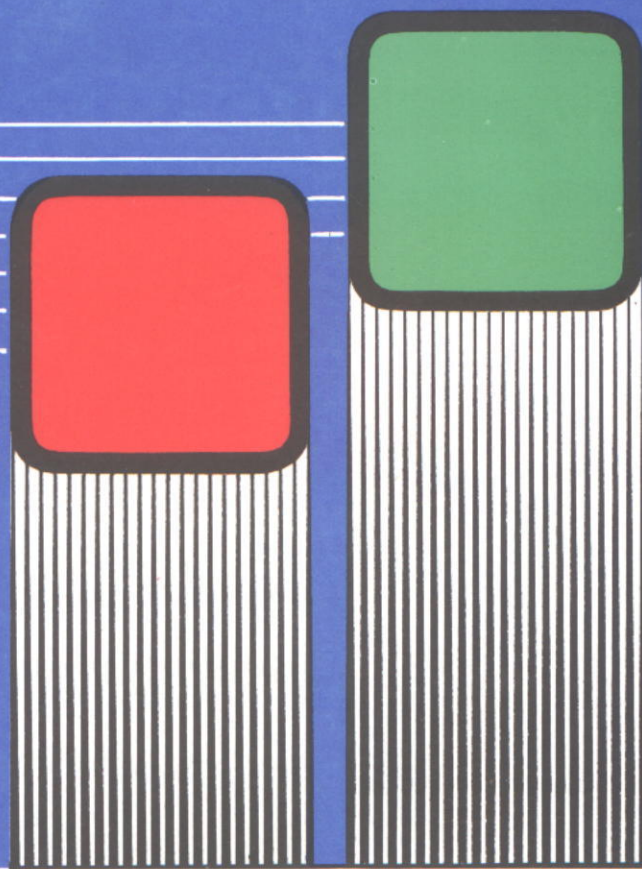


BENEDIKTI ISTVÁN - HUCZKA BÉLA

ISMERKEDÉS AZ  
**ENTERPRISE**  
SZÁMÍTÓGÉPPEL

ENTERPRISE



ISMERKEDÉS AZ  
**ENTERPRISE**  
SZÁMÍTÓGÉPPEL



Benedikti István–Huczka Béla

---

# Ismerkedés az **ENTERPRISE**-szal

Műszaki Könyvkiadó, Budapest, 1987

Lektorálta:  
Pampuk Imre  
okl. vill. üzemmérnök  
SCAN: GAFZ, 2003

ETO: 681.31 ENTERPRISE  
ISBN: 963 10 7766 7

© Benedikti István, Huczka Béla, 1987

Ezzel a könyvvel az OKKFT G-1-es programját támogatjuk

# Tartalom

---

<u>Előszó</u> .....	7
<u>Bevezetés</u> .....	9
Az ENTERPRISE vásárlása .....	9
Géptípusok .....	11
Gépösszeállítások .....	13
Üzembe helyezés; a hardver ismertetése .....	16
<u>A perifériák</u> .....	23
Billentyűzet .....	23
A billentyűk értelmezése .....	27
Kazettás magnetofon .....	29
Mágnesszalagos kazetta .....	32
Botkormány .....	34
Hajlékony mágneslemezes meghajtóegység .....	35
Hajlékony mágneslemez .....	37
Nyomtató .....	37
Külső programtár .....	39
A perifériák felcserélhetősége (kapcsolata) .....	40
<u>Az ENTERPRISE programozása</u> .....	42
Programozási nyelvek .....	43
Gépi kód .....	43
BASIC nyelv .....	48
Programok készítése .....	50
BASIC alapszavak .....	53
BASIC programok készítése .....	145
Assembly nyelvű programozás .....	152

Z80 assembly utasítások .....	154
Aritmetikai és logikai utasítások .....	154
Adatmozgató utasítások .....	164
Be- és kiviteli utasítások .....	172
Programszervező utasítások .....	175
Vezérlésátadó utasítások .....	178
A programok beírása .....	180
Gépi kódú eljárások használata .....	182
<u>Beépített segédprogramok</u> .....	184
Szövegszerkesztés ENTERPRISE számítógéppel .....	184
Szövegek beírása .....	185
Szövegek módosítása, javítása .....	187
Szövegek tárolása .....	190
Grafika a képernyőn .....	190
A grafikus lap .....	191
Függvények rajzolása .....	192
Rajzolás a képernyőn .....	193
Hanggenerálás .....	195
<u>Függelék</u> .....	199

# Előszó

---

Napjainkban nagyon sokféle számítógép van az országban. Vannak, akik többet is használtak már, sőt esetenként néhányat "ki is nőttek". De akadnak olyanok is, akik még nem találkoztak egyetlen számítógéppel sem, akik számára az ENTERPRISE az első.

Könyvünket elsősorban nekik szánjuk. Amit leírtunk, az a könyvtárnyi szakirodalomból kiválasztott, a kezdeti ismerkedéshez nélkülözhetetlen ismeretanyag, kiegészítve egy-két érdekességgel. Célunk a számítástechnika és az ENTERPRISE bemutatása, de csak "első közelítésben". A tudományos igényű, mélyebb ismeretek, és a számítógépben rejlő összes lehetőség feltárását más könyvekre hagyjuk. Arra törekedtünk, hogy az itt leírtak segítségével -- ha nem is a legügyesebben, de -- használni lehessen az ENTERPRISE-t, és a továbbhaladásra több utat vázoljunk, hogy közülük mindenki a neki legjobban megfelelőt választhassa. A Tv-Híradó egyik adásában láthattuk: "indul az ENTERPRISE-akció". A nézők megtudhatták, hogy ezen a néven egy újabb számítógép kerül a budapesti és a vidéki Centrum Áruházak polcaira. Olyan, amelyik házi számítógépként használható. Ez a gép az eddig hozzáférhetőek közül kiemelkedik grafikai és akusztikai tulajdonságaival, 17 000 Ft körüli ára pedig átlagos. (Volt már előtte forgalomban ennél drágább és olcsóbb házi számítógép is.)

Hetekkel az "ENTERPRISE-akció" indításának bejelentése után, a sajtóból újabb érdekességeket tudhattunk meg a gép és magyarországi elterjedésének történetéről. Például azt, hogy van némi rokonság közte és a Videoton TV-Computer között, sőt vannak magyar szakemberek, akik már régóta ismerik az ENTERPRISE-t. Várható tehát, hogy ők Magyarországon "nagyra teszik" ezt a gépet: építenek hozzá különböző hardver- és szoftverillesztőket, ki-



egészítik a már meglévő gyári programokat újakkal. Érdekesen indult tehát az ENTERPRISE magyarországi pályafutása. Érdekes maga a gép is, ismerkedjünk meg vele! A hasonló nevű repülőgép-anyahajót a Tv-Híradó egyik adásából ismerhetjük. Szólnunk kell a név egy más jellegű használatáról is, például a HONG KONG ENTERPRISE című árukatalógusról. Ennek az érdekes kiadványnak (egy-egy példányai nálunk is kaphatók) és a csatahajó-  
nak az azonos elnevezése látványos bizonyítéka, hogy az angol enterprise szónak több jelentése értelmezhető. Az elsősre, a merészségre a csatahajó emlékeztethet, a másodikra, a vállalkozásra a vállalkozók árukatalógusa. Vajon Magyarországon számítógépként mit fog jelenteni az ENTERPRISE? Csak nem azt: merész vállalkozás!? Könyvünkkel azoknak a vállalkozóknak szeretnénk segíteni, akik még nem "számítógépesek", de elég bátrak ahhoz, hogy egy ENTERPRISE ONE TWO EIGHT típusú számítógéppel megpróbáljanak bejutni a számítógép-kedvelők egyre bővülő táborába.

# Bevezetés

---

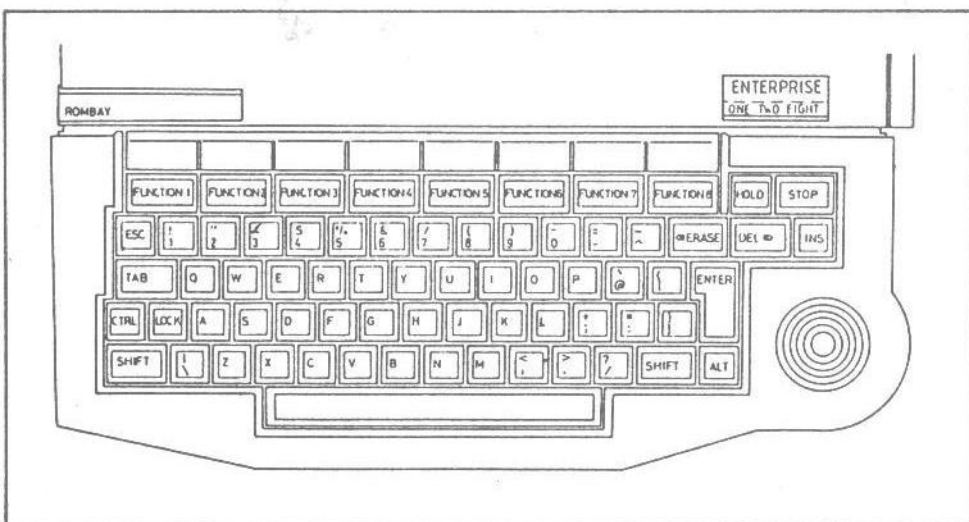
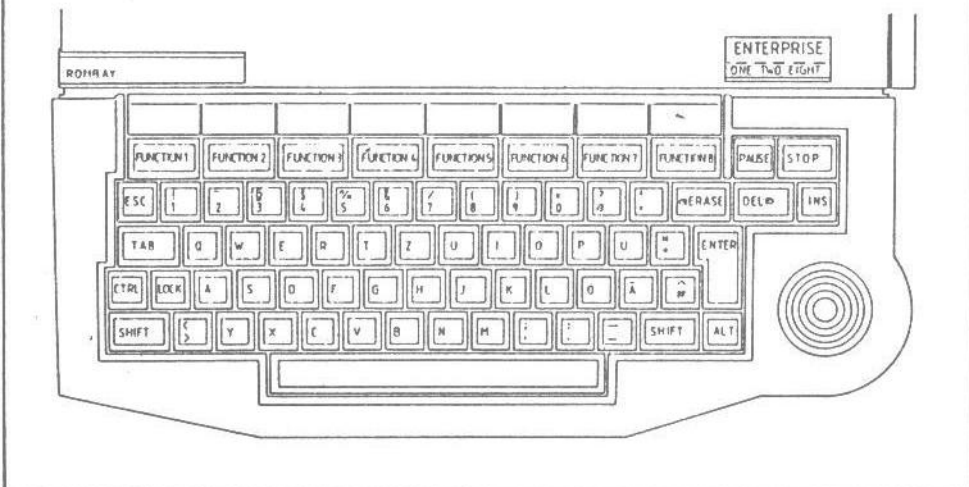
## Az ENTERPRISE vásárlása

A gép vételára nem kis összeg; ekkora összeg kiadásakor általában megfontoltan kell cselekednünk. Könnyű a dolgunk, ha napon-ta benézhetünk az üzletkebe, ahol próbálgathatjuk a gépeket, és nyugodtan érdeklődhetünk az eladótól. (Ő hozzánk képest beavatatott szakember.) Természetesen a vásárlás előtt célszerű min-denféle, a gépet érintő információt -- még a reklám célú szóró-lapokat is -- összegyűjtenünk.

Az ENTERPRISE 128 (pontosabban ENTERPRISE ONE TWO EIGHT) számító-gép a Centrum áruházakban egy, a központi egységgel egybeépít-tett billentyűzetet, egy tápegységet és egy kazettás magneto-font jelent. Ezt az egységcsomagot így, együtt kell megvennünk, annak ellenére, hogy a tápegységről már működtethető központi egység nemcsak ezzel a magnetofonnal, hanem egy már otthon meg-lévővel, köztük magyar gyártmányú kazettás, rádiós magnetofon-nal is használható.

Már a vásárlás előtt feltűnhet, hogy a számítógépeket kétféle billentyűzettel árulják (1. ábra), az angol, ill. a német ábécének megfelelő billentyűzettel. A vásárlás előtt el kell dön-tenünk, hogy gépünket melyikkel tudjuk jobban használni.

Az áruházi egységcsomagnak az az előnye, hogy vásárláskor azt a gépösszeállítást próbálhatjuk ki, amelyiket otthon használni fogunk. Ezen választhatjuk ki a géppel együtt vásárolt progra-mokat is. Nem szabad azonban megfeledkezniünk a leendő képernyő-ről sem, ami különösen a színes gyári (játék-) programok ese-tén okozhat csalódást. Az áruház eladója a gép és a programok előnyösebb bemutatása érdekében általában egy színes tv-hez kapcsolja a gépet. Ha otthon egyszerű fekete-fehér tv-hez csat-lakoztatjuk a gépet, akkor néhány program, amely az áruházban a



1. ábra. Különböző billentyűzetek

bemutatáskor elnyerte tetszésünket, otthon szinte használhatatlanná, de mindenképpen csökkent értékűvé válik.

Gyári programok vásárlásakor feltétlenül próbáljuk ki azokat! Találtunk már olyan "originál angol" programkazettát is, amelyekben csak befűzőszalag volt, a mágnesszalag pedig kimaradt. Ez úgy tűnt fel, hogy kipróbálásakor mindkét oldalon azonnal a szalag végére értünk. Lehetnek olyan kazeták is, amelyeknek

csak az egyik oldala használható. Fontos tehát, hogy a kazettán lévő összes programot beolvassuk, és legalább elindulásukat ellenőrizzük.

A programok kipróbálásakor figyelniük kell arra is, hogy azt a saját gépünkkel azonos típusú gépen végezzük. Eddig csak ránézéssel tettünk különbséget a gépek között, de néhány gyári program is észreveszi a különbséget, és ezt időnként tudomásunkra is hozza, pl. az egyik gépen hibátlanul futó program a másikon csupán hibaüzenetet képes kiírni. Elég kellemetlen, ha a program használhatatlansága csak otthon derül ki, ráadásul a programkazettákat árusító helyeken gyakran látható a "Programkazettákat nem cserélünk." felirat.

### Géptípusok

Akik megvették az ENTERPRISE-t, azok a géppel együtt kapott Felhasználói kézikönyvből megtudhatják, hogy az országba a különböző forrásból beérkező gépek -- a beszerzési forrásoktól függően -- német, angol vagy mindkét nyelvű üzenetek kiírására képesek. Ez több géptípust jelent, mint amennyit a szemlélődő a billentyűzetek különbözőségéből fel tud fedezni. A gépek billentyűzetén alapuló megkülönböztetés tehát csupán felszínes csoportosítás; az igazán típusmeghatározó a gép csak olvasható tárába (ROM-ba) beégetett program. (A ROM a Read Only Memory rövidítése.)

Az ENTERPRISE ROM-ja két részből áll: a belső tárban és a külső programtárban (cartridge) lévő ROM-ból. A géptípusok közötti eltéréseket kutatva mindkettőt külön-külön vizsgáltuk.

Ezt egyszerűen megtehetjük, ha gépünket a hozzá kapott külső tár nélkül próbáljuk használni. Ekkor -- amint azt a Felhasználói kézikönyv is írja -- csak a szövegszerkesztőt tudjuk használni. Mi egy angol és egy német billentyűzetű géppel kísérletezve a következőket tapasztaltuk: Az angol nyelvű gép rendelkezés nélkül működött; igaz, BASIC üzemmódba nem tudtunk áttérni (F8, a nyolcas funkcióbillentyű), ez csak a géphez kapott külső programtár csatlakoztatásával sikerült. A német billentyűzetű géppel ellenben bajban voltunk. Az első nem várt jelen-

ség, hogy a képernyőn megjelent információk (a szövegszerkesztő fejléce, a funkcióbillentyűk értelmezése) és az esetenként felvillanó hibaüzenetek angol nyelvűek voltak. Ennél sokkal zavaróbb volt a szöveg bevitelénél a billentyűzet karakterkiosztásának átrendezése, így nem az általunk beütött karakter jelent meg a képernyőn. Nem volt ugyan nehéz rájönnünk, hogy a korábban német nyelvű billentyűzet angol nyelvűre változott, de így a gépet már nagyon nehéz használni. A két gép belső ROM-ja tehát -- a billentyűzettől függetlenül -- angol nyelvű.

A gép üzeneteinek nyelvét, pontosabban a különböző nyelvű billentyűzetek működtetését és karaktereinek értelmezését tehát a külső programtárban lévő ROM határozza meg. Ezek a csatlakoztatható tárcák külsőleg nagyon egyformának tűnnek, csak alaposabb szemrevételezéssel vehetünk észre közöttük különbséget: pl. az egyik dobozon van egy jel (színes korong), a másikon nincs. A jelölt dobozban levő program kezeli a német nyelvű billentyűzetet, míg a jelöletlen az angolt. Ha a jelöletlen tárcát a német billentyűzetű géphez csatlakoztatjuk, akkor az üzenetek nyelvét és a billentyűzetet angollá változtatjuk, és többé nem tudjuk visszaállítani németre. Nem ismeri továbbá a billentyűzet karakterkiosztását megváltoztató :BRD és :UK parancsokat sem.

Az ENTERPRISE 128 változatait tehát a külső tárc felcserélésével állíthatjuk elő. Az egyes változatok között még lényeges különbség a felhasználók számára biztosított szabad tárterület nagysága. A géppel kapott külső tárcnak, a gép BASIC bővítőjének a típusa a BASIC rendszer bejelentkezésekor megjelenő üzenetből könnyen felismerhető.

