bennszülöttek békében és nyugalomban éltek addig, amíg egy idegen, távolról érkezett faj rájuk nem erőltette saját civilizációjának "aldásait". Az őslakókat leigázták, és rabszolgaként dolgoztatják. ... A leigázott bennszülöttek nem nyugodtak bele sorsukba. ... Szervezkedésük fő célpontja az idegenek generátora, de ennek megközelítése - az erős sugárzás miatt - az egyszerű szerszámokkal felszerelt bennszülötteknek egyenlő a lehetetlennel. Az ősi mondák regélnek ugyan ... Antiriad szent fegyveréről, ... de hogy ez hol található, azt senki sem tudja. Egy nap az egyik törzs vezetői úgy döntenek, hogy fiataljaik legkiválóbbját, Talt megbízzák az Antiriad megkeresésével ..." És itt kapcsolódunk be a történetbe.

Ebben a cikkben folytatnám az újság 16. számának 4. oldalán leírtakat.

NOP: Ores utasítás, semmit sem csinál. Érdeemes megjegyezni, mert bizonyos esetekben szükség van rá!

DJNZ X: Csökkenti B regiszter értékét, és ha nem nulla, akkor ugrik X-re (relatív ugrás). Megvalósítható: DEC B és JR NZ,X Nagyon jó ciklusszervezésre, mert csak be kell rakni a B-be a ciklusok számát, és a végéről egy DJNZ-vel visszaugorni!

Mennyi lesz A értéke?

LD A,10
 LD B,2

CIKLUS:

INC A
 DJNZ CIKLUS

Megoldás: $A=10+2=12$

PUSH X: Az X regiszterpár értékét a verem tetejére teszi. Ez arra jó, hogy eltároljuk a regiszter pillanatnyi értékét, hogy később visszaállíthassuk. Csak ezek vannak: PUSH HL, PUSH DE, PUSH BC, PUSH AF

POP X: A PUSH ellentéte, a verem tetején lévő értéket az X-be teszi. Arra kell ügyelni, hogy mindig a legutoljára betett értéket lehet a leghamarabb kivenni. És hogy a PUSH-ok és POP-ok száma megegyezzen, mert különben lefagy a program! Ezek használhatóak: POP HL, POP DE, POP BC, POP AF

Mennyi lesz HL értéke?

| | |
|-------------|-------------|
| 1. LD HL,10 | 2. LD HL,20 |
| PUSH HL | LD DE,19 |
| INC HL | PUSH HL |
| POP HL | PUSH DE |
| | POP HL |
| | POP DE |

Megoldás: HL=10

Megoldás: HL=19

Apróságok

Ha RST rutint használsz, akkor ügyelj arra, hogy az RST tönkreteszi a regiszterek értékét, tehát ha pl. az RST hívás előtt B=10, akkor biztos, hogy az RST után valami más lesz benne. A PUSH és a POP nagyon jó, de a legkisebb hiba tönkreteheti a programot. Figyelj oda, hogy minden POP-nak legyen PUSH párja, illetve hogy mindig csak a legfelső adattal dolgozhatsz. Szubrutinnak soha ne adj át adatot PUSH-al, illetve szubrutin ne adjon vissza adatot a verembe! Ciklusszervezésnél ügyelj a ciklusszámlálóra, mert egy végtelen ciklusból csak reszettel lehet kilépni!

Videomemória

Tegyük fel, hogy a programozó valami képtrükköt akar csinálni, egy CLS-t, vagy csak egy emberkét szeretne kirajzolni. Basicben adott a CLS utasítás, a kirajzolásra a PRINT AT. Gépi kódban pedig az RST48 rutinok, de most ezeket ne használjuk!

Szóval a feladat adott (egy CLS), és ehhez szükség van a videomemória (röviden VID) ismeretére. Ez egy 16 kbájtos memórialap, és tulajdonképpen ezt látod. Mondjuk, hogy 4-színű

üzeműben vagy. Ekkor egy bájt 4 képpontot tárol, mégpedig úgy, hogy a 7. és a 3. bit együtt adja, hogy a bal szélső pont milyen színű (0-3). A következő képpontot a 6. és a 2. bit adja. Egy sorban 32 karakter van, egy karakter 8 képpontból áll, vagyis 32*8 pont egy sor, és ha egy bájt 4 pont, akkor egy sor 32*2=64 bájt. Ezek a bájtok a memóriában egymás mellett vannak, vagyis a legfelső bájt sor 32768-tól tart 32831-ig, a következő bájt sor 32832-től kezdődik. Az egymás alatt lévő bájtok 64 bájt távolságban vannak. 24 karaktorsor van, egy ilyen sor 10 képpontsor, vagyis 240 képsor van, az összes bájt tehát $240*64=15360$.

