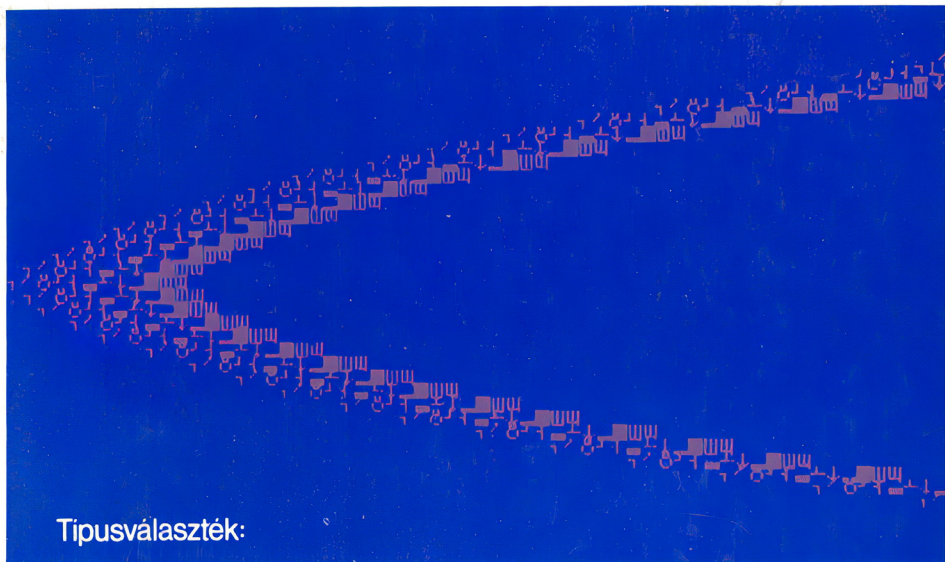


SZEMÉLYI SZÁMÍTÓGÉPEK

kezelése, programozása és alkalmazása

VARGA JÓZSEF



Típusválaszték:

COMMODORE-64

ABC-80

HT-1080Z, 2080Z

TEXAS-99

PRIMO-32, 64

TV COMPUTER-32, 64

VT-16

Terra · Budapest

TÁVOKTATÁSI PROGRAM ÉS KÉZIKÖNYV

**SZEMÉLYI
SZÁMÍTÓGÉPEK KEZELÉSE,
PROGRAMOZÁSA
ÉS ALKALMAZÁSA**

Írta és szerkesztette:

DR. VARGA JÓZSEF

TÍPUSVÁLASZTÉK:

ABC-80

HT-1080 Z, -2080Z

TEXAS-99

COMMODORE-64

PRIMO-32, -64

TV COMPUTER-32, -64

VT-16** PROPER-16** IBM PC

TERRA • BUDAPEST 1986

A kéziratot lektorálta:

DR. BALÁZS KATALIN
a matematikai tudományok kandidátusa

Grafikus:

SZABÓNÉ TÓTH ÁGNES

ISBN 963 205 158 0

© Terra, Budapest 1986 – Dr. Varga József

Printed in Hungary

LEVÉL A FELHASZNÁLÓHOZ	9
1. AZ ALAPGÉP KEZELÉSE ÉS PROGRAMOZÁSA	15
1 A televízió bekapcsolása	15
2 A számítógép bekapcsolása	15
3 A belső memória kiürítése	15
4 A gép, mint kalkulátor	15
4' Egy mondat megjelenítése a képernyőn	17
5 A program tömörítése	21
6 Program listázása a képernyőre	21
7 Egy szám megjelenítése a képernyőn	21
8 Normál-alakú szám használata	23
9 Számsor beolvasása ciklussal	23
10 A program tömörítése	25
11 Valós szám konvertálása egész számmá	25
12 Egyváltozós tömb beolvasása ciklussal	27
13 A tömb dimenzionálása	27
14 Számpárok beolvasása	29
15 Kétváltozós tömb beolvasása cikluspárral	31
16 A cikluspár tömörítése	31
17 Szövegírás a képernyőre	33
18 Karakter tömb beolvasása	33
19 Karakter tömbök átszabása	33
20 Értékdás karakter változóhoz	35
21 Karakter konstansok összefűzése	35

22	Értékkadás és aritmetika, valós változókkal	37
23	Utasítások összevonása	37
24	Értékkadás numerikus kifejezéssel	37
25	Értékkadás konverzióval	39
26	Algebrai kifejezés kiértékelése	39
27	Vezérelt adatbeolvasás	41
28	Program módosítás átírással	41
29	Adat INPUT és hatványozás	43
30	Utasítások összevonása	45
31	A nem értelmezett művelet	45
32	Befogóival adott derékszögű háromszög megoldása	45
33	A végtelen ciklus megszakítása	47
35	Feltételes munkarend	47
36	Elemi utasítások tesztelése	51
37	Az első k természetes szám szorzata	53
38	A változók kezdőértéke	53
39	Függvények értéktáblázata	55
40	Értéktáblázat megadása ciklussal	55
41	Szakadós függvény értéktáblázata	57
42	A képernyő törlése	57
43	Hibajelek összevonása	59
44	A másodfokú egyenlet megoldása	59
45	Tájékoztató szövegek a programban	61
46	A kör általános megoldása	63
47	Az egyszerű átlagok keresése	65
48	Beolvasott adatok átlagai	67
49	Idősorok bázis és lánc viszonyszámai	69
51	Egyváltozós tömbök alpműveletei	69
52	INPUT tömbök aritmetikája	73
53	Számok prímtenyezős felbontása	75
54	Bank ügyeletek általános megoldása	75
55	Szubrutinok hívása kódokkal	79
56	Program-bővítés	81

2. A KAZETTÁS MAGNETOFON KEZELÉSE, PROGRAMOZÁSA ÉS FELHASZNÁLÁSA	85
57 Program rögzítése kazettára	85
58 Rögzített program ellenőrzése	87
59 Program betöltése kazettáról	89
60 Programok elhelyezése és azonosítása	91
61 Címmel azonosított program megkeresése	93
62 Adatfile (névsor) rögzítése és betöltése	95
63 Adattömb (forgalmi adatok) rögzítése	97
64 Rekord-okba szervezett tömb rögzítése	99
65 A rekord -ok átszervezése és rögzítése	101
66 Vegyes rekord -ok (készlet) rögzítése	103
67 String alakú rekord -ok rögzítése	103
68 Ismeretlen szerkezetű file betöltése	105
69 Tartalomjegyzék lekérdezése kazettáról	107
3. A LEMEZE GYSÉG KEZELÉSE, PROGRAMOZÁSA ÉS FELHASZNÁLÁSA ..	109
70 A lemezegység (Floppy) bekapcsolása	109
71 A mágneslemez (disk) formázása	109
72 Program átvitele kazettáról disk -re	113
73 Program betöltése és futtatása	115
74 Program felülírása disk -en	117
75 Tartalomjegyzék lekérdezése disk -ről	119
76 Véletlen számok (lottó) keltése és rögzítése	123
77 Adatfile betöltése és monoton rendezése	125
78 Soros adatfile végleges betöltése	127
79 Soros adatfile bővítése	129
80 Vegyes rekord -ok rögzítése disk -en	131
81 Vegyes rekord -ok végleges betöltése	133
82 Rekord -ok elválasztása soros file -ban	133
83 Készlet-nyilvántartás használata	135
84 Soros adatfile diszkrét feldolgozásai	141

85	File betöltés hibacsapdával	145
86	Magyar–angol szótár alapítása és használata	149
87	A dinamikus STOP programozása	155
88	Relatív file telepítése	157
89	Rekord -ok lekérdezése relatív file -ből	161
90	Személyzeti nyilvántartás alapítása és használata	163
91	Karbantartó program a személyzeti file -hoz	167
92	Relatív file hézagos feltöltése	171
93	Relatív file közvetlen rögzítése	173
94	Soros file átvitele relatív file -ra	175
95	Adatfile átvitele kazettáról disk -re	179

4. A PRINTER KEZELÉSE, PROGRAMOZÁSA ÉS FELHASZNÁLÁSA 185

96	A printer bekapcsolása	185
97	Program kinyomtatása	185
98	Disk tartalomjegyzék kinyomtatása	187
99	Nyomatványok címzése	189
100	A nyomtatott programok címezése	189
101	Levélcím kinyomtatása	191
102	Levelek folyamatos címezése	191
103	Számsorozat négyszögbe nyomtatása	193
104	A gép karaktereinek a kinyomtatása	197
105	Betű és írásminták programozása	201
106	Nyomatvány programozása	205
107	Disk -ről betöltött file nyomtatványban	209
108	A másodfokú függvény ábrázolása	213
109	Paraméteres függvény (görbesereg) ábrázolása	217
110	Egy változós függvények ábrázolása	219
111	Karakter tervező program	221
112	Gépirás reguláris és tervezett karakterekkel	225
113	Oszlopdiaagramm rajzolása	229

5. A GÉPRENDSZER BELSŐ ADOTTSÁGAI	233
114 A gépi idő mérése	233
115 A középeurópai idő mérése	233
116 A szabad memória mérése	235
117 Memória kísérletek	237
118 Memória rekeszek lekérdezése	239
119 Memória rekeszek átírása	245
120 Karakterek színezése	247
121 Egyváltozós függvények karakterizálása a képernyőn	249
122 Függvények mikro-képe	253
123 Programok összefűzése	267
124 Programok montírozása	275
125 A „LEGÓ” DISK és alkalmazása	281

6. MELLÉKLETEK

1. „LEGÓ” rutinok C-64-re	281
2. Utasításkészletek	295
3. Hibajelek gépenkénti reprezentációban	309
4. Grafikai útmutató	317

7. IRODALOMJEGYZÉK	323
--------------------------	-----

8. ANGOL-MAGYAR SZÓTÁR	324
------------------------------	-----

ALKALMAZOTT KONFIGURÁCIÓK

COMMODORE-64:	1530 DATASETTE UNIT. MODEL C2N. VC-1541 FLOPPY DISK-ek (8 és 9). MPS-801 DOT MATRIX PRINTER. VT SUPER COLOR TELEVIZIÓ. COMMODORE-64 MICRO COMPUTER
ABC-80:	ABC-80 IKER KAZETTÁS EGYSÉG. VT SUPER COLOR TELEVIZIÓ. ABC-80 SZÁMITÓGÉP
HT-... 80 Z:	A SZÁMITÓGÉPBE BEÉPÍTETT KAZETTÁS EGYSÉG. VT SUPER COLOR TELEVIZIÓ. HT-1080Z, VAGY HT-2080Z SZÁMITÓGÉP.
TEXAS-99:	CASSETTE RECORDER SANYO. MODEL 2502 U. H/C PERIPHERIAL EXPANSION SYSTEM. VT SUPER COLOR TELEVIZIÓ. TI-99/4A TEXAS INSTRUMENTS.
PRIMO:	CASSETTE RECORDER SANYO, MODEL 2502 U. VIC-1541 FLOPPY DISK, VAGY FDU 1109 FLOPPY. DATACoop-DCD-PRT42 PRINTER. VT SUPER COLOR TELEVIZIÓ. PRIMO-A-32, VAGY PRIMO-A-64 M. SZÁMITÓGÉP.
TV COMPUTER:	CASSETTE RECORDER SANYO. MODEL 2502 U. TV COMPUTER IKER FLOPPY (A és B). WM2000 PRINTER. VT SUPER COLOR TELEVIZIÓ. VIDEOTON TV COMPUTER-32, VAGY 64.
VT-16:	COMPUPRO WFLOPPY (C). VT DISPLAY ÉS IKER FLOPPY (A és B). WM4000 PRINTER. VT-16 PROFESSZIONÁLIS SZEMÉLYI SZÁMITÓGÉP.

TISZTELT KOLLÉGA!

Ha Ön megtisztelt azzal, hogy könyvem a kezébe vette, mindent meg szeretnék tenni, hogy ötletét ne bánja meg.

Könyvem rendhagyó. Nyitott állapotában a **jobboldali lapon találja a tananyagot**, amit COMMODORE-64-re (C-64) alkalmazunk. A **baloldali lapon regiszterek állnak**, amelyekben a címlapon felsorolt gépekre módosítjuk a C-64-re adott megoldásokat.

A tananyag mintegy 125 alkalmazási feladatra épül. A feladatok didaktikai rendszert képeznek, az egyszerűtől a bonyolult felé vezetnek. Minden feladat megoldása igényel valami újat. Így a kezelési, programozási és alkalmazási ismeretek téra állandóan bővül, az ismert elemek pedig rutinírózódnak.

A tananyag feltételezi, hogy Ön személyi számítógép elé ül és „utasításai” szerint dolgozik. Ha a problémákat **egymásután** megoldja, készséget szerez a személyi számítógép manuális kezelése, programozása és alkalmazása területein. Minden probléma egy lehetséges megoldását közöljük. Próbálja ki, szerezzen tapasztalatokat. **Csak akkor menjen az anyagban tovább, ha megértette** a megoldás minden részletét és ismeri funkcióját.

A programok **BASIC**, illetve **EXTENDED BASIC** nyelven íródtak. Egyes gépek **PASCAL**, **FORTRAN** stb., nyelveken megírt programokat is kezelni tudnak, de ezzel nem foglalkozunk. A **BASIC** megoldások háttérter magyarázatok világítják meg. Ezek C-64-re érvényesek, de ritka kivételtől eltekintve konvertálhatók a regiszterekben szereplő gépekre is, ha az ott közölt módosításokat értelmezik. Alapvető eltérések esetén irodalmi forrást jelölünk meg.

A könyvben található programokat a címlapon felsorolt gépek mindegyikén teszteltük, működésüket garantáljuk. A felhasznált adatokat nem közöljük, mert az Ön által választott, bemenő adatoknak is meg kell felelnie, ha eleget tesznek a feladat követelményeinek.

HOGYAN DOLGOZZON?

Ha a számítógép-ismeretben kezdő, akkor kövesse a szerzőt az első oldaltól az utolsóig. **Csak akkor lapozzon előre, ha erre felkérjük és térjen vissza, ha azt ajánlják.**

Induljon ki az [1] problémából és oldja meg. A közölt megoldásban jelek utalnak a tennivalókra. A \gg jel mindig gépkezelési (pl. kapcsolás), a $>$ jel pedig gépirási feladatra kéri fel. Sok bosszúságtól menekül meg, ha a $>$ jelet követő karaktereket (betűk, számok, szóközök stb.) hiba nélkül kopogja le, mert mindegyiknek, az adott gépre jellemző funkciója van. Minden gépen található kezelő gombok (pl. **SHIFT**). Ezek lenyomására nevük bekeretezésével (pl. **SHIFT**) utalunk.

A személyi számítógép párbeszédre képes. Értelmezi közléseinket és válasz-lépésekkel reagál. Eredményeit tőlünk függetlenül (pl. hibajelek), illetve utasításainknak megfelelően a képernyőn közli. **Figyelje a képernyőt.** Esetenként megadjuk, hogy mit kell a képernyőnek mutatnia. A kettőzött vonalpár || a képernyőre utal.

A számítógépek munkáját parancsok és utasítás sorozatok (programok) irányítják. A **parancsok** végrehajtása lekopogásukat követően kezdeményezhető, szemben az **utasításokkal**, amik **programba szervezve** aktiválhatók. A parancsok jelzésére a *-ot használjuk.

A program meghatározott munkarendet hordoz. A munkarendet az utasítások programban elfoglalt helye és tartalma határozza meg. A munkarend grafikusán ábrázolható. Ilyen grafikon (munkagráf) található egyes programok baloldalán, amelyben a ▼-ek a mellettük fekvő utasítás végrehajtását, a nyilak pedig a soronkövetkező utasításra való átugrást jelzik. **Tapogassa végig a munkagráfokat** és értelmezze a kiváltó utasításokat.

A gépekhez kiegészítő berendezések, képernyő (display), magnetofon (tape recorder), lemezegység (floppy disk), nyomtató (printer) stb. tartoznak. Ezek munkáját is parancsok, illetve program részek vezérlik. A gép munkájába való bekapcsolódásukat a munkagráfon jelezzük. A ○—○ jel a magnetofonra, a □ jel a lemezegységre és a Λ jel a nyomtatóra utal.

Ha egy probléma megoldását érti és sikeresen keresztül tudja vinni a gépen, áttérhet a következő problémára. **A között megoldásokban vastagbetűs részeket talál.** Így emeljük ki az új ismereteket első előfordulásuk alkalmából.

Néhány órai munka után már gondolhat arra, hogy **megoldásainkat saját megoldásokkal** helyettesíti és a feladatok bővítésével vagy átfogalmazásával új program megírására és alkalmazására vállalkozik. Például egy területszámítási program az adatok és képletek megfelelő változtatásával, vagy térfogat számítási programmá fejlesztésével stb., nyújthatja Önnek a saját munka élményét.

Ha nem C-64 előtt ül, hanem a szóbanforgó gépek közül egy másikkal dolgozhat, akkor is érvényesek az elmondottak azzal a kiegészítéssel, hogy a tüköroldal, gépével egyező című regiszterét is tanulmányoznia kell. Az Ön gépére vonatkozó megoldások ugyanis kisebb-nagyobb mértékben eltérhetnek a C-64-re adott megoldásoktól. A regiszterben csak az **eltérést** találja. Ha ezt tételesen **behelyettesíti** a C-64 megoldásba az Ön gépére érvényes megoldáshoz jut. A tételeket * választja szét a regiszterben és számjelek teszik azonosíthatóvá a behelyettesítési pontokat. Ha egy utasítás csak részben tér el a C-64-re érvényesőtől, akkor előfordul, hogy csak a helyettesítendő rész kerül a regiszterbe. Erre a rész előtt vagy után elhelyezett : figyelmeztet (a részt megelőző, illetve követő elemek C-64 szerint érvényesek). Az eltéréseket „csak” **első előfordulási helyükön** jelezzük a regiszterben, később már ismertnek tételezzük fel. Itt is alkalmazzuk a vastag betűs kiemélést, mert az előbbi „csak” a nehezebben megjegyezhető eltérésekre nem vonatkozik. Ha a regiszterben a probléma száma után **OK*** jelzés (okay) áll azt jelentheti, hogy a szóbanforgó gép a C-64 megoldást egy az egyben **elfogadja**, de jelentheti azt is, hogy **új eltérés nem** merült fel.

A gépek használata programok és adatok bevitelét feltételezi. Ez időrabló gépirási munkát jelent. Mivel egyes **programok** és **adatok** többször is felhasználásra kerülnek könyvünkben, sőt a gyakorlati alkalmazásukra is gondolhat (pl. munkahelyén), **gondoskodjon a megőrzésük-ről.** Ezt a lehetőséget a kiegészítő berendezések (magnetofon, lemezegység stb.) biztosítják. Alkalmazásuk időben való megismerését előre-lapozással és visszatéréssel oldjuk meg.

Anyagunk olyan C-64-re szól, amelyhez kazettás magnetofon, floppy disk, printer és képernyő tartozik. A címlapon felsorolt gépek, ipari vagy kereskedelmi adottságok miatt ettől eltérő konfigurációban is előfordulhatnak. Egységek (floppy disk, printer, kazettás magnetofon) hiányozhatnak, egységek (kazettás magnetofon, floppy disk) duplikáltak stb. A hiányzó egységeket más, rendszeresített elemekkel **helyettesítjük**, hogy Ön ne maradjon ki a programozás és alkalmazás szempontjából fontos problémák megoldásából. A helyettesítésről a regiszterek a felmerülési pontban tájékoztatnak. Ha egy feladat megoldása nem nélkülözheti a helyettesített egységet, akkor a feladat számát a regiszterben a NEM* szó zárja.

A kiegészítő berendezések kis **lámpákkal** jelzik a készenlétet, az effektív munkát és zava-

raikat. Kiegészítő jelzéseként követhetjük mozgásaikat, zajaikat és hangjelzéseiket. **Figyelje a jelzéseket**, mert tennivalóra utalhatnak.

Gyakorlott számítástechnikus Ön? Ebben az esetben három dolgot tekintsen át: milyen gépek, milyen alappéldák és milyen gépkezelési-programozási utasítások szerepelnek a könyvben. Az elsőt a címlapon, a másodikat a könyv elején lévő tartalomjegyzékben, a harmadikat pedig a könyv végén elhelyezett utasításjegyzékben találja. A két utóbbi oldalszámokkal irányítja a keresett dolog megtalálásában. Bízom benne, hogy ez a mű kézikönyvként fogja szolgálni Önt alkotó munkájában.

Ne marasztalja el a szerzőt, amiért a programokból kihagyta az irracionális elágazásokat (pl. negatív távolság, idő stb.), mert a programok áttekinthetősége volt a cél. Nem mutatjuk be a gépek összes képességeit, mert egyrészt úgy ítéljük meg, hogy a felvett ismeretek birtokában könnyen megtanulhatja minden érdeklődő például a zenei anyagok megjelenítését vagy játékképek programozását a gépkönyvből, másrészt nem titkoljuk, hogy olvasóink figyelmét a számítógépek érdemi felhasználására szeretnénk inkább irányítani, nem lebecsülve hanem megalapozva a gépek játék funkcióját is.

HOGYAN KEZELJE A GÉPÉT?

A gyártó gondosan ügyel arra, hogy gépe és kiegészítő berendezései ne kapcsolódhassanak egymáshoz és az áramforráshoz a leégés veszélyével. Ezt a dugaszoló kapuk és dugók páronként eltérő formáival érik fel. Idegen kábelt ne tűrjön meg a gép körül. A géprendszer összeállítását megkönnyítik a gépkönyvek rajzos útmutatói.

Az ember és a számítógép párbeszédében az iniciális közvetítő szerepét a klaviatúra (írógép) tölti be. Használati tudnivalók:

- Az egyjeles (pl. **A**, **B** stb.) gombok leütésekor a képernyő visszatükrözi a jelet.
- A két fedőjeles (pl. **≤** stb.) gombok leütésekor az alsó jelek (>), a **SHIFT** gombbal együtt leütve pedig a felső jelek (<) érvényesülnek.
- Egyes gépeken (pl. C-64, TEXAS-99 stb.) homoljeles (pl. **□** stb.) gombokat is találunk, ezek az **FCTN**, illetve **SHIFT** kezelő gombbal együtt leütve aktiválódnak.
- A gépek automatikusan váltanak sort, ha a képernyő-sor végére érünk. **A sor hossza gépenként változik.** Ha a sorváltást a sor vége előtt kezdeményezzük, le kell ütnünk a sorváltót, ami **RETURN**, **NEW LINE**, **ENTER** stb. neveket viselhet.
- A képernyőn egy villogó alakzat, (pl. négyzet, szakasz, háromszög stb.) a kurzor jelöli ki, hogy a következő leütésünk hol jelenne meg. A kurzor pozícióját megváltoztatja a sorváltó, a lécbillentyű, a nyílás (pl. **←** stb.) gombok és gépenként változó gomb kombinációk leütése.
- Ha egy gombot vagy a lécet lenyomva tartjuk, akkor a kérdéses jel, illetve szóköz sorozódik a képernyőn és felülírja, vagy eltolja az útjában álló jeleket.
- A sorváltó a gép legfontosabb billentyűje. Parancs lekopogása után leütve elindítja a parancs végrehajtását, utasítás lekopogása után leütve elindítja az utasítás memorizálását, feltéve, hogy bizonyos hibák az utasításban nem fordulnak elő. Az utasítások sorszámmal kezdődnek (pl. 10 READ A). Ez biztosítja a program rendezését és nyilvántartását.
- A gépek csak azt a nyelvet (szókészlet és nyelvtan) értik, amire képesítették (beégetett, betöltött BASIC). **Ha törjük a nyelvet**, utasításaink hibásak lehetnek és megértés hiányában **végrehajtásuk elmarad.** A gép a képernyőn közli a hibát, körvonalazva jellegét (pl. SYNTAX ERROR: helyesírási hiba), vagy azonosítási lehetőségét (kód).
- A hibák egyik forrása a melléütés, amit a klaviatúrán követhetünk el. Ha rögtön észrevesszük a melléütést, akkor a **←** gomb leütéseivel visszaléptetjük a kurzort és a hibás jelet felülírjuk.

- Ha egy parancs vagy utasítás a végrehajtás során bizonyul hibásnak, ismételten lekopogjuk hibátlanul (az utasítások sorszámát is) és a sorváltó leütésével a végrehajtás, illetve a programban való behelyettesítés veszi kezdetét.
- A javítás és módosítás további lehetőségeit az anyagban fogjuk megismerni.
- A géprendszert **védje** az ütődésektől, a sugárzó hőtől, a por és hamutól, a folyadékoktól, a mágnesezett idegen tárgyaktól és a figyelmetlen kapcsolásoktól.
- A tároló szalagok és lemezek nem állják a -10 fok alatti hideget, a hajlítást és gyűrést, a mágnesezett tárgyakat és csupasz részeik fogdosását.
- Ha a géprendszer bármelyik eleme meghibásodik, szakemberre van szüksége, kivétel a magnetofon tisztítása.

A TÍPUSVÁLASZTEKRŐL

A magyar gazdaság és társadalom intenzív fejlődésének egyik kulcsa a programozható automaták általános alkalmazása. Ehhez importból származó és hazai gyártású gépek szolgáltatják a bázist. Az alkalmazás mindannyiunk megszerezhető készségére vár.

A TEXAS-99, a C-64 és az IBM PC (Personal Computer) a hasonló nevű tőkés cégek terméke.

Az ABC-80 (Budapesti Rádiótechnikai Gyár), a HT-1080Z és -2080Z (Híradástechnikai Szövetkezet), a PRIMO-32 és -64 (Számítástechnikai és Automatizálási Kutatóintézet-Cosy), a TV COMPUTER (VIDEOTON), a VT-16 (VIDEOTON) és a PROPER 16 (Számítástechnikai Koordinációs Intézet) hazai gyártmány.

A felsorolt gépekből tekintélyes mennyiség van családok, intézetek és vállalatok bitokában. Tanulásra egyformán alkalmasak, de a gépkezelés és programozás **ugyanolyan jól** megtanulható az ABC-80, a HT-1080Z, a HT-2080Z, a PRIMO-32 gépeken, mint a kiépítettségük magasabb foka miatt a gazdasági munkába is beállítható TV-COMPUTER, PRIMO-64, TEXAS-99 és C-64 gépeken. A VT-16, a PROPER-16 és az IBM PC alapozó tanulásra való használata már luxus, mert professzionális gépekről van szó, amik kisebb vállalatokat és intézményeket képesek kiszolgálni.

Miért kerültek mégis egy gyékényre ezek a gépek? Azért, hogy az Ön horizontja nyitott legyen.

AZ OPERÁCIÓS RENDSZERRŐL

Parancsaink és programjaink értelmezését és végrehajtását a géprendszer operációs rendszere bonyolítja. Az operációs rendszer értelmező, szerkesztő, vezérlő és végrehajtó programból áll és a gép típusára jellemző. Az operációs rendszer és a gép viszonya többféle lehet:

- Az operációs rendszer a gép **belső tartozéka** (beégetett BASIC), tehát bekapcsolás után a géprendszer azonnal munkakész. Ilyen az ABC-80, a HT-1080Z, a C-64, a PRIMO-32 és a TV COMPUTER.

- Az operációs rendszer egy **beégetett** és egy **betölthető** részből áll. Az első rész az előbbieket szerint működik. A második rész bázisa speciális rendszerlemez (TEXAS-99 : EXTENDED BASIC), magnetofon kazetta (PRIMO-64 : CDOS), vagy floppy lemez (PROPER 16/a : PROPOS-16 V 2.00). Betöltés után aktivizálható, a gép kikapcsolásával törlődik. A betöltésre a regiszterek figyelmeztetnek.

– Beégetett operációs rendszerrel nem számolhatunk. **Floppy lemezről olvassuk** be, a gép bekapcsolását követően az operációs rendszert (VT-16 : UDOS V 2.00, VT BASIC), mely azonnal munkakész. A gép kikapcsolásával törlődik.

A betölthető operációs rendszerek lehetnek a gépek **tartozékai**, vagy külön **vásárolhatók**. Ez utóbbi kategóriába tartoznak a géprendszer elemeit ellenőrző (C-64 : TEST DEMO) és adathordozókat formalizáló (TEXAS-99 : DISK MANAGER COMMAND MODULE) eljárások segítő programjai, stb.

A PROGRAMRÓL

A gondolkodó ember valamiből következtet valamire, vagyis input-transzformáció-output sémával él. Ha a következtetési folyamatot számítógépre szervezzük a séma nem változik, de nekünk kell gondoskodni a három tevékenység olyan elemekre bontásáról, amikor a számítógép képes megoldani és következtetésünk rendjét tartani. Az elemi tevékenységek **utasításokkal**, a folyamat utasítások sorozatával, a **programmal** váltható ki.

Az input (adatbevitel) és az output (eredményközlés) **módja** a géprendszer kiépítésétől függ. A transzformáció (átalakítás) a gyakorlat igényeit és összefüggéseit kielégítő **technológia** végigvitele. A technológia matematikai és logikai formulák sorozatában testesül meg.

Programunk hibás lesz, ha rosszul választjuk meg a **technológiát**, ha nemlétező **perifériáról** biztosítjuk az adatokat, vagy ilyen helyre irányítjuk az eredményeket, ha valamely elemi tevékenység **kiváltására** más tevékenység utasítását vesszük igénybe, ha utasításaink **sorrendje** eltér a kívánatostól, ha **hibásan írunk** BASIC-ül stb.

Az első hibát próbaszámítások eredményei jelzik. Olyan adatokkal vizsgáltatjuk programunkat, hogy kézzel is kiszámolhassuk az eredményt. Az említett egyéb hibákról maga a gép tájékoztat a képernyőn vagy a perifériák a jelzőlámpáik által. A képernyőn a hiba megnevezése, illetve kódszáma jelenik meg. Értelmezéséhez a gépkönyvek végén elhelyezett **hibajegyzék** nyújt segítséget.

A hibakeresés egyik eszköze a munkaközi **PRINT**, ami képernyőre viszi a program paraméterek változásait (a kész programból töröljük). Sokan kedvelik a **TRON-TROFF** utasításpárt, melyek hatására az összefüggő utasítások címkei jelennek meg a képernyőn. Léteznek **hiba-csapidák** amiket a programba állítunk be.

Tisztelt Kolléga! Ha valahol megakad és nem segíti ki sem a gépkönyv, sem az ajánlott irodalom, **írjon** egy lapot (IX., Dimitrov tér 8.) és válaszolni fogok.

Akár fiatal, akár idős embert üdvözölhetek Önben, számítok kitarására, hogy végigcsinálja.

Üdvözlettel



Utóirat:

Köszönettel tartozom mindazoknak, akik a mű létrehozásában közreműködtek és erényeit gyarapították. A könyv gyengeségei a szerzőt terhelik.

ABC: 1 OK*
 2 A **POWER** feliratú gomb benyomása a kazettás egység hátoldalán*
 3 OK*
 4 A ?-et a **PRINT** szó helyettesíti* A hatványozás jele** pl. 3**2* A gép munkára kész, ha ABC-80 jelenik meg a képernyőn*

HTZ: 1 OK*
 2 I/O → I **NEW LINE***
 3 OK*
 4 OK* A ?-et a **PRINT** szó is helyettesítheti*

TEXAS: 1 OK*
 2 **ON** ; **ENTER** ; 1 leütése ebben a sorrendben*
 3 OK*
 4 >PRINT (10+30) **ENTER*** Az aritmetikai művelet zárójelbe kerül*
A ?-et a **PRINT** szó helyettesíti* A hatványozás jele ^*

PRIMO: 1 OK*
 2 **ON** gomb lenyomása a tápegységen* || PRIMO BASIC SYSTEM '84,1 ||*
 3 OK*
 4 **UPPER** leütésekor nagy, ismételt leütésekor kisbetűkre vált a gép*

TV COM: 1 ~~OK~~*
 2 **ON** gomb lenyomása a tápegységen* || TV COMPUTER V || **A***
 3 OK*
 4 **SHIFT** **LOCK** leütésekor nagy, **LOCK** leütésekor kisbetűkre vált a gép* A ?-et a **PRINT** szó helyettesíti* **IRJON KISBETUVEL!*** A hatványozás jele ^*

VT-16: 1 Törölve* 2 tegye be a baloldali (A) **FLOPPY**-ba az "UDOS 2.00" című operációs lemezt és **F1*** || DATUM? HO/N/EV :>12/04/86 || **ENTER***
|| IDOPONT? O : P : MP :>12 : 15 : 30 || **ENTER*** Tegye az (A) **FLOPPY**-ba a "VT BASIC 1.00" c lemezt és hívja a BASIC-et || A : \> VT BASIC ||
 ENTER* F1-F10 gombok értéke a kép alján* 3 OK* 4 A kétjeles gombok felső jele ↑ -el aktiválható* A hatványozás jele ^*

1. AZ ALAPGÉP KEZELÉSE ÉS PROGRAMOZÁSA



1 Kapcsolja be a televízió 36. csatornáját:

2 Kapcsolja be a számítógépet:

»ON/OFF → ON

```

***COMMODORE 64 BASIC V2***
64 K RAM SYSTEM 38911 BASIC BYTES FREE
READY
    
```

3 Űrtse ki a számítógép memóriáit:

* >NEW

RETURN

△△ – A számítógép használatát a memória üríté-
sével kezdjük. Lekopogjuk a NEW (új)
szót és lenyomjuk a RETURN gombot.
A READY (kész) szó a művelet végét
jelzi.

001 A gép, mint kalkulátor.

4 Kopogja le az alábbi sorokat és nyomja le a RETURN gombot:

* >? 10+30 40

RETURN

* >? 15-13 2

RETURN

* >? 5 * 3 15

RETURN

* >? 20 / 5 4

RETURN

* >? 3 ↑ 2 9

RETURN

ABC: A gyakorlat OK* A memóriát és a képernyőt a **RESET** gomb (a gép hátoldalán balképről) benyomásával törölhetjük* A gép nem érzékeny a szóközre *

HTZ: ? 2 ; 3 2 3 *

TEXAS: A gyakorlat utolsó 3 feladata az alábbi formában OK* > PRINT "2", "3"
* >PRINT "2", "3" * >PRINT "2" & "3" *
A gép érzékeny a szóközre*

PRIMO: A gyakorlat OK* A képernyőt a **RESET** gomb (a gép hátoldalán jobbkézre) benyomásával törölhetjük* A gép a szóközre nem érzékeny *

TV COM: A gyakorlat az utolsót leszámítva OK* || *** NOT UNDERSTOOD || (Nem értem)* A képernyőt a **RESET** billentyű (jobboldalon alul) felnyomásával törölhetjük* A gomb (hátraljára) színesre vált, ha a TV színesre van állítva*

VT-16: A gyakorlat OK* A gép érzékeny a szóközre* Próbálja ki az alábbiakat:

>OFF * >ON *

A VT-16 a betöltött operációs rendszertől függően 8, illetve 16 bites rendszerben dolgozik (UDOS : 16 bit -es r), vagyis 8, illetve 16 bit alkot egy magasabb tároló egységet*

△△

– A számtani műveletek a ? leütésével indíthatók és a RETURN gomb lenyomását követően megjelenik a végeredmény. A számtanban szokásos műveleti jeleket a gép csak részben fogadja el. Számára a * a szorzás, a / az osztás és a ↑ a hatványozás jele. A számok írásánál figyeljünk a 0 és az 1-re, nehogy az O, illetve L betűkkel helyettesítsük (ezért írjuk a nullát áthúzva: \emptyset), valamint a tizedesvesszőre, amit ponttal kell jelölni.

Gyakoroljon:

PRINT

*	>? 2+3-4+7-1	RET	7	
*	>? (35+25) / (2+4)	RET	\emptyset	
*	>? (35+25) / 2+4	RET	34	
*	>? 35+25 / 2+4	RET	51.5	
*	>? (5+4) * (7-2)	RET	45	
*	>? (5+4) * 7-2	RET	61	
*	>? 5+4 * 7-2	RET	31	
*	>? 5+ (4 * 7-2)	RET	31	
*	>? (2+3) ↑ 2/5+4	RET	9	
*	>? 2+3 ↑ 2 / (5+4)	RET	3	
*	>? 2 ; 3	RET	23	
*	>? 2 , 3	RET	2	3
*	>? 2 3	RET	23	

002

A gép, mint robot.

4'

Utassítsa a számítógépet, hogy írja ki a képernyőre az Ön nevét:

ABC: >10 READ A >30 PRINT A A nap-jel () szöveg azonosítására teszi alkalmassá az A numerikus változót* Belső nyelvtana szerint egy adat \equiv egy szó pl. KISILONA. Ezt a >20 DATA "KISILONA" védi ki*

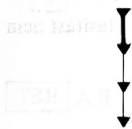
HTZ: Az utasítások címkéit az >AUTO parancs rendre generálja*
Kilépés a leütésével történik* A gép szóközre nem érzékeny pl. >10 READ A\$ \equiv >10 READ A\$*

TEXAS: Az utasítások címkéit a >NUM10 parancs rendre generálja.
Kilépés az -el* A szóköz fontos pl. >10 READ A\$*

PRIMO: Az utasítások címkéit az >AUTO parancs rendre generálja*
Kilépés a -val* Próbálja ki az >AUTO 10, 20 parancsot*

TV COM: OK* Elütés esetén a **BOTKORMÁNY** balra dütésével léphet vissza és felül írja a hibás jelet. A botkormány balra, jobbra, le és fel képes a kurzort mozgatni*
OK* Szóközre nem érzékeny a gép*

VT-16: Az utasítások címkéit az >AUTO parancs rendre generálja*
Kilépés -vel* Próbálja ki az >AUTO 10, 20 parancsot*
A >RUN helyettesíthető az -el*



```
>1Ø READ A$  
>2Ø DATA KIS ILONA  
>3Ø PRINT A$  
>4Ø END
```

RETURN

RETURN

RETURN

RETURN

△△ – A gép megjegyezte az Ön utasítás sorozatát és végrehajtási parancsra vár:

*

```
>RUN
```

RET

```
|| KIS ILONA ||  
|| READY ||
```

△△ – A négy címkézett utasítás programot alkot. Egyező címke a programban nem szerepelhet, mert a később lekopogott felülírja a korábbi. A program BASIC nyelven íródik. Neve a "BEGINNERS' ALL PURPOSE SYMBOLIC INSTRUCTION CODE"¹ kezdő-betűiből származik. Jelkészlete az angol nyelvhez kötődik. Programunkban kulcsszavak: READ (olvas), DATA (adat), PRINT (nyomtat), END (vége), változó: A\$ és adat: KIS ILONA szerepelnek. Az A változó valós számokat, a \$ jellel kiegészítve szövegeket (string) azonosíthat. A program végrehajtását a RUN (fut) parancs indítja el. Az A\$ változóhoz hozzárendeli a DATA készletét (A\$ = KIS ILONA), majd képernyőre írja A\$ értékét. A programok END-el zárulnak.

¹ Kezdek minden igényét kielégítő, szimbólikus utasítás gyűjtemény.

ABC: [5] >NEW parancsot a RESET gomb helyettesíti* Beékelt data utasítást nem fogad el a gép. Megoldás: >1Ø READ A [6] : PRINT A [6]*
>2Ø DATA KIS ILONA : END*[6] » TARTÓSAN NYOMJA LE A [RET]
GOMBOT*[7] >3Ø PRINT A *

HTZ: [5] OK*
[6] A képernyőt a [CLEAR] törli* CLEAR (tisztá)*
[7] OK*

TEXAS: [5] A TI BASIC-ban a program nem tömöríthető*
[6] A képernyőt a >CALL CLEAR [ENT] paranccsal töröljük* CALL (hiv)*
CLEAR (tisztá)*
[7] >3Ø PRINT A *

PRIMO: [5] OK* Az utasításhossz ≤ 5 sor*
[6] [CLS]* >LIST és tartós [RET]*
[7] OK* Ha rosszul lát próbálja ki a >NEW : PRINT CHR\$(2) [RET]
parancsot *

TV COM: [5] OK* Az utasításhossz ≤ 8 sor* Próbálja ki a >GRAPHICS 2 [RET] pa-
rancsot. Ezesetben 4 képernyő hossz a határ* Ha rosszul lát próbálja ki a
>GRAPHICS16 [RET] parancsot* A >GRAPHICS4 [RET] alapállásba
visz* [6] » CLS [RET]* [7] >3Ø PRINT A *

VT-16: [5] OK* Az utasításhossz <3 sor*
[6] »CLS [ENT]* » [F1] [ENT]*
[7] OK*

5

Írja fel a 4 programot egyetlen utasítás sorozatba, majd aktiválja:

*	>NEW	
▼	>1Ø READ A\$: DATA KIS ILONA : PRINT A\$: END	RET
*	>RUN	RET

△△ – Több utasítást egy címke alá vonhatunk kettősponttal elválasztva, a sorrend megtartása mellett. Az utasítás-sor hossza legfeljebb 2 képernyő-sor lehet.

003

A CLR/HOME SHIFT gombok és a LIST utasítás használata.

6

Törölje a képernyőről az 5 programot, majd hívja elő ismét:

	»USSE LE EGYIDEJŰLEG A CLR/HOME és SHIFT GOMBOKAT	
*	>LIST	RET

7

Írjon fel egy számot:

*	>NEW	RET
▼	>1Ø READ A	RET
▼	>2Ø DATA 125	RET
▼	>3Ø ? A	RET
▼	>4Ø END	RET
*	>RUN	RET

ABC: >1Ø READ A : PRINT A ** >2Ø DATA 1.25E2 : END *
 OK **

HTZ: >AUTO 1Ø ** OK **
 A hátlapon álló gombot benyomva a képernyő 32 karaktert fogad egy sorba, a gép pedig 64 karaktert ír. A lenyomása a másik 32 karaktert láttatja **

TEXAS: Egy címke – egy utasítás, így kopogja le >NUM 1Ø bekezdéssel *
 OK **

PRIMO: OK **
 OK ** Ha jobban kedveli a fehér alapon fekete képet a >NEW után adja a >PRINT CHR\$(3) utasítást * Szemkimélő a >PRINT CHR\$(4), CHR\$(2) után dolgozni *

TV COM: OK **
 OK **

VT-16: OK **
 OK **

△△ – A PRINT kulcsszó helyett kérdőjelet is írhatunk.

004 Normál alakú számok használata.

8 Tömörítse a 7 programot 2 címke alá és a számot normál alakban adja meg:

```
* >NEW  
▼ >1Ø READ A : DATA 1.25E2  
▼ >2Ø ? A : END  
* >RUN
```

△△ 125 = $1,25 \cdot 10^2$, amit 1.25E2 alakban fogad el a gép. Minden tizedes szám átírható ilyen alakra, pl. $-0,0056 = -5,6 \cdot 10^{-3}$, innen $-5.6E-3$ adódik.

005 A FOR–NEXT ciklus utasítás alkalmazása.

9 Írjon fel 7 valós számot egymásutáni beolvasással:

```
* >NEW  
▼ >1Ø FOR I=1 TO 7  
▼ >2Ø READ A  
▼ >3Ø DATA 4, 5.2, 3.85, 2E3, -4.15, -2E4, -3E-1  
▼ >4Ø PRINT A;  
▼ >5Ø NEXT I  
▼ >6Ø END  
* >RUN
```

ABC: 10 OK* Egy utasítás ≤ 3 képernyő sort ($3 \cdot 40$ karakter) tölthet ki
 11 OK* Ha % helyett $\text{\textcircled{X}}$ szerepelne, string számokkal dolgoznánk.
Ezek memorizálása karakterenként történik*

HTZ: 10 OK* Nyomja ki a "VIDEO CUT" gombot. Δ képernyő most 64 karaktert fogad. A PAGE gombnak esik a funkciója*
 11 OK* Ha % helyett # szerepelne, "dupla pontosságú" (≥ 7 jegy) számokkal dolgoznánk*

TEXAS: 10 Van "EXTENDED BASIC" kazettája? Kapcsolja a géphez^{2*} 1 ; 2
|| READY ||* Ekkor >10 FOR I=1 TO 7 :: READ B* >30 PRINT B ::
NEXT I :: END*
 11 B%-ot a gép nem tudja kezelni*

² A továbbiakban EXTENDED BASIC-et feltételezünk!

PRIMO: 10 OK*
 11 OK* Próbálja meg a % helyett a # jelet szerepeltetni*

TV COM: 10 OK*
 11 Csak valós típusú számokkal dolgozhatunk*

VT-16: 10 OK*
 11 OK* Próbálja meg a % jel helyett a >5 DEFINT B utasítást alkalmazni*
Próbálja meg a %-ot # jellel helyettesíteni, illetve >5 DEFDBL B utasítással élni*

△△

– A FOR–NEXT utasításpár a közrefogott utasítások ismételt végrehajtását biztosítja. Az egyszeri végrehajtást ciklusnak nevezzük. A ciklusok számát a FOR–TO utasítás deklarálja. A ciklusváltozó (I) bármely betű lehet. A változó kezdő és végértékét (1;7) megadjuk. Léptetését a NEXT oldja meg. A lépéshossz +1, ha mást nem adunk meg. (Lásd később). A gép munkarendjét a gráf szemlélteti. Programunk A numerikus változója eszerint 7-szer vált értéket: $A=4$; ... ; $A=-3E-1$.

10 Tömörítse a **9** programot 3 címke alá:

```
* >NEW  
↓ ↓ ↓  
↓ ↓ ↓  
↓ ↓ ↓  
* >10 FOR I=1 TO 7: READ B  
>20 DATA 4,5.2,3.85,2E3,-4.15,-2E4,-3E-1  
>30 ? B, : NEXT I : END  
>RUN
```

→ Ha van kazettás egysége, előre lapozhat az **57** programhoz!!!

11 Módosítsa a **10** programot, hogy B egész típusú változó legyen:

```
>10 FOR I = 1 TO 7 : READ B%  
>30 ? B% : NEXT I : END  
>RUN ○ ↓ ○
```

△△

– A numerikus változót követő %-jel egész (integer) értékek felvételét biztosítja. Ha B%-ot valós (real) számmal azonosítanánk, csak egészértékét őrizné meg (pl. $A=3.25$; $A\%=3$).

ABC: 12 OK **

13 OK **

HTZ: 12 OK **

13 OK **

TEXAS: 12 OK **

13 OK ** DIM A(11) 12 elemű tömböt deklarál a $\emptyset, 1, \dots, 11$ indexszel **

NEM vezet hibajelre, ha a \emptyset indexet nem használjuk fel (>FOR I=1 TO 11)**

PRIMO: 12 OK **

13 OK **

TV COM: 12 >5 DIM A(7)* A tömb méretét mindig meg kell adni (DIM)* A DIM

A(7) 8 elemű tömböt deklarál $\emptyset, 1, \dots, 7$ indexszel* NEM vezet hibajelre, ha

a \emptyset indexet nem használjuk fel* 13 OK **

VT-16: 12 OK **

13 OK **

A %-jelet a programban a változó ismételt előfordulása esetén is ki kell írni. Az egész típusú számok tárolására a gép meghatározott méretű (2B) memória elemet biztosít. Eszerint a tárolható legnagyobb egész típusú szám 32767. Egy valóstípusú szám férőhelye 2,5-szer nagyobb (5B).

007

Az egyváltozós tömb használata. A DIM utasítás.

12

Olvasson be 7 számot egy numerikus tömbbe³ :

```

*
>NEW
>10 FOR I=1 TO 7
>20 READ A (I)
>30 DATA 350,800,4.5E2,53E2, -60, -850, 1E4
>40 PRINT I, A (I)
>50 NEXT I
>60 END
*
>RUN 0 ↓ 0

```

△△ – Az A(I) numerikus változót tömbváltozó-
nak, benne az I-t indexnek nevezzük. I a
természetes számok véges sorozata. Az
azonosítás rendezett (pl. A(1)=350, A(2)=800,
..., A(7)=1E4) és ciklussal oldható meg.
A ciklusok száma legfeljebb 10 lehet?

13

Módosítsa a 12 programot pl. 11 szám beolvasására. Kérjen listát a módosított programról. Válasszon 11 helyett más számot és kísérletezzen:

³ Az egyváltozós tömb vektornak felel meg műveleteiben. dr. Halmi Erzsébet: Lineáris algebra. Tk. Budapest 1979.

ABC: 14 >5Ø NEXT I : END* A NEXT indexe NEM hagyható el*

HTZ: 14 Kapcsolja be a gép "EXTENDED BASIC"-jét. >NEW* > SYSTEM*
? / 12288 NL* A gép villogó kurzor -ral válaszol* A továbbiakban
EXTENDED BASIC-kel dolgozunk* A gép bekapcsolásakor mindig gondoljon
rá* OK*

TEXAS: 14 >5Ø NEXT I : : END* A NEXT indexe NEM hagyható el*

PRIMO: 14 OK*

TV COM: 14 OK*

VT-16: 14 OK*

```

>5 DIM A(11)
>10 FOR I=1 TO 11
>31 DATA 80, 60, 40, 250
* >LIST
* >RUN  O↓O

```

RET

- △△ – Ha 10-nél több eleme van a tömbnek, ezt DIM utasításban deklarálni kell. A kész programba beszúrhatunk új utasításokat. A DIM utasítás helyét úgy választjuk ki, hogy a program futása során csak egyszer kerüljön végrehajtásra és megelőzze a tömb felhasználását.
- Egy programban több DATA utasítást is megadhatunk. Felhasználásuk rendjét címkék rendje adja meg. A LIST (lista) végrehajtási parancs, melyre a gép rendezett programot ír ki a képernyőre.

14

Írjon fel 4 számpárt egy oszlopba:

```

* >NEW
>10 FOR I=1 TO 4
>20 READ A1, A2
>30 DATA 10, 9, 8, 7, 6, 5, 4, 3, 2
>40 PRINT A1; A2
>50 NEXT : END
* >RUN  O↓O

```

- △△ – Az A1, . . . , A9-et indexelt változóknak nevezzük. AI' és A(I) abban térnek el, hogy I'=1, . . . , 9, míg I=1, . . . , N értékeket vehet fel. A kiírás formáját jelek biztosítják: A ; szorosan melléírást (A1; A2), a jelhiány (A2 után) új sort biztosít ciklusonként.

ABC: 15 >65 PRINT* Elválasztó sort generál az üres PRINT az I ciklusok között.
Enélkül 3 oszlopba rendezné a képernyőn az adatokat (13 karakter térközzel)*
 16 >60 NEXT J, I nem használható*

HTZ: 15 >65 PRINT* Elválasztó sort generál az üres PRINT az I ciklusok között*
 16 Általában OK, most nem, mert a PRINT után álló , 4 oszlopba rendezné
az adatokat a képernyőn (16. karakter térköz)*

TEXAS: 15 OK*
 16 NEXT J :: NEXT I* A NEXT NEM vonható össze*

PRIMO: 15 >65 PRINT* Elválasztó sort generál az I ciklusok között. Enélkül 3 osz-
lopba rendezné a gép az adatokat a képernyőn (16 karakter térköz)*
 16 Általában OK, most >65 PRINT miatt NEM*

TV COM: 15 >5 DIM A(4,2)* >50 PRINT A(I, J),,*
 16 OK*

VT-16: 15 >65 PRINT* Elválasztó sort generál az I ciklusok között* Csatolja a prog-
ramhoz az >5DEFDBL A utasítás és figyelje meg a változást (a nyolc vagy annál
többjegyű számokat nem kerekíti a gép?)*
 16 Általában OK, most >65 miatt NEM*

15

Olvasson be számokat egy 4x2-es tömbbe⁴.

```

*      >NEW
      >1Ø FOR I=1 TO 4
      >2Ø FOR J=1 TO 2
      >3Ø READ A (I, J)
      >4Ø DATA -3, 2.5, -5E6, 0.2, .05, 2ØØØØ.
      >45 DATA -256318.256, -1E-9
      >5Ø PRINT A (I, J),
      >6Ø NEXT J
      >7Ø NEXT I : END
*      >RUN   ○ ↓ ○

```

△△

– $M \times N$ -es tömbnek az M sorból és N oszlop-
ból álló számnégyszöget nevezzük. Két-
indexes tömbről van szó. Általános ele-
mét $A(I, J)$ jelöli, ahol $I=1, \dots, M$ és
 $J=1, \dots, N$ értékeket vehet fel. Sorfolytonos
beolvasás esetén $I=1$ -hez $J=1, \dots, N$; $I=2$ -höz
 $J=1, \dots, N$ stb. tartozik. Ezt fejezi ki a
munkagráfon a beágyazott ciklus. Eszerint
 $A(1,1) = -3$, $A(1,2) = 2.5$, $A(2,1) = -5E6$
stb. hozzárendelés történik. Ha felcserél-
jük a FOR–NEXT utasítások indexeit,
akkor oszlopfolytonos beolvasásra kerül
sor (FOR $J=$, FOR $I=$, NEXT I , NEXT J).

16

Módosítsa 15 -öt az alábbi módon:

```

*      >6Ø NEXT J, I
      >7Ø END
*      >RUN   ○ ↓ ○

```

⁴A kétváltozós tömb mátrixnak felel meg műveleteiben. dr. Halmi Erzsébet: Lineáris algebra. TK. Bp. 1979.

-
- ABC: 17 OK *
- 18 Az üres STRING legalább 1 karakter " " hosszú* OK *
- 19 Az adatokat a gép összevonja, ezért >30 DATA "AZ IDEZOJELBE TETT SZOVEGET", "KARAKTER KONSTANSNAK", "NEVEZZUK"*
-

- HTZ: 17 OK *
- 18 Az üres STRING legalább 1 karakter " " hosszú* OK *
- 19 OK *
-

- TEXAS: 17 Idézőjel a P gomb homlokán található és FCTN P -vel aktiválható*
- 18 Az üres STRING legalább 1 karakter " " hosszú*
- 19 OK *
-

- PRIMO: 17 OK *
- 18 Az üres STRING legalább 1 karakter helyet " " tartalmaz* OK *
- 19 OK *
-

- TV COM: 17 OK *
- 18 >5 DIM A\$(7)* Az üres STRING " " legalább 1 karakter hosszú*
- 19 >5 DIM A\$(3)*27* A karakter konstans >18 karakter esetén dimenziójuk (*H)*
-

- VT-16: 17 OK *
- 18 Az üres STRING " " hossza legalább 1 karakter* OK *
- 19 OK *
-

17

Irassa ki az alábbi mondatot: az idézőjelbe tett szöveget karakter konstansnak nevezzük:

```

*      >NEW
      >1Ø PRINT "AZ IDEZOJELBE TETT SZOVEGET KARAKTER
      KONSTANSNAK NEVEZZUK"
      >2Ø END
*      >RUN

```

009

Az egydimenziós karakter tömb használata.

18

Oldja meg a 17 feladatot szavankénti beolvasással:

```

*      >NEW
      >1Ø FOR I=1 TO 7
      >2Ø READ A$( I)
      >3Ø DATA Az, IDEZOJELBE, TETT, SZOVEGET,
      KARAKTER, KONSTANSNAK, NEVEZZUK
      >4Ø PRINT A$( I); " ";
      >5Ø NEXT I
      >6Ø END
*      >RUN   ○ ↓ ○

```

△△ – A karakter tömb számok helyett szöveg-
részekkel dolgozik. A DATA sorokban a
vesszők jelölik ki a változók „értékeit”.
Az üres ” ” STRING, szóközt biztosít a
kiírásban.

19

Módosítsa a 18 programot arra az esetre, ha a szöveget három részre daraboljuk:

ABC: 20 A karakter konstans max. 3 képernyősor foglalható el, ha definiáljuk, de
 21 eredménye akár 6 sor is lehet *
 21 OK * Az egyesítés string számok (pl. A $\text{X} = "30"$: B $\text{X} = "600"$)
esetén is érvényes *

HTZ: 20 A karakter konstans max. 4 képernyősor 64 oszlopos üzem esetén
(VIDEO CUT ki) *
 21 OK *

TEXAS: 20 A karakter konstans max. 4 képernyősor *
 21 +jel \equiv & ; " " helyett K\$= " " ; így >20 PRINT A\$&K\$&B\$ *

PRIMO: 20 A karakter konstans max. 5 képernyősor *
 21 OK * Az egyesítés string számok (pl. A\$="30" : B\$="600") esetében is
elvégezhető *

TV COM: 20 OK *
 21 >20 PRINT A\$& " " & B\$ *

VT-16: 20 OK *
 21/A OK * Egészítse ki a programot >5 DEFSTR A, B-vel és a dollár jeleket
törölje a >10 és >20 utasításokból. Az >5 utasítás deklarálja az egész program
számára a változók string jellegét *

```
>10 FOR I=1 TO 3
>30 DATA AZ IDEZOJELBE TETT SZOVEGET, KARAKTER
    KONSTANSNAK, NEVEZZUK
*
>RUN  ⓪↓⓪
```

010 A karakter konstans alkalmazása.

20 Adjon értéket egy karakter változónak és írassa ki tartalmát:

```
*
>NEW
>10 A$="A TUDAS"
>20 PRINT A$ : END
*
>RUN  ⓪↓⓪
```

△△ – BASIC jelekből idézőjelbe szerkesztett karakter sorozatot karakter konstansnak nevezünk. Ilyen konstans karakter változóval azonosítható. A konstans hossza 2 képsorig terjedhet.

21 Adjon össze karakter konstansokat:

```
/A
>10 A$="A TUDAS":B$=" HATALOM"
>20 PRINT A$+" "+B$ : END
*
>RUN  ⓪↓⓪
```

```
/B
>10 A$="A TUDAS"
>20 PRINT A$+" HATALOM" : END
*
>RUN  ⓪↓⓪
```

△△ – Karakter konstansok összeadása értelmes mondatok szerkesztésére és string-ek bővítésére használható fel.

ABC: 22 OK* Próbálja meg az alábbi módosítással!

>1Ø A Ø = "2" : B Ø = "6"

>2Ø C Ø =ADD Ø (A Ø , B Ø , Ø):D Ø =SUB Ø (B Ø , A Ø ,

1) : E Ø = MUL Ø (A Ø , B Ø , 1) : F Ø = DIV Ø (A Ø , B Ø ,

6)* >3Ø PRINT C Ø , D Ø , E Ø , F Ø* 23 OK* 24 OK*

HTZ: 22 OK*

23 OK*

24 OK*

TEXAS: 22 OK*

23 OK*

24 OK*

PRIMO: 22 OK*

23 OK*

24 OK*

TV COM: 22 OK*

23 OK*

24 OK*

VT-16: 22 OK*

23 OK* Próbálja ki a >25 PRINT A\B, B\A, A MOD B, B MOD A

utasítást is*

24 OK*

011

Számok aritmetikája. Értékadás közvetlenül és LET-tel.

22

Adjon értéket az A és B változóknak. Számítsa ki az $A+B$, $A-B$, $A \cdot B$, A^B számértéket:

```
*
_____
>NEW
>1Ø A=2 : B=6
>2Ø C=A+B : D=A-B : E=A*B : F=A/B : G=A↑B
>3Ø PRINT C, D, E, F, G : END
*
>RUN   ○↓○
_____
```

△△ – Az $A=2$ (vagy **LET** $A=2$) értékadó utasítás, a **READ** utasítást helyettesíti. Az aritmetikai műveletek eredményeit külön változókkal (C, D, E, ...) azonosíthatjuk.

012

Műveletek összevonása. Algebrai kifejezések.

23

Oldja meg a 22 feladatot az eredmény azonosítók elhagyásával:

```
*
_____
>2Ø PRINT A+B, A-B, A*B, A/B, A↑B
>3Ø END
*
>RUN   ○↓○
_____
```

△△ – A **PRINT** alá aritmetikai műveletek is bevihetők.