Angol nyelvű gép:

Két nyelvű gép:

IS BASIC program 0  
131072 bytes in system  
116121 bytes unused system  
OK

IS BASIC program 0  
131072 bytes in system  
114858 bytes unused system  
OK

Ennek a felhasználható tárterületnek az eltérése magyarázza azt a meglepő eseményt, amit néhány, vásárolt program betöltésekor

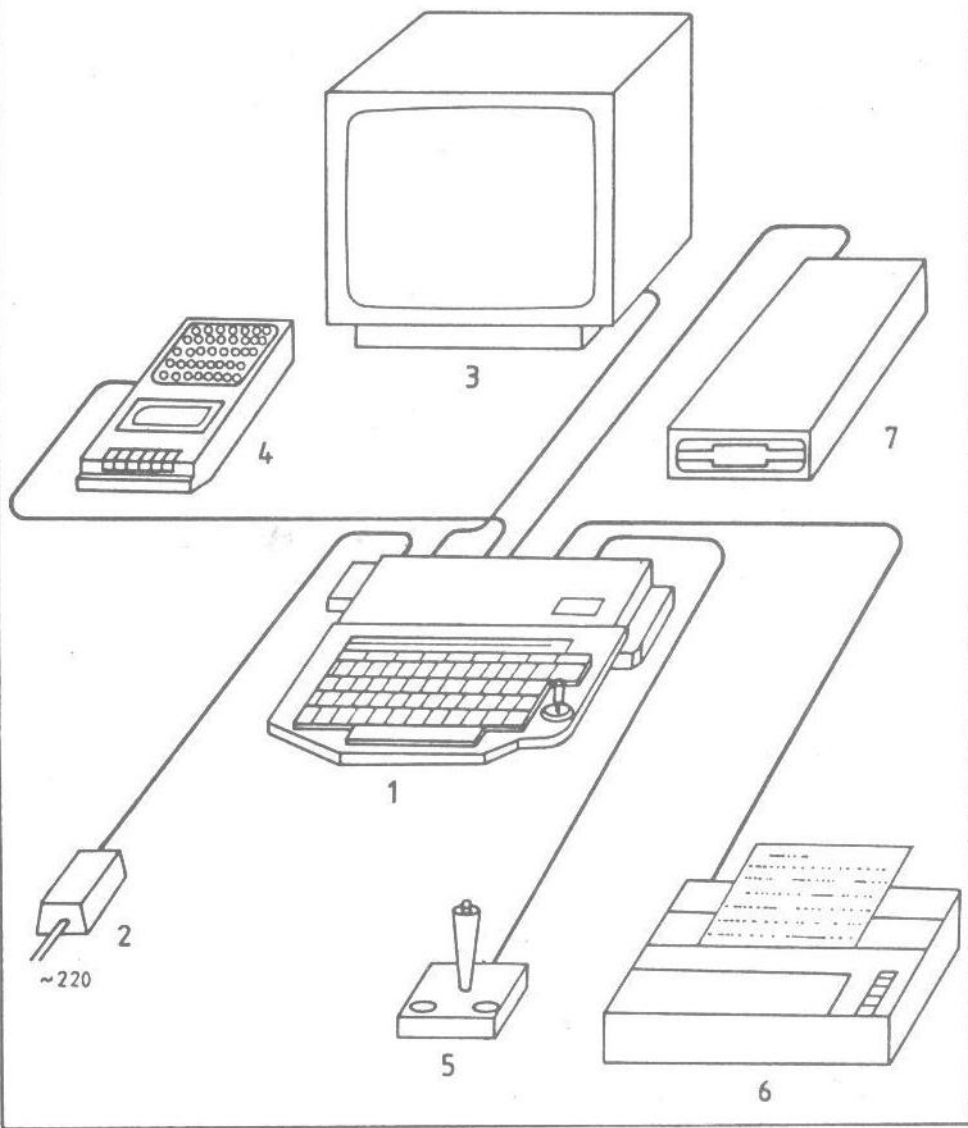
tapasztalhatunk: a program betöltése után annak elindulása helyett a kis tárterületre utaló hibüzenet jelenik meg. Ez tapasztalható például a billiárd játékprogramnál a kétnyelvű gépeken. Az egyes változatok eltéréseiből adódó kellemetlenségek nagy része -- a géphez képest méretben és értékben apróságnak tűnő különböző külső táruk beszerzésével -- elkerülhető lenne, de a külső tár egyenlőre csak a géppel együtt kapható. A változatok azonban egyáltalán nem okoznak gondot, amikor a saját gépünkre magunk írunk programot. Csak a vásárolt vagy cserélt programokra, ill. a más gépre is szánt programjainkra kell vigyáznunk. Figyelnünk kell arra, hogy a kétnyelvű, kisebb felhasználható tárterületű gép nemcsak az üzenetek szövegében más, mint az egynyelvű, hanem a karakterek kezelésében is van beépített többletszolgáltatása. Nem biztosít tehát feltétlen használhatóságot az, ha a program a "kisebb" gépen készült. Mit tegyünk? Programszerző útjainkra legalább gépünk külső tárát vigyük magunkkal!

### Gépösszeállítások

A legegyszerűbb és legigénytelenebb gépösszeállítás (konfiguráció), ha az áruházi egységcsomagot egy Junoszty tv-vel kapcsoljuk össze. E tv előnye, hogy olcsó, ezért érdemes a géphez külön képernyőként megvásárolni. Kényelmesebb, ha a gépet nagyképernyős készülékekhez kapcsoljuk; erre minden, a nálunk forgalomban lévő nem túl régi készülék is alkalmas.

Az ENTERPRISE 128 grafikai és színábrázolási tulajdonságait jól kihasználó legegyszerűbb összeállítás a színes tv-hez kapcsolt és kazettás magnetofon használó gép. Igaz, így a lehetőségeihez képest még egy kicsit lassú az adat- és programbevitel, sőt a kezelése is lehetne kényelmesebb, de a rajzok megjelenítése már kifogástalan.

Az ENTERPRISE billentyűzetével egybeépítve találjuk a botkormányt (pontosabban a kurzormozgató botkormányt), amelyet különösen a játékprogramokban kell sokat használnunk. A kényelmesebb "játékgép" kialakításának egyik kézenfekvő megoldása, hogy a géphez egy különálló botkormányt (joystick, amellyel a játékprogramok számos más gépen is irányíthatók) kapcsolunk (2.



2. ábra. Gépösszeállítás

ábra, 5. bővítegység). Ennek nincs különösebb akadálya, hiszen például a nálunk is közsímert, Videoton gyártmányú botkormány egy megfelelő csatlakozóval azonnal a géphez kapcsolható. A külső botkormány nemcsak kényelmesebb játékot tesz lehetővé, hanem a gép használatának biztonságát is növeli. A beépített

Dotkormány bármilyen hibájának - a játék hevében a meghibásodás veszélye nem kicsi -- javítási ideje alatt számunkra maga a számítógép is használhatatlan.

A munkaeszközzé válás első lényeges lépése a nyomtató illesztése (2. ábra, 6. bővítőegység). Ez -- elsősorban a nyomtatók viszonylag magas ára miatt -- már nehezebb feladat. A számítógépek árusításának megkezdésekor még nem volt meg a géphez ajánlható, árban megfelelő és egyszerűen csatlakoztatható nyomtatótípus. A nyomtatókimenetre azonban a Centronics illesztésű nyomtató -- sajátkezűleg vagy segítséggel készült csatlakozóval -- azonnal ráköthető. Az ára viszont jóval meghaladja magának az ENTERPRISE egységcsomagnak az árát.

A komoly segítőtárrá válás következő lépése a mágneslemezes háttértár -- floppymeghajtó -- illesztése (2. ábra, 7. bővítőegység). Ez sem tartozik az olcsó bővítések közé. Annak éri meg, aki így nagymennyiségű adat gyors mozgatását (tárolását, beolvasását) akarja elérni. A lemezegységgel kiegészített számítógéppel feldolgozott adatok más gépekre is átvihetők, így ott további feldolgozásuk, nyomtatásuk vagy más eszközökön való megjelenítésük válik lehetővé. (A gépkönyv és a gép hátlapjának felirata szerint egy, már létező hálózatba való csatlakozási lehetőség a gép Magyarországon való megjelenésének idején az országban még nem számít valódi lehetőségnek, ezért a cserélhető lemez veszi át az átviteli vonal szerepét.) Ha ehhez még tárbővítés is társul, az egységcsomagban vásárolt játékgépből egy komoly "házi számítóközpontot" hozhatunk létre.

Az ENTERPRISE-hoz kaphatók, ill. készíthetők olyan kiegészítők amelyekkel más, esetleg korábban használt számítógépek megszo- kott programjait is futtathatjuk. Például a Sinclair gépeken futtatott programokat ilyen kiegészítővel gond nélkül használhatjuk az ENTERPRISE-on is.

A továbbiakban a legegyszerűbb -- és legolcsóbb -- összeállítást feltételezve beszélünk az ENTERPRISE-ról.



Ha már megvásároltuk a gépet, bizonyosan szeretnénk otthon mielőbb összeállítani és bekapcsolni. A Felhasználói kézikönyv ajánl egy összeállítási módszert, de ezt nem árt egy-két pontban korrigálni és kiegészíteni. A legegyszerűbb összeállításhoz, a tv készüléket kivéve az egységcsomagban megtalálhatunk mindent. Igaz, a magnetofon csatlakozókábele nem mindkét végén 4 dugaszos, hanem két darab két-két dugaszos kábel, de ez nincs háttással a használhatóságára. Mielőtt bármit is tennénk, és ha van kedvünk barkácsolni, a számítógép tápegységének csatlakozóvezetékére (de lehetőleg a kiefeszültségű oldalon) szereljük a csatlakozóhoz közel egy kapcsolót. Erre a célra bármilyen, vezetékre szerelhető kapcsoló megfelel. Ugyanis nem tudni milyen megfontolásból, de a számítógépen nincs hálózati kapcsoló, és bizony sokszor csak a gép kikapcsolásával lehet az alapállapotot újra beállítani. A hálózati csatlakozó ki-be dugdosása nem célszerű, mert előbb-utóbb az károsodást szenvedhet, és nem is túl kényelmes.

Ha elvégeztük ezt a munkát, lássunk hozzá az összeszereléshez! A legfontosabb szempont, hogy minden perifériát és bővítményegyet a gép bekapcsolása előtt csatlakoztassunk. Ha a BASIC programtárat (cartridge) használni akarjuk, akkor először -- és feltétlenül feszültségmentes állapotban -- ezt helyezzük el a ROMBAY felíratú csatlakozóaljzatba.

A magnetofont csatlakoztathatjuk mind felvételre, mind lejátszásra. (Nem kell kihúzogatni és átrakosgatni a vezetékeket.) A két darab vezeték közül az egyik vezeték vastagabb dugaszolói a gép IN és a magnetofon EAR jelű csatlakozójába dugjuk, a vékonyabbakat pedig a REM 1 és a REM távvezérlő aljzatokba. Ezzel készek vagyunk a programbeolvasásra. De szeretnénk néha programokat elmenteni is! Erre a célra használhatók a második vezeték vastagabb dugaszai. Az egyik végét a számítógép OUT jelű kimenetébe, a másik végét pedig a magnetofon MIC jelű aljzatába helyezzük be. Így felvételeket is tudunk készíteni. Igaz, így két dugó a "levegőben fog lógni", de ezt az esztétikai hibát viseljük el a könnyebb kezelhetőség érdekében. A tv-készüléket vagy a MONITORT a megfelelő felíratú csatlakozókra kell az e

célra készített vezetékkel dugaszolni. A többi perifériát (ha van) tetszőleges sorrendben csatlakoztathatjuk. Ha ezzel készen vagyunk, kapcsoljuk be a tv-t, a perifériákat, majd a számítógépet. Ha a tv jól van behangolva (36-os csatorna), meg fog jelenni a képernyőn a memóriatesztelő, majd az ENTERPRISE felirat; ekkor nyomjunk meg egy tetszőleges billentyűt, és be fog jelentkezni a BASIC rendszer.

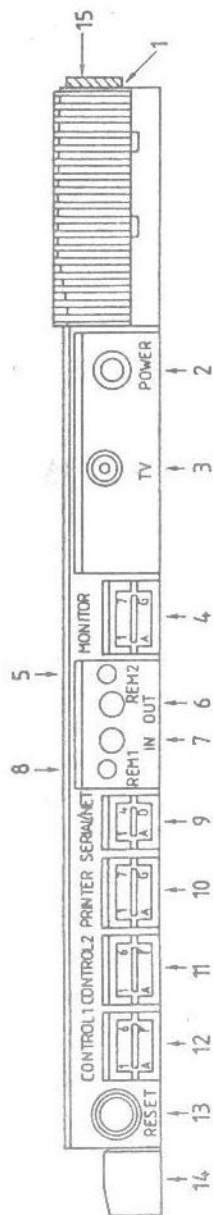
Könyvünknek nem célja a gép belső áramkörének (az un. harvdré-  
rének) részletes ismertetése, de azért egy-két dolog a felhasználók számára érdekes lehet. Az ENTERPRISE számítógép központi egysége Z80 A jelű mikroprocesszorból és a hozzá tartozó kiegészítésekből áll. Ezeket két panelre helyezték el, egy nagyobbra és egy felette található kisebbre. Az utóbbin a második 64 K-s tárrész van. A 3. ábrán csak a fontosabb csatlakozókat tüntettük fel. Ezek a következők:

1. Az 5 V-os stabilizátorok, a hűtőfelülettel. A gép működéséhez szükséges stabil feszültségeket ez az egység állítja elő.
2. A tápegység csatlakozóaljzata.
3. Modulátoregység. A tv 36-os csatornájára ez szolgáltatja a videójelet.
4. A monitor csatlakozósávja. Erre csatlakoztatjuk a monitort egy megfelelő csatlakozó segítségével. A különböző értékek jelentése:

1	2	3	4	5	6	7
A	B	C	D	E	F	G

1 üres	A zöld szín
2 testpont	B testpont
3 kék szín	C összetett videójel
4 piros szín	D vízszintes szinkron
5 szinkronizáló jel	E függőleges szinkron
6 üzemmód kapcsoló	F üres
7 jobb hangcsatorna	G bal hangcsatorna

5. Kettes számú magnetofon-távvezérlő dugaszolóaljzat (REM 2).
6. Kazettás magnetofon csatlakoztatásakor a kimenetet ide lehet csatlakoztatni, és egy 3,5 mm átmérőjű dugasszal fejhallgatót is (OUT).



3. ábra. Csatlakozók

7. Kazettás magnetofonról való programbetöltésre szolgáló csatlakozó (IN).

8. Egyes számú távvezérlő (REM 1).

9. Soros vonal és más számítógépekhez való csatlakoztatás pontja. Ez megfelel az RS 232 szabványnak (SERIAL/NET). A csatlakozósáv érintkezési pontjainak a jelentése következő:

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>
A	B	C	D

1 testpont

A referencia

2 üres

B üres

3 TXD adatjelkimenet

C RTS adatkérés

4 RXD adatjelbemenet

D CTS adattörlés, nyugtázás

10. A párhuzamos nyomtató csatlakozósáv-érintkezői (PRINTER). A D0-tól a D7-ig az adatkimenet csak egyirányú. Jelentésük:

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>
A	B	C	D	E	F	G

1 testpont

A testpont

2 Adatvétel-nyugtázó

B Jelzés, adat van a kimeneten

3 D4 (adatkimenet)

C D3

4 üres

D üres

5 D5

E D2

6 D6

F D1

7 D7

G D0

11. és 12. Külső botkormányok csatlakozóaljzata (CONTROL 2 és CONTROL 1). A két csatlakozósávon az 1, 2, 4, A, B csatlakozási pontok össze vannak kötve. Érintkezőik jelentése:

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>
A	B	C	D	E	F

1 testpont

A közös pont

2 üres

B üres

3 üres

C üres

4 +5V

D jobbra

5 balra

E le.

6 fel

F tűzgomb

13. RESET gomb.

14. A bővítés csatlakozósávja. Az érintezők jelentése:

1	2	3	4	.....	33
34	35	.....	66		

Erre a csatlakozósávra gyakorlatilag a Z80 mikroprocesszor összes kivezetése, valamint a bővítéshez szükséges összes pont ki van vezetve.

1 bal hangcsatorna	34 jobb hangcsatorna
2 $\overline{WR}$	35 $\overline{RFSH}$
3 $\overline{IORQ}$	36 $\overline{RD}$
4 üres	37 üres
5 $\overline{NMI}$	38 $\overline{MREQ}$
6 A9	39 A8
7 A11	40 A10
8 A13	41 A12
9 A15	42 A14
10 A1	43 A0
11 A3	44 A2
12 A5	45 A4
13 A7	46 A6
14 D1	47 D0
15 D3	48 D2
16 D5	49 D4
17 D7	50 D6
18 $\overline{INT}$	51 $\overline{RESET}$
19 testpont	52 $\overline{WAIT}$
20 testpont	53 $\overline{M1}$
21 testpont	54 1 MHz órajel
22 testpont	55 4 MHz órajel
23 testpont	56 8 MHz órajel
24 EC1	57 EC0
25 EC3	58 EC2
26 A16	59 EXTC
27 A18	60 A17
28 A20	61 A19
29 14 MH órajel	62 A21
30 függőleges szinkron	63 testpont
31 Üres	64 testpont
32 testpont	65 vizszintes szinkron
33 +9V	66 +9V

Azoknak a jelennek az értelmezését, amelyek közvetlenül a Z80 A processzor kivezetései, később megtalálhatjuk. Az A16-tól A21-ig terjedő kivezetések a tár bővítésére használhatók.