Basicben a POKE/PEEK 49152-65535 a VID-re dolgozik. Gépiiben a VID-et be kell lapozni OUT (2)-vel, és így a kép a 32768-49151 területre kerül. Egy CLS rutin annyit csinál, hogy belapozza a VID-et, és 32768-tól beír 15360 darab 0 bájtot, majd visszalapozza a memóriát. Íme a forráslista:

```
LD A,80 ;lapozási állapot: U0-U1-VID-SYS
LD (3),A ;PAGE változó
OUT (2),A ;PAGE regiszter
LD HL,32768 ;VID kezdőcím
LD DE,32769 ;VID kezdőcím+1
LD (HL),0 ;0-t ír a kezdő bájtba
LD BC,15359 ;15360-1 a hossz
LDIR ; a 0 bájtot írja 32768-tól 15360 hosszán
LD A,112 ;lapozási állapot: U0-U1-U2-SYS
LD (3),A
OUT (2),A
RET
```

Képkonverter /TVCM

Már volt róla szó az újság 13. számában, elég röviden. Hogy mit is tud? Betöltesz egy képet, és automatikusan megállapítja a formátumát, azonosítja a színeket és a grafikus módot. Ha betöltötted, akkor bármilyen formátumban kimenthető (akár loaderként is!). Tud nyomtatni, a színbeállítás automata. VT21200-ast és DataCoopot ismer. A nyomtatást STOP-pal lehet megállítani. A többi videomemóriát is kezeli (olvas, ír bele), a kép megnézhető, közben G-vel a grafikus mód, 0-3 billentyűkkel a szín állítható.

Kívánságra bármilyen nyomtatót beleteszek.
 Megrendelhető: Szentés, Pf. 143.

R.I.P.

Életének 14. évében örökre elhagyta a TVC-sek tábort Kurucz Lajos - Kurisoft -, egy szerencsétlen közlekedési baleset miatt. Fő területe a zene, a zeneszerkesztés volt (pl. War2100 zenéje).
 Eggyel kevesebben vagyunk...

UPM lemezmeghajtó (3)

Erre a számra már csak az UPM kompatibilis funkcióhívások felsorolása maradt. Emlékeztetőül: a hívás a \$56-os funkcióhívással történik. A C-ben lesz a kód, amely alapján elkülöníti a feladatokat. Visszatéréskor az A regiszter a hibakódot tartalmazza.

Egy fontos eltérést tapasztaltunk a kézikönyvtől: a fájl nevében nem csak nagybetűk és számok szerepelhetnek, hanem gyakorlatilag bármilyen karakter. Most pedig a lista. Az első sor a funkció megnevezése, a második a bemenő adatok felsorolása, a harmadik a visszakapott paramétereké. Többször hivatkozunk az FCB-re. Ennek felépítése gyakorlatilag megegyezik a lemez tartalomjegyzékének egy bejegyzésével. Az egyik eltérés, hogy a 0. bájtt nem a USER kód, hanem a meghajtó megadása (0=aktuális, 1=A meghajtó, 2=B meghajtó, stb.). A másik az, hogy szekvenciális illetve random fájlműveleteknél szükség van 32.-35. bájtra is.

-Diszk rendszer reset

IN: C=13 OUT: ---

-Aktuális lemez kijelölése

IN: C=14 E=kijelölt lemez (0=A, 1=B)

OUT: ---

-Fájl nyitás

IN: C=15 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Fájl zárás

IN: C=16 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Fájl keresés a könyvtár elejéről

IN: C=17 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Fájl keresés adott belépési ponttól

IN: C=18 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Fájl törlés

IN: C=19 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Sorrendi olvasás

IN: C=20 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Sorrendi írás

IN: C=21 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Fájl létrehozás

IN: C=22 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Fájl átnevezés

IN: C=23 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Drive on-line státusz lekérdezés

IN: C=24

OUT: B,A= on-line státusz

-Aktuális drive lekérdezés

IN: C=25

OUT: A=aktuális drive (0=A, stb)

-DMA címbeállítás

IN: C=26 DE=DMA cím OUT: ---

-Diszk foglaltsági térkép kezdőcímének lekérdezése

IN: C=27

OUT: B,A=diszk foglaltsági térkép kezdőcíme

-frásztöltés a lemezen

IN: C=28

OUT: ---

-Drive R/O lekérdezés

IN: C=29

OUT: B,A= R/O vektor

-Fájl státusz állítás

IN: C=30 DE=FCB kezdőcím

OUT: A=DIR-kód vagy hibajelzés

-Diszk paraméter-tábla kezdőcímének lekérdezése

IN: C=31

OUT: B,A=kezdőcím

-USER kód lekérdezés és állítás

IN: C=32 E=\$FF vagy USER kód

OUT: A=aktuális USER kód (lekérdezésnél)