24

Számítsa ki az $(5.8+2,3^2)^3 : (5^3 - 4.6, 1)$ kifejezés értékét:

```
*
_____
>NEW
>1Ø A=(5*8+2.3↑2)↑3/(5↑3-4*6.1)
>2Ø PRINT "A KIFEJEZES ERTEKE=" ; A:END
*
>RUN   ○↓○
_____
```


ABC: [25] OK* A módosítás menete: >ED1Ø [RET]*
[→] lenyomásával KURZOR az = felett* >%*
[→] lenyomásával KURZOR az utasítás végéig fut* [RET]* >ED 2Ø
hasonlóan*
[26] OK*

HTZ: [25] A módosítás menete: >EDIT1Ø [NL]* Lécbillentyű leütések;
KURZOR az = felett* >I* >% [NL]* >EDIT 2Ø [NL]* LÉCB;
KURZOR a : felett* >I* >% [NL]*
[26] OK*

TEXAS: [25] A gép csak a valós számokat ismeri. A programot lépje át, vagy írja át az
alábbiak szerint:
>1Ø A = (5*8+2.3 ^ 2) ^ 3/(5 ^ 3-4*6.1)
>2Ø PRINT INT (A) : : END*
[26] OK*

PRIMO: [25] OK* A módosítás menete: >EDIT1Ø [RET]*
[→] lenyomásával a KURZOR az = felett* >%*
[→] lenyomásával a KURZOR az utasítás végéig fut* [RET]* >EDIT 2Ø
hasonlóan*
[26] OK* A ?-et PRINT-re váltja a gép*

TV COM: [25] A módosítás menete: >LIST1Ø [RET]* A botkormánnyal a
KURZORT az = fölé állítja, majd [INS]* Leüti a % jelet és [RET]*
Tartósan lenyomva az [INS] gombot az üres pozíciók száma növekedik*
NEM*
[26] OK*

VT-16: [25] A módosítás menete: [8 ↑] lenyomásával KURZOR a >1Ø sorig lép*
[6 →] lenyomásával KURZOR az = jel alá áll* [INS⁰] leütése után >% [ENT]*
[2 ↓] leütésével KURZOR a >2Ø sorra lép és a fenti ismétlődik* Túlfutás esetén
[4 ←] leütésével lépünk vissza. A javítandó utasítások LIST paranccsal állíthatók
a képernyőre*
[26] OK*

△△

- A műveletek sorrendje: hatványozás, szorzás, osztás, összevonás. Ez a rend érvényesül a zárójelen belül és zárójelek között is. Az egynemű műveletek rendje, balról jobbra, egyenként kerülnek végrehajtásra. A zárójelek abszolút prioritása van.
- Ha a numerikus változó (A) jellege eltér a számolt érték jellegétől, az érték konvertálódik (pl. $A=5,26 \rightarrow A\%=5$).

25

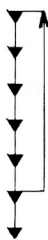
Módosítsa a 24 programban egész jellegűre az A változót és ismétlje meg a számolást:

```
>10 A%=(5*8+2.3↑2)↑3/(5↑3-4*6.1)
>20 PRINT "A KIFEJEZES ERTEKE=";A%;END
*
>RUN   ○↓○
```

26

Számítsa ki az $E=(A \cdot B^2 + C) : (D \cdot F - F^2)$ algebrai kifejezés értékét a [3, 7, 2, 4, 5] és a [6, 4, 4, 8, 9] pontokban:

```
*
>NEW
>10 FOR P=1 TO 2
>20 READ A,B,C,D,F
>30 DATA 3, 7, 2, 4, 5, 6, 4, 4, 8, 9
>40 LET E=(A*B↑2+C)/(D*F-F↑2)
>50 ? "A KIFEJEZES ERTEKE="; E
>60 NEXT P
>70 END
*
>RUN   ○↓○
```



ABC:

OK*

>ED5Ø *KURZOR E-re (törlendő mögé)* leüt*

KURZOR az utasítás végére** Több karaktert többszöri leütésével
törlünk, majd *

HTZ:

OK*

>EDIT5Ø * KURZOR ;-re* * >LIST* >RUN*

DELETE=töröl. minden leütése egy karakter törlését biztosítja* OK*

TEXAS:

OK*

≧5Ø * Megjelenik az utasítás*

≧ * KURZOR-t a vesszőre állítjuk*

≧ *

A ; elhagyása szintaktikai hibára vezet*

PRIMO:

OK*

>EDIT5Ø* KURZOR a ; mögé* * * Ha a KURZOR túlszalad,
akkor leütése után az EDIT-álás ismételhető*

TV COM:

OK*

>LIST5Ø* Álljon a KURZOR-ral a ; mögé (A botkormányt használja),
majd és * Tartósan lenyomva a gombot, az egész
előzmény törlődik* ; NEM törölhető*

VT-16:

OK*

>LIST 5Ø * leütésével KURZOR a ; alá, majd és
 , vagy >EDIT5Ø és ugyanaz, mint előbb*

013

A **RESTORE** utasítás használata.

27

Módosítsa a [26] programot úgy, hogy ismételten az első pontot vegye vizsgálat alá:

 >15 **RESTORE**

*

 >RUN

014

Szöveg-szerkesztés.

28

Törölje a [27] program >5Ø utasításából a pontosvesszőt és futtassa a programot:

*

 >**LIST 5Ø**
RET
 ≧ **SHIFT CR↑SR** : KURZOR AZ 5Ø CIMKÉRE.

 ≧ **CR→SR** : KURZOR A ; MÖGÉ.

 ≧ **INST/DEL**
RET
 ≧ **CR↓SR** : KURZOR A PROGRAM ALÁ.

*

 >RUN

△△

– A LIST 5Ø utasításra a javítandó utasítás megjelenik a képernyőn. A SHIFT és CR↑SR gombok egyidejű lenyomásakor a kurzor egy sorral fentebb ugrik, így elérhető a javítandó sor. A CR→SR gomb lenyomásakor a kurzor egy pozícióval jobbra lép, így elérhető a javítandó jel, ha a kurzor-t a javítandó jel mögé állítjuk és az INST/DEL gombot lenyomjuk, akkor a javítandó jel törlődik. A törlés csak a RET gomb lenyomásával memorizálódik. A kurzor a CR↓SR lenyomásával vihető le a program alá. Több jel törlését az utolsóval kezdjük. Másik megoldás:

ABC: INPUT után karakter konstans nem állhat * >1Ø megoldása:
>5 PRINT "A PARAMETER ERTEKE"*
>1Ø INPUT A*

HTZ: OK*

TEXAS: >1Ø INPUT "A PARAMETER ERTEKE?" : A*

PRIMO: OK* Próbálja ki az alábbi utasításokat -ben
>5 DEFINT B,C,D : DEFDBLE,F,G*
>5 DEFINT E,F,G : DEFDBL B,C,D
Az első esetben a B, C és D számok egész, az E, F, G számok pedig >7 jegy pontosak*

TV COM: INPUT után karakter konstans nem állhat >1Ø megoldása: >1Ø PRINT"
A PARAMETER ERTEKE ? " : INPUT A* Próbálja ki a >1Ø INPUT PROMPT
"A PARAMETER ERTEKE ? " : A megoldást is*

VT-16: OK* Próbálja ki az >5 DEFDBL G, illetve az >5 DEFDBL A-G utasítást*

>50 ? "A KIFEJEZES ERTEKE=" E

>RUN

*

△△ – Ez utóbbi csak egyszerű és rövid utasítások esetében ajánlható.

015

Az **INPUT** utasítás használata.

29

Írjon fel egy paraméterre hatványozási műveleteket, (pl. a^2 , \sqrt{a} , $\sqrt[3]{a^2}$, a^0 , a^{-3} , $a^{-\frac{4}{5}}$ stb.), és végezzen számításokat a >0 értékek behívásával:

*

>NEW

>10 INPUT "A PARAMETER ERTEKE?" ; A

>20 B=A↑2

>30 C=A↑(1/2)

>40 D=A↑(2/3)

>50 E=A↑0

>60 F=A↑(-3)

>70 G=A↑(-4/5)

>80 PRINT "A HATVÁNY=" ; B, C, D, E, F, G

>90 END

*

>RUN O↓O

△△ – Az **INPUT** bemenetet jelent. A választott számot (A) a gép kérésére le kell kopogni. A kérést a képernyőn megjelenő szöveg (pl. **A PARAMETER ERTEKE??**), ennek hiányában egy kérdőjel közvetíti. A lekopogást a kérés sorában kezdjük el. Legyen $A=400$:

|| A PARAMETER ERTEKE??400 ||

RET

ABC:

OK*

|| ERR 4 LINE 20 ||* A gép 0-66 hibakódokat ismeri. A 4 a NEM megengedett művelet jele* >10 INPUT A, B* >40 GOTO 5* A ciklusból

leütése visz ki*

HTZ:

OK*

|| ? FC ERROR IN 20 ||* FC NEM megengedett műveletet jelez
(-4) ↑ (1/2)*

,, helyett , így 32 oszlopos képernyő esetén is látjuk az eredményt*

TEXAS:

>10 :A*

|| BAD ARGUMENT IN 20 || (hibás argumentum)*

>10 :A, B* >20 PRINT "KERULET", "TERULET"* >30 , A*B/2*

A képernyő ugyanis 28 oszlopos*

PRIMO:

OK*

|| FC ERROR IN 20 ||*

OK* A ciklusból a leütése visz ki*

TV COM:

OK*

|| *** BAD ARGUMENT. ||
|| 20 PRINT A ^ 2 ... ||

>10 PRINT "A BEFOGOK HOSSZA?": INPUT A, B* >10 INPUT
PROMPT "A BEFOGOK HOSSZA?": A, B*

VT-16:

OK*

|| ILLEGAL FUNCTION CALL IN 20 ||* CALL = hiv*

OK*

30

Egyesítse a 29 programban a műveleti és kiírási fázisokat:

```

*      >NEW
▼      >1Ø INPUT "A PARAMETER ERTEKE?" ; A
▼      >2Ø PRINT A↑2, A↑(1/2), A↑(2/3), A↑Ø, A↑(-3), A↑(-4/5)
▼      >3Ø END
*      >RUN   ○↓○

```

31

Ismételje meg a futtatást negatív értéket adva A-nak (pl. A=-4):

```

|| ? ILLEGAL QUANTITY ERROR IN 2Ø ||

```

△△ – A hibajel értelmetlen műveletre utal ($\sqrt{-4}$).

016

A GOTO α , STOP és CONT utasítások használata.

32

Írjon alkalmazási programot két befogóval adott derékszögű háromszög kerület és területének a kiszámítására:

```

*      >NEW
▼      >1Ø INPUT "A BEFOGOK HOSSZA?"; A, B
▼      >2Ø PRINT "KERULET", "TERULET"
▼      >3Ø PRINT A+B+(A↑2+B↑2)↑(1/2), A*B/2
▼      >4Ø GOTO 1Ø
▼      >5Ø END
*      >RUN   ○↓○

```

```

|| A BEFOGOK HOSSZA??3,4
|| KERULET           TERULET
|| 12                6
|| A BEFOGOK HOSSZA??

```

ABC: ⇒ leütése "futás" közben hatásos. Ha a gép adatra vár, akkor összeférhetetlen adat (most pl. string) megadása vagy a leütése hatásos*

HTZ: ⇒ *

TEXAS: ⇒ *

PRIMO: ⇒ *

TV COM: ⇒ *

VT-16: ⇒ *

△△

- Egy input utasításban több paraméter is szerepelhet. Ezeket vesszővel kell elválasztani az utasításban és az értékadás lekopogásánál is. Ha két kérdőjel jelenik meg a képernyőn, a sor elején, akkor vagy hibásan gépeltük az értéket, vagy kevesebb adatot gépeltünk, mint ahány paramétert deklaráltunk. A hiba ismételt értékadással korrigálható.
- A GOTO α vezérlés-átadó utasítás. Hatására nem a soronlevő utasítást hajtja végre a gép, hanem az α sorszámú utasításra lép és onnan halad tovább, követve a sorszámok természetes rendjét.

33

Állítsa le a program futását:

» **STOP RESTORE**

△△

- Ha a GOTO α címkéje nagyobb α -nál, akkor az α -val kezdődő program részlet a végtelenségig ismétlődne. Ezt, ha INPUT is van a program részletben, mindig a jelzett két gomb egyidejű lenyomásával állítjuk meg.

→

Ha van lemezes egysége, előrelapozhat a **70** programhoz!!!

017

Az **IF–THEN** és a **STOP** utasítások használata.

35

Írjon alkalmazási programot arra az esetre, amikor a derékszögű háromszög átfogója és egyik befogója adott:

ABC: >60 **STOP** leállítja a futást, de a folytatásra **nincs** utasítás. END≡STOP*
>60 *

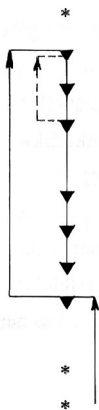
HTZ: OK*

TEXAS: >10 : A* >20 : C* >30 :: GOTO 10* >40, "TERULET"* >50,
((C ^ 2 - ...)*) >60 **BREAK*** >CON*

PRIMO: OK*

TV COM: >CONTINUE * Akkor is alkalmazható, ha -vel
állítjuk meg a futást és a program érintetlen marad, (NEM javítjuk, módosítjuk
stb.*

VT-16: OK*



```

>NEW
>1Ø INPUT "A BEFOGO HOSSZA?"; A
>2Ø INPUT "AZ ATFOGO HOSSZA?"; C
>3Ø IF A>-C THEN PRINT "DEGENERALT
      HAROMSZÖG" : GOTO 1Ø
>4Ø PRINT "KERULET", "TERULET"
>5Ø PRINT A+(C↑2-A↑2)↑(1/2)+C ,, ((C↑2-A↑2)↑(1/2)) * (A/2)
>6Ø STOP
>7Ø GOTO 1Ø
>8Ø END
>RUN   ◯↓◯   → ◻
>CONT

```

RET

- △△ – Egy INPUT utasításban eltérő „megnevezéssel” két paraméter nem szerepelhet (>1Ø és 2Ø).
- A program végrehajtási rendjét feltételek szabályozhatják. Ezeket relációkkal fejezzük ki: <, ≤, =, ≥, >, ≠. Ilyen relációk változók, konstansok és kifejezések értékei között állhatnak fenn⁵ (pl. A>C). A feltételt az IF vezeti be. Ha az IF-et követő reláció teljesül, akkor a THEN kulcsszót követő utasítás vagy utasítások, ellenkező esetben a soronkövetkező (4Ø) kerül végrehajtásra.
 - A STOP megállítja a program futását. Az újraindítás a >CONT parancs lekopogásával érhető el.

⁵ Minden reláció logikai függvénynek felel meg, ami kétértékű: –1, ha a reláció igaz (pl. 2<3) és 0, ha nem igaz (pl. 2>3).

ABC: [36] >PRINT 5EQV NOT 3* >PRINT NUM $\text{\textcircled{X}}$ (255)* >PRINT "5" +
+ NUM $\text{\textcircled{X}}$ (255)* >PRINT VAL (NUM $\text{\textcircled{X}}$ (52))* LOG az e, LOG10 a 10
alapú log. függvény* $\pi=PI$ * >PRINT SGN (SIN(PI)) pontatlan* >PRINT
"A" CHR $\text{\textcircled{X}}$ (10) "B"* SPC funkció nincs* FRE és TI nem működik*

HTZ: [36] "ALMA">"ALBA"<"ALFA"* A XOR logikai műveletet nem ismeri*
TAB(35) "A" jobboldali képernyőn áll* SPC () funkcióval nem rendelkezik*
A PEEK fgv max. címe: 32767* FRE(\emptyset) \equiv MEM is működik* TI, TI\$ nem
értelmezett*

TEXAS: [36] "ALMA">"ALBA"<"ALFA"* LEFT\$ helyett SEG\$(" ", 1, 3)*
RIGHT\$ helyett SEG\$(" ", LEN(" ") - 4, 5)* MID\$ = SEG\$*
TAB(2 \emptyset); "A"* TAB(35); "A" = TAB(7); "A"* "A" CHR\$(13) "B" = "A"
: "B"* A — — — alatt nem értelmezhetők*

PRIMO: [36] XOR logikai műveletet nem ismeri* $\pi=PI$ * SPC funkciót nem ismeri*
PEEK (I), I<32768, ha PRIMO-32-vel dolgozik*

TV COM: [36] >PRINT ("KATI">"VIRAG")* >PRINT ("ALMA"<"ALBA")*
>PRINT(5=NOT3)* LEFT\$, RIGHT\$, MID\$ helyett egységes szerelendő utasítás:
>A\$ = "BOKRÉTA": PRINT A\$ (:3), A\$(:4), A\$(3:), A\$(5:), A\$(2:3), A\$(4:6),
vagy >PRINT "BOKRETA" (:3) stb.* ASC \equiv ORD* $\pi=PI$ *
CHR\$(13), CHR\$(8) nem működik* SPC(4) \equiv STRING\$(4, " ")* >PRINT
FREE* TI nem értelmezett* VAL ("52ALMA") nem működik*

VT-16: [36] $\pi=3.1415$ * A szóközökre érzékeny a gép*

>PRINT 2<3

```

2>3
"KATI">"VIRAG"
"ALMA"<"ALBA"
"ALMA"<"ALFA"
2<3 AND 5<8
2<3 AND 5>8
2<3 OR 5<8
2>=3 OR 5>8
2<3 XOR 5<8
2<3 XOR 5>8
5=NOT 3
3=NOT 3
-4=NOT 3
NOT 5
LEN("BOKRETA")
LEŇ("KRETA")
LEFT$("BOKRETA",3)
LEFT$("BOKRETA",4)
RIGHT$("BOKRETA",5)
RIGHT$("BOKRETA",3)
MID$("BOKRETA",2,2)
MID$("BOKRETA",4,3)
STR$(255)
5+STR$(255)
"5"+STR$(255)
ASC("A")
ASC("1")
ASC("BOKRETA")

```

RET

>PRINT VAL("52")

```

VAL(STR$(52))
CHR$(0)
CHR$(1)
ABS(-8.6)
EXP(2)
TAB(20)"A"
TAB(35)"A"
LOG(2.7)
LOG(10)
SIN( $\pi/2$ )
SIN( $90*\pi/180$ )
SGN(-6)
SGN(+6)
SGN(SIN( $\pi$ ))
TAN( $\pi/4$ )
"A"CHR$(13)"B"
"A"CHR$(8)"B"
"A"SPC(4)"B"
"A"SPC(8)"B"
PEEK(53280)
PEEK(53281)
PEEK(60000)
FRE(0)
FRE(1)
TI
TI$
VAL("52ALMA")
VAL("52AL25")

```

RET

ABC: 37 OK*

38 OK*

HTZ: 37 OK*

38 OK*

TEXAS: 37 OK*

38 OK* vagy >20 FCTN X FCTN 3 *

PRIMO: 37 OK*

38 OK*

TV COM: 37 OK*

38 OK*

VT-16: 37 OK* Futtassa $K \leq 10$ -el és próbálja meg >5 DEFINT A-Z mellett $K \leq 10$ -el*

Mérje a futások időtartamát.*

38 OK*

018

A változó léptetése értékadással. Segédváltozók.

37

Írjon alkalmazási programot a $k!$ (k elemű permutációk⁶ száma) kiszámítására:

```

*          _____
          >NEW
          >1Ø INPUT "AZ ELEMEEK SZAMA?"; K
          >2Ø A=1 : N=1
          >3Ø A=A*N : N=N+1
          >4Ø IF N<=K THEN 3Ø
          >5Ø PRINT "K!="; A
          >6Ø END
*
          >RUN   ○↓○   →   □•
  
```

△△ – A $K!$ az első K természetes szám szorzatával egyenlő. A természetes számokat az N segédváltozó szállítja az $N=N+1$ értékadás útján (N új értéke egyenlő N régi értéke + 1), amíg a $>4Ø$ teljesül. A már megkapott számok szorzatát az A segédváltozóhoz rendeljük (A új értéke egyenlő régi értéke szer új N). A segédváltozók kezdőértékét deklarálni kell ($A=1$; $N=1$), mert ellenkező esetben nullának számítanak. Az értékadás a változó bármely függvényére⁷ támaszkodhat (pl. $A=\text{SIN}(A)$ stb.)

38

Törölje a [37] programból a $>2Ø$ utasításokat és futtassa ismét a programot:

```

          _____
          >2Ø                                     RET
*          >RUN
  
```

△△ – Bármely K esetén nullát kap eredményül, mert a futás kezdetén $A=Ø$ $N=Ø$ stb.

⁶ dr Denkinger Géza: Valószínűség-számítás. TK. Budapest, 1978.

⁷ dr. Szép Jenő: Analízis. KJK. Budapest, 1972.

ABC: >30 LET $Y = 3 * X * X + 2 * X - 1$, ha negatív argumentum is előfordul, mert a gép $2 * \text{LOG}(X)$ alapon hatványoz*

OK *

HTZ: OK*

OK*

TEXAS: OK*

OK*

PRIMO: OK*

OK*

TV COM: OK*

OK*

VT-16: OK*

OK*

019

A FOR-TO-STEP-NEXT utasítás használata.

39

Írjon leképező (értéktáblázat készítő) programot adott halmaz és leképező függvény esetére Legyen $x(-1, 0, 1, 2, 3, 4, 5)$; $y=3x^2+2x-1$:

```

*      >NEW
      >1Ø FOR I=1 TO 7 : READ X
      >2Ø DATA -1, 0, 1, 2, 3, 4, 5
      >3Ø LET Y=3*X^2+2*X-1
      >4Ø PRINT X; Y : NEXT I
      >5Ø END
*      >RUN  ○↓○ → □.

```

40

Oldja meg a 39 feladatot X léptetésével:

```

/A      >1Ø FOR X=-1 TO 5 STEP 1
      >2Ø
      >4Ø PRINT X;Y : NEXT X
*      >RUN  ○↓○ → □.

```

RET

```

/B      >1Ø X=-1
      >4Ø PRINT X; Y: X=X+1
      >45 IF X<=5 THEN 3Ø
*      >RUN  ○↓○ → □.

```

- △△ – A FOR X=A TO B STEP L utasítás az [A, B] intervallum A, A+L, A+2L, ..., A+KL pontjait rendeli a ciklus-változóhoz. Így K számú ciklust generál a NEXT X közreműködésével. Ha L=1 a STEP 1 el is hagyható.
- A bemutatott megoldások bizonyítják a program alternációk lehetőségét.

ABC:

OK*

\gg TARTÓSAN NYOMJA LE A GOMBOT**

HTZ:

OK*

, ha SYSTEM / 12288-al dolgozik*

TEXAS:

$>3\emptyset$ IF $X + 1 = \emptyset$ THEN 55* $>4\emptyset$ IF $((X \wedge 2 - 1)/(X+1)) < \emptyset$ THEN 56*
 $>5\emptyset$ PRINT "X="; X, "Y="; $((X \wedge 2 - 1)/(X+1)) \wedge (1/2)$:: GOTO 60*
 >55 PRINT "AZ OSZTO NULLA" :: GOTO 60*
 >56 PRINT "A GYOKALAP NEGATIV"* $>$ CALL CLEAR*

PRIMO:

$>5\emptyset$ PRINT "X="; X, $((X \uparrow 2 - 1)/(X+1)) \uparrow (1/2)$ *

$>$ CLS*

TV COM:

OK*

$>$ CLS* tartósan lenyomva*

VT-16:

OK*

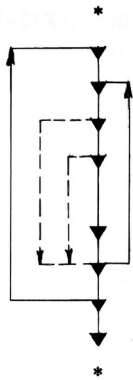
**

020

AZ IF lánc alkalmazása. A logikai kifejezés.

41

Készítsen értéktáblázatot az $y = \sqrt{\frac{x^2-1}{x+1}}$ függvényhez az $a \leq x \leq a+h$ intervallum $h/10$ lépéshosszal adott pontjairól:



>NEW

>10 INPUT "INTERVALLUM ADATOK"; A, H

>20 FOR X=A TO A+H STEP H/10

>30 IF X+1=0 THEN PRINT "AZ OSZTO NULLA" : GOTO 60

>40 IF ((X↑2-1)/(X+1))<0 THEN PRINT

"A GYOKALAP NEGATIV" : GOTO 60

>50 PRINT "X="; X,, "Y="; ((X↑2-1)/(X+1))↑(1/2)

>60 NEXT X : STOP

>70 GOTO 10

>80 END

>RUN ○↓○ →

△△

– A FOR utasításban algebrai kifejezések ($A+H$, $H/10$) is szerepelhetnek. A gép kiszámítja értéküket, ha a paramétereket rögzítjük (A , H).

– Az IF láncot akkor alkalmazzuk, ha egyidejűleg több lehetőséget kell megvizsgálni és eltérnek a következtetések, (>30, >40, >50). Ha valamely lehetőség nem következik be, a soronkövetkező kerül vizsgálat alá.

42

Törölje a képernyőt, majd kérjen listát a 41 programból:

 ≧ CLR/HOME SHIFT

*

>LIST

ABC: OK*

HTZ: Nullával való osztásra a gép \emptyset saját hibajelét jelzi ki. Így $>3\emptyset$ IF $((X \uparrow 2 - 1) / (X + 1)) < \emptyset$ THEN PRINT "A FUGGVENY NINCS ERTELMEZVE" :
GOTO 6 \emptyset *

TEXAS: $>3\emptyset$ IF $X+1=\emptyset$ OR $((X \wedge 2 - 1) / (X + 1)) < \emptyset$ THEN 55* $>4\emptyset$ *
 >55 PRINT "A FUGGVENY NINCS ERTELMEZVE"* >56 *

PRIMO: OK* A nullával való osztásra \emptyset ERROR IN α jellel reagál a gép, ezért $>3\emptyset$ -ból törölhetjük az $X+1=\emptyset$ feltételt*

TV COM: OK* A nullával való osztásra \parallel *** CAN NOT DIVIDE BY \emptyset \parallel jellel reagál a gép*

VT-16: OK*

```
>30 IF X+1=0 OR ((X↑2-1)/(X+1))<0 THEN PRINT "A FUGGVENY
NINCS ERTELMEZVE": GOTO 60
```

```
>40
```

RET

```
*
```

```
>RUN ○↓○ → □
```

△△

- Logikai függvények (relációk) között értelmezhető a konjunkció (műveleti jele: AND), a diszjunkció (műveleti jele: OR) és a negáció (műveleti jele: NOT). R_1 AND R_2 csak akkor igaz, ha R_1 és R_2 is teljesül. R_1 OR R_2 igaz ha, legalább az egyik reláció teljesül. Not R akkor igaz, ha R nem teljesül.
- Az AND, OR és NOT műveletekkel összetett logikai függvény (logikai kifejezés) jön létre, ami szintén kétértékű: -1 (igaz) és 0 (nem igaz, hamis). A THEN kulcsszót a -1 aktiválja, (részletesen a 2. számú mellékletben).
- Esetünkben akár a nevező nulla, akár a tört negatív valamely x helyen, ott a függvény nincsen értelmezve.

021

A $SQR(X)$ függvény alkalmazása

44

Írjon alkalmazási programot az $ax^2+bx+c=0$ egyenlet megoldására. Ellenőrzésként építse be a gyökök és együtthatók összefüggéseit:

ABC: 44 >1Ø PRINT'' AZ '''* >15 INPUT A, B, C*
>2Ø LET D = B * B - 4 * A * C* >5Ø : GOTO 1Ø*
>7Ø Törölve*
 45 OK*

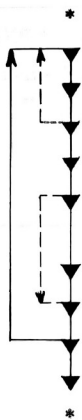
HTZ: 44 OK*
 45 OK*

TEXAS: 44 >1Ø : A, B, C* >3Ø IF D<Ø THEN 1Ø* >5Ø THEN 65*
>6Ø PRINT ''PONTATLAN MEGOLDAS''; X1, X2 : : GOTO 7Ø*
>65 PRINT ''X1=''; X1, ''X2 =''; X2* >7Ø BREAK*
 45 OK*

PRIMO: 44 OK*
 45 OK*

TV COM: 44 OK*
 45 OK*

TV-16: 44 OK*
 45 OK* A REM '-vel helyettesíthető, ha a megjegyzést nem követi : és újabb utasítás*



```

>NEW
>10 INPUT "AZ A, B, C EGYUTTHATOK ERTEKE?"; A, B, C
>20 LET D=B↑2-4*A*C
>30 IF D<0 THEN PRINT "NINCSEN VALOS GYOK": GOTO 10
>40 X1=(-B+SQR(D))/(2*A) : X2=(-B-SQR(D))/(2*A)
>50 IF X1+X2=-B/A AND X1*X2=C/A THEN PRINT "X1=" X1,
      "X2=" X2 : GOTO 70
>60 PRINT "PONTATLAN MEGOLDAS" X1, X2
>70 STOP
>80 GOTO 10
>90 END
>RUN  ◯ ↓ ◯ → ◻

```

- △△
- A négyzetgyökvonás gyakorisága miatt beépítették a gépbe a \sqrt{x} függvényt, amely az SQR jellel aktiválható. A gyökalap (változó vagy kifejezés) mindig zárójelbe kerül.
 - Az >50 utasítás R1 AND R2 típusú logikai kifejezésre épül. A megoldás akkor pontos, ha mindkét $(X1+X2=-B:A ; X1 \cdot X2=C:A)$ összefüggést kielégíti. Egyéb esetben az X1 és X2 közelítően pontos (esetleg hibás), ami a gépi kerekítések miatt gyakori.

45

Építsen be értelmező címet a 44 programba és futtassa újra:

44

/A

```
>5 REM A MASODFOKU EGYENLET MEGOLDASA
```

*

```
>RUN
```

/B

```
>5 PRINT "A MASODFOKU EGYENLET MEGOLDASA"
```

*

```
>RUN
```

ABC: >20 PRINT "A KOR SUGARA CM?": INPUT R*: >50 A = A * PI/180*
>70 I = R*A : K1 = I* R/2 : K2 = (R * R/2) * (A - SIN(A))* : >90 : K1 = INT
(100 * K1 + .5)/100 : K2 = INT (100 * K2 + .5)/100* : >110 PRINT "I =": I,
"K1 =": K1, "K2=": K2*

HTZ: OK*

TEXAS: >120 BREAK*

PRIMO: >50 A = A * PI/180*

TV COM: >50 A = A * PI/180*

VT-16: OK*

△△

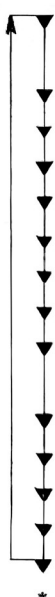
- A REM megjegyzést jelent. A megjegyzés szövege tájékoztatja az alkalmazót, ha a program listáját felírhatja.
- A PRINT alatt közölt „megjegyzés” a program futásakor a képernyőn jelenik meg, tehát akkor is elének kerül, ha alkalmazás előtt listát nem kérünk.

→ Ha van printere, előre lapozhat a 96 programhoz!!!

022 A SIN(X) és INT(X) függvények használata.

46 Írjon alkalmazási programot a kör alkotórészeinek (körív, körcikk, körszelet)⁸ kéttizedes pontosságú kiszámítására:

```
*
>NEW
>10 PRINT "AZ IVHOSSZ, A KORCIKK ES KORSZELET
    TERULETE"
>20 INPUT "A KOR SUGARA CM?"; R
>30 INPUT "A KOZEPPONTI SZOG FOK?"; A
>40 REM A SZOG ATSZAMITASA RADIANBA
>50 A=A*3.14/180
>60 REM IVHOSSZ ES TERULETEK
>70 I=R*A : KC=I*R/2 : KS=(R↑2/2)*(A-SIN(A))
>80 REM KEREKITES KET TIZEDESRE
>90 I=INT(100*I+0.5)/100 : KC=INT(100*KC+0.5)/100 : KS=INT(100*
    KS+0.5)/100
>100 REM EREDMENY KOZLES
>110 ? "I="; I, "KC="; KC, "KS="; KS
>120 STOP
>130 GOTO 10
>140 END
>RUN
```



⁸ dr. Farkas Miklós (szerk): Matematikai Kislexikon. MK. Budapest, 1974.

ABC: >90 PRINT "SZAMTANI", "MERTANI"* >103 PRINT "HARM",
"KRONO"*

HTZ: OK*

TEXAS: >30 DIM (100)*

Dimenzióként változó nem szerepelhet, így megfelelően nagy, valós típusú, egész
számot használunk*

PRIMO: >90 PRINT "SZAMTANI", "MERTANI"* >103 PRINT "HARM",
"KRONO"*

TV COM: OK*

VT-16: OK*

ABC: >15 PRINT " >51 DATA. . . (KITOLTESE), MAJD : REM STOP ES RUN"
 : STOP* RUN utasítás közbenső címkétől nem adható, így a STOP-ot magyarázat-
 tá degraláljuk a >15-ben az adatok lekopogása után*

HTZ: >106 PRINT ">51 NEW LINE" : END*

TEXAS: >106 PRINT">51 ENTER" : : END*

PRIMO: OK *

TV COM: OK *

VT-16: >106 PRINT ">51 ENTER" : END*

- △△ – Egy programot „alkalmazási” jelzővel akkor illetünk, ha változó helyzetben is alkalmazható. Változhat pl. az adatok száma, értéke stb. a DIM utasítás parametrikusan is megadható: DIM D(N). Az N-et INPUT-tal adjuk meg a DIM utasítást megelőzően.
- A tömelemek behívása (INPUT D(I)) FOR–NEXT ciklussal történik. Az adatokat a gép kérésére, egyenként kopogjuk le RET-el zárva.

48

Módosítsa 47-et adatbeolvasással való megoldásra:

```

>>TOROLJE 105-BOL AZ END-ET:
>15 PRINT ">51 DATA . . . (KITOLTESE), MAJD
>RUN 20" : STOP
>50 READ D(I)
>106 PRINT ">51 RET (TOROL)" : END
>RUN 0 ↓ 0 → : ^

```

*

- △△ – A >15 és >106 parancsot ír a képernyőre, amit STOP után teljesítünk. Lekopogjuk az >51 utasítást (adatokkal), majd RUN20-al folytatjuk a program futását. A program teljesítése után >51-et töröljük a gép kérésére. Ha N nagy és az adatok egy DATA utasításba nem férnek be, akkor >52 DATA, . . ., >59 DATA utasítások is megnyithatók, de törlésükről is gondoskodni kell a futás végén.

024

A GOSUB–RETURN utasításpár használata.

ABC: OK *

HTZ: OK *

TEXAS: >80 PRINT "LANCV =" ; L (J), "BAZISV =" ; B(J) *

PRIMO: OK *

TV COM: OK *

VT-16: OK *

ABC: >5Ø FOR I = 1 TO 15 * >9Ø NEXT I : GOTO 2ØØ * A ciklusindex nem fut a határon (15) túl * >22Ø PRINT I ; TAB (11) "C (I) = " ; C (I); TAB(21) "D(I) = " ; D(I); TAB(31) "E(I) = " ; E(I) * >25Ø PRINT I ; TAB(4) "F(I) = " ; F(I) *

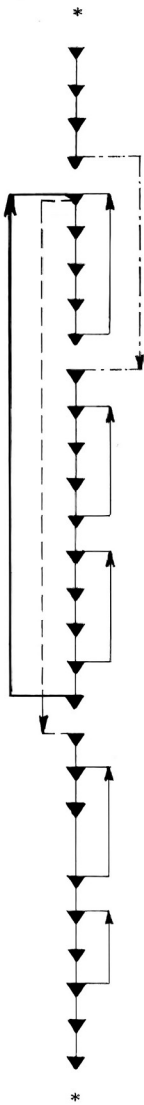
HTZ: OK *

TEXAS: >22Ø PRINT I ; TAB(1) ; "C(I) = " ; C(I) ; TAB(11) ; "D(I) = " ; D(I) ; TAB(21) ; "E(I) = " ; E(I) * >25Ø PRINT I ; TAB(4) ; "F(I) = " ; F(I) *

PRIMO: >5Ø FOR I=1 TO 15 * >9Ø NEXT I : GOTO 2ØØ *

TV COM: : >9Ø NEXT I : GOTO 2ØØ * >22Ø PRINT I ; TAB(11) ; "C(I) = " ; C(I) ; TAB(21) ; "D(I) = " ; D(I) ; TAB(31) ; "E(I) = " ; E(I) * >25Ø PRINT I ; TAB(4) ; "F(I) = " ; F(I) * Próbálja meg >GRAPHICS 2 után futtatni a programot * OUTPUT stop: *

VT-16: >5Ø FOR I=1 TO 15 * >9Ø NEXT I : GOTO 2ØØ * megállapítja az OUTPUT listát és bármely karakter folytatást biztosít *



```

>NEW
>10 PRINT "VEKTOR ARITMETIKA"
>20 INPUT "ADJA MEG LAMBDAT"; L
>30 DIM A(15), B(15), C(15), D(15), E(15), F(15)
>40 GOSUB 100 : G=0
>50 FOR I=1 TO 15 : IF I>15 THEN 200
>60 C(I)=A(I)+B(I) : D(I)=A(I)-B(I)
>70 E(I)=L*C(I) : F(I)=A(I)*B(I)
>80 G=G+F(I)
>90 NEXT I
>100 PRINT "ADATBEOLVASO SZUBRUTIN"
>110 FOR I=1 TO 15
>120 READ A(I)
>130 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
>140 NEXT I
>150 FOR I=1 TO 15
>160 READ B(I)
>170 DATA 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
>180 NEXT I
>190 RETURN
>200 PRINT "ADATKIIRO SZUBRUTIN"
>210 FOR I=1 TO 15
>220 PRINT I TAB(11) "C(I)="; C(I) TAB(21)
      D(I)="; D(I) TAB(31) "E(I)="; E(I)
>230 NEXT I: FOR P=1 TO 2000 : NEXT P
>240 FOR I=1 TO 15
>250 PRINT I TAB(4) "F(I)="; F(I)
>260 NEXT I
>270 PRINT "G="; G
>280 END
>RUN

```

ABC: 52 >8∅ nem módosul 51 -ben* OK*

HTZ: 52 OK*

TEXAS: 52 >3∅ DIM A (1∅∅), B (1∅∅), ...*

PRIMO: 52 >8∅ nem módosul 51 -ben*

TV COM: 52 >8∅ G = G + F(I)*

Próbálja meg az adatkiíró szubrutin PRINT-jei helyére LPRINT-eket
helyettesíteni *

VT-16: 52 OK*

- ΔΔ
- Egy program bevezető elemekből (pl. >1Ø), főprogramból (pl. >2Ø-9Ø, >19Ø, >28Ø) és szubrutinokból (pl. >1ØØ-14Ø, >15Ø-18Ø, >21Ø-23Ø, >24Ø-27Ø) állhat. A szubrutin valamilyen komplett részfeladatot megoldó „program”. A szubrutin behívása a GOSUB α (α a szubrutin kezdő címkéje) utasítással történik, elbocsájtásáról a RETURN gondoskodik.
 - A TAB(C) a formalizálás eszköze. Biztosítja, hogy az utána írt adat a képernyő c-edik oszlopában kezdjen megjelenni. A képernyő 4Ø oszlopra van felosztva, így $c < 4Ø$ lehet.
 - A képernyő méretét meghaladó PRINT a STOP gombbal megszakítható (ezt a >CONT parancs semlegesíti), vagy CTRL -el lassítható.

52

Módosítsa 51 -et n elemű és INPUT-tal behívható tömbök esetére:

```

>15 INPUT "A TOMB ELEMSZAMA?" ; N
>3Ø DIM A(N), B(N), C(N), D(N), E(N), F(N)
>5Ø FOR I=1 TO N
>8Ø G=G+F(I) : IF I=N THEN GOTO 2ØØ
>11Ø FOR I=1 TO N
>12Ø INPUT "A KOVETKEZO A(I) ELEM?" ; A(I)
>13Ø
>14Ø NEXT I : PRINT : PRINT
>15Ø FOR I=1 TO N
>16Ø INPUT "A KOVETKEZO B(I) ELEM?" ; B(I)
>17Ø
>21Ø FOR I=1 TO N
>24Ø FOR I=1 TO N
>27Ø PRINT "A VEKTOROK SZORZATA="G

```

RET

RET

ABC: OK*

HTZ: A kiírás az gomb lenyomásával lassítható* OK*

TEXAS: >1Ø :: PRINT N; "'=' ;* >31 PRINT I; "'.' ;* >55 :: PRINT TAB(25)
; K*

PRIMO: OK*

TV COM: >1Ø : PRINT N ; "'=' ;* >31 PRINT I; "'.' ;*
>55 : PRINT TAB(3Ø) ; K*

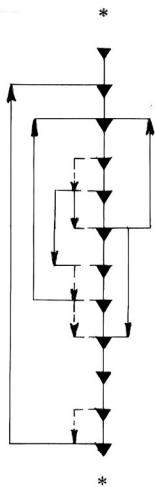
VT-16: OK* A program 25 sec alatt fut végig. Ez független attól, hogy DEFINT
A-Z, illetve egyszeres pontosságú valós számokkal dolgozunk*

026

IF–THEN–ELSE utasítás használata.

53

Bontsa fel az első 100 természetes számot prímtenyezőire. Számlálja meg a valódi osztóval nem rendelkező számokat:



>NEW

>5 A=2 : K=0

>10 N=A : P=0 : PRINT : PRINT N'=';

>20 FOR I=2 TO A^(1/2)

>30 IF INT (N/I)=N/I THEN 31 ELSE 40

>31 PRINT I'.'; : P=P+1 : GOTOT 50

>40 NEXT I : GOTO 55

>50 IF N/I>1 THEN 51 ELSE 55

>51 N=N/I : I=0 : GOTO 20

>55 IF P=0 THEN K=K+1 : PRINT TAB(30) K

>56 IF INT (N/I) <> N/I OR I>A^(1/2) THEN PRINT N

>60 A=A+1 : IF A=100 THEN END

>70 I=0 : GOTO 10

>RUN

△△ – Egy N természetes szám valódi osztója I , ha $\text{INT}(N/I)=N/I$ és $1 < I < N$ teljesül. Az I számok a $2 \leq I \leq \sqrt{N}$ intervallumban kereshetők. Az **IF R THEN α ELSE β** utasítást akkor használjuk, ha az „igaz” következménye 2-nél több utasítás. Az α , illetve β mutatja, hogy hová helyeztük el az „igaz”, illetve a „hamis” következményeit.

027

Logikai függvény karakter változókkal. Karakter konstansok **INPUT**-ja.

54

Írjon alkalmazási programot kamatos-kamat, annuitás és kölcsöntörlesztés, valamint inverzeik¹² kiszámítására:

¹² Matematikai közgazdasági gimnáziumok II–IV. osztálya számára. TK. Budapest.

ABC: >2Ø PRINT "MI KERUL KISZAMITASRA?" : INPUT A Ø* STOP törölve* >6Ø IF A Ø <> "FELKAMATOLTTOKE" THEN 11Ø*
Azonosító szöveget egybeírjuk* >7Ø : INPUT K(Ø)* Ugyanaz a jel számot (KØ) és tömböt (K(N)) nem azonosíthat* Hasonló a helyzet AØ, A(N) és TØ, T(N) esetében, ami A(Ø) és T(Ø) jelöléssel oldható fel*

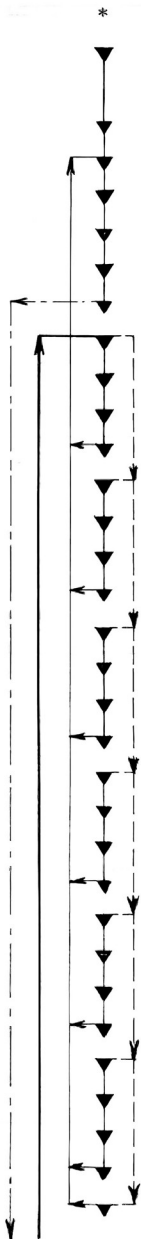
HTZ: OK*

TEXAS: OK*

PRIMO: OK*

TV COM: >5 DIM R(3Ø), D(3Ø), E(3Ø), K(3Ø), T(3Ø), A(3Ø) >5Ø TÖRÖLVE*
Ismételt számolás esetén az >5Ø dimenzionálás hibajelre vezetne* Ha GO TO 2Ø helyett **GOTO 4ØØ**-at alkalmazunk a szubrutinok végén, akkor
 OK*

VT-16: >2Ø CLEAR : INPUT "MI KERUL KISZAMITASRA?" ; A\$*
A CLEAR törli a változók értékét és így ismételt számítások esetén az >5Ø DIM nem vezet tilos átdimenzionálásra*



> NEW

```

> 10 PRINT "SZAMITHATO: FELKAMATOLT TOKE,
DISZKONTALT TOKE, JARADEKOSSZEG, JARADEKTAG"
> 15 PRINT "KOLCSONOSSZEG, TORLESZTOTAG"
> 20 STOP: INPUT "MIKERUL KISZAMITASRA?"; A$
> 30 INPUT "HANY% A KAMATLAB?"; P
> 40 INPUT "MENNYI AZ EVEK SZAMA?"; N
> 50 DIM R(N), D(N), E(N), K(N), T(N), A(N)
> 55 GOSUB 370
> 60 IF A$ <> "FELKAMATOLT TOKE" THEN 110
> 70 INPUT "KEZDOTOKE"; K0
> 80 K(N)=K0* R(N)
> 90 PRINT "FELKAMATOLT TOKE="; K(N):GOTO 20
> 110 IF A$ <> "DISZKONTALT TOKE" THEN 160
> 120 INPUT "ALAPTOKE"; K(N)
> 330 K0=K(N)*D(N)
> 140 PRINT "A DISZKONTALT TOKE="; K0 : GOTO 20
> 160 IF A$ <> "JARADEKOSSZEG" THEN 210
> 170 INPUT "JARADEKTAG"; A0
> 180 A(N) = A0*E(N)
> 190 PRINT A$;"="; A(N) : GOTO 20
> 210 IF A$ <> "JARADEKTAG" THEN 260
> 220 INPUT "JARADEKOSSZEG"; A(N)
> 230 A0=A(N)*E(N)↑(-1)
> 240 PRINT A$;"="; A0 : GOTO 20
> 260 IF A$ <> "KOLCSONOSSZEG" THEN 310
> 270 INPUT "TORLESZTOTAG"; T0
> 280 T(N) = T0*D(1)*D(N)*E(N)
> 290 PRINT A$;"="; T(N): GOTO 20
> 310 IF A$ <> "TORLESZTOTAG" THEN 360
> 320 INPUT "KOLCSONOSSZEG"; T(N)
> 330 T0=T(N)*R(1)*R(N)*E(N)↑(-1)
> 340 PRINT A$;"="; T0: GOTO 20
> 360 PRINT "A KEREST NEM ERTEM": GOTO 20

```

ABC: 55 OK, ha A \emptyset helyett A(\emptyset) és T \emptyset helyett T(\emptyset) áll *

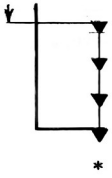
HTZ: 55 OK *

TEXAS: 55 OK *

PRIMO: 55 OK *

TV COM: 55 >6 DIM R(3 \emptyset), ... *

VT-16: 55 >2 \emptyset CLEAR: *



```

>37Ø R(1)=1+P/1ØØ : R(N) = R(1)↑N : D(1) = R(1)↑(-1)
>38Ø D(N) = R(N)↑(-1) : E(N) = (R(1)*(R(N)-1))/(R(1)-1)
>39Ø RETURN
>40Ø END
>RUN  ⓪ ↓ ⓪ → □ ⓪

```

- △△
- A BASIC karakter kifejezések között is értelmezi a <, ≤, =, ≥, > és <> (nem egyenlő) relációkat. Ezt úgy oldja meg, hogy minden karakterhez azonosító kódot (ASC kód) rendel és ezeket páronként összehasonlítja. Az első eltérés alapján dönti el a relációt. Ha pl. kód (A) <... <kód(Z), akkor BARNA<BARTA, mert "N"<"T". Így string-ek is szerepelhetnek logikai függvényekben.
 - INPUT utasítás karakter kifejezésekre is felírható, ha string (pl. A\$) változót adunk meg. Az idézőjelbe tett szám (pl. "1985") karakter konstansnak minősül az input-nál. A szöveget nem tesszük ""-be (pl. BARNA; B12; "12").
 - Programunk 6 féle bankügyletet foglal magában. Ezek egy-egy rutin-nal kezelhetők. A kiválasztás a név megadását követő IF lánc feladata. A 6 rutin mindegyikét kiszolgálja a >37Ø-38Ø szubrutin.

028 Közvetett logikai függvény szerkesztése.

55 Oldja meg 54 -et numerikus azonosítással:

ABC: OK* A GOTO 20-ból kiléphetünk a egyszeri leütésével*

HTZ: OK*

TEXAS: OK*

PRIMO: OK*

TV COM: >7 DIM J(30), M(30)*

VT-16: OK*

```

>5 PRINT "RUTIN VALASZTEK 1-6"
>10 PRINT 1"*FELKAMATOLT TOKE"
>11 PRINT 2"*DISZKONTALT TOKE"
>12 PRINT 3"*JARADEK OSSZEG"
>13 PRINT 4"*JARADEKTAG"
>14 PRINT 5"*KOLCSONOSSZEG"
>15 PRINT 6"*TORLESZTOTAG"
>20 STOP: INPUT "MELYIKET SZAMOLJA?"; Z
>60 IF Z <> 1 THEN 110
>110 IF Z <> 2 THEN 160
>160 IF Z <> 3 THEN 210
>190 PRINT "JARADEKOSSZEG="; A(N): GOTO 20
>210 IF Z <> 4 THEN 260
>240 PRINT "JARADEKTAG=" A0: GOTO 20
>260 IF Z <> 5 THEN 310
>290 PRINT "KOLCSONOSSZEG="; T(N): GOTO 20
>310 IF Z <> 6 THEN 360
>340 PRINT "TORLESZTOTAG="; T0: GOTO 20
* >RUN 0 ↓ 0 → [ ] Λ

```

△△

– Ha a rutinokat számokkal azonosítjuk (pl. 3 ↔ járadék összeg), akkor numerikus input-tal (>20) és logikai függvényekkel (>60, >110, stb.) aktiválhatók.

56 Egészítse ki az 55 programot egy törlesztő-tervet kidolgozó szubrutinnal (Z=6 esetben értelmezhető):

```

>340 PRINT "TORLESZTOTAG="; T0
>341 INPUT "HA KER TORLESZTO TERVET,
IRJA: IGEN, MASKOR: NEM"; L$
>342 IF L$ <> "IGEN" THEN 0
>343 DIM J(N), M(N)
>344 PRINT "M(I)=ADOSSAG; J(I) AZ ADOSSAG
CSOKKENESE AZ I. EVBEN"
>345 FOR I=1 TO N: M(0)=T(N)
>346 M(I)=M(I-1) * R(1) - T0
>347 J(I)=T0 - M(I-1) * (P100)
>348 PRINT I, M(I), J(I)
>349 NEXT I: GOTO 20
* >RUN 0 ↓ 0 → [ ] Λ

```

ABC: Programok összefűzését lásd P123-nál *

HTZ: Programok összefűzését lásd P123-nál *

TEXAS: Programok összefűzését lásd P123-nál *

PRIMO: A programok összefűzését lásd P123-nál *

TV COM: Programok összefűzését lásd P123-nál *

VT-16: A programok összefűzését csak akkor tudjuk megoldani, ha ASC kódban rögzítjük őket, azaz >SAVE "NEV", A*. Az összefűzést lásd P123-nál *

△△

- A kész program bármikor bővíthető új utasításokkal és rutinokkal. Ha a program címkézése és az új utasítások száma megengedi, akkor az új elemeket beszúrjuk logikai helyére, ellenkező esetben a program végéhez fűzzük. Ez utóbbit akkor is megtehetjük, ha a beszúrás is megoldható volna például:

```
>342 IF L$ <> "IGEN" THEN 2Ø ELSE 41Ø  
>41Ø DIM J(N), M(N)  
.  
.  
.  
>47Ø NEXT I : GOTO 2Ø
```

- Ha egy programot egyszeri kísérlethez módosít, háttér-tárolón csak akkor őrizze meg, ha fontos változathoz jutott.
- Ha háttér-tárolóról tölti be az alap-programot, a LOAD-ot ne előzze meg a módosító utasítások lekopogása, mert a LOAD törli azokat.
- Ha két vagy több programot szeretne összefűzni úgy, hogy háttér tárolóról egymásután betölti őket, a LOAD törlő hatása miatt kudarcot vall (a megoldást lásd a P123-nál). Töltse be az összefűzésben szóbjavó legnagyobb programot, a többit pedig kopogja le, figyelve a címkézésre.

ABC: [57] >LIST [RET] és tartósan nyomja [L]-et* Munkarend: ≥1, 3, 2
>SAVE CAS : P10* CAS a kazettás egység azonosítója* || ABC-80 ||*
≥STOP* A szalag első 3 fordulatnyi részét hagyja üresen*

HTZ: [57] Munkarend: ≥1, 3, 2 >CSAVE#-1, "P10"* -1 a kazettás egység
azonosítója* || READY ||* ≥STOP* A szalag elején 3 fordulatnyi rész marad-
jon üresen* Nyomja le az F1 gombot mielőtt csévé (pirolámpa világít)*

TEXAS: [57] >SAVE CS1* CS1 a kazettás egység azonosítója* A rögzítés munka-
rendjét a KÉPERNYŐ diktálja* Kulcsszavak: REWIND=visszacsevé, PRESS=
lenyom, RECORDING=felvesz, CHECK=ellenőriz, EXIT=kilép*

PRIMO: [57] >SAVE "P10"* || OK ||* ≥ STOP*

TV COM: [57] >SAVE#5:"P10" [RET]* A #5 a kazettás egység azonosítója* || OK ||*
Próbálja meg a >SAVE "P10" [RET] utasítást alkalmazni*

VT-16: Keménylemezes (Wdisk) egységgel (Wfloppy) dolgozunk*
Bekötés a WINCHESTER kapuba* ≥WFLOPPY ON* ≥VT-16 ON* [F1]*
Datum és idő kérés [ENT] -el elodázható* || c>||* Az UDOS operációs rend-
szer automatikusan betöltődik a Wdisk -ről a gépbe* || c>a: || [ENT] kap-
csolódás A disk-re* VT BASIC lemez A-ban* || a > vt basic || [ENT]*
|| OK || A gép BASIC parancsra vár* [57] 1,2,3 törölve* >SAVE "C : P10";
[ENT]* || OK ||*

029

Program rögzítése kazettára. **SAVE** és **VERIFY** utasítások.

57

Kérjen listát a képernyőre a gépben lévő programról, majd rögzítse kazettára:

* >LIST

```

1Ø FOR I=1 TO 7 : READ B
2Ø DATA 4, 5.2, 3.85, 2E3, -4.15, -2E4, -3E-1
3Ø PRINT B, : NEXT I : END
READY
    
```

1. ≧TEGYEN EGY ÜRES KAZETTÁT A MAGNETOFONBA.

A SZALAG LEGYEN BALOLDALT:

2. ≧KOPOGJA LE AZ ALÁBBIAKAT:



* >SAVE "P1Ø", 1,1

RET

3. ≧NYOMJA LE A MAGNETOFON **REKORD** ÉS **PLAY**

BILLENTYŰIT:

```

OK
SAVING P1Ø
READY
    
```

- ΔΔ – A programoknak nevet ad az alkotó, minden programnak mást. A **10** program neve lehet pl. "P1Ø", "ADATOLVASO" stb. A **SAVE** (megőríz) utasítás a program nevét, a kazettás egység 1 azonosítóját (első 1-es) és programhoz „vége” jelet generáló 1-est (második 1-es) tartalmazza.

ABC: Ha a név 8 karakternél hosszabb, hibajeleket kapunk*. A technikai hibákra a gép nem figyelmeztet*. Teljes ismétléssel korrigálhatunk*
 58 ellenőrzés nem végezhető a rögzítés minőségéről*

HTZ: 58 Munkarend: ≥1, 2, 5, 3 >CLOAD? "P10" NL* || READY ||* Az ellenőrzési folyamatot a teljes képernyő jobb felső sarkában két csillag jelzi, az egyik villog*. A gép csak akkor vezérli a magnót, ha F1 nincs lenyomva.

TEXAS: 58 Az ellenőrzést a gép kezdeményezi. A képernyőn a rögzítés végén CHECK TAPE (Y OR N)? jelenik meg. Válaszunk Y (yes), vagy N (no). Mindkét esetben a képernyő diktálja további teendőinket.

PRIMO: 58 3>> >TEST "P10" RET* A gép válasza: || 02 00 FOUND: P10 ||, ahol az első adat a rekordok, a második a hibák számát jelöli a rögzített P10-ben*. Ha a hibaszám nagyobb 0-nál a rögzítést megismételjük*.

TV COM: 58 > VERIFY#5: "P10" RET* || SEARCHING || || READING || || P10 || || OK ||* A #5: elhagyható*

VT-16: 58 A Wfloppy automatikusan ellenőrzi a program rögzítését*. VERIFY beadására a gép hibajellel válaszol*. NEM*
A képernyő alján F1,...,F10 gombok utasítás értéke áll*. F9 >OFF ENT eltávolítja a képernyőről*. Próbálja ki F9 F1 ENT leütéseket*.

A név hossza ne haladja meg a 16 karaktert.

A nevet időzítőjelbe tesszük.

– A SAVE "NÉV", 1,1 utasítás hatására a szalagon megjelenik a név, program és végjel. A folyamatról a SAVING, befejezéséről a READY tájékoztat.

– A rögzítés technikai akadályait a gép közli. Ha pl. nem nyomtuk meg a REKORD és PLAY billentyűket, akkor figyelmeztet: **|| PRESS RECORD AND PLAY ON TAPE ||** erre a mulasztást pótoljuk és a rögzítés végbemegy.

58 Ellenőrizze az **57** rögzítés pontosságát:

1. **»NYOMJA LE A REWIND BILLENTYŰT A MAGNÓN (VISSZACSÉVÉLI A SZALAGOT): STOP.**
2. **»NULLÁZZA A MAGNÓ SZÁMLÁLÓJÁT:**
3. **»KOPOGJA LE AZ ALÁBBIKAT:**



>VERIFY "P10", 1

RET

4. **»FIGYELJE A KÉPERNYŐT:**

|| PRESS PLAY ON TAPE ||

5. **»NYOMJA LE A PLAY BILLENTYŰT:**

```
|| SEARCHING FOR P10 ||  
|| FOUND P10 ||  
|| VERIFYING ||  
|| OK ||  
|| READY ||
```

△△

– A VERIFY (átvizsgál) utasítás a gépben és a

ABC: [59] Munkarend: $\geq 1, 2, 4, 3$ >LOAD CAS:P1Ø [RET] || ABC 8Ø ||* Hibás névadás esetén végigkeresi a szalagot a gép* Ha n_k ismeretlen, a szalag elejéről indulhatunk*

HTZ: [59] Munkarend: $\geq 1, 2, 4, 3$ >CLOAD#-1, "P1Ø" [NL]*A töltési folyamat is két csillag jelzi a képernyőn. A villogást || READY || követi* Ha n_k ismeretlen, a szalag elejéről indulhatunk, a gép megkeresi a programot az adott név alapján*

TEXAS: [59] >OLD CS1 [ENT]* A program betöltéséhez kapcsolódó munkarendet a képernyő diktálja* A program neve alapján a gép nem találja meg a programot, ezért alapvetően fontos n_k pontos beállítása*

PRIMO: [59] >LOAD "P1Ø" [RET]* || Ø2 ØØ FOUND:P1Ø* Hibás névadás esetén a teljes szalagot átvizsgálja a gép és a talált program címeket SKIP:"CIM" formában közli* Ha n_k ismeretlen, a szalag elejéről indulhatunk*

TV COM: [59] >LOAD#5: "P1Ø" [RET]* || SEARCHING || || READING || || P1Ø || || OK ||* Hibás névadás esetén a teljes szalagot átvizsgálja a gép és a talált programok címeket FOUND: "CIM" formában közli* A #5: elhagyható*

VT-16: [59] $\geq 1, 2, 3, 4$ törölve* LOAD "C:P1Ø" [ENT]*A Wdisk beépített mágneslemez ~ 5 MEGABYTE ($2 \wedge 24$ byte = 1 MEGABYTE) kapacitással* Igen gyors* Nagyméretű programok és adathalmazok rögzítésére használják* A Wdisk -ről >KILL" C: P1Ø. BAS" [ENT] parancs törli* P1Ø-et* BAS a gép által generált típus jel, a program tartozéka*

szalagon lévő program egybevetését kezdeményezi. A képernyőn megjelenő OK a jó, a VERIFY ERROR hibás rögzítést jelez. Ezesetben az 57 és 58 feladatot megismételjük. Ha a rendszer összekapcsolása jó, a dugaszok kontaktusa megfelelő, a környezet hőmérséklete nem magas (<24 °C), akkor a sikertelenség okát a technika meghibásodásában kereshetjük.