15. ROMBAY csatlakozóaljzat. Ide csatlakoztatható a külső ROM.

A központi egység panelján található még:

- Csatlakozóaljzatok a billentyűzetről jövő vezetékek számára.
- A második áramköri lap. Ez alatt helyezkedik el a grafikus- és hangáramkör.
- A második nyomtatott áramköri lapon lévő 64 K-s RAM.
- Az alaplapon lévő 64 K-s RAM.
- A belső ROM. Ez tartalmazza a beépített szövegszerkesztőt és az operációs rendszert.
- A Z80 A mikroprocesszor, amely a számítógép lelke, 4 MHz-es órajellel működik. Részletesen itt nem foglalkozunk vele, csak azokat a kivezetéseket közöljük, amelyek a bővítőcsatlakozóra is ki vannak vezetve.

A0--A15 (Address Bus): 16 bites címbusz. Ezen jelenik meg a tár olvasásakor vagy írásakor a tárcím. Az A0 a legkisebb helyértékű bit.

D0--D7 (Data Bus): 8 bites kétirányú adatvonal. A processzor és a külső eszközök közötti adatforgalom bonyolódik rajta.

$\overline{M1}$  (Machine Cycle One): Utasításlehívó gépi ciklust jelző kimenet.

$\overline{MREQ}$  (Memory Request): Háromállapotú kimenet. Jelzi, hogy a címvonalon érvényes tárcím van.

$\overline{IORQ}$  (Input/Output Request): Azt jelzi, hogy a címvonal alsó felén érvényes I/O cím van. A címvonal felső részén (A8-A15) perifériacímzéstől függően az A vagy a B regiszter tartalma található.

$\overline{RD}$  (Read): Ez a kimenet azt jelzi, hogy a processzor a periféria vagy a tár felől kész az adatfogadásra.

$\overline{INT}$  (Interrupt Request): A maszkolható megszakításkérés bemenete. Ezt az I/O elemek kérhetik.

$\overline{NMI}$  (Non Maskable Interrupt): A nemmaszkolható megszakításkérés vonala. Hatására a processzor az éppen futó programot megszakítja.

$\overline{\text{RESE}}\overline{\text{T}}$ : Ezen a bemeneten a megfelelő alacsony szint hatására a processzor alaphelyzetbe kerül.

$\overline{\text{WR}}$  (Write): Ennek a kimenetnek a jele azt jelzi, hogy az adatvonalon érvényes kimenő adat van.

$\overline{\text{RFSH}}$  (Refresh): E kimenet állapota jelzi, hogy a címvonal alsó 7 bitje frissítési címet tartalmaz. A címvonal felső részén az I regiszter tartalma jelenik meg.

$\overline{\text{WAIT}}$ : Az e bemenetre adott alacsony jelszinttel lehet a processzornak jelezni, hogy a címzett tár vagy a periféria nem kész az adatátvitelre.

Aki többet szeretne megtudni a Z80 mikroprocesszorról, az a szakirodalomban részletes leírásokat találhat.

-- A videó--grafikus integrált áramkör (NICK).

-- Speciális hanggeneráló IC (DAVE).

A panel mellé szerelték be a gép saját hangszóróját.

# A perifériák

---

A számítógépek közül ismertek például a ZX81, a SPECTRUM, a Commodore-64, a Commodore-16, a Commodore PLUS 4, a VT TV-Computer vagy a most még nálunk újnak számító ENTERPRISE 128 típusú gépek. Megkülönböztetésük alapja a központi egység, ami az esetek többségében megfelelő megkülönböztetés, de akik használják is valamelyik géptípust, azok tudják, hogy ugyanaz az alapgép a hozzá kapcsolt kiegészítő egységektől függően más-más tulajdonságú géppé válhat. (Gép-összeállítási példáinkkal az ENTERPRISE néhány eltérő tulajdonságú változatát mutattuk be.) Ezeket a kiegészítő eszközöket -- amelyek gyakran nemcsak egyféle számítógéppel kapcsolhatók össze -- perifériáknak nevezük, az alapgépet pedig központi egységnek.

A perifériák közül a billentyűzettel (gyakran nevezik klaviatúrának vagy tasztatúrának is), a képernyővel (monitor vagy display), a kazettás magnetofonnal (egyszerűen a magnóval) és a kazettával, a nyomtatóval (printer), a mágneslemezes meghajtóval (floppy drive) és a mágneslemezekkel, valamint néhány egyéb eszközzel foglalkozunk.

## Billentyűzet

A számítógép-tulajdonosok egy része -- például az ENTERPRISE tulajdonosok -- számára a billentyűzet nem igazi periféria, ugyanis nem csatlakoztatható a központi egységhez, hanem azzal elválaszthatatlanul egybe van építve. Igazi periféria viszont az IBM számítógépeknek és sok más gépnek a billentyűzete, amelyek a központi egységhez kábellel csatlakoznak. Ez fontos tulajdonsága azoknak a gépeknek, amelyeket a gyártóik különböző



nyelvtérületeken kívánunk értékesíteni. (Ez elég gyakran előfordul, és nem csak nemzetközi cégeknél.) Például a magyar gyártmányú TAP-34 intelligens terminálhoz is lehet latin betűs és cirill betűs billentyűzetet csatolni aszerint, hogy Magyarországra vagy a Szovjetunióba készül.) Az ENTERPRISE billentyűzetének cseréje magának a gépnek a cseréjét jelenti.

A billentyűzetet arra használjuk, hogy a gép tárolójába adatokat beírjunk, ill. az egyes feladatok elvégzésére utasítsuk a gépet. Vannak olyan adatbeviteli feladatok, amelyek könnyű elvégzéséhez különleges billentyűzetet alakítottak ki. Ezek közül kiemelkedik a kisméretű numerikus billentyűzet, amelyen csak számjegyek és néhány, a számok leírásánál használt jel (+, -, . és szóköz) van. Ezek előnye, hogy rajtuk az összes billentyű az egyik kézzel, a kéz felemelése nélkül elérhető. Az összeadógépeken végzett, kizárólag számokból álló adatrögzítés korában nélkülözhetetlenek voltak. Amennyiben betűt és jeleket egyaránt tartalmazó billentyűzetet kell kialakítani, akkor ez az írógépek billentyűzetére fog emlékeztetni. Ilyennel működik az ENTERPRISE, és valamennyi személyi számítógép is.

Az ENTERPRISE billentyűzetét látva azonnal feltűnik, hogy a billentyűk különböző színűek. Vannak fekete, zöld, kék billentyűk és egy piros, ráadásul vannak olyanok is, amelyek alakra is eltérnek a többiektől. Az alakok és az egyes színek megválasztása formatervezői feladat, ami fontos kérdés, de ezzel mi itt nem foglalkozunk. Számunkra most az azonos színű billentyűk csoportjai érdekesek; a színek szerinti csoportosítás ugyanis egyben a (szokásos) használatuk szerinti csoportosítás is.

-- Fekete billentyűk (karakterbillentyűk). Ezekből van a legtöbb; ilyenek a betűk, számok és a különböző jelek. Elhelyezésük a két általánosan elfogadott, szabványosnak tekinthető QRTY és QRTZ elrendezés. (QRTY az angol, és QRTZ a német nyelvű gépeken). Minden billentyűről leolvashatjuk, hogy önmagában melyik betűnek, számnak vagy jelnek, azaz melyik karakternek a bevitelére használható. A számok és jelek billentyűin két karaktert találunk. Ezek az írógépeken szokásos kisbetű--nagybetű váltó használatának lehetőségéből adódnak. Az írásjeleknek, mint tudjuk, nincs két alakjuk, viszont sok jel van, így elhe-

lyezésüknél természetes volt kibaszálni a váltóbillentyűzet változtatási lehetőségét. (Ezért már az írógépeken is a kisbetűket jelentő váltóállást alsó állásnak, a nagybetűs váltóállást pedig felső állásnak nevezték.) A billentyűzeten látható feliratok csak az általánosan használt jelentéseket mutatják. Természetesen az egyes programok kivételesen megváltoztathatják az általános karakterelrendezést, vagy egészen más, sem az írógépen, sem a kézírásban nem szokásos karakterként is értelmezhetik az egyes billentyűket. Az ilyen átértelmezéseknek két alapvető típusa van. Az egyik típus -- a kis- és nagybetű váltójához hasonlóan -- valamilyen kiegészítő billentyű lenyomása után változtatja meg a billentyűzet eredeti jelentését, majd felengedésekor visszaállítja azt. Ilyen megoldást használ például az ENTERPRISE szövegszerkesztője is. Másik megoldás, ha a programban írjuk át a billentyűkön lévő karaktereket, mintha a programba építenénk be az előzőekben a gépkezelőre bízott váltó lenyomását és felengedését. Ilyen például a programba is beépíthető :BRD és :UK utasítás. Ezek érdekes hatását akkor is tapasztalhatjuk, ha nem kétnyelvű gépünk van, de hozzájutunk a német nyelvű DEMO kazettához. Ha ezt elindítjuk, a program hibajelzéssel megáll. Indítsuk újra, de most a :BRD utasítást követő utasításról ...!

-- Zöld billentyűk (vezérlőbillentyűk). Az ENTERPRISE használatához -- mint minden, párbeszédre alkalmas személyi számítógép használatához -- elengedhetetlen néhány irányító-, ill. vezérlőjel bevitele is. Erre valók ezek a felirattal (névvel) ellátott zöld színű billentyűk. Hatásuk programonként változhat ugyan, de van egy szokásos értelmezésük. Általában a fekete billentyűk jelentését változtatják meg -- tehát váltónak tekinthetők. Ezek a SHIFT (mindkét oldalon), ALT és CTRL billentyűk. A SHIFT legjellemzőbb értelmezése: nagybetű- és kisbetű-váltás. Az ALT és a CTRL különleges karakterkiosztást -- például ALT a szövegszerkesztőben -- létrehozó váltóként értelmezhető. A LOCK billentyű lenyomása a váltók lenyomott állapotát rögzíti. Az ENTER jellegzetes és szokásos jelentése: a billentyűzés vége (például egy szám összes számjegyének beírását a programban legtöbbször az ENTER billentyű karakterének felisme-

résével állapítják meg). A DEL, INS, ERASE és TAB billentyűket a karaktersorozatok bevitelének megkönnyítésére használjuk (például a szövegszerkesztőben). Ilyenkor ez a jelentésük: a DEL a kurzor előtti karaktert törli, és a jobb oldali karaktereket felzárkóztatja. Az INS a jobb oldali karaktereket egy hellyel eltolva egy karaktert beszúr; az ERASE a sorban a kurzortól jobbra álló összes karaktert törli; a TAB a karakterek beírását a sor egy előre meghatározott helyétől folytatja (tabulál). A HOLD (német nyelvű billentyűzeten a PAUSE) leütésével általában a program futását lehet felfüggeszteni addig, amíg azt ismét le nem nyomjuk. Az ESC legjellemzőbb alkalmazása a programok futásának irányítása (például a DEMO programokban a következő programra való áttérés jelzésére használták).

-- Piros billentyű. A STOP az egyetlen piros színű billentyű. Hatása az éppen futó program azonnali -- mondhatnánk "ész nélküli" -- leállítása. Természetesen van olyan program is, amelynek futása alatt ez nem igaz. Próbáljuk ki például a sakkprogramban!

-- Kék billentyűk (funkcióbillentyűk). A felső sorban található nyolc funkcióbillentyű használatára olyan kevés az elfogadott szabály, hogy állandó nevet sem kaptak. Az ENTERPRISE beépített programjaiban lévő alapértelmezésüket a billentyűsorozat feletti tárolóban elhelyezett cserélhető feliratsor mutatja. Fontos jellemzőjük, hogy lenyomásukat a programban akkor is lehet érzékelni, ha éppen nem a billentyűzetről beérkező jeleket fogadó és feldolgozó programrész fut; azaz a funkcióbillentyűk megszakíthatják a programok futását. Ezeknek a megszakításoknak az értelmezése, kezelése az egyes programok feladata, ami azonnal érthetővé teszi, hogy miért nincs szabványos vagy elfogadott jelentésük.

Programok készítésekor fontos lehet számunkra, hogy hogyan működnek az egyes billentyűk. Ezt azonban itt nem ismertetjük részletesen, csupán a működési módoknak a programozók számára legfontosabb sajátosságait mutatjuk be.

Képzeljünk el egy olyan gyárbejáratot (portát), ahol van egy nagykapu az autók számára és egy személybejárat. A nagykapu előtt pedig egy parkoló. A portás a nagykaput időnként kinyit-

ja, és beengedi az előtte várakozót, vagy megvárja, amíg jön valaki. A személybejáraton jönnek az itt dolgozók és az ügyfelek. Az ügyfelek azonnal benyitnak a portáshoz, megtudni, hogy hova menjenek és mit csináljanak. A dolgozók az érkezéskor a bejáratnál lévő táblán a nevük mellé elhelyezik az "itt van", távozáskor pedig az "elment" jelzést. A portás akkor nyit kaput, ha szólnak neki (például a processzor). Majd a dolgozók jelenléti tábláját figyelembe véve igazítja el az érkezőt.

-- Ügyfelet képviselő, megszakítást okozó billentyűk: a HOLD (PAUSE), a STOP és a funkcióbillentyűk.

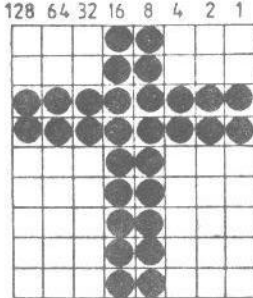
-- Dolgozót képviselő módosítóbillentyűk: SHIFT, ALT, CTRL és LOCK.

-- Nagykapun érkezőt képviselő az összes többi billentyű.

## A billentyűk értelmezése

Tapasztalhattuk, hogy az ENTERPRISE billentyűzetén olvasható feliratok időnként nem érvényesek, megváltozik a karakterek elrendezése. Feltételezhetjük tehát, hogy az egyes billentyűket és jelentésüket programmal kapcsolták össze. Ezt az összekapcsolást legegyszerűbben ismét egy hasonlattal írhatjuk le. Számozzuk meg az összes billentyűt és tekintsük úgy, mintha a feliratok helyén a billentyűk sorszáma lenne. Minden egyes billentyű lenyomásával annak sorszámát adjuk meg. Az értelmezőprogrammal ezt a sorszámot cseréljük át az éppen adott karakterkészlet felsorolásában a neki megfelelő sorszámú karakterré. A billentyűzet átrendezése eszerint nagyon egyszerű: kicseréljük a karakterek felsorolását. Az ENTERPRISE billentyűzetek alapértelmezésben az egyik szabványos karakterkészlet, az ASCII karakterkészlet kódjait kapják. (L. a függelékét!)

A párbeszéden alapuló -- interaktív -- számítógép használatának nélkülözhetetlen eszköze egy, a felhasználó beavatkozásait és azok hatását visszajelző berendezés. Az ENTERPRISE-hoz, mint a legtöbb hasonló számítógéphez, erre a célra képernyőt használunk. Általában ez egy tv-készüléket jelent, de a MONITOR felirátú csatlakozón keresztül különleges, a számítógépek feladatainak ellátására tervezett képernyős megjelenítő (monitor) is



4. ábra. Karakterkialakítás

csatlakoztatható a géphez. Működési elvük azonos: a megjelenítéssel soronként pontokra osztják, ezeknek a pontoknak a színe, ill. a fényereje által alakul ki az üzenet. Az ENTERPRISE képernyőjén alapértelmezésben 24 sor, soronként 40 karakterhelyen lehet elhelyezni az üzeneteket. Minél több pontra osztjuk a képernyőt, és minél állandóbbá tudjuk tenni az egyes pontok tulajdonságait, annál kellemesebb képet kapunk, amit esetleg még a színek megválasztásával javíthatunk. A különböző képernyők felépítésével, működési elvükkel részletesebben nem foglalkozunk, ismertetjük viszont az ENTERPRISE képernyőkarakterjeinek kialakítási módját.

A képernyőn megjelenő üzenet számunkra értelmes jeleit (karakterjeit) a ROM-ban lévő programmal tesszük láthatóvá. A karakterképet egy kilenc sorból és nyolc oszlopból álló, téglalap alakú rész pontjainak kitöltésével alakítjuk ki. Ehhez soronként leírjuk, hogy melyik pont legyen -- az egyszerűség kedvéért fekete-fehér tv-t feltételezve -- fehér és melyik fekete. Például ha egy olyan keresztet ábrázoló karaktert akarunk leírni, amilyen a 4. ábrán látható, akkor ezt a következő számsorozattal tehetjük: 24, 24, 255, 255, 24, 24, 24, 24, 24. A karakterképkialakítási lehetőségek száma az elmondottak alapján igen nagy, még akkor is, ha a képernyő előtt ülve néhányat egymástól csak kevéssé különbözönek látunk. Közülük a szokásos karakterképek leírását automatikusan használhatjuk a ROM-ban lévő programmal, nekünk csak a saját különlegességeinket kell leírni. A DEMO kazettákon lévő karakterdefiniáló programmal tervezhetünk karakterképeket, és terveinket a valóságos méretben ellenőriz-

hetjük is. A jónak ítélt karakterkép leírását is leolvashatjuk a képernyőről. (A DEMO programmal ugyanis a képernyőn -- a programhibákból származó, esetenként kicsit zavaró üzenetmaradványokkal kiegészítve -- megjeleníthetjük az utoljára definiált karakter sorszámát és leírását is. A sorszám a képernyőn látható 32 karakter sorrendjére vonatkozik, ahol az első karakter száma: 0.)

A német nyelvű gépek :BRD és :UK utasítása is átírja a karakterképeket. Ez különösen jól látszik, ha egy jól teleírt képernyő alsó sorainak egyikében adjuk ki a nyelvválasztó parancsot. A korábban értelmes szöveg -- a betűk helyére kerülő furcsa jelek hatására -- olvashatatlanává válik. Saját programjainkban ehhez hasonló, tudatosan előírdézett karakterkészlet-átalakítással sokkal érdekes hatásokat érhetünk el.