-Fájl random olvasás

IN: C=33 DE=FCB kezdőcím

OUT: A=visszatérési kód

-Fájl random írás

IN: C=34 DE=FCB kezdőcím

OUT: A=visszatérési kód

-Fájl méret számítás

IN: C=35 DE=FCB kezdőcím OUT: ---

-Random mező beállítás

IN: C=36 DE=FCB kezdőcím OUT: ---

Oktatóprogramok

Mindenki tudja, hogy a számítógépet a játékon kívül másra is felhasználhatjuk. Így saját célunk érdekében "munkára kényszeríthetjük" a gépet, sőt tudásunkat is fejleszthetjük. Eszerint megkülönböztethetünk felhasználói - ezen belül több kategóriát lehet említeni -, oktatói és szemléltető (demo) programokat. Ez a cikksorozat azért indul, mert TVC-re eddig nem sok oktató jellegű program készült. Igaz, találkozhatunk egy-két próbálkozással, viszont kevés olyan program van, amely a kategóriájának megfelel. A cikk bemutat régebbi és újabb oktatóprogramot, de jobb lenne, ha csak új programokkal foglalkozhatnánk. A későbbiekben még leírnám, hogy milyen elvárásoknak kellene megfelelni egy oktatóprogramnak.

Azt hiszem, megfelelő az alkalom, hogy bejelentsek egy olyan stúdiót, amely a jövőben az elvárásoknak megfelelő, jó minőségű és természetesen felhasználható oktatóprogramokkal fog előállni. A csapat - amely 2 személyből tevődik össze - "One Unlimited TVC Studio" névre hallgat. Remélem egyre többet fogunk találkozni a nevükkel a jövőben.

Elnézést kérek, de a témától jól elkanyarodtam, most már foglalkozunk egy kicsit a programokkal. Először a térképészeti programokkal érdemesebb megismerkedni, mert van egy-pár ezekből is, bár nem mindegyik képviseli a "magasabb színvonalat". Ilyen programok valójában nem nagy sikert arattak, nézzük őket név szerint: Európa, Afrika, USA, Szovjetunió (most már FAK), Magyarország 1-2.

Ezek közül csak az Európa nevezhető kidolgozottnak. Nem is akarok róla többet írni, mert már mindenki ismeri, évekkel ezelőtt "piacra" került. Annyit illik róla tudni, hogy Commodore átírat és nem valami óriási agyszülemény. Tudása kitűnő, bár nem kell összehasonlítani a PC GLOBE-bal. A többi program színvonala silány. Igaz, a grafika jó, de nem olyan sokrétűek, mint az előző program. A városok, megyék, államokon kívül nem tudnak semmit (pl. autógyártás, folyóirat, történelem, stb.). Viszont akiket érdekelnek ezek a programok, megkaphatják tőlem, valamennyit azért lehet belőlük tanulni.

GRIFF - 3 dimenziós grafika

Ez a program nálam a reklám áldozata lett. Amikor keresni kezdtem TVC-re a programokat, ezt olvashattam a Novotrade katalógusában: "BASIC-ból hívható grafikus szubrutinydíjtemény. A 3-dimenziós grafikai rendszerben a rajzolás a cursor segítségével történik. Részletes leírással. Ara 601 Ft."

Erre azt mondtam: egy sima térbeli ábrázolás nekem nem ér meg ennyi pénzt, a szükséges néhány képletet majd megkeresem valamelyik könyvben. Most azonban megkaptam a részletes leírását, és egy kissé megdöbbentem. A hirdetett térbeliség még véletlenül sem fedi a program lényegét. Szerintem azzal lehetne legjobban jellemezni a program erőseit, hogy megvalósítja BASIC program számára a LOGO programnyelv technógrafikájának lényeges pontjait.

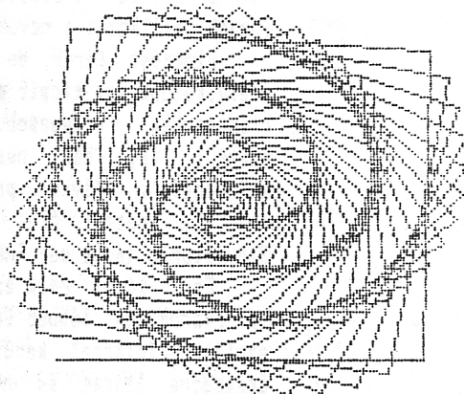
Jelen esetben a teknőcöt egy griffel helyettesítették, ezzel is jelezve, hogy térbeli mozgásokat tehet. Az utasítások két fő csoportra oszthatók. Az egyik csoport a mozgást, a másik a megjelenítést csinálja. A mozgásnál mindig tárolódik a griff térbeli helyzete és iránya. Tudjuk a saját magához igazított koordináta-rendszerben előre-hátra, fel-le, jobbra-balra mozgatni, valamint elfordítani valamelyik tengely körül. A megjelenítésnél szabályozni tudjuk a láthatóságot és az alkalmazott leképezési rendszert (perspektivikus illetve párhuzamos vetítés). Az utasítások részletes felsorolása meghaladja a terjedelmi lehetőségeinket, de gondolom izelítőnek ennyi elég. A programhoz tartozik egyébként egy demo és egy előtétprogram, amelyekből elég sok információ kileshető.