- Az OK-al zárult VERIFY programunk megőrzését és ismételt alkalmazási lehetőségét nyugtázza.

»IRJA FEL A KAZETTA REGISZTERÉRE A PROGRAM HELYÉT (n_k-n_v ; A SZÁMLÁLÓ KEZDŐ ÉS VÉGÁLLÁSA) ÉS CÍMÉT:
 »CÍMEZZE MEG EGYEZŐEN A KAZETTÁT ÉS A REGISZTERT (PL. P001):

△△ – A SAVE és VERIFY utasításokban a név és az eszközzazonosítók megegyeznek. Eltérést csupán a végjel (l) értelemszerű kezelése jelez.

030 Program betöltése kazettáról. LOAD utasítás.

59 Töltse be a P001-ről a P10 programot és futtassa:

1. »TEGYE A P001-ET A MAGNÓBA:
2. »CSÉVÉLJE A SZALAGOT n_k -RA: STOP.
3. »KOPOGJA LE AZ ALÁBBIKAT:

0
 1
 2
 3
 4
 5
 6
 7
 8
 9
 *
 *

>LOAD "P10", 1, C

RET

|| PRESS PLAY ON TAPE ||

4. » LENYOMJA A PLAY BILLENTYŰT.
-

ABC: [60] >SAVE CAS: ADATOLV [RET]*>VERIFY nem működik*

HTZ: [60] A program nevének csak első karakterét azonosítja a gép. Ezért elegendő, ha >CSAVE#-1, "A", illetve >CLOAD? "A" utasításokkal dolgozunk. Program keresésnél az "A" megjelenik az egyik csillag helyén*

TEXAS: [60] >SAVE CS1 [ENT]* Ha a program címét megadjuk, a gép szintaktikai hibát jelez* A program kezdetét a fordulat-számlálón n_k ismeretében a kezelő állítja be*

PRIMO: [60] >SAVE "ADATOLVASO"* A program címe legfeljebb 16 karakterből állhat. Túllépés esetén FC ERROR lép be*

TV COM: [60] >SAVE#5: "ADATOLVASO"* A gép csak 16 karakterig jegyzi fel a címet*

VT-16: [60] >SAVE "C: ADATOLVASO" [ENT]* A címből 8 karaktert jegyez fel a Wfloppy **

```

OK
SEARCHING FOR P1Ø
FOUND P1Ø
LOADING
READY

```

△△

- A **LOAD** utasításban a cím ("P1Ø"), a kazettás egység azonosítója (1) és a magnetofon motor indítója (C) szerepel. A gép üzenetei: **SEARCHING** (keresem), **FOUND** (megtaláltam), **LOADING** (töltöm), és **READY** (kész). A gép megkeresi a szalagon az adott című programot és a gép memóriájába tölti. Hibás címadás esetén (vagy ha az író-olvasó fej hibás) **FILE NOT FOUND** jelzi, hogy nem találja.
- A **LIST** utasítás megjeleníti a programot. Enélkül is futóképes a program, de jobb meggyőződni a **RUN** előtt, hogy a kívánt programot töltöttük-e be a kazettáról.

```

* >LIST
* >RUN

```

60 Módosítsa a 10 programot 2Ø adat beolvasására. Adja meg az adatokat, majd rögzítse a programot "ADATOLVASO" címmel az előző program után 1 fordulatra:

```

>1Ø FOR I=1 TO 2Ø : READ B
>21 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9, 1Ø, 11, 12, 13

```

↺

```

>SAVE"ADATOLVASO", 1, 1
>VERIFY"ADATOLVASO", 1

```

```

RET
RET

```

ABC: A keresést a **RESET** gomb (gép hátoldala) vagy a **STOP** (kazettás egység) állítja le. Az első megoldás gyorsabb*
*

HTZ: >CLOAD# -1, "ADATOLVASO" * A számláló túlhaladást jelez, ha az F1 gomb lenyomva maradt, ha a keresett program sérült, hiányzik stb. (**RESET**, **STOP**)* || **READY** || jelzi a sikeres betöltést (, **STOP**)*

TEXAS: A gép **NEM TUDJA** a feladatot megoldani. A szalagot az „ADATOLVASO” c. program rögzítési pontjára kell csévélni*
*

PRIMO: A keresést a kazettás egység **STOP** billentyűjével szakítjuk meg*
*

TV COM: **OK*** A keresést szakítja meg*
*

VT-16: **NEM***
A Wfloppy feljegyzést készít a tárolt programokról (és adat file -okról).
Betöltéskor a feljegyzésből állapítja meg a file -ok helyét a lemezen*
A Wfloppy gyors lemezegység (fix disk -kel látták el)*
*

△△

- **VIGYÁZAT!** A szalag felvétel törléssel kezdődik. A kazettás egység nem védi a korábban rögzített programokat. A szalag beállításáról, a számlálót figyelve, magunknak kell gondoskodni. Ezért regisztráljuk a felvitt programokról a számláló kezdő és végállását. Minden program n_k kezdőállása megegyezhet az előző program n_v végállásával. A beállítás a SAVE RET-je előtt eszközzendő.
- A VERIFY csak akkor működik ha a RET leütése előtt a szalagot visszacsévéltük a program kezdőpontja elé.

61

Csévélje teljesen vissza a szalagot és adja a LOAD "ADATOLVASO" utasítást:



>LOAD"ADATOLVASO", 1, C

RET

△△

- A szalag addig megy, amíg a gép a keresett címet meg nem találja. Addig is közli a talált címeket. A READY és a szalag leállása sikeres műveletet jelez. Ha a számláló túlhaladást jelez, akkor a RUN/STOP gombbal megállítjuk a kereső folyamatot, mert sikeres betöltésre nem számíthatunk.
- Számítsunk rá, hogy a kazettás egység munka fejei (törlő, író-olvasó) 15–20 órai használat alatt felmágneseződnek. A demagnetizálást a fejek alkoholos áttörlésével végezzük. Pamut kendőt használjunk és tiszta alkoholt.

ABC: [62] >3Ø PREPARE "NEVSOR" AS FILE 62: FOR I=1 TO N : PRINT#62, A X(I): NEXT I * /A >2Ø OPEN "NEVSOR" AS FILE 15 : FOR I=1 TO N: INPUTLINE#15, B X(I) : PRINT B X(I) : NEXT I * Ha N-et pontatlanul adjuk meg, betöltés OK, de túlméretezéskor hibajel kerül a képernyőre és a magnetofon nem áll le *

HTZ: [62] >3Ø PRINT# -1, "NEVSOR" : FOR I=1 TO N : PRINT# -1, A\$(I) : NEXT * >4Ø PRINT# -1, "VEGE" : END * /A >2Ø INPUT# -1, NEVS\$: IF NEVS\$ <> "NEVSOR" THEN 2Ø * >21 PRINT NEVS\$ * >22 FOR I=1 TO N: INPUT# -1, B\$(I) : PRINT B\$(I) : NEXT * >3Ø INPUT# -1, VEGES\$: PRINT VEGES\$: END * OPEN-CLOSE utasítások nincsenek. A file számot nem kap *

TEXAS: [62] >3Ø OPEN# 62 : "CS1", FIXED, OUTPUT :: : PRINT#62 : A\$(I) :: * >4Ø CLOSE#62 * /A >2Ø OPEN#15 : "CS1", FIXED, INPUT:: : INPUT#15: B\$(I):: * 3ØCLOSE#15 * A FIXED≤64 karakteres rekord -ok rögzítését jelzi * Az OUTPUT rögzítésre, az INPUT betöltésre nyit *

PRIMO: [62] >5 CLEAR1ØØØ * Tágitjuk a gép string tárolóját 1Ø5Ø byte -ra (5ØB adott) * >3Ø CREATE "NEVSOR" : FOR I=1 TO N : PRINT#A\$(I) : NEXT * >4Ø CLOSE : END * /A >2Ø OPEN "NEVSOR" : FOR I=1 TO N : INPUT# B\$(I) : * >3Ø CLOSE: * A betöltés érzékeny a rekord -szám túllépésére (INPUT N) * >CLEAR 1ØØØ kikapcsolásig él *

TV COM: [62] >1Ø PRINT "A NEVEK SZAMA" : INPUT N : DIM A\$(N) *3Ø * >3Ø OPEN OUTPUT "NEVSOR" : FOR I=1 TO N : PRINT#5 : A\$(I) : NEXT * >4Ø CLOSE OUTPUT: * /A >1Ø : DIM B\$(N)*3Ø * >2Ø OPEN INPUT "NEVSOR" : INPUT#5 : B\$(I) : * >3Ø CLOSE INPUT: *

VT-16: [62] >3Ø OPEN "O", #62, "C : NEVSOR" : * >4Ø CLOSE#62 : END * "O" OUTPUT-ra megnyitott file a C Wdisk -en * Az adatok rögzítése soros * /A >2Ø OPEN "I", #15, "C : NEVSOR" : * "I" INPUT-ra megnyitott file a C Wdisk -en * A CLOSE#15 helyett CLOSE is állhat * A visszaolvasás az első N adatra szól és nem kezdődhet pl. a második vagy későbbivel *

Folytassa a munkát a 11 feladatnál. A fontosabb programokat rögzítse a P001-re.

Erre a programok alatt megjelenő $\circ \downarrow \circ$ jel figyelmezteti!!!

031

Adat, rekord, file. OPEN-PRINT#-CLOSE utasítások.

62

Rögzítsen névsort kazettára, majd tölts be a gépbe és írassa képernyőre:

```

*
>NEW
>1Ø INPUT "A NEVEK SZAMA"; N : DIM A$(N)
>2Ø FOR I=1 TO N : INPUT "A KOVETKEZØ NEV
?"; A$(I) : NEXT
>3Ø OPEN#62, 1, 2, "NEVSOR" : FOR I=1 TO N :
PRINT#62, A$(I) : NEXT
>4Ø CLOSE#62 : END

```

» TEGYEN ÜRES KAZETTÁT A GÉPBE:

```

*
>RUN  $\circ \downarrow \circ$  →  $\boxed{\cdot}$   $\wedge$ 
» REGISZTERBE:  $n_k - n_v^1$ , "NÉVSOR"
» CIMEZZE MEG A KAZETTÁT (PL. F1Ø1)

```

```

/A *
>NEW
>1Ø INPUT "A NEVEK SZAMA"; N : DIM B$(N)
>2Ø OPEN#15, 1, Ø, "NEVSOR" : FOR I=1 TO N :
INPUT#15, B$(I) : PRINT B$(I) : NEXT
>3Ø CLOSE#15 : END

```

» CSÉVÉLJE n_k -RA A KAZETTÁT

```

*
>RUN  $\circ \downarrow \circ$  →  $\boxed{\cdot}$   $\wedge$ 

```

△△

– A program a rögzítendő adatok input-jával kezdődik. A bevitt adatok file-t képeznek. A file számmal, pl. 62 és névvel, pl. "NEVSOR" kerül azonosításra. A szám technikai segédeszköz, tehát helyettesíthető, de a név az tartozék.

ABC: [63] >2Ø PREPARE "FORGALOM" AS FILE 63: * >3Ø PRINT#63, F(I,J)
: NEXT J : NEXT I : CLOSE#63 : END * A >3Ø F(I, J), ";"-vel is működik *

HTZ: [63] >2Ø PRINT# -1, "FORGALOM" : FOR I=1 TO 3 : FOR J=1 TO 4 :
PRINT# -1, F(I, J) : NEXT J, I * >3Ø, PRINT# -1, "VEGE" : END * "VEGE"
helyett ØØ is alkalmazható *

TEXAS: [63] >1Ø::NEXT J::NEXT I * >2Ø OPEN#63: "CS1", FIXED1ØØ, OUTPUT*
>3Ø :: CLOSE#63 :: * Ha FIXED 1ØØ-at írunk, a gép 128-at ért, mert három
(64, 128, 192) rekord - hosszot értelmez és fölfelé kerekít *

PRIMO: [63] >2Ø CREATE "FORGALOM" : * >3Ø PRINT#F(I, J) : NEXT J, I :
CLOSE : END *

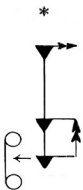
TV COM: [63] >5 DIM F(3,4) * >2Ø OPEN OUTPUT "FORGALOM": * >3Ø PRINT#5
: F(I, J) : NEXT J, I : CLOSE OUTPUT : END * A rekord -ok elválasztásá-
ról a gép gondoskodik *

VT-16: [63] >2Ø OPEN "O", #63, "C : FORGALOM" : * >3Ø PRINT#63, F(I, J)
: NEXT J, I : CLOSE#63 : * A rekord -ok elválasztásáról a Wfloppy gondos-
kodik (". " hibás betöltésre vezet) *

- Az adatfile rögzítése megnyitással (OPEN62,1,2, "NEVSOR") kezdődik. Ebben az 1 a kazettára, a 2 pedig a rögzítésre utal. A "név" azonosítja a file-t. A PRINT#62, A\$(I) oldja meg az adatok listászerinti rögzítését. Ezt követően a file -t lezárjuk (CLOSE62). A file jelenlegi száma (62) csak arra való, hogy a három utasítást összekapcsolja vonatkozás szempontjából.
- Adatfile kazettáról való betöltését az OPEN-INPUT# -CLOSE utasításokkal oldjuk meg. Látható, hogy a file szám (15) elfelejthető. Betöltés esetén 2-es helyett Ø szerepel a nyitó utasításban. A # jel perifériáról való adat input-ot jelöl (közvetlenül csatlakozik az INPUT szóhoz).
- A file adatai (rekord-ok) ciklus utasítással, egyenként kerülnek betöltésre.

63

Egy kereskedelmi vállalat 4 boltja havi bontásban adja meg negyedévi forgalmát (3x4-es tömb). Rögzítse kazettára a tömböt, majd töltsse be a gépbe:



```

>NEW
>1Ø FOR I=1 TO 3:FOR J=1 TO 4 : INPUT  "FORGALOM?";
      F(I, J) : NEXT J, I
>2Ø OPEN 63, 1, 2, "FORGALOM" : FOR I=1 TO 3 : FOR J=1 TO 4
>3Ø PRINT#63, F(I, J),"," : NEXT J, I : CLO SE63 : END

```

≧F1Ø1 A MAGNÓBAN, SZÁMLÁLÓ n↓-re ÁLLITVA:

```

>RUN  ⓪↓⓪ →  Λ

```

ABC: 63/A >1Ø OPEN "FORGALOM" AS FILE 5 : FOR K=1 TO 3 : FOR L=1
TO 4 : INPUT#5, P(K, L)* INPUTLINE# string -ek, INPUT# számok betölté-
sénél használható*

64 >1Ø J=1 TO 4* >2Ø FOR I=1 TO 3 : PRINT "FORG" : INPUT F(I, J) :
NEXT I : NEXT J*

HTZ: 63/A >1Ø INPUT# -1, NEV\$: IF NEV\$ <> "FORGALOM" THEN 1Ø
* >2Ø PRINT NEV\$: FOR K=1 TO 3 : FOR L=1 TO 4 : INPUT# -1, P(K, L)*
>3Ø PRINT P(K, L), : NEXT L, K : INPUT# -1, VEGE\$: PRINT VEGE\$:
END* ØØ-nál v *

64 >25 PRINT -1, F(1, J) F(2, J), F(3, J)*

TEXAS: 63/A >1Ø OPEN#5 : "CS1", FIXED 1ØØ, INPUT :.*

64 >25 PRINT#64 : P(1); ",,"; P(2); ",,"; (P3)*

PRIMO: 63/A >1Ø OPEN "FORGALOM" : FOR K=1 TO 3 : FOR L=1 TO 4 :
INPUT# P(K, L) : NEXT L, K : PRINT* >2Ø FOR I=1 TO 3 : FOR J = 1 TO 4 :
PRINT P(I, J) ; : NEXT J : PRINT : NEXT I : CLOSE : END* 64 >25
PRINT#P(1), P(2), P(3)* OK* Az adat rögzítés 256 byte -os pufferból törté-
nik. Célszerű az adatokat együtt gépre vinni és utána rögzíteni. Így a szalag ki-
használása jobb*

TV COM: 63/A >5 DIM P(3,4)* >1Ø OPEN INPUT "FORGALOM" : INPUT#5 :
P(K, L)* >2Ø : CLOSE INPUT : END*

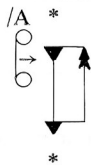
64 >5 DIM F(3,4), P(3)* >25 PRINT#5 : P(1); ",,"; P(2); ",,"; P(3)*

VT-16: 63/A >1Ø OPEN "I", #5, "C : FORGALOM" :.* >2Ø : NEXT L : PRINT :
NEXT K : CLOSE#5 :.*

64 >25 PRINT#64, (P1), P(2), P(3)*

»KOPOGJA LE A TÖMB ADATAIT SORFOLYTONOSAN:

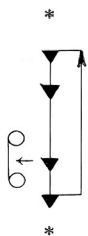
»REGISZTERRE $n_k^2 - n_v^2$ ÉS "FOGALOM" :



```
>NEW
>1Ø OPEN 5, 1, Ø, "FOGALOM" : FOR K=1 TO 3 : FOR L=1
    TO 4 : INPUT#5, P(K, L)
>2Ø PRINT P(K, L), : NEXT L, K : CLOSE5 : END
>RUN  O↓O  → [ ]  Λ
```

△△ — A file nem kötődik szorosan numerikus azonosítójához és a mozgatasakor használt változókhöz. Így a rögzítés és betöltés azonosítói és ciklus változói (pl. $I \leftrightarrow K$; $J \leftrightarrow L$; $F \leftrightarrow P$) egymástól függetlenül jelölhetők ki.

64 Képezzen egy-egy bolt forgalmi adataiból rekord-ot és így rögzítse kazettára, majd töltsse be a gépbe.



```
>NEW
>1Ø OPEN64, 1, 2, "FOGALOM" : FOR J=1 TO 4
>2Ø FOR I = 1 TO 3 : INPUT. "FOGALOM?" ; F(I, J)
    : P(I)=F(I, J) : NEXT I
>25 PRINT#64, P(1) ; CHR$(13) P(2); CH R$(13) P(3); CHR$(13)
>3Ø NEXT J : CLOSE 64 : END
>RUN  O↓O  → [ ]  Λ
```

»AZ ADATOKAT OSZLOPFOLYTONOSAN KOPOGJA LE:

ABC: >25 PREPARE "FORG" AS FILE 64 : FOR J=1 TO 4 : PRINT#64, F(1, J)
; " , " ; F(2, J) ; " , " ; F(3, J) * >3Ø NEXT J : CLOSE64 : END*
 64/A OK* 64/B NEM működik* A 64 átalakítását a hézagmentes
rögzítés igénye indokolja. A hézagok betöltésnél STOP-ként működnek*
 65 lásd 64*

HTZ: 64/A OK*
 64/B A 64 így nem olvasható vissza* A rekord -ok első tagjait tölti be
a gép*
 65 >25 PRINT# -1, S(1), S(2), S(3), S(4)*

TEXAS: 64/B A 64 így NEM olvasható vissza*
 65 >25 PRINT#65 : S(1); " , " ; S(2); " , " ; S(3); " , " S(4)*

PRIMO: 64/A OK* 64/B NEM*
 65 OK*

TV COM: 64/A OK*
 64/B Így nem olvasható vissza*
 65 >25 PRINT#5 : S(1); " , " ; S(2); " , " ; S(3); " , " ; S(4)*

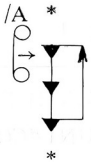
VT-16: 64/A OK*
 64/B >3Ø NEXT I : PRINT : NEXT J : * OK*
 65 OK*

A P63, P64 és P65 rögzítési módok a Wfloppy számára **egyenértékűek**.
Bármelyik betölthető a P63A, P64A vagy P65A programmal*

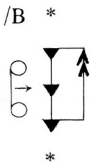
△△

– Az adat INPUT most oszloponként történik.

Egy-egy oszlop adatait a P(I) tömbváltozó segítségével rekord-ba szervezzük (a rekord tehát adatokból álló szervezeti egység pl. [P(1), P(2), P(3)]). A rögzítés (és betöltés) tárgyaként így adat helyett a rekord funkcionál. A jelen rekord csak akkor válik betölthetővé, ha a rögzítésnél CHR\$(13)-at iktatunk a rekord adatai közé és a rekord végére. A betöltés ugyanis rekord-onként, az azonosítás viszont adatonként történik (pufferes betöltés).

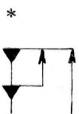


```
>NEW
>1Ø OPEN 3Ø, 1, Ø, "FORGALOM" : FOR J=1 TO 4
>2Ø INPUT#3Ø, F1, F2, F3 : PRINT F1, F2, F3
>3Ø NEXT J : CLOSE3Ø : END
>RUN  Q↓O → [ ] Λ
```



```
>NEW
>1Ø OPEN3Ø, 1, Ø, "FORGALOM" : FOR J=1 TO 4: FOR I=1 TO 3
>2Ø INPUT#3Ø, F(I, J) : PRINT F(I, J),
>3Ø NEXT I, J : CLOSE3Ø : END
>RUN  Q↓O → [ ] Λ
```

65 Módosítsa 64 -et úgy, hogy az egyes hónapok bolti adatai alkossanak rekord-okat:



```
>NEW
>1Ø OPEN65, 1, 2, "FORGALOM" : FOR I=1 TO 3 : FOR J=1 TO 4
>2Ø INPUT "FORGALOM?" ; F(I, J) : S(J)=F(I, J) : NEXT J
```

ABC: 66 >4Ø FOR I=1 TO N : INPUT C(I), M Ø (I), V(I), E(I), D Ø (I) : NEXT I
* >5Ø PREPARE "KESZLET" AS FILE 66 : FOR I=1 TO N : PRINT#66, C(I),
", " ; M Ø (I) ; ", " ; V(I) ; ", " ; E(I) ; ", " ; D Ø (I) *
67 >4Ø FOR I=1 TO N : INPUT K Ø (I) : NEXT I *

HTZ: 66 >5Ø PRINT# -1, C, M\$, V, E, D\$ *
67 >5Ø PRINT# -1, K\$ (I) : NEXT *

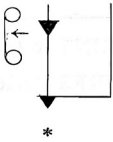
A program futásakor OS hibajel léphet fel (a gép 5Ø byte -os string tárolója betelt). Bővebb tárolót >RUN előtt >CLEAR N NL foglal *

TEXAS: 66 OK, mint 65* A program nagyon lassú *
67 >5Ø PRINT#67 : K\$ (I) : : NEXT I : : *

PRIMO: 66 >5 CLEAR1ØØØ* >4Ø FOR I=1 TO N : INPUT C(I), M\$(I), V(I), E(I),
D\$(I) : NEXT * >5Ø CREATE "KESZLET" : FOR I=1 TO N : PRINT#C(I),
M\$(I), V(I), E(I), D\$(I)*
67 >4Ø FOR I=1 TO N : INPUT K\$(I) : NEXT >5Ø : PRINT#K\$(I)*
Ellenörzés betöltéssel *

TV COM 66 OK, mint 65* Próbálja meg az adatbehívást és a rögzítést szétválasztani,
így sok tárolóhelyet takarít meg *

VT-16: 66 >5Ø PRINT#66, C, M\$, V, E, D\$ *
66/A Betöltéshez P66 módosítható: >4Ø OPEN "1", #66, "C : KESZLET" :
FOR I=1 TO N* >5Ø INPUT#66, A\$: PRINT A\$ * Vegyes (numerikus és
string adatok) rekord -ok tételes INPUT-ja hibás azonosításhoz vezethet *

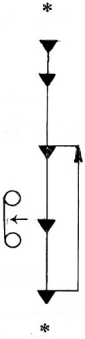


```

>25 PRINT#65, S(1); CHR$(13) S(2); CHR$(13) S(3); CHR$(
      (13) S(4); CHR$(13)
>30 NEXT I : CLOSE65 : END
>RUN   O↓O   → [ ] Δ

```

66 Irjon alkalmazási programot raktárkészlet (cikkszám, megnevezés, mennyiség, egységár, mértékegység) kazettára rögzítéséhez:



```

*
>NEW
>10 INPUT "A TETEEK SZAMA" ; N
>30 PRINT "CIKKSZ, MEGNEV, MENNYI, EGYSEGAR, MERTEKE"
      : PRINT
>40 OPEN66, 1, 2, "KESZLET" : FOR I=1 TO N : INPUT C, M$,
      V, E, D$
>50 PRINT#66, C;CHR$(13) M$;CHR$(13)V;CHR$(13)E;
      CHR$(13) D$; CHR$(13)
>60 NEXT : CLOSE66 : END
>RUN   O↓O   → [ ] Δ

```

△△ – Egy raktári tétel (pí. anyag, szerszám, gép stb.) adatai rekord-ot képeznek. Így az INPUT és PRINT# nem adatokra, hanem rekord-okra vonatkoznak (egy kérdőjel itt 5 adatot hív). Az adatok sorrendjét felirattal adjuk meg (>30), hogy az adatbevitelt támogassuk. Az INPUT adatokat vesszővel elválasztjuk!

67 Módosítsa a 66 programot arra az esetre, amikor a rekord-ot karakter konstansként definiáljuk:

```

>20 DIM K$(N)
>35 PRINT " *.....*.....*.....*.....*.....*"

```

ABC: >5Ø PREPARE "KESZLET" AS FILE 67 : FOR I=1 TO N : PRINT#67,
K ⚡ (I)* >6Ø NEXT I : CLOSE67 : END* A STRING URES KARAKTE-
REIT PONTOZZUK KI*
[68] >1Ø OPEN "NEV" AS FILE 1* >2Ø INPUT#1, A ⚡ : PRINTA\$*>3Ø
GOTO 2Ø* >4Ø CLOSE1 : END* A futás ERR34-el áll le a file betöltése után*

HTZ: [67] Pl. CLEAR 1ØØØ pontosan 1ØØØ byte tárolóhelyet biztosít a gép memó-
riájából*
[68] >2Ø INPUT# -1, A\$: PRINT A\$: IF A\$ = "VEGE" THEN END* >3Ø
GOTO2Ø* Ha a file "VEGE" végjelet kapott rögzítéskor és a file rekordjai
string -ek*

TEXAS: [68] NEM oldható meg. Hiányzik a GET# utasítás és kazettás egységre nem
alkalmazható az ST=64-et helyettesítő EOF*

PRIMO: [68] >1Ø OPEN "NEV" * >2Ø INPUT#A\$: PRINTA\$;* >3Ø GOTO2Ø
* >4Ø CLOSE : END* A futás a file végén akad meg. RESET benyomásával
állítjuk le*

TV COM: [67] >2Ø DIM K\$(N)*4Ø*>35 A karakterek száma adja meg a string dimen-
zióját >2Ø alatt* >5Ø PRINT#5 : K\$(I); ", " : NEXT : CLOSE OUTPUT : *
[68] >2Ø GET#5 : A\$: PRINT A\$; : IF PEEK(2926) <> Ø THEN CLOSE
INPUT : END* >3Ø GOTO 2Ø* A gép 2926-os rekesze nullától eltérő értéket
vesz fel, ha a file végére ér a betöltésnél*

VT-16: [67] OK*
[68] >2Ø INPUT#68, A\$: PRINT A\$* A betöltés nem karakter, hanem
rekord-onként történik* >3Ø IF NOT EOF(68) THEN 2Ø 'EOF(68) FILE
VEGE*>4Ø CLOSE#68 : END* >5Ø [ENT]*

ABC: [69] Nem oldható meg, mert a **file neveket a gép nem írja ki**, csak összehasonlítja a keresett névvel és megy tovább, ha azonosságot nem talál*

HTZ: Ha számokból álló file-okról van szó, akkor **A\$** helyett **A**, **"VEGE"** helyett **Ø** végjel alkalmazandó*
[69] OK* || A* ||*

TEXAS: [69] **NEM** oldható meg, mert a file neveket a gép nem azonosítja*

PRIMO: [69] **>LOAD "EBEK"** [RET] a programok címeit, a **>1Ø OPEN "EBEK"**:
INPUT#A\$: CLOSE* **>RUN** pedig az adat-file-ok neveit jeleníti meg **SKIP :**
NEV formában* **"EBEK"** helyett más, címként nem használt szó is állhat*

TV COM: [69] OK*

VT-16: [69] A megoldás közvetlen, eltérően a kazettás egység közvetett és lassú eljárásától: **>SYSTEM** [ENT]* || **A > C :** || [ENT]* || **C > DIR** || [ENT]*
Próbálja ki a **>DIR/W** [ENT] parancsot* A megoldás **UDOS**-ban zajlik*
Visszatérés a BASIC-ba: || **C > A :** || [ENT]* || **A > VT BASIC** || [ENT]*

△△

- A GET# utasítás egyesével tölti be a karaktereket. A karakterek (betű, szám, szóköz stb.) egy string (A\$) változóhoz kötődnek. A betöltést egy ciklus szabályozza, de a ciklus számát előre nem ismerjük. Ehhez nyújt támogatást a gép úgynevezett status (ST) paramétere. Ez a paraméter 0-tól 64-ig vesz fel értékeket. Ezek közül az ST=64 a file végét, az ST=0 a zavartalan betöltést és 0<ST<64 valamilyen szalaghibát jelez.
- A file karakterenként kerül a képernyőre. Itt kirajzolódik a rekord-ok összetétele. Néhány rekord megjelenése után ki tudjuk jelölni az INPUT# paramétereiket és a betöltés az ismert módon elvégezhető (pl. 64/A, B).



Vizsgálja meg a 68 program alkalmazásával a "NEVSOR", a "FORGALOM" és a "KESZLET" file-ok szerkezetét.

69

Állapítsa meg egy kazettán található programok, illetve file-ok címeit:

»SZALAG BALRA CSÉVÉLVÉ:

*

>LOAD "EBEK", 1, C

RET

△△

- Ha "EBEK" című file nincs a kazettán, akkor a gép végigpergeti a szalagot és minden program, illetve adat file nevét képernyőre írja pl.:

```
|| FOUND P10 ||  
|| FOUND ATLAGOK ||
```

- Ha a szalag megáll, újabb LOAD utasítással lökjük tovább.

ABC: 70 Lemezegység nincsen rendszeresítve. Egyes feladatok a két **kazettás egység** felhasználásával megoldhatók*
 71 NEM*

HTZ: 70 Lemezegység nincsen rendszeresítve. Egyes feladatokat a **kazettás egység** felhasználásával meg tud oldani*
 71 NEM*

TEXAS: 71 **»HELYEZZEN BE EGY KÁRTYÁT** (címké jobboldalon, lent)* **»HELYEZZE BE A SZÁMÍTÓGÉPBE A "DISK MANAGER COMMAND MODULE" BLOKKOT*** 1 és 3 leütése után a képernyő vezeti az eljárást* **YOUR CHOICE?** kérdésre 2 majd 4 a válasz, **MASTER DISK (1,3)?-ra 1***
TRACKS (Y/N)? kérdésre Y **40**, N **35 track felépítését kezdeményezi.**

PRIMO: 70 Töltse be kazettáról a **CDOS** periféria illesztő programot, ami 19 rekordból áll (a kazetta vásárolható)* A gép **"COMMODORE SERIAL BUS HANDLER 1.0 (C) BME-KSZK, 1985"** kiírásával nyugtázza a program működési készségét* 71 **>CMDNEW "VARGA, 01"** RET* **CLOSE15** lekopogása **szükségtelen*** A CDOS a gép kikapcsolásakor törlődik* Több floppy bekapcsolo-

TV COM: A TV COM iker-floppy fejlesztés alatt áll* Kapcsolja a floppy dugaszát (HANDLER) a gép fedelén nyitható, balról nézve az utolsó kapuba* 70 OK*
 71 VT-16-on oldható meg, ha UDOS (16. B) helyett **UPM V.2.1** operációs rendszer (8 B) kerül beolvasásra* A formázott lemezt az A, vagy B floppy-ba helyezhetjük (címké jobbra, lent)* A jelen fejlettségi fokon a BASIC-ből nem

VT-16: 70 A géppel együtt bekapcsol* A baloldali egység azonosítója az A, a jobboldalié a B* 71 **»GÉP BEKAPCSOLÁSA*** **UDOS BEOLVASÁSA*** A lemez segédprogramokat is tárol. Listájuk : **|| A:\>DIR ||** ENT, vagy **>DIR\W**
 ENT* Töltse be a **FORMAT** nevű programot: **|| A:\>FORMAT ||** ENT*

3. A LEMEZEGYSÉG KEZELÉSE, PROGRAMOZÁSA ÉS FELHASZNÁLÁSA

70 Kapcsolja be rendre: 1. a televíziót, 2. a lemezeget, 3. a számítógépet.

- △△ – Ez a kapcsolási rend a készüléket védi az elektronikus sérülésektől. Szigorúan tartsa be. A kikapcsolás fordított (3., 2., 1.) rendet követ.
- A lemezeget mágneses kártyákkal (disk) dolgozik. A kártya érzékeny hajlításra, hőre, hidegre és mágneses hatásokra. A kazettától eltérően, originál állapotban nem alkalmas adatok és programok tárolására, előtte formalizálni kell.

033 A DISK formalizálása. TRACK, SECTOR és BLOCK.

71 Formázzon meg egy mágneses lemezt:

»HELYEZZEN BE EGY LEMEZT (CIMKE FENT, JOBBKÉZRE)
A FLOPPY-BA ÉS ZÁRJA LE A FEDELÉT:
>OPEN15,8,15, "NØ : VARGA, Ø1" RET

 ← *

- △△ – A lemezen végzett művelet idején piros lámpa ég a floppy (meghajtó) elülső homlokán. Ha kialudt, akkor:

*

>CLOSE 15

RET

|| READY ||

ABC: —

HTZ: —

TEXAS: Így **360**, illetve **315** sector keletkezik, egyenként 256 BYTE kapacitással. A disk nevet kap (<**10 karakter** hosszú), ami betűvel kezdődik*

PRIMO: lása esetén az azonosítás >**CMD SET** α utasítással történik ($\alpha=9,10,\dots,15$)* A 8-as azonosítás technikailag megoldott*

TV COM: érhetünk el minden olyan szolgáltatást, amivel az iker-floppy potenciálisan rendelkezik*

VT-16: A képernyő vezeti tovább. Válaszai: **B** , **Y** , **3** , **ENT** , **N** legyenek az első próbánál* A lemez **egyik** vagy **mindkét** oldala formázható. Max. kapacitás 712 KBYTE*

△△

- A disk formázása abból áll, hogy a floppy írókarja 35 koncentrikus kört (track) és közöttük 683 cikket (sector) szerkeszt a lemezre (a legnagyobb sugarú körhöz 21, a legkisebbhez 17 sector tartozik). A track-ek és sector-ok sorszámot (1, ..., 35; \emptyset , 1, ..., 21) kapnak. A legnagyobb sector-sorszám 21.
- A 18-as számú track a disk regisztere. Itt tartja számon a floppy a disk címét (pl. VARGA), azonosítási számát (pl. \emptyset 1) és terhelési adatait (directory file).
- Minden szektor-ban 256 adat-byte és meghatározott számú regiszter-byte található. A 256 adat-byte-ot block-nak nevezzük. Egy disk 664 block-ot hordoz.
- A floppy 16 csatornával dolgozik (\emptyset , 1, ..., 15). A 15-ös csatorna kiemelt. Ezen továbbítjuk a floppy-nak szóló parancsokat. A floppy maga is számítógép, tehát bizonyos feladatok önálló megoldására képes.
- A floppy, mint eszköz a 8-as számmal azonosítható a rendszerben. Több floppy bekapcsolása esetén a 8, 9, ..., 15 azonosítók kerülnek bevezetésre.
- A formalizáló utasítás (OPEN15,8,15) eszerint megnyitja a 8-as floppy 15-ös csatornáját. Az első 15-ös itt is, mint a kazettás egységnél file azonosító (célszerűségi okokból mindig 15-nek választjuk). A megnyitást a parancs követi. A parancsot mindig karakteres formában kell megadni ("N \emptyset : VARGA, \emptyset 1"). Ebben N a NEW szó kezdőbetűje, \emptyset az írókar „meghajtó”, motorjának az indítója, VARGA a disk címe és \emptyset 1 az azonosítási száma. A név nem alapvetően

ABC: [72] Két kazettás egységgel oldja meg * >LOAD CAS : P10 [RET] * Átkapcsolás CA2-re >OUT58,16 [RET] * >OUT58,24 [RET] * ≧KAZETTÁT TESZ CAS2-BE* >SAVE CAS : P10 [RET] * A gép RESET gombjával visszakapcsol CAS1-re *

HTZ: [72] NEM *

TEXAS: [72] >OLD CS1* SAVE''DSK1. P10'' [ENT]*A rögzítés helyességét a gép nem tudja ellenőrizni*

PRIMO: [72] >LOAD ''P10''* >CMD SAVE ''P10''* >CMD TEST ''P10''*
P10 ≠ p10 erre vigyázzunk*

TV COM: [72] >POKE 2821,5 : POKE 2829,5 : POKE 2823, 0 : POKE2831,0 [RET] *
Az első két utasítás a kazettás egységhez rendeli az 5-ös, a második kettő pedig a floppy-hoz a 6-os csatornát * >LOAD#5: ''P10'' [RET] * >SAVE#6: ''P10'' [RET] * >VERIFY#6: ''P10'' [RET] * || READING A : P10 . CAS OK || hibátlan rögzítést jelez * A poke utasításokkal biztosított csatornák a gép kikapcsolásáig, illetve a kettős [RESET] -ig élnek. Nélkülük a floppy veszi a parancsokat *

VT-16: [72] >SAVE ''A : P10'' [ENT] Az A floppy-ban lévő lemezre rögzíti a P10 programot * >SAVE ''A : P10'' , A [ENT] ASC kódban rögzíti a P10 programot * >SAVE ''B : P10'' [ENT] a B floppy-val dolgozik * >SAVE ''P10'', P [ENT] az A floppy-val dolgozik és listázásra hozzáférhetetlenné teszi a programot (P lopás elleni védelem)* A rögzítés ellenőrzése automatikus*

fontos tartozék, tehát elhagyható, illetve ugyanaz a név akárhányszor felhasználható (pl. a lemezek a tulajdonos nevét viselik). A disk azonosítója viszont létfontosságú tartozék. Nélküle a lemez munkára nem fogható. A formázás megismétlésével rehabilitálhatjuk. Célszerű lemezeinknek a 01.....99 azonosítókat adni.

- Valamely csatorna megnyitása egyben file megnyitását és a floppy adminisztrációban block foglalást jelent, feltéve hogy lezárási (CLOSE) parancsot is adtunk. A CLOSE parancs elmaradása esetén munkánk kárba vész.
- A megformázott disk programok és adatok fogadására kész.

034

Program rögzítése DISK-re. SAVE és VERIFY utasítások.

72

Töltse be kazettáról a P10-es programot. Rögzítse mágneslemezre a P10-es programot és ellenőrizze a rögzítés pontosságát:



>LOAD "P10", 1, C

RET

|| READY ||



>SAVE "P10", 8

RET

|| SAVING FOR P10 ||
|| READY ||

*

>VERIFY "P10", 8

RET

ABC: CAS2-ről ugyanúgy kérdez le mint a CAS1-ről *

A kazetták közötti átkapcsolás címkézett utasítással is megoldható:

CAS2-re >1 \emptyset OUT58,16 : OUT 58,24 *

CAS1-re >1 \emptyset OUT58,0 : OUT 58,8 *

HTZ: NEM *

TEXAS: >OLD "DSK1.P1 \emptyset " * Több (≤ 3) floppy esetén DSK1; DSK2; DSK3 vagy "DSK. DISZKNEV. PROGRAMNEV" hivatkozás is lehetséges *

PRIMO: >CMD LOAD "P1 \emptyset " *

TV COM: >LOAD#6 : "P1 \emptyset " * Tegye át a lemezt a B floppy-ba és >LOAD#6 : "B: P1 \emptyset " utasítással próbálkozzék *

VT-16: >LOAD "A : P1 \emptyset " vagy >LOAD "P1 \emptyset " * Ha >LOAD "P1 \emptyset ", R utasítást adunk, akkor a betöltés után a program futása is elkezdődik * Védett programok a fentiekhez hasonlóan tölthetők be, de a LIST-re "ILLEGAL FUNCTION CALL" hibajellel válaszolnak * Védett programhoz a készítő sem férhet hozzá, ezért nem védett másolatot is rögzítsen *

```

SEARCHING FOR P10
VERIFYING
OK
READY

```

- »NYISSON EGY KARTONT A 01-ES DISK SZÁMÁVAL ÉS JEGYEZZE FEL A TÁROLT PROGRAM CIMÉT:
 »IRJA FEL CIMKÉRE A DISK AZONOSÍTÁSI SZÁMÁT ÉS RAGASSZA FEL A DISKRE (csak gyári, öntapadó címkét használjon):

- △△
- A SAVE és VERIFY utasítások csak az eszköz fizikai számában (1 helyett 8) térnek el a kazettás egység megfelelő utasításaitól.
 - Több floppy esetén a 8-at a rögzítést végző floppy azonosítási számával helyettesítjük. Gyárilag minden floppy 8-as. Átállítása fizikai és programozási eszközökkel is megoldható. A fizikai megoldás technikus feladata.

035 *Lekérdezés a DISK-ről. A LOAD utasítás.*

73 Kérdezze le a 01 disk -ről a P10 programot és futtassa:

```

. → * >LOAD "P10", 8 RET

|| READY ||

* >LIST RET

|| READY ||

* >RUN

```

ABC: [74] NEM*

HTZ: [74] NEM*

TEXAS: [74] >SAVE "DSK1.P1Ø" utasítást a gép **elfogadja** és **felülírja a P1Ø** programot.
A block-kok foglaltsága névre szól*

PRIMO: [74] >CMD SAVE "P1Ø" [RET]* || 63, **FILE EXISTS, ØØ, ØØ** || a gép válasza (Már létezik ilyen című file a lemezen)* A floppy nem villogtat, ha mégis, akkor >CMD RESET : CMD CLEAR [RET] állítja helyre a floppy munkakészségét*

TV COM: [74] >SAVE#6: "P1Ø" felülírja a korábban rögzített P1Ø programot*

VT-16: A név nem védi a programot. >SAVE "P1Ø" felülírásra vezet, ha P1Ø című program már van a lemezen* Felülírás ellen sem az A, sem a P nem nyújt védelmet* A címet 4Ø karakterig azonosítja a gép, feltéve hogy hosszú címetek adunk. Egyetlen eltérés elég ahhoz, hogy új programként rögzítse és ne felülírja a hasonló címet*

- △△
- A floppy veszi a gép LOAD parancsát. Kikeresi regiszteréből a helyfoglalás adatait, majd olvasó (egyben író) karja aktiválja a tárolt programot és a gép memóriájának adja át. LIST utasítással a képernyőn is megjeleníthető. Futtatása nem függ attól, hogy képernyőre kiírtuk-e. Ha a LOAD utasításban megadott cím az adott disk-en nem található "FILE NOT FOUND" szövegű jelzést kapunk.
 - Ha a LOAD utasításból a 8-as kimarad, a gép a kazettás egységre vonatkozó parancsnak fogja fel és "PRESS PLAY ON TAPE" felhívással él. A **RUN/STOP** gomb leütése után a LOAD parancs megismételhető.

74 Módosítsa a P10 adatait és rögzítse ismét P10 címmel a 01 disk-en:

 ← *

>SAVE "P10", 8

RET

|| ?FILE OPEN ERROR ||

- △△
- A floppy piros lámpája villogásával tiltakozik a már tárolt cím ismételt alkalmazása ellen és nem hajtja végre a parancsot. Megnyugtatása az alábbi parancsokkal történik:

 → *

>OPEN15,8,15 "10"

RET

>CLOSE15

RET

ABC: NEM**

HTZ: NEM**

TEXAS: »HELYEZZE BE A SZÁMITÓGÉPBE A "DISK MANAGER COMMAND MODULE" BLOCK-KOT. Üsse le a DISK MANAGER , majd DISK COMMANDS? és végül a MASTER DISK? kódokat, mire a disk katalógus megjelenik a képernyőn*

PRIMO: >CMD SAVE "É :P10" *
 >CMDLOAD "S" * A disk címlistáját a ismételt leütésével hívjuk a képernyőre*

TV COM: NEM* A disk tartalomjegyzékét magunknak kell vezetni* Az operációs rendszer fejlesztési stádiumban van*

VT-16: >SAVE " @ : P10" rögzítési parancs is működik*
 >FILES "A : " vagy csak >FILES ha a lemez az A floppy-ban fekszik* >FILES "B : " a B floppy-ra érvényes*

- A floppy alapállását elérve kezdeményezhetjük a módosított program rögzítését, elhatározástól függő módokon:

/A * >SAVE "P1Ø/A", 8

/B * >SAVE "@Ø : P1Ø", 8

- Az A esetben a módosított programot újnak ítéli a floppy és rögzítését a P1Ø program érintetlen meghagyásával, szabad block-on oldja meg. A két program külön-külön lekérdezhető.
- A B esetben megkeresi a floppy a P1Ø programot és felülírja. Ezzel az eredeti P1Ø program elveszik. A felülírást a @ jellel kell kérni. A szándékos felülírást tehát elfogadja a floppy, de a tárolt program védelmet élvez.

75 Állapítsa meg egy disk-en található programok neveit:

□ → * >LOAD "\$", 8

RET

|| READY ||

* >LIST

	VARGA	Ø1
1	P1Ø	PRG
1	P1Ø/A	PRG
662 BLOCKS FREE		
READY		

ABC: -

HTZ: -

TEXAS: Az **1** gomb lenyomásával leállítható a katalógus listázása*. A disk katalógussal kapcsolatos fogalmak : MASTER=gazda; DISKNAME=kártyanév; SIZE=terjedelem; AVAILABLE=felhasználható; USED=felhasznált*

PRIMO: A disk hiányát **||74, DRIVE NOT READY, 00, 00 ||** jelzi*

TV COM: A disk hiányát a **||SYSTEM ERROR 254. ||** jelzi*

VT-16: A **CTRL NUM/LOCK** leütésével megszakíthatjuk a listázást, de **bármelyik karakter** leütésekor a listázás folytatódik*
A floppy-ra vonatkozó programokat írja át wfloppy-ra is és dolgozzon velük*

△△

- A parancs lemeztől független. A § jel a directory file neve. A képernyőn egy fejléc, alatta 3 oszlopos információ jelenik meg. Az első oszlop a programok helyfoglalását (blockszám), a második a nevét, a harmadik pedig a jellegét (PRG≡program) közli. Az utolsó sorban a még szabad block-ok száma jelenik meg.
- Ha a lista túllépi a képernyő méretét, akkor a **RUN/STOP** gomb leütésével a kívánt helyen megállítható, vagy a **CTRL** gomb lenyomásával a kiírás lassítható. Ha lista helyett a vörös lámpa villogása a válasz, akkor valamilyen hiba áll fenn:
 1. nincsen disk a floppy-ban,
 2. hibásan kopogtuk le a LOAD ”§”, 8-at,
 3. a vizsgált disk ”sérült” stb.

Az első két eset szemmel eldönthető. A harmadik eset a LOAD ”§”, 8 utasítás megismétlésével válhat bizonyossá. Ezt megelőzően **RUN/STOP**-al leállítjuk a floppy-t, majd a vörös villogást is megszüntetjük. A hibásnak mutakozó lemezt ismételt formalizálással próbáljuk rehabilitálni. Ez viszont a tárolt program törlésével jár.

- A lemezhibák gyakori oka, hogy a disk a floppy-ban reked a rendszer kikapcsolása idejére, vagy a bekapcsolás előtt becsúztatjuk a használni kívánt lemezt. Hibás gyakorlat, ha a vörös lámpa villogását a floppy ki-be kapcsolásával hártjuk el.



Folytassa a munkát a **35** feladatnál. Rögzítse a 01 disk-en a további programokat. Erre a programok alatt megjelenő →☐ jel figyelmezteti!!!

ABC: 76 OK * Minden futásnál ugyanazokat a véletlen számokat kapjuk * >15
RANDOMIZE esetén a számok futásonként változnak * Végjelet is írhatunk a
file-hoz >45 PRINT#76, 100 *

HTZ: 76 >30 PRINT# -1, "LOTTO" * >40 :PRINT# -1, L(I) : NEXT : >50
PRINT# -1, "VEGE" : END * "VEGE" helyett valamilyen szám, pl. 100
is lehet végjel. Célszerű ugyanazt a végjelet használni, hogy ne tévesszünk *

TEXAS: 76 >20 L(I)=INT (1+RND*90)::* >30 OPEN#76 : "DSK1.LOTTO",
SEQUENTIAL, OUTPUT, VARIABLE 5 * >40 ::PRINT#76 : L(I) :: *
VARIABLE 5 azt jelzi, hogy a rekord-ok hossza változó, de legfeljebb 5 byte *

PRIMO: 76 >30 CMD CREATE 5, "LOTTO" * File megnyitása * >40 FOR I=1 TO 5 :
CMD PRINT#5, L(I) : NEXT * >50 CMD CLOSE5 : END * A rögzítési utasítá-
sokat a csatornaszám (5) kapcsolja össze *

TV COM: 76 >5 DIM L(5) : RANDOMIZE * >30 OPEN#6 : OUTPUT "LOTTO"*
>40 : PRINT#6:L(I)::* >50 CLOSE#6:OUTPUT : END Próbálja meg a prog-
ramot adatfile-ként (ASC kódban) rögzíteni * >OPEN#6 : OUTPUT "P76" :
LLIST#6 :: CLOSE#6 : OUTPUT RET *

VT-16: 76 >20 L(I) = INT (1+ RND (1) * 90) : PRINT L(I); : NEXT * >30 OPEN
"O", #76, "LOTTO" * "O" OUTPUT-ra, tehát rögzítésre való megnyitást
jelent * >40 FOR I=1 TO 5 : WRITE#76, L(I) : NEXT * WRITE=ír * >50
CLOSE#76 * Wfloppy-ra OK *

ABC: 77 >1Ø OPEN "LOTTO" AS FILE 77* >11Ø PREPARE "LOTTO" AS FILE 77* Eredeti területén az n_k pontos beállításával rögzíthetjük. >85 STOP lehetővé teszi a beállítást*

HTZ: 77 Kazettás egységekkel OK* Az eredeti területen való rögzítés csak akkor megy, ha >82 PRINT M(I) : STOP alatt visszacsévélünk*

TEXAS: 77 >1Ø OPEN#77 : "DSK1. LOTTO", SEOUENTIAL, INPUT, VARIABLE 5* >2Ø :: INPUT#77 : K(J) : :* >11Ø OPEN#77 : "DSK1. LOTTO", OUTPUT, VARIABLE* SEQUENTIAL deklaráció elmaradhat* VARIABLE max. 8Ø byte-os rekord-okra épít* Az adatfile névazonosság esetén felülíródik, védelme megoldatlan*

PRIMO: 77 >1Ø CMD OPEN 3, "LOTTO"* >2Ø FOR J=1 TO 5 : CMD INPUT#3, K(J) : M (J) = K (J) : NEXT : CMD CLOSE3* >11Ø CMD CREATE6, "É : LOTTO"* >12Ø FOR P=1 TO 5 : CMD PRINT#6, M(P) : NEXT* >13Ø CMD CLOSE6 : END* Az adat file felülírható az É alkalmazásával*

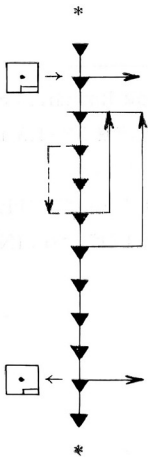
TV COM: 77 >5 DIM K(5), M(5)* >1Ø OPEN#6 : INPUT "LOTTO"* >2Ø : INPUT#6 : K(J) : : CLOSE#6 : INPUT* >11Ø OPEN#6 : OUTPUT "LOTTO"* >12Ø : PRINT#6 : M(P) : :* >13Ø CLOSE#6 : OUTPUT : END* Az adatfile nem védett, azonos címmel felülírható a lemezen*

VT-16: 77 >1Ø OPEN "1", #77, "LOTTO"* "1" INPUT-ra, tehát betöltésre való megnyitást jelent* >2Ø : CLOSE77 vagy CLOSE#77* >11Ø->13Ø mint 76 alatt* A szekvenciális file nem védi a neve az azonos nevű felülírástól* Wfloppy-ra OK*

- A file szekvenciális rögzítése az OPEN–PRINT#–CLOSE utasításokra épül. Mindhárom utasításban szerepel a file száma (pl. 76). Az OPEN utasítás rögzíti a floppy azonosítóját (8), a munkához lekötött csatorna (5) számát, az írókar (Ø) meghajtóját, a file (LOTTO) nevét, a rögzítés (SEQ) típusát és azt, hogy rögzítésről (WRITE=ir) van szó. A PRINT# a rögzítésre kerülő adatok azonosító (L(I)) változóit foglalja magában. A CLOSE utasítás lezárja a file-t, ezzel területét védetté válik. A CLOSE elhagyása a file elvesztését jelenti.

037 Szekvenciális file betöltése disk-ről. Számsor rendezése.

77 Töltse be a "LOTTO" nevű file-t. Rendezze a számokat növekvő sorrendbe, majd rögzítse a rendezett file-t eredeti területén:



```

>NEW
>1Ø OPEN77,8,3,"Ø : LOTTO, S, R"
>2Ø FOR J=1 TO 5 : INPUT#77, K(J) : M(J)=K(J) : NEXT : CLOSE77
>5Ø FOR I=1 TO 4 : FOR J=I+1 TO 5
>6Ø IF M(I)<=M(J) THEN 8Ø
>7Ø S=M(I) : M(I)=M(J) : M(J)=S
>8Ø NEXT J
>81 PRINT M(I); : NEXT I
>82 PRINT M(I)
>9Ø PRINT "A RENDEZETT FILET ROGZITJUK"
>11Ø OPEN77,8,6,"@Ø : LOTTO, S,W"
>12Ø FOR P=1 TO 5 : PRINT#77, M(P) : NEXT
>13Ø CLOSE77 : END
>RUN → [ ] ^

```

ABC: —

HTZ: A rekord-ok a bemutatásra kerülő P77 programnál csak 1–1 adatot tartalmaznak* Feltesszük, hogy -ban >50 PRINT# –1, 100 végjelet tartalmaznak* Többtagú rekordok esetén többtagú végjelre gondoljunk*

TEXAS: —

PRIMO: —

TV COM: Helyezze át a floppy dugaszát a balszélső kapuba. A csatornák hozzárendelését most a >POKE 2821,5 : POKE 2829,5 : POKE 2823, 3 : POKE 2831,3 utasítások látják el* Az utolsó két rekesz tartalma kapunként 1-gyel nő*
* Próbálja meg ASC kódban rögzített program betöltését*>OPEN#6 : INPUT "P76" : POKE2818,6 * || OK OK ||*>CLOSE#6 : INPUT *

VT-16: —

△△

- A szekvenciális file betöltése az OPEN–INPUT#–CLOSE utasításokra épül. A rögzítésnél alkalmazott utasítás-hármasból két helyen találunk eltérést. Az OPEN utasításban a W helyett R (READ=olvas) szerepel és a PRINT# helyére INPUT#-ot helyettesítettünk.
- A programozónak nem kell emlékeznie a file rögzítésénél alkalmazott file számra, a használt csatornára, a ciklus indexére és az adatok változóira. Ezek a betöltésnél szabadon választhatók. A file nevét és a rekord-ok összetételét (adatszám, típus) viszont ismerni kell. A rekord-ok számának az ismerete kellemes, mert akkor ciklussal tölthető be. Ennek hiányában az ST megfigyelésére vagyunk utalva.
- A számok rendezése párok összehasonlítására ($M(I) < M(J)$?) és a tagok helycseréire épül. Ez utóbbit az S segédváltozó (>70) közvetíti.
- A floppy védi a disk-en rögzített file-okat. Egy disk nem fogad azonos nevű file-okat. Az ilyen rögzítési kísérlet csak akkor sikeres, ha a file név elé felülíró @ jelet illesztünk. Ekkor viszont az eredeti file felülíródik, tehát eltűnik.

038 Az ST használata adatfile-ok betöltésénél.

78 Módosítsa a **77** programot arra az esetre, ha az adatok (rekord-ok) számát nem ismerjük:

ABC: [78] >2Ø INPUT#77, L : K=K+1 : M (K)=L : IF L=1ØØ THEN 3Ø ELSE 2Ø
* Ezt akkor alkalmazhatjuk. ha végjelet (itt 1ØØ) rögzítettünk a file végére *
[79] A gép saját végjelet generál a file végére.

HTZ: [78] >2Ø INPUT# -1, L : K=K+1 : M (K)= L : IF L <> 1ØØ THEN 2Ø * >3Ø
REM A RENDEZÉS * >5Ø FOR I=1 TO K-1 : FOR J=I+1 TO K * >12Ø FOR
P=1 TO K : *
[79] Ha [76] -nál nem alkalmazunk végjelet. a bővítés nem oldható meg a jel-
zett módon *

TEXAS: [78] >2Ø ::IF EOF(77)=Ø THEN 2Ø * EOF(77)=1, ha a file utolsó rekord-ját
beolvasta a gép (előtte EOF(77)=Ø) *
[79] >OLD DSK1. P76 * >3Ø OPEN#76 : "DSK1. LOTTO", SEOUENTIAL,
APPEND, VARIABLE 5 *

PRIOMO: [78] >2Ø CMD INPUT#3, L : K=K+1 : M(K)=L : IF NOT EOF(3) THEN 2Ø
* >3Ø CMDCLOSE3 * >12Ø FOR P=1 TO K : CMD PRINT#6, M(P) : NEXT
* END OF FILE=vége a file-nak * EOF(3) vége a 3-as csatornán megnyitott
file-nak * [79] >3Ø CMD CONT5, "LOTTO" *

TV COM: [78] Ha csatorna hozzárendelésünk (POKE) elmarad, a floppy a kazettás
egység utasításaira reagál * >2Ø INPUT#5 : L : K=K+1 : M(K)=L : GET#5 : X\$
: S= PEEK (2926) : IF S>Ø THEN GOTO 3Ø : ELSE 2Ø * >3Ø CLOSE#5 :
INPUT * A GET#5 A rekord elválasztókat köti le *

VT-16: [78] >2Ø : IF NOT EOF(77) THEN 2Ø * >12Ø : WRITE#77, M(P) : * END OF
FILE=vége a file-nak * EOF(77) vége a 77-es logikai számú file-nek * Próbálja
meg az IF EOF(77)=Ø THEN 2Ø utasítást használni * Wfloppy-raOK *
[79] >3Ø OPEN "A", #76, "LOTTO" * "A" APPEND=csatol * >3Ø OPEN
"A", #76, "C : LOTTO" OK *

```

> 5 DIM M(255)
> 2Ø INPUT#77, L : K=K+1 : M(K)=L : IF ST < 64 THEN 2Ø
> 3Ø CLOSE77
>5Ø FOR I=1 TO K-1 : FOR J=I+1 TO K
>12Ø FOR P=1 TO K : PRINT#77, M(P) : NEXT
>RUN → [ ] Δ

```

*

ΔΔ – A rekord-ok számát nem ismerve a FOR–NEXT ciklusos beolvasásról le kell mondani. A ciklust most az ST<64 kezdeményezi és a GOTO 2Ø (THEN 2Ø) ismételteti. Mivel a számhalmaz rendezéséhez indexelt azonosítókát használunk, a betöltött adatoknak indexet kell adni (K=K+1 : M(K)=L.) A rendező és rögzítő ciklusok számát a K utolsó értéke adja meg >2Ø-ban, de ehhez csak a program futása közben férünk hozzá. Ezért szerepel >5Ø és >12Ø-ban K. A DIM utasításra az M tömb miatt van szükség. Mivel dimenzió vissza nem datálható, (pl. >2Ø-ból >5-be), a szóbjáöhető legmagasabb dimenzió számot (255) adtuk M-nek.