### Kazettás magnetofon

A hangrögzítés régóta ismert eszköze a kazettás magnetofon, vagy egyszerűen a kazettás magnó. Az áruházi ENTERPRISE egységcsomagban megvásárolt készülék a kortársai körében szerénynek látszik, amit jól mutat az ára is: központi egység nélkül (ha önállóan lehetne kapni) 3000 Ft alatt lenne az ára, szemben a gyorsmásolós kazettás sztereó magnetofonok 12 000 Ft feletti átlagárával. A számítógép perifériájaként használt magnónak tehát nem szükséges a legjobbnak lennie. Egyetlen elengedhetetlen feltételt kell csupán teljesítenie: legyen csatlakoztatható a központi egységhez. Az egységcsomagban árult -- megkülönböztetésül nevezzük ezt eredetinek -- magnó ilyen. Számítógép-perifériaként való használhatóságát mindenki számára jelzi a "computer program recorder" felirat. Előnyös a telepről való működése, mert akkor kevesebb csatlakozókábelre van szükség. Az eredeti magnónkhoz kapott hálózati kábel elég rövid, emiatt hálózatról működtetve a csatlakozóaljzattól nem lehet nagyobb távolságra elhelyezni. Időnként pedig erre határozottan szükségünk van, ilyenkor is segíthet az elemről való működtetés. Elsősorban a már rögzített programok és adatok beolvasását érdemes így megkísérelnünk, hiszen nagy baj nem történhet; adatot, különösen programot jobb a hálózatról működő magnóval rögzíte-

ni. Az -- esetleg gyengülő -- elemekkel készített felvételeket ajánlatos azonnal ellenőrizni, sőt bizonytalanság esetén azonnal megismételni, így későbbi kellemetlenségeket előzhetünk meg.

Az ENTERPRISE-hoz perifériaként minden további átalakítás nélkül használhatók azok a magnók, amelyeken külső mikrofon (általában MIC jelzésű) és fejhallgató (általában EAR jelzésű) csatlakoztatására kialakított aljzat van. Ez nem különleges igény, inkább szabványosnak tekinthető elvárás. Könnyen kapcsolhatunk tehát a géphez más magnókat is perifériaként, és még külön összekötőkábel sem kell, jó az eredeti magnóé. Például a Videoton gyár RM 4640 jelzésű kazettás magnós táskarádiója is így csatlakoztatható az ENTERPRISE-hoz.

Az ENTERPRISE-t megelőzően is voltak már magnóval működő személyi számítógépek az országban. Érdekes, hogy az ezek jelentős hányadát kitevő Commodore gépek olyan, kifejezetten számítógéperiféria céljára tervezett magnókat használnak, amelyeket csak különleges átalakítóval tudunk más gépekhez kapcsolni. (Igaz, az eredeti magnónk sem kapcsolható közvetlenül a Commodore gépekhez.) Eddigi tapasztalataink alapján ezek a különleges kazettaegységek olyan minőségi, szolgáltatásbeli előnyöket nem tudnak nyújtani, ami miatt érdemes volna ragaszkodni csatlakoztatásukhoz.

A perifériaként egyszerűen használható, MIC és EAR csatlakozóaljzattal ellátott magnók közül nem mindegyiket lehet távvezérelni, például a Videoton említett készülékét sem. Az eredeti magnónk REM feliratú csatlakozóját ezeknél a magnóknál kézi vezérléssel kell pótolnunk. Ez ugyan egy kis kényelmetlenséget okoz, de némi odafigyeléssel a magnós rádiót is sikerrel használhatjuk a géphez kapott DEMO kazetta programjainak futtatására. Természetesen sokkal jobb, ha automatikusan, a gép beolvasóprogramjával vezéreljük a magnót, de ha nincs más, átmenetileg kézi vezérléssel is dolgozhatunk.

A bevitel megkezdése előtt ajánlatos a szalagot a keresett program elé vagy ahhoz közelre állítani. Ehhez a számlálószerkezetet célszerű használnunk. Hiányuk általában nem okoz gondot, annál inkább a mutatott számértékük! Ha magnót cserélünk,

rendszerint nem bízhatunk a számáló által mutatott számértékekben. A rosszul beállított magnószalag miatt a beolvasáskor esetleg nincs meg a keresett, de a szalagon valójában meglévő program. Ez a kisebbik baj. Sokkal kellemetlenebb következményekkel jár, ha a szalagra íráskor a szabad hely helyett egy már foglalt rész elé állítjuk a kazettát. Ekkor ugyanis a szalagra írással az ottlévő értékes programot vagy adatokat tünethetjük el.

Az elmondottakból adódik: jobb, ha a számítógép magnóját nem cserélgetjük. Emellett szól még a kazetta felülírása elleni védelem alkalmazása is, amely a kazetta és a magnó közös lehetősége. Ha a kazetta hátoldalán lévő füleket kitörjük, akkor egy átlagos magnóval nem tudunk a kazettára jeleket felírni. Figyelnünk kell arra, hogy nem minden magnó ilyen. Az írásvédelem működéséről a következőképpen egyszerűen meggyőződhetünk: a feszültségmentes magnón egy írásvédett, majd egy írható kazetta behelyezése után próbáljuk meg lenyomni a felvétel gombját. Az írásvédett kazetta esetén a gomb nem működik, ellenáll a lenyomásnak. Írható kazetta behelyezése után viszont könnyedén lenyomhatjuk. Ez az írásvédelemmel ellátott magnó normális működése.

A magnóknál gondolnunk kell arra is, hogy a használatuk során -- bármilyen gondosan járunk is el -- megváltoznak a tulajdonságaik. Ez leggyakrabban a fejek és a továbbítógörgők szennyeződéséből adódik. A füllel is érzékelhető hangváltozás, és az ezzel járó olvasási és írási bizonytalanság megelőzésére a magnókat rendszeresen tisztítanunk kell. Ez a vele kapott használati utasítás alapján nem nagy feladat, csupán arra kell vigyáznunk, hogy a durva beavatkozásokat elkerüljük. Gondoljunk arra, hogy minden magnó finommechanikai szerkezet. A karbantartást megkönnyítik az erre a célra gyártott és időnként kapható tisztítószalagok és fejbeállító kazetták. Különösen a por okoz sok bajt, ezért a magnót és a kazettákat elsősorban ettől óvjuk. Ha lehet, a gyakrabban használt kazetták legyenek jó minőségűek, és a hibás kazettákat a lehető leghamarabb cseréljük ki hibátlanra.

A programokat és az adatokat biztonságosan akkor őrizhetjük



még, ha azokat egy másik kazettára is felírjuk. Ehhez két magnóra, vagy egy gyorsmásoló magnóra van szükség. A másolandó programokat vagy adatokat ebben az esetben egy kényes hangfelvételnél kell tekintenünk. A másolást a hangfelvételek másolásának szokásos módján is elvégezhetjük.

Kipróbált másolási módszereink a következők:

1. Eredeti magnónk 5-pólusú DIN csatlakozóját és az RM 4640 5-pólusú DIN csatlakozóját kötöttük össze.
2. Eredeti magnónkat egy jó minőségű Hi-Fi torony bemenetével kötöttük össze.

Két eredeti magnó MIC és EAR csatlakozóinak összekötése veszélyes, szerencsétlen esetben még a magnókat is elronthatja! A legkönnyebben és a legjobb minőségben a 2. módszerrel sikerült másolatot előállítanunk, de az 1. esetben is lehetett használható másolatot készíteni, viszont a felvétel túlságosan zajos volt. A tapasztalatok alapján megállapítható, hogy a kazetták másolásának ezek is járható útjai.

## Mágnesszalagos kazetta

A személyi számítógépek elterjedt programtárolója a zenei és egyéb hangfelvételek készítéséhez régóta használt mágnesszalagos kazetta, ill. annak néhány célszerűen átalakított változata. Legkönnyebben a hangfelvételek készítésére, megőrzésére szánt kazettákhoz -- ezeket megkülönböztetésül itt zenei kazettáknak fogjuk nevezni -- juthatunk hozzá. Két igen fontos, számunkra meghatározó tulajdonságuk van: a kazettákban lévő mágnesszalag hossza (amit a lejátszási idővel mérünk) és a kazetta minősége (amiről csupán a gyártó cég neve és a mágnesezhető felület jelzése tájékoztat).

A kazetta lejátszási ideje -- a nálunk leggyakrabban kapható zenei kazetták esetében -- 60 és 90 perc. A számítógépesek számára mind a kettő egy kicsit hosszú, ezért leginkább a 60 percest használjuk. A 60 perc a teljes műsoridőt jelenti, tehát a szabványos 4,75 cm/s szalagtovábbítási sebesség mellett a kazetta egyik oldalának lejátszása 30 percig tart (azaz kb. 8550 cm hosszú a kazettában lévő szalag). Még a gyors szalagtovábbí-

tással is elég sok időt igényel tehát a szalag másik végén lévő program megkeresése, hiszen csaknem az egész szalagot át kell tekercselnünk. Különösen a gyakran használt programjaink tárolására előnyösebb a rövidebb (például 2x10, 2x5 perces vagy még rövidebb) kazetta. Ezeket programkazettáknak fogjuk nevezni. Üres programkazettákat egyenlőre ritkábban lehet beszerezni, a vásárolt programok azonban szinte kizárólag ilyeneken vannak. Egy-két program tárolására, főleg ha ezek a programok elég nagyok, ezek a programkazetták kifogástalanul használhatók. Beolvasandó program keresése során legfeljebb a kazetta végére érünk, de ekkor egyszerűen úgy folytathatjuk a keresést, hogy megfordítjuk a kazettát. A gyári kazetták ugyanis úgy készülnek, hogy mind a két oldalukra ugyanazt írják fel, tehát egy fordítással az összes, ezen a kazettán lévő programot megtaláljuk. Vannak olyan kazetták is, amelyekkel kétszeres programfelvétel és kazettaforgatás nélkül elérhetjük ugyanezt, azaz egy programnév keresése során a kazettán -- pontosabban a kazetta egyik oldalán -- lévő összes programot megvizsgálhatjuk mindaddig, amíg a keresettet meg nem találjuk. Ezek a végtelenített kazetták. (Sajnálatos módon az ilyen kazetta beszerzése nálunk elég nehéz.)

A kazetta minőségét kétféleképpen jellemezhetjük: a szalag mágneses és a szalagtovábbítás mechanikai tulajdonságaival. A mágneses tulajdonságokról a kazettákon van -- ugyan nem túl sok -- tájékoztató. Tapasztalatunk szerint jól használhatók a normál (NORMAL BIAS 120  $\mu$ s EQ vagy FE 120  $\mu$ s jelzésű) szalagok. Egyenlőre azonban sokkal többet mond számunkra a gyártó cég neve és szalagjelzése. Jól működnek a következő típusú kazetták:

AGFA SUPERFERRO HDX 60+6

BASF

HITACHI DL 60

MAXELL LN

SONY HL 60.

Ezek a kazetták nem tartoznak a legolcsóbbak közé. A vásárolt programokat gyakran gyengébb minőségű szalagokon kapjuk. A gyenge szalagminőséget a magnó sűrűbb tisztogatásával ellensú-

lyozhatjuk, de ha lehet, új szerzeményeinket másoljuk át egy már bevált minőségű kazettára. Ezt mindamellelt javasoljuk, hogy az egyes kazettatípusokon belül is előfordulhatnak gyengébbek.

A biztonságos adat- és programbevétel érdekében tehát -- a magnó karbantartásán kívül -- a kazetták kezelésére, tárolására is figyelniünk kell.

Már említettünk néhány mechanikai behatást, amitől a kazettákat óvniunk kell. Nem említettük még a mágnesezettséget befolyásoló hatásokat. Ezek alattomos hibákat, zavarokat okozhatnak, ugyanis a kazettákon ránézésre a hiba semmi jele sem látható, csupán a számítógép jelzi a bajt; például programeolvasás helyett hibajelzést kapunk. A mágnesezési hibákat a kazetta időnkénti újraírásával -- frissítésével -- és a külső mágneses hatások kizárásával kerülhetjük el. Mágneses hatások villamos motorok, erősáramú elektromos vezetékek, elektromágnesek stb. közelében érhetik a kazettákat. A magnó perifériaként való használatának bizonytalanságai nagyrészt a nem megfelelő kazettatárolásra és a magnókarbantartás elhanyagolására vezethetők vissza, ezért célszerű az apróságoknak tűnő rendszabályokat betartani.

### Botkormány

Az ENTERPRISE központi egységével egybeépített billentyűzet szerves része a botkormány. Ez a szerkezet éppúgy nem tűnik perifériának, mint maga a billentyűzet. Segítségével a képernyőn villogó kurzort mozgathatjuk függőleges és vízszintes irányban. (Ezért nevezzük kurzormozgató botkormánynak.)

Jobban használható, kényelmesebb botkormánnyal is kiegészíthetjük a számítógépet. Ehhez egy külön erre a célra kialakított csatlakozót is be kell szereznünk. Ezt a csatlakozót be kell helyezni a CONTROL 1 feliratú aljzatba, és erre kell csatlakoztatni a botkormány összekötővezetékét. Így marokra fogható, "tűzgombbal" is ellátott, különösen a játékprogramok irányítására alkalmas botkormányt kaphatunk.

Több játék (például a lesiklás vagy az autóverseny) a játékos-tól "kétkezes" vezetést igényel. Ehhez a botkormányt rögzíteni

kell, amit a gyárilag a botkormányra felszerelt tapadókorongok segíthetnek. Gondolnunk kell azonban arra, hogy ezek csak sima felületen (például lakkozott asztallapon vagy üveglapon) rögzítenek jól.

### Hajlékony mágneslemezes meghajtóegység

Az adatok és programok felírását és beolvasását a magnóhoz képest több nagyságrenddel gyorsíthatja meg a mágneslemezes háttértár. A személyi számítógépek mágneslemezes perifériái két nagy csoportra oszthatók:

1. A meghajtóval egybeépített, rögzített mágneslemezes egység.
2. A cserélhető, hajlékony mágneslemezes egység.

Az elsőt Winchesternek is szokták nevezni. (Nevét egy közismert gyorstüzelő fegyverről kapta). Ezzel itt részletesebben nem foglalkozunk, egy kicsit ismerkedünk azonban a másodikkal, amit floppy egységnek, ill. floppy drive-nak vagy diszknek is szoktak nevezni. A továbbiakban egyszerűen csak lemezek és lemezegységnek nevezzük. Egyenlőre az ENTERPRISE-hoz az még nem kapható közvetlenül csatlakoztatható lemezegység, tervezik azonban Commodore lemezegység működtetésére alkalmas illesztők árusítását. A terv megvalósításáig az egyetlen -- a későbbiekben pedig várhatóan az olcsóbb -- beszerzési lehetőség az egyedi fejlesztések megvásárlása, vagy a sajátkezü illesztés. A lemezegységek legszembeütőbbben a lemez méretében különböznek. Vannak 8 collos nagy, 5 1/4 collos közepes és 3 1/2 collos minilemezzel működő egységek. A lemezegységek egy része a lemezek csak az egyik oldalát mágnesezi. Ezeket egyoldalú lemezegységeknek nevezzük, megkülönböztetve a mindkét oldalt mágnesező, kétoldalú lemezegységektől. Ezekről a tulajdonságoktól azonban a működési elvük független. Minden lemezegység egy, a lemezt forgató és a mágneses jeleket író-olvasó fejet működtető részből áll.

A lemezt egy tengelyvég hozza forgásba. A tengelyvéget a lemezegység lezárásával szorítjuk a lemezre. A lemezről olvasást és a lemezre írást a számítógép indítójellel kezdi. Ennek a jelnek a hatására kezd forogni a tengely, ami a súrlódási erő közvetítésével forgatja a lemezt.

A fejműködtető rész a fejet egy egyenesszakasz mentén mozgatja, és a lemezhez közelíti. A jól beállított fej nem ér hozzá a lemezhez, de nagyon megközelíti. A forgó lemez és a fej között légpárna alakul ki, ami védi a lemezt a sérüléstől. Ez természetesen a jól beállított és nem túlságosan elszennyeződött fej esetén jelent védelmet. A hibás meghajtók a lemezt megkarcolják, amivel örökre használhatatlanná tehetik. Különösen az egyoldalas lemezmeghajtók szoktak ilyen hibát elkövetni, de szerencsére általában a lemeznek a fejjel átellenes, nem mágnesezett oldalát karcolják. Az így kialakuló körpályák azonban akkor okoznak bajt, ha a lemezt megfordítva annak mindkét oldalát használni akarjuk. Mivel a mindkét oldalra való írással a lemezterületet olcsón megkétszerezhetjük, ez gyakran használt fogás. Lehetnek azonban kellemetlen következményei is, ezért jobb, ha óvatosan élünk vele. A kétoldalas lemezmeghajtóknál a lemezek sérülése sokkal ritkább.

A lemezmeghajtók ma már megszokott szolgáltatása a lemez felülírása elleni védelem. Hatása ugyanaz, mint amiről a kazettáknál már beszéltünk. Ki- és bekapcsolása azonban nem mechanikus, hanem fényérzékelővel végezhető. Ha a lemezmeghajtó a behelyezett lemez meghatározott helyén "átlát", akkor a lemezre írhatunk, ellenkező esetben nem; de létezik fordított eset is. A működéséről azonban sokkal nehezebb meggyőződnünk, mint a magnók esetében. Kívülről semmi sem látszik, a lemezmeghajtó borítása még a vezetékeket is eltakarja, így marad a tényleges kipróbálás: veszünk két lemezt, az egyiket az adott helyen leragasztjuk, a másikat nem, és írásvédettnek azt a lemezt tekintjük, amelyikre nem tudunk írni. Ezt a kísérletet minden gyanús jel vagy komolyabb beavatkozás után célszerű elvégezni.