A programhoz mellékelte tájékoztató (melléklettel) több mint 40 oldal, ennek egy rövidített változatát szeretném egy szövegfájlba elkészíteni, így könnyebben eljuttatható lesz az érdeklődőkhöz. Egy másik ötlet: talán érdemes lenne "kiemelni" a programot a BASIC környezetből, és egy LOGO-hoz hasonló feldolgozórendszert készíteni hozzá. Akinek van kedve - és tudása - megpróbálhatná, azt hiszem megérné.

TVC 1.2.3 (Vékony Sándor)

A program neve nem sokat sejtet, de ha zárójelben azt

Griff = demo



```
10 SZOG=88
20 X=USR(RESET)+USR(POSX,260)+USR(POSY,215)
30 FOR I=460 TO 10 STEP -3
40 X=USR(FWARD,I)+USR(TURN,SZOG)
50 NEXT I
```

irnám mellé, hogy "IBM - Lotus", gondolom sokaknak mondana valamit. Na jó! Elárulom, egy kiváló táblázatkezelő programról van szó, amely majdnem mindent tud. Tudásával és természetesen gyorsaságával persze nem lehet összehasonlítani a Lotus-sal.

Ismerkedjünk meg a használatával. Akinek birtokában van ez a program (nem feltételezem, hogy sokaknak meglesse) és a tájékoztató is, annak semmi újat nem fogok leírni.

A táblázat 26*26 mezőből áll. A mezőkben 3féle dolgot helyezhetünk el: szám (max. 10 jegyű), ciáke (10 karakter hosszú szöveg) és formula (műveletek, stb.). A mezőkre névvel lehet hivatkozni. A táblázatban egy 10%-os ablakon keresztül dolgozhatunk. A pozíciót mindig megtudhatjuk a 2. sorból, az ablakon belül joy-jal mozgathatjuk a kurzort. Az adatbevitel a következőképpen történik:

-ha számot akarunk rakni a mezőre, akkor egyszerűen begépeljük

-ha címkét akarunk elhelyezni, azt jelezni kell a gépnek

-formulával úgy dolgozhatunk, hogy leütjük a "E" billentyűt.

A formulában csak mezőnevek helyezkedhetnek el.

(P1: AB+CD+KF*TU/FR - helyes AB+100+DD - helytelen.)

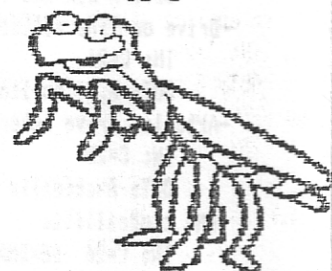
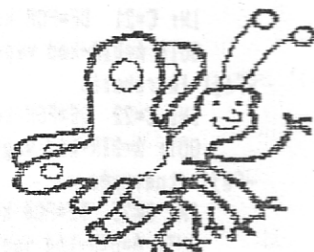
A műveletekben nem lehet zárójelet használni!

Egy sorban vagy oszlopban álló mezők értékét összegezni is lehet. Amely mezőket összegeztünk, nem lehet formulát írni. Ha meg akarjuk tudni az egyik mező tartalmát, oda kell menni a kurzorral. A képernyő második sorában ki van írva az aktuális mező tartalma. A táblázatban belül bármely mezőre is ugorhatunk a "<" jel, és egy paraméter segítségével. A "!" átszámol díra mindent, az "ESC" hatására elköszön a program. "k"-val ki lehet menteni, "b"-vel be lehet tölteni a táblázatot. "t" töröl, "s" a táblázatban szereplő számokat törli. "." után meg kell adni, hány tizedesjegyre számoljon a gép, és végül az "o" összegzi az adott mezőre a felette, vagy töle balra lévő mezőket.

Ennyit a programról dióhéjban.

-60ra-

```
1 REM - Csikok
10 FOR I=1 TO 99
20 FOR J=99 TO I STEP -1
30 OUT 100,I: OUT 111,I: OUT 100,J
40 NEXT J,I
```



Néhány rovar a HEWSROOM sorozatból.