039 *File-ok bővítése (APPEND).*

79 Töltse be disk-ről a P76-ot és futtassa újra. Az új számokat rögzítse a "LOTTO" nevű file bővítményeként:

```

* >LOAD "P76", 8
* >LIST
  »MODOSITSA A >3Ø UTASITÁST:
  >3Ø OPEN76,8,5, "Ø : LOTTO, S, A"
* >RUN → [ ] Δ

```

ABC: A bővítés nem oldható meg a jelzett módon* Ha betölti a meglévő file-t és kiegészíti újabb számokkal, rögzítve bővített file-t nyer*

80 NEM*

HTZ: Ha betölti a meglévő file-t és kiegészíti újabb számokkal, rögzítve bővített file-t nyer*

80 NEM*

TEXAS: 80 >40 OPEN#66 : "DSK1. KESZLET", OUTPUT : :* SEQUENTIAL ÉS VARIABLE a megnyitásnál elmaradhat, a pár automatikusan érvényesül*
>50 PRINT#66 : C ; " , " ; M\$; " , " ; V ; " , " ; E ; " , " ; D\$ *

PRIMO: 80 >40 CMD CREATE 2, "E : KESZLET" : :* >50 CMD PRINT#2, C, M\$, V, E, D\$* >60 NEXT : CMDCLOSE2 : *

TV COM: 79 NEM*

80 >LOAD#5: "P66" RET* >40 OPEN#5 : OUTPUT "KESZLET"
:* >50 PRINT#5 : C ; " , " ; M\$; " , " ; V ; " , " ; E ; " , " ; D\$ >60 NEXT
: CLOSE#5 : OUTPUT : END* HA >50 alatt D\$; " , " végződés szerepel, a gép elfogadja*

VT-16: 80 >40 OPEN "O", #66, "KESZLET" : :* >50 WRITE# 66, C, M\$, V, E, D\$ *

△△

– Adatfile-ok bővíthetők. Bővítés esetén az OPEN utasításban a W jel helyett az A (APPEND=hozzáfűz) kezdőbetű szerepel. A hozzáfűzés mint eljárás fontos lehetőség. Nagytömegű adatot célszerű részletekben felvinni a gépbe és a disk-re. Bizonyos adat-együttesek időről-időre reprodukálódhatnak és a kumulálódó adattömeget ugyanabban a file-ban tartjuk nyilván. Nyilvántartások állandóan változnak, belső tartalmuk módosulhat, méretük is kiterjedhet.

80

Olvassa be disk-ről a P66 programot. Módosítsa oly módon, hogy a raktárkészletet disk-en rögzítse:

*

```
>LOAD "P66", 8
```

RET

*

```
>LIST
```

```
10 INPUT "A TETEEK SZAMA"; N
30 PRINT "CIKKSZ, MEGNEV, MENNYI, EGYSEGA,
MERTEKE" : PRINT
40 OPEN66,1,2, "KESZLET" : FOR I=1 TO N: INPUT C, M$,
V, E, D$
50 PRINT#66, C; CHR$(13) M$; CHR$(13) V; CHR
$(13) E; CHR$(13) D$; CHR$(13)
60 NEXT : CLOSE66 : END
```

```
>40 OPEN66,8,2, "0 : KESZLET, S, W" : FOR
```

*

```
>RUN →  Δ
```

→

Vigyen fel adatfile-okat disk-re. Változtassa a rekord összetételét.

ABC: 81 Ha rögzítésnél **PRINT#66, 0, "M", 0, 0, "D"** végjelet alkalmazunk, akkor
 >30 IF C <> 0 THEN 20 megoldással OK *

 82 Lásd P66-os megoldást *

HTZ: 81 Ha rögzítésnél **PRINT# -1, 0, "M", 0, 0, "D"** végjelet alkalmazunk,
 akkor >30 IF C <> 0 THEN 20 megoldással OK *

 82 Lásd P66-os megoldást *

TEXAS: 81 >10 OPEN#81 : "DSK1. KESZLET", INPUT * >20 :: PRINT C; M\$;
 V ; E ; D\$ * >30 IF EOF(81) THEN 20 * A feltétel teljesül, ha EOF(81)=0,
 ELSE ha EOF(81)≠0.

 82 A törlésnek nincs alapja *

PRIMO: 81 >10 CMDOPEN 10, "KESZLET" * >20 CMD INPUT#10, C, M\$, V, E,
 D\$ * >30 IF NOT EOF(10) THEN 20 *

 82 P66 miatt hiba nem merül fel *

TV COM: 81 >10 OPEN#5 : INPUT "KESZLET" * >20 INPUT#5 : C, M\$, V, E, D\$:
 PRINT C, M\$, V, E, D\$: GET#5 : X\$ * >30 S=PEEK (2926) : IF S>0 THEN
 40 : ELSE 20 >40 CLOSE#5 : INPUT : END *

 82 OK *

VT-16: 81 >10 OPEN "I", #81, "KESZLET" * >30 IF NOT EOF(81) THEN GOTO
 20 * Próbálja ki az alábbi formát: >10 OPEN "I", #81, "KESZLET" : WHILE
 NOT EOF(81) * >30 WEND * A WHILE-WEND ciklusszervező utasítás * Próbálja ki
 >20 INPUT#81, R\$: PRINT R\$ utasítást *

 82 NEM *

ABC: [83] A nyilvántartás minden bővítésénél a **végjelet változtatni** kell, hogy az újabb betöltés sikeres legyen* Betöltés előtt >21-et **aktualizálni** kell* A >470 miatt a **C=0** NEM alkalmas végjel*

HTZ: [83] A nyilvántartás minden bővítésénél a **végjelet változtatni** kell, hogy az újabb betöltés sikeres legyen* Ujabb betöltés előtt >21-et módosítani kell az **aktuális** végjelre* A >470 miatt a **C=0** NEM alkalmas a végjelben*

TEXAS: [83] >12 OPEN#83 : "DSK1. KESZLET", INPUT* >15 INPUT#83 : C1, M1\$, V1, E1, D1\$:: I=I+1* C és C(I) nem fér meg egy programban, de C1 és C(I) igen* >16 C(I)=C1 :: M\$(I)=M1\$ stb..* >210 :: IF V(I)<K THEN 215 ELSE 220* >215 PRINT C(I) ; M\$(I) ; V(I) ; E(I) ; D\$(I)*

PRIMO: [83] OK* >CLEAR N utasítás a RUN előtt direkt formában, illetve program utasításként is beadható. Az utoljára beadott kapacitás foglalás él* A >NEW nem törli a helyfoglalást*

TV COM: [83] >15 INPUT#5 : C1, M1\$, V1, E1, D1\$: GET#5 : X\$: I=I+1* >16 C(I)=C1 : M\$(I)=M1\$ stb.* >21 S=PEEK (2926) : IF S>0 THEN 25 : ELSE 15* >120 PRINT "KESZLETERTEK = " ; S ; "FT" :* >201 INPUT PROMPT "KORLAT?" : K :* C és C(I) nem fér meg egy programban*

VT-16: [83] >12 OPEN "I", #83, "KESZLET": WHILE NOT EOF(83)* >21 WEND* WFLOPPY >15 miatt nem OK* P67 szellemében át kell írni az egész programot* A vizsgálatokhoz MID\$ és VAL függvényekkel választjuk le a megfelelő adatokat* Törlés esetén az üres (" ") string-et rögzítjük stb.*

Írjon alkalmazói programot a P82-vel rögzített készlet file kezelésére az alábbiak szerint:

1. A raktárkészlet értékének a kiszámítására,
2. Adott "k" korlát alatti készletek kilistázására,
3. A nyilvántartás "w" számú cikkel való bővítésére,
4. A 00–40-nel végződő cikkszámok rekordjainak a törlésére,
5. Az eredmény file visszarögzítésére.

```
*
>NEW
>10 PRINT "A KESZLET FILE BETOLTESE" : PRINT
>11 DIM C(255), M$(255), V(255), E(255), D$(255)
>12 OPEN#83,8,12, "0" : KESZLET, S, R"
>15 INPUT#83, C, M$, V, E, D$ : I=I+1
>16 C(I)=C : M$(I)=M$ : V(I)=V : E(I)=E : D$(I)=D$
>20 PRINT C(I), M$(I), V(I), E(I) ; D$(I)
>21 IF ST <>64 THEN 15
>25 CLOSE#83 : N=I

>100 PRINT "A RAKTARKESZLET ERTEKSZAMITASA"
: PRINT : PRINT "*****"
>110 FOR I=1 TO N : S=S+ V(I) * E(I) : NEXT
>120 PRINT "KESZLETERTEK ="S"FT" : PRINT

>200 PRINT "A KORLAT ALATTI CIKKEK JEGYZEKE"
: PRINT : PRINT "*****"
>201 INPUT "A MEGADOTT KORLAT?"; K : PRINT
>210 FOR I=1 TO N : IF V(I) < K THEN PRINT C(I), M$(I),
V(I), E(I); D$(I) : PRINT
>220 NEXT : PRINT "*****"
* * *
```

ABC: -

HTZ: A >465-öt töröljük*

TEXAS: >4Ø1 DIM Q(255)* >465 OPEN #83 : "DSK1. KESZLET", OUTPUT *

PRIMO: >521 CMD PRINT#12, C(I), M\$(I), V(I), E(I), D\$(I) *

TV COM: >465 OPEN#5 : OUTPUT "KESZLET"* >521 PRINT#5 : C(I); ", ";
M\$(I); ", "; V(I); ", "; E(I); ", "; D\$(I)* >53Ø NEXT : CLOSE#5 :
OUTPUT **

VT-16: >465 OPEN "O", #83, "KESZLET"* >521 WRITE#83, C(I), M\$(I),
V(I), E(I), D\$(I)* Próbálja meg az >521-et változatlanul hagyni. OK *

```

>300 PRINT "A NYILVANTARTAS BOVITese" : PRINT
>301 INPUT "HANY TETELLEL BOVIT?"; W : PRINT
>310 FOR I=N+1 TO N+W : INPUT C(I), M$(I), V(I), E(I), D$(I)
      : NEXT
>320 PRINT "A BOVITETT NYILVANTARTAS:" : PRINT
>330 FOR I=1 TO N+W : PRINT C(I), M$(I), V(I), E(I) ; D$(I)
      : NEXT

>400 PRINT "CIKKEK TORLESE * * * 00 - * * * 40-IG"
      : PRINT : PRINT " * * * * * "
>401 DIM Q(N+W)
>410 FOR I=1 TO N+W : Q(I) = C(I)/100-INT (C(I)/100)
>420 IF Q (I) >0.40 THEN 450
>430 C(I)=0 : M$(I)=" " : V(I)=0 : E(I)=0 : D$(I)=" "
>450 NEXT

>460 PRINT "AZ EREDMENY FILE" : PRINT " * * * * * "
>465 OPEN83,8,12, " @ 0 : KESZLET, S, W"
>470 FOR J=1 TO N+W : IF C(J)=0 THEN 530
>480 I=I+1 : C(I)=C(J) : M$(I)=M$(J) : V(I)=V(J)
      : E(I)=E(J) : D$(I)=D$(J)
>481 PRINT C(I), M$(I), V(I), E(I); D$(I) : PRINT
>521 PRINT#83, C(I); CHR$(13)M$(I); CHR$(13) V(I)
      ; CHR$(13) E(I); CHR$(13) D$(I)
>530 NEXT : CLOSE83 : END
>RUN      [ ]      ^

```

*

△△

- A program futtatása előtt gondoskodni kell a megfelelő (P82 alkalmazásával nyert) "KÉSZLET" file-ról.
- Az ST=64-re alapozott file betöltési módnál a rekord-okat megszámlálhatjuk (>15).

ABC: —

HTZ: —

TEXAS: —

PRIMO: —

TV COM: —

VT-16: —

Ez nem kerülhető el, ha a további feldolgozás érdekében a rekord-ok adatait tömbökbe (C(I),...,D\$(I)) kell sorolni (>16). Mivel a tömbök retrográd (visszamenőleges) módon nem dimenzionálhatók a megengedett legnagyobb (255) dimenziót kölcsönözzük számukra. Ha az így dimenzionált tömbök száma elég sok, a központi memória túlélése állhat elő. A dimenzió ugyanis előzetes memória foglalást biztosít. Ilyen esetben lefele engedünk (pl. C(50)).

- A file bővítése új rekord-okat, esetünkben bővített tömböket is jelent. Ezért olyan ciklusos input-tal dolgozunk, ahol a számláló N+1-el indul.
- Rekordok törlése olyan értékadást jelent, ahol a numerikus változó a nulla, a karakteres változó pedig az üres (" ") string-gel azonosul. A kiválasztásra általános receptet adni nem lehet, de annyi bizonyos, hogy valamilyen logikai függvény alapján történhet. Példánkban a két utolsó jegyet le kell választani, hogy a logikai függvénybe helyettesíthető legyen. Erre a törtalakban való kinyerés látszik célravezetőnek (pl. 35418; 354,18–354=0,18).
- A törlések után a bővített file \emptyset " " \emptyset " " alakú és eredeti rekord-okból áll. A disk-re rögzítés előtt kiszűrjük a törölt elemeket. Ezt átindexeléssel (>47 \emptyset , >48 \emptyset) oldjuk meg, de ne felejtjük el, hogy a file utolsó rekordjai eredeti indexükkel is fennmaradhatnak. Ezért az átindexelő cikluson belül kell a disk-re rögzítést is megoldani.

ABC: >11 * >600 * >331 N=N+W : GOSUB 610* >610 lásd
P66*

HTZ: >11 * >600 * >331 N=N+W : GOSUB 610*

TEXAS: >11 *
>600 OPEN#83 : "DSK1. KESZLET", OUTPUT*
>610 :: PRINT#83 : C(I) ; " , " ; M\$(I) ; " , " stb.*

PRIMO: >11 *

TV COM: >11 OK*

VT-16: >11 *

Képesítse a 83 programot a feladatok egymástól független megoldására:

```

* >LOAD "P83", 8 : LIST-100 : LIST 100-200 : LIST 200 -
>1 DIM C(100), M$(100), V(100), E(100), DS(100), Q(100)
>2 PRINT "A SZOLGALTATASOK JEGYZEKE" : PRINT
>3 PRINT "***1**KESZLETERTEK SZAMITAS"
>4 PRINT "***2**KORLAT ALATTI CIKKEK LISTAJA"
>5 PRINT "***3** A NYILVANTARTAS BOVITESE"
>6 PRINT "***4**TORLES A CSZ 2 UTOLSO JEGYE A"
>7 PRINT "MI AZ ON KIVANSAGA? " : INPUT Z
>8 ON Z GOSUB 100, 200, 300, 400
>9 FOR I=1 TO N : C(I)=0 : M$(I)=" " : V(I)=0 : E(I)=
  0 : D$(I)=" " : NEXT : N = 0 : I = 0 : GOTO 2
>26 RETURN
>101 GOSUN 10
>120 PRINT "KESZLETERTEK = "S" FT" : PRINT : S =0
>121 RETURN
>201 GOSUB 10
>202 INPUT "A MEGADOTT KORLAT?"; K : PRINT
>221 RETURN
>301 GOSUB 10
>302 INPUT "HANY TETELLEL BOVIT? "; W : PRINT
>331 N = N+W : GOSUB600
>332 RETURN
>401 GOSUN 10
>531 N=1 : FOR P=1 TO N : Q(P)=0 : NEXT
>532 RETURN
>600 OPEN8 3,8,2" @0 : KESZLET, S, W" : X$=CHR$(13)
>610 FOR I=1 TO N : PRINT#83, C(I); X$M$(I); X$V(I)
  ; X$E(I); X$D$(I) : NEXT
>620 CLOSE 83 : RETURN
* >RUN → □ Λ

```

ABC: -

HTZ: -

TEXAS: -

PRIMO: -

TV COM: -

VT-16: -

- Összetartozó, de egymástól függetlenül aktíválható programok együttesét program-csomagnak nevezzük. A csomag bevezető utasításai a befoglalt szolgáltatásokról és aktiválásuk módjáról tájékoztatnak (>2->7). Az aktivitást egy INPUT paraméter (Z) kötött értékadása (Z=1,2,3,4) mellett az ON Z GOSUB C₁,..., C₄ – RETURN utasításpár oldja meg. Az ON GOSUB ugyanis Z=1 esetben C₁-re, Z=2-nél C₂-re, stb. adja át a vezérlést. A C₁, C₂,... címkéket pedig a megfelelő szubrutinok nyitó címkéivel azonosítjuk.
- A szubrutin is behívhat rutinokat. Ilyen megoldást találunk pl. a 3. feladatnál, ahol a file beolvasását, majd a bővített file rögzítését külső rutinok oldják meg. Ezek a GOSUB C – RETURN utasításpár hatására lépnek majd ki. A RETURN akár az ON GOSUB, akár a GOSUB páryaként funkcionál, ugyanoda a vezérlést átadó utasítást követő címkére adja vissza a vezérlést. Kövessük nyomon a Z=3-hoz tartozó vezérlést a 83 és 84 programokon: A >8 átadja a munkát >300-nak. A >300 aktiválja A >10-et. A file betöltése >25-nél befejeződik. >26 vissza léphet >302-re. Az új rekordok bevezetése >331-ben végetér és >600 kapja a vezérlést. A bővített file disk-re vitele megtörténik és >620 átadja >332-nek a lépés jogot és mivel ez az ON-GOSUB RETURN párja, a vezérlést >9-re adja át. Itt törlik a használt változókat és a gép a >2-be lépve ismét felajánlhatja szolgálatait. Ügyeljünk a DIM utasítások elhelye-

ABC: [85] NEM*

HTZ: [85] NEM*

TEXAS: [85] A [84] -et csak 3 helyen kell módosítani: >11 **ON ERROR 25 :: REM HIBAFELTÁRÁSHOZ UTASÍTÁS*** >24 **CLOSE#83 :: N=1*** >25 **CALL ERR (W, X, Y, Z) :: PRINT W; X; Y; Z*** W=hibakód; X=file szám; Y=-1 (utasítás); 9 (egyéb); Z=hiba címkéje* Az **ON ERROR-CALL ERROR** a program egészét ellenőrzi hiba szempontjából. Észlelés esetén 25-re kerül a vezérlés és a kép-

PRIMO: [85] >11 Törölve* >16 **GOSUB 25 *** >21 **IF NOT EOF(12) THEN 15*** >22 **CMD CLOSE 12 : N=1*** >25 **CMD ERROR EN, EM\$, ET, ES : IF EN >1 THEN PRINT EN; EM\$; ET; ES*** A 15-ös "parancs" csatornát nem kell megnyitni (így lezárni sem) a hiba lekérdezéshez*

TV COM: [85] NEM*

VT-16: [85] Ne vegye figyelembe a jobboldali módosításokat. Helyette: >11 **ON ERROR GOTO 630*** >630 **PRINT "HIBAKOD=" ERR, "A HIBAS UTASITAS CIMKEJE=" ERL*** >640 **PRINT "JAVITSA KI A HIBAT" : STOP***

- zésére, mert az ismételt dimenzionálást a gép visszautasítja és leáll a programmal.
- Vegye birtokba a P84-et és dolgozzon vele. Változtassa a feladatok feltételeit. Ha közben kiürülne a disk-ről hívott "készlet" file, töltsse fel a P82 alkalmazásával.
 - Programcsomagunk további feladatokkal és a feladatok változataival is bővíthető. Ez utóbbi esetben a 100, 200, 300 és 400 címke kiinduló pontja egy-egy al-programcsomagnak.

042

Az **OPEN15,8,15 – INPUT#15, EN, EM\$, ET, ES** utasítások használata.

85

Módosítsa a **84** programot, hogy ellenőrizhesse a file betöltés kimenetelét:

```
*
      >LOAD "P84", 8 : LIST-100

      _____
      >11 OPEN15,8,15
      >13 GOSUB 25
      >16 RS=ST : GOSUB 25
      >17 C(I)=C : M$(I)=M$ : V(I)=V : E(I)=E : D$(I)=D$
      >21 IF RS <> THEN 15
      >22 CLOSE83 : N=I : CLOSE 15
      >23 RETURN
      >25 INPUT#15,EN,EM$,ET,ES : IF EN >1 THEN
          PRINT EN, EM$, ET, ES
      >26 RETURN
      _____
```

△△ – A file műveletek az **OPEN15,8,15 – INPUT#15, EN, EM\$, ET, ES – CLOSE15** utasításokkal ellenőrizhetők.

ABC: —

HTZ: —

TEXAS: ernyőn megjelenik a hiba kódja, amit a felhasználói kézikönyvből azonosíthatunk.
X, Y, Z kiegészítés*

PRIMO: —

TV COM: —

VT-16: —

- Ha a floppy a >11 , illetve >12 utasításokat valamelyik disk hiba miatt nem tudja végrehajtani, akkor jelzést ad a hiba száma (EN), és megnevezése (EM\$), valamint fellépési helye (ET=track szám; ES=sector szám) megjelölésére. Ezeket az információkat INPUT#15-tel kérdezhetjük le a 15-ös csatornán, amit a lekérdezés előtt meg kell nyitni. Ha a file művelet zavartalan, akkor \emptyset , OK, \emptyset , \emptyset a hibaváltozók értéke. Ellenkező esetben a négy adat tájékoztat a hiba jellegéről, ami egyben a betöltés megghiusulását is jelentheti. Ilyen esetben megismételjük a betöltési műveletet, feltéve, hogy a file nem hibásodott meg és vált használhatatlanná.
- Példánkban a >13 csak a file megnyitását, a >16 viszont a rekordok beolvasását tételesen ellenőrzi. Itt az a furcsa jelenség tapasztalható, hogy az ST= \emptyset az ellenőrzés alatt ST=64-re vált, holott a betöltésnek még csak a kezdeténél tartunk és egy rekord beolvasása után, a >21 leállítja a műveletet. Ezt úgy oldjuk fel, hogy egy segédváltozóval azonosítjuk az ST betöltéskor felvett értékét, (RS=ST) és csak utána kérdezzük le a hibajeleket. Így RS=64 csak akkor áll elő, amikor ST a file végét érzékeli.
- A betöltési művelet végén nemcsak a file-t hanem a parancs csatornát is le kell zárni. Erre szolgál a CLOSE15. Fontos, hogy a 15-ös csatorna megnyitása és lezárása bérgyazóként jelenjen meg a file megnyitása és lezárása vonatkozásában (azaz OPEN15–OPEN83–CLOSE83–CLOSE15).

ABC: NEM* Szalag használata igen nehézkes*

HTZ: NEM* Szalag használata igen nehézkes*

TEXAS: OK*

PRIMO: OK* A program futtatása előtt biztosítson a stringek számára elegendő tároló kapacitást* >CLEAR 1000 *

TV COM: 18 karakternél hosszabb szópárok esetén T\$(N)*30 kerül >5-be*

VT-16: OK* Wfloppy OK*

A CLOSE15 ugyanis az addig megnyitott munka file-okat is lezárja. Gyakorlatilag a CLOSE83 el is hagyható. Ezt mégsem tanácsoljuk, mert ha a program futási idejét erőteljesen megnövelő hibalekérdésezést elhagyjuk a programból akkor a CLOSE15-re nem számíthatunk a file-ok lezárásánál és nyitva maradhatnak, ami mint tudjuk, a felülírásukkal fenyeget.

- A hibalekérdézés és kijelzés (INPUT#15-PRINT) utasításait szubrutinként kezelhetjük (>25). Ezt az is indokolja, hogy nemcsak a betöltési, hanem a rögzítési műveleteket is ellenőrizhetjük ugyanabban a programban. Ilyen okok miatt találjuk az OPEN15-öt a program elején, a CLOSE15-öt pedig a program végén, gondosan ügyelve arra, hogy a vezérlés megnyitás nélküli lezárást, illetve lezárás nélküli megnyitást ne kezdeményezhessen, mert a program hibajellel fog leállni.

043

STRING függvények. A LEN, LEFT\$ és RIGHT\$ használata.

86

Írjon programot a COMMODORE64 programozását és működését kísérő angol szavak¹⁴ és magyar jelentésük disk-en való rögzítésére és kétnyelvű használatára:

```

*
>NEW
>1 PRINT "ANGOL-MAGYAR; MAGYAR-ANGOL SZÓTAR"
>2 INPUT "HANY ANGOL SZOT TESZ A TARBA?"; N
>5 DIM A$(N), M$(N), X(N), Y(N), T$(N)
>10 PRINT "X(I) AZ ANGOL A$(I), Y(I) A MAGYAR M$(I)
      SZAVAK HOSSZAT MERI" : PRINT
>20 FOR I=1 TO N : INPUT "ANGOL SZO?"; A$(I) :
      X(I)=LEN (A$(I))

```

ABC: -

HTZ: -

TEXAS: [86] >35 ON ERROR 180* >40 :: Z\$ törölve* >41 törölve* >100 ::
OPEN15,8,15 törölve* >110 :: IF SEG\$ (T\$(P), 1, X (P))=K\$ THEN 115
ELSE 120* >115 PRINT K\$, "="; SEG\$ (T\$(P), X (P) + 1, Y(P)):: GOTO 60*
>150 OPEN15,8,15 törölve* >160 :: IF SEG\$ (T\$(P), X(P) + 1, Y(P)) = K\$
THEN 165 ELSE 170*

PRIMO: [86] >35 törölve* >40 CMD CREATE2, "SZOTAR"* >50 FOR I=1 TO N :
CMDPRINT#2, X(I), A\$(I) + M\$(I), Y(I) : NEXT : CMDCLOSE2* >180
CMDERROR EN, EM\$, ET, ES : * >200 CMDOPEN2, "SZOTAR"* >210
CMDINPUT#2, X, T\$, Y:*

TV COM: [86] >35 törölve* >40 OPEN#5 : OUTPUT "SZÓTÁR"* >41 törölve*
>50 : T\$(I)=A\$(I) & M\$(I) : PRINT#5 : X(I); " , " ; T\$(I); " , " ; Y(I) : NEXT
:CLOSE#5 : OUTPUT* >56END* >59 DIM A\$(200), M\$(200), T\$(200),
X(200), Y(200)* >110 FOR P=1 TO I : A\$(P)=T\$(P) (:X(P)) : IF (A\$(P)=K\$)
THEN PRINT K\$, "="; T\$(P) (X(P)+1:) : GOTO60* OPEN15,8,15 törölve*
>160 FOR P=1 TO I : M\$(P)=T\$(P) (X(P)+1:) : IF (M\$(P)=K\$) THEN PRINT
K\$, "=";T\$(P) (:X(P)) : GOTO 60* >180, 190 törölve* >201 törölve*

VT-16: [86] >35 ON ERROR GOTO 240* >40 OPEN "O", #86, "SZOTAR"*
>41 törölve* >50 :WRITE#86, X(I), A\$(I) + M\$(I), Y(I) : NEXT : CLOSE#86*
>100 és >150-ből : OPEN15,8,15 törölve* >180 ->190 törölve* >201 töröl-
ve*

ABC: —

HTZ: —

TEXAS: [86] >165 PRINT K\$; " = " ; SEG\$(T\$(P),1,X(P)) :: GOTO 60* SEG\$ (" " ,
I, J) a szöveg I-edik karakterétől kezdve J karaktert olvas ki* >180 CALL ERR
(W1, X1, Y1, Z1) :: PRINT W1; X1; Y1; Z1* >201 törölve* >220 IF EOF(86)
THEN 210* >230 CLOSE#86*

PRIMO: [86] OPEN 15,8,15 és CLOSE15 mindenütt törölve* A >RUN a szótár
felhordása, a >RUN60 a szótár használata során alkalmazható*

TV COM: [86] >210 INPUT#5 : X1, T1\$, Y1 : GET#5 : V\$: I=I + 1 : X(I) = X1 : T\$(I)=
T1\$: Y(I)=Y1* >220 S=PEEK(2926) : IF S>0 THEN 230 : ELSE 210* >230
CLOSE#5 : INPUT* >RUN* >RUN59*

VT-16: [86] >220 IF NOT EOF(86) THEN 210* >230 CLOSE#86* >231 RETURN*
>240 PRINT "HIBAKOD =" ERR, "CIMKE =" ERL, "JAVITSA A HIBAT" :
STOP* >RUN60 a szótár használatára szolgál. Ha helyette RUN kerül beütésre
a szótár **törlődik**. Ez kivédhető, ha >40-ben WRITE helyett APPEND-et szere-
peltetünk az alapítás után*

- A program első része (>1–>56) kétnyelvű szótár disk-re vitelét oldja meg. A szópáron (A\$, M\$) kívül a szavak hosszát (betűszámát) is rögzítjük (X, Y). A szóhosszt a LEN függvény adja meg. Célszerűségi okokból a szópárt egy string-be szorítjuk (A\$+M\$) és így rögzítjük.
- A program második része (>60–>231) a szótár használatát biztosítja. Kiindulhatunk akár magyar, akár angol szóból. Az igény rögzítését a "SZOTAR" file betöltése követi (>200–>231). Ezután jön az azonosítás. Ha angol szóból indul ki az igény, akkor az A\$+M\$ balfelét elemről-elemre összevetjük az adott szóval és ahol egyezés keletkezik, ott az A\$+M\$ jobbfelét válaszként írjuk ki. Szerencsés körülmény, hogy a LEFT\$(T\$, X) függvény a T\$ szöveg baloldaláról az első X, a RIGHT\$(T\$, Y) pedig a jobb oldaláról az utolsó, Y karaktert választja le. A magyar szóból való kiindulás fordított eljárást igényel.
- A betöltési és rögzítési műveleteket tételeken ellenőriztük a [85]-ben. Itt csak a file megnyitás ellenőrzésére szorítkoztunk.
- A parancs csatornát 3-szor nyitjuk és zárjuk a programban. Ezt a program-csomag jelleg magyarázza.
- A program egyszer indítható RUN-nal, amikor a szótárt felhordjuk. A lekérdezést RUN 60-nal indítjuk.

ABC: 87 P86-tól függetlenül* OK >RUN-nal* P87-et bármely programba beépíthetjük, ha valamilyen céllal várakoztatjuk a program futását (pl. kazetta csere, döntés a menüből való választásról stb.)*

HTZ: 87 Külön* >65 G\$ = INKEY\$: IF G\$ < "A" THEN 65* >66 IF G\$ = "N" THEN END* >67 PRINT "SZEP"* >RUN* Bármely programba beépítve szabályozhatjuk a program futását*

TEXAS: 87 >65 INPUT G\$ (:GET és :IF törölve)* A GET utasítás hiányzik. A teljes 65-öt INPUT G\$ helyettesíti*

PRIMO: 87 >65 INPUT G\$: IF G\$ = "N" THEN END* >66 törölve* Az egyszerű GET utasítás hiányzik a gép BASIC-jéből* Próbálja ki a >65 G\$= "" : G\$= INKEY\$: IF G\$ < "A" THEN 65 utasításokat*

TV COM: 87 >65 G\$= "" : GET G\$: IF (G\$= "") THEN 65* >66 IF (G\$="N") THEN END* Próbálja ki önálló programként, vagy megfelelő címkével bármely program részeként*

VT--16: 87 >65 G\$= "" : G\$=INKEY\$: IF G\$= "" THEN 65* KEY=billentyű*

Egészítse ki a P86-ot futás-szabályozó utasításokkal:

```
>LOAD "P86", 8 : LIST-100
```

```
>63 PRINT "HA EGY BETUT LEUT, AKKOR RUN 65, HA AZ
      N-ET, AKKOR END KOVETKEZIK"
```

```
>65G$=" " : GET G$ : IF G$ = " " THEN 65
```

```
>66 IF G$=" N" THEN END
```

```
>RUN 60 → [ ] ^
```

△△

- A >65 egy, a G\$-től függő ciklusba vezeti a gépet. Ha G\$ üres, akkor >65 végrehajtása ismétlődik. A GET olyan értékadó utasítás, amely a klaviatúra bármely gombjának (GET G számjegyre; GET G\$ egyébre) a leütésével teljesíthető. Hatására a G, illetve G\$ értéke a leütött karakterrel azonosul. Ekkor megszűnik G\$ üres jellege és a gép >66-ra lép. Olyan esetben alkalmazzuk a >65-öt, amikor időre van szükségünk a gép közleményeinek az értelmezésére, lemásolására, tárolók cserélésére stb.
- A >65-höz kötődő >66-ot kiegészítésül adjuk meg olyan esetben, amikor a program ciklus futását a munka befejezésével regulárisan akarjuk leállítani. A RUN/STOP alkalmazása kedvezőtlen a floppy számára, mert a file lezáratlan marad. Erre a célra kitüntetjük valamelyik karaktert (pl. N). Ennek leütése a >65-ből való kilépést és >66-ban az END végrehajtását eredményezi.

ABC: 88 NEM*

HTZ: 88 NEM*

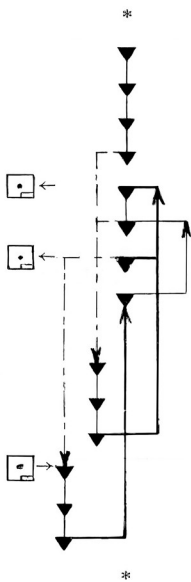
TEXAS: 88 >101 ON ERROR 600* >110 OPEN#88 : "DSK1. DIREKT", **RELATIVE 200, FIXED 254*** >130 törölve* >140 :: GOSUB 510* >150 törölve* >160 : CLOSE15 törölve* >500 **PRINT#88, REC 200 : -1** :: RETURN* >510 **PRINT#88, REC J : -1*** >600 CALL ERR (W, X, Y, Z)* >601 PRINT W; X; Y; Z* A megnyitásban szerepel a (**RELATIVE**) jelleg, a (**200**) rekordszám, a szóbjöhető (**FIXED**) rekord típus és a (**254 BYTE**) rekordhossz*

PRIMO: 88 >101 törölve* >110 **CMDDEF5, "DIREKT, L,"** + CHR\$(254)* >130 **CMDPRINT#5, -1*** >135 **CMDCLOSE5*** >140 FOR I=1 TO 199: **CMDDEF5, "DIREKT"** : GOSUB 500* >160 **CMDCLOSE5** : NEXT I* >510 **CMDPRINT "P"** + CHR\$(5) + CHR\$(L) + CHR\$(H) + CHR\$(1)* >600 **CMDERROR EN, EM\$, ET, ES*** A parancs csatornát nem kell megnyitni és lezárni. A **>CMDPRINT** utasítás automatikusan a **15-ös** csatornát veszi igénybe*

TV COM: 88 NEM*

VT-16: 88 >101 ON ERROR GOTO 600 : **DIM F\$(200)*** >110 **OPEN "R", #1, "RELATIV", 154*** A rekord logikai száma **relatív file esetén csak 1-3** (≤3 file lehet nyitva egyidőben), a rekordhossz csak **1-180** (később kiterjesztjük) lehet* >130 **PUT#1, I: PUFFERBÖL LEMEZRE VISZI AZ I-EDIK REKORDOT*** >150 **PUT#1, I*** >160 NEXT I : **CLOSE#1 : END*** >500 **FIELD#1, 154 AS F\$(I)*** >510 **LSET F\$(I) = MKIS(-1) :** ' A REKORD ADATAI SZÁMÁRA HELYET FOGLAL A PUFFERBEN*

Írjon alkalmazási programot relatív file disk-en történő telepítésére:



```

>NEW
>100 PRINT "***RELATIV FILE TELEPITESE***"
>101 OPEN15,8,15
>110 OPEN88,8,5,"DIREKT,L,"+CHR$(254)
>120 I=200 :GOSUB 500
>130 PRINT#88,-1
>140 FOR I=1 TO 199 :GOSUB 500
>150 PRINT#88,-1 :GOSUB 600
>160 NEXT I :CLOSE88 :CLOSE15 :END
>170 REM * A SZERVIZ RUTINOK *
>500 H = INT(I/256) : L = I - H * 256
>510 PRINT#15, "P"+CHR$(5)+CHR$(L)+CHR$(H)+CHR$(1)
>520 RETURN
>600 INPUT#15, EN, EM$, ET, ES
>601 IF EN <>0 THEN PRINT EN, EM$, ET, ES
>610 RETURN
>RUN → [ ] ^

```

△△

- A relatív vagy direkt elérésű file soros elrendezését illetően olyan, mint a szekvenciális file, azzal az eltéréssel, hogy ismerjük a rekordok fekvését, tehát olyan koordinátákat, amelyek alapján a rekordok bármelyike közvetlenül felkereshető anélkül, hogy a sorban előzőket letapogatónánk.
- A relatív file telepítése a parancs csatorna megnyitásával kezdődik. A file megnyitása két dologban tér el a szekvenciális file megnyitásától. Itt egy L betű (>110) közli a géppel (operációs rendszerével), hogy relatív file megnyitását kezdeményezzük. a

ABC: —

HTZ: —

TEXAS: Az INPUT-OUTPUT elhagyása mindkettő működését jelenti. A PRINT#-ben a **REC J** a pointer állító*

PRIMO: —

TV COM: —

VT-16: 88 A string-gé konvertált adatokat, balra igazítva (LSET) helyezi a pufferbe* >600 PRINT "HIBAKOD" ERR, "HIBACIMKE" ERL, "JAVITSON" : STOP* >601 törölve* Relatív file telepítése kétlépcsős művelet:
REKORD → **PUFFER** → **LEMEZ*** Ilyen file-ba csak string telepíthető. A számokat (egész, E pontos, D pontos) az **MKIS** (), **MKSS** (), **MKDS** () függvények konvertálják*

CHR\$(254) pedig a tervezett rekordkép helyigényét (karakterszám) rögzíti. A legkisebb rekordkép1 (CHR\$(1)), a legnagyobb 254 karakterből állhat. A szóköz is karakternek számít.

- A rekordok számát előre meg kell adni. Ha üres disk-et használunk és 254 karakterből álló rekordokkal kell dolgoznunk, akkor legfeljebb 664 rekorddal számolhatunk. A két paraméter fordított tendenciában mozog. A file túlméretezése hibajellel párosul.
- A file telepítését a megnyitással, majd az utolsó (példánkban 2000-adik) rekord letelepítésével indítjuk. Deklaráljuk a sorszámát (I=2000) majd kiszámíttatjuk a helyzetjelző (pointer) disk-re vonatkozó pozícióját. Erre szolgál az >500 és >510 utasítás. A H és L segédparaméterek láthatóan a rekord sorszámától (I) és a disk formátumától függenek. Az >510 a parancs (15) csatornát használva beállítja az írókart a kiszámított rekordterület 1-es pozíciójába, hogy az 5-ös (CHR\$(5)) csatornán érkező parancsokat végrehajtsa. A >130 utasítás –1-el foglalja le az utolsó rekord területét. Ezzel a file terület lefoglalása befejeződött. A file belső rekordjainak hasonló eljárással foglalunk helyet. A már biztosított területen. A foglalást FOR ciklussal vezéreljük. Kedvező, hogy a pointer állásokat nem kell feljegyeznünk, mert az >500 és >510 utasítások bármikor rekonstruálják azokat, ha a sorszámot (I) közöljük.
- A disk műveletek zavarait a szokásos módon ellenőrizhetjük.

ABC: [89] NEM*

HTZ: [89] NEM*

TEXAS: [89] >31Ø OPEN#88 : "DSK1. DIREKT", RELATIVE 2ØØ, FIXED 254*
>33Ø :: GOSUB 515* >34Ø törölve* >36Ø NEXT I :: CLOSE#88 :: END*
>515 INPUT#88, REC I : A\$* A lekérdezésnél az INPUT# alatt állítjuk a
pointert (a keresett rekord indexe alapján)*

PRIMO: [89] >31Ø törölve* >33Ø FOR I=1 TO R : CMDDEF5, "DIREKT" : GOSUB
5ØØ* >34Ø CMDINPUT#5, A\$: GOSUB 6ØØ* >36Ø CMDCLOSE5 : NEXT
: END* A file-t >CMDKILL "RELATIV" [RET] utasítással törölhetjük a
disk-ről*

TV COM: [89] NEM*

VT-16: [89] >31Ø OPEN "R", #2, "RELATIV", 154* >33Ø : GOSUB 8ØØ* >34Ø
PRINT A\$: A% = CVI(A\$) : ' PUFFERBŐL OLVAS EGY 154 KARAKTERES
STRING-ET, MAJD EGÉSZ SZÁMMÁ FORDITJA* >35Ø PRINT A% : ' KÉP-
ERNYŐRE IR* >36Ø NEXT : CLOSE#2 : END* >8ØØ FIELD#2,154 AS
A\$* >81Ø GET#2,I : ' LEMEZRŐL PUFFERBE TÖLTI AZ I-EDIK REKOR-
DOT* >82Ø RETURN* Relatív file betöltése kétlépcsős művelet: LEMEZ-
PUFFER-REKORD*

```

*      >LOAD "P88", 8 : LIST

      >300 PRINT "*** A RELATIV FILE LEKERDEZESE***"
      >310 OPEN15,8,15 : OPEN88,8,5, "DIREKT"
      >320 INPUT "HANY REKORDRA KIVANCSI?" ; R
      >330 FOR I=1 TO R : GOSUB 500
      >340 INPUT#88, A$ : GOSUB 600
      >350 PRINT A$
      >360 NEXT : CLOSE88 : CLOSE15 : END
*      >RUN 300      →   Δ

```

- △△
- Ha P88-al 200 rekordból álló relatív file-t telepített, akkor $R \leq 200$ rekordot kérdezhet le. A file megnyitása egyszerűsödik, amennyiben az "L" és a CHR\$(254) elmarad a >310-ből. A gép a megnyitáskor feljegyezte, hogy a 88-as "DIREKT" file relatív (REL) típusú és 254 karakterből állnak rekordjai.
 - A rekordok betöltését a pointer állások kiszámítása előzi meg, (GOSUB 500). A betöltő utasítás (>340) látszólag azonos a szekvenciális file betöltésével használatossal. Meglepő eltérés, hogy a relatív file rekordjai karakter konstansként viselkednek és így A\$-el azonosíthatók.
 - Korábban a parancs (15) csatornát csak akkor nyitottuk meg, ha ellenőrizni akar-

ABC: NEM*

HTZ: NEM*

TEXAS: A relatív file megnyitása OUTPUT és INPUT esetén azonos. A file-szám és rekord-hossz változtatható (most 200 és 45. Nagyságukat tervezzük az információk száma és 1 információ hossza alapján)*

PRIMO: >43 INPUT X\$* >110 CMDDEF5, 'SZEMELYZETI, L, '+CHR\$(45)*

TV COM: -

VT-16: A betöltött string-eket vissza kell konvertálni számokká a CVI (), CVS (), CVD () függvényekkel* Wfloppyra OK*

>4 DEFINT B, G : DEFSNG M : 'ÉRTELEMSZERŰEN*

tuk a disk műveleteket. Relatív file esetén a 15-ös csatornának funkcionális szerepe van. A pointer állításában vesz részt, tehát megnyitása nélkül nem boldogolunk.

- A RUN300 az ismételt file telepítést hivatott elkerülni, jöllehet az egyező paraméterek (rekordhossz, file hossz, név) mellett megismételt telepítés észrevehető változást nem idéz elő a disk-en. Más a helyzet ha a rekordhosszt módosítjuk. Ezt a floppy visszautasítja és hibajelet ad, miközben a file érintetlen marad. Ha ugyanakkor a nevet is megváltoztatjuk, akkor új file telepítése veszi kezdetét, feltéve, hogy van szabad terület. Az egyetlen paraméter a file hossz, ami szabadon növelhető újratelepítéssel. Az eljárás nem érinti a már feltöltött rekordok tartalmát.

90

Egészítse ki a P89-et a telepített file feltöltésére alkalmas olyan programmal, amely személyzeti anyagra – név, végzettség, munkabér, gyerekek száma – specifikált és oldja meg a 3 rutin függetlenítését:

```
* >LOAD "P89", 8 : LIST

>5 PRINT "RELATIV FILE PROGRAM"
>10 PRINT "****SZOLGALTATASAIM****" : PRINT
>20 PRINT "****1**RELATIV FILE TELEPITESE"
>30 PRINT "****2** RELATIV FILE FELTOLTESE"
>40 PRINT "****3** RELATIV FILE LEKERDEZESE"
>41 PRINT "====="
```

ABC: NEM*

HTZ: NEM*

TEXAS: >43 INPUT X\$* >45 ON ERROR 600* >101 törölve* >110 GOSUB 700* >160 : CLOSE15 : END törölve* >170 RETURN* >260 : GOSUB 500 törölve* >270 PRINT#88, REC I : N\$; B; M; G* >270 : GOSUB 600 törölve* >280 : CLOSE15 törölve* >340 törölve* >360 :: END törölve* >510 :: RETURN* >700 OPEN#88 : 'DSK1. SZEM', RELATIVE 200, FIXED 45*

PRIMO: >160 CMD CLOSE5 : NEXT* >210 törölve* >250 FOR J=1 TO R : CMDDEF5, "SZEMELYZETI"* >270 CMDPRINT#5, I, N\$, B, M, G : * >280 CMDCLOSE5 : NEXTJ* >310 törölve* >340 CMDINPUT#5, I, N\$, B, M, G : * >350 PRINT I; N\$; B; M; G* >360 CMDCLOSE5 : NEXT* >700, 710 törölve*

TV COM: NEM*

VT-16: >43 : X\$ = INKEY\$: * >110 GOSUB 900* >160 NEXT : CLOSE#1* >210 GOSUB 900* >260 I=J : GOSUB 700 >270 PUT# 1, I* >280 NEXT J : CLOSE#1* >310 GOSUB 900* >340 B=CVI (B\$) : M=CVS(M\$) : G=CVI (G\$) : ? STRING-SZAM KONVERZIO* >350 PRINT N1\$; B, M, G* >360 NEXT : CLOSE#1* >500 FIELD#1,45 AS F\$ (I) : *

```

>42 PRINT "N" LEUTESE : END ; MAS BETUE : RUN"
>43 X$=" " : GET X$ : IF X$=" " THEN 43
>44 IF X$="N" THEN END
>50 INPUT "MELYIK SZOLGALTATAST KERI?" ; V
>60 ON V GOSUB 100, 200, 300
>70 GOTO 10
>110 OPEN88,8,5, "SZEMELYZETI, L, "+CHR$(45)
>160 NEXT I : CLOSE88 : CLOSE 15
>170 RETURN

```

```

>200 PRINT "****RELATIV FILE FELTOLTESE****"

```

```

>210 GOSUB 700

```

```

>220 PRINT "VEGZETTSEG AZONOSITOK : 1=ALT. ISK.
, 2=SZAKM, 3=ERETT, 4=FOISK, 5=EGYETEM"

```

```

>230 PRINT "VALTOZOK : N$=NEV, B=VEGZETTSEG,
M=MUNKABER, G=GYEREKEK SZAMA"

```

```

>240 INPUT "A DOLGOZÓK SZAMA?" ; R

```

```

>250 FOR J=1 TO R

```

```

>251 INPUT "A KOVETKEZO DOLGOZO SZEMELYI ADATAI
?" ; N$, B, M, G

```

```

>260 I=J : GOSUB 500

```

```

>270 PRINT#88, I; N$; B; M; G : GOSUB 600

```

```

>280 NEXT J : CLOSE88 : CLOSE 15

```

```

>290 RETURN

```

```

>310 GOSUB 700

```

```

>340 INPUT#88, A$ : GOSUB 600

```

```

>350 PRINT A$

```

```

>360 NEXT : CLOSE88 : CLOSE 15

```

```

>370 RETURN

```

```

>700 OPEN15,8,15 : OPEN88,8,5, "SZEMELYZETI"

```

```

>710 RETURN

```

```

>RUN →  Λ

```

*

ABC: [91] NEM*

HTZ: [91] NEM*

TEXAS: [91] >45 INPUT X\$*

PRIMO: [91] >45 INPUT X\$*

TV COM: [91] NEM*

VT-16: >700 FIELD#1, 30 AS N1\$, 3 AS B\$, 9 AS A\$, M\$, 3 AS G\$: LSET N1\$=N\$
: LSET B\$=MKI\$(B) : LSET M\$=MK\$\$(M) : LSET G\$=MKI\$(G)*
>800 FIELD#1, 30 AS N1\$, 3 AS B\$, 9 AS M\$, 3 AS G\$* >900 OPEN "R",
#1, "SZEMELYZETI", 45 : RETURN* Wfloppyra OK*

△△

- A program >5->70 utasításai a rutinok függetlenítését, a >200->290 pedig a szerkezeti file rögzítését szolgálják. A P89 egyes utasításait érinti a file címváltozása, a rekordhossz, CHR\$(45) módosulása és a függetlenséggel kapcsolatos vezérlés. Új elem a >700, amely szubrutinként biztosítja a file munkára való megnyitását. A >340-et csak annak rögzítésére írtuk ki, hogy a rekordok tartalmától független az INPUT# A\$. Ez a betöltési mód nem jelenti azt, hogy a rekordok a gépben karakter konstansként viselkednének.
- A P90 tetszőleges tartalmú relatív file alapítására és lekérdezésére alkalmas, feltéve egyes utasítások módosítását. Erre szorul a >110, 120 és 700 (nev, file és rekordhossz), valamint a >200->300 utasítások.
- A rögzítés érdekessége (>270), hogy I értékét a ciklus biztosítja. A pontosvessző tömör rekordképet generál.

91 Egészítse ki a P90 programot rekordok törlésére és új rekordok rögzítésére alkalmas rutinokkal:

```
*  
>LOAD "P90",8 : LIST  
>42 PRINT "****4**DOLGOZO TORLESE"  
>43 PRINT "****5** UJ DOLGOZO FELVETELE"  
>44 PRINT "*N* LEUTESE : END ; MAS BETUE : RUN"  
>45 X$="" : GETX$ : IF X$="" THEN 45  
>46 IF X$= "N" THEN END
```

ABC: -

HTZ: -

TEXAS: [91] >130 törölve* >140 :: GOSUB 510* >415 I=K :: GOSUB 515* >420
törölve* >425 törölve* >430 PRINT A\$:: J=I* >435 GOSUB 510* >440
törölve* >445 CLOSE#88 :: RETURN* >475 törölve* >480 PRINT#88,
REC J : N\$; B; M; G *

PRIMO: [91] >140 FOR I=1 TO R-1 : CMDDEF5, "SZEMELYZETI" : GOSUB 500*
>340 CMDINPUT#5, A1\$, A2\$, A3\$, A4\$* >350 PRINT A1\$; A2\$; A3\$;
A4\$* >410 CMDDEF5, "SZEMELYZETI"* >420 CMDINPUT#5, A1\$,
A2\$, A3\$, A4\$* >430 PRINT A1\$; A2\$; A3\$; A4\$* >470 CMDDEF5,
"SZEMELYZETI"* >480 CMDPRINT#5, I, N\$, B, M, G : GOSUB 600*

TV COM: -


VT-16: [91] >130 PUT#1, I* >410 GOSUB 900* >415 I=K : GOSUB 800* >420
B=CVI (B\$) : M=CVS (M\$) : G=CVI (G\$)* >425 és 435 törölve* >430 PRINT
N1\$; B, M, G* >440 N\$=" " : B=0 : M=0 : G=0 : GOSUB 700* >441 PUT#1, I*
>445 CLOSE#1 : RETURN* >470 GOSUB 900* >475 I=J : GOSUB 700*
>480 PUT#1, I* >485 CLOSE#1 : RETURN* Wfloppya OK*

```

>60 ON V GOSUB 100, 200, 300, 400, 450
>111 INPUT "A FILE HOSSZA? "; R
>120 I=R : GOSUB 500
>130 PRINT#88, -1
>140 FOR I=1 TO R-1 : GOSUB 500

>400 PRINT "***DOLGOZO TORLESE***"
>405 INPUT "A TORLENDO NEV INDEXE? "; K
>410 GOSUB 700
>415 I=K : GOSUB 500
>420 INPUT#88, A$
>425 GOSUB 600
>430 PRINT A$
>435 GOSUB 500
>440 I=0 : N$ = "" : B=0 : M=0 : G=0 : PRINT#
      88,I; N$; B; M; G
>445 GLOSE88 : CLOSE15 : RETURN

>450 PRINT "***UJ DOLGOZO FELVETELE***"
>455 PRINT "NEV, VEGZETTSEG, BER, GYEREK"
>460 INPUT "A DOLGOZO SORSZAMA?"; J
>465 INPUT "A DOLGOZO: "; N$, B, M, G
>470 GOSUB 700
>475 I=J : GOSUB 500
>480 PRINT#88, I; N$; B; M; G : GOSUB 600
>485 CLOSE88 : CLOSE15 : RETURN

* >RUN →  Λ

```

△△

– A két új szolgáltatás miatt átszervezzük a P90 >40–>50 utasításait és kibővítjük a >60 hatókörét. Az általános érvényű programot megközelítendő, paraméterként kezeljük a file hosszát (>111).

ABC: NEM*

HTZ: NEM*

TEXAS: >330 FOR I=K TO K + R - 1 :: GOSUB 515*

PRIMO: >250 FOR J=1 TO 4*R STEP 4 : CMDDEF5, "SZEMELYZETI"* >330
FOR I=K TO K+R - 1 : CMDDEF5, "SZEMELYZETI" : GOSUB 500*
Az üres (-1-el foglalt) rekordokat a program nem tölti be*

TV COM: NEM*

VT-16: >330 FOR=I=K TO K+R - 1 : GOSUB 800* Próba után állítsuk vissza
>250 FOR J=1 TO R alakba*
A file-ok törlése : >KILL "SZEMELYZETI" *
A törlés ellenőrzése: >FILES * Wfloppyra OK*


- A rekord törlésére szolgáló rutin erősen hasonlít a 3-as (relatív file lekérdezése) rutinra, csak rekord törlő és az üres rekordot rögzítő (>440) utasítással kel kiegészíteni.
- Új rekordot bármikor felvihetünk a korábban telepített direkt file-ra, akár valamely rekordot felülírva, akár az utolsó feltöltött rekordot követő helyre, sőt a file határt áttörve APPEND jelleggel is. A realizáció az index (>460) megadásától függ. Előzetesen tájékozódniuk kell a file képéről a 3-as rutin bevetésével, nehogy fontos információt felülírjunk. Sok nyilvántartásnál számolni kell utólagos beszúrásokkal. Erre akkor van lehetőség, ha az első feltöltésnél üres rekordokat biztosítunk, vagyis hézagosan töltünk.
- A 3-as rutin kedvezőbben használható, ha alkalmas a file egyes részeinek a lekérdezésére is.

92

Módosítsa a P91-et oly módon, hogy a 2-es rutin 4–4 hézagot biztosítson a feltöltött file-ban, a 3. rutin pedig a file bármely szeletét képes legyen betölteni:

```

* >LOAD "P91", 8 : LIST 250 : LIST 300-330

>250 FOR J=1 TO 4*R STEP 4
>320 PRINT "EGY REKORD, REKORDSOR, ES A TELJES FILE IS
LEKERDEZHETO!"
>325 INPUT "A REKORDSOR ELSO INDEXE : K, ES
TAGSZAMA : R ?" ; K, R
>330 FOR I=K TO K+R-1 : GOSUB 500
* >RUN →  Λ

```

ABC: [93] NEM*

HTZ: [93] NEM*

TEXAS: [93] >210;>280;>281 marad [92] szerint* >330;>331 marad [92] szerint* >350 PRINT A\$: IF I=K+R THEN 361* >360;>361 marad [92] szerint*

PRIMO: [93] >DELETE 20* >DELETE 100-170* >210 CMDDEF5, "SZEMELYZETI, L," + CHR\$(45)* >250 FOR J=1 TO R* >330 FOR I=K TO K+R - 1 : CMDDEF5, "SZEMELYZETI"* >350 PRINT A1\$; A2\$; A3\$; A4\$* >360 CMD CLOSE5 : NEXT* >361 törölve*

TV COM: [93] NEM*

VT-16: [93] >20 [ENT]* DELETE 100-170 [ENT]* >210 GOSUB 900* >280 NEXT J : CLOSE#1* >281 törölve* >330 FOR I=K TO K+R - 1 : GOSUB 800* >331 törölve* >350 PRINT N1\$; B, M, G* >360 NEXT : CLOSE#1* >361 törölve*

△△

- Ha a file feltöltését FOR-TO-STEP α -NEXT szervezéssel oldjuk meg, akkor két töltött rekord között α számú rekord hely marad üresen. Ezek később az 5-ös rutinnal tölthetők fel. Ha a ciklus kezdőpontjával és a STEP lépésközzel manőverezünk, tet-szőleges file feltöltést érhetünk el.
- A teljes file lekérdezése a K=1 és a file hossz, egy szeleté a kezdő index és a szelet rekord száma, egy rekordé pedig a rekord index és R=1 megadásával érhető el.

93

Törölje a P92-ből a file telepítésére vonatkozó utasításokat és egészítse ki relatív file közvetlen rögzítésére alkalmas programmá:

```
* >LOAD "P92", 8 : LIST -200

>20, 100, 101, 110, 111, 120, 130, 140, 150, 160, 170

>210 OPEN15,8,15 : OPEN88,8,5, "SZEMELYZETI,L,"+CHR$(45)
>280 NEXT J
>281 CLOSE88 : CLOSE15
>330 FOR I=K TO K+R-1
>331 GOSUB 500
>350 PRINT A$ : IF I=K+R THEN 361
>360 NEXT
>361 CLOSE88 : CLOSE15
* >RUN → [ ] Λ
```

ABC: -

HTZ: -

TEXAS: -

PRIMO: -

TV COM: -

VT-16: Terjessze ki a megnyitható file-ok számát **15**-re és a rekord-hosszt **512**-re :
>SYSTEM A gép a VT BASIC betöltésére kész állapotba kerül*
>VT BASIC B : P93 /F : 15 /S : 512 * Az A floppy-ban elhelyezett
VT BASIC lemez betöltődik és a B-ben lévő P93 program RUN parancsot kap.
OK után módosíthatjuk a rekord-hosszt, a logikai számot, a file nevét stb.*

△△

- A közvetlen alapítású relatív file számos gyengeséget mutat a telepítettel szemben. Nyitottsága gondot okoz a lekérdezésnél, ha nem tudjuk az utolsó rekord indexét. Túl-lépés esetén hibajeleket kapunk és a lekérdezés az utolsó rekordot követő 2–5-ik üres rekordnál megáll. Nem biztosít puffer területet az utolsó rekord mögött, így a későbbi APPEND, területileg elszakadhat a file-tól, ami lassítja a file kezelését. A gép (file kezelő rendszere) nehezen boldogul az utasítás sorokkal. Ezért bontottuk le a >280, 330, >350 és >360 utasításokat. Ilyen lebontással kivédhetők a virtuális hibajeleket, amik főként a rekordok rögzítését kísérik. A hibajeleket értékéről lekérdezéssel tájékozódhatunk.
- A file felülírását és módosítását illetően ugyanazt mondhatjuk el, mint a telepített filenél. Megjegyezzük, hogy a közvetlen alapítású file is telepítetté válik, ha a feltöltés után alkalmazzuk az 5-ös rutint és egy fiktív rekordot töltünk be egy meghatározott méretű tartalék terület mögé. Ezzel a file védett területét kiterjesztjük. Ez természetesen felesleges, ha egy disk-et egyetlen file tárolására használunk, ami ritkán fordul elő.

046

File-ok konvertálása.