A személyi számítógépek perifériájaként ma használt mágneslemez egységekben programok segítségével (szoftverrel) tájékozódunk arról, hogy a lemeznek éppen melyik területességét olvasuk. Ehhez általában csak annyi hardvertámogatást használunk, amennyi az induláshoz kell: a lemez kezdőpontját megadó un. indexjel érzékeléséhez szükséges ablakot is ki kell alakítania. Természetesen léteznek olyan lemezmeghajtók is, amelyek az indexjel felhasználása nélkül kezelik a lemezt. Ezeket -- például

juk állítani, és ezen a helyen a fej előtt elforduló szektor tudjuk írni vagy olvasni. A mágnesszalag és a lemez közötti sebességkülönbség ebből adódik, hiszen például a magnó olvasófeje előtt a szalagon lévő és a keresett adatokat megelőző, de ahhoz nem tartozó adatoknak is sorban el kell haladniuk (azaz az adatelérés soros).

### Hajlékony mágneslemez

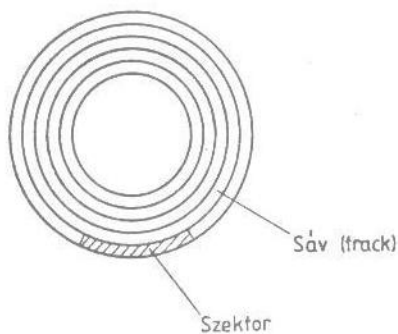
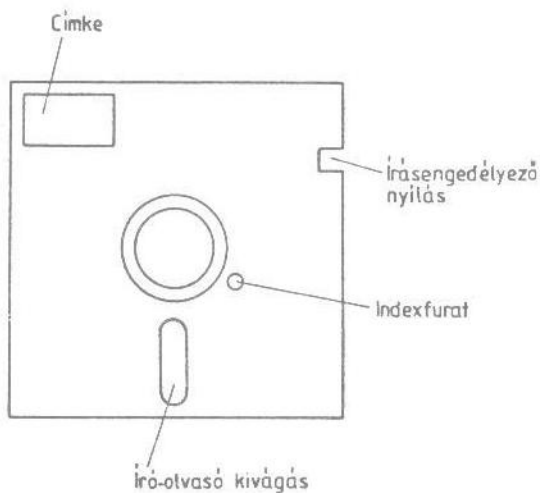
A személyi számítógépek kedvelt eszköze a hajlékony mágneslemez. Ha egy lemezt a kezünkbe veszünk, rendszerint csak a borítóját látjuk, amin a lemezmérettől függően meghatározott helyeken kivágásokat láthatunk (5. ábra).

Ha a kivágások elhelyezése szimmetrikus, akkor a lemezt már a gyártáskor az egyoldalas lemezmeghajtókban is mindkét oldalon használhatóra szánták. (Házilagos kétoldalasítás során ezt utánozzuk.)

A lemezterületet koncentrikus körpályákra osztják. A körpályákat sávoknak (track) nevezik. A körpályákat körívekre osztják, ezek a szektorok. A szektorok a legkisebb címezhető egységek. A meghajtóval egy címet közvetlenül el tudunk érni; ez azt jelenti, hogy a fejet a címben adott szektornak megfelelő sávba tudjuk állítani, és ezen a helyen a fej előtt elforduló szektort tudjuk írni vagy olvasni. A mágnesszalag és a lemez közötti sebességkülönbség ebből adódik, hiszen például a magnó olvasófeje előtt a szalagon lévő és a keresett adatokat megelőző, de ahhoz nem tartozó adatoknak is sorban el kell haladniuk (azaz az adatelérés soros).

### Nyomtató

Számítógéppel készített számolásaink, rajzaink és egyéb írásaink jó részét nem csak mi, és mi sem mindig csak a képernyőn akarjuk látni. Ezt általában nyomtatóval érhetjük el. A közeljövőben árusításra kerülő mátrixnyomtatót az ENTERPRISE PRINTER feliratú csatlakozóján keresztül, egyszerűen, minden további beavatkozás nélkül használhatjuk. A barkácsoló számítógépek elektronikus írógép illesztésével is próbálkozhatnak. (Például a soros vonalon keresztül a ROBOTRON 6011 írógép illesztése.)



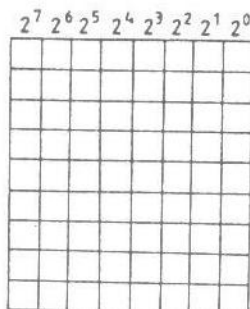
5. ábra. Mágneslemezek

Napjaink elterjedt mátrixnyomtatóinak (például EPSON) karaktereit egy függőlegesen elhelyezett túsorozat a képernyőhöz hasonlóan alakítja ki az ASCII kódnak megfelelően (6. ábra). A mátrix sorait egy-egy tű, oszlopait a fej vízszintes elmozdulása során a megadott helyzetben való működtetésével határozzuk meg. A működtetés azt jelenti, hogy a tű elektromágnes az adott pillanatban a tűt a papírra üti, és a köztük lévő festékszalag nyomot hagy a papíron.

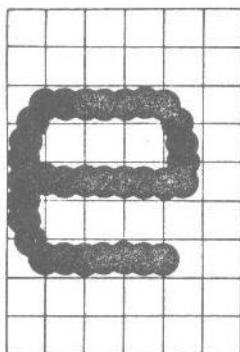
A mátrixnyomtatók írásképe nem elég szép, sőt olvasása időnként zavaró és fárasztó, ezért gyakran a jóval szebb írásképet adó,

$$\begin{array}{l} 2^7 = 128 \\ 2^6 = 64 \\ 2^5 = 32 \\ 2^4 = 16 \\ 2^3 = 8 \\ 2^2 = 4 \\ 2^1 = 2 \\ 2^0 = 1 \end{array}$$

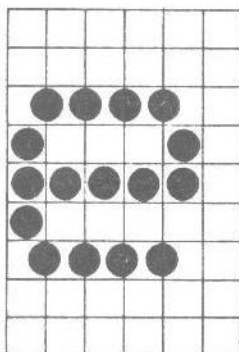
Grafikus tű



6. ábra. Matrixyomtató tűi



Javitott



Normál

7. ábra. Levélminőségű nyomtatás

de sokkal lassúbb elektronikus írógépet használjuk nyomtatóként. Az írógépeken kialakított betűtestek a kiíratáskor a levelezéshez megfelelő minőségű megjelenést biztosítanak. Igaz, szoftverrel is lehet szép írásképet készíteni, de ez komoly feladat, és a nyomtatót is túlzottan igénybe veszi (7. ábra).

### Külső programtár

Az ENTERPRISE programjai általában BASIC nyelven készülnek. Ennek szinte kizárólag az az oka, hogy a géphez kapott külső programtárral éppen BASIC értelmezőt kaptunk. Ha megnézzük a programtárunk belsejét, láthatjuk, hogy az egy olyan integrált



áramköri elemből áll, amelyikbe egy programot maszkoltak be. Természetesen többféle ilyen programtár készíthető, közöttük olyan is, amely más programnyelv fordítóját, értelmezőjét vagy egy meghatározott célra készített egyedi programot tartalmaz. Így egy külső programtár cseréjével más-más feladat megoldására használható gépet kaphatunk. (Igaz, hogy például a DEMO kazetta BASIC programjai nem fognak működni, ugyanúgy, ahogy a német nyelvű DEMO programok sem futtathatók az angol nyelvű gépen.) A külső programtár készítését lehetőleg kerüljük el!

### A perifériák felcserélhetősége (kapcsolata)

A számítógép központi egységéhez vezetékkel csatlakozó perifériák fontos jellemzője, hogy egy másik, ugyanolyan funkciójú perifériával helyettesíthetők. A magnetofon ismertetésénél már elmondtunk erre egy példát. Azokat a programokat például, amelyeket az eredeti magnetofonnal írtunk szalagra, a lényegesen eltérő tulajdonságú helyettesítő magnetofonnal -- némi kézi beavatkozás után -- be tudtuk olvasni, és a számítógép mindkét magnójával ugyanazt a programot ugyanúgy olvastuk.

Könnyen meggyőződhetünk a képernyők felcserélhetőségéről is. Nemcsak a különböző típusú fekete-fehér tv-eket cserélhetjük más fekete-fehér tv-re, hanem színes tv-készülékekkel, sőt a számítógépek céljára kifejlesztett sajátos képernyővel (monitorral) is kicserélhetők. A különböző típusok sajátosságai a felcserélés után természetesen észlelhetők, de a számítógép minden esetben működőképes, üzenetei olvashatóak, érthetőek anélkül, hogy programunkban bármit is változtatnánk.

Sokkal több kellemetlenséget okoznak a nyomtatók. Felcserélésük gyakran jár együtt a működtető programok cseréjével is. Ezt egyrészt a karakterkészletük különbözősége, másrészt az eltérő papírtovábbítás és papírméret okozza. Jóllehet a karakterkódoknak kialakult szabványaik vannak, mégis gyakran előfordul, hogy az egyik nyomtatón kiírt karakter a másikon értelmezhetetlen jelként hibát okoz, vagy a nyomtató betűközt ír a karakter helyett. A kellemetlenségeket fokozza, hogy a magyar nyelv ékezetes betűinek és hosszú magánhangzóinak jó része nem szabványka-

rakter, így nem egységes -- még ASCII, tehát ugyanazon szabványos kódrendszert használó nyomtatóknál sem (az eltéréseket a szakkönyvekben megtalálhatók). Könnyen érthető az eltérés, ha például egy mátrixnyomtatót cserélünk egy betűtárcsás elektromechanikus írógépre. Ezeket már eleve két különböző papírtovábbító szerkezettel látták el. A nyomtatókat általában az egybefüggő lapokból álló leporellóra, míg az írógépeket a különálló írólapokra való nyomtatásra tervezték. A nyomtatóktól ezek az írógépek abban is eltérnek, hogy a betűtárcsa vagy gömbfej egyszerű cseréjével ugyanazok a kódok egészen más karakterképhez kapcsolódhatnak, és az egyik vezérlőjele a másikon értelmezhetetlen kódokra változik. Mindebből számunkra az következik, hogy nyomtatót csak akkor cseréljünk, ha a működtető programot is cseréljük.

Itt kell beszélnünk arról, hogy a perifériák használata során melyek azok a körülmények, amelyek megváltoztatják a jellemzőiket:

A szennyeződés a legtöbb periféria megváltozását előidézi.

A melegedés következtében fellépő tulajdonságváltozásokkal is csaknem minden eszköznél találkozhatunk. Ezek oka rendszerint valamilyen géphiba, időnként azonban csak a hűtőrendszer átmeneti zavara (például a szellőzőnyílás eltakarása) vagy tartós túlterhelés.

A fej elállítódása (előfordulhat lemezmeghajtókban, magnetofonban) jellegzetes perifériamegváltozás, ami a melegedéssel együtt különleges bajt okozhat: a reggel felvett programot délután már nem tudjuk olvasni.

A perifériák beállítását meghatározó szabványok például a fejek beállítására bizonyos tűréshatárokat engednek meg. Ha a programkat az egyik gép tudja olvasni, a másik viszont nem, akkor ez a hiba a rossz fejbeállításból is eredhet. A fejek fokozatos elmozdulását korrigáló esetleges újraállítás után az említett olvasási gond egy gépen is jelentkezhet. Megelőzése elég nehéz, hiszen a bajt csak utólag észleljük. A fejek fokozatos beállítása és az újraírás azonban kisegíthet a bajból.

# Az ENTERPRISE programozása

---

A számítógépek igazi értékét nem a beszerzési árak, hanem a használhatóságuk adja. Ezt minden számítógépes szakember a programjainak készítésével maga is tovább növelheti. Ehhez természetesen meg kell tanulnia -- legalább a saját gépét -- programozni. Programozási ismeretek birtokában a gépet bizonyos korlátok között csaknem minden, a számunkra fontos feladat megoldásában felhasználhatjuk. A korlátokat a hardver jelenti. Azt mondhatnánk, hogy gépünk értékének két összetevője van, a "tehetsége" és a "tudása". A "tehetsége" a hardverben rejlik. Meghatározza, hogy milyen eszközöket milyen sebességgel lehet vele működtetni, mennyi adatot és mekkora programot lehet benne tárolni, az adatokkal milyen számtani (aritmetikai) és egyéb műveleteket lehet végrehajtani. Ennek döntő többségét a gyártók építik be a gépbe, így a gép megvásárlásával együtt ezekhez is hozzájutunk. Ha vállalkozunk otthoni barkácsolásra, akkor lehetőségeinket egy kissét kibővíthetjük, vagy számunkra előnyösebb tulajdonsággá alakíthatjuk valamelyik gyári lehetőséget. A gép "tehetségéből" számunkra csak az a jelentős, amit ki is tudunk használni, azaz amit "tudássá" tudunk változtatni. A tudássá változtatás a programozás, amelynek egy részét -- a gyári szoftver készítését -- a gép tervezői, előállítói elvégezték, egy másik jelentős részét mások -- a vásárolható programok előállításával -- már megcsinálták. Gépünk azonban akkor válik sajátunkká, ha magunk "tanítjuk meg" néhány nekünk tetsző dologra. Ehhez viszont meg kell ismerkedni az ENTERPRISE programozásával.

Az ENTERPRISE számítógéppel és programozásával ismerkedők számára igen fontos kérdés: hogyan lehet vagy kell a gépet programozni? Ezért most a programozási lehetőségekkel ismerkedünk. Kétféle programozásról, a BASIC nyelvűről és a gépi kódúról szólunk. A BASIC nyelvről azért, mert ezt kapjuk a géppel, a gépi kódról pedig azért, mert minden program, amit a géppel végre tudunk hajtatni, az ebben a nyelvben meglévő utasításokból áll; ráadásul ha megismerjük, segítségével apró, de ügyes programokat, eljárásokat tudunk magunknak írni.

### Gépi kód

Ebben a részben a gépi kódról azok számára szeretnénk képet adni, akik még nem próbálkoztak a vele való programozással. A számítógépek központi egysége számára a végrehajtandó utasításokat tömören, egyértelműen kell leírunk. A leírással szemben támasztott legfontosabb követelmény, hogy egyszerűen, gyorsan és gépiesen felismerhető legyen. Könnyen belátható, hogy erre a legalkalmasabb megoldás, ha az egyes utasításokat számokkal, például valamilyen sorrendbe állított műveletek sorszámával jelöljük.

Tekintsük bizonyításként a következő műveleteket:

1. a és b szám összegének elhelyezése a c számba.
2. a szám kétszeresének elhelyezése a c számba.
3. a és b szám szorzatának beírása a c számba.
4. c szám értékének bemásolása az a számba.
5. a szám értékének bemásolása a b számba.
6. a szám értékének beolvasása.
7. a szám értékének megjelenítése (kiírása).
8. a szám értékének beírása a tárolóba.
9. a tároló értékének beírása az a számba.

Láthatjuk, hogy egy-egy sorszámmal egy, sok betűből és egyéb jelből álló szöveget (karakter sorozatot) helyettesíthetünk, már ebben a nagyon leegyszerűsített esetben is. A számítógép mikro-

processzora azonban az általunk érthetőnek ítélt szöveget általában nem érti elég pontosan, és így további magyarázatot igényel. A Z80 A mikroprocesszor -- helyesebben a tervezői -- számára az "a szám" kifejezés még nem mond semmit, ugyanis nem adja meg például a szám hosszát, amely nélkül pedig a szám kezelhetetlen. Ráadásul ha valamilyen feladatot akarunk kiszámolni, akkor a számoknak értéket kell adnunk, valamint időnként részeredményeket kell megőrizni. Példaként tekintsük egy téglalap kerületének kiszámítását! (Előzőleg az utasításainkat már a példánkhoz megfelelően állítottuk össze.) A kitűzött feladat megoldásának lépései (algoritmusa):

1. Megadjuk a két szám értékét.
2. Kiszámítjuk a két szám összegét.
3. Kiszámítjuk az összeg kétszeresét.
4. Kiíratjuk a kerületet.

Ahhoz, hogy ezeket az apró lépéseket az utasításkészletünkkel leírassuk, tovább kell azokat fimoítanunk:

- 1.1. Az a szám értékének beolvasása (az utasítás száma 6).
- 1.2. Az a szám értékének beírása a b számba (5).
- 1.3. Az a szám értékének beolvasása (6).
- 2.1. Az a és a b szám összegének elhelyezése a c számban (1).
- 3.1. A c szám értékének beírása az a számba (4).
- 3.2. Az a szám kétszeresének beírása a c számba (2).
- 4.1. A c szám értékének beírása az a számba (4).
- 4.2. Az a szám megjelenítése (7).

Egyszerű jelölési rendszerünkkel a következőképpen írhatjuk le a feladat megoldását: 6 5 6 1 4 2 4 7. Ezzel előállt a feladat gépi kódú programja. Példaként említett feladatunk megoldása azonban így még nem eléggé tipikus! Gondoljunk arra, hogy a kerület meghatározásával általában együttjár a területszámítás is, ezért a mi megoldásunkat is ki kellene egészíteni ezzel. Egy megoldás: a kerület megjelenítése után a két oldal szorzatát is megjelenítjük, de ehhez a kerületszámításhoz összeállított lépéseinket (programunkat) módosítani kell.

A módosítás a 2. lépésnek a kiegészítését jelenti:

- 2.2. Az a szám értékének beírása a tárolóba (8).