94

Irjon másoló programot, amely szekvenciális file-t relatív file-ba visz át. Specifikálja a programot a "LOTTO" c. file-ra. Ellenőrizze az eredményt:

ABC: 94 NEM*

HTZ: 94 NEM*

TEXAS: 94 >3Ø OPEN#77 : "DSK1. LOTTO", SEQUENTIAL, INPUT, VARIABLE
5* >4Ø :: INPUT#77 : A :: D(K)=A :: IF EOF(77) THEN 4Ø* >9Ø OPEN#94 :
"DSK1. RLOTTO", RELATIVE K, FIXED 3* >11Ø, >12Ø törölve* >13Ø
PRINT#94, REC I : D(I) :: * >15Ø CLOSE#94 :: END*

PRIMO: 94 >3Ø CMDOPENS, "LOTTO" : * >4Ø K=K+1 : CMDINPUT#5, D : D(K)=D
: IF NOT EOF(5) THEN 4Ø* >9Ø CMDDEF6, "RLOTTO, L, " + CHR\$(3)*
94/B >25 FOR I=1 TO N : CMDDEF3, "RLOTTO"* >3Ø törölve* >6Ø
CMD INPUT#3, Q\$: PRINT Q\$* >7Ø CMDCLOSE3 : NEXT : END*

TV COM: 94 NEM*

VT-16: A file újra alapítható. A programban 15 nyitott file dolgozhat egyidőben, (pl.
törölő, bővítő, karbantartó stb., file-ok)* 94 >2Ø PRINT "LOTTO AZ
A, RLOTTO A *B* floppy-ban"* >3Ø OPEN "I", #1, "LOTTO" : DIM
D(1ØØ), D\$(1ØØ) : GOSUB 9Ø* >4Ø K=K+1 : INPUT#1, D : D(K)=D : GOSUB
11Ø* >5Ø IF NOT EOF(1) THEN 4Ø* >6Ø GOTO 14Ø* >7Ø - >8Ø töröl-
ve* >9Ø OPEN "R", #2, "B : RLOTTO", 3 : RELATIV FILE MEGNYITA-
SA"* >1ØØ RETURN* >11Ø FIELD#2, 3 AS D\$(K)*

ABC: [95] NEM* A programot átírhatja két kazettás egységre és szekvenciális file-ra* A CAS1-et használja a file betöltésére és CAS2-t a rendezett file rögzítésére* Ezesetben >115 OUT 58,16 : OUT 58,24*
[95] A rögzítés ellenőrzéséhez különálló betöltő programot kell írnia, mert a >RUN 120 utasítás nem működik*

HTZ: [95] NEM* Kísérlelje meg két kazettás egységgel a problémát megoldani* Csak szekvenciális rögzítésre gondolhat* A gépet használja a file betöltésére és a CAS2-n végezze a rögzítést*

TEXAS: [94/B] >25 OPEN#94 : "DSK1. RLOTTO", RELATIVE, FIXED* >40;
>50 törölve* Relative file esetén a rekord csak FIXED típusú lehet, tehát jelzése elhagyható. Ez nem gazdaságos a 80 BYTE-os rekord-ok miatt*

PRIMO: —

TV COM: [95] NEM* A gép nem tud stringeket lexikografikusan rendezni. Ha ORD(B\$(1)) <= ORD(B\$(K)) összehasonlítást alkalmazunk, akkor a kezdőbetűk ASC kódjai alapján rendezünk. Ezt alkalmazzuk [95] -ben*

VT-16: >120 LSET D\$(K) = MKI\$(D(K))* >130 PUT#2, K : RETURN* >140 PRINT "N=" K* >150 CLOSE : END*
[94/B] >25 OPEN "R", #3, "RLOTTO", 3* >40 FIELD#3, 3 AS Q\$* >50 GET#3, I* >60 Q%=CVI(Q\$) : PRINT Q%, : NEXT* >70 CLOSE#3 : END*
Wfloppy OK*

△△

- A szekvenciális file-t ST figyeléssel olvastatjuk be a disk-ről és számláljuk (K) a rekordokat. Ezt felhasználjuk a relatív file rögzítésénél (>100) a ciklus vezérlésére.
- A szekvenciális file bázis disk-jét felhasználhatjuk a relatívva konvertált file közvetlen telepítésére. Ellenkező esetben a lemez cserélése időt vesz igénybe. Ezt nagyszámú üres (>70) ciklus beépítésével biztosíthatjuk. A >70-et késleltető utasításnak nevezzük. A kettős disk műveletet ugyanazon a csatornán bonyolíthatjuk, de eltérő számú (5;6) csatorna választása biztonságosabb.
- A konvertálás ellenőrzéséhez a >140 utasítás közli a szekvenciális file rekordszámát, ami egyenlő a relatív file rekordszámával. Ezt a B., >20 INPUT-nál használjuk fel.
- A pointer pozíciószáma nem lehet magasabb a betöltésnél, mint a rögzítésnél (>50; >120).
- A rögzítést ellenőrző program utasításai (B), nagyrészt megtalálhatók a telepítést végző programban is. Célszerű a két programot összedolgozni. Egy lehetséges megoldást bemutatunk a 95-ben.

047

File-ok mozgása háttértárolók között.

95

Olvassa be kazettáról a „NEVSOR” c file-t. Rendezze a neveket A, B, C sorrendben és rögzítse disk-en a NEVSOR c. relatív file-ba. A rögzítést ellenőrizze:

ABC: —

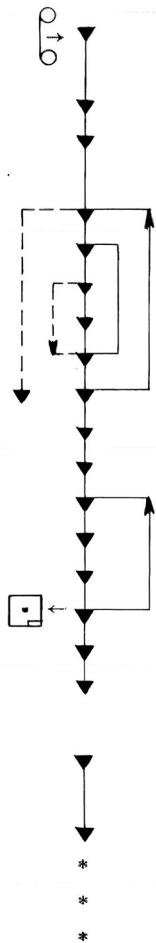
HTZ: —

TEXAS: [95] >30 CLOSE#62 ::* >110 :: BREAK* >120 OPEN#95 : "DSK1. NEV
SOR", RELATIVE, FIXED 30* >140; >150 törölve* >160 PRINT#95,
REC I : B\$(I) :: PRINT B\$(I) ::* >170 PRINT "INPUT#, VAGY PRINT#?
IRJA BE 160 ALA A MEGFELELOT ES RUN 120"* A 170-es címkét nem
tudja törölni* >180 CLOSE#95 :: END* >200; >210 törölve* >RUN
200 törölve*

PRIMO: [95] >110 törölve* >160 CMDPRINT#7, B\$(I) : NEXT* >170 REM >120
INPUT N : FOR I=1 TO N : CMDDEF7, "NEVSOR, L,"* >171 REM >160
CMD INPUT#7, N\$: PRINT N\$* >172 REM 180 CMDCLOSE7 : NEXT : END*
>180 CMDCLOSE7* >200 PRINT "TOROLJE A 170, 171, 172 CIMKEKET
ES AZ UTANUK KOVETKEZO REM SZOKAT (EDIT)"* >201 PRINT"
TOROLJE A >130 UTASITAST, MAJD RUN 120"*

TV COM: [95] >LOAD#5 : "P62A" : LIST* >30 CLOSE INPUT : * >40 : PRINT
"A KEZDOBETUK SZERINT RENDEZEK"* >70 IF ORD(B\$(I)) <= ORD
(B\$(K)) THEN 90* >DELETE 110-210* >RUN* n_k a kazetta nyilvántar-
tásból kikereshető*

VT-16: [95] >10 : DIM B\$(N), D\$(N), K\$(N)* >30 : GOSUB 120* >180 SWAP
B\$(I), B\$(K)* >100 PRINT B\$(I) : GOSUB 140* >110 NEXT I : END* >120
OPEN "R", #2, "NEVSOR", 30* >131 FIELD#2, 30 AS D\$(I) : LSET
D\$(I) = MKI\$(-1) : PUT#2, I : NEXT : CLOSE#2 : RETURN* >140 OPEN
"R", #3, "NEVSOR", 30 : FIELD#3, 30 AS K\$(I) : LSET K\$(I) = B\$(I) :
PUT#3, I : CLOSE#3 : RETURN* >150- >200 törölve. Betöltés P95A-val*
>RUN*



>LOAD "P62/A",8 : LIST

>30 CLOSE15 : PRINT "A NEVEK BETOLTESE KESZ"

>40 PRINT : PRINT "***LEXIKOGRAFIKUSAN
RENDEZEK***"

>50 FOR I=1 TO N : IF I=N THEN 100

>60 FOR K=I+1 TO N

>70 IF B\$(I) <= B\$(K) THEN 90

>80 S\$=B\$(I) : B\$(I)=B\$(K) : B\$(K)=S\$

>90 NEXT K

>100 PRINT B\$(I) : NEXT I

>110 PRINT "TEGYEN DISZK-ET A FLOPPY-BA!" : STOP

>120 OPEN15,8,15 : OPEN95,8,7, "NEVSOR, L, " + CHR\$(30)

>130 FOR I=1 TO N

>140 H=INT (I/256) : L=I-H*256

>150 PRINT#15, "P" + CHR\$(7) + CHR\$(L) + CHR\$(H) + CHR\$(4)

>160 PRINT#95, N\$(I) : NEXT

>170 REM 160 INPUT#95, N\$: PRINT N\$: NEXT

>180 CLOSE95 : CLOSE15 : END

>200 PRINT "TOROLJE A 170-es CIMKET ES A REM-ET
(LIST170 ; TORLES), MAJD RUN 120"

>210 END

>RUN → ^

>RUN 200

>RUN 120

△△

– A 62/A programból indulunk ki. Töröljük az END-et, majd rendező (>50→100), rögzítő (>120→180) és átalakító programot fűzünk hozzá.

ABC: —

HTZ: —

TEXAS: —

PRIMO: —

TV COM: —

VT-16: 95/A >1Ø INPUT "NEVEK SZAMA?"; N : OPEN "R", #2, "NEVSOR", 3Ø*
>2Ø FOR I=1 TO N* >3Ø FIELD#2, 3Ø AS A\$* >4Ø GET#2, I : PRINT A\$:
NEXT* >5Ø CLOSE#2 : END* >RUN*

- A [77]-ben megírt számrendező program stringekre is érvényes. Az IF bevezetése az >50 alatt csak arra szolgál, hogy más változatot is bemutassunk a rendezésre.
- A rögzítő programunk relatív file-t alapít. A pointer pozíciót 4-nek adtuk meg. Ez pazarlás a tároló kapacitással.
- A >170 REM egy utasítást hordoz, ami a 170 és a REM törlése után >160-as utasításként lép fel, törölve a PRINT#95, N\$(I) :NEXT utasítást. A >200 arra szólít fel, hogy ezzel a törléssel a >120->180 rögzítő programot betöltő programmá konvertálja és RUN 120-al aktiválja.
- A program működése: RUN elindítja a nevek kazettáról való betöltését, névsort állít elő majd disk-en rögzíti. REDY után RUN 200 felvilágosít a konverzió módjáról. A LIST 170-re kapott programsor címkéjét és a REM kulcsszót törölve RUN 120 betölti a disk-ről a névsort majd leáll. Ismételt futtatás előtt a 160-at eredeti formába állítjuk vissza.
- A [95] programját már korábbról ismert rutinokból állíthattuk össze. Kedvező lenne, ha disk-ről, vagy szalagról behívhatnánk a szükséges rutinokat és programmá szerkeszthetnénk. Sajnos minden LOAD parancs egyben NEW parancs is, ezért a dolog nem megy. Racionális eljárásnak azt tekinthetjük, hogy a legnagyobb méretű rutint betöltjük és a kisebb rutinokat a klaviatúráról hozzáfűzzük. Bizonyos körülmények között azért fogunk tudni programokat egyesíteni. Erre visszatérünk.

ABC: 96 Printer nincsen rendszerezítve* Egyes feladatokat **képernyőre** értelmezve oldunk meg*
 97 NEM*

HTZ: 96 Printer nincsen rendszerezítve* Egyes feladatokat **képernyőre** értelmezve oldunk meg*
 97 NEM*

TEXAS: 96 Printer alig van a hazai állományban* Egyes feladatokat **képernyőre** értelmezve oldunk meg*
 97 NEM*

PRIMO: 96 **DATA COOP-DCD-PRT-42** printer-rel dolgozunk* A/4-es írólapokat használunk, mint az írógépen*
 97 >CMDLOAD "P45" RET* ≧"ON LINE" KÖRMÖK MEGÉRINTÉSE A PRINTER HOMLOKÁN (sárga lámpa világít)* >LLIST RET*

TV COM: 96 **WALTERS MICROSYSTEMS LTD. MODEL WM 2000** printer-rel dolgozunk*
 97 >LOAD#5 : "P45" RET* Kapcsoló (hátsó, balra) ON* SEL* zöld lámpa ég (elől balra)* Ha a PE lámpa ég igazítsa a papíron* >LLIST RET*

VT-16: 96 **WALTERS MICROSYSTEMS LTD. MODEL WM 4000** printer-rel dolgozunk* Papírtovábbító henger állítás 1-re (skála a henger bal végén), ha szimpla papírt fűz be* 97 >LOAD "P45" ENT* Kapcsoló (hátsó balra) ON* SEL* zöld lámpa ég (elől balra)* Ha a PE lámpa ég, igazítsa a papíron* >LLIST ENT* Ha ERROR lámpa ég SEL* , RESET*

4. A PRINTER KEZELÉSE, PROGRAMOZÁSA ÉS FELHASZNÁLÁSA.

96 Kösse össze a printer-t a számítógép rendszerrel. Tűzzön be szabványos írópapírt, ellenőrizze az írószalag kazettát (épség, festék, mozgás). Kapcsolja be a rendszert:

048 *Program kinyomtatása. OPEN-CMD-LIST és a PRINT#-CLOSE utasítások használata.*

97 Nyomtassa ki a 45 programot:

```
* >LOAD "P45", 8
^ ← * >OPEN97,4 : CMD97 : LIST RET
5 PRINT "A MASODFOKU EGYENLET MEGOLDASA"
10 INPUT "AZ A,B,C EGYUTTHATOK ERTEKE?":A,B,C
20 LET D=B^2-4*A*C
30 IF D<0 THEN PRINT "NINCSEN VALOS GYOK":GOTO10
40 X1=(-B+SOR(D))/(2*A):X2=(-B-SOR(D))/(2*A)
50 IF X1+X2=-B/A ANDX1*X2=C/A THEN PRINT "X1="X1,"X2="X2:GOTO 70
60 PRINT "KOZELITO MEGOLDAS"X1,X2
70 STOP
80 GOTO 10
90 END
```

READY.

```
* >PRINT#97 : CLOSE97 RET
```

△△ – A printer megnyitása és lezárása a háttértárolókhoz hasonlóan történik. Az OPEN hivatkozik a nyomtatásra tervezett file numerikus azonosítójára (pl. 97) és a printer hívószámára, ami mindig 4. A CMD a programot egy tárolóba viszi, melynek a LIST

ABC: NEM*

HTZ: NEM*

TEXAS: NEM*

PRIMO: >CMD LOAD "\$" * >LLIST * A printer-t nem kell megnyitni és lezárni* Ha a papír elfogy RESET-tel állítjuk meg a listázást és >LLIST C1-el folytatjuk, ha a C1 címke előtt állt le a nyomtatás*

TV COM: NEM*

VT-16: A printer-t nem kell megnyitni és lezárni. Az LLIST mindkét funkciót ellátja* >FILES * * A PRTS a képernyő tartalmát viszi printer-re. Ha a disk tartalomjegyzéke nem fér el a képernyőn, akkor a gombokkal a listát megszakítjuk, printeljük, majd folytatjuk a listázást és ismét printelünk stb.* Ismétlés elkerülhető, ha a képet feltoljuk -el*

nyitja ki a zárját és a tároló tartalma a nyomtatóra sorolódik. Meglepő, hogy a lezárást egy üres PRINT#-nek kell megelőznie. A printert a CLOSE bocsátja el, amikor a vezérlést ismét a gép veszi át. Ha elmulasztjuk a lezárást a rendszer figyelmeztet. A floppy például vörös villogással válaszol minden megnyitási kísérletre. Ilyenkor a PRINT#97 : CLOSE97, majd az OPEN15,8, 15, "1" adja vissza a rendszer munkaképességét. A lezárás elmulasztását jelzi az is, hogy a gépnek szánt LIST utasítást a printer hajtja végre, vagyis képernyő lista helyett nyomtatványt kapunk.

- A printer 80 karaktert írhat egy sorba, így a program utasításai egy-egy sort foglalnak le. A sorváltás automatikus, de jelekkel is szabályozható.
- A program egésze helyett (LIST), bármely összefüggő részlete is kinyomtatható. Ekkor a LIST C1-C2 utasítást adjuk, ahol a C1, illetve C2 címke el is maradhat.

98

Nyomtassa ki egy disk tartalomjegyzékét:

```

* >LOAD "$", 8
^← * >OPEN4,4 : CMD4 : LIST

0 "P35" PRG
1 "P35" PRG
1 "P37" PRG
1 "P39" PRG
1 "P40/B" PRG
1 "P40/A" PRG
1 "P41" PRG
1 "P43" PRG
1 "P44" PRG
656 BLOCKS FREE.

READY.

* >PRINT#4 : CLOSE4

```

ABC: NEM *

NEM *

HTZ: NEM *

NEM *

TEXAS: NEM *

NEM *

PRIMO: >1Ø törölve* >2Ø LPRINT "P45: A MASODFOKU EGYENLET MEGOLDASA." : LPRINT* >3Ø END* >ON LINE MEGÉRINTÉSE* >RUN* >ON LINE KIKAPCSOLÁSA (megérintés)* >CMDLOAD "P45"* >ON L* >LLIST*

OK *

TV COM: >1Ø, >3Ø törölve* >RUN* >LOAD#5: "P45" * >LLIST
A printer-t nem kell megnyitni és lezárni. Mindkettő hibajelre vezet*

>4 DIM C\$*4Ø* >5 PRINT "A PROGRAM CIME?" : INPUT C\$* >2Ø PRINT#4 : C\$*

VT-16: >1Ø törölve* >2Ø LPRINT "P45- A MASODFOKU EGYENLET MEGOLDASA.", CHR\$(13)* >3Ø törölve* >RUN* >LOAD "P45"

* >LLIST* Másik megoldás: >1Ø OPEN "O", #4, "LPT1:"* LPT1: a printer (mint file) neve* >3Ø CLOSE#4 : *

OK a második megoldásával*

049

ADATOK kinyomtatása. OPEN-PRINT#-CLOSE utasítások.

99

Írjon programot a P45 megcímezésére majd címmel együtt nyomtassa ki újra:

```

*
  >NEW
  >1Ø OPEN4,4
  >2Ø PRINT#4, "P45 : A MASODFOKU EGYENLET MEGOLDASA."
    : PRINT#4
  >3Ø CLOSE4 : END
*
  >RUN

```

```

*
  >LOAD "P45" , 8
  >OPEN4,4 : CMD4 : LIST
  >PRINT#4 : CLOSE4

```

△△

– A cím nem programnak, hanem adatnak számít. Az adatok kinyomtatása címkézett programmal történik. Ebben nem szerepelnek a CMD és LIST utasítások. A nyomtatást a PRINT#4 utasítás vezérli, ami a befoglalt idézőjeles szövegre (később adatlistára is) terjed ki. Az üres PRINT#4 sorok között biztosít a nyomtatványban.

100

Módosítsa a P99-et alkalmazói programmá:

```

>5 INPUT "A PROGRAM CIME?" ; C$
>2Ø PRINT#4, C$
*
>RUN →  Λ

```

←

Folytassa a munkát a **46** feladatnál. A fontosabb programokat nyomtassa ki. Erre a programok alatt megjelenő Λ jel figyelmeztet!!!

ABC: 101 >1Ø törölve* >2Ø PRINT TAB (2Ø) "KOVACS ISTVAN", CHR\$(1Ø)
TAB (2Ø) "NAGYREDE", CHR\$(1Ø) TAB (2Ø) "TAS U 2"* >3Ø PRINT
TAB (34) 5385* A képernyősor utolsó két pozícióját lefedő TAB új sorba viszi át
a számot (string-et nem)*

HTZ: 101 >1Ø törölve* >2Ø;>3Ø PRINT* >4Ø CLOSE4 törölve* A CHR\$(13)
≡ CHR\$(1Ø), tehát CHR\$(13) maradhat a programban*
102 OK*

TEXAS: 101 >1Ø törölve* >2Ø PRINT TAB(1Ø); "KOVACS ISTVAN" :TAB(1Ø);
"NAGYREDE" :TAB(1Ø); "TAS U 2"* >3Ø PRINT TAB(2Ø); 5385* Más
megoldás: >2Ø DISPLAY AT (2Ø, 1Ø) : "KOVACS ISTVAN" : : DISPLAY AT
(22, 1Ø) : "NAGY REDE" :: DISPLAY AT (24, 1Ø) : "TAS U 2" stb.*

PRIMO: 101 OK* LPRINT UTÁN, NEM ÁLL* **PRÓBÁLJA MEG A KÉPERNYÖRE**
ÁTIRNI A PROGRAMOT (PRINT)*
102 OK*

TV COM: 101 >1Ø törölve* >2Ø PRINT#4 : STRING\$(2Ø, " "); "KOVACS ISTVAN"
: LPRINT : PRINT#4 : STRING\$(2Ø, " "); "NAGYREDE" : LPRINT: PRINT#4
:STRING\$(2Ø, " "); "TAS U 2"* >3Ø PRINT#4 : STRING\$(35, " "); 5385*
>4Ø törölve*
102 >4 DIM N\$*2Ø, T\$*2Ø, C\$*4Ø, Y\$*8Ø*

VT-16: 101 OK* **PRÓBÁLJA MEG KÉPERNYÖRE IS ÁTIRNI A PRINTER-RE**
VONATKOZÓ PROGRAMOKAT. Itt elegendő a #4 törlése a >2Ø és >3Ø utasítá-
sokban*
102 OK

Írjon programot egy levélcím nyomtatásához:

```

*
  ▼
Λ ← ▼
    ▼
Λ ← ▼
    ▼
*
  >NEW
  >1Ø OPEN4,4
  >2Ø PRINT#4, TAB(2Ø) "KOVACS ISTVAN" CHR$(13)
    TAB(2Ø) "NAGY REDE" CHR$(13) TAB(2Ø) "TAS U 2"
  >3Ø PRINT#4, TAB(35) 5385
  >4Ø CLOSE4 : END
  >RUN

```

△△

- A TAB az írópapír baloldali margójától számított karakter pozíciót biztosít a szöveg kezdőpontja számára.
- A CHR\$(13) sorváltásra utasítja a printer-t a megelőző szöveg kinyomtatása után.
- Egy printer megnyitásban többezer PRINT# követheti egymást.

Fejlessze a P1Ø1-et alkalmazói programmá:

```

  >5 INPUT "NEV?" ; N$
  >6 INPUT "TELEPULES NEVE? " ; T$
  >7 INPUT "KERULET, UTCA, HAZSZAM " ; C$
  >8 INPUT "IRANYITOSZAM" ; I
  >9 X$=CHR$(13)
Λ ← >2Ø PRINT#4, TAB(2Ø) N$X$TAB(2Ø)T$X$TAB(2Ø)C$
Λ ← >3Ø PRINT#4, TAB(35)I
* >RUN →  Λ

```

△△

- A TAB és a CHR\$ utasítások láthatóan írásjelek (; , ; ,) mellőzésével illeszkednek be az utasításba.

ABC: 103 Kód készlet: 32–127* >20 X X = SPACE X (3) : Y X = SPACE X (1)
:PRINTX X ; Y X ; * >30 : PRINT LEFT X (NUM X (K) + X X , 4) ;
NEXTK : PRINT CHR X (10)* >50 PRINT LEFT X (NUM X (I) + Y X , 3);
Y X ; * >60 FOR J=31 + I TO 111 + I STEP 16* >70 PRINT LEFT X
(NUM X (J) + X X , 4) ; Y X ; * >100 END*

HTZ: 103 Kódkészlet : 32–255* >20 PRINT " " ; * >30 FOR K=1 TO 14 :
PRINT USING "####" ; K ; NEXTK : PRINT CHR\$(13)* >40 FOR I=1 TO
16 : PRINT USING "####" ; I ; * >50 FOR J=31+I TO 239+I STEP 16* >60
PRINT USING "####" ; J ; NEXTJ : PRINT CHR\$(13) : NEXT I : END*

TEXAS: 103 Kódkészlet: 32–127* >20 PRINT USING "####" : " " ; * >30
FOR K=1 TO 6 :: PRINT USING "####" : K ; :: NEXT K* >40 FOR I=1 TO
16 :: PRINT USING "####" : I ; * >50 FOR J=31+I TO 111+I STEP 16* >60
PRINT USING "####" : J* NEXT J :: NEXT I :: END* >80–100 törölve.

PRIMO: 103 Kódkészlet: 1–151* >20 X\$=" " : Y\$=" " : LPRINT X\$ Y\$; * >30
FOR K=1 TO 8 : LPRINT RIGHT\$ (X\$ + STR\$ (K), 4) ; : NEXT K : LPRINT*
>40 FOR I=1 TO 20 : IF I=21 THEN 100* >60 FOR J=I–1 TO 140 + I STEP
20* A printer 42 karakteres*

TV COM: >9 Y\$ = STRING\$ (20, " ")* >20 PRINT#4 : Y\$; N\$: LPRINT : PRINT#4 :
Y\$; T\$: LPRINT : PRINT#4 : Y\$; C\$: LPRINT*
103 >20 PRINT#4 : STRING\$ (3, " ")* >30 FOR K=1 TO 8 : PRINT#4,
USING "####" : K ; : NEXT K : LPRINT* >50 PRINT#4, USING "####" : I ; *
>60 FOR J=31 + I TO 143 + I STEP 16

VT–16: 103 >5 WIDTH "LPT1 : ", 100* A standard 80-ról 100-ra emeljük az egy sor-
ba nyomtatható karakterek számát* >20 : OPEN "O", #4, "LPT1 : " : *
>30 : PRINT#4, USING "####, " ; K ; : * >50 PRINT#4, USING "##, " ; I ; *
>70 PRINT#4, USING "####, " ; J ; * >80 : PRINT#4, CHR\$(13)*

- Ha P62-vel rögzítjük ügyfeleink címjegyzékét (rögzítés kazettára P64, >25-re tekintettel), akkor a P102; >5->8 utasításait a P62/A programmal helyettesíthetjük, feltéve hogy címkéit és paramétereit aktualizáljuk.

050 Számok jobbra igazítása. Az STR\$ függvény használata.

103 Irjon programot, amely egy 16 soros négyszögbe nyomtatja a 0,1,..., 255 számokat (ASC II kód-okat), megszámozva a sorokat és oszlopokat:

```

*
>NEW
>10 PRINT "***ASC II KOD-OK RENDEZESE**"
>20 X$=" " : Y$=" " : OPEN 4,4 : PRINT#4, X$Y$;
>30 FOR K=1 TO 16 : PRINT#4, RIGHT$(X$ + STR$(K), 4);
      : NEXT K : PRINT#4, CHR$(10)
>40 FOR I=1 TO 16 : IF I=17 THEN 100
>50 PRINT#4, TAB(1) RIGHT$( Y$ + STR$(I),2) Y$ ;
>60 FOR J=I-1 TO 239 + I STEP 16
>70 PRINT#4, RIGHT$( X$ + STR$(J), 4) ;
>80 NEXT J : PRINT#4
>90 NEXT I
>100 CLOSE 4 : END
>RUN → [ ] Δ
  
```

- △△
- A BASIC-ben minden karakterhez rendeltek egy-egy kódszámot. Ezt P54-ben ki is használtuk. A kódok 0, 1,..., 255 értéket vesznek fel. Ahhoz, hogy a karakterek és kódok megfelelését áttekinthetően szemléltethessük, a kódokat egy 16*16-os tömbbe rendezzük, majd hasonlóan járunk el a karakterek-

ABC: Számok jobbra tartása nem oldható meg, mert a RIGHT X (A X , 4) a 4. pozíciótól jobbra álló karaktereket választja ki. Így $\text{X} + \text{NUM}(\text{K})$ jobboldalra K növekedésével egyre hosszabb, ha a pozíciószám állandó*

HTZ: $>80 - >100$ törölve* A PRINT USING számok jobbra tartását biztosítja. Lásd a TEXAS regiszterben*

TEXAS: A PRINT USING számok és térközök jobbra tartását biztosítja. Tizedesek esetén kerekít (pl. PRINT USING "#.#": X, ha X=5.38, 5.4-et ad). Helyiérték túlfolyás esetén **nem írja ki az egész számokat** (pl. 15.38, .4-et ad)*

PRIMO: A printer közvetlenül csak a 33–127 kódokat képes azonosítani* Az $>\text{LPRINT CHR}\$(27) \text{ CHR}\$(82) \text{ CHR}\$(3) : \text{LPRINT} " \text{A B C X Y Z a b c x y z} "$ RET a printer grafikai jeleit nyomtatja ki*

TV COM: $>70 \text{ PRINT}\#4, \text{ USING} "####" : \text{J};*$ $>80 : \text{LPRINT}*$ $>100 \text{ END}*$ A PRINT#4, USING számok és térközök jobbratartását biztosítja. A "#.#" string a szám helyét jelöli ki. Ha a szám **balra** túlfolyik, **hibajel**, ha **jobbra**, **kerekítés** áll elő.

VT-16: A PRINT#4, USING számok és térközök jobbra tartását biztosítja. A jegyek helyét a # jellel adjuk meg. Ha több egész jegy van, mint hely, **csontkítás nem áll elő**, de tört jegyeknél **kerekítéssel** számolhatunk Pl. "#.#", és 56.56 esetén 56.6 jelenik meg*

kel is. Erre szükségünk lesz a nyomtatás gyakorlatában.

- Számoszlopok nyomtatása esetén gondoskodni kell az egyező helyiértékek egymás alá kerüléséről. Egész számok esetén az akkor merül fel, ha egy oszlopba eltérő nagyságrendek kerülnek (pl. 1 ; 16). A megoldást jobbra igazításnak nevezik szemben a karakter konstansokkal (pl. névsor), ahol balra igazításról kell általában gondoskodnunk.
- A jobbra igazításhoz deklarálnunk kell egy üres string-et annyi pozícióval, amely fedezi a legkisebb és legnagyobb szám karakterbeli különbségét és az oszlopok között tervezett minimális távolságot. Példánkban az X\$ 3 pozíciós. Ebből 2 a nagyságrend 1 pedig az oszloptávolság miatt került felvételre. Az üres string-et balról hozzáadjuk a számok karakter konstans alakjához, majd az így keletkező string-ek jobboldalát leválasztjuk a RIGHT\$ függvényvel. A leválasztásnál annyi karaktert nevezünk meg, hogy a legnagyobb szám sem csonkuljon és az oszlopközt is hordozza. Az egyenlő mértékű karakter-ek egymás alá kerülve biztosítják a helyiértékek rendezettségét. Ezt alkalmaztuk >30-ban az oszlopok és >50-ben a sorok számozásánál, valamint >70-ben a kódok pozicionálásánál is.
- Külön figyelmet kell fordítani az első oszlopindex elhelyezésére, amit >20-ban az X\$Y\$ előíratásával oldottunk meg. A print-ek végén látható ;-k a sorképzést biztosítják.

ABC: [104] Az ABC [103] programot módosítjuk * >7Ø PRINT Y ☒ ;
CHR ☒ (J); X ☒ ; * Próbálja ki a >7Ø PRINT CHR ☒ (151); Y ☒ ;
CHR ☒ (J); X ☒ ; utasítást is a [104] programban (Az ASC kódok grafikus
megfelelőit kapja) * A CHR ☒ (151) a grafikus kódváltó * Visszaváltóként
a CHR ☒ (135) alkalmazható *

HTZ: [104] A HTZ [103] programot módosítjuk * >6Ø PRINT " " ; CHR\$(J);
: * A 92-es Ő, a 93-as A' és a 94-es az Ű kódja (ékezetes) * Használat: >PRINT
"KAB" CHR\$(93) "T" || KABÁT || *

TEXAS: [104] A TEXAS [103] programot módosítjuk * >6Ø PRINT " " ;
CHR\$(J); :: *

PRIMO: [104] >65 IF J <= 32 OR J > 127 THEN 8Ø * >66 törölve *
>8Ø LPRINT X\$Y\$; *

TV COM: [104] >65,66 törölve * >7Ø PRINT#4 : STRING\$(3, " "); CHR\$(J); *
>75, >85 törölve * >8Ø marad a P1Ø3 szerint * Irja át képernyőre is a prog-
ramot (#4, #4 : és L törlése a printekből) *

VT-16: [104] >66 törölve * >7Ø PRINT#4, X\$Y\$ CHR\$(J); * >85 : PRINT#4,
CHR\$(13) * Irja át képernyőre és futtassa a programot * Vesse össze a prin-
ter-en és a képernyőn keletkező, azonos kódszámú karaktereket? *

- A jobbra igazítás hasonló módon történik olyan valós számok esetén is, amelyekben a tizedesjegyek száma azonos (pl. kerekített). Egyéb esetben a tizedespontok azonos pozicionálását kell megoldani (l: irodalomjegyzék [18]).
- A BASIC kétféle minőségben használja a számokat. A numerikus formát ismerjük, azzal számoltunk eddig is. A másik minőség a karakteres forma, ami szöveggént viselkedik. Ha egy számot idézőjelbe teszünk karakteres formáját kapjuk (pl. 52; "52"). Ha viszont egy numerikus változó felvehető értékeit akarjuk karakteres formára transzformálni, akkor az STR\$ függvényt kell használnunk (pl. I; STR\$(I)). A transzformáció inverzét a VAL (STR\$(I)) függvény képezi, amit idézőjeles esetben is alkalmazhatunk (pl. VAL ("20")=20). Számok karakteres alakjával aritmetika műveletek nem végezhetők az "összeadást" kivéve, de az eredmény itt is eltér a megszokottól, mert pl. "19" + "85" = "1985", ugyanakkor "19" + 85 eredmény nélküli művelet.

051

Kódok és karakterek megfelelése. A CHR\$ függvény használata.

104

Módosítsa a P103-at úgy, hogy az ASCII kódok helyén karaktereik jelenjenek meg a négyesjögben:

```
>10 PRINT "***KODOK ES KARAKTEREK***"
>65 IF J<32 THEN GOTO 80
>66 IF J>127 AND J<160 THEN GOTO 80
>70 PRINT#4, X$ CHR$(J);
```

ABC: -

HTZ: -

TEXAS: -

PRIMO: -

TV COM: -

VT-16: -

```

>75 GOTO 85
>80 PRINT#4,"      ";
>85 NEXT J : PRINT#4
*
>RUN → □ Δ

```

ΔΔ – A kódok és karakterek táblázatai nyomtatásban:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
2	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
3	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242
4	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243
5	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244
6	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245
7	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246
8	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247
9	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248
10	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249
11	10	26	42	58	74	90	106	122	138	154	170	186	202	218	234	250
12	11	27	43	59	75	91	107	123	139	155	171	187	203	219	235	251
13	12	28	44	60	76	92	108	124	140	156	172	188	204	220	236	252
14	13	29	45	61	77	93	109	125	141	157	173	189	205	221	237	253
15	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
16	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1				0	@	P	—	┌								
2		!	1	A	Q	—	•				▀	└	—	•	▀	└
3		"	2	B	R	—	—	—			▀	└	—	—	▀	└
4		#	3	C	S	—	—	—			▀	└	—	—	▀	└
5		\$	4	D	T	—	—	—			▀	└	—	—	▀	└
6		%	5	E	U	—	—	—			▀	└	—	—	▀	└
7		&	6	F	V	—	—	X			▀	└	—	—	▀	└
8		/	7	G	W	—	—	0			▀	└	—	—	▀	└
9		<	8	H	X	—	—	+			▀	└	—	—	▀	└
10		>	9	I	Y	—	—	—			▀	└	—	—	▀	└
11		*	8	J	Z	—	—	+			▀	└	—	—	▀	└
12		+	7	K	[—	—	—			▀	└	—	—	▀	└
13		,	6	L]	—	—	—			▀	└	—	—	▀	└
14		-	5	M	^	—	—	—			▀	└	—	—	▀	└
15		.	4	N	_	—	—	—			▀	└	—	—	▀	└
16		/	3	O	`	—	—	—			▀	└	—	—	▀	└

ABC: 105 NEM *

HTZ: 105 NEM *

TEXAS: 105 NEM *

PRIMO: CHR\$(8) nem értelmezett* CHR\$(14) ≡ CHR\$(87) CHR\$(1)* CHR\$(15) ≡
CHR\$(64)* CHR\$(16) ≡ TAB()* CHR\$(17) ≡ UPPER* CHR\$(18) nem
ért* CHR\$(26, 145, 146) nem ért* CHR\$(62) dült karakterek* CHR\$(45)
CHR\$(1) aláhúzott krakterek* CHR\$(82) CHR\$(2) Cirill krakterek*

TV COM: CHR\$(8) nem graf* CHR\$(13) ≡ LPRINT* CHR\$(16) ≡ STRING\$(N, " ' ")*
>CHR\$(17, 18, 26, 27) nem értelmezett* CHR\$(9) vízszintesen tabulál*
CHR\$(11) függőlegesen tabulál*

VT-16: CHR\$(8) ≡ CHR\$(27) CHR\$(14)* CHR\$(16) = TAB ()* CHR\$(17) ≡
CAPS LOCK* CHR\$(18) nem ért* CHR\$(26) ≡ STRING\$(N,
CHR\$(K))* CHR\$(27) CHR\$(76) 12 sor/2,5 cm : finom sorfelbontás*

– A CHR\$ függvény a Ø,..., 255 kódokhoz karakter megfelelőiket rendeli hozzá. A karakter négyszögből látható, hogy a 96–127 kódok a 192–233 kódok karakterivel rendre megegyeznek. Hasonló a helyzet a 16Ø–19Ø és 224–254 kódokkal is. A négyszögből láthatóan 4 oszlop hiányzik. Ezeket a >65 utasítással kizártuk, mert egyrészüknek nincs karakter megfelelője, másrészüknél viszont nem karakter, hanem vezérlő kód pl.:

>CHR\$(8) : A GÉPET GRAFIKUS ÜZEMMÓDBA ÁLLITJA,
>CHR\$(1Ø) : SORKÖZT GENERÁL,
>CHR\$(13) : SORVÁLTÁST GENERÁL,
>CHR\$(14) : DUPLASZÉLES KARAKTER ÜZEMMÓDRRA ÁLLIT,
>CHR\$(15) : NORMÁL KARAKTERES MÓDRRA VISSZAÁLLIT,
>CHR\$(16) : NYOMTATÁS POZICIONÁLÓ,
>CHR\$(17) : LEVELEZÉSI KARAKTERES MÓDRRA ÁLLIT,
>CHR\$(18) : NEGATIV KARAKTERES MÓDRRA ÁLLIT,
>CHR\$(26) : KARAKTERISMÉTLŐ NYOMTATÁSI MÓDRRA ÁLLIT,
>CHR\$(27) : NYOMTATÁSI KÉPPONT POZICIONÁLO ELEM,
>CHR\$(145) : NORMÁL KARAKTERES MÓDRRA VISSZAÁLLIT,
>CHR\$(146) : NEGATIV KARAKTERES MÓDRÓL VISSZAÁLLIT.

– Az utasítások hatását egy nyomtatási programon tanulmányozzuk:

105

Nyomtassa ki a saját nevét, a printer "betűkészletét" kihasználva különböző formákban és pozíciókban:

ABC: NEM *

HTZ: NEM *

TEXAS: NEM *

PRIMO: >1Ø törölve* >4Ø LPRINT CHR\$(27) CHR\$(87) CHR\$(1) : LPRINT
"4Ø"; "VARGA"* >45 LPRINT CHR\$(27) CHR\$(87) CHR\$(1)
CHR\$(27) CHR\$(62) : LPRINT "45"; "VARGA" : LPRINT CHR\$(27)
CHR\$(64)* >5Ø LPRINT "5Ø" TAB(2Ø) "VARGA" : LPRINT CHR\$(27)
CHR\$(87) CHR\$(2)* >6Ø LPRINT "6Ø" TAB(1Ø) " "*

TV COM: >1Ø X\$=CHR\$(15)* >3Ø PRINT#4 : "3Ø"; "VARGA" : LPRINT :
PRINT#4 : "JOZSEF" : LPRINT * >4Ø PRINT#4 : "4Ø"; CHR\$(14);
"VARGA JOZSEF"; X\$* >5Ø PRINT#4 : "5Ø"; CHR\$(14); STRING\$(
2Ø, " "); "VARGA JOZSEF"* >6Ø lásd >5Ø * >7Ø
PRINT#4 : "7Ø"; "varga j" >9Ø PRINT#4 : "9Ø"; "VARGA"
; CHR\$(9); "JOZSEF"* CHR\$(9) helyett próbáljon CHR\$(11), vagy
STRING\$(3Ø, CHR\$(127))-el dolgozni*

VT-16: >3Ø : PRINT#4, " "* >5Ø PRINT#4, "5Ø"; CHR\$(14) TAB(2Ø)
"VARGA JOSEF"* >7Ø PRINT#4, "7Ø"; "varga jozsef"*
>8Ø PRINT#4, "8Ø"; CHR\$(14) "varga" * >9Ø
PRINT#4, "9Ø"; CHR\$(27) CHR\$(14) "VARGA JOSZEF" X\$* >1ØØ és
>11Ø helyett állítsa be az alábbiakat : >1ØØ PRINT#4, STRING\$(2Ø,
CHR\$(95)) "VARGA" STRING\$(2Ø, CHR\$(95))* >11Ø PRINT#4,
STRING\$(4Ø, " ") "VARGA" *

*

```

>NEW
>5 PRINT "***NYOMASPROBA**"
>10 OPEN4,4 : X$=CHR$(15)
>20 PRINT#4, "20";"VARGA JOZSEF": PRINT#4, CHR$(10)
>30 PRINT#4, "30";"VARGA" CHR$(13) "JOZSEF" : PRINT#4
>40 PRINT#4, "40"; CHR$(14) "VARGA JOZSEF" X$
>50 PRINT #4, "50"; CHR$(14) CHR$(16) "20 VARGA JOZSEF"
>60 PRINT#4, "60"; CHR$(16) "10 VARGA JOZSEF" X$
>70 PRINT#4, "70"; CHR$(17) "VARGA JOZSEF" X$
>80 PRINT#4, "80"; CHR$(14) CHR$(17) "VARGA" X$
>90 PRINT#4, "90"; CHR$(18) "VARGA JOZSEF" X$
>100 PRNTI#4, "100"; CHR$(17)CHR$(18) "VARGA JOZSEF" X$
>110 PRINT#4, "110"; CHR$(14) CHR$(18) "VARGA JOZSEF" X$
>120 CLOSE4 : END
>RUN

```

*

△△ – A nyomáspróba eredménye:

20VARGA JOZSEF

30VARGA
JOZSEF

```

40VARGA JOZSEF
50          VARGA JOZSEF
60  VARGA JOZSEF
70var9a Jozsef
80var9a
90var9a
100var9a
110var9a

```

Az eredményeket vessék egybe a kiváltó utasításokkal. Látható az egyes CHR\$ és a CHR string-párok utasítás értéke. A CHR\$(15) univerzális feloldójel, amit mindig ki kell tenni, ha más betűtípusra akarunk áttérni.

ABC: NEM *

HTZ: NEM *

TEXAS: NEM *

PRIMO: NEM *

TV COM: >10 : DIM A\$(4)* >20 X1\$ = STRING\$(5, " ") : X2\$ = STRING\$(8, " ") : X3\$ = STRING\$(2, " ") : X4\$ = STRING\$(4, " ") : X5\$ = STRING\$(3, " ") : X6\$ = STRING\$(16, " ") * >25 Y\$ = CHR\$(127) >30 PRINT#4 : CHR\$(14); STRING\$(14, " ") "SZEM ..."* >60 PRINT#4 : Y\$; : FOR J=1 TO 4 : PRINT#4 : STRING\$(16, CHR\$(45)); Y\$; : NEXT : LPRINT * >70 PRINT#4 : Y\$; X1\$; AS(1);


VT-16: >20 : X\$=CHR\$(127)* >30PRINT#4, CHR\$(14) TAB(14) "SZEM"CHR\$(10) CHR\$(15)* >60 FOR J = 1 TO 4 : PRINT#4, X\$; STRING\$(15, CHR\$(176)); : NEXT J : PRINT#4, X\$ * >70 PRINT#4, X\$TAB(6) A\$(1) TAB(17) X\$ TAB(19) A\$(2) TAB (33) X\$TAB (36) A\$(3); * >80 PRINT#4, TAB(49) X\$ TAB(51) A\$(4) TAB(65) X\$ * >90 azonos >60 * >110 PRINT#4, X\$ TAB(17) X\$ TAB(33) X\$ TAB(49) X\$ TAB(65) X\$: NEXT J *

Írjon programot, amely címet, keretet, feliratokat és oszlopelválasztó vonalakat nyomtat a P90 személyi nyilvántartáshoz:

```

*
>NEW
>10 PRINT "***TABLAZAT FORMALIZALAS**"
>20 OPEN4,4 : X$ = CHR$(16)
>30 PRINT#4, CHR$(14)X$ "14 SZEMELYZETI
      NYILVANTARTAS" CHR$(10) CHR$(15)
>40 FOR I=1 TO 4 : READ A$(I) : DATA NEV, VEGZETTSEG,
      MUNKABER, GYEREKEK SZ : NEXT I
>60 PRINT#4, "┌-----┐
      ┌-----┐-----┐"

>70 PRINT#4, X$"001"; X$"05"A$(1);X$"161 "; X$
      "18" A$(2); X$ "321"; X$ "35" A$(3);
>80 PRINT#4, X$ "48 I "; X$ "50" A$(4); X$ "64 I" CHR$(15)
>90 PRINT#4, "└-----┘
      └-----┘-----┘"

>100 FOR J=1 TO 10
>110 PRINT#4, X$ "00 I"; X$ "161"; X$ "321"; X$
      "48 I"; X$ "64 I" : NEXT K
>120 CLOSE4 : END
*
>RUN →  Δ

```

△△

– A táblázatokat a programírás előtt gondosan meg kell tervezni. Ebben az adatoszlopok és a fejléc szélessége, a cím kezdő pozíciója és betűtípusa, a fejléc grafikai elemei (pl. ┌, ┐, └, ┘, ┌, ┐, └) és pozícióik, a feliratok kezdő pozíciói és betűtípusaik,

ABC: —

HTZ: —

TEXAS: —

PRIMO: —

TV COM: X2\$; Y\$; X3\$; A\$(2); X4\$; Y\$; X5\$; A\$(3); * >80 PRINT#4 : X1\$; Y\$; X3\$;
A\$(4); X5\$; Y\$ * Az X1\$, ... szóközöket jelöl ki * >90 Lásd >60, de CHR\$(
(45) helyett CHR\$(95) szerepel * >110 PRINT#4 : Y\$; : FOR L=1 TO 4 :
PRINT#4 : X6\$; Y\$; : NEXTL : LPRINT : NEXT J *

VT-16: A grafikai elemek ASC kódjaik (pl. 172–191 stb) alapján vihetők be a . program-
ba *

valamint a táblázat hossza kerül eldöntésre.
A P106 például az alábbi tervből (részlet)
indult ki:

”14” SZEMELYZETI NYILVANTARTAS

”00”	”16”	”32”
”05”nev	”18”vezettség	”35”munkaber

Tudjuk, hogy a grafikai elemek a klaviatúra útján vihetők be a gépbe (a gombok homlokzatán található négyzetekbe foglalva). A baloldali homlok jel a **G** a jobboldali pedig a **SHIFT** gomb közreműködésével aktiválható. A grafikai elemek önállóan vagy konfigurációkban jelenhetnek meg (pl. >60). Mindkét esetben karakter konstansként kezeljük, vagyis idézőjelbe tesszük őket. A fejlécen elhelyezett feliratokat READ-del olvassuk be a gépbe (>40) és CHR\$(16)-al pozicionálva írjuk ki a printerrel (>70 ; >80). A printer ugyanazt a sort csak egyszer érinti, ezért a feliratok és elválasztó vonalak felváltva szerelepnek a megfelelő utasításokban (>70 ; >80). Mivel mindent pozicionálni kell az ilyen szerkesztésnél, a CHR\$(16)-ot célszerű rövid jellel (pl. X\$) azonosítani.

- A pozicionálás idézőjelbe tett és X\$ után elhelyezett kétjegyű számmal valósítható meg. Szöveg pozicionálása esetén az idézőjelet a szám előtt és a szöveg után elegendő

ABC: 107 NEM *

HTZ: 107 NEM *

TEXAS: 107 NEM *

PRIMO: 107 NEM *

TV COM: -

VT-16: 107 >340 B = CVI (B\$) : M = CVS (M\$) : G = CVI (G\$) : GOSUB 600 * >350
PRINT #4, X\$ TAB(6) LEFT\$ (N1\$, 10) TAB(17) X\$ TAB(19) B ; TAB(33) X\$
TAB(36) " " ; * >355 PRINT#4, USING "#####.#"; M ; : PRINT#4,
TAB(49) X\$ TAB(51) G ; TAB(65) X\$ * >500 FIELD#1, 30 AS N1\$, 3 AS
B\$, 9 AS M\$, 3 AS G\$ * >510 GET#1, 1 * >700 OPEN" R ", #1, "SZEMELY-
ZETI", 45 *

kitenni (pl. "ØØ I"), de a karakter változókat ki kell vonni az idézőjel alól (pl. "Ø5" A\$(1)). Ez utóbbi vonatkozik a számokra és numerikus változókra is.


107

Egészítse ki a P1Ø6-ot olyan utasításokkal, hogy alkalmas legyen disk-re rögzített "SZEMÉLYZETI" anyag kinyomtatására:

>1ØØ, 11Ø, 12Ø

RET

```

>3ØØ PRINT "A RELATIV FILE BETOLTESE ES NYOMTATASA"
>31Ø GOSUB 7ØØ
>32Ø INPUT "HANY REKORDOT NYOMTAT?"; R
>33Ø FOR I=1 TO R : GOSUB 5ØØ
>34Ø INPUT#1Ø7, N$, B, M, G : GOSUB 6ØØ
>35Ø PRINT#4, X$ "ØØ I"; X$ "Ø3" N$; X$ "16 I"; X$ "23" B ;
      X$ "32 I";
>355 PRINT#4, X$ "37" RIGHT$( " " + STR$(M), 5)
      ; X$ "48 I"; X$ "53" G; X$ "64 I" : NEXT I
>36Ø CLOSE 1Ø7 : CLOSE 15 : CLOSE : END
>5ØØ H = INT (I/256) : L=I-H * 256
>51Ø PRINT#15, "P" + CHR$(5) + CHR$(L) + CHR$(H) + CHR$(1)
>52Ø RETURN
>6ØØ INPUT#15, EN, EM$, ET, ES : IF EN <>Ø THEN PRINT EN,
      EM$, ET, ES
>61Ø RETURN
>7ØØ OPEN 15, 8, 15 : OPEN 1Ø7, 8, 5, "SZEMÉLYZETI"
>71Ø RETURN
>RUN →  Λ

```

*

ABC: —

HTZ: —

TEXAS: —

PRIMO: —

TV COM: NEM**

VT-16: A >350 utasításban **LEFT\$(N1\$, 10)** a neveket **balra** igazítja, a **TAB(36)** ””;
pedig a **PRINT#4, USING** kezdetét pozicionálja**

△△

- A kiegészítés a P90 utasításaiból adódik. Hiszen a "SZEMÉLYZETI" file adatait be kell tölteni a gépbe, hogy azután nyomtathassuk. A betöltést a >340 kezdeményezi. A >350 és >355 utasítások a rekord-ok adatait a táblázat megfelelő rubrikáiba sorolják, ugyanakkor tovább építik a keretet is, helyettesítve a P106 >100 és >110 utasításait.
- A futás próba meglepő eredményre vezet, amennyiben a rekord-ok az első oszlopba sorolódnak, pedig adataikat külön-külön pozicionáltuk, mégpedig N\$-et az első, B-t a második stb. oszlopokba. Ennek a hibának a magyarázata a relatív file természetéből adódik. A relatív file a rekord adatait karakter konstanssá egyesíti a lekérdezés számára. Ezért alkalmaztuk az INPUT# utasításokban az A\$ változót a P89 és P90 esetében. A mostani >340 utasításban felesleges a B, M és G kiírása, hiszen N\$-be tölti a program a rekord-okat. Ha pl. B-vel kezdődött volna a rekord és INPUT# B, N\$, M, G betöltést alkalmaztunk volna, sikertelen vállalkozás lett volna. A relatív file rekord-jaiban foglalt adatok akkor kezelhetők külön, ha a rekord-ból kiemeljük. Erre szolgálnak a már ismert LEFT\$ és RIGHT\$ függvények, valamint a MID\$, ami a karakter konstans belsejéből képes kiemelni karakter szeleteket. A kiemelés csak akkor lesz pontos, ha tudjuk az adatok helyét a string-ben. A P67-nél megmutattuk

ABC: 108 NEM * Lásd 121 *

HTZ: 108 NEM * Lásd 121 *

TEXAS: 108 NEM * Lásd 121 *

PRIMO: 108 A CHR\$(27) használata bizonytalan, ezért karakter grafikát alkalmazunk * >1Ø törölve * >2Ø LPRINT TAB(12) "MASODFOKU FUGGVENY"*
>3Ø LPRINT TAB(14) "Ø"* >6Ø Y = 1Ø + 3 * X ↑ 2 + 2 * X - 1 * >7Ø ->8Ø
törölve * >9Ø LPRINT RIGHT\$ (" " + STR\$(X), 4) TAB(Y) " * "*
>11Ø END *

TV COM: 108 >1Ø törölve * >2Ø PRINT#4 : CHR\$(14); STRING\$(12, " ");
"MASOD . . ." * >3Ø PRINT#4 : CHR\$(15); STRING\$(1Ø, " "); Ø *
>6Ø Y = 1Ø + 3 * X ^ 2 + 2 * X - 1 * >7Ø IF Y < 5 OR Y > 77 THEN 1ØØ *
>8Ø törölve * >9Ø PRINT#4, USING "##.#" : X ; : PRINT#4 : STRING\$(Y - 4, " "); ". "*

VT-16: 108 >6Ø Y = 1Ø + (3 * X ^ 2 + 2 * X - 1) * >7Ø IF Y <= 5 OR Y >77 THEN
1ØØ * >8Ø töröl * >9Ø PRINT#4, RIG HT\$ (" " + STR\$(X), 4) TAB(Y) " "*
>11Ø CLOSE#4 : *

a megoldást, ami nemcsak kazetta, hanem disk esetében is célravezető. Az alkalmazhatóság szekvenciális és relatív file-ok esetében is fennáll.



Tervezzen különböző tartalmú és formátumú táblázatokat. Írjon rögzítő programokat és alkalmazza azokat:

053

Függvények grafikus ábrázolása. A CHR\$(27) használata.

108

Írjon programot a $3x^2+2x-1$ másodfokú függvény képének a $[-2; 2]$ intervallumban való kinyomtatásához:

<p>*</p> <p>Λ ←</p> <p>←</p> <p>←</p> <p>Λ ←</p> <p>*</p>	<pre> >NEW >10 OPEN4,4 >20 PRINT #4, CHR\$(14) CHR\$(16) "12 MASODFOKU FUGGVENY" CHR\$(10) >30 PRINT #4, CHR\$(15) CHR\$(16) "10" 0 >40 PRINT#4, "X" "-----" "-----" >50 FOR X=-2 TO 2 STEP 0.1 : X=INT (10*X + .5) / 10 >60 Y=6*10 + 6 *(3*X↑2+2*X-1) >70 IF Y<6 OR Y>474 THEN 100 >80 H=INT (Y/256) : L=Y-H*256 >90 PRINT#4, STR\$(X) + CHR\$(27) + CHR\$(16) + CHR\$(H) + CHR\$(L) "*" >100 NEXT X >110 CLOSE4 : END >RUN → [F1] Λ </pre>
---	---

ABC: —

HTZ: —

TEXAS: —

PRIMO: A TAB funkció printer esetében térközt generál, tehát **nem az első** pozícióhoz igazodik*
*

TV COM: —

VT-16: —

△△

- A printer alkalmas függvény grafikon nyomtatására is. Programozási szempontból egyszerűbb, ha a koordináta rendszert 90° -al elforgatjuk, vagyis az X tengelyt függőleges, az Y tengelyt pedig vízszintes helyzetűnek tekintjük. Első közelítésben a tengelyek elhelyezését is csak érzékeltetjük azáltal, hogy kitűzzük az $Y=0$ helyet ($>3\emptyset$); [itt állna az X tengely] és az X szóabajövő értékeit a nulladik oszlopba nyomtatjuk [ahol $X=0$, ott fekédné az Y tengely]. A függvényértéket a $>6\emptyset$ utasítás számolja az X-re vonatkozó FOR ciklusra támaszkodva.
- A függvényérték számításhoz tudnunk kell, hogy a printer $8\emptyset$ karakter pozíciót ural. Minden karakter pozícióban 6 oszlopot (mini-karakter) és minden mini karakterben 7 pontot (mikro-karakter) különböztet meg. Így egy printer sor $7*48\emptyset$ elemű pontmátrixnak tekinthető.
- A függvényértéket mini karakterben kell számolni. A függvényérték abszolút ($6*Y$) és relatív ($6*D+6*Y$) módon számítható. Az abszolút az X tengelyt a $0\emptyset$, a relatív pedig a D-edik karakteren fekvőnek tekinti (példánkban $D=1\emptyset$).
- A $>7\emptyset$ utasítás a printer-t védi a lehetetlen végrehajtású parancsoktól.
- A $>8\emptyset$ és $>9\emptyset$ utasítás a printer-t pozicionálja az X helyettesítési értékének ($STR\$(X)$) és a függvényértéket hordozó pontnak ("*") a kinyomtatásához. A két utasítás hasonlít a pointer pozicionálásánál megismert utasításokhoz. A $>7\emptyset$ – $>9\emptyset$ utasítások szubrutint képeznek.

ABC: Lásd *

HTZ: Lásd *

TEXAS: Lásd *

PRIMO: >52 LPRINT RIGHT\$(" " + STR\$(X), 4); * >55 FOR C = -5 TO 3 *
>70 IF Y < 4 OR Y > 41 THEN 95 * >80 IF C > -5 THEN 92 * >85 A = Y *
>90 LPRINT TAB(Y) " * "; * >92 IF Y - A <= 0 THEN 95 * >93 LPRINT
TAB(Y-A) " * "; * >94 A = Y * >95 NEXT C : LPRINT *

TV COM: >60 Y = 10 + 3 * X ^ 2 + 2 * X + C * >80 IF C > -5 THEN 90 *
>85 PRINT#4, USING "###.#" : X ; : PRINT#4 : STRING\$(Y-4, " "); " .";
: GOTO 95 * >90 PRINT#4 : STRING\$(.1, " "); " ."; *
>95 NEXTC : LPRINT *

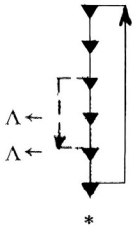
VT-16: >60 Y = 10 + (3 * X ^ 2 + 2 * X + C) * >80 IF C > -5 THEN 90 *
>85 PRINT#4, RIGHT\$(" " + STR\$(X), 4); * >90 PRINT#4, TAB(Y) CHR\$(
(44)); * >95 NEXT C : PRINT#4, " " * Próbálja meg a printer hátoldalán lévő
16 körmös kapcsolót egyenként felső állásba kapcsolni és printelni *

054

A görbesereg ábrázolása.

109

Módosítsa a P108-at arra az esetre, amikor a másodfokú függvény $3x^2+2x+c$ alakú, ahol c a $[-5 ; 15]$ intervallumban mozog:



```

>55 FOR C= -5 TO 15
>60 Y=6*10+6*(3*X^2+2*X+C)
>80 H=INT(Y/256):L=Y-H*256:IF C<-5 THEN 90
>85 PRINT#4,STR$(X);
>90 PRINT#4,CHR$(27)+CHR$(16)+CHR$(H)+CHR$(L)"";
>95 NEXT C
>RUN → [ ] Λ

```

△△

- Kétváltozós függvényt a síkon szintvonalalival tudunk ábrázolni. Az egyik változót $0, 1, 2, \dots$ értékeken rögzítjük és az így keletkező egyváltozós függvényeket ábrázoljuk ugyanabban a koordináta rendszerben. Így a kétváltozós függvényt jellemző görbesereg jön létre.
- A görbesereg nyomtatása fordított eljárást igényel. Minden görbét egyszerre kell elindítani és lépésről-lépésre kifejleszteni. Példánkban a parabolák alkotnak sereget. A sereg elemei a C rögzített értékeihez kapcsolódnak. Ezért ágyazzuk be a C ciklust az X ciklusba. Ezért választottuk le X nyomtatását (>85) a >90 utasításról. Az elemek sorozódását ;-k biztosítják.