A területszámítás:

5.1. A tároló értékének beírása az a számba (9).

5.2. Az a és a b szám szorzatának beírása a c számba (3).

6.1. A c szám értékének beírása az a számba (4).

6.2. Az a szám értékének megjelenítése (7).

A kerület és terület értékét is számító programunk tehát a következő:

6 5 6 1 8 4 2 4 7 9 3 4 7.

Példánkon megfigyelhetjük a gépi kódú programozás sajátosságait:

A gépi kód első szembetűnő kellemetlensége, hogy a programozónak olyan számsorozatot kell előállítania, amit könnyen össze tevészthet, elhibázhat. Gondoljuk csak meg, ha a számsorozatban felcseréljük a 8 és a 4 sorrendjét, azaz 8 4 helyett 4 8-at írunk, akkor az így leírt program mennyire mást csinál, mint amit terveztünk, és ezt milyen nehéz -- főleg a program futtatása nélkül -- észrevenni. Pedig ez az adatrögzítés, a gépirás egyik tipikus hibája! A programozók ezt a kellemetlenséget természetesen hamar felismerték. Mivel tudták használni a számítógépet, olyan leírási szabályokat alakítottak ki, amelyek betartásával a maguk számára sokkal jobban olvasható programokat készíthettek, és ezeket nem nagyon boholyult munkával, a számítógép egy programjával átalakíthatták -- lefordíthatták -- gépi kódra. A szabályokat assembly nyelvnek, az átalakító programot assemblernek nevezzük. Ma már természetes, hogy nemcsak egy assembly nyelv van, hanem szinte minden processzornak különböző, valamint ugyanannak a nyelvnek is számos assemblerével találkozhatunk. Az ENTERPRISE Z80 A mikroprocesszorának assembly nyelvével később részletesebben is foglalkozunk.

Példánk másik furcsasága, hogy nem a, b és c változókról beszéltünk, hanem a számról, b számról, c számról. Ezzel a furcsa elnevezéssel jelezni akartuk, most nem a matematikai tanulmányokból ismerős változókról, hanem valami másról, a regiszterekről van szó. Ezek a műveletek elvégzését segítő rendkívül gyorsan elérhető, a mikroprocesszorban lévő táruk. (Példánkban a c szám csak a műveletek eredményét tárolta, kiírni pedig csak az a szám értékét lehetett.) Megkülönböztetésükre neveket (a,

d, c stb.) használunk, ill. maga az utasítás meghatározza a regiszterre vonatkozott.)

Azokhoz az utasításokhoz, amelyekben nem határozták meg, hogy mely regiszterekkel fog dolgozni, a programozónak ezt az utasításkód után meg kell adnia. Ezeket a kiegészítéseket "operandusoknak" nevezzük. A Z80 A gépi kódjában csak egy vagy két operandus lehet. Egyes utasításokban az operandusok nemcsak regiszterek, hanem azok tartalmával egyenértékű, közvetlenül felhasználható számértékek is lehetnek. Az operandusok miatt a gépi kódokban -- a példaként említett utasításrendszerrel eltérően -- egy szám nem mindig írja le az utasítást, így változó utasításhosszra kell felkészülnünk. Ez nem csak a gépi kódú programok írását nem könnyíti, de olvasásukat, megértésüket határozottan megnehezíti.

A regiszterek száma korlátozott -- a Z80 A-nál 16 --, amely már az egyszerűbb számításokhoz sem elegendő. A regiszterek ismételt felhasználása érdekében időnként ki kell üríteni azokat, tartalmukat "menteni" kell. (Példánkban a terület kiszámításához az a szám kellett újra.) Kimentés után a regiszter újra felhasználható. Korábbi tartalmuk az adott helyen a mentésből újra előállítható. A mentés általában a mérhető tárterületre való átmásolást, a visszaállítás pedig a mentéskor megadott című tárterület tartalmának a regiszterbe másolását jelenti.

A regiszterek és a tár címzésére felhasználható tárterület a mai személyi számítógépekben általában egy bájt, amely nyolc, kettős számrendszerbeli számjegy hosszúságú. (Ezért nevezzük ezeket a gépeket bájtszervezésűeknek.) Leírásukra a 16-os számrendszert használjuk. A műveleteket elvégző egység is ilyen regiszterekkel dolgozik, ami azt is jelenti, hogy az utasítások kódja is ekkora. Egy utasításkóddal tehát 256 különböző utasítást adhatunk meg. A Z80 A utasításainak száma ennél több, ezért két utasításkóddal leírt -- azaz kétszer két hexadecimális számmal megadott -- utasításai is vannak.

A programokat a számítógép a tárolójában a beírás sorrendjében elhelyezve hajtja végre. Példaprogramunkban az utasításokat a megadott sorrendben kellett végrehajtani. Ez csak kivételesen egyszerű esetekben van így. A programokban általában szükséges

olyan elágazás, ahol valamilyen feltételtől függően a következőként vagy a sorban következőt, vagy az utasításban megadott helyen lévő utasítást kell végrehajtani. Ezt vezérlésátadásnak is nevezik. Megvalósítására a műveleti egységekben feltételjelzőket és utasításslámlálót építettek be. Az utasításkód meghatározza az utasítás hosszát, így a következő utasításnak a tárbeli helyét -- az előzőt ismerve -- könnyű meghatározni. Ha az utasítás elágazást hoz létre, az új címet a művelet végrehajtásaként határozzuk meg, és beírjuk az utasításslámlálóba.

A feltételek kiértékelését jelzőbitek segítségével végezzük, amelyeket minden utasítás számára az előző utasításokkal állítunk be. Ha például két szám egyenlősége esetén a végrehajtás sorrendjét meg akarjuk változtatni, akkor először a két számot kivonjuk egymásból. A kivonáskor a különbség kiszámítása mellett beállítjuk -- többek között -- azt is, hogy az eredmény zérus-e vagy sem. Ezt rögzítjük egy jelzőben, amelynek értéke 1, ha az eredmény zérus, és 0, ha az eredmény nem zérus. A vezérlésátadó utasítások közül most azt kell használnunk, amellyel a jelző 1-es értéke esetén térünk át az utasításban megadott helyen lévő utasítás végrehajtására.

Az itt elmondottak szemléltetik, hogy milyen aprólékos munkát jelent a gépi kódú programozás, és közben milyen könnyű hibázni. Ennek ellenére nem tartjuk használhatatlannak, főleg olyan házi számítógépeknél, mint az ENTERPRISE, amelynek nincs hozzáférhető jó assembler és assembly nyelvű fejlesztőrendszere.

Azonban azt ajánljuk, hogy az első néhány programot senki se akarja gépi kódban írni! Sokkal gyorsabban, könnyebben juthat mindenki jól működő programhoz, ha azt BASIC nyelven írja. Kezdetben ennek a magasszintű programnyelvnek a lehetőségei is korlátlanak tűnnek. Ha már akadályoz a feladat megoldásában, akkor vegyük elő a gépi kódot! Addig a gépet és a BASIC nyelvű programozást is megismerjük. Ismeretek birtokában a gép előnyös tulajdonságait néhány, a gépi kódú programozást megkönnyítő BASIC program segítségével tudjuk kihasználni. (Az ENTERPRISE gépi kódú utasításkészletét a Z80 assembly nyelvének ismertetésével együtt fogjuk megadni.) Most foglalkozzunk egy kicsit a BASIC nyelvvel!



A számítógépek programozását megkönnyítették a magasszintű programozási nyelvek. Kezdetben a COBOL, FORTRAN, Algol nyelveket, majd a mai számítógépeken a Pascal, C, PL/1 nyelveket használják szívesen a programozók. A több nyelvhez is értők megtehetik, hogy a nyelvek közül mindig a feladatnak legjobb megfelelőt válasszák ki. 1963-ban a Dartmouth College munkatársai -- köztük a magyar származású Kemény professzor -- olyan nyelvet dolgoztak ki, amelyikkel a számítástechnikával ismerkedőket tovább segítették a programjaik elkészítésében. Céljukat a programnyelv elnevezésében -- Beginners, All-purpose Symbolic Instruction Code (kezdők általános célú szimbolikus utasításkódja) -- is megfogalmazták. Az angol elnevezés kezdőbetűiből álló BASIC betűszónak azonban önálló jelentése is van: alapvető.

A BASIC nyelv egyik legfontosabb jellemzője a párbeszéd (interaktív) jelleg. Tervezői feltételezték, hogy a programozó ott ül a számítógépnél és beírja a programját. A beírt (bebillentyűzött) utasításokat azonnal értelmezheti, és ellenőrizheti azokat a gép BASIC nyelvű utasításait gépi kódú utasításokká átalakító programjával. Az azonnali kiértékeléssel magyarázható, hogy ezek a programok a legtöbbször értelmezőprogramok. Ezekkel nem állítjuk elő a BASIC utasításoknak megfelelő gépi kódú utasítások sorozatát, hanem mindig végrehajtjuk a soron következő BASIC utasításnak megfelelő gépi kódú programrészletet. Az utasítások így azonnal végrehajthatóak, hatásukat a programozó azonnal láthatja. A számítógép használatának e módja a parancs üzemmód. A másik üzemmód a program üzemmód, amelyben az utasításokból a gép tárolójában programokat állítunk össze és külön paranccsal -- a RUN paranccsal -- indítjuk el azok végrehajtását, vagyis a program futtatását. Program üzemmódban az utasításokat a begépeléskor kialakított, számunkra is érthető, betűkkel leírt alakban tároljuk.

A BASIC nyelv utasításait úgy alakították ki, hogy azok az ismert és megszokott matematikai írásmódhoz lehetőleg hasonlóak legyenek. Használhatjuk a változókat, amelyek valamilyen egész vagy törtszámot jelölnek, és konkrét számokat -- számkonstanso-

kat -- is. A gépi kódú programpéldákban említett kerületszámítási feladatot BASIC nyelven  $k=2*(a+b)$ , a területet pedig  $t=a*b$  alakban is leírhatjuk. Ezeket, és az ezekhez hasonló, esetleg trigonometrikus és más, általánosan használt függvényt is tartalmazó képleteket, amelyekkel számértéket (most a kerület és a terület értékét) határozunk meg, aritmetikai kifejezéseknek nevezzük. A programokjainkban használunk még másféle kifejezéseket is: logikai kifejezéseket és karakterlánc (sztring) kifejezéseket is. A logikai kifejezés eredménye igaz vagy hamis logikai érték. Például, ha a megadott reláció teljesül, akkor a kifejezés értéke igaz, ha nem teljesül, akkor hamis. A logikai kifejezés egy logikai változóból, vagy logikai változók, esetleg relációk (kisebb, nagyobb, egyenlő stb.) között kijelölt logikai műveletekből állhat. A karakterlánc-kifejezések betűkkel, számjegyekkel és egyéb karakterekkel leírt szövegekből, ezeket jelölő változókból és a rájuk vonatkozó műveletekből állnak. A programjaink ilyen kifejezésekből és az utasítások végrehajtási sorrendjét előíró, valamint a gép és a külvilág kapcsolatát biztosító -- be- és kimeneti -- utasításokból állnak.

Az utasítások végrehajtását a BASIC nyelvben a vezérlésátadó és ciklusutasításokkal szabályozhatjuk.

Ciklusutasításokkal, pontosabban utasításpárokkal azokat az utasításcsoportokat adjuk meg, amelyeket egy meghatározott feltétel teljesüléséig kell végrehajtani.

Vezérlésátadó utasításokkal azokat a programrészleteket adjuk meg, amelyeket a megadott feltétel teljesülése esetén kell végrehajtani.

A matematikában megszokott jelölések használatát a BASIC nyelv a beépített függvényekkel és a vektorok, mátrixok leírását lehetővé tevő tömbökkel segíti. Ez azonban már a programozótól is igényel egy kis többletmunkát. A programban használni kívánt tömböket például előre értelmezni kell, meg kell adni az indexek számát és a legkisebb, legnagyobb indexszámokat. Amíg meg nem szokjuk, hibajelzés fog emlékeztetni a hiányokra. Eleinte okoz rendszerint nagyobb gondot a beépített függvények használata, ugyanis meg kell szoknunk használatuk szabályait. Például

a  $45^\circ$  fokos szöghöz tartozó tangens szögfüggvény értékét írhatjuk  $\text{TAN}(45)$  és  $\text{TAN}(\text{PI}/4)$  formában is, számunkra mindkettő ugyanazt jelenti; látjuk, hogy az elsőben a szöveget fokban, a másodikban radiánban adtuk meg. A BASIC beépített függvénye azonban csak egyféle szögértékkel számol, így a két forma két különböző függvényértéket jelent. (A BASIC beépített függvényeinek és egyéb utasításainak jelentésével, használatuk szabályai val az utasítások ismertetésekor foglalkozunk.)

A BASIC nyelv párbeszédese jellege a be- és kiviteli (I/O: Input/Output) utasításaiban mutatkozik meg. Az adatok bevitelekor például feltételezi a képernyő előtt ülő és a program által kért adatokat bebillentyűző programozót, ill. felhasználót. A "menet közben" beírt adatokkal igen jól lehet irányítani a programot (ezt használják ki az olyan "menüt" kínáló programok, mint pl. a DEMO program második része), de nagyon kellemetlen lehet, ha a program ritkán kér adatot, és az egyszerre megadható adatokkal sokáig számol. Természetesen az ilyen hosszú várakozások is elkerülhetők, ugyanis a I/O utasítások a program végén a futtatás előtt felsorolt adatokat is használni tudják.

Programjainkban felváltva használhatunk mindkét típusú I/O utasítást.

### Programok készítése

A személyi számítógépesek általában saját maguknak írják a programjaikat. Ez a programozás egyik kivételes esete, hiszen a programozó egyben a felhasználó, aki egyedül rendelkezik a teljes géppel. A programoknak nem kell mások számára olvashatóknak, javíthatóknak lenniük, nem kell a csoportmunkához nélkülözhetetlen szabványokat betartani, és a dokumentációkat sem kell elkészíteni. A számítógép adottságai -- különösen az első időben -- nem jelentenek a programok számára korlátozást, így nem kell a legjobb (például a legrövidebb, a leggyorsabb stb.) programot megtalálni. Programjainknak egyetlen követelményt kell kielégíteniük: a kitűzött feladatot kell elvégezniük! A megoldásban sokat segít, hogy az esetleg adódó programhibákat magunk vesszük észre, és mivel értünk hozzá, azonnal javítani is tudjuk, sőt, a menet közben szerzett tapasztalataink alapján

akár át is alakíthatjuk vagy kiegészíthetjük azokat. A könnyítések ellenére a feladatmegoldásokat célszerű jól átgondolva megterveznünk, a programokat pedig emlékeztető megjegyzésekkel bőven ellátnunk.

A feladatok megoldását kezdjük először a feladat pontosabb megfogalmazásával! A feladatot mi magunk találtuk ki magunknak, tehát kicsit furcsa, hogy mégis pontosítani akarjuk. Tapasztalatok alapján állítható, hogy a fejben kialakított elképzelések általában nemcsak túlságosan elnagyoltak, hanem még "menet közben változnak" is. Már a leírás is sokat segíthet, még akkor is, ha az szinte csak a leendő program nevének a megfogalmazása. Nagyon sok esetben a megoldás lépéseinek meghatározása véglegesíti és egyben pontosítja is a feladatot. A megoldás lépései -- az algoritmus -- vagy azok egy része adja a programot, tehát célszerű alaposan szemügyre venni. A szemrevételezést a különböző ábrázolási módszerek segítik. A következőkben az algoritmusok folyamatábráival és azok elkészítésével foglalkozunk.

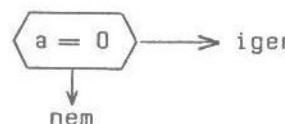
A folyamatábrán a megoldás lépéseit és azok sorrendjét ábrázoljuk. A sorrendet vonalakkal befejezett nyilakkal jelöljük, a tevékenységeket három csoportra osztjuk, és az e csoportokra utaló keretekbe rajzoljuk.

1. csoport: start, stop, az algoritmus végrehajtásának megkezdése és befejezése:

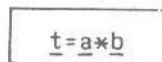
start

stop

2. csoport: döntés, az algoritmus következő lépését egy feltétel kiértékelése után két lehetséges lépésből választjuk ki:



3. csoport: végrehajtandó tevékenységek, amelyek után feltétel nélkül a kijelölt tevékenység következik:



A folyamatábrák megrajzolásuk a feladatmegoldás első, még nem programozható változatát is, amelynek egyes részleteit -- az egész megoldás átgondolása után -- pontosíthatjuk. A nagy feladatokat így kisebbekre bontva oldhatjuk meg, sőt a korábban már elkészített és kipróbált programjaink között is találhatunk olyant, amelyik valamelyik jelenlegi részfeladatunkat már megoldja, vagy "majdnem" megoldja. Az utóbbiakat is kitűnően tudjuk használni, ha elég körültekintően járunk el. A programok és leírásaik ugyanis általában nem elég részletesek -- nemcsak a mi saját magunknak készített programjaink, hanem időnként még a másoknak készített, szabályosan dokumentált programok sem --, ezért könnyen félreérthetőek. Az ismert sajátosságokra azonban már a folyamatábra összeállítására során számíthatunk.

A folyamatábrák jelölései -- mint a megállapodásoknál lenni szokott -- nem egységesek; vannak, akik több csoportot különböztetnek meg, más jelöléseket (leggyakrabban a döntések jelölésénél), ill. a többféle csoporthoz többféle jelölést használnak. Ennek ellenére azoknak a programoknak, amelyeket időt állónak vagy másokat is érintőeknek ítélünk, jó, ha megrajzoljuk a folyamatábráját is. Saját jelölésrendszerünk ugyanis magunk számára érthető marad, még akkor is, ha időnként "fejlesztjük". Másoknak viszont sokkal kisebb feladat a jelöléseink megfejtése, mint a programlistáink értelmezése. Mindezt magunk is tapasztalhatjuk, ha mások programját -- amely esetleg számunkra ismeretlen vagy gépünkön használhatatlan nyelven íródott -- akarjuk átvenni. Különösen a programozással hivatásszerűen foglalkozók ábrázolási módszerei térnek el az általunk említett folyamatábra-készítés módszerétől. Az ő céljuk, a munkájukkal szemben támasztott követelmények egészen mások, mint a gépével éppen ismerkedő, időnként még programot is író amatőröké. Ráadásul a lehetőségeik, a számítógépes fejlesztői rendszereik a mienktől lényegesen eltérő körülményeket biztosítanak a számukra. A programtervezési és -készítési módszereiknek része a programábrázolás is, amit mindig a módszerhez igazítanak. A sokféle ábrázolási technika közötti eligazodást megkönnyíti, hogy nagyon kevés -- a struktúrált programozási módszerek például 3-4 --, különböző alapelemből állítják össze a programok ábráit.

Most a programok minősége, a programozási módszerek helyett csak gépünk, az ENTERPRISE használatával foglalkozunk. Megelég- szünk a programozásnak a "folytonos átalakítás" módszerével, ami meglévő programjaink egyes részleteinek kimásolását, néhány utasítás módosítását, kihagyását vagy hozzáírását jelenti. Ez- zel ugyan a programok számos változatát állítjuk elő, így a tá- rolásuk, megőrzésük gondot okozhat, de kezdetben ez még nem el- viselhetetlen.

Ha majd a programjaink rengetegében folyton-folyvást eltévedve rendrakásra kényszerülünk, akkor gépünk és összegyűjtött prog- ramozói tapasztalataink felhasználásával olyan saját fejlesztő- rendszert barkácsolhatunk magunknak, amely programgyűjteményün- ket is kezelni fogja.

Ismerkedésünket a rendelkezésünkre álló BASIC nyelv utasításai- val folytatjuk.

### BASIC alapszavak

A BASIC szavak leírásánál a hagyományos módszert követtük: ábé- césorrendbe szedtük az utasításokat. Gondolkodtunk azon, hogy milyen megoldást válasszunk. Mivel a Felhasználói kézikönyv és külön kapható BASIC segédlet az utasításokat a funkciójuk sze- rint csoportosított formában közli, ezért úgy döntöttünk, hogy egy másfajta megoldást választunk. Az utasítások leírását úgy próbáltuk összeállítani, hogy minden alapinformációt közöljünk. Sajnos, a Felhasználói kézikönyv néhol több lehetőséget említ egy utasításnál, mint amennyit a valóságban használni lehet. Gyakorlatilag itt minden utasításról szólunk, ezért az átfedé- sek elkerülése végett -- a kézikönyv és ezen mű között -- hi- vatkozunk az irodalmakra. A lehetőségekhez képest próbáltuk a kézikönyv hiányosságait és pontatlanságait kijavítani.

### Felépítés:

Az első sorban közöljük az utasítás NEVÉT, hogy az milyen üzem- módban használható és milyen típusú.

A második sor az utasítás szintaktikáját írja le (ahogy a gép elfogadja tőlünk). A különböző nyelvű gépeken a csatornaszám

előle más-más jel kerül, a németnél a # (hashmark), az angolnál a \$ (fontjel).

A következőkben az utasítás hatását írjuk le, példaprogramokkal is segítve a megértést, majd a hibajelzésekről szólnunk néhány szót.

A BASIC végre tud hajtani bizonyos műveleteket is, ezek sorrendje a következő:

#### Aritmetikai műveletek:

1. 1 hatványozás.
2. \*, / szorzás, osztás.
3. +, - összeadás, kivonás.

#### Logikai műveletek:

1. <, =, > összehasonlító műveletek.
2. NOT logikai NEM.
3. AND logikai ÉS.
4. OR logikai VAGY.
5. BNOT bináris NEM.
6. BAND bináris ÉS.
7. BOR bináris VAGY.

Az ENTERPRISE BASIC interpreterje a különböző utasításokat súlyozza, és a kilistázáskor formázza. A példaprogramjainkat mi is így formáztuk.

#### Rövidítések:

aritm. kif. -- aritmetikai kifejezés.  
kif. -- kifejezés.

Most pedig lássuk az utasításokat:

ABS parancs üzemmód, program üzemmód numerikus függvény  
ABS(aritm. kif.)

Az ABS utasítás az aritmetikai kifejezés abszolút értékét számolja ki.

Példák:

- a) PRINT ABS(-188.7)
- b) 100 LET AB=-388  
110 PRINT ABS(AB)
- c) 200 LET X=48:LET X1=56:LET K=2  
300 IF ABS(X-X1)<K THEN GOTÓ 600

a) példa: az ABS közvetlen használatát mutatja be. Kiszámítja a zárójelben lévő konstans abszolút értékét.

b) példa: egy változó abszolút értékének kiszámítása.

c) példa: egy program részleteként való felhasználást mutatja be. Az 300-as sor függvényközelítéseknel (ha a különbség K-tól kisebb, akkor a 600-as sorra lép) használható fel.

Hibajelzés: Ha nem aritmetikai kifejezés van a zárójelben, vagy rossz a kifejezés, valamint ha a kifejezés értéke túl nagy (túlcsordulás).

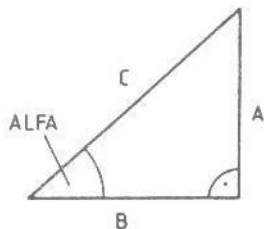
ACOS parancs üzemmód, program üzemmód trigonometrikus függvény

ACOS(aritm. kif.)

Azt a szögértéket számítja ki, amelynek koszinusza az aritmetikai kifejezés (arkusz koszinusz).

Példák:

- a) 100 OPTION ANGLE DEGREES  
110 LET X=ACOS(1/SQR(2))  
120 PRINT X
- b) 100 OPTION ANGLE DEGREES  
110 INPUT B,C  
120 LET ALFA=ACOS(B/C):LET BETA=90-ALFA  
130 PRINT "ALFA=" ALFA "BETA=" BETA





a) példa: kiszámítja a megadott értékhez tartozó szöget. Ügyeljünk azonban arra, hogy a gép alapértelmezésben radiánban számol, tehát ha az értéket fokban akarjuk megkapni, akkor a gépet DEGREES üzemmódba kell állítani.

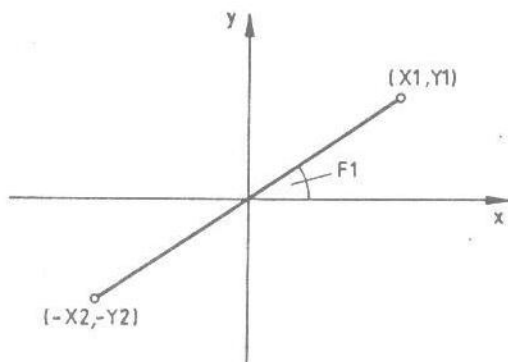
b) példa: kiszámítja a háromszög szögeit C és B értékéből. Az aritmetikai kifejezés értéke -1-től +1-ig terjedhet.

Hibajelzés: Ha a kifejezés értéke -1-nél kisebb vagy +1-nél nagyobb.

ANGLE parancs üzemmód, program üzemmód trigonometrikus függvény

ANGLE(aritm. kif., aritm. kif.)

Kiszámítja a (0,0) pontból induló, (aritm. kif., aritm. kif.) ponton átmenő egyenes és az x tengely közötti szöget az ábrán látható módon.



F1 = ANGLE(X,Y)

Alapértelmezésben radiánban kapjuk az értéket. Ha fokban akarjuk megkapni, akkor a szögszámítás típusát állítsuk át DEGRESS üzemmódba. Az értékek negatívak is lehetnek. A kapott szögérték maximum +-180 fok lehet.

Példák:

a) PRINT ANGLE(33,33) :REM 45 fok

```
b) 100 OPTION ANGLE DEGREES
    110 INPUT PROMPT "X=":X
    120 INPUT PROMPT "Y=":Y
    130 LET F1=ANGLE(X,Y):PRINT F1
```

a) példa: a direkt módon való használatot mutatja be.

b) példa: programba beépített utasításként használható, szögek kiszámítására.

Hibajelzés: Az aritmetikai kifejezés bármekkora lehet, de egy utasítássorban el kell férnie.

ALLOCATE parancs üzemmód, program üzemmód utasítás  
ALLOCATE(aritm. kif.)

Csak program elején használható, a programban megadja a gépi kódú rutin nagyságát, valamint az aritmetikai kifejezésnek megfelelő helyet hagy szabadon a gépi kódú program számára a memóriában.

Példa:

```
100 ALLOCATE(115)
```

Az aritmetikai kifejezésnek megfelelő számú bájtot foglal le a memóriából.

Hibajelzés: Ha túl nagy tárterületet foglalunk le, akkor az "ALLOCATE tár betelt" üzenetet kapjuk.

ASIN parancs üzemmód, program üzemmód trigonometrikus függvény

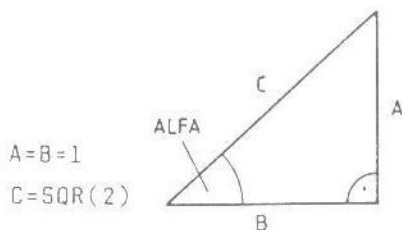
ASIN(aritm. kif.)

Azt a szögéréket számítja ki, amelynek szinusza az aritmetikai kifejezés (arkusz szinusz).

Példa:

```
100 OPTION ANGLE DEGREES
110 LET ALFA=ASIN(1/SQR(2))
120 PRINT ALFA
```

példaprogram kiszámítja egy derékszögű, egyenlő befogójú háromszög szögét fokokban, az ábra szerint:



Hibajelzés: Ha az aritmetikai kifejezés értéke  $-1$ -nél kisebb, vagy  $+1$ -nél nagyobb.

ASK parancs üzemmód, program üzemmód rendszerutasítás  
ASK utasítás változó

Az utasítással a különböző értékű gépi feltételeket lehet lekérdezni. Ha nem állítottuk be a feltételeket, akkor az alapértelmezés szerinti értékeket kapjuk válaszul.

Példák:

- a) ASK EDITOR BUFFER A  
PRINT A :REM 8
- b) ASK EDITOR VIDEO B  
PRINT B :REM 102
- c) ASK SERIAL BAUD S  
PRINT S :REM 15

A példák önmagukért beszélnek, az előre beállított értékeket kapjuk meg. A különböző értékek értelmezése a megfelelő utasításoknál megtalálhatók.

Hibajelzés: Ha a változónak nincs értéke.

ATN parancs üzemmód, program üzemmód trigonometrikus függvény

ATN(aritm. kif.)

Azt a szögértéket számítja ki, amelynek tangense az aritmetikai kifejezés (arkusz tangens).

Példák:

```
a) PRINT ATN(1)
b) 100 LET A=10:LET B=8
    110 OPTION ANGLE DEGREES
    120 LET ALFA=ATN(A/B):LET BETA=90-ALFA
    130 PRINT "ALFA=" ALFA "    ""BETA=" BETA
```

a) példa: a direkt módon való használatot mutatja be, ha átváltunk fokokra, akkor 45 fokot kapunk.

b) példa: kiszámítja egy derékszögű háromszög szögeit fokokban, amelynek befogói A és B.

Hibajelzés: Ha az aritmetikai kifejezésnek nincs értéke, vagy túllépi a megengedett határt.

ATTRIBUTES parancs üzemmód, program üzemmód grafikus utasítás

SET ATTRIBUTES aritm. kif.

Az utasítással a különleges, attributum típusú videólap működését lehet irányítani. A különböző aritmetikai kifejezések értékei megtalálhatók a Felhasználói kézikönyvben. Gyakorlatilag ez a típusú lap átmenet a grafikus és a szöveges lap között. Lehet rá írni és rajzolni is. Meghívni a GRAPHICS ATTRIBUTE utasítással lehet, a színszám megválasztására nincs lehetőség.

Példa:

```
100 GRAPHICS ATTRIBUTE
110 SET ATTRIBUTES 64
120 PRINT # vagy £ 101,AT 5,15:"IDE IRNI LEHET"
130 PLOT 100,100;500,300
140 PRINT # vagy £ 101,AT 18,18:"IDE RAJZ"
```

A példa önmagát mutatja be. A 64-es kód jelentése: írás a tintaszínek megváltoztatása nélkül.

Hibajelzés: Az aritmetikai kifejezés értelmezése közben sok hibalehetőség van (rossz összeadás, nem létező mód stb.). Használatakor figyelni kell, mert törölheti a képernyőt.

AUTO csak parancs üzemmódban használható utasítás.

AUTO

AUTO AT aritm. kif. STEP aritm. kif.

Bekapcsolja a program írásához az automatikus sorszámozást. Az F3-as funkcióbillentyű alapértelmezése: a kezdő sorszám 100, és 10-esével lépteti a sorszámot. Hatását megszüntetni vagy a STOP billentyű, vagy az ENTER újbóli leütésével lehet.

Példák:

a) AUTO

b) AUTO AT 500 STEP 5

c) AUTO STEP 5

a) példa: alapértelmezés, indítás 100-ról 10-esével.

b) példa: az első sorszám 500, a lépésköz 5.

c) példa: 100-ról 5-ösével indul.

Ha bekapcsoltuk az automatikus sorszámozást, akkor a programsor begépelése és az ENTER-rel való lezárása után kiíródik az újabb sorszám. Csak akkor célszerű használni, ha a program első változatát vagy újabb nagy programrészletet gépelünk be.

Hibajelzés: Ha az aritmetikai kifejezés túl nagy vagy negatív.

BEAM csak program üzemmódban használható grafikus utasítás  
SET # vagy & csatornaszám: BEAM ON vagy OFF

Csak grafikus üzemmódban használható, a rajzolósugarat lehet be-, ill. kikapcsolni vele.

Példa:

```
100 GRAPHICS HIRES 4
```

```
110 PLOT 10,10;400,100:SET BEAM ON:
```

```
    PLOT 300,500;1000,600
```

A példa a grafikus lapon egy kettős töréspontú görbét rajzol fel. Gépeljük be a SET BEAM ON utasításban ON helyett OFF-ot: ekkor a 400, 100 koordinátájú pontból a 300, 500 koordinátájú pontba nem rajzol, ugyanez történik akkor is, ha kihagyjuk ezt az utasítást.

Hibajelzés: Ha nem grafikus üzemmódban vagyunk, a gép kifrja, hogy a csatorna nem létezik.

BIAS parancs üzemmód, program üzemmód grafikus utasítás  
SET # vagy  $\xi$  csatornaszám: BIAS színkód

Amikor grafikus üzemmódban 16 színt lehet használni (GRAPHICS HIRES 16) és beállítottuk a palettánk első 8 színét, akkor a második nyolcat ezzel az utasítással állíthatjuk be. (Részletebben 1. a PALETTE utasításnál.) A csatornaszám grafikus üzemmódban elhagyható. Csak egymás utáni kódszámú színek 8-as csoportjai állíthatók be.

Példa:

```
SET BIAS 35
```

Az ötödik 8-as színcsoport lesz a paletta második 8 színe. A paletta 9. színe a 32-es kódszámú, és így tovább, a 15. szín a 39-es kódszámú lesz.

Hibajelzés: Ha nem grafikus üzemmódban vagyunk és nem nyitottuk meg a csatornát.

BIN parancs üzemmód, program üzemmód konverziós (átalakító) függvény  
BIN(aritm. kif.)

Az aritmetikai kifejezésben megadott bináris szám tizes számrendszerbeli értéke. A legnagyobb bináris szám 12 jegyű lehet.

Példák:

a) PRINT BIN(1010111) :REM 87

b) A Felhasználói kézikönyvben a karakterátgeneráló program.

a) példa: kiszámítja a zárójelben lévő bináris szám decimális értékét.

Hibajelzés: Ha az aritmetikai kifejezés nem bináris szám, vagy

meghaladja a gép által használható legnagyobb karakterszámot (12).

BLACK parancs üzemmód, program üzemmód szín-segédfüggvény  
SET utasítás BLACK

Önmagában nem használható függvény. A fekete színt állítja elő. Az alapszíneket ezen színtfüggvények megadásával is elő lehet állítani. (Azonos RGB(0,0,0).)

Példák: 1. a BORDER utasításnál és az RGB függvénynél.

Hibajelzés: Ha az utasítás első része jó, akkor hiba nélkül végrehajtódik.

BLUE parancs üzemmód, program üzemmód szín-segédfüggvény  
SET utasítás BLUE

Önmagában nem használható, a kék színt állítja elő (azonos RGB( 0,0,1)).

Példák: 1. a BORDER utasításnál és az RGB függvénynél.

Hibajelzés: Ha az utasítás első része jó, nincs.

BORDER parancs üzemmód, program üzemmód grafikus utasítás  
SET # vagy £ csatornaszám:BORDER függvény vagy kódszám

A képernyőkeret színét lehet beállítani vele. Parancs üzemmódban a csatorna megadása nem szükséges.

Példák:

- a) SET BORDER BLUE  
SET BORDER RGB(0,0,1)
- b) SET BORDER 255

a) példa: a keret színét mindkét utasítás kékre állítja. A színállításkor használhatjuk a szín nevét (beépített függvény),

a kódszámát (0--255-ig) vagy az RGB(aritm. kif.,aritm. kif., aritm. kif.) színkeverő függvényt.

b) példa: a keretet fehér színűre állítja.

Hibajelzés: Ha nincs meg a szín neve, vagy az aritmetikai kifejezés nem a 0-255 intervallumba esik, akkor hibajelzést kapunk.

CALL parancs üzemmód, program üzemmód programkezelő utasítás

CALL függvény

CALL függvény (kif.,kif.)