ABC: 110 Lásd 121 *
*

HTZ: 110 Lásd 121 *
*


TEXAS: 110 Lásd 121 *
*

PRIMO: 110 >2Ø törölve* >4Ø LPRINT TAB(15) N\$ * >5Ø LPRINT * >6Ø
LPRINT TAB(2Ø) "Ø"* >7Ø LPRINT "X" " - - -" * >11Ø Y = 16 + Y *
>12Ø IF Y < 4 OR Y > 41 THEN 15Ø* >13Ø törölve* >14Ø LPRINT
RIGHT\$ (" " + STR\$(X), 4) TAB (Y) " . " *

TV COM: 110 >2Ø REM* >5Ø LPRINT * >9Ø A lépéshosszot változtassuk * >1ØØ
Y = N * SIN(X) * Az ordinátát nagyíthatjuk (N)* >11Ø Y = 2Ø + Y *

VT--16: 110 »FÜZZÖN BE MAXIMÁLIS SZÉLESSÉGŰ PAPIRT A PRINTER-BE*
>12 WIDTH" LPT1 : " , 14Ø* >2Ø OPEN "O" , #4,"LPT1:" * >4Ø PRINT#4,
CHR\$(14) TAB(15) N\$ CHR\$(1Ø) * >5Ø PRINT#4, " " * >6Ø PRINT#4,
CHR\$(15) TAB(2Ø) Ø* >11Ø Y = 2Ø + Y * >12Ø IF Y <= 5 OR Y > 14Ø
THEN 15Ø* >13Ø PRINT#4, RIGHT\$ (" " + STR\$(X), 4); * >14Ø
PRINT#4, TAB(Y) " . " *

```

*
>NEW
>10 PRINT "***FUGGVENYEK NYOMTATASA***"
>15 PRINT "100-AS CIMKEVEL KOPOGJA LE A FUGGVENYT
      : PL. Y= X↑2-SIN(X) MAJD RUN20"
>16 STOP
>20 OPEN4,4 : X$ = CHR$(16)
>30 INPUT "A FUGGVENY NEVE? " ; N$
>40 PRINT#4, CHR$(14) X$ "15" N$ CHR$(10)
>50 PRINT#4
>60 PRINT#4, CHR$(15) X$ "20" 0
>70 PRINT#4, "X" " ----- "
>80 INPUT "AZ X HATARAI?" ; A, B
>90 FOR X=A TO B STEP .1 : X=INT (10*X+.5)/10
>100 REM
>110 Y=6*(20+Y)
>120 IF Y<6 OR Y>474 THEN 150
>130 H=INT (Y/256) : L=Y-H*256
>140 PRINT#4, STR$(X) + CHR$(27) + X$ + CHR$(H) +
      +CHR$(L) ". ."
>150 NEXT
>160 CLOSE4 : END
*
>RUN →  Δ

```

△△ – A P110 fregoli program, mert a >16 STOP alatt akarmilyen egyváltozós függvényt (pl. $X \uparrow N$; $X \uparrow (-N)$; $(1 - X \uparrow 2) \uparrow (1/2)$; $ABS(F(x))$, $COS(F(X))$; $EXP(FX)$; $INT(F(X))$; $LOG(X)$, $SQR(F(X))$; $X(T)$, $Y(T)$ stb.) bevihetünk a programba (értékkadó utasítás

ABC: NEM *

HTZ: NEM *

TEXAS: >20 PRINT "A GYOKJEL MINI KODJAI 8x2-es BONTÁSBAN:
Ø, F, C, 8, 4, 8, 4, 8, 2, 8, 2, 8, 1, 8, 1, 8" * >25 X\$ = " Ø F C 8 4 8 4 8 2 8 2 8
1 8 1 8" * >30 CALL CHAR (140, X\$) * >35 CALL HCHAR (5, 1, 140, 32) *
>40 END * Lásd 4. számú melléklet *

PRIMO: NEM * Képernyőn megoldható *

TV COM: NEM * Képernyőn megoldható *

VT-16: NEM *

100-as címkével) és RUN 20-al elindíthatjuk a nyomtatását. A programban az X tengely pozíciója között (20), de RUN 20 előtt módosítható a >60 és >110 utasításokban.

- Ha a függvényt különböző intervallumokban akarjuk vizsgálni, akkor RUN20-al ismételnünk és az X határait módosítva adjuk meg.
- Ha a függvény X(T), Y(T) alakú, akkor 100 alatt X=X(T) : Y=Y(T) értékadó utasításokat adjuk meg. Változik >90-ben a ciklus paramétere is (FOR T =).


055

Karaktergrafika. A CHR\$(8) használata.

111

Írjon programot a gép karakter készletéből hiányzó jel, például a gyök jel kinyomtatásához:

```

* >NEW
  >10 PRINT "***KARAKTER GRAFIKA**"
  >20 OPEN4,4 : X$=CHR$(8)
  >25 FOR I=1 TO 80
  >30 PRINT#4, X$+CHR$(130)+CHR$(254)+CHR$(255)+CHR$(129)+
    CHR$(129)+CHR$(130);
  >35 NEXT : PRINT#4, CHR$(15)
  >40 CLOSE4 : END
* >RUN →  Δ

```

△△

- Az elképzelt karakter nyomtatását a CHR\$(8) vezeti be, ami grafikus üzemre állítja át a printer-t. Az ezt követő stringek a gyökjel által lefedett mini és mikro karaktereket adják meg. A megadás mini

ABC: -

HTZ: -

TEXAS: -

PRIMO: -

TV COM: -

VT-16: -

karakter kódokkal történik. Ennek megértéséhez rajzoljunk ki egy printer karaktert részletesen:

	1 2 3 4 5 6		1 2 3 4 5 6
1	○ ○ ○ ○ ○ ○	1	○ ○ ● ● ● ○
2	○ ○ ○ ○ ○ ○	2	● ● ● ○ ○ ●
4	○ ○ ○ ○ ○ ○	4	○ ● ● ○ ○ ○
8	○ ○ ○ ○ ○ ○	8	○ ● ● ○ ○ ○
16	○ ○ ○ ○ ○ ○	16	○ ● ● ○ ○ ○
32	○ ○ ○ ○ ○ ○	32	○ ● ● ○ ○ ○
64	○ ○ ○ ○ ○ ○	64	○ ● ● ○ ○ ○

A mikro karakterekhez helyiérték mutatók (1, 2, 4, 8, 16, 32, 64) tartoznak. Ezek összege a mini karakterekben 127. Ennek nincsen jelentősége. Ha berajzoljuk a karakter mezőbe az általunk elképzelt jelet, az meghatározott mikro karaktereket lefed. Ha a mini karakterekben a jel foglalta mikro karakterek helyiérték mutatóit (példánkban: 2, 126, 127, 1, 1, 2) összeadjuk; majd az összegeket 128-al még megnöveljük (példánkban: 130, 254, 255, 129, 129, 130), akkor jelünk mini karakter kódjait nyerjük. A mini karakterek sorrendje kötött. Ezt a string-ek összeadásánál (>30) érvényesítjük. A string-ek összeadása tehát nem kommutatív.

- A grafikus üzemmódból CHR\$(15)-el kell kilépni. Enélkül a printer nem teljesít további normál (pl. LIST) utasítást.
- Az utóbbi programokból észre kellett venni, hogy a CHR\$ függvénynek számos funkciója van: ASCII kódokhoz rendel karaktereket, karakterek pozícióját, jelek karakterét stb. hozzáadja. Funkcióváltók a CHR\$(8), CHR\$(16), CHR\$(27) stb.

ABC: 112 NEM **

HTZ: 112 NEM **

TEXAS: 112 NEM **

PRIMO: 112 NEM **

TV COM: 112 NEM **

VT-16: 112 NEM **



Dolgozzon ki mini karakter kódtáblázatot az Ön által használt, de a gép karakter készletéből hiányzó jelek (Cirill, Görög, Gót stb. betűk; műszaki, matematikai, zenei stb. jelek) grafikus nyomtatásához pl.:

JEL	1	2	3	4	5	6
√	130	254	255	129	129	130

112

Írjon alkalmazói programot olyan írás nyomtatására, ahol számok, gépi karakterek és saját kódolású grafikai jelek is szerepelhetnek:

```

*
>NEW
>10 PRINT "***VEGYES KARAKTEREK***"
>20 OPEN4,4 : X$=CHR$(8)
>30 IF I=40 THEN I=1
>40 PRINT "NEM SZAM KOVETKEZIK? : Q, SZAM? USSE LE AZ
      A BETUT"
>50 D$=" " :GETD$ : IF D$ <>" " AND D$="Q" THEN 80
>55 IF D$=" " THEN 50
>60 INPUT "KOPOGJA LE A KOVETKEZO JEGYET" ; N
>65 U$=STR$(N) : GOSUB 150
>70 GOTO 30
>80 PRINT "SAJAT JEL? : Q, HA GEPIJELAZT USSE LE?"
>90 D$=" " : GETD$ : IF D$ <>" " AND D$="Q" THEN 110
>95 IF D$=" " THEN 90
>100 U$=D$ : GOSUB 150
>105 GOTO 30
>110 INPUT "SAJAT JEL KODSORA? " : K1, K2, K3, K4, K5, K6
>120 G$=CHR$(K1)+CHR$(K2)+CHR$(K3)+CHR$(K4)+CHR$(K5)+
      CHR$(K6)
>125 J=6*(I-INT(I/180))*80 : H=INT(J/256) : L=J-H*256
>130 P$=CHR$(27)+CHR$(16)+CHR$(H)+CHR$(L) :
      U$=X$+P$+G$+CHR$(J/6)+CHR$(15)

```

ABC: -

HTZ: -

TEXAS: -

PRIMO: -

TV COM: -

VT-16: -

```
>14Ø GOSUB 15Ø
>145 GOTO 3Ø
>15Ø I=I+1 : IF I < 4Ø THEN T$=T$+U$ : RETURN
>16Ø PRINT#4, T$ + U$ : RETURN
>17Ø CLOSE4 : END
*
>RUN
```

△△ — A program akármilyen ABC-t és jeleket használó szövegek soronkénti nyomtatására alkalmas. A program utasításai számokra, gépi és grafikai karakterekre irányulnak, az első kettő a GET-re, a marmadik pedig a karakter grafikára alapoz. Számok, gépi, illetve grafikai karakterek bevitelét a >6Ø, >9Ø, illetve >11Ø utasítások oldják meg. A számok jegyenkénti input-tal, a gépi karakterek egyenkénti GET utasítással kerülnek a gépbe. Ez utóbbi a "Q" grafikai karakterre nyilvánítását vonja maga után. A grafikai karakterek 6–6 mini kódja közös input-tal jut a gépbe. A bevitt elemeket U\$ változóba (>65, >1ØØ, >13Ø) olvassuk be, majd rendre T\$-ben összegezzük. Így a karakterek az előgyűjtő sorban gyülekeznek és 8Ø vagy ennél kevesebb (itt 4Ø) elemű sorokat egyszerre nyomtatunk (>16Ø) ki. Ettől a gyűjtő besorolástól függetlenül a grafikai elemeket pozicionálni kell. (>125, >13Ø) a grafikus mód CHR\$(8)-al kezdődik, a kilépést CHR\$(15)-el kell biztosítani.

ABC: 113 >20 törölve * >40, >50 törölve * >80 DIM S(N) : FOR I = 0 TO N-1 * >90 PRINT "A KOVETKEZO ADATOT KEREM" : INPUT S(I) : IF S(I) >= 40 - D - LEN (NUM χ (S(I))) THEN S(I) = 0 * >95 NEXT I * >100 FOR K = 1 TO 24 : PRINT : NEXT K * >105 FOR I = 0 TO N - 1 * >110 PRINT CUR (3 + I, 1); T + I *

HTZ: 113 >20 törölve * >40; >50 törölve * >80 FOR I = 0 TO N - 1 * >100 NEXT I : CLS * 110 FOR I = 0 TO N - 1 : PRINT T + I; * >115 FOR J = 1 TO S(I) : IF J > S(I) THEN 130 * >120 SET (2 * D + J, 1 + 3 * I) : NEXT J >130 PRINT * CLS törli a képernyőt *

TEXAS: 113 >20 DIM S(100) * >40; >50 törölve * >90 INPUT " " : S(I) * >100 NEXT I :: CALL CLEAR * >110 FOR I = 0 TO N - 1 : : DISPLAY AT (I + 1, 1) : T + I * >120 CALL CHAR (140, "FFFFFFFFFFFFFFFF") :: CALL HCHAR (I + 1, D, 140, S(I)) * >130 DISPLAY AT (I+1, D+S(I)) : S(I) * >150 FOR K = 1 TO 1 E 4 :: NEXT K *

PRIMO: 113 Képernyőn oldjuk meg * >30 PRINT "MEGJELENITES" >40-50 törölve * >75 Q = LEN (STR\$ (T + N)) : IF D < 6 * Q THEN PRINT "D > " 6 * Q : GOTO 30 * >80 FOR K = 0 TO N - 1 * >90 INPUT "A KOV ADAT?"; S(K) : NEXT K * >95 CLS * >100 FOR K = 0 TO N - 1 * >105 IF S(K) >251 - D - 6 * (LEN (STR\$(S(K))) + 2 THEN 140 *

TV COM: 113 >20 törölve * >40, >50 törölve * >100 IF S > 80 - D THEN 140 * >120 PRINT#4 : STRING\$ (D, " "); STRING\$ (S, CHR\$ (127)); * >130 PRINT#4 : " "; S * Ha D túlságosan kicsi (D < LEN (STR\$ (T + I)), akkor sem ütközik a grafikon a >110; je miatt * Oldja meg a feladatot képernyőre is * Próbálja ki >GRAPHICS 2 RET után is a programot *

VT-16: 113 >20 OPEN "O", #4, "LPT1: " * >30 : IF D < 4 THEN 30 * >40, 50 törölve * >60 : IF T < 0 THEN 60 * >70 : IF N <= 0 THEN 70 * >90 : IF S < 0 THEN 90 * >100 IF S > 136 - D THEN 140 * >120 PRINT#4, TAB (D) STRING\$ (S, CHR\$(255)); * >130 PRINT#4, " " S * Próbálja meg a >130 PRINT#4, SPACE\$(2) S utasítást használni * Hagyja el >120-ban a CHR\$ karaktereket [STRING\$(S, 255)] *

- A program kényelmesebbé tehető, ha a karakter kódtáblázatunkat, U\$ tartalommal relatív file-ba rögzítjük. Ezesetben sorszám-mal hívhatjuk a grafikai karaktereket.


056

Diagram grafika. A CHR\$(26) használata.

113

Írjon oszlopdiagram rajzoló programot egy tetszőleges idősorhoz:

```

*
>NEW
>10 PRINT "***OSZLOPDIAGRAM NYOMTATAS**"
>20 OPEN4,4 : X$=CHR$(8) : R$=CHR$(26)
>30 INPUT "AZ X TENGELY POZICIOJA?" ; D
>40 H=INT (6*D/:256) : L=6*D-H*256
>50 P$=CHR$(27)+CHR$(16)+CHR$(H)+CHR$(L)
>60 INPUT "MELYIK AZ ELSO IDOSZAK?LP1980" ; T
>70 PRINT : INPUT "HANY IDOSZAK VAN?" ; N
>80 FOR I=0 TO N-1 : IF I=N THEN 150
>90 INPUT "A KOVETKEZO ADATOT KEREM" ; S
>100 IF S>480-6*D THEN 140
>110 PRINT#4, STR$(T+I);
>120 PRINT#4, X$+P$+R$+CHR$(S)+CHR$(255) ;
>130 PRINT#4, CHR$(15)" "S
>140 NEXT I
>150 CLOSE4 : END
*
>RUN →  Δ

```

- △△ – Az oszlopdiagram egy mini karakter (CHR\$(255)) S-szer ismétlődő (CHR\$(S)) nyomtatásával áll elő. A nyomtatás (>120) grafikus üzemmódban, a mini karakter képének (CHR\$(255)), a kép ismétlésének (R\$) és az ismétlések számának (CHR\$(S)) a pontos megadásával zajlik. A diagrammot

ABC: 113 >12Ø PRINT CUR (3 + I, D); CHR ∩ (151); : FOR J=1 TO S(I) : PRINT
CHR ∩ (127); : NEXT J ** >13Ø PRINT CUR (3 + I, D + S(I) + 1); CHR ∩
(135); S(I) ** >15Ø END ** >1ØØ törli a képernyőt ** >11Ø A 3 + I, 1 pozíció-
ba írja az adatot ** >12Ø Lásd 4/1 számú mellékletet **

HTZ: SET a képernyőt Ø–127 oszlopra és Ø–47 sorra bontja. SET (X, Y) az X-edik
oszlop Y-adik elemét aktiválja. Szakaszok ciklussal aktiválhatók **

TEXAS: –

PRIMO: 113 >11Ø PRINT STR\$ (T + K); * >115 PRINT TAB ((D + S (K)) / 6 + 1)
S(K) * >12Ø FOR I = 179 – (F–1)* 12 TO 179–F * 12 STEP –1 : FOR
J = D TO D + S(K) : SET (J, I) : NEXT J, I : F = F + 1 * >14Ø NEXT K : END *
A TAB karakterre, a SET mikrokarakterre hat **

TV–COM: –

VT–16: –

balról az időszakkal (>110), jobbról pedig az adattal (>130) egészíthetjük ki. A grafikus üzemmódot természetesen fel kell oldanunk. A diagram pozitív és negatív adatra egyaránt vonatkozhat. Negatív esetben a tengely pozíciója 6*D-S. Ha törtszámokhoz nyomtatunk diagrammot, akkor $\text{INT}(10 \uparrow K * S)$ felszorzással és kerekítéssel élünk, ahol $K=1,2,\dots$, tekintettel a >100-ra választható meg.

A printer még további képességekkel is rendelkezik, amikről nem beszéltünk. Próbálják meg például az OPEN4,4,7 megnyitást, vagy CHR\$(17)-et nem feloldani, illetve a CHR\$(17)-et CHR\$(145)-el párban használni egy printer soron belül stb.

ABC: [114], [115] A gép órával nem rendelkezik *

HTZ: [114], [115] A gép órával nem rendelkezik *

TEXAS: [114], [115] A gép órával nem rendelkezik *

PRIMO: [114], [115] A gép órával nem rendelkezik *

TV COM: [114], [115] A gép órával nem rendelkezik *

VT-16: [114] >PRINT TIMER : 'SECUNDUM * Próbálja ki az alábbi programot
>10 LOCATE 25,2 : PRINT TIME\$: GOTO 10 * >RUN * Ha a bekapcsoláskor
megadtuk a pontos időt, akkor a képernyő alsó sorában a pontos idő olvasható *
[115] lásd [2], vagy TIME\$ = "HH : MM : SS" [ENT]. Ez utóbbi bármikor
alkalmazható * >PRINT TIME\$ [ENT] * >LPRINT TIME\$ *

5. A GÉPRENDSZER BELSŐ ADOTTSÁGAI

114

Állapítsa meg, hogy mennyi idő telt el a gép bekapcsolása óta:

* >PRINT TI

RET

△△ – A gép a secundum 60-ad részében számlálja az időt, tehát 1 percet 3600 egységnek jelez. A TI időmérésre lekötött változó.

115

Vigye be a gépbe az aktuális közép-európai időt, majd időnként kérdezze le a pontos időt:

* >NEW

>10 INPUT "MENNYI AZ IDO?"; TI\$

* >RUN

|| MENNYI AZ IDO?? 080540 ||

RET

△△ – Az aktuális időt óra–perc–másodperc-ben 2–2 számjeggyel, elválasztó jelek nélkül kell bevinni. A pontos időt bármikor lekérdeshetjük a bevitel után:

* >PRINT TI\$

RET

Ha a gépet kikapcsoljuk, a pontos idő törlődik.

ABC: >PRINT 64*1024 - (PEEK (65054) + 256*PEEK (65055)) *
A programozó BASIC területe 16 KB. A rekeszek 49152–65535 címeket viselik* 32767–49151-ig csak bővített géprendszerben dolgozhatnánk* 32 KB a gép területe*

HTZ: >PRINT FRE(0)* >PRINT MEM* A programozó BASIC területe 15,2 KB* A 32 KB megosztása: 0–12287 cím BASIC ROM; 12288–13823 EXTENDED BASIC ROM; 13824–15359 klaviatúra; 15360–16383 video; 16384–32767 az ön BASIC területe*

TEXAS: >SIZE * A programozó BASIC területe 13,6 KB. Ez "MEMORY EXPANSION" nevű periféria bekapcsolásával 24511 BYTE-tal bővíthető (beszerezhető).

PRIMO: >PRINT FRE(0) * A programozó BASIC területe 16 KB (32; 48 KB). Ebből 9271 BYTE program és adattár, 6144 BYTE a mikrografika területe, és a maradék a billentyűzet, magnetofon stb. kiszolgálására és nyilvántartásokra szolgál*

TV COM: >PRINT FREE * A programozó BASIC területe 32 KB (48, 64 KB). Ebből ~25 KB program és adattár. A videót 16 KB támogatja* Rendszer nyilvántartások az első ~7 KB-on találhatók*

VT-16: >PRINT FRE (0) * A belső memória 256 KBYTE* Ebből az első 61454 BYTE az ön BASIC területe, ettől 65535 BYTE-ig regiszterek állnak. A maradék 192 KBYTE az operációs rendszerek és nyelvek (BASIC, PASCAL, FORTRAN, C) befogadására szolgál* Beégetett BASIC nincs*

Állapítsa meg, hogy a gép hány BYTE szabad kapacitással áll az Ön rendelkezésére:

*

>NEW

*

>PRINT 64*1024 + FRE(0)

RET

△△ – A normál üzemre állított gép (bekapcsolás utáni állapot) 64 KILOBYTE (1 KILOBYTE= =1024 BYTE) kapacitású belső memóriája cím szerint az alábbi funkciókra van osztva:

1.	1 K	0	RENDSZER "NYILVÁNTARTÓ"	1023	
2.	1 K	1024	KÉPERNYŐ KAR. KÓD TÁROLÓ	2047	
3.	38 K	2048	AZ ÖN BASIC TERÜLETE	40959	
4.	8 K	40960	BASIC ROM	49151	
5.	4 K	49152	RAM	53247	
6.	12 K	53248	GÉPI-	55296 KÉPE. COLOR	56295
		53248 VIDEO 53294		– ROM	65536

- A FRE(0) függvény csak az Ön területét vizsgálja. Ha a gép üres 38909 BYTE szabad kapacitásról kap jelzést. Ha programot visz be a szabad hely csökken. Ugyanez áll elő, ha a programot futtatja és adatokat, műveleti eredményeket memorizál.
- A gép 26K BYTE kapacitást foglal le. Ebben tárolja operációs rendszerét (6. sáv), az INTERPRETER (az Ön programját értelmező rutin) nyilvántartásait (1. sáv), a képernyő aktuális tartalmát (2. sáv) és itt biztosít munkaterületet alpműveletek (4. sáv) számára. A 6. sávban két regiszter is

ABC: >2Ø A DATA utasítás külön címkét igényel * OK *

HTZ: >4Ø " " string-et nem tud beolvasni. Irjon helyette más * OK *

TEXAS: >1Ø törölve * >2Ø A DATA utasítás külön címkét igényel (>2Ø FOR , >21 DATA , >22 NEXT) OK * A karakter 1, a valós szám 8 BYTE-ban tárolódik. Elméletileg 1741 valós szám, illetve 13928 "betű" rögzíthető a memóriában *

PRIMO: OK *

TV COM: >1Ø DIM X(3), Z\$(3) * >3Ø törölve * >6Ø INPUT A, C\$ * * >8Ø DIM V (1Ø, 1Ø) *

VT-16: OK *

helyet kapott. Ezek a képernyő kép és **szín** tartalmát rögzítik.

- Több sávban találjuk a ROM megjelölést. Azokat a memóriákat nevezik így amelyekbe kitörölhetetlen, "beégetett" információkat rögzít a gyártó (pl. INTERPRETER): A beégetett információk is felülírhatók, de tartásuk időleges. A gép kikapcsolásakor visszaáll a beégetett tartalom.

117

Tanulmányozza utasítások, változók és adatok kapacitás foglalását az alábbi programon (vigye be egyenként az utasításokat. Minden utasítás bevitelét kövesse a szabad kapacitás megmérése, majd a programrész futtatása után ismételjük meg a szabad kapacitás mérését):

```
* >NEW
|
| >1Ø A=64*1Ø24+FRE(Ø)
|
| >2Ø FOR I=1 TO 3 : READ X (I) : DATA 32767, Ø,
| -32767 : NEXT I
|
| >3Ø FOR J=1 TO 3 : READ Y%(J) : DATA 32767,
| Ø, -32767 : NEXT J
|
| >4Ø FOR K=1 TO 3 : READ Z$(K) : DATA BOLHA,
| B, " " : NEXT K
|
| >5Ø PRINT "START OF PROCEDURE"
|
| >6Ø INPUT A, B%, C$
|
| >7Ø DIM U (1ØØ)
|
| >8Ø DIM V% (1Ø, 1Ø)
|
| >9Ø DIM W$ (1ØØ)
|
| >1ØØ END
* >PRINT FRE (Ø)
* >RUN
```

ABC: >20 : INPUT A, F * >30 FOR I = A TO F * OK *

HTZ: OK *

TEXAS: >40 A **PEEK** függvény ilyen célra nem használható. Szerepe más *

PRIMO: Ha nemlétező címre kérdezzük **OV** hibajeleket kapunk * OK *

TV COM: OK * Ha **ROM** területet kérdezzük le, a **VIDEO** rekeszek tárolnak fel helyette *

VT-16: Csak az első **64 KBYTE** kérdezhető le * OK *
|| OVER FLOW IN 40 ||*

△△


- Az elemzéshez vegyük figyelembe, hogy a karakter 1, a valós szám 5, az integer típusú szám pedig 2 BYTE-ban tárolódik. A RUN utasítás törli az addig tárolt adatokat, tehát a 1Ø; 1Ø, 2Ø; ...; 1Ø, 2Ø, ..., 1ØØ utasításokból álló, növekvő programok futása kumulált memória igénnyel jár, ugyanez nem vonatkozik a program utasításaira.
- Elméletileg 7782 valós, vagy 19455 egész típusú szám, illetve 38911 "betű" rögzíthető a memóriában. Mivel az adatok a memóriában változókhoz kötődnek, azok helyfoglalásával is számolni kell. Nagyméretű programok, illetve adathalmazok géprevitele esetén érdemes durva számvetést csinálni, mert ha a gép ||OUT OF MEMORY ||jelzést ad, akkor a munka memória hiány miatt félbeszakad.

057

A rekeszek feltárása. A PEEK függvény használata.

118

Irjon programot cím szerint adott rekeszek tartalmának a kimutatására:

```
* >NEW
>1Ø PRINT "***A REKESZK FELTARASA**"
>2Ø INPUT "A CIMLISTA ELSØ ES UTOLSO ELEME ? "; CA, CF
>3Ø FOR I=CA TO CF
>4Ø K=PEEK (I) : PRINT I ; K,
>5Ø NEXT I : END
* >RUN →  Δ
```

ABC: Tanulmányozza az irodalomjegyzék [5] adatait *

HTZ: Tanulmányozza az irodalomjegyzék [6] – [8] adatait *

TEXAS: Tanulmányozza az irodalomjegyzék [9] adatait *

PRIMO: Tanulmányozza az irodalomjegyzék [11] adatait *

TV COM: Tanulmányozza az irodalomjegyzék [13] – [14] adatait *

VT--16: Tanulmányozza az irodalomjegyzék [15] adatait *

A P118 bármely rekesz (0–65535), vagy rekeszszorozat tartalmát kimutatja, Űsse fel a COMMODORE kézikönyv 160-ik oldalát, ahol a rekeszek rendeltetését részletesen ismertetik. Ebből válassza a rekesz címeket a model input-jához.

A rekeszek vagy üresek, vagy számokkal vannak feltöltve. A gép kettes rendszerben tárol (számjegyei a 0 és az 1), de 10-es-ben jelez ki. Egy BYTE 8 BIT-ből áll. Minden bit tartalma 0, vagy 1. A bit-ek helyiértéke 10-es rendszerben számolva, jobbról-balra haladva 1, 2, 4, 8, 16, 32, 64, 128. Ha minden lehető, 8 elemű ismétléses variációt elkészítünk a 0 és 1 jegyekre, akkor minden 0 és 255 közötti szám előáll 2-es rendszerben. A rekeszek tehát [0,255] intervallum egész számait hordozhatják.

Minden gépi karakterhez [0,255] közötti kódot rendeltek. Ugyanaz a kódszám mást-mást jelenthet attól függően, hogy milyen memória sávban foglalhat helyet. Űsse fel a COMMODORE kézikönyv 133, 135, 139 oldalait, ahol a képernyő jelek (1024–2047 rekeszekben) a BASIC ASCII jelkódok (2048–40959) és a színkódok (53248–53294) azonosítását találja. A rekeszek tartalma akarunktól (program bevitel, adatok memorizálása, színezés előírása stb.), illetve a gép operációjától függ. A [0, 255] kódok között számos olyat is találunk, amelyek valamilyen speciális függvény argumentumaként megjelenve a gép üzemmódját váltják.

ABC: OK*

HTZ: OK*

TEXAS: NEM*

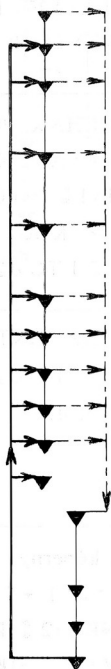
PRIMO: OK*

TV COM OK* mint *

VT-16: >1Ø A=FRE(Ø) : GOSUB 2ØØ* >2ØØ INPUT "ELSO ES UTOLSO REKESZCIM
: PL. 6161Ø, 65535"; A, F*

△△

A P117 és 118 összeépíthető az alább látható módon. Az összevont program alkalmas a rekeszek aktív működésének a követésére, hiszen a rekeszek lekérdezését a P117 minden egyes utasításának a végrehajtása után ismételjük. (Célszerű egy futás alatt ugyanazt az A és F rekeszcímet ismételni):



```

>5  GOSUB 200
>10 A=64*1024 + FRE(0) : GOSUB200
>20 FOR I=1 TO 3 : READ X(I) : DATA 32767, 0, -327
    67 : NEXT I : GOSUB 200
>30 FOR J=1 TO 3 : READ Y%(J) : DATA 32767, 0,
    -32767 : NEXT J : GOSUB200
>40 FOR K=1 TO 3 : READ Z$(K) : DATA BOLHA, B,
    " " : NEXT K : GOSUB200
>50 REM "START OF PROCEDURE : GOSUB 200
>60 INPUT A, B%, C$ : GOSUB200
>70 DIM U(100) : GOSUB 200
>80 DIM V% (10, 10) : GOSUB200
>90 DIM W$(100) : GOSUB 200
>100 END
>200 INPUT "ELSO ES UTOLSO REKESZCIM : PL. 40,
    50 " ; A, F
>210 FOR S=A TO F
>220 K=PEEK(S) : PRINT S ; K,
>230 RETURN

```

Amikor a printer-t megismeri, a rekeszek tartalmát ki is írhatja papírra és tételesen összehasonlíthatja az egy-egy utasítás végrehajtása során keletkező memória változásokat.

ABC: 119 >1 PRINT "SZOVETEK"* >2 PRINT "A KARAKTERKOD ALSO ES
FELSŐ HATÁRA?":INPUT A,F* A képernyő 24x40-es* >6 PRINT CUR(I,J) ;
CHR ♂ (K) : NEXT J : NEXT I * >7 FOR P=1 TO 24 : PRINT : NEXT P *
>8 - >10 törölve* A képernyő nem színezhető *

HTZ: 119 >4 FOR I=15360 TO 16383 * >5 törölve* >6 POKE I, K : NEXT I :
CLS * >7 NEXT K : END * A képernyő nem színezhető, így >8 - >11 töröl-
ve* A 15360-16383 című memória rekeszek a képernyő 16x64 karakterét
kezelik* A képernyő kódok az ASC II. kódokkal azonosak* 129-191 kódok-
hoz grafikai jelek tartoznak* PRINT TAB(H) CHR\$(K) ábrák rajzolására
alkalmas*

TEXAS: 119 POKE utasítás nincs. A CALL SCREEN, COLOR, HCHAR, VCHAR stb.
gépi rutinok helyettesítik* >3 K=A :: GOTO 10 :: REM A=ASC KÓD* >4
T=P : CALL SCREEN (1)* >5 L=L+1 :: IF L>15 THEN L=1* >6 CALL COL
OR (T, L, L+1)* >7 CALL HCHAR (1, 1, K, 768)* >8 K=K+1 :: IF K>F
THEN END* >9 CALL CLEAR :: GOTO 14* >10 FOR C=1 TO 8*

PRIMO: 119 >1 PRINT "SZOVETEK"* >2 INPUT "A KARAKTERKOD
ALSO ES FELSO HATARA?":* >4 FOR I=1 TO 16* >5 FOR J=1 TO 42*
>6 PRINT CHR\$(K); : NEXT J >8 CLS* >9, >10 törölve* A képernyő
nem színezhető*

TV COM: 119 >4 FOR I=1 TO 24* >5 FOR J=1 TO 32* A képernyő 24x32 fel-
bontású >6 PRINT CHR\$(K); : NEXT J* >8 CLS : IF L + 2 >4 THEN
L=0* >10 SET BORDER L : SET PAPER L+1 : SET INK L+2* Keret, alap,
karakter színbeállítás* >105 V=V+1 : IF V>3 THEN V=0* >106 ON V+1
GOSUB 120, 121, 122, 123* V a kiválasztott 4 szín sorszáma* >120 SET
PALETTE 0, 1, 4, 5 : RETURN* 4 szín kiválasztása a szobajövő 16-ból*

VT-16: 119 >2 INPUT "A KARAKTERKOD ALSO ES FELSO HATARA?":
A, F: A > = 3, F < = 254* >4 FOR I = 1 TO 24* >6 LOCATE I, J : PRINT
CHR\$(K) : NEXT J* >8 CLS* >10 IF K/2 = INT (K/2) THEN COLOR 0,15
ELSE COLOR 23,15:SZINVALASZTAS* A karakter színek kódja 0; 7; 8; 15, a
háttéré 0; 7; 15. 16-al növelve villogást látunk* >4 : IF I=24 THEN 8 kiegészít-
és?*

Írjon a képernyőre színes szövetmintákat rajzoló programot:

```

*
>NEW
>1 PRINT "***SZINES SZOVETEK**"
>2 INPUT "A KEPRENYOKOD ALSO ES FELSO
      HATARA? "; A, B
>3 FOR K=A TO F
>4 FOR I=1 TO 25
>5 FOR J=1 TO 40
>6 POKE 983+40*I+J, K: NEXT J
>7 NEXT I
>8 PRINT "  ▣  ": IF L>14 THEN L=0
>9 L=L+1
>10 POKE53281, L+1 : POKE53280, L
>11 NEXT K : END
>RUN → [ ] Λ

```

△△

– A COMMODORE–64 képernyője 25×40 karaktert jeleníthet meg, ami 25 sort és egy sorban 40 karaktert takar. A karakterek címe $1024-2023$ a memóriában. Ezek bármelyikét az alábbi kétváltozós függvény állítja elő:

$$C=983+40*I+J.$$

Ahol I a sor, J pedig az oszlop futóindexe.

– A **POKE** C, K függvény a C című rekeszbe $K=1-255$ számok (a képernyő kódok) bármelyikét elhelyezheti, amire a képernyő adott karakterében a kód megfelelője (kézikönyv 133 o.) válik láthatóvá.

– A video rekeszek (6. sáv) közül az 53280 és

ABC: Vonja össze az >5, >6-ot és egészítse ki az alábbiak szerint: >6 PRINT CUR (I, 1); CHR Ǿ (151); : FOR J=1 TO 38 : PRINT CHR Ǿ (K); : NEXT J : NEXT I * A PRINT CUR (I, 1); CHR Ǿ (151) az egész I-edik sorra érvényes, a CHR Ǿ (K) csak 1 pozícióra *

HTZ: Nem oldható meg * A gép színekkel nem dolgozik *

TEXAS: >11 READ D(C), E(C) * >12 DATA 32, 39, 40, 47, 48, 55, 56, 63, 64, 71, 72, 79, 80, 87, 88, 95 * >13 NEXT C * >14 FOR P=1 TO 8 :: IF D(P) <= K AND K <= E(P) THEN 4 * >15 NEXT P :: GOTO 1 * Lásd a 4. sz. mell. *

PRIMO: NEM *

TV COM: >121 mint >120 a 16, 17, 20, 21 kódokkal * >122 a 64, 65, 68, 69 kódokkal * >123 a 80, 81, 84, 85 kódokkal * Próbálja meg >GRAPHICS 2 után futtatni a programot * Törölje a >105, >120, >121, >122, >123 utasításokat * >80 CLS : IF L >15 THEN L = 0 * Adja a >GRAPHICS 16 parancsot és futtassa újra * L a 16 szín sorszáma *

VT-16: Színes adapter nem tartozik a géphez. A PROPER 16 és az IBM PC-vel megoldható *

53281 a képerket, illetve a kép egységes színezését látja el. (A képernyő karakter mélységű színezését az 55296–56295 című rekeszek biztosítják). A színeket a 0–15 kódokkal azonosították. Ha a kódok egyikét a POKE utasítással bevisszük a jelzett rekeszek valamelyikébe, akkor vagy a keret, vagy a kép felveszi a jelzett színt a képernyőn.

- A P119 a teljes képernyőt beszövi az első, a második,..., a 255-ik karakterrel. A képváltás a képernyő törlésével kezdődik. Erre szolgál a print "☛" utasítás. Ez az idézőjeles negatív szívalak az alábbi gomb kombinációval vihető a gépbe:

SHIFT" , CTRL RVS/ON , SHIFT☛ ,
SHIFT RVS/OFF , SHIFT" ,

ahol az RVS/ON megnyitja, RVS/OFF pedig lezárja a negatív karakteres írásmódot.

- A program három beágyazott ciklusa belülről (>5), kifelé (>3) haladva az oszlop, a sor, illetve a karakter váltást biztosítja. A színváltást léptetéssel, a színsorozat ismétlődését IF-el oldjuk meg. A futó program a szövést imitálja.

059

Karakterek színezése.

120

Rajzolja fel a magyar zászlót a képernyőre.

```
* >NEW
>10 PRINT "***MAGYAR ZASZLO**"
>20 FOR T=1 TO 3 : READ S : DATA 2, 1, 5 : L=L+1
>30 FOR I=1 + (L-1)*8 TO 8*L : FOR J=0 TO 39
>40 POKE 984+40*I+J,240 : POKE55256+40*I+J,S
>50 NEXT J, I, T : END
* >RUN → [ ] Λ
```

ABC:

NEM *

>20 PRINT "'>70-ES CIMKEVEL KOPOGJA LE A VALASZTOTT
FUGGVENYT, TOROLJE >21-ET ES RUN"'* >30 FOR P=1 TO 24 : PRINT :
NEXT P* >21 STOP *

HTZ:

>30 CLS *

TEXAS:

>5 CALL CLEAR :: CALL SCREEN(2) * >10 FOR I=1 TO 3 :: READ
K (I) * >20 DATA 7, 16, 4 * >30 FOR J=8*(I-1) + 1 TO 8*I * >40 FOR
L=1 TO 32 * >50 CALL COLOR (1, K (I), 2) * >60 CALL HCHAR (J, L,
33 + I) * NEXT L :: NEXT J :: NEXT I :: END *

PRIMO:

>30 CLS *

TV COM:

>10 : CLS * >20 SET PALETTE 0, 68, 85, 16 : SET BORDER 0 : SET
PAPER 2 : L=1 * >30 FOR J=1 TO 32 * >40 SET INK L : PRINT CHR\$(127)
; >50 NEXT J, I : L = L + 1 : IF L <= 3 THEN 30 : END *

>30 CLS * >40 PRINT AT 1,15 : N\$ * Első sor 15. pozícióját definiál-
ja a kurzor számára * >50 PRINT AT 2, 20 : 0 *

VT-16:

>30 CLS *

△△

- A képernyő karakterek egyenként is színezhethők. A kívánt színkódot az 55296–56295 című rekeszek bármelyikébe POKE utasítással vihetjük be. Sorozatos bevétel esetén a

$$C=55256+4\emptyset*I+J ; I=1,\dots,25 ; J=\emptyset,\dots,39$$

függvénnyel állítjuk elő a címeket. Fontos körülmény, hogy a színek csak akkor válnak láthatóvá, ha valamilyen jelet is viszünk a képernyőre. Ezért szerepel $>4\emptyset$ alatt egy a színre és egy a karakterre vonatkozó, szinkronizált (azonos I és J) POKE utasítás.

- A zászlót 3*8 képernyő sorra terveztük. A színsávok kezdő sorindexeit az L segédváltozó léptetése mellett az $1+(L-1)*8$ függvény állítja elő. A színkódokat ciklussal olvassuk be.

060

Karakter grafika a képernyőn.

121

Irjon programot egyváltozós függvények képernyőn való grafikus ábrázolására:

*

```
>NEW
>1\emptyset PRINT "***FUGGVENYEK A KEPERNYON***"
>2\emptyset PRINT "7\emptyset-ES CIMKEVEL KOPOGJA LE A VALSZTOTT
      FUGGVENYT, MAJD RUN 25" : STOP
>25 INPUT "A FUGGVENY NEVE; AZ X ALSO ES FELSO
      HATARA ?" ; N$, A, F
>3\emptyset PRINT "  "
>4\emptyset PRINT TAB(15) N$ : PRINT
>5\emptyset PRINT TAB(2\emptyset) "\emptyset"
>6\emptyset FOR X= A TO F STEP.2 : X= INT(1\emptyset*X+.5)/1\emptyset
>7\emptyset REM
```

ABC: 121 >11Ø PRINT X; TAB(19) CHR ♂ (151); CHR ♂ (1Ø6); TAB(Y) CHR ♂ (135); " * " : GOTO 15Ø * >14Ø PRINT X; TAB(Y) " * "; TAB(19) CHR ♂ (151); CHR ♂ (1Ø6) *Egy soron belül vált és visszaválthat a grafikus módból. CUR (I, J) és TAB(J) pozíciónál *

HTZ: 121 >8Ø : IF Y < 1 OR Y > 63 THEN 15Ø * Gyakran nagyítjuk a torz képet pl. Y = 1Ø * SIN (X) *

TEXAS: 121 >3Ø CALL CLEAR * >5Ø CALL VCHAR (1,15, 46, 24) * >54 CALL HCHAR (12, 1, 46, 32) * >8Ø U = 15 + X :: IF U < 1 OR U > 32 THEN 12Ø * >9Ø V=12 - Y : IF V < 1 OR V >24 THEN 12Ø * >1ØØ törölve * >11Ø CALL HCHAR (V, U, 46, 1) * >12Ø NEXT X * >14Ø, >15Ø törölve *

PRIMO: 121 >11Ø PRINT X TAB(2Ø) CHR\$(138) TAB(Y) " * " : * >14Ø PRINT XTAB(Y) " * " TAB(2Ø) CHR\$(138) *

TV COM: >8Ø: IF Y < 4 OR Y > 3Ø THEN 15Ø * Y>6Ø, ha >GRAPHICS 2-vel dolgozunk * >11Ø PRINT X; TAB(2Ø); CHR\$(139); TAB(Y); " * " : * >14Ø PRINT X; TAB(Y); " * "; TAB(2Ø); CHR\$(139) *

VT-16: 121 >8Ø : IF Y < 4 OR Y > 76 THEN 15Ø * >1Ø5 IF Y=2Ø THEN 145 * >11Ø PRINT X TAB(2Ø) CHR\$(179) TAB(Y) " * " : GOTO 15Ø * >14Ø PRINT X TAB(Y) " * " TAB(2Ø) CHR\$(179) : GOTO 15Ø * >145 PRINT X TAB(2Ø) " * " *

```

>80 Y=20+Y : IF Y <4 OR Y >36 THEN 150
>90 REM AZ X TENGYELY POZICIONALASA
>100 IF Y <20 THEN 140
>110 PRINT X TAB(20) "I" TAB(Y) "*" : GOTO 150
>140 PRINT X TAB(Y) "*" TAB(20) "I"
>150 NEXT X : END
>RUN → □ Δ

```

*

- △△
- A normál üzemmódban álló gép alkalmas arra, hogy karakter felbontású képeket rajzoljon a képernyőre. A képrajzolás sorfolytonosan történik. Ezért például a koordináta rendszer a szokásos állás helyett 90 fokkal elforgatva rajzolható, tehát az X tengely lesz függőleges.
 - Az X tengely pozícióját az >50 utasítás állítja be. Az X aktuális értékét és a tengelyt a >110 és >140 utasítások íratják ki. Az Y aktuális értéke ugyanezen utasításokban pozicionálja, tabulálja a függvény megfelelő pontját.
 - A programba akármilyen egyváltozós függvény bevihető. Változtathatjuk a vizsgálati tartományt és a lépésközt is. Kisértékű lépésköz degenerált képhez vezet, mert ha $R \leq Y < R+1$, akkor $TAB(Y) = TAB(R)$, feltéve, hogy R egész szám. Így a görbe kiegyenesedik. A jelenség nagytással kiküszöbölhető (pl. $Y = \cos(X * \pi / 180)$; $Y = N * \cos(X * \pi / 180)$).

ABC: -

HTZ: -

TEXAS: -

PRIMO: -

TV COM: -

VT-16: -

Módosítsa a P121-et oly módon, hogy a koordináta rendszer a hagyományos állásban maradjon:

△△

– Előzetes megegyezések:

1. Egész számokat kettes rendszerbe egy osztás sorozat maradékai visznek át. Pl. 38
 $38:2=19:2=9:2=4:2=2:2=1:2=\emptyset$; $38=1\emptyset\emptyset11\emptyset$

$\emptyset \quad 1 \quad 1 \quad \emptyset \quad \emptyset \quad 1$

az első maradék a legkisebb helyiértékű.

2. Kettes rendszerű számokat a tizedes rendszerűek mintájára adunk össze. Pl.
 $38+13$

$1\emptyset\emptyset11\emptyset+$

$\underline{11\emptyset1}$

$11\emptyset\emptyset11$

3. Kettes rendszerű számokat tizedes rendszerbe a számjegyek és helyiértékek szorzatösszege visz át. Pl. $11\emptyset\emptyset11$

$1*2^5+1*2^4+\emptyset*2^3+\emptyset*2^2+1*2^1+1*2^0$

$1\cdot32+1\cdot16+0\cdot8+0\cdot4+1\cdot2+1\cdot1=51$

4. Egy byte 8 bit-je kettes rendszerben

az alábbi helyiérték táblázattal dolgozik:

8 7 6 5 4 3 2 1

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

A bit-ekre helyiértékükkel és sorszámmal is hivatkozhatunk pl. 32-es helyiértékű bit ; 6-os bit. Szokásos alsó (1–4) és felső (5–8` bit-ekről is beszélni.

5. Egy byte \emptyset –255 közötti számokat hordozhat. Egy byte szám (N) komple-

ABC: —

HTZ: —

TEXAS: —

PRIMO: —

TV COM: —

VT-16: —

mense (\bar{N}) teljesíti az $N+\bar{N}=256$ azonosság-
got pl.:

$$\begin{array}{r} N=51 \quad 00110011 + \\ \bar{N}=205 \quad \underline{11001101} \\ 10000000 \end{array}$$

A komplement képzést jobbról-balra, az első értékes jegyet (1) követő jegyek megfordításával (0 helyett 1 és viszont) intézhetjük el.

6. Két byte A, illetve B tartalmának az $A-B$ különbségét az $A+\bar{B}$, vagy az $\bar{A}+B$ összeg adja meg. A választás A és B közötti relációtól függ. Pl.:

$$\begin{array}{r} A = 137 \quad 10001001 -; \quad 10001001 + \\ B = 51 \quad 00110011 \quad \underline{11001101} \\ \quad \quad \quad \quad \quad \quad \quad 01010110 \end{array}$$

Az összegezésnél a byte túlsordul.

7. Egy byte alsó és felső 4 bit-je más-más jelentést közvetíthet a gép számára. A felső 4 bit helyiértéke úgy tekinthető, mint az alsó 4 bit helyiértékének a 16-szoros nagyítása. Az alsó 4 bit max.15-öt, a felső 4 max. 16*15-öt tárolhat. A két féltékére azonos számokat (Pl. 5) az alábbi módon vihetünk be:

$$\begin{array}{r} 16*5 +5 \\ \hline 0101 | 0101 \end{array}$$

8. Egy üres byte. bármelyik bit-jébe tehetünk egy 1-et:

$$\text{POKE cím, } 2^{\uparrow}N ; \text{ AHOL } 0 \leq N \leq 7$$

9. Két byte-ra értelmezzük az AND és OR logikai műveleteket. Az értelmezés bit paritásra vonatkozik. Eszerint Pl.

ABC: 11. Magyarázat az irodalomjegyzék [5]-ben *

HTZ: 11. Magyarázat az irodalomjegyzék [8]-ban *

TEXAS: 11. Magyarázat az irodalomjegyzék [9]-ben és a 4. számú mellékletben*

PRIMO: 11. A [122] feladat normál üzemmódban oldható meg a **SET (X, Y)** függvény alkalmazásával, ahol $X \in \{0-255\}$, $Y \in \{0-191\}$ értéket vehet fel. X a képernyő oszlopait, Y a képernyő sorait jelöli *

TV COM: 11. Magyarázat az irodalomjegyzék [14]-ben *

VT-16: 11. A **>SCREEN 2** [ENT] állít grafikus üzemmódra, amit **>SCREEN 0** [ENT] von vissza* **>LINE (X1, Y1) – (X2, Y2)** [ENT] szakaszt rajzol két adott pont között. $0 \leq X \leq 639$ és $0 \leq Y \leq 299$ a képernyő felbontása* **>LINE (X3, Y3) – STEP (D1, D2)** [ENT] az $X3, Y3$ pontból az $X3 + D1, Y3 + D2$ -be vezet a szakaszt* **>LINE (X1, Y1) – (X2, Y2), 0,B** [ENT] téglalapot rajzol a szakasz, mint átló köré* **>CIRCLE (U, V), .A** [ENT] egy *

11111111 AND 11111111 OR

11110000

11110000

11110000

11111111

Az AND kimenete csak akkor 1-es (igaz), ha a bemenetek is 1-esek voltak, az OR kimenete viszont csak akkor nulla (hamis), ha a bemenetek is 0-ák voltak. A két művelet 10-es rendszerű számokra is felírható a

0–255 intervallumban pl.:

45 AND 198 = 4

45 OR 198 = 239

00101101 AND

00101101 OR

11000110

11000110

00000100

11101111

10. Ha egy byte tartalma K (PRINT PEEK(C)=K), és az N-edik üres bit-jébe 1-et akarunk beírni, akkor

POKE C, K OR 2↑(N–1), vagy POKE C,
K+2↑(N–1)

megoldások jöhetnek szóba.

Ha K=0 és két vagy több bit-et akarunk 1-re állítani, akkor

POKE C, 2↑(L–1) OR 2↑(N–1), vagy
POKE C, 2↑(L–1) + 2↑(N–1)

megoldások egyikét alkalmazzuk. A második megoldásban a számokat összeadjuk pl.:

POKE C, 2↑3 + 2↑5 = POKE C, 40

11. A 122 feladat normál üzemmódban nem oldható meg. A gépet grafikus üzemmódra kell állítani. Ezt az 53265-ös című video regiszter módosításával érjük el. Eredeti tartalma 00011011, vagyis 27. A 32-es helyiértékű bit-je üres. Ha ezt 1-esre állítjuk:

ABC: >20 PRINT ">180-AS CIMKEVEL KOPOGJA LE A FUGGVENYT,
TOROLJE >21-ET, MAJD RUN"* >21 STOP * >30 PRINT "AZ X TEN-
GELY POZICIOJA?": INPUT I * >40 PRINT "AZ Y TENGELY POZICIO-
JA?": INPUT J * >50 PRINT "X INTERVALLUMA, A, F ES LEPESHOSSZA
S?":

HTZ: —

TEXAS: —

PRIMO: 12. A mikro karakterek kezelésére 6 KBYTE nagyságú puffer tároló áll rendelkezésünkre*

TV COM: —

VT-16: U, V középpontú és A NAGYTENGELYU ELLIPSZIST RAJZOL * >PSET
(X, Y), 1 kigyűjtja az X, Y koordinátájú pontot, a >PRESET
(X, Y), 0 viszont kioltja azt * A képernyő 640x300 képpontot hordoz *

akkor a gép grafikus módra áll át.

Grafikusüzemmódban a gép felbontja a képernyő karaktereit 8–8 sorra (mini karakterek) és minden mini karaktert 8 mikro karakterre (8*8-as felbontás). A mini karaktereket közvetlenül (címezett byte), a bennük foglalt mikro karaktereket közvetve (a címzett byte bit-jei) éri el. A mini karakterek száma $200 * 40 = 8000$. A képernyőn tehát 200 sor és egy sorban 320 mikro karakter, tehát 320 oszlop áll rendelkezésre. A mini sorokat MI=0,1 ,..., 199, a mikro oszlopokat pedig MJ=0,1 , ..., 319 jelöli. A mikro karakter pontnak felel meg.

12. A mini karaktereknek memóriát kell biztosítani. Erre szolgál az 53272 című video regiszter. Ha felső 4 bit-jébe 1-et, az alsóba pedig 8-at teszünk:

POKE 53272, 16*1+8

Akkor a gép memóriája az alábbiak szerint funkcionál:

1.	1 K	0	RENDSZER NYILVÁNTARTÁS	1023
2.	1 K	1024	KÉPERNYŐ KARAKTER SZINTÁR	2047
3.	14 K	2048	8192 MINI KARAKTER TÁR	16383
	24 K		AZ ÖN BASIC TERÜLETE	40959
4.	8 K	40960	BASIC ROM	49151
5.	4 K	49152	RAM	53247
6.	12 K	53248	GÉPI	
		53248	VID 53294	ROM

A mini karakterek elérésére 8K nagyságú terület került kijelölésre a felhasználói sávban. A képernyő karakterek eddigi

ABC: INPUT A, F, S * >60 - >80 törölve* >90 FOR P=0 TO 23 : PRINT : NEXT
P * >100 - >110 törölve * >120 REM AZ X TENGELY RAJZOLASA *
>130 PRINT CUR (I, 0) CHR 151; : FOR K=2 TO 79 : SETDOT 3*I+1, K
: NEXT K * >140 törölve * >150 REM AZ Y TENGELY RAJZOLASA *
>160 FOR K=0 TO 23 : PRINT CUR (K, J) CHR\$ (151); : SETDOT 3*K,
2*(J+1) : SETDOT 3*K+1, 2*(J+1) : SETDOT 3* K+2, 2*(J+1) : NEXT K *

HTZ: -

TEXAS: 122 >10 PRINT "FUGGVENYEK MIKRO GRAFIKONJA" * >20 PRINT
"90-ES CIMKEVEL KOPOGJA LE A FUGGVENYT : PL. Y=X ^ 2, MAJD RUN
30" :: STOP * >30 CALL CLEAR :: Z\$ = "0000001818000000" :: CALL SCR
EEN (12) * >40 INPUT "AZ X ES Y TENGELY POZICIOJA?": MX, MY *
>50 CALL VCHAR (1, MY, 46, 24) : CALL HCHAR (MX, 1, 46, 32) :: REM
TENGELYEK * >60 INPUT "AZ X HATARAI ES LEPESKOZE?": A, F, S *

PRIMO: -

TV COM: -

VT-16: 122 >10 PRINT "FÜGGVÉNYEK MIKRO GRAFIKONJA" * >60 CLS :
SCREEN2 : REM A KEPERNYOT 300*640 PONTRA BONTJA * >70-
>90 törölve * >100 LINE (0, 12*MY) - (639, 12*MY) : REM AZ X TEN-
GELY RAJZOLASA * >110 - >120 törölve * >130 LINE (8*MX, 0) -
(8*MX, 299) : REM AZ Y TENGELY RAJZOLASA * >140 MJ=MX * >150
törölve * >160 MJ=8* (MJ + INT (A/S))

rekeszei funkciót váltanak, amennyiben a képernyő karakterek színezését látják el, mégpedig a felső 4 bitjük a pontok (mikro karakterek), alsó 4 bitjük pedig a háttér (pontközök) színezését biztosítja. Az I. J karakterre vonatkozó színezési utasítás:

POKE 983+40*I+J, 16*P+H,

ahol P és H 0 és 15 között választott színkódokat jelöl. Fontos körülmény, hogy a pontszín csak akkor válik láthatóvá, ha a pontot gerjesztik, vagyis a mini karakter memória rekeszének a megfelelő bit-jét 1-essel töltik fel.

12. Grafikus üzemmódban a mini karakterek címezhetőek. A képernyő bal felső sarkában a 8192-es, a jobb alsó sarkában viszont a 16192-es című mini karakter fekszik. A mini karakterek relatív címzését az alábbi módon oldották meg:

	J = 0	J = 1	J = 38	J = 39
I = 1	0	8	304	312
	1	9	305	313
	7	15	311	319

$(I-1)*320+J*8+k$

I = 25	7680	7688	7984	7992
	7681	7689	7985	7993
	7687	7695	7991	7999

Tehát a relatív cím a mini karaktert befoglaló képernyő karakter sor (I) és oszlop

ABC: >165 J1=J+INT (A/(2*S))* >170 FOR X=A TO F SPTEP S * >180 REM A Z
FUGGVENY HELYE * >190 I1=I-Z : IF 3*I1 <1 OR 3*I1 >78 THEN 220 *
>200 PRINT CUR (I1, 1)CHR (151) : SETDOT 3*I1+1, 2*(J1+1) * >210
J1=J1+.5 : IF 2*(J1+1) >78 THEN END * >220 NEXT X * >230 END *
>240 - >250 törölve *

HTZ: -

TEXAS: >70 FOR X=A TO F STEP S :: IF X > F THEN END *
>80 U=8*(MY+X) :: IF U < 1 OR U > 256 THEN 110 *
>90 REM A FUGGVENY * >95 V=8*(MX-Y) :: IF V <1 OR V >192 THEN
110 * >100 CALL CHAR (140, Z\$) : CALL SPRITE (#1, 140, 7, V, U) *
>110 NEXT X *

PRIMO: -

TV COM: 122 >60 GRAPHICS4 : REM GRAPHICS 2 VAGY GRAPHICS 16 is de-
klarálható * >70 SET PALETTE 0, 1, 16, 68 : REM GRAPHICS 2 ESETÉN
CSAK KÉT KÓD ÊL, 16 ESETÉN >70 elmaradhat * >80 SET BORDER 0
: SET PAPER 2 : SET INK 1 * >90 CLS * >100 SET STYLE 1 : REM A
VONAL FORMA 1, ..., 14 KÓDOKKAL VÁLASZTHATÓ KI *

VT-16: >190 MI=MY - Z : IF MI < 1 OR MI > 24 THEN 210 * >200 PSET (MJ,
12*MI), 1 : REM PONT KIVILAGITASA * >210 : IF MJ >639 THEN 250 *
>220 - >240 törölve * >250 END * A kép nem színezhető *

(J) idenjeiből, valamint a mini karakter, karakteren belüli sorszámából ($K=0, \dots, 7$) meghatározható az

$$(I-1)*32\emptyset+J*8+K$$

képlettel. Ha viszont egy mikro karaktert adunk meg MI, MJ indexekkel, akkor a befoglaló mini karakter relatív címét a

$$32\emptyset*INT(MI/8) + (MI \text{ AND } 7) + 8*INT(MJ/8)$$

képlet adja az előzővel összehangban. Pl. MI=1, Mj=8, akkor: INT (1/8)=0; INT (8/8)=1; 1 AND7 = 1 a cím tehát 9.

Ha a relatív címeket a mini karaktertár kezdő címével (8192) növeljük, akkor abszolút címeket kapunk, ahol a mini karaktereket a gép ténylegesen eléri.

13. A mikro karakter a befoglaló mini karakter címén érhető el, aktiválása pedig, ha a byte üres, a

$$2\uparrow((7-MJ) \text{ AND } 7)$$

byte -ba való helyezésével oldható meg. pl. MI=1; MJ=10, akkor $7=00000111$; $10=00001010$; $7-10=00000111+11110110=1111101$.

$(7-10) \text{ AND } 7=1111101 \text{ AND } 0000111=00000101=5$ vagyis $2\uparrow 5$ helyiértékű bit kerül gerjesztésre a 9-es című (relatív cím) byte -ban. Ha ezután még MJ=11. Akkor $2^5 \text{ OR } 2^4=00110000$ megoldással két pont is gerjeszthető stb.

ABC: -

HTZ: -

TEXAS: -

PRIMO: 122 >6Ø, ->9Ø törölve * >95 CLS * >1ØØ FOR MJ=1 TO 255 * >11Ø SET (MJ, MY) * >13Ø FOR MI=191 TO 1 STEP -1 * >14Ø SET (MX, MI) * >19Ø MI=MY+Z : IF MI <1 OR MI >191 THEN 21Ø * >2ØØ SET (MJ, MI) * >21Ø MJ=MJ+1 : IF MJ >255 THEN 25Ø * >215 NEXT X *

TV COM: 122 >11Ø MI=MY * >12Ø PLOT Ø, MI; 1Ø23, MI : REM A MIKRO KARAKTEREK Ø-1Ø23 (VIZSZ.) ÉS Ø-96Ø (FÜGG) KOORDINATÁKKAL AKTIVÁLHATÓK * >13Ø törölve * >14Ø MJ=MX * >15Ø PLOT MJ, 959; MJ,Ø * >16Ø MJ=MX+4*INT (A/S) : REM GRAPHICS 2 ESETÉN 2-ES; 16 ESETEN

VT-16: 122 >1Ø 'FÜGGVÉNYEK MAKRO GRAFIKONJA * MAKRO GRAFIKON AZ ADOTT OPERÁCIÓS RENDSZERBEN SZINEZHETŐ * >6Ø CLS * >7Ø COLOR Ø,7 * >8Ø, >9Ø törölve * >1ØØ FOR MJ=1 TO 8Ø : IF MJ=8Ø THEN 13Ø * >11Ø MI=MY : LOCATE MI, MJ : PRINT " " * >13Ø FOR MI=1 TO 24 : IF MI=24 THEN 16Ø * >14Ø MJ=MX : LOCATE MI, MJ : PRINT " " * >19Ø MI=MY-Z : IF MI <1 OR MI >24 THEN 21Ø * >2ØØ LOCATE MI, MJ : PRINT " " * >22Ø COLOR 7,15 * >23Ø törölve *

14. A grafikus üzemmódból a

STOP	RESTORE
------	---------

 leütésével, vagy a

POKE53265, 27 : POKE53272,21

utasításokkal térhetünk vissza a normál üzemmódra.

– Ezután tekintsük a P122 megoldását:

*

```
>NEW
>1Ø PRINT "***FUGGVENYEK MIKRO GRAFIKONJA**'"
>2Ø PRINT "18Ø-AS CIMKEVEL KOPOGJA LE A FUGGVENYNT
      : PL Z=X↑2, MAJD RUN 3Ø'" : STOP
>3Ø INPUT "AZ X TENGELY POZICIOJA?" ; MY
>4Ø INPUT "AZ Y TENGELY POZICIOJA?" ; MX
>5Ø INPUT "X INTERVALLUMA : A, F ES A LEPES HOSSZ : S ?"
      ; A, F, S
>6Ø POKE53265, 27 OR 32 : REM 32 GRAF.
>7Ø POKE53272, 16 OR 8 : REM 1 SZIN, 8 PONT
>8Ø FOR C=1Ø24 TO 2Ø23 : POKE C, 16*Ø+3 : NEXT
      C : REM SZINPAR Ø ; 3
>9Ø FOR C=8192 TO 16383 : POKE C, Ø : NEXT C :
      REM PONTTARAK KIURITESE
>1ØØ FOR MJ=Ø TO 319 : IF MJ=32Ø THEN 13Ø
>11Ø MI=MY : GOSUB 22Ø
>12Ø NEXT MJ : REM AZ X TENGELY RAJZOLASA
>13Ø FOR MI=Ø TO 199 : IF MI=2ØØ THEN 16Ø
>14Ø MJ=MX : GOSUB 22Ø
>15Ø NEXT MI : REM AZ Y TENGELY RAJZOLASA
>16Ø MJ= MX+INT (A/S)
>17Ø FOR X=A TO F STEP S : IF X > F THEN 25Ø
>18Ø REM Z A FUGGO VALTOZO!
>19Ø MI=MY-Z : IF MI < 1 OR MI > 198 THEN 21Ø
>2ØØ GOSUB 22Ø
>21Ø MJ=MJ+1 : IF MJ > 319 THEN 25Ø
>215 NEXT X : GOTO 25Ø
```

ABC: 123 Előzetes megjegyzések: 1. OK * 2. A felhasználói BASIC kezdő címkéjét a 65052 és 65053 rekeszek tárolják. Tartalmuk Ø és 192 * L a páros, H a páratlan című rekesz * 3. A programozó kiszolgálásában éppen következő rekesz címét a 65054 és 65055 című rekszek tárolják *

HTZ: 123 Előzetes megjegyzések: 1. OK * 2. Csak annyi igaz, hogy a felhasználói BASIC terület kezdő címkéjét a 16548 és 16549 rekeszek tárolják * 3. A programozó kiszolgálásában éppen következő rekesz címét a 16633 és 16634 című rekeszek tárolják *

TEXAS: 123 Az előzetes megjegyzések 1. kivételével érvénytelenek *

PRIMO: >22Ø – >24Ø törölve * >25Ø END* A képernyő sorok alulról felfelé számozódnak, az oszlopok balról jobbra * A képek >CMD SAVE SCREEN” CIM,” RET utasítással disk-re rögzíthetők és >CMD LOAD” CIM” RET olvassa vissza *


TV COM: 8-AS SZORZOT ALK * >19Ø MI=MY + 4*Z : IF MI <1 OR MI >96Ø THEN 21Ø : REM Z NAGYÍTÁSA MINDHÁROM GRAFIKÁBAN 4 * >2ØØ PLOT MJ, MI * >21Ø MJ=MJ + 4 : REM 2, 4, 8 A VIZSZ. LÉPÉS HOSSZ. IF MJ >1Ø23 THEN 25Ø*>22Ø, 23Ø, 24Ø törölve * >25Ø END*

VT-16: >25Ø COLOR 7,15 : END*

A kurzort ki kell vinni a képből, mielőtt új akciót kezdeményeznénk *

123 Az előzetes megjegyzéseket figyelmen kívül hagyjuk * A BASIC terület áthelyezésére nincs szükség * A >MERGE ”NEV” utasítás betölt lemezen fekvő és ASC kódban rögzített programot és egyesíti a gépben lévő programmal * Azonos címkék felülíródnak, ezért a gépben lévő program cím-

```

>220 P=8192+320*INT(MI/8)+(MI AND7)+8*INT(MJ/8)
>230 POKE P, PEEK(P) OR 2↑ ((7-MJ) AND 7)
>240 RETURN
>250 POKE53272, 27 : POKE53272, 21 : END
* >RUN →  Λ

```

- △△
- Az előzetes megjegyzések alapján a program alig igényel magyarázatot. Feltűnhet, hogy minden INPUT a grafikus üzemmódra való áttérés (>60) előtt szerepel, utána ugyanis a PRINT és INPUT szövegek nem olvashatók a képernyőn. A függvény ábrázolás érdekessége, hogy az X tengelyen nem a helyettesített X érték, hanem a STEP-pel szinkronban léptetett MJ sorozat (>210), az Y tengelyen pedig az X tengely pozíciójának a Z(X)-el modulált értéke jelenik meg.
 - A >220–>240 szubrutin oldja meg mini karakterek elérését (P) és a megfelelő mikro karakterek gerjesztését.
 - A program egyszerű módosításokkal alkalmas figurák és mozgó alakzatok ábrázolására, sőt a SPRITE technikát is előkészíti, amivel itt nem foglalkozunk.

062

Programok összefűzése. POKE44, PEEK(46) ; POKE 43, PEEK(45)

123

Töltse be kazettáról vagy lemezről a P10 programot és egyesítse a később betöltött P11-el úgy, hogy működőképes P123 álljon elő:

- △△
- Előzetes megjegyzések:
 1. A LOAD utasítás egyik funkciója, hogy törli a gépben élő programot.

ABC: —

HTZ: —

TEXAS: —

PRIMO: A BASIC terület kezdő címét a **16548** és **16549** című rekeszpár, a programozó kiszolgálásában éppen következő rekesz címét viszont a **16633** és **16634** című rekeszpár tárolja*

TV COM: AZ ELŐZETES MEGJEGYZÉSEK FIGYELMEN KIVÜL HAGYHATÓK* Az >OPEN#6 : INPUT "NEV" : POKE 2818,6 * ||OK OK|| >CLOSE#6 : INPUT betölt lemezen fekvő és ASC kódban rögzített programot és összefűzi a gépben lévő programmal*

VT-16: kékít >RENUM C1, C2 paranccsal a másik program címkéi alá vagy fölé toljuk (C2 címkét C1-el helyettesíti, a lépésköz pedig 1Ø), attól függően, hogy az egyesített programban melyik felel meg a monotonitásnak*

2. A rendszer nyilvántartás (0–1023 című rekeszek) területén szóló és páros rekeszek találhatóak. Köztük a 43–44-es a felhasználói BASIC első rekeszének, (normál állapotban 2048) a címét regisztrálja. PEEK(43)=1 ; PEEK(44)=8. A 43-as a LOWER (L), a 44-es a HIGH (H) szerepet viseli. Ez azt jelenti, hogy H a cím főrészét blokkban (1 blokk=2[↑]8 BYTE), az L pedig a töredéket BYTE-ban regisztrálja:

$$8 * 2^{\uparrow}8 + 1 = 8 * 256 + 1$$

A rekeszek töltése változtatható. Ha például az 1-et 1 hellyel balra visszük, akkor

L 00000001 43 H 00001000 44

L-ben 2, 4, 8 stb. BYTE, H-ban pedig 4K, 8K, 16K stb. címeltolódás regisztrálódik. Ez egyben a felhasználó munkaterületének a fokozatos csökkenését jelenti.

Ha megadjuk a terület kezdőcímét (K), akkor $H = \text{INT}(K/256)$; $L = K - H * 256$ a regiszterek tartalmát 10-es rendszerben adja meg. Pl. ha $K = 4000$, akkor $H = 15 = 00001111$ és $L = 160 = 10100000$. Ezen esetben 2048–4000-ig terjedő rekeszek nem vesznek részt a programozó kiszolgálásában.

3. A rendszer nyilvántartás területén fekvő másik fontos rekeszpár a 45–46-os. Ez a pár a 43–44-eshez hasonlóan azt a rekeszcímet tárolja, amely a programozó kiszolgálásában éppen következne a BASIC területen. Ha tehát eddig tolnánk ki a BASIC

ABC: 4. > POKE 65052, PEEK (65054) *
> POKE 65053, PEEK (65055) *
> NEW * ÁTCIMKÉZÉS: >REN α,10 RET * >LIST *

5. > POKE 65052, 0 *
> POKE 65053, 192 *

HTZ: 4. > POKE 16549, PEEK (16634) *
> POKE 16548, PEEK (16633) -2 *
> NEW *

5. > POKE 16548, 233 *
> POKE 16549, 66 *

TEXAS: -

PRIMO: 4. > POKE 16549, PEEK (16634) *
> POKE 16548, PEEK (16633) -2 *
> NEW *

5. > POKE 16548, 234 *
> POKE 16549, 67 *

TV COM: Azonos címkék felülíródnak, ezért a címkék logikai rendjét előzetesen biztosítjuk. Az eredmény program második összetevőjét címezzük át az első blokk utolsó címkéjét felülmúlva. A vezérlést átadó címkékre is megoldjunk *

VT-16: -

terület kezdő címét, például a LOAD kénytelen lenne a címét követő üres területet használni és nem érintené a korábban felhasznált területeket. A címtolás nem érinti a programozó manőverezési lehetőségeit, leszámítva a felhasználható terület nagyságát.

4. A címtolás a 43–44 rekeszpáron hajtható végre. A címtolás mértékét a 45–46 rekeszpár tartalma határozza meg, azaz a

>POKE44, PEEK(46)

RET

>POKE43, PEEK(45) –2

RET

>NEW

RET

utasítások a programozó munkaterületét át helyezik, miáltal az addig használt terület védetté válik, tartalmát megőrzi. A –2 a két terület láncolását biztosítja.

5. Ha programot viszünk be az új területre az futthat, rögzíthető, nyomtatható stb. Ha a védett területen program fekszik és az új területre bevitt programmal egyesíteni akarjuk, akkor az új programrész címkeit monoton emelkedő sorozatként kell megadnunk, illetve így kell átcímkezni, ha például betöltött programról van szó. Az egyesítést a BASIC kezdőcím visszaállításával

>POKE44, 8

RET

>POKE43, 1

RET

érhetjük el. Az egyesített program futtatható, módosítható, rögzíthető és törölhető.

ABC: >LOAD CAS : P10* >POKE65052, PEEK(65054)*
>POKE 65053, PEEK(65055)* >NEW*
>LOAD CAS : P11* >LIST* >REN40, 10 [RET]*
>LIST* >POKE65052, 0* POKE 65053, 192*
>LIST* A régi címkéket a REN törli*

HTZ: >CLOAD#-1, "P10" : LIST*
>POKE16549, PEEK(16634) : POKE16548, PEEK(16633) - 2 : NEW [NL]*
>CLOAD#-1, "Q11" : LIST* >RE40, 10 [NL]*
>LIST* POKE16548, 233 : POKE16549, 66 [NL]* >LIST*

TEXAS: Betöltjük kazettáról P11-et >OLD "CS1"* Átcímezzük P11-et, hogy P10
címkéit kövesse >RESEQUENCE 100* Disk-re rögzítjük P11-et >SAVE
DSK1. P11, MERGE* Kazettáról betöltjük P10-et >OLD "CS1"* Betöltjük
a disk-ről P11-et >MERGE DSK1. P11* >LIST* >RUN. MERGE=egyesít

PRIMO: >LOAD "P10" : LIST*
>POKE16549, PEEK(16634) : POKE16548, PEEK(16633) - 2 : NEW
[RET]* >LOAD "P11"* Átcímkezés EDIT módusban; régi címkék törlése*
>POKE16548, 234 : POKE16549, 67 [RET]* >LIST*

TV COM: P123 megoldása: >LOAD#6 : "P11" [RET]* ≧ÁTCIMKÉZÉS 40-60-ra*
RÖGZÍTÉS ASC kódban* >LOAD#6: "P10"* Töröljük az END-et*
>OPEN#6 : INPUT "P11" : POKE2818, 6 [RET]* >CLOSE#6 : INPUT*
Az összefűzés befejeződött*

VT-16: >LOAD "C : P10" [ENT] >LIST* >SAVE "C : P10", A [ENT]*
>LOAD "C : P11" [ENT] >LIST* >RENUM 40, 10 [ENT]* A régi
címké eltűnik* >MERGE "C : P10" [ENT] >LIST* >END törölve*

```
* >NEW
* >LOAD "P10", 1, C : LIST
```

```
|| 10 FOR I=1 TO 7 : READ A
|| 20 DATA 4, 5, 2, 3.85, 2E3, -4.15, -2E4
|| 30 PRINT A. : NEXT I : END
```

»TÖRÖLJUK END-ET

```
* >POKE44, PEEK(46)
* >POKE43, PEEK(45) -2
* >NEW
* >LOAD "P11", 1, C : LIST
```

RET
RET
RET

```
|| 10 FOR I=1 TO 7 : READ A%
|| 20 DATA 4, 5.2, 3.85, 2E3, -4.15, -2E4
|| 30 PRINT A% : NEXT : END
```

»ÁTCIMKÉZUNK 40, 50, 60-RA.

»TÖRÖLJUK A RÉGI CIMEKET : >10, 20, 30

RET

```
* >POKE44, 8
* >POKE43, 1
* >LIST
```

RET
RET

```
|| 10 FOR I=1 TO J : READ A
|| 20 DATA 4, 5.2, 3.85, 2E3, -4.15, -2E4
|| 30 PRINT A, : NEXT I
|| 40 FOR I=1 TO 7 : READ A%
|| 50 DATA 4, 5.2, 3.85, 2E3, -4.15, -2E4
|| 60 PRINT A% : NEXT I : END
```

```
* >RUN
```

ABC: [124] A P113 ABC-re érvényes alakjában a számozás más a REN 210, 10 után*
Illesztése: >240 törölve* >260 S(J) = B(J)* Célszerű a B(J) értékét nagyítani
pl. S(J)=5*B(J)*

HTZ: [124] Átcímkezni csak kiterjesztett BASIC-ben lehet (>SYSTEM [NL]*
>/12288 [NL] Erről (bekapcsolásról) mindig gondoskodunk*
RESEQUENCE = sorrendre* >RE 210, 10 [NL] címkezi az utasításokat
210, 220 stb.-re* OK*

TEXAS: [124] P49 és P113 egyesítése és illesztése ad megoldást* >OLD DSK1. P113*
>RESEQUENCE 200* >SAVE DSK1. P113, MERGE* >OLD DSK1.P49*
>MERGE DSK1.P113* LIST* Illesztés >65 NEXT J :: GOTO 200* I(N)
helyett A(N)* >250; >260; >270; >280 törölve* >290 FOR I=2 TO N ::
S(I)=B(I) :: IF S(I) >32 - D THEN 320*

PRIMO: [124] >CMDLOAD "P49" : LIST* >DELETE 10-150* P113 illesztése:
>250 törölve* >270 - >280 törölve* >290 CLS* >300 FOR K=1 TO N-1:
S(K)=10*B(L) : L=L+1* 10*B(L) nagyítás* >350 NEXTK : END* P49
illesztése: >65 NEXT J : L=2 : GOTO 210* >66 törölve*

TV COM: [124] >LOAD#6: "P49"* >LIST* Címke leolvasás* >LOAD#6: "P113"*
>LIST* Átcímkezés* >OPEN#6 : OUTPUT "P113" : LLIST#6: : CLOSE
OUTPUT* >LOAD#6: "P49"* >OPEN#6 : INPUT "P113" : POKE2818,6 :
CLOSE#6 : INPUT* >ILLESZTKÜK A KÉT PROGRAMOT (1, 2, 3; 1, 2)*

VT-16: [124] >LOAD "P49"* >SAVE "@: P49", A* >LOAD "P113"* >RENUM
210* >270 S=50*B(J) : J=J+1* B(J) 50-szeres nagyítással jobb képet ad*
>250 törölve* >330 CLOSE#4: RETURN* >MERGE "P49"* >65 NEXT
J : J=1 GOSUB 210* >66 END*

△△

– Ha az összefűzött programban egyező cím-
kék maradnának, vagy a címke sorozat
növekvő rendjét elhibáznánk, az egyesítési
folyamatot meg kell ismételni.

063


Programok felhasználása szubrutinként. Programok szerkesztése.

124

Nyomtassa ki a P49 bázisviszonyszámaait diagrammal együtt:

```
* >LOAD "P49", 8 : LIST
* >POKE44, PEEK(46)
* >POKE43, PEEK(45) -2
* >NEW

* >LOAD "P113", 8 : LIST
  »ÁTCHIMKÉZUNK (PL. 21Ø-35Ø)
  »TÖRÖLJUK A >1Ø ->15Ø-ET
  »ILLESZTJUK A KÉT PROGRAMOT : P113 SZUBRUTIN
    1. P113 >29Ø S=B(J) : J=J+1
    2. P113 >27Ø törölni
    3. P113 >35Ø CLOSE4 : RETURN

* >POKE44, 8
* >POKE43,1
* >LIST
  »ILLESZTJUK A KÉT PROGRAMOT : P49 FŐPROGRAM
    1. P49 >65 NEXT J : J=1 : GOSUB21Ø
    2. P49 >66 END
* >RUN →  Λ
```



Írjon alkalmazási programokat a könyvben szereplő és háttértárolón rögzített prog-
ramok összefűzésével:

ABC: Az 1. számú mellékletben található R0, R1, ... mini programokat, adaptálva vigye kazettára * A >NEW csak direkt utasításként használható R0 után *

HTZ: Az 1. számú mellékletben található R0, R1, ... mini programokat, adaptálva vigye kazettára: >PRINT# -1, "RI" : CSAVE# -1, "RI" NL betöltés: >10 INPUT# -1, N\$: IF N\$ <> "RI" THEN 10 * >20 CLOAD# -1, "NI" * >RUN * (I=0, 1, ...) * A rögzítés kettős címmel történik. Így a betöltés többkarakteres cím (pl. R42) azonosításával történhet *

TEXAS: Ha B(I) <1, célszerű S(I)=10*B(I) szerepeltetése >290-ben * A "programozó" disk-re az elemi szubrutinokat >SAVE DSK1. CIM, MERGE utasítással kell rögzíteni * Betöltésük >MERGE DSK1. CIM *

PRIMO: Az 1. számú mellékletben található R0, R1, ... mini programokat, adaptálva vigye disk-re * Bővítse a mini programtárát *

TV COM: Az 1. számú mellékletben található R1, R2, ... mini programokat, adaptálva vigye disk-re * A rögzítés ASC kódban történjen (OPEN#6 : OUTPUT "R1" : LLIST#6 :: CLOSE#6 : OUTPUT) * Bővítse a mini programtárát *

VT-16: Az 1. számú mellékletben található R1, R2, ... mini programokat, adaptálva vigye disk-re * A rögzítés ASC kódban történjen (>SAVE "R1", A). Ha Wdisk-et használ az sokkal gyorsabb *

Az eddig megismert 124 programot nyersanyagként tekinthetjük újabb programok felépítéséhez. Két vagy több programot is összefűzhetünk és ha kell kiegészíthetjük klaviatúráról. Figyeljünk arra, hogy címkét csak azon a BASIC területen módosíthatunk, amelyen a program behívása történt. Újabb feltételeket viszont bármikor beszúrhatunk meglévő címkék közé.

- A személyi számítógépek szűk keresztmetszetét a lassú INPUT (adatok és programok bevétele a gépbe) és OUTPUT (adatok és programok nyomtatása) képezi. Ezt részben feloldja a háttértárolókon rögzített programokból való építkezés, de ez a programkészlet alapos ismeretét tételezi fel.
- A kezdő programozók számára a "programozó" disk alkalmazását ajánlhatjuk. A programozó disk a gyakran előforduló és sok karakterből álló utasítások és elemi szubrutinok tárháza. Ilyen tármintát mutat be az 1-es számú melléklet. Az elemi szubrutinokat és utasításokat mini programokként, R0, R1, ..., RI, ... stb. névvel ellátva rögzítjük a disk-en. A mini programok >LOAD "RI", 8 utasítással tölthetők be a kijelölt BASIC területekre.
- Ha bármilyen programot megterveztünk fejben és már alapítottunk programozó disk-et akkor a program gépen való megírása az alábbi lépésekben történhet:

ABC: >LOAD CAS R0 * >RUN * >NEW *
>LOAD CAS R1 * >REN α , 10 * >RUN 104 helyett >103 törölve és >RUN *
Ha visszaálltunk a BASIC terület kezdőpontjára óvakodjunk utasítások tör-
lésétől *

HTZ: Az n_k pontos beállításától eltekinthetünk * A programépítés kazettás egy-
séggel lassú *

TEXAS: "R0" nem értelmezhető * Program építés: >MERGE DSK1. RI * RI a prog-
ram első blokkja. Kiegészítő utasítások lekopogása * >SAVE DSK1. PI,
MERGE * >MERGE DSK1. RJ * ÁTINDEXELÉS >RESEQUENCE α (PI) *
>SAVE DSK1. PJ, MERGE * És így tovább, majd >MERGE DSK1. PI *

PRIMO: >CMD LOAD "R0" : RUN * >CMD LOAD "RI" : LIST * >POKE 16548,234
: POKE 16549,67 : LIST : RUN * >CMD LOAD "RJ" : LIST * >POKE
16548,234 : POKE 16549,67 : LIST : RUN stb. * "R0" >101 POKE 16549,
PEEK(16634) * >102 POKE 16548, PEEK (16633) -2 *

TV COM: RI betöltése * Átindexelés * Rögzítés * RJ betöltése * Átindexelés * Rö-
gzítés * RK betöltése * Átindexelés * Rögzítése stb. * RI, RJ, RK, ... BETÖL-
TÉSE * Nem szereplő utasítások lekopogása (beszúrás), átírás, törlés * >RUN *

VT-16: Az RI, RJ, RK program felépítése * >LOAD RI * LIST * Utolsó címke C1 *
>LOAD "RJ" * >RENUM C1+10 * Utolsó címke C2 * >SAVE "@: RJ",
A * >LOAD "RK" * >RENUM C2+10 * >MERGE "RJ" * >MERGE "RI"
>LIST * >RUN * Az R0 program nem funkcionál *

```

* >LOAD "RØ", 8 : RUN
* >LOAD "RI", 8 : LIST
  »ÁTINDEXELÉS ÉS TÖRLÉS, PARAMÉTEREK EGYEZTETÉSE.
  »DISK-EN NEM SZEREPLŐ UTASÍTÁSOK LEKOPOGÁSA
* >POKE44, 8 : POKE43, 1 : LIST : RUN
*
* >LOAD "RJ", 8 : LIST
  »ÁTINDEXELÉS ÉS TÖRLÉS, PARAMÉTEREK EGYEZTETÉSE
  »DISK-EN NEM SZEREPLŐ UTASÍTÁSOK LEKOPOGÁSA
* >POKE44, 8 : POKE43, 1 : LIST : RUN
*
* >LOAD "RK", 8 : LIST ...
  ÉS IGY TOVÁBB A PROGRAM VÉGÉIG, MAJD
* >POKE44,8 : POKE43,1 : LIST
* >RUN 1Ø4

```

– Az RØ mini program:

```

* >1Ø1 POKE44, PEEK(46)
* >1Ø2 POKE43, PEEK(45) -2
* >1Ø3 NEW : STOP

```

a programok mindegyikének a bevezető eleme. Futtatása STOP-ig lehetséges, így akárhol tartunk a program építésben egy RUN a BASIC terület feltöltését oldhatja meg. Minden betöltést átindexelés és törlés követ. A mini programokban szabvány paraméterek szerepelnek. A program igényei szerint esetleg át kell írni, vagy aktuális értékekkel kell helyettesíteni őket. Ezt nevezzük egyeztetésnek. A programozó disk nem tartalmazhat apró utasításokat (pl. DIM, STOP, stb.). Ha ilyenekre van

ABC: OK *

HTZ: OK *

TEXAS: >MERGE DSK1. PJ * És így tovább* >LIST*Kiegészítések * >RUN *
Tehát betöltjük a logikai rendbe állított blokkokat, monoton átcímkezzük, kiegészítjük és rendre rögzítjük P1, PJ, ... névvel. Végül egymásután betöltjük MERGE utasítással. Ha a címkék rendje nem szigorúan monoton növekvő, a később betöltött felülírhatja a korábbi. A felépített program teszőlegesen módosítható, beleértve a törlést is. OK *

PRIMO: Az 1. számú mellékletben C-64-re érvényes programokat talál, ezeket először adaptálnia kell *

TV-COM: Az 1. számú mellékletben C-64-re érvényes programokat talál. Adaptálja és ASC kódban rögzítse őket *

VT-16: Az 1. számú mellékletben C-64-re érvényes programokat talál, ezeket először adaptálnia kell *
FONTOS: A gép elemei összedolgoznak INPUT-OUTPUT feladatok esetén*
Lemezről hívhatjuk a programot. A program INPUT igényét (adat) lemezzől veheti le (ha előbb odatettük). A program PRINT-jére lemezt, printert, képernyőt hívhatunk *

szükség a programokban, címkével együtt lekopogjuk. A címke mutathat a mini program elé (de a védett programrész mögé), közé és után. A mini programok között természetesen a P10–P123 is szóba jöhet. Ezekből akármilyen részek törölhetők az át-címkézés előtt, vagy után. Ugyanez vonatkozik az új utasításokkal való kiegészítésre is.

A programok géprevitele eszerint szerkesztési munkává egyszerűsödik. A szerkesztés annál egyszerűbb, minél bővebb a program tár és a "programozó" disk.

A program futtatása figyelmet igényel. Az egyszerű RUN az R0 programot aktiválja és CONT a STOP elhárítására ugyan alkalmas, de a programot nem hordozó BASIC területre vonatkozik és így REDY választ indukál. A programot tehát RUN104-el lehet elindítani. Erre az ismétléseknél is figyelni kell. R0 törlése is megoldható egy trükkel. Elegendő a REM szót a címkék után beszúrni. Ez más utasítások törlésére is alkalmazható.

Alapítson "programozó" disk-et és rögzítse az R0,..., R30,... mini programokat az 1. mellékletből. Szerkessze meg az eddig megismert P10–P122 programokat:

ABC: —

HTZ: —

TEXAS: —

PRIMO: —

TV COM: —

VT-16: Próbálja ki az alábbi rendszerkapcsolatokat: VT BASIC lemez az A floppy-ban*
Program és adatlemez a B floppy-ban* >LOAD "B : P62" * >10
N=10 : DIM A\$(N)* >SAVE "B : P62D" * >SAVE "C :P62D"
* >SYSTEM * >VTBASIC B : P62D > B : NEVEK
* >SYSTEM* >VTBASIC B : P62D <B : NEVEK >LPT1 *
SYSTEM* >VTBASIC B : P62D <C : MINEK >LPT1* SYSTEM*

LEGÓ RUTINOK C-64-RE.

R0	<pre>> 100 PRINT "BASIC POINTER FELTOLAS" > 10 POKE44, PEEK(46) > 102 POKE43, PEEK(45) -2 > 103 NEW : STOP</pre>
R1	<pre>> 100 PRINT "M*N TOMB BEOLVASASA" > 101 DIM A(M, N) > 102 FOR I=1 TO M : FOR J=1 TO N > 103 READ A(I, J) : PRINT A(I, J); : NEXT J, I > 104 DATA</pre>
R2	<pre>> 100 PRINT "M*1 TOMB BEOLVASASA" > 101 DIM A(M) > 102 FOR I=1 TO M > 103 READ A(I) : PRINT A(I); : NEXT > 104 DATA</pre>
R3	<pre>> 100 PRINT "M*N TOMB INPUTJA" > 101 DIM A(M, N) > 102 FOR I=1 TO M : FOR J=1 TO N > 103 INPUT "ADAT? "; A(I, J) > 104 PRINT A(I, J); : NEXT J, I</pre>
R4	<pre>> 100 PRINT "Y=F(X) ERTEKTABLAZATA" > 101 INPUT "X KORLATAI ES LEPESSKOZE"; A, F, S > 102 FOR X=A TO F STEP S > 103 REM Y= A FUGGVÉNY > 104 PRINT X ; Y, : NEXT</pre>

R5 >100 INPUT "N KEREKITSE P TIZEDESRE" ; N, P
>101 X=N : X=INT (10↑P*X+.5)/10↑P
>102 PRINT X

R6 >100 PRINT "ADATSOR OSSZEGE"
>101 FOR I=1 TO M : H=H+A(I)
>102 NEXT : PRINT H

R7 >100 PRINT "R2 ES R6 ALKALMAZASA"
>200 PRINT "M*1 TOMB BEOLVASASA"
>201 DIM A(9)
>202 FOR I=1 TO 9
>203 READ A(I) : PRINT A(I) ; : NEXT
>204 DATA 5, 7, 4, 8, 3, 9, 1, 9, 6
>300 PRINT "ADATSOR ÖSSZEGE"
>301 FOR I=1 TO 9 : H=H+A(I)
>302 NEXT : PRINT H

R8 >100 PRINT "ADATSOR SZORZATA"
>101 H=1
>102 FOR I=1 TO M : H=H*A(I)
>103 NEXT : PRINT H

R9 >100 PRINT "VEKTOROK SKALARIS SZORZATA"
>101 FOR I=1 TO M : H=H+A(I)*B(I)
>102 NEXT : PRINT H

R10 >100 PRINT "MATRIXOK ÖSSZEGE"
>101 FOR I=1 TO M : FOR J=1 TO N
>102 A(I, J) = A(I, J) + B(I, J) : PRINT A(I, J);
>103 NEXT J, I

R11

```
>100 PRINT "HASONLO EGYBDEK MEGSZAMLALASA"  
>101 FOR I=1 TO M : IF A(I)<B THEN N=N+1  
>102 NEXT : PRINT N
```

△△

Az R11 a B számnál kisebb elemeket számolja meg. Ha IF alatt $B \leq A(I) \leq C$ szerepelne, akkor a [B, C] zárt intervallumba eső adatok számát mutatná ki. Az egész számokat az $INT(A(I))=A(I)$, a D-vel osztható számokat az $INT(A(I)/D)=A(I)/D$, az egyenlő hosszúságú szavakat a $LEN(A\$(I))=B$ stb. logikai függvények segítségével számolná.

R12

```
>100 PRINT "OSZTALYOZAS ES SZAMLALAS"  
>101 FOR I=1 TO M  
>102 IF A(I)<B THEN N=N+1 : GOTO 105  
>103 IF A(I)=B THEN P=P+1 : GOTO 105  
>104 R=R+1  
>105 NEXT : PRINT N, P, R
```

R13

```
>100 PRINT "A(M, N) ROGZITese KAZETTARA"  
>101 OPEN2, 1, 2, "FILE NEV"  
>102 FOR I=1 TO M : FOR J=1 TO N  
>103 PRINT#2, A(I, J) : NEXT J, I  
>104 CLOSE2
```

R14

```
>100 PRINT "A(M, N) BETOLTESE KAZETTAROL"  
>101 OPEN2, 1, 0, "FILE NEV" : DIM A(M, N)  
>102 FOR I=1 TO M : FOR J=1 TO N  
>103 INPUT#2, A(I, J) : PRINT A(I, J);  
>104 NEXT : CLOSE2
```

△△

R13 és R14 számsor vagy számoszlop esetén is alkalmazható, ha M, illetve N-et 1-gyel azonosítjuk.

R15 >100 PRINT "REKORDOK (A, B, C ADATOKKAL) ROGZITese
KAZETTARA"
>101.OPEN2, 1, 2, "FILE NEV" : X\$=CHR\$(13)
>102 FOR I=1 TO M
>103 PRINT#2, A ; X\$B ; X\$C ; X\$
>104 NEXT : CLOSE2

R16 >100 PRINT "SEQ FILE ROGZITese DISK-EN"
>101 OPEN6, 8, 6, "Ø : FILE NEV, S, W"
>102 FOR I=1 TO M
>103 PRINT#6, A(I)
>104 NEXT : CLOSE6

R17 >100 PRINT "SEQ FILE BETOLTESE DISK-ROL"
>101 OPEN6, 8, 6, "Ø : FILE NEV, S, R"
>102 FOR I=1 TO M
>103 INPUT#6, A(I) : PRINT A(I);
>104 NEXT : CLOSE6

R18 >100 PRINT "ISMERETLEN HOSSZUSAGU ADATSOR
SEQ BETOLTESE DISK-ROL"
>101 OPEN6, 8, 6, "Ø : FILE NEV, S, R" : DIM B(255)
>102 INPUT#6, A : K=K+1 : B(K)=A : PRINT A;
>103 IF ST <> 64 THEN 102
>104 CLOSE6

R19 >100 PRINT "TOBBTAGU REKORDOK SOROS ROGZITese
DISK-EN"
>101 OPEN6, 8, 6, "Ø : FILE NEV, S, W" : X\$=CHR\$(13)

R19 >102 FOR I=1 TO M
>103 PRINT#6, A ; X\$B ; X\$C
>104 NEXT : CLOSE6

R20 >100 PRINT "HIBAINFORMACIO A LEMEZEGYSEGROL"
>101 OPEN15, 8, 15
>102 INPUT#15, EN, EM\$, ET, ES
>103 IF EN>1 THEN PRINT EN, EM\$, ET, ES
>104 CLOSE15 : RETURN

R21 >100 PRINT "POINTER ALLITAS FLOPPY-N"
>101 OPEN15, 8, 15
>102 H=INT (I/256) : L=I-H*256
>103 PRINT#15, "P"+CHR\$(6)+CHR\$(L)+CHR\$(H)+
CHR\$(1) : CLOSE15 : RETURN

R22 >100 PRINT "RELATIV FILE MEGNYITASA MUNKARA"
>101 OPEN15, 8, 15 : OPEN6, 8, 6, "FILE NEV"
>102 RETURN

R23 >100 PRINT "FUTAS SZABALYOZAS"
>101 X\$ = " " : GETX\$: IF X\$ = " " THEN 101
>102 IF X\$ = "N" THEN END

△△

A 101-et >101 INPUT X\$ is helyettesí-
heti, ha X\$-nek más funkciója nincs a prog-
ramban.

R24 >100 PRINT "LEXIKOGRAFIA"
>101 FOR I=1 TO M-1 : FOR K=I+1 TO M
>102 IF B\$(I)<=B\$(K) THEN 104

R24 >103 S\$=B\$(I) : B\$(I)=B\$(K) : B\$(K)=S\$
>104 NEXT K
>105 PRINT B\$(I) : NEXT I

R25 >100 PRINT "CIM NYOMTATASA"
>101 OPEN4,4
>102 PRINT#4, "CIM"
>103 CLOSE4

R26 >100 PRINT "SZAMOK JOBBRA IGAZITASA"
>101 X\$=" "
>102 FOR I=1 TO M
>103 N\$(I)=RIGHT\$(X\$+STR\$(A(I),H)
>104 PRINT N\$(I) : NEXT

R27 >100 PRINT "SZAVAK BALRA IGAZITASA"
>101 X\$=" "
>102 FOR I=1 TO M
>103 N\$(I)=LEFT\$(A\$(I)+X\$,H)
>104 PRINT N\$(I) : NEXT

△△

R26 és R27 printerre is alkalmazható az OPEN4,4, CLOSE4 és (PRINT)#4 kiegészítéssel.

R28 >100 PRINT "POINTER ALLITASA PRINTEREN"
>101 H=INT(Y/256) : L=Y-H*256
>102 PRINT#4, CHR\$(27)+CHR\$(16)+CHR\$(H)+CHR\$(L) ". "

R29 >100 PRINT "MI, MJ MIKRO KARAKTER KIGYUJTASA A
KEPERNYON"

```

R29      >101 P=8192 + 230*INT(MI/8) + (MI AND 7) + 8*INT (MJ/8)
          >102 POKE P, PEEK(P) OR 2*((7-MJ) AND 7)
          >103 RETURN


---


R30      >1 PRINT "BASIC POINTER ALAPALLASA"
          >2 POKE44, 8 : POKE 43,1


---


R31      >100 PRINT " LEGO PROGRAMOZAS RENDJE"
          >101 REM : LOAD "R0", 8 : RUN
          >102 REM : LOAD "RI", 8 : LIST : ATINDEXELES :
              REGI INDEX TORLES : AKTUALIZALAS
          >103 REM : POKE44, 8 : POKE43, 1 : LIST : RUN
          >104 REM : LOAD "RJ", 8 : LIST : ATINDEXELES : REGI INDEX
              TORLESE : AKTUALIZALAS
          >105 REM : POKE44,8 : POKE43,1 : LIST : RUN
          >105 REM : LOAD "RK", 8 : LIST : ATINDEXELES :
              REGI INDEX TORLES : AKTUALIZALAS
          >107 REM : POKE44, 8 : POKE43, 1 : LIST
          >108 REM : UTASITASOK BESZURASA 103-NAL NAGYOBB
              CIMKEVEL
          >109 REM : HA A PROGRAM KESZ, AKKOR RUN104,
              ELLENKEZO ESETBEN RUN


---


R32      >100 PRINT "J-EDIK HO I-EDIK NAPJA AZ EV D(I, J)-EDIK
              NAPJA"
          >101 DIM D(31, 12) : N=1
          >102 FOR J=1 TO 12
          >103 IF J=1 OR J=3 OR J=5 OR J=7 OR J=8
              OR J=10 OR J=12 THEN M=31 : GOTO 106
          >104 IF J <> 2 THEN M=30 : GOTO 106
          >105 M=28 : REM 29
          >106 FOR I=1 TO M : D(I, J)=N : N=N+1 : NEXT I
          >107 NEXT J : INPUT "J, I?" ; J, I : PRINT D(I, J)


---



```

R33 >100 PRINT "I-EDIK HO J-EDIK NAPJAN BETETT T TOKE
P%-OS KAMATA EV VEGEIG"
>101 INPUT "I, J, T, P? "; I, J, T, P : N=(12-I)*30+30-J
: K = T*P*N/36000
>102 PRINT "I="I, "J="J, "T="T, "P="P, "KAMAT="K

R34 >100 PRINT "SZOVEG ATTETELE ASC KODBA"
>101 INPUT "SZOVEG"; A\$: N=LEN(A\$) : PRINT N
>102 FOR C=1 TO N
>103 IRAS=ASC(MID\$(A\$, C, 1)):PRINT IRAS ; : NEXT C
: GOTO 101

R35 >100 PRINT "N ELEMU MINTA VETELE AZ (A, B)
INTERVALLUMBOL"
>101 INPUT "N, A, B? "; N, A, B
>102 FOR I=1 TO N : X=A+(B-A)*RND(0) : PRINT X ; : NEXT

R36 >100 PRINT "VEGTELEN SOROZAT KOZELITO HA
TAREKTEKE"
>101 PRINT "A 101 CIMRE IRJA BE A SOROZATOT
PL. DEF FNA(N)=(1+1/(N+1))^(N+1)"
>102 N=1
>103 N=N+1 : IF FNA(N)<=FNA(N-1) AND FNA(N+1) <= FNA(N)
THEN 106
>104 IF FNA(N) >= FNA(N-1) AND FNA(N+1) >= FNA(N)
THEN 106
>105 PRINT "A SOROZAT NEM MONOTON"
>106 IF ABS(FNA(N+1)-FNA(N)) < 1 E-6 OR N >= 1E6 THEN
PRINT FNA(N) : END ELSE 103

ΔΔ

A DEF FN jellel definiált A(N) függvény helyettesítési értékét a gép kiszámítja anélkül, hogy a vezérlést 101-re adnánk át, elegendő az argumentumot beírni pl. A(100).

R37 >100 PRINT "VEGES MERTANI SZORZAT N-EDIK TAGJA ES AZ
 ELSON TAG ÖSSZEGE"
 >101 INPUT "A1, Q, N?"; A1, Q, N
 >102 PRINT "AN="A1*Q↑(N-1) CHR\$(13) "SN="A1*(Q↑N-1)/
 (Q-1)

R38 >100 PRINT "PARALLELOGRAMMAK ES HAROMSZOGEK
 TERULETE KET OLDAL ES ALTALUK ZART SZOGBOL"
 >101 INPUT "A, B, S SZOG?"; A, B, S
 >102 PRINT "PT="A*B*SIN(S*π/180), "HT="A*B*
 (SIN(S*π/180))/2

R39 >100 PRINT "M*N ES N*R TIPUSU TOMBOK SZORZATA"
 >101 I=1
 >102 K=1
 >103 J=1
 >104 F(J)=A(I, J)*B(J, K) : C(I, K) = C(I, K) + F (J)
 >105 J=J+1 : IF J<= N THEN 104
 >106 PRINT C (I, K) : K=K+1 : IF K<=R THEN 103
 >107 I=I+1 : IF I<= M THEN 102

R40 >100 PRINT "N*N TIPUSU MATRIX INVERZE"
 >101 FOR I=1 TO N
 >102 FOR J=1 TO N : IF A(I, J) = 0 THEN NEXT J
 >103 FOR K=1 TO N : C(K)=A(I, K) : A(J, K) = C(K)
 >104 C(K)=B(I, K) : B(J, K) = C(K) : NEXT K : D=1/A(I, I)
 >105 FOR K=1 TO N : A(I, K)=D*A(I, K) : B(I, K) = D*B(I, K)
 : NEXT K
 >106 FOR L=1 TO N : IF L=J THEN 108
 >107 E = - A(L, J) : FOR K=1 TO N : A(L, K) = A(L, K) + E*A(I, K)
 : B(L, K) = B(L, K) + E*B(I, K) : NEXT K
 >108 NEXT L, I
 >109 FOR I=1 TO N : FOR J=1 TO N : PRINT B(I, J) : NEXT J;
 : PRINT : NEXT I

-
- R41 >100 PRINT "M*N MERETU TOMB NYOMTATASA"
 >101 OPEN4,4 : REM SOROK NYOMTATASA
 >102 FOR I=1 TO M
 >103 FOR J=1 TO N : B(J)=A(I, J) : NEXT J :
 PRINT#4, B(1); B(2); . . . ; B(N)
 >104 NEXT I : CLOSE4
-
- R42 >100 PRINT "SULYOZOTT SZAMTANI ATLAG"
 >101 FOR I=1 TO M : INPUT "X ADAT, S GYAKORISAG?" ; X, S
 >102 A=A+X*S : B=B+S : NEXT
 >103 PRINT "ATLAG=" A/B
-
- R43 >100 PRINT "ATLAGOS FEJLODESI MUTATO"
 >101 FOR I=1 TO M : INPUT "P FEJLODESI RATA,
 S GYAKORISAG?" ; P, S : G=1
 >102 G=G*P↑S : N=N+S : NEXT : PRINT "MUTATO=" G↑(1/N)
-
- R44 >100 PRINT "A+BX EGYENES ILLESZTESE PONTSORHOZ"
 >101 INPUT "A PONTOK SZAMA?" ; N
 >102 FOR I=1 TO N : INPUT "X, Y KOORD?" ; X, Y : U=U+X
 : V=V+Y : K=K + X↑2 : L=L+Y↑2
 >103 S=S+X*Y : NEXT : U=U/N : V=V/N : PRINT U, V, K, L, S
 >104 B=(S-N*U*V)/(K-N*U↑2) : A=V - B*U :
 PRINT A "+" B " * X=Y"
 >105 PRINT "R =" (S-N*U*V)/SQR((K-N*U↑2) * (L-N*V↑2))
-
- R45 >100 PRINT "KEY INPUT, DISK ES PRINTER OUTPUT"
 >101 INPUT "ADATSZAM?" ; N : DIM A(N) : FOR I=1 TO N
 : INPUT "ADATOK?" ; A(I) : NEXT
 >102 OPEN6, 8, 6, "0" : FILE CIM, S, W" : FOR I=1 TO N : PRINT#6,
 A(I) : NEXT : CLOSE6
 >103 OPEN4,4 : PRINT#4, "CIM" : FOR I=1 TO N : PRINT#4,
 I ; A(I) : NEXT : CLOSE4
-

R46 >100 PRINT "DISK INPUT, PRINTER OUTPUT"
>101 OPEN6, 8, 6, "FILE CIM, S, R" : DIM A(1000)
>102 INPUT#6, A : A(I) = A : I = I + 1 : IF ST <> 64 THEN 102
>103 CLOSE6 : OPEN4,4 : FOR J=0 TO I-1 : PRINT#4, J ; A(J)
: NEXT J : CLOSE4

R47 >100 PRINT "FIX SZERKEZETU STRINGEK BEVITELE"
>101 INPUT "REKORDSZAM? " ; M : DIM R\$(M)
>102 PRINT "Δ-TOL Δ-IG IRHAT EGY ADATOT"
>103 PRINT " **Δ*****Δ*****Δ"
>104 FOR I=1 TO M : INPUT R\$(I) : NEXT
>105 OPEN4,4 : FOR I=1 TO M : PRINT#4, R\$(I) : NEXT : CLOSE4

R48 >100 PRINT "F LOGIKAI SZAMMAL, M REKORD (C KARAKTER
HOSSZU) ROGZITese DISK-EN"
>101 INPUT "F SZAM, C HOSSZ, M REKORDSZAM?"; F, C, M
>102 OPEN15, 8, 15 : OPEN F, 8, 6, "NEV, L,"+CHR\$(C)
>103 FOR I=1 TO M : GOSUB 106
>104 PRINT# F, R\$(I) : NEXT : END
>105 CLOSE F : CLOSE15
>106 H=INT (I/256) : L = I - H*256
>107 PRINT#15, "P" + CHR\$(6) CHR\$(L) CHR\$(H) CHR\$(1)
: RETURN

R49 >100 PRINT "FIX SZERKEZETU STRINGEK ÖSSZETEVOKRE
BONTASA"
>101 INPUT "REKORDSZAM M, OSSZETEVOK SZAMA
N? " ; M, N : DIM X\$(M, N)
>102 FOR I=1 TO N : INPUT "KEZDO POZICIO ES HOSSZ AZ
ÖSSZETEVOKNEL?"; P(I), C(I) : NEXT
>103 FOR J=1 TO M : FOR I=1 TO N : X\$(J, I) = MID\$(R\$(J), P(I),
C(I))
>104 PRINT X\$(J, I) ; : NEXT I, J

R50

```
>100 PRINT "8-AS FLOPPY 9-ESRE ALLITASA"  
>101 OPEN15, 8, 15  
>102 PRINT#15, "M-W" CHR$(119) CHR$(0) CHR$(2)  
      CHR$(41) CHR$(73)  
>103 CLOSE 15
```

△△

A LEGO programtárból csak mintát adtunk. A gyakorlati munka során keletkező programrészletek közül azokat, amelyek gyakori előfordulására számíthatunk, mindig rögzítjük a LEGO DISK-en. A programozási feladatok változatossága miatt a LEGO programok nem illenek be mindig eredeti összetételükben, de törlés, átírás, beszúrás révén alkalmassá tudjuk tenni. Az is előfordul, hogy ugyanezt a LEGO rutint többször is behívjuk és törlések után más-más részletét építjük be az adott helyre.

2. UTASÍTÁSKÉSZLET

Ha a személyi számítógépek változóit, műveleteit, függvényeit, parancsait és utasításait gépenként adnánk meg, gyakran ismétlésekbe bocsájtkoznánk. Ezt el akarjuk kerülni oly módon, hogy az egyező elemeket egyszer írjuk le, kiemelve az esetleges kivételeket. A parancsok és utasítások esetén ez hármas tagozódást eredményez:

egyező utasítások,
egyező funkciót betöltő utasítások,
géptípusra jellemző utasítások.

Az egyező utasítások a megengedett kiegészítések által eltérő, illetve más hatókörű funkciót nyerhetnek géptípusonként, ezért a második csoportban is megjelenhetnek. Az egyező funkciót betöltő utasításokat a funkcióhoz kapcsolva mutatjuk be. A géptípusra jellemző utasításokat, a teljesség igénye nélkül, hatókörük jelzésével soroljuk fel.

1. Adatok és azonosítók

A személyi számítógépek az alábbi adatokat képesek kezelni:

- Egész számok (kiv: TEXAS, TV COM). Azonosításuk $A\%, \dots, Z\%$; $A0\%, \dots, A9\%, \dots, Z0\%, \dots, Z9\%$ változókkal történik.
- Valós számok. Azonosításuk $A, \dots, Z; A0, \dots, A9; \dots; Z0, \dots, Z9$ változókkal történik.
- A valós számok egyszeres (≤ 7 számjegyű), vagy kétszeres (≥ 8 számjegyű) pontosságúak lehetnek (kiv: C-64, ABC, TEXAS, TV COM). Azonosításuk $A!, \dots, Z!; A0!, \dots, Z9!$, illetve $A\#, \dots, Z\#; A0\#, \dots, Z9\#$ változókkal történik.
- A számok eredeti (fixpontos pl. 530.25) és normál (lebegőpontos pl. 5.3025E2; 5.3025D2) alakban vihetők be és kaphatók output-ként.
- A számok 10-es és 16-os (HEXADECIMÁLIS)¹⁶ rendszerben vihetők be a gépekbe. Az utóbbi a TEXAS és VT-16-ra jellemző (pl. &H2457).
- Karakter konstansok (STRING-ek). Azonosításuk $A\$, \dots, Z\$, A0\$, \dots, Z0\$$ (kiv: ABC, ahol $A\$, \dots$) változókkal történik.
- A karakter konstansok lehetnek szöveg, (pl. "BABA") és szám, (pl. "5.3025") jellegűek.
- Tömbök (számok vagy STRING-ek véges hosszúságú (oszlopa), táblázata, vagy táblázat sora). Azonosításuk a tartalom jellege és típusa szerint $A\%(N)$, $B(N)$, $C\#(N)$, $D\$(N)$; $E\%(M, N)$, $B(M, N)$, $C\#(M, N)$, $D\$(M, N)$ stb.

2. Numerikus műveletek.

Összeadás	műveleti jele +	és ADD ☒	(ABC-80)
Kivonás	műveleti jele -	és SUB ☒	(ABC-80)
Szorzás	műveleti jele *	és MUL ☒	(ABC-80)
Osztás	műveleti jele /	és DIV ☒	(ABC-80)
Hatványozás	műveleti jele ^, ↑, **		
Gyökvonás: = törtkitevős hatványozás.			

– A számok műveletei ismertek. A számtömbök műveleteire a mátrixok (vektorok) szabályai érvényesek.

¹⁶ Részletesebben a 4. számú mellékletben.

3. Számok és string-ek viszonya

$A < B$, $A \leq B$, $A = B$, $A \geq B$, $A > B$, $A <> B$, $A \$ < B \$$, $A \$ \leq B \$$, $A \$ = B \$$, $A \$ \geq B \$$, $A \$ > B \$$, $A \$ <> B \$$. Két szám vagy két string viszonya tehát kisebb ($<$), kisebb vagy egyenlő (\leq), egyenlő ($=$), nagyobb vagy egyenlő (\geq), nagyobb ($>$) és nem egyenlő ($<>$) lehet.

4. Logikai függvények

Két szám vagy string között kijelölt viszony függvényként viselkedik a számítógépben. Ha ugyanis a kijelölt viszony igaz, akkor a függvény -1 , ha nem igaz, akkor \emptyset értéket vesz fel, azaz

$$\begin{array}{ll}
 Y=(A < B); & Y = \begin{cases} -1, & \text{ha } A < B \\ \emptyset, & \text{ha } A \geq B \end{cases} & Y=(A > B); & Y = \begin{cases} -1, & \text{ha } A > B \\ \emptyset, & \text{ha } A \leq B \end{cases} \\
 Y=(A \leq B); & Y = \begin{cases} -1, & \text{ha } A \leq B \\ \emptyset, & \text{ha } A > B \end{cases} & Y=(A \geq B); & Y = \begin{cases} -1, & \text{ha } A \geq B \\ \emptyset, & \text{ha } A < B \end{cases} \\
 Y=(A = B); & Y = \begin{cases} -1, & \text{ha } A = B \\ \emptyset, & \text{ha } A <> B \end{cases} & A=(A <> B); & Y = \begin{cases} -1, & \text{ha } A <> B \\ \emptyset, & \text{ha } A = B \end{cases}
 \end{array}$$

– Két vagy több logikai függvény logikai kifejezéssel fűzhető össze az AND, OR, XOR, NOT stb. "műveleti" jelekkel. A logikai kifejezés is kétértékű az alábbiak szerint:

$$\begin{array}{ll}
 Y=(Y1 \text{ AND } Y2); & Y = \begin{cases} -1, & \text{ha } Y1 = -1 \text{ és } Y2 = -1 \\ \emptyset, & \text{egyéb esetekben} \end{cases} \\
 Y=(Y1 \text{ OR } Y2); & Y = \begin{cases} \emptyset, & \text{ha } Y1 = \emptyset \text{ és } Y2 = \emptyset \\ -1, & \text{egyéb esetekben} \end{cases} \\
 Y=(Y1 \text{ XOR } Y2); & Y = \begin{cases} -1, & \text{ha } Y1 = -1 \text{ és } Y2 = \emptyset \text{ vagy } Y1 = \emptyset \text{ és } Y2 = -1 \\ \emptyset, & \text{egyéb esetekben} \end{cases} \\
 Y = \text{NOT } Y1; & Y = \begin{cases} -1, & \text{ha } Y1 = \emptyset \\ \emptyset, & \text{ha } Y1 = -1 \end{cases}
 \end{array}$$

– Eszerint a -1 -es kimenethez AND esetén minden, OR esetén legalább egy és XOR esetén csak egy összetevő -1 -es kimenete felel meg, (pl. $(2 < 3 \text{ AND } 3 > 5) = \emptyset$, de $(2 < 3 \text{ OR } 3 > 5) = -1$ és $(2 < 3 \text{ XOR } 3 > 5) = -1$, ugyanakkor $(2 < 3 \text{ AND } 3 < 5) = -1$, $(2 < 3 \text{ OR } 3 < 5) = -1$ és $(2 < 3 \text{ XOR } 3 < 5) = \emptyset$). A logikai függvények lényegében számok, így numerikus műveletekben is résztvehetnek (pl. $2 + (3 < 5) * 12 + 4 * (7 < 4)$ stb.). Fő alkalmazási területük az ágaztatás, ahol $A - 1$ vonzata a THEN, a nulláz az ELSE ág.

5. Matematikai függvények

– Számításainkban gyakran előforduló elemi függvényeket definiáltak a számítógépek mindegyikére (pl. DEF ABS(X)=, DEF COS(X)=, stb.). Így elegendő a független változót a függvény nevébe behelyettesíteni, hogy a megfelelő függvény értéket nyerjük (pl. ABS (-2)=2; COS (3.14)= -1 stb.). Standard függvények az alábbiak:

ABS(X)	x szám abszolút értéke.
ATN(X)	y radián tangense x.
COS(X)	x radiános szög cosinusa.
EXP(X)	e alap x-edik hatványa ($e \approx 2,71$).
INT(X)	a legnagyobb egész szám x-től balra.
LOG(X)	x szám e alapú logaritmus. Visszakeresés: EXP (LOG(X)) = X.
RND(X)	egyenletes eloszlású véletlen szám a (0,1) intervallumból.
SGN(X)	x szám fekvését jelzi a számegyenesen: -1, ha $x < 0$, +1, ha $x > 0$ és 0, ha $x = 0$.
SIN(X)	x radiános szög sinusa.
SQR(X)	x szám pozitív négyzetgyöke.
TAN(X)	x radiános szög tangense.

– A matematika függvények numerikus jellegűek, így műveletekre is alkalmasak (pl. $3+5 * \text{SIN}(X)$). Fontos körülmény, hogy az argumentum (X) függvény is lehet (pl. $\text{EXP}(-X \uparrow 2)$), SQR ($1 - \text{SIN}(X) \uparrow 2$) stb.).

6. STRING változós függvények

STRING-ek alapítását, konverzióját és darabolását a számítógépek STRING függvényekkel oldják meg. Ezek az alábbiak:

YS="KARAKTEREK" ..	STRING-ek alapítása karakterekből.
Y\$=STR\$(X)	x számból STRING számot képez (kiv.: ABC, ahol Y\$=NUM (X)).
Y\$=CHR\$(K)	k ASC kódhoz karaktert rendel.
Y\$=STRING\$(N, K)	k ASC kódhoz n elemű STRING-et rendel. (Kiv.: C-64; TEXAS; Y\$=RPT\$(N, K)).
Y=VAL(X\$)	Ha X\$ "szám", valós számmá változtatja.
Y=ASC(X\$)	X\$ első karakteréhez tartozó ASC kódot állítja elő (kiv.: TV COM; Y=ORD(X\$)).
Y\$=MID\$(X\$, J, N)	X\$-ből n karaktert emel ki a j-edik pozícióval kezdve (kiv.: TEXAS; Y\$=SEG\$(X\$, J, N). TV COM; Y\$=X\$(J : J + N - 1)).
Y\$=RIGHT\$(X\$, N) ...	X\$ utolsó n karakterét emeli ki (kivételek: ABC; Y\$ = MID\$(X\$, K-N+1, N), ahol k a string hossza. TEXAS; Y\$=SEG\$(X\$, K-N+1, N). TV COM; Y\$=X\$(K-N+1 :)).
Y\$=LEFT\$(X\$, N)	X\$ első n karakterét emeli ki (kiv.: TV COM; Y\$=X\$(:N).
Y = LEN(X\$)	X\$ karaktereit számlálja meg.

7. Memória függvények

Y=FRE(\emptyset)	a BASIC terület szabad kapacitását méri (kiv.: ABC, TEXAS; Y=SIZE, TV COM; Y=FREE).
Y=PEEK(R)	a memória R című rekeszének a tartalmát adja.
Y=POKE R, X	x számot ($x \leq 255$) helyezi el az R című rekeszbe.

8. Egyező utasítások

CLOSE	lezárja a nyitott FILE-t.
DATA	a READ adattára.
DELETE C1-C2	törli a program C1-től C2-ig címzett utasításait (kiv.: C-64, ABC).
DIM	tömbök méretét közli a géppel.
END	leállítja a program futását, törli változóit a munkaterületen és lezárja a nyitott FILE-okat.
FOR-TO-STEP-NEXT	utasítások ismételt végrehajtását szervezi.
GOSUB C-RETURN	vezérlést a C címkével kezdődő szubrutinra adja át majd visszatér a következő címkére.
GOTO C	a vezérlést a C címkére adja.
IF Y THEN-ELSE	a vezérlést az Y logikai függvény felvett értéke alapján adja át.
INPUT	Adatot hív be a billentyűzetről, illetve háttér tárolóról.
LET	változókhoz értéket rendel.
LIST C1-C2	programot listáz ki képernyőre, illetve printerre (kiv.: PRIMO, TV COM, VT-16; LLIST C1-C2).
LOAD	programot tölt be háttér tárolóból (kiv.: HTZ; CLOAD).
NEW	törli a BASIC terület tartalmát.
ON V GOSUB C1-RETURN	V=1,2, ... felvett értéke szerint átadja a vezérlést a C1, C2, ... címkével induló szubrutinra, majd visszatér.
ON V GOTO C1,	V=1, 2, ... felvett értéke szerint C1, C2, ... címkére adja a vezérlést.
OPEN	FILE-t, illetve csatornát nyit meg háttértárolón.
PRINT	adatokat és függvényeket jelenít meg, illetve rögzít háttér tárolókon.
PRINT USING" "	adatokat előírt (" ") formátumban jelenít meg, illetve rögzít. (Kiv.: C-64, ABC).
RANDOMIZE	a véletlen szám generátort indulás előtt pozicionálja (kiv.: C-64).
REM	megjegyzések tára. A program futásban nem vesz részt.
READ	változókat azonosít DATA-beli adatokkal.
RESTORE C	C alatti DATA adattár ismételt felhasználása READ-hez.
RUN C	program végrehajtását indítja a C címkétől, illetve C hiányában az első utasítással kezdve.
SAVE	programot rögzít háttér tárolóra (kiv.: HTZ; CSAVE).
STOP	megszakítja a program futását.

9. Egyező funkciót betöltő utasítások

a) Program címkéző és átcímkéző parancsok.

C-64	-	-
ABC	-	REN C, D
HTZ	AUTO C1, D	RE C, D
TEXAS	NUM C1, D	RESEQUENCE C, D
PRIMO	AUTO C1, D	-
TV COM	-	-
VT	AUTO C1, D	RENUM C, D

- Amelyik gépnél valamelyik utasítás hiányzik, címkéről-címkére, kézzel kell helyettesítenünk, beleértve a vezérlés átadásokban szereplő címkeket is.

b) Kurzor pozicionáló parancsok.

- A kurzort az aktuális sor J-edik pozíciójába, a képernyő I, J pontjára és I pozícióval odébb állíthatjuk:

C-64	TAB(J)	-	SPC(T)
ABC	TAB(J)	CUR(I, J)	SPACE ☒ (T)
HTZ	TAB(J)	-	-
TEXAS	TAB(J)	DISPLAY AT I, J	-
PRIMO	TAB(J)	-	SPACE\$ (T)
TV COM	TAB(J)	PRINT AT I, J	STRING\$(T, " ")
VT-16	TAB(J)	LOCATE I, J	SPACE\$ (T)

- A számítógépek alkalmazási lehetősége a háttér tárolók bekapcsolásával kitágul. A mágneses tárolók (kazetta, lemez, merev lemez) program és adat FILE-okat rögzíthetnek, ezeket megőrzik és a gép számára ismételten hozzáférhetővé teszik. A nyomtató passzívan őrzi meg ugyanazokat. Programok rögzítése és betöltése a SAVE és LOAD kulcsszavakra épül, amit géptípuson és eszközként eltérő módon egészítenek ki. A rögzített programot ellenőrizhetjük a gépben és a tárolón fekvő jelek egybevetésével. Kulcsszóként a VERIFY, CHECK, TEST stb. használatosak.

- Az adat FILE-ok rögzítése és betöltése hármastagozódású utasítás rendszerrel oldható meg:

- a FILE megnyitása a tárolón,
- a FILE rögzítése (betöltése) tárolóba (ból),
- a FILE lezárása a tárolón.

Az utasítások rendre az OPEN, a PRINT# (INPUT#) és CLOSE kulcsszavakra támaszkodnak, bár az OPEN számos esetben PREPARE, CREATE szavakkal helyettesítődik. Relatív FILE-ok esetén a rögzítést (betöltést) a REKORD helyének a meghatározása és a rögzítést (betöltést) közvetítő puffer tároló aktivizálása készíti elő.

UTASITÁSKÉSZLET

c) Kazettás egység. Program rögzítés, ellenőrzés, betöltés.

C-64	SAVE "NEV", 1,1	VERIFY "NEV", 1	LOAD "NEV", 1,C
ABC	SAVECAS: NEV	-	LOADCAS : NEV
HTZ	CSAVE# -1, "NEV"	CLOAD? "NEV"	CLOAD# -1, "NEV"
TEXAS	SAVE CS1	CHECK TAPE	OLD CS1
PRIMO	SAVE "NEV"	TEST "NEV"	LOAD "NEV"
TV COM	SAVE#5: "NEV"	VERIFY#5: "NEV"	LOAD#5: "NEV"
VT-16*	SAVE "C : NEV"	-	LOAD "C : NEV"

d) Adat FILE rögzítés kazettára.

C-64	OPEN α , 1, 2, "NEV" : PRINT# α , VALTOZOK : CLOSE α
ABC	PREPARE "NEV" AS FILE α : PRINT# α , VALTOZOK : CLOSE α
HTZ	PRINT# -1, "NEV" : PRINT# -1, VALTOZOK : PRINT# -1, VEGJEL
TEXAS	OPEN# α : "CS1", FIXED, OUTPUT :: PRINT# α , VALTOZOK :: CLOSE# α
PRIMO	CREATE "NEV" : PRINT# VALTOZOK : CLOSE
TV COM	OPEN OUTPUT "NEV" : PRINT#5 : VALTOZOK : CLOSE OUTPUT
VT-16	OPEN "O", # α , "NEV" : PRINT# α , VALTOZOK : CLOSE# α

e) Adat FILE betöltés kazettáról.

C-64	OPEN α , 1, 0, "NEV" : INPUT# α , VALTOZOK : CLOSE α
ABC	OPEN "NEV" AS FILE α : INPUT# α , VALTOZOK : CLOSE α
HTZ	INPUT# -1, NEV\$: INPUT# -1, VALTOZOK : INPUT# -1, VEGJEL
TEXAS	OPEN# α : "CS1" FIXED, INPUT :: INPUT# α , VALTOZOK :: CLOSE# α
PRIMO	OPEN "NEV" : INPUT# VALTOZOK : CLOSE
TV COM	OPEN INPUT "NEV" : INPUT#5 : VALTOZOK : CLOSE INPUT
VT-16	OPEN "I", # α , "NEV" : INPUT# α , VALTOZOK : CLOSE# α

1; CAS; -1; CS1; 5 ≡ kazettás egység azonosítók; α ≡ FILE-szám; 0 ≡ output. * WFLOPPY-ra vonatkozó utasítások.

f) FLOPPY DISK. Program rögzítés, ellenőrzés és betöltés.

C-64	SAVE "NEV", 8	VERIFY "NEV", 8	LOAD "NEV", 8
ABC	-	-	-
HTZ	-	-	-
TEXAS	SAVE "DSK1.NEV" SAVE "DSK1. NEV", MERGE	-	OLD "DSK1. NEV" MERGE "DSK1. NEV"
PRIMO	CMDSAVE "NEV"	CMDTEST "NEV"	CMD LOAD "NEV"
TV COM	SAVE#6: "NEV"	VERIFY#6 : "NEV"	LOAD#6 : "NEV"
VT-16	SAVE "A : NEV" SAVE "B: NEV", A, P	-	LOAD "A : NEV"

g) Adat FILE szekvenciális rögzítése DISK-en.

C-64	OPEN α , 8, k, "Ø : NEV, S, W" : PRINT# α , VALTOZOK : CLOSE α
ABC	-
HTZ	-
TEXAS	OPEN# α :"DSK1.NEV", SEQUENTIAL, OUTPUT, VARIABLE N :: PRINT# α :VALTOZOK :: CLOSE α
PRIMO	CMDCREATEk, "NEV" : CMDPRINT#k, VALTOZOK : CMDCLOSEk
TV COM	OPEN#6:OUTPUT "NEV" : PRINT#6: VALTOZOK : CLOSE#6: OUTPUT OPEN#6:OUTPUT"NEV" : LLIST#6 :: CLOSE#6 : OUTPUT
VT-16	OPEN "0", # α , "NEV" : WRITE# α , VALTOZOK : CLOSE# α

h) Szekvenciális adat FILE betöltése DISK-ről.

C-64	OPEN α , 8, k, "Ø : NEV, S, R" : INPUT# α , VALTOZOK : CLOSE α
TEXAS	OPEN# α : "DSK1.NEV", SEQUENTIAL, INPUT, VARIABLE N :: INPUT# α : VALTOZOK : : CLOSE α
PRIMO	CMDOPENk, "NEV" : CMDINPUT#k, VATOZOK : CMDCLOSEk
TV COM	OPEN#6 : INPUT "NEV" : INPUT#6 : VALTOZOK : CLOSE#6 : INPUT OPEN#6 : INPUT "NEV" : POKE 2818,6 : CLOSE#6 : INPUT
VT-16	OPEN "I", # α , "NEV" : INPUT# α , VALTOZOK : CLOSE α

6; 8; DSK1; A; B ≡ FLOPPY azonosítók; α ≡ FILE-szám; k ≡ CSATORNA-SZÁM; S ≡ SEQUENTIALIS; W ≡ WRITE;
N ≡ REKORD-hossz; 0 ≡ OUTPUT; R ≡ READ; I ≡ INPUT.

UTASITÁSKÉSZLET

i) Relatív FILE rögzítése DISK-en.

C-64 OPEN 15, 8, 15 : OPEN α , 8, k, "NEV, L," +CHR\$(N) : H=INT (I/256) : L=I-H*256
: PRINT#15, "P" + CHR\$(k) + CHR\$(L) + CHR\$(H) + CHR\$(1) : PRINT# α ,
VALTOZOK : CLOSE α : CLOSE15
ABC -
HTZ -
TEXAS OPEN# α : "DSK1. NEV", RELATIVE N1, FIXED N : : PRINT# α , REC I :
VALTOZOK :: CLOSE α
PRIMO CMDDEFk, "NEV, L," + CHR\$(N) : H=INT (I/256) : L=I-H*256 : CMDPRINT "P"
+ CHR\$(k) + CHR\$(L) + CHR\$(H) + CHR\$(1) : CMDPRINT#k, VALTOZOK :
CMDCLOSEk
TV COM -
VT-16 OPEN "R", # α , "NEV", N : FIELD# α , N AS V\$: LSET V\$=MKI\$(V) : PUT# α , I :
CLOSE# α

j) Relatív FILE betöltése DISK-ről.

C-64 OPEN15, 8, 15 : OPEN α , 8, k, "NEV" : H=INT(I/256) : L=I-H*256 : PRINT#15,
"P" + CHR\$(k) + CHR\$(L) + CHR\$(H) + CHR\$(1) : INPUT# α , A\$: PRINTA\$:
CLOSE α : CLOSE15
TEXAS OPEN# α : "DSK1. NEV", RELATIVE N1, FIXED N : : INPUT# α , REC I : A\$: :
PRINT A\$: CLOSE α
PRIMO CMDDEFk, "NEV" : H=INT (I/256) : L=I-H*256 : CMDPRINT "P" + CHR\$(k) +
CHR\$(L) +CHR\$(H) + CHR\$(1) : CMDINPUT#k, A\$: PRINT A\$: CMDCLOSE k
VT-16 OPEN "R", # α , "NEV", N : FIELD# α , N AS V\$: GET# α , I : PRINT A\$: A%=CVI
(A\$) : PRINT A% : CLOSE# α

k) Program nyomtatása printer-en.

C-64 OEPN α ,4 : CMD α : LIST : PRINT# α : CLOSE α
ABC -
HTZ -
TEXAS -
PRIMO LLIST
TV COM LLIST
VT-16 LLIST

α \equiv FILE-szám; k \equiv csatorna-szám; R, L \equiv a relatív jelleg mutatója; N \equiv REKORD-hossz; N1 \equiv FILE-hossz;
DSK1 \equiv FLOPPY azonosító.

l) Adat FILE nyomtatása printer-en.

C-64 OPEN α , 4 : PRINT# α , VALTOZOK : CLOSE α
 PRIMO LPRINT VALTOZOK
 TV COM PRINT#4 : VALTOZOK
 VT-16 LPRINT VALTOZOK
 OPEN "0", #4, "LPT1:" : PRINT# 4, VALTOZOK : CLOSE#4

α \equiv FILE-szám; 4 \equiv PRINTER azonosító szám; 0 \equiv OUTPUT; LPT1 \equiv a PRINTER FILE neve.

10. Géptípusra jellemző utasítások

a) C-64.

BREAK	megszakítja a program futását.
CONT	feloldja a STOP utasítást.
DEF FNY1(X)	Y1 (X) függvény bevezetése.
GET A\$	egy karakter hívása klaviatúráról.
GET# α , A\$	FILE betöltése karakterenként.
LOAD "\$", 8	a DISK nyilvántartás betöltése.
OPEN15,8,15,"NØ:NEV,Ø1"	DISK-et formalizálja.
,"IØ:"	FLOPPY alapállása.
,"SØ:NEV"	a "NEV" FILE-t törli a DISK-ről.
,"SØ:*"	törli a DISK-et.
,"VØ:"	látszólag foglalt BLOCK-ok szabaddá tétele.

b) ABC

CLEAR	törli a változókat és lezárja a FILE-okat.
CLR DOT (X, Y)	törli az X, Y pontot az ernyőn.
DOT (X, Y)	logikai függvény. -1 ha az X, Y pont világít és Ø, ha nem.
ED(C)	C címkes: utasítás kiírása javítás céljából.
GET A Ø	egy karakter hívása klaviatúráról.
INPUTLINE	STRING INPUT.
LOGIØ(X)	X tizesalapú logaritmus.
MERGE "NEV"	program betöltés és egyesítés a gépben lévő programmal.
RUN"NEV"	program betöltés kazettáról futtatással egybekötve.
SET DOT(X, Y)	kigyújtka képernyőn az X, Y pontot.

c) HTZ

CLEAR N	N rekeszt lefoglal STRING-EK részére.
CLS	törli a képernyőt.
CON	feloldja a STOP utasítást.
DEF DBL A-Z	dupla pontos változókat definiál.
DEF INT A-Z	egészértékű változókat definiál.
DEF SNG A-Z	egyszeresen pontos változókat definiál.
DEF STR A-Z	STRING változókat definiál.
EDIT C	C című utasítás képernyőre íratása javítás céljából.
ERL	hibás utasítás címkéjét azonosítja.
ERR	a hiba jellegére utaló kódot ad.
INKEY\$	klaviatúráról fogad 1 karaktert.
ON ERROR GOTO C	hibacsapdát definiál C alatt.
POINT (X, Y)	logikai függvény a grafikában: -1, ha ég és Ø, ha nem ég az X, Y koordinátákhoz tartozó pont.

RESUME C	hibacsapdából kilépés C-re.
RESET (X, Y)	kioltja az X, Y koordinátákhoz tartozó pontot.
SYSTEM	betölt, illetve elindít gépi kódban írt programot.
SET (X, Y)	kigyűjti az X, Y koordinátákkal adott pontot a képernyőn.
TROFF	kikapcsolja az utasítások összefüggés vizsgálatát.
TRON	bekapcsolja az utasítások összefüggés vizsgálatát.
VARPTR (X)	X változót tároló rekesz címét szolgáltatja.

d) TEXAS

BREAK C	C címkénél megszakítja a program futását.
CALL CLEAR	törli a képernyőt.
CALL COLOR (JIK, JCK, HCK)	karakter és háttér színezése JIK=1-14, JCK=1-16, HCK=1-16.
CALL CHAR (JK, J\$)	a definiált J\$ karakter JK kóddal való megidézése.
CALL ERR (W, X, Y, Z)	W hibakód, X hibatípus, Y erősség, Z utasítás címke azonosítása.
CALL GCHAR (I, J, X)	a képernyő I, J helyén talált karakter kódját adja meg.
CALL HCHAR (I, J, JK, R) ..	R-szer jeleníti meg a JK kódú karaktert az I-edik sorban, a J-edik pozíción kezdve. I=1-24. J=1-32.
CALL SCREEN (CP)	a képernyő keretét színezi. CP=1-16.
CALL VCHAR (I, J, JK, R) ..	R-szer jeleníti meg a JK kódú karaktert a J-edik oszlopban, az I-edik pozíción kezdve.
CONTINUE	feloldja a BREAK utasítást.
DEF A(X)	A(X) függvény definiálása.
DELETE "DSK1. NEV"	FILE törlése DISK-ről.
DISPLAY AT (I, J) : USING " ,,:	a képernyő I, J pozíciójába formalizált adatot ír.
EDIT C	C című utasítás képernyőre íratása javítás céljából.
EOF (α)	α file végét jelzi.
MAX (A, B)	kiválasztja a nagyobb számot.
MERGE "DSK1. NEV"	betölti lemezről a "NEV" programot és egyesíti a gépben álló programmal.
ON ERROR C	hibacsapdát definiál C alatt.
OPTION BASE 1	a tömbindex 1-el kezdődik.
RUN "DSK1. NEV"	program betöltés DISK-ről futással egybekötve.
SAVE "DSK1, NEV", MERGE SAVE "DSK1. NEV", PROTECTED	MERGE utasítás által betölthető programot rögzít DISK-en. képernyőre nem írható program.
TRACE	bekapcsolja az utasítások összefüggés vizsgálatát.
UNTRACE	kikapcsolja a TRACE utasítást.

UTASÍTÁSKÉSZLET

e) PRIMO

CLEAR N	N rekeszt lefoglal stringek számára.
CLS	törli a képernyőt.
CMD NEW "NEV, Ø1"	DISK-et formalizálja.
CMD ERROR EN, EM\$, ET, ES	hiba azonosítás DISK műveletek esetén.
CMD KILL "NEV"	"NEV" FILE-t törli a DISK-ről.
CMD LOAD "\$"	DISK nyilvántartás betöltése.
CMD SET E	FLOPPY azonosítási számát módosítja E-re (E=9–15).
DEF DBL A–Z	dupla pontos változókat definiál.
DEF INT A–Z	eégszértékű változókat definiál.
DEF SNG A–Z	egyszeresen pontos változókat definiál.
DEF STR A–Z	STRING változókat definiál.
EDIT C	C című utasítás képernyőre íratása javítás céljából.
EOF (k)	k csatornán betöltött FILE vége.
INKEY\$	klaviatúráról fogad 1 karaktert.
ON ERROR GOTO C	hibacsapdát definiál C alatt.
RESET (X, Y)	kioltja az X, Y koordinátákhoz tartozó pontot. X=1–255, Y=191–1.
RESUME C	hibacsapdából kilépés C-re.
SET (X, Y)	kigyűjtja az X, Y koordinátákkal adott pontot a képernyőn.
TROFF	kikapcsolja az utasítások összefüggés vizsgálatát.
TRON	bekapcsolja az utasítások összefüggés vizsgálatát.
VARPTR (X)	X változót tároló rekesz címét adja.
VARPTR (X\$)	X\$ változót tároló rekesz címét adja.

f) TV COMPUTER.

CLS	törli a képernyőt.
CONTINUE	feloldja a STOP utasítást.
DEF A(X)	A(X) függvény definiálása.
GET A\$	egy karakter hívása klaviatúráról.
GRAPHICS 2	szin:2, 24*64-es képernyő, 24Ø*512 auto karakter.
GRAPHICS 4	szin:4, 24*32-es képernyő, 24Ø*256 auto karakter.
GRAPHICS 16	szin:16, 24*16-os képernyő, 24Ø*128 auto karakter.
INKEY \$	klaviatúráról fogad 1 karaktert.
PLOT X1, Y1	X1, Y1 koordinátával r. pont kigyűjtása.
PLOT X1, Y1; X2, Y2	pontok közötti szakasz kigyűjtása.
SET CHARACTER K, E1, E2, ..., E1Ø	tervezett karakterek megjelenítése.
SET INK S(CK)	karakter színezése. CK=Ø,...,15 színkód. S(CK)=1, 2; 1, 2, 3, 4; 1, 2,..., 16 a kiválasztott színek sorszáma.
SET MODE V	keresztvező vonalak eredő színe. V=1, 2, 3.
SET PAPER S(CK)	a képernyő színezése.

SET PALETTE CK, CK	színek kiválasztása GRAPHICS 2-höz.
SET PALETTE CK, CK, CK, CK	színes kiválasztása GRAPHICS4-hez.
SET STYLE VK	vonaltípus kiválasztása PLOT-hoz. VK=1–14.
SET BORDER CK	képernyő keret színezése CK=0, ..., 15.
TRACE ON	bekapcsolja az utasítások összefüggés vizsgálatát.
TRACE OFF	kikapcsolja az utasítások összefüggés vizsgálatát.

g) VT–16.

CDBL(X)	X-et dupla pontosságú számmá konvertálja.
CHAIN "A : NEV", C, ALL, DELETE C1–C2	"NEV" program betöltése futó program végén, a C címkével folytatódik a "NEV" megoldása, átvéve az első program összes változója értékeit. A "NEV" program egy részét törölhetjük is.
CINT(X)	X-et egészértékűvé konvertálja.
CLEAR	törli a változók értékét a memóriában.
CLS	képernyőt törli.
COLOR CK, CK	színezés CK=0, 7, 8, 15.
COMMON VALTOZOK	a megadott változók értékeivel számolhat a CHAIN "NEV"-je.
CONT	feloldja a STOP utasítást.
CSNG(X)	X-et egyszeres pontosságú számmá konvertálja.
CSRLIN	a KURZOR sor pozícióját azonosítja.
CVI, CVS, CVD	2, 4, 8 BYTE-os STRING számok numerikus formára alakítása.
DATA\$	azonosítja a dátumot.
DEF FN A(X)	A(X) függvény definiálása.
DEF DBL A–Z	dupla pontos változókat definiál.
DEF INT A–Z	egészértékű változókat definiál.
DEF SNG A–Z	egyszeresen pontos változókat definiál.
DEF STR A–Z	STRING változókat definiál.
EDIT C	C című utasítás képernyőre írása javítás céljából.
EOF(α)	–1, ha vége van az α számú FILE betöltésének.
ERASE A(N)	A(N) tömb törlése program futása közben, ha már felesleges.
ERR, ERL	hibakód és címke azonosítók.
FILES	DISK nyilvántartás betöltése.
HEX\$(N)	számot 16-os rendszerben fejez ki.
INKEY\$	klaviatúráról fogad 1 karaktert.
KEY ON, OFF, LIST	F1, ..., F10 billentyűk értékét megjeleníti, törli, listázza.
KEY N, X\$	FN billentyű értékét A\$-re változtatja.
KILL "A : NEV. BAS"	FILE törlése DISK-ről az A FLOPPY-val.
LINE(X1, Y1) – (X2, Y2)	szakasz rajzolás képernyőn.
LOAD "A : NEV", R	betölti DISK-ről a megjelölt programot és futtatni kezdi.
LOF (α)	az α számú FILE-ben foglalt BYTE-ok számát méri.

UTASÍTÁSKÉSZLET

MERGE "B : NEV"	betölti lemezlől a "NEV" programot és egyesíti a gépben lévő programmal.
MKD\$(A#), MKI\$(A%), MKSS\$(A!)	számok STRING formáját adják.
NAME "A : NEV. BAS" AS "NEV1. BAS"	NEV FILE új nevet kap : NEV1.
ON ERROR GOTO C	hibacsapdát definiál C alatt.
ON KEY (N) GOSUB C	FN gomb leütésekor a vezérlés C-re adódik.
OPTION BASE 1	a tömbindex 1-el kezdődik.
POS(1)	a KURZOR oszlop pozícióját adja.
PSET(X, Y), 1	kigyűjtja az X, Y pontot a képernyőn X=0-639, Y=0-299.
PRESET(X, Y), 0	X, Y pont kioltása a képernyőn.
RESUME C	hibacsapdából kilépés C-re.
SAVE "B : NEV", P	titkosított program rögzítés DISK-en.
SCREEN 2	átállás grafikus módra. 300*640 pont a képernyőn.
SWAP A, B	a két változó értékét felcseréli.
SYSTEM	kilépés a BASIC-ból. Program törlődik. File-ok lezáródnak.
TIME \$	a pontos időt azonosítja.
TIMER	másodpercben adja meg a gép bekapcsolása óta eltelt időt.
TROFF	kikapcsolja az utasítások összefüggés vizsgálatát.
TRON	bekapcsolja az utasítások összefüggés vizsgálatát.
VARPTR(X)	X változót tároló rekesz címét szolgálja.
VT BASIC A : PROGRNEV < B : ADAT NEV > LPT1 /F : 15/S : 512	a BASIC beolvasását egy program követi az A FLOPPY-ból. A program elindul. Adat igényét INPUT útján a ' < B FLOPPY-ban levő lemez ADATNEV FILE-ből veszi. A program OUTPUT-ját a > PRINTER-re viszi ki. A futás előtt 15-re terjeszti ki a megnyitható FILE-ok számát és 512-re emeli a direkt puffer méretét.
WHILE-WEND	feltételes ciklus szervező pár.
WIDTH "LPT1 : "N	sorhosszt állítja be printer-en.

COMMODORE-64 hibajegyzék

A hiba jelzése:

BAD DATA
 BAD SUBSCRIPT
 CAN'T CONTINUE
 DEVICE NOT PRESENT
 DIVISION BY ZERO
 EXTRA IGNORED
 FILE NOT FOUND
 FILE NOT OPEN
 FORMULA TOO COMPLEX
 ILLEGAL DIRECT
 ILLEGAL QUANTITY
 NEXT WITHOUT FOR
 NOT INPUT FILE
 OUT OF DATA
 OUT OF MEMORY
 OVERFLOW
 REDIM'D ARRAY
 REDO FROM START
 RETURN WITHOUT GOSUB
 STRING TOO LONG
 SYNTAX ERROR
 TYPE MISMATCH
 UNDEF'D STATEMENT

A hiba oka:

változó és adat összeférhetetlen.
 hibás index.
 STOP után módosított a programon (>RUN).
 a hivatkozott peréfiria nem él.
 az osztó nulla.
 túlteljesített INPUT.
 adott DISK-en nemlétező FILE-ra hivatko-
 zunk.
 FILE művelet megnyitás nélkül.
 STRING függvények egymásba skatulyázása
 sokszoros.
 parancsként nem működő utasítás.
 ellentmondó értékadás, vagy nem értelmezhető
 művelet.
 FOR hiányzik a FOR-NEXT párból.
 FILE utasítások összeférhetetlenek.
 fedezetlen adatigény READ esetén.
 a memória potenciálisan foglalt.
 túlméretezett adat.
 tömb ismételt dimenzionálása törlés nélkül.
 típusidegen adat.
 GOSUB hiányzik a GOSUB-RETURN pár-
 ból.
 túlméretezett STRING.
 helyesírási hiba.
 függvényidegen argumentum vagy típusidegen
 azonosítás.
 vezérlésátadás nemlétező címre vagy para-
 méterrel.

További hibaüzenetek vonatkoznak a FLOPPY egységre, amik a hiba kódját (0-80),
 illetve nevét tartalmazzák. Ezek az irodalomjegyzék [3]-ban található.

ABC–80 hibajegyzék

A hiba kódja:	A hiba oka:
Ø	nem megengedett indext használ.
1	nem következetes tömbazonosítás.
2	címke nélküli utasítást adott.
3	memória betelt, adat túlcsondult.
4	nem értelmezhető művelet.
5	nem megengedett változót, indexet stb. használ.
6	GOTO címkéje nemlétező utasításra vonatkozik.
7	nemlétező memória címet használ.
9	karakter konstans nemlétező részét próbálja kiemelni.
10	túlhosszú karakter konstans.
11	hiányos utasítás, nem értelmezett operáció (géptidegen).
12	numerikus változóhoz STRING-et rendel.
13	operandusok és műveleti jelek párosítása hibás.
15	hiányzik az = jel.
16	ugrás sehová, címkehiány a GOTO vagy GOSUB utasításokban.
17	típuskeveredés változók és adatok azonosításában, összehasonlításában.
18	zárójel hiány vagy többlet.
19	FILE lezárása elmaradt.
20	120 karaktert túllépő utasítás.
21	hiányos utasítás, pontatlan hivatkozás. A keresett FILE nincs a kazettán.
23	TO hiány a FOR–NEXT utasításban.
24	NEXT hiány a FOR ciklusban.
25	ON GOSUB utasítás szerkezete hibás.
26	ON V GOSUB C1, ... utasítás V-je túlmutat a C soron.
27	FOR hiány a FOR–NEXT párban.
28	A FOR és NEXT indexe nem azonos.
29	GOSUB érintése nélkül RETURN-ba futó program.
30	a DIM vagy DATA adtalehetőséget meghaladó beolvasási utasítás.
32	A FILE nyitása elmaradt.
33	a FILE megnyitó utasítás hiányos.
34	a FILE állományán túlmutató betöltési utasítás.
35	hibás FILE.
50	negatív számból von négyzetgyököt.
60	a program címe 8 karakternél hosszabb (rögzítésnél).
63	AS hiányzik a FILE megnyitásából.

A hiányzó hibakódok vagy a teljes kiépítésű rendszerre (FLOPPY, PRINTER) vonatkoznak, vagy ritkán lépnek fel.

HTZ hibajegyzék

A hiba jelzése:	A hiba oka:
BD	adatkiolvasási hiba a kazettánál.
BS	nagyobb index a FOR-ban, mint a DIM-ben.
CN	END után adott CON parancs.
DD	kétszer deklarált DIM, eltérő index.
FC	nem értelmezhető művelet.
FD	adatrögzítési hiba a kazettánál.
ID	indirekt utasítást adtunk direktre.
LS	túlméretezett STRING (h > 255).
MO	operandus hiány (pl. 5*).
NF	NEXT hiányzik a FOR–NEXT párból.
OD	adathiány a READ, illetve INPUT#-nál.
OM	a memória potenciálisan betelt.
OS	betelt a STRING-ekhez rendelt standard memória.
OV	túlméretezett adat.
RG	GOSUB hiányzik a GOSUB–RETURN párból.
SN	helyesírási hiba.
ST	túlzottan összetett STRING függvény.
TM	összeférhetetlen adattípusok.
UL	vezérlésátadás nemlétező címkére.
/0	a nulla osztóként lép fel.

TEXAS hibajegyzék

A hiba jelzése:	A hiba oka
BAD ARGUMENT	adott argumentumra a függvény nincs értelmezve.
BAD LINE NUMBER	hibás címke (<1; >32767).
BAD SUBSCRIPT	hibás index.
BAD VALUE	értelmetlen mutatók vagy kódok használata.
CAN'T DO THAT	végrehajthatatlan utasítás.
COMMAND ILLEGAL IN PROGRAM	adott utasítás a programban nem szerepelhet.
DATA ERROR	hibás vagy kevés vagy típusidegen adat.
FILE ERROR	hibás, a FILE állapotának ellentmondó Művelet.
FOR-NEXT NESTING	a FOR-NEXT mutatók hibásak.
INCORRECT STATEMENT	karakterhibás utasítás.
I/O ERROR	adat vagy program-FILE bevitele, rögzítése, betöltése hibás.
LINE NOT FOUND	hibás vezérlés-átadás.
MEMORY FULL	a lefoglalt vagy a teljes BASIC területünk betelt.
NAME TOO LONG	túl hosszú név (>15 karakter).
NAME CONFLICT	többértelmű azonosítás.
NEXT WITHOUT FOR	FOR lemaradt a FOR-NEXT párból.
NO PROGRAM PRESENT	nincs program a gép memóriájában.
NUMERIC OVERFLOW	adat túlsordulás a memóriában.
RETURN WITHOUT GOSUB	GOSUB lemaradt a GOSUB-RETURN párból.
STRING NUMBER MISMATCH	összeférhetetlen típusjelzések.
SYNTAX ERROR	szintaktikai (helyesírási) hiba.
UNRECOGNIZED CHARACTER	hibás karakter használat.

További hibáüzenetek találhatóak az irodalomjegyzék [10]-ben.

PRIMO hibajegyzék

A hiba jele:	A hiba oka:
NF	FOR lemaradt a FOR—NEXT párból.
SN	színtaktikai (helyesírási) hiba.
RG	GOSUB lemaradt a GOSUB—RETURN párból.
OD	kevés az adat az igényhez képest.
FC	adott argumentumra a függvény nincsen értelmezve.
OV	adat túlcsoordulás a memóriában.
OM	BASIC területünk betelt.
UL	hivatkozás nemlétező címkére.
BS	túllépte a tömb deklarált dimenzióját.
DD	egy tömböt kétszer dimenzionált.
/Ø	nullával próbál osztani.
ID	az utasítás csak programban állhat.
TM	összeférhetetlen típusjelzések.
OS	STRING tárolónk betelt.
LS	túlhosszú STRING.
ST	túlbonyolult STRING függvény.
UE	ismeretlen hiba.
FE	FILE kezelési hiba.
FD	kevés az adat a FILE-ban /a FILE rögzítése nem ellenőrizhető.
BO	újabb csatorna nem nyitható meg (>1Ø).
FO	nyitott csatornát akar megnyitni.
NO	zárt csatornát akar használni.
BD	írásra/olvasásra használja az olvasásra/írásra megnyitott csatornát.
DEVICE NOT PRESENT	CDOS, eszköz, lemez, dugaszolás stb. hiányzik a rendszerből.
FILE NOT FOUND	adott című FILE nincs a DISK-en.

TV-COMPUTER hibajegyzék

A hiba jelzése:

BAD ARGUMENT
 BAD FILE
 BAD SUBSCRIPT
 CANNOT CONTINUE
 CANNOT READ
 LINE MISSING
 NO DATA
 NO FOR
 NO GOSUB
 NO MEMORY
 NOT UNDERSTOOD
 OVERFLOW
 SYSTEM ERROR α
 TYPE MISMATCH
 VARIABLE DECLARED TWICE

A hiba oka:

adott értékre nem értelmezhető függvény vagy művelet.
 rögzítési hiba kazettán vagy DISK-en.
 ellentmondó értékadás pl. DIM A(2) : PRINT A(3).
 STOP után módosított a programon (>RUN). az olvasott változótól eltérő típusú adat. vezérlésátadás nemlétező címre vagy paraméterrel.
 fedezetlen adat igény.
 hiányzik a NEXT FOR párja.
 hiányzik a RETURN GOSUB párja.
 a belső memória megtelt. Pl. sok, vagy magas dimenziójú tömb.
 helyesírási hiba a kijelzett, vagy a kapcsolódó utasításban.
 túlméretezett adat.
 a perifériák rendellenessége (α kódok a [13] gépkönyvben).
 függvényidegen argumentum vagy típusidegen azonosítás.
 hibás változó használat pl. DIM A(2 \emptyset), A1(2 \emptyset), A\$(20), vagy törlés nélkül ismételt dimenziálás.

VT-16 hibajegyzék

A hiba jelzése:

BAD FILE MODE
 BAD FILE NAME
 BAD FILE NUMBER
 CAN'T CONTINUE
 DEVICE I/O ERROR
 DEVICE UNAVAILABLE
 DISK FULL
 DISK NOT READY
 DISK WRITE PROTECT
 DIVISION BY ZERO
 DUPLICATE DEFINITION
 FILE ALREADY OPEN
 FILE NOT FOUND
 FIELD OVERFLOW
 FOR WITHOUT NEXT
 ILLEGAL FUNCTION CALL
 ILLEGAL DIRECT
 INPUT PAST END
 INCORRECT DOS VERSION
 LINE BUFFER OVERFLOW
 MISSING OPERAND
 OVERFLOW
 OUT OF DATA
 OUT OF MEMORY
 OUT OF PAPER
 RETURN WITHOUT GOSUB
 SYNTAX ERROR
 STRING TOO LONG
 STRING FORMULA TOO COMPLEX
 SUBSCRIPT OUT OF RANGE
 TYPE MISMATCH

A hiba oka:

FILE utasítások összeférhetetlenek.
 FILE művelet hibás névadással.
 a megengedettnél több FILE-t nyit meg, vagy számát elírja.
 STOP után módosított a programon (>RUN), sikertelen periféria művelet.
 a hivatkozott periféria nem él.
 a DISK betelt.
 a DISK-et a FLOPPY nem érzékeli.
 védett lemezre akartunk rögzíteni.
 az osztó nulla.
 törlés nélkül ismételt dimenzionálás.
 nyitott FILE megnyitása.
 adott DISK-en nemlétező FILE-ra hivatkozunk.
 a direkt FILE utasításai összeférhetetlenek (rekord-hossz).
 A NEXT hiányzik a FOR-NEXT párból.
 ellentmondó értékadás vagy nem értelmezhető művelet.
 parancsként nem működő utasítás.
 fedezetlen adatigény egy FILE-val szemben.
 idegen parancs vagy utasítás.
 túlméretezett utasítás vagy sorozat egy cím-ben.
 adathiányos művelet.
 túlméretezett adat.
 fedezetlen adatigény READ esetén.
 a memória potenciálisan foglalt (betelt).
 hibásan fekszik vagy nincs papír a PRINTER-ben.
 GOSUB hiányzik a GOSUB-RETURN párból.
 helyesírási hiba a kijelzett vagy kapcsolódó utasításokban.
 túlméretezett STRING.
 STRING függvények egymásba skatulyázása sokszoros.
 ellentmondó értékadás tömbök esetén.
 függvényidegen ARGUMENTUM vagy típus-idegen azonosítás.

VT-16 HIBAJEGYZÉK

A hiba jelzése:

TOO MANY FILES
 UNDEFINED LINE NUMBER
 WHILE WITHOUT WEND

A hiba oka:

a FLOPPY nem tud újabb FILE-t feljegyezni adott DISK-re, mert telített, illetve a FILE szabálytalan.
 vezérlésadás nemlétező címre vagy paraméterrel.
 a WEND hiányzik a WHILE-WEND párból.

GRAFIKAI ÚTMUTATÓ

A személyi számítógépek alapállapotban karakter méretű jeleket vihetnek képernyőre. A képernyő felbontása, (sor*oszlop) az alábbi:

C-64	ABC	HTZ	TEXAS	PRIMO	TV COM	VT-16
25*40	24*40	16*64	24*32	16*42	24*32	25*80
					24*16	
					24*64	

A karakterek mikro karakterekre bonthatók. A felbontás vagy külön utasítással, vagy az alkalmazási utasításokba beépítve oldható meg. A képernyő és karakterek felbontása (mikro sor * mikro oszlop az alábbi:

C-64	ABC	HTZ	TEXAS	PRIMO	TC COM	VT-16
200*320	72*80	48*128	192*256	192*256	960*1024	300*640
8*8	3*2	3*2	8*8	—	40*32	12*8
					40*64	
					40*16	

A mikro karakteres felbontást ábrák, grafikonok és képek megjelenítése esetén alkalmazzuk. A rajztechnikai utasítások kódokkal (képernyő, szín, karakter stb. kódok) dolgoznak. A kódok 2-es (BINÁRIS), 10-es (DECIMÁLIS) és 16-os (HEXADECIMÁLIS) rendszerben kerülnek kifejezésre. Ez utóbbit a memória rekeszcímek megadására is gyakran alkalmazzák.

Egész számokat 16-os rendszerbe egy osztás sorozat maradékai visznek át.

$$\text{Pl. } 38 : 16 = 2 : 16 = 0$$

$$\text{maradék: } 6 \quad 2$$

Igy $38 = 26$, mert $2*16 + 6 = 38$. A HEXADECIMÁLIS rendszer 16 számjeggyel dolgozik: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, ahol A=10, B=11, C=12, D=13, E=14, F=15. Például az 5F3B szám az $5*16^3 + 15*16^2 + 3*16^1 + 11$ -nek felel meg tizes rendszerben.

A rajztechnikai eljárásokat a grafika szó foglalja egybe. Így kétféle grafikáról beszélhetünk, úgymint karakter grafika és mikro karakter grafika. Mindkettőhöz járulhat fekete-fehér, vagy színes technika. Ez utóbbi a számítógép és képernyő közös adottsága.

A karakter grafikát a gép alapállása esetén alkalmazhatjuk. A rajz készítés a kurzor pozícionálásából, a megfelelő grafikai karakter megjelenítéséből, a képernyő keret és lap, valamint a karakter jel és háttér színezéséből áll.

A mikro karakter grafika vagy közvetlenül, vagy a karakterek közvetítésével érvényesül. A bonyolultabb mikro grafikát egyes gépek esetében összefoglaljuk, illetve hivatkozunk korábbi ismertetésükre.

ABC-80 grafika

A képernyő normál (24x40) és grafikus (72x80) üzemmódban dolgozhat. A grafikus sorok és oszlopok indexelése nullával kezdődik. A grafikus üzemmód nem az egész képernyőre, hanem a deklarált karakterpozíciótól jobbra fekvő karakter sorszáma áll fenn. Itt férhetünk hozzá a mikro karakterekhez. Normál üzemmódból a >PRINT CUR (I, J) CHR $\text{\textcircled{X}}$ (151) utasítással érjük el az I+1-edik sor J+1., J+2.,...,40. Karaktereiben a 3x2-es mikro felbontást. Ugyanezt érjük al a >PRINT TAB (J) CHR $\text{\textcircled{X}}$ (151) utasítással a gép számára aktuális sorban. Az élő mikro karakterek indexeit a kurzor pozíció indexeiből az $I1=3*I+\emptyset;1;2$ és $J1=2*J+\emptyset;1$ képletekkel határozhatjuk meg, ahol $J=\emptyset;...;39$ bármelyikével számolhatunk.

A grafikus üzemmódra átállított karakter pozícióban kétféle eljárással rajzolhatunk : 1., >PRINT CHR $\text{\textcircled{X}}$ (K) az ASC KÓD (K) grafikus képét jeleníti meg a kijelölt pozícióban vagy pozíciókban. 2., a SETDOT(I1, J1) utasítás kivilágítja az I1, J1 mikro karaktert.

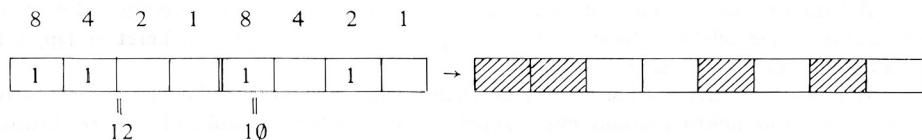
Grafikus üzemmódra állított sorrész, bármelyik pozíciótól kezdve, feltéve, hogy ábrától mentes, normál üzemmódra állítható vissza. Ezt a >PRINT CUR (I, L) CHR $\text{\textcircled{X}}$ (135) utasítással érjük el ($L < 40$). Az I, L;I, L + 1 stb. pozíciókba karakterek írhatók.

HTZ grafika

A képernyő mikro karakteres felbontását az alkalmazási utasítások közvetlenül oldják meg. Így a SET (X, Y) 48*128 mikro karakterre bontja a képernyőt és egyben kigyűjtja az X, Y koordinátával ($0 \leq X \leq 127, 0 \leq Y \leq 47$) rendelkező mikro karaktert. Hasonlóan funkcionál a RESET (X, Y) és a POINT (X, Y) a kioltás és ellenőrzés esetén. Ha a pontokat megfelelően pozicionáljuk, geometriai vagy más pl. képi alakzatok hozhatók létre.

TEXAS grafika

A mikro grafika a karakterek közvetítésével érvényesül. Egy képernyő karakter 8*8 mikro és 8*2 mini karakterre bontható. Egy mini karakter eszerint 4 mikro karakterből áll. Egy mini karakter 1-4 mikro karakterét az 1,2,...,15 DECIMÁLIS számokkal aktiválhatjuk. Ahol a szám BINÁRIS alakjában 1-es áll ott képelem keletkezik a mini karakterben. Pl.:



A DECIMÁLIS számokat HEXADECIMÁLIS rendszerbe átvive a mini karakterek képe egy-egy számjeggyel azonosítható (példánkban "CA"). Eszerint egy karakterben foglalt mikro karakterek képe egy 16 karakterből álló STRING-gel adható meg. A karakterek felülről-lefelé és balról-jobbra olvasva kerülnek be a STRING-be. Az így megtervezett karakterek 128-159 közötti kódot kapnak. (Pl. 130, J\$).

A mikro karakterből összeállított karakter megjelenítése a >CALL CHAR (JK, J\$) és a >CALL HCHAR (I, J, JK, R), illetve a >CALL VCHAR (I, J, JK, R) utasítás-párral lehetséges. Ha R, vagyis a karakter ismétlődési száma 1, akkor a JK kódú J\$ karakter az I-edik karakter sor (I=1–24) J-edik (J=1–32) pozíciójában jelenik meg. Ha viszont R>1, akkor a kezdő pozíciótól jobbra, illetve lefelé R-szer ismétlődik. Például J\$= 000000FFFFFF000000”, JK=130 esetén a >CALL CHAR (130, J\$) :: CALL HCHAR (10, 1, 130, 32) vízszintes vonalat rajzol a képernyő 10-edik sorába. Ha viszont R=768 és I=J=1, akkor a teljes képernyőt lefedi a JK karakter akár a HCHAR, akár a VCHAR utasítással dolgozunk. Ha a gép karaktereit használjuk a rajzolás során, akkor a CALL CHAR utasítás elmarad és JK-ként a gépi karakter ASC kódját használjuk.

Színes technika

>CALL SCREEN (CP) a képernyő keret színét adja meg a CP = 1–16 színkódok szerint. >CALL COLOR (JIK, JCK, HCK) a jel (JCK) és háttere (HCK) színkódok, valamint a jel-intervallum (JIK) kód alapján a >CALL HCHAR (I, J, JK, R)-ban megadott JK karaktert és hátterét színezi. A JIK az alábbi táblázatból olvashatók ki a JK ismeretében:

JK	32–39	40–47	48–55	56–63	64–71	72–79	80–87
JIK	1	2	3	4	5	6	7

JK	88–95	96–103	104–111	112–119	120–127	128–135	136–143
JIK	8	9	10	11	12	13	14

A színező utasítások sorrendje:

>CALL SCREEN (CP) :: CALL COLOR (JIK, JCK, HCK) :: >CALL CHAR (JK, J\$) :: >CALL HCHAR (I, J, JK, R) Az utasításokban szereplő kódok változókként is megadhatók, beleértve a JIK is, amit a JK intervallumokkal adott feltételből határozhat meg a program.

PRIMO grafika

A képernyő mikro karakteres felbontását az alkalmazási utasítások közvetlenül oldják meg. Így a SET (X, Y) 192*256 mikro karakterre bontja a képernyőt és egyben kigyűjtja az X, Y koordinátákkal (0≤X≤255, 191≥Y≥0) rendelkező mikro karaktert. Hasonlóan funkcionál az X, Y pontot kioltó RESET (X, Y) és a POINT (X, Y) logikai függvény, ami 1-el az aktivitást jelzi.

Alakzatok előállítására az X, Y pontok megfelelő pozicionálásával jár együtt.

TV COM grafika

A mikro grafika önállóan és a karakterek közvetítésével is megvalósítható. A >PLOT X, Y utasítás ($0 \leq X \leq 1023$ és $959 \geq Y \geq 0$) kigyűjti a képernyő X, Y koordinátájú pontját attól függetlenül, hogy GRAPHICS 2, 4, 16 állapotban működik a gép. A PLOT utasítás alkalmas szakasz: >PLOT X1, Y1; X2, Y2, pontsor: >PLOT X1, Y1, X2, Y2, ... és zárt poligon (háromszög, négyszög stb.) ábrázolására is. Előírhatjuk azt is, hogy a szakasz összefüggő szaggatott vonallal rajzolódik ki a két végpont között. Erre szolgál a >SET STYLE VK utasítás, ahol VK=1–14 a vonal mintakódja.

A mikro grafika karakterek közvetítésével is megoldható. Itt szerepet játszik a gép működési állapota, ami meghatározza a képernyő karakter felbontását. A PLOT utasítás minden I, J koordinátához automatikusan a $10 * I, 8 * J$ koordinátákat rendel. Ha az automatikus koordinátákat megfelelően nagyítjuk, mikro koordinátákat nyerünk:

Állapot	Kar. felb.	Aut. felb.	Mikro kar. felb.
GRAPHICS 2	24*64	240*512	4*240x2*512
GRAPHICS 4	24*32	240*256	4*240x4*256
GRAPHICS 16	24*16	240*128	4*240x8*128

Szerencsés körülmény, hogy az I, J karakter koordináták a programozó számára csaknem közvetlenül (>PLOT 2*I, 4*J; >PLOT 4*I, 4*J; >PLOT 8*I, 4*J) alakulnak mikro karakter koordinátákká.

Ha az I, J koordinátákkal adott karakter belsejében akarunk egy mikro karaktert elérni, akkor a

$$>PLOT NJ * (J-1) + MJ, NI * (I+1) + MI$$

utasítást alkalmazzuk, ahol N és M a gép állapotától és a koordináta jellegétől függ:

Állapot	NI	MI	NJ	MJ
GRAPHICS 2	4	1–39	2	1–15
GRAPHICS 4	4	1–39	4	1–31
GRAPHICS 16	4	1–39	8	1–63

Színes technika

>SET BORDER CK a képernyő keret színét adja meg a CK=0–15 színsorszámának megfelelően. A választható színek száma függ a gép állapotától. A GRAPHICS 2, 4, 16 állapotok 2, 4, 16 szín alkalmazását engedik meg. A színválasztásról a >SET PALETTE CK, CK, illetve >SET PALETTA CK, CK, CK utasítással gondoskodunk. A GRAPHICS 16 esetben nem kell színt választani. A CK a megengedett színek kódszámaival azonosítható. Kódszámok és sorszámok: 0,0; 1,1; 4,2; 5,3; 16,4; 17,5; 20,6; 21,7; 64,8; 65,9; 68,10; 69,11; 80,12; 81,13; 84,14; 85,15.

Ha 2 vagy 4 színt választhatunk, akkor a fenti kódokat használjuk, de a választott színek saját sorszámot (1,2; 1, 2, 3, 4) kapnak. Mindhárom állapotban színezhajjuk a képernyőt: >SET PAPER S(CK), valamint a jeleket : > SET INK S (CK) azoknak a színeknek a sorszámai közül választva, amelyeket a SET PALETTE utasítással lekötöttünk.

Az utasítások sorrendje: >SET BORDER CK : GRAPHICS V : SET PALETTE ,, ,, ,, : SET PAPER S(CK) : SET INK S(CK) : PLOT X, Y vagy PRINT AT I, J : A stb.

VT-16 grafika

A P122-nél részletes magyarázatot adtunk.

7. IRODALOMJEGYZÉK

Gépkönyvek

- [1] *Commodore-64 Micro Computer HANDBUCH.
- [2] MPS-801 DOT MATRIX PRINTER USER'S MANUAL.
- [3] FLOPPY DISK VC1541 Bedienunghandbuch.
- [4] 1530 DATASETTE UNIT OPERATING INSTRUCTIONS. Commodore &.
- [5] *ABC-80 USER'S MANUAL. B. Rádiótechnikai Gyár.
- [6] *HT-1080Z és HT-2080Z Használati útmutató.
- [7] Programozás kezdőknek. Híradástechnikai Szövetkezet.
- [8] BASIC kézikönyv HT-80Z SZÁMÍTÓGÉPEKHEZ. HSZ.
- [9] *Texas Instruments TI-99/4 A Computer Bedienungsanleitung.
- [10] TI Extended BASIC FOR THE TI-99/4 HOME COMPUTER. Texas &.
- [11] *PRIMO felhasználói kézikönyv.
- [12] Commodore típusú soros perifériák kezelése PRIMO számítógéppel (kézirat). COSY.
- [13] *TV COMPUTER kezelési útmutató. (Garai G., Gáspár Cs., Velkei Z.). VIDEOTON.
- [14] BASIC programozási segédlet (Balogh László. Sz. B.: Garai G., Gáspár Cs., Velkei Z.). VIDEOTON.
- [15] *VT-16 BASIC (kézirat). VIDEOTON.
WM 4000 OPERATORS MANUAL. WALTERS M. I.
- [16] Donald Alcock: ISMERD MEG A BASIC NYELVET. MK. Budapest. 1983.
- [17] KOCSIS A.: A BASIC I-II-III. SZÁMALK. BP. 1983.
- [18] I.ÁNGOS István: A COMMODORE-64 mikrogép kezelése és programozása. COMPORGAN R. BP. 1983.
- [19] LSI Alk. Techn. Tanácsadó Szolgálat: Mikrogépes Operációs Rendszerek a CP/M operációs rendszer.
- [20] Ury László: COMMODORE-64 Basic felhasználói kézikönyv.
- [21] Englisch-Szczepanowski: A VC-1541-es lemezegység programozása. DATABECKER-NOVOTRADE. BP. 1985.
- [22] Angerhausen-Englisch-Gerits: Tippek és trükkök a COMMODORE-64-eshez. DATABECKER-NOVOTRADE. BP. 1985.
- [23] Angerhausen-Becker-Gerits-Schellenberger: A BASIC programozás magas iskolája a C-64-esen. DATABECKER-NOVOTRADE. BP. 1985.
- [24] L Englisch: Gépi kódú programozás a COMMODORE-64-esen. DATABECKER-NOVOTRADE. BP. 1985.

8. ANGOL–MAGYAR SZÓJEGYZÉK

A		C	
already	– már	control	– ellenőrzés
and	– és	continue	– folytat
append	– csatol	create	– teremt
argument	– független	cursor	– kurzor
array	– tömb		
at	– nál, nél	D	
auto	– ön ...	data	– adat
available	– felhasználható	date	– dátum
		decimal	– tizes
B		definition	– meghatározás
bad	– hibás	delete	– tölör
base	– kiindulópont	device	– eszköz
bit	– falat	direct	– közvetlen
block	– blokk	directory	– címtár
border	– szegély	disk	– lemez
break	– megszakítás	display	– kirakat
buffer	– puffer	division	– osztás
		do	– csinál
C		dot	– pont
call	– hív	duplicate	– kétszeres
can't	– nem tud		
carriage	– kocsi	E	
cartridge	– kazetta	edit	– szerkeszt
chain	– láncol	else	– különben
character	– betű	end	– vége
check	– ellenőriz	enter	– bejegyez
choice	– választás	erase	– kitöröl
circle	– kör	erros	– hiba
clear	– tiszta	exit	– kilép
close	– bezár	exist	– létezik
code	– kód	extended	– kiterjesztett
command	– parancs	extra	– külön
color	– szín		
common	– közös	F	
compatible	– összeegyeztethető	feed	– agagolás
complex	– összetett	field	– mező
computer	– számítógép	file	– adatgyűjtő
conflict	– összeütközés	fixed	– állandó

F	
floppy	– laza
for	– át
format	– formátum
found	– talált
free	– szabad
from	– tól, óta
full	– teli
function	– függvény

G	
get	– kap, szerez
go	– megy
graphics	– rajzolt

H	
handler	– kezelő
hardware	– fizikai gépr.
head	– fej
hexadecimal	– tizenhatos
home	– otthon

I	
if	– ha
ignored	– nem vette figyelembe
illegal	– törvénytelen
inch	– 2,54 cm
incorrect	– hibás
ink	– tinta
input	– betesz
insert	– beszúr
instruction	– útbaigazít
instrument	– eszköz
integer	– egész
interact	– egymásra hat
interpreter	– tolmács

K	
key	– billentyű
kill	– megsemmisít

L	
left	– bal
length	– hosszúság
let	– enged, hagy
line	– sor, vonal
list	– lista

L	
load	– betölt
locate	– telepít
long	– hosszú

M	
manager	– menedzser
master	– gazda
manual	– kézikönyv
many	– sok
memory	– memória
merge	– egybeolvaszt
mid	– közötti
mismatch	– hibás párosítás
missing	– hiányos
mode	– alak

N	
name	– név
new	– új
next	– következő
no, not	– nem
number	– szám
numeric	– számszerű

O	
off	– le
okay	– rendben
old	– régi
on	– fel
open	– kinyílik
operator	– kezelő
option	– szabad választás
or	– vagy
out	– vége
output	– kitesz
overflow	– túlfolyik

P	
palette	– festéklap
paper	– papír
past	– túl valamin
peek	– kukucskál
pen	– toll
play	– működésbehoz
plot	– ábrázol
point	– pont
pointer	– mutató

P

poke	–	eltesz
position	–	fekvés
power	–	energia
prepare	–	előkészít
present	–	jelen van
press	–	lenyom
print	–	nyomtat
priority	–	elsőbbség
protect	–	véd
put	–	tesz

R

random	–	találomra
rang	–	hatókör
read	–	olvas
ready	–	kész
real	–	valós
record	–	feljegyzés
recorder	–	hangfelvevő
recording	–	felvesz
redimensioned	–	újra dim.
redo	–	átalakít
relative	–	viszonylagos
remakr	–	megjegyzés
reset	–	helyretesz
restore	–	visszaad
return	–	visszatér
rewind	–	visszacsvél
right	–	jobb
run	–	fut

S

save	–	megőríz
screen	–	képernyő
searching	–	keres
sector	–	kőrcikk
select	–	választ
sequential	–	soros
serial	–	soros
set	–	helyez
shift	–	váltás
size	–	terjedelem
software	–	program
space	–	térköz
sprite	–	szellem
square	–	négyzetes
statement	–	utasítás

S

step	–	lépés
stop	–	megállás
string	–	fűzér
style	–	írásmód
sub	–	alatt
subscript	–	index
swap	–	cserél
syntax	–	mondattan
system	–	rendszer

T

table	–	táblázat
tape	–	hangszalag
test	–	próba
that	–	az
then	–	akkor
time	–	idő
too	–	túláságosan
to	–	ig
track	–	pálya
true	–	igaz
twice	–	kétszeres
type	–	típus

U

unavailable	–	elérhetetlen
undefined	–	nem meghatározott
understood	–	értette
unit	–	egység
used	–	felhasznált
user	–	használó

V

value	–	érték
variable	–	változó
verify	–	átvizsgál
version	–	változtat

W

wend	–	irányít
while	–	míg
width	–	szélesség
without	–	nélkül
write	–	ír

Y

yes	–	igen
zero	–	nulla

A kiadásért felelős az Akadémiai Kiadó és Nyomda főigazgatója
A kötéstervezés Fodor Attila munkája
Műszaki szerkesztő: Marton Andor
Terjedelem: 29.3 (A/5) ív – AK 1972 k 8688
A nyomtatást a Nógrád Megyei Nyomdaipari Vállalat, Salgótarján,
a kötetet a Franklin Nyomda, Budapest végezte