Az általunk a DEF utasításban definiált, vagy egy beépített függvény, ill. gépi kódú rutin hívásának egyik lehetséges eszköze. A programunkban bárhol elhelyezett függvényt a számítógép a CALL utasítással meghívja, végrehajtja a függvény utasításait, majd visszatér a CALL utasítás utáni sorra és folytatja a program futtatását. Ha a programban függvényt definiálunk, a végrehajtás során a számítógép átugorja azt, és a következő utasítással folytatja.

Példa:

```
100 DEF HAROMSZOGTER(REF A,REF B,REF C,REF T)
110 LET K=(A+B+C)/2
120 LET D=K*(K-A)*(K-B)*(K-C)
130 LET T=SQR(D)
140 END DEF
150 CLEAR SCREEN
160 PRINT TAB(10) "TERÜLETSZAMITAS"
170 PRINT " Oldalak hosszai ?"
180 INPUT PROMPT " A =":A
190 INPUT PROMPT " B =":B
200 INPUT PROMPT " C =":C
210 LET T=0
220 CALL HAROMSZOGTER(A,B,C,T)
230 PRINT " TERÜLET=" T
240 INPUT PROMPT " Akarod folytatni ?":AS
250 IF AS="i" THEN GOTO 150
260 END
```



A példa az utasítás egyik használatát mutatja be, kiszámítja egy háromszög területét az oldalak hosszából. További példákat a Felhasználói kézikönyv DEF utasításánál találhatunk.

Hibajelzés: A változók kiértékelése közben számos hibalehetőség adódhat. Leggyakoribb hiba, hogy a változóknak elfelejtünk értéket adni.

CAPTURE parancs üzemmód, program üzemmód adatkezelő utasítás

CAPTURE FROM # vagy £ csatornaszám TO vagy csatornaszám

Az utasítás az első csatornára érkező jellel helyettesíti a második csatornáról érkező jelet. A második csatorna jelét akkor kezdi el olvasni, ha a STOP billentyűt leütjük, vagy fájl vége jel érkezik, vagy hiba keletkezik. Hatástalanítani úgy lehet, hogy a második csatornaszámmra 255-öt írunk.

Hibajelzés: Ha a csatornák nincsenek megnyitva.

CASE csak program üzemmódban használható feltételutasítás

Önállóan nem szerepelhet, csak SELECT CASE - END SELECT blokkban. A CASE sorokban tudunk egy változó értékei szerint többféle elágaztatni. Részletesebben 1. a SELECT utasításnál.

Hibajelzés: Ha önálló utasításként gépeljük be, vagy a programban belépünk a CASE sorba, akkor kapunk hibát.

CAUSE EXCEPTION parancs üzemmód, program üzemmód hibageneráló utasítás

CAUSE EXCEPTION aritm. kif.

Az utasítással a programban hibát hozhatunk létre. Attól függően kapunk hibajelzést, hogy az aritmetikai kifejezésben milyen szám áll. Program közben használhatjuk, ha megszakítást akarunk létrehozni, vagy ha értékadó utasításnál egyből észre akarjuk venni a rossz értékek begépelését. Ha az aritmetikai

kifejezés 1--999-ig terjed, akkor az adott sorszámú hibajelzést kapjuk.

Példák:

```
100 PRINT "Nem jó"  
110 CAUSE EXCEPTION 9511  
120 REM A folytatás
```

Ebben az esetben azt az üzenetet kapjuk, hogy "Hibás EXOS funkciókód", és a program futása leáll.

Hibajelzés: Ha túl nagy számot gépelünk be, az 1000 kódszámú hibaüzenetet kapjuk.

CEIL parancs üzemmód, program üzemmód numerikus függvény  
CEIL(aritm. kif.)

Az aritmetikai kifejezésben megadott érték egész számra való felkerekítését végzi el.

Példák:

```
PRINT CEIL(8,0054) :REM 9-et ad  
PRINT CEIL(8,999) :REM szintén 9-et ad  
PRINT CEIL(-7,89) :REM -7-et ad
```

A példákban látható, hogy használatával vigyáznunk kell, mert meghamisíthatja számításunkat. Az aritmetikai kifejezés 12. karakterétől a számokat nem veszi figyelembe a gép. Tehát a

```
PRINT CEIL (1000000000.9) esetén 1000000000-t kapunk.
```

Hibajelzés: Ha az aritmetikai kifejezésnek nincs értéke, vagy ha túl nagy számot gépelünk be.

CHAIN parancs üzemmód, program üzemmód programkezelő utasítás

CHAIN program száma

A számítógép egyszerre több programot is tud kezelni. Az éppen futó programból meghívhatunk vele egy másik programot (a változólistával adatokat is adhatunk át a másik programnak, l. EDIT utasítás)

Példa:

```
100 CHAIN 3
```

A példa a 3. programot hívja meg.

Hibajelzés: Ha nincs a megadott sorszámú program.

CHARACTER parancs üzemmód, program üzemmód utasítás  
SET # vagy £ csatornaszám: CHARACTER X,A,B,C,D,E,F,G,H,I

A karakter formáját megváltoztató utasítás. Az X ASCII kódszámú karaktert lehet újradefiniálni vele. Az A az első sorát, a B a másodikat és így tovább, az I a kilencedik sorát adja meg felülről lefelé haladva. Az A...I értékét decimálisan kell megadni. Az összes videólapra vonatkozik, alapértelmezésben a normál karakteres csatornát használja. Ha csak a képernyőn akarjuk használni, a csatornaszám elmaradhat.

Példa:

```
SET CHARACTER 35,0,66,102,102,102,36,24,24,126
```

A példa egy kupát rajzol a német nyelvű gépen az # , ill. az angol nyelvűn a £ karakter helyére.

Hibajelzés: Ha nem ASCII kódú az X értéke , vagy az A...I-ben túllépjük a szintén 0--255-ös határt.

CHR\$ parancs üzemmód, program üzemmód sztring függvény  
CHR\$(aritm. kif.)

Egy tetszőleges  $0 < X <= 255$  intervallumba eső számból előállít egy sztringet, amelynek ASCII kódja az adott szám. (Ez az egyetlen lehetőség olyan karakterek meghívására, amely nem a tényle-

ges karakterkészletbe tartozó karakter (32--159), hanem rendszerutasítás.)

Felhasználható karakterek hívására, ha átgeneráljuk a karaktereket.

Példák:

a) PRINT CHR\$(33)

b) 100 FOR X=32 TO 159

110 PRINT ,CHR\$(X) "=" X;

120 NEXT X

a) példa: a felkiáltójelet írja ki.

b) példa: kiírja az összes karaktert az ASCII kódokkal együtt (32-től 126-ig a szabványos jeleket, utána pedig a bővített karakterkészletet). Érdekes kísérleteket végezhetünk, ha a számot változtatjuk.

Hibajelzés: Ha parancs üzemmódban nem 0--255, vagy programban nem 32--159 közötti számot írunk be, hibát kapunk. Használatára figyelni kell.

CLEAR parancs üzemmód, program üzemmód törlőutasítás

CLEAR # vagy £ csatornaszám

CLEAR feltétel

A megadott csatornát, feltételt vagy opciót törli.

CLEAR # vagy £ csatornaszám :REM a megadott csatorna pufferét törli.

CLEAR ENVELOPE :REM a megadott hangcsatorna pufferét törli.

CLEAR FKEYS :REM ha átalakítottuk a funkcióbillentyűk jelentését, az eredeti állapotot állítja vissza.

CLEAR FONT :REM az átírt karakterkészletet visszaállítja eredetire.

CLEAR GRAPHICS :REM törli a grafikus képernyőt.

CLEAR QUEUE :REM az x-edik hangforrás működését leállítja, és törli a SOUND utasítást.

CLEAR SCREEN :REM törli a képernyőt.

CLEAR SOUND :REM törli a zenei puffert.  
CLEAR TEXT :REM törli a karakteres képernyőt.

Hibajelzés: Ha szintaktikailag helyes, nincs. De a használatával vigyázni kell, mert törölheti a változókat.

CLOSE parancs üzemmód, program üzemmód csatornakezelő utasítás

CLOSE # vagy £ csatornaszám

Lezárja a megadott csatornát, felszabadítja a csatorna puffert.

Példa:

CLOSE # vagy £ 102

Bármilyen is volt írva a képernyőre, lezárja a 102-es szöveges csatornát, törli a képernyő puffert. A begépelt programok nem vesznek el.

Hibajelzés: Ha nincs megnyitva a megadott csatorna.

CODE parancs üzemmód, program üzemmód gépi kódú segédutasítás

CODE = sztring változó

CODE rutinnév = sztring változó

Az utasítással gépi kódú rutin nevét adhatjuk meg, és eltárolja a memóriában a gépi kódú rutint. Később a `USR` függvényel hívhatjuk meg, ha a rutinnak van bemenő adata és értéket ad vissza a programnak. Ha nincs, akkor a `CALL` utasítással is meghívhatjuk.

Példa: 1. Felhasználói kézikönyv.

Hibajelzés: Amikor meghívja a program, akkor adódhat a kiértékelésben hiba.

COLOUR parancs üzemmód, program üzemmód grafikus utasítás  
SET # vagy £ csatornaszám:COLOUR palettaszám, színkód  
SET COLOUR palettaszám

Grafikus üzemmódban a paletta első 8 színe közül valamelyiknek a színe állítható át az utasítással (0--7-ig), így nem kell az egész palettát átállítanunk, ha csak egy színt szeretnénk ideiglenesen megváltoztatni. A csatornaszám elhagyható, mert valamennyi videólapra kifejti hatását; alapértelmezésben a grafikus (101) csatornát használja.

Példa:

SET COLOUR 3,255

A példában a 3-as sorszámú palettaszínt állítja át a 255 kódszámú színre. A kódszám helyett a szín neve (beépített függvény) vagy az RGB függvény is használható. (Részletesebben l. a PALETTE használatánál).

Hibajelzés: Ha nincs megnyitva a grafikus lap és csatornaszám nélkül adjuk meg.

CONTINUE parancs üzemmód, program üzemmód segédutasítás  
CONTINUE

A SHIFT F1 funkcióbillentyű alapértelmezésben ez. Parancs üzemmódban a STOP billentyűvel megállított program futtatását folytatja onnan, ahol megállítottuk a programot. Programutasításként a handlerекből (megszakítás, "különleges eset") lehet vele visszatérni oda, ahol a program futása megszakadt. Ha a leállított programot újraszerkesztjük, vagy NEW, esetleg törlőutasítást adunk, a program futtatása nem folytatható.

Hibajelzés: A leírt eseteket kivéve nincs.

COPY parancs üzemmód, program üzemmód adatkezelő utasítás  
COPY FROM # vagy £ csatornaszám TO # vagy £ csatornaszám

Az egyik csatornáról adatokat másol át a másikra. A csatornák-  
nak nyitva kell lenniük. A műveletet a STOP vagy a fájl vége  
jel állítja le.

Példa:

```
100 OPEN # vagy £ 101:"VIDEO:"  
110 COPY FROM # vagy £ 0 TO # vagy £ 101
```

A 0 csatornán lévő adatokat átmásolja a grafikus csatornára.

Hibajelzés: Ha a csatorna nincs nyitva, vagy esetleg rosszul  
adjuk meg.

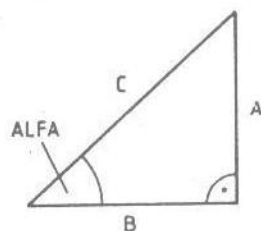
COS parancs üzemmód, program üzemmód trigonometrikus függ-  
vény

COS(aritm. kif.)

Az aritmetikai kifejezésben (alapértelmezésben radiánban) mega-  
dott szög koszinuszát számítja az ábra szerint.

Példák:

```
a) PRINT COS(1.5)  
b) 100 OPTION ANGLE DEGREES  
110 INPUT C, ALFA  
120 PRINT "B=" C*COS(ALFA)
```



a) példa: a radiánban megadott szögérték koszinuszát számítja  
ki (7.073720167E-02).

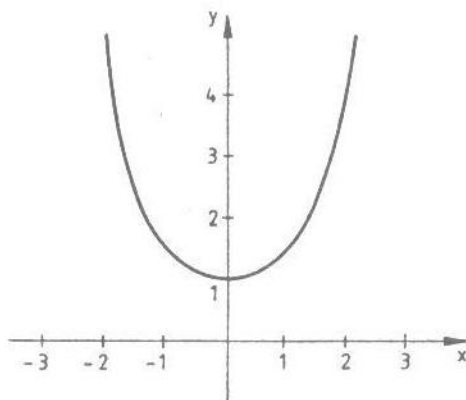
b) példa: kiszámítja a B oldal értékét az ALFA szög és a C ol-  
dal ismeretében. Értelmetlen eredményt kaphatunk, ha ALFA =90  
fok.

Hibajelzés: Ha az aritmetikai kifejezés nem kiértékelhető  
(nincs értéke).

COSH parancs üzemmód, program üzemmód trigonometrikus  
függvény

COSH(aritm. kif.)

Az utasítás kiszámítja az aritmetikai kifejezés koszinusz hiperbolikusát. A függvény értelmezése az ábrán látható.



$Y = \text{COSH}(X)$

Példák:

```
a) PRINT COSH(0)           :REM 1-et ad
b) 100 CLEAR SCREEN
    110 FOR X=-3 TO 3 STEP .25
    120 LET Y=COSH(X)
    130 PRINT " Y ( " X TAB(11)" ) ="Y
    140 NEXT X
```

a) példa: parancs üzemmódban való használatát mutatja be.

b) példa: a program kiszámítja a függvény értékeit -3-tól +3-ig 0.25-os lépésekben, és kiírja az Y értékeit a képernyőre.

Hibajelzés: Ha az aritmetikai kifejezés értéke túl nagy.

COT parancs üzemmód, program üzemmód trigonometrikus függvény

COT(aritm. kif.)

Az utasítás kiszámítja az aritmetikai kifejezés kotangensét. Értelmezése az ábrán látható.

$\text{COT}(\text{ALFA}) = B/A$



A két utasítással a kurzor alakját és színét változtathatjuk meg. Az első utasítás az ASCII kódszám megadásával az alakját határozza meg. A második utasítással a kurzor színét választ-hatjuk meg a palettáról megadott szín kódjával.

Példa:

```
SET CURSOR CHARACTER 35
```

Ezután a kurzor a német nyelvű gépeken # lesz, az angol nyelvűn a £ jelet kapjuk. Ha előtte átgeneráljuk azt a karaktert, amelyiknek a kódszámát használni szeretnénk, akkor bármilyen alak-zatot használhatunk kurzorként.

Hibajelzés: Ha nem esik a kódszám a 0--255 intervallumba.

CYAN parancs üzemmód, program üzemmód szín-segédfüggvény  
SET utasítás CYAN

Önmagában nem használható; a lila színt állítja elő (azonos RGB(0,1,1)).

Példa: 1. RGB függvénynél.

Hibajelzés: Ha az utasítás első része jó, akkor nincs.

DATA csak program üzemmódban használható adatkezelő utasí-tás

DATA konstanslista

Lehetővé teszi, hogy a programban, a READ utasítással olvasható számokat és sztring kifejezéseket tároljuk. A READ utasítás ab-ban a sorrendben olvassa az adatokat, ahogy azok a DATA utasí-tásban egymást követik. A konstansokat vesszővel kell elválasz-tani egymástól, ezek egyaránt lehetnek szöveg- és számkonstan-sok. Idézőjelbe akkor kell tenni, ha a konstans vesszőt is (,) tartalmaz.

Példa:

```
100 CLEAR SCREEN
110 FOR X=1 TO 6
120   READ X$
130   PRINT ,X$
140 NEXT X
150 DATA ENTERPRISE,ONE TWO EIGHT," ",
    Ez a példa," ",Tetszik ?
```

A példa szerint a gép egyenként beolvassa az adatokat, majd kiírja a képernyőre.

Hibajelzés: Ha a READ utasításban nincs hiba, a DATA-ban sem lehet, csak akkor, ha túl sok karaktert gépelünk be.

DATE parancs üzemmód, program üzemmód parancs  
DATE "ÉÉÉÉHHNN"

A gép által tárolható dátumot állítja be, ha megadjuk az időt is, éjfélkor automatikusan továbblépteti. A megadás formája: Év, Hó, Nap. A DATE-val lehet lekérdezni.

Példa:

```
DATE "19871130"
```

Beállítja 1987. november 30-ra a dátumot.

Hibajelzés: Ha nem 8 számjegy értékű a kifejezés.

DATE\$ parancs üzemmód, program üzemmód belső függvény  
DATE\$

A számítógép által tárolt aktuális dátumot lehet lekérdezni vele. A tárolt idő formája:

```
"ÉÉÉÉHHNN"
```

Példa:

```
PRINT DATE$ :REM 19871130
```

Ha előzőleg beállítottuk az aktuális időt, akkor a következő formában kapjuk: Év,Hó,Nap. Programok, rajzok, íráskopírozásnál, printelésnél hasznos, mert így megtudhatjuk, hogy mikor készült a program, rajz stb.

Hibajelzés: Nincs.

DEF csak program üzemmódban használható alprogramkezelő utasítás

DEF név/aritmetikai változó (változók)

DEF név/sztring változó (sztring változók)

Az utasítással egy- vagy többváltozós aritmetikai és sztring típusú függvényt vagy alprogramot definiálhatunk.

A függvény vagy alprogram definiálását az END DEF sorral kell befejezni. A programban, ha a számítógép DEF utasítást talál, akkor azt átugorja és az END DEF sor után folytatja a program végrehajtását. A CALL utasítással a függvény nevére hivatkozva lehet meghívni a definiált alprogramot. Ha alprogramból adatokat, értékeket akarunk átadni a főprogramnak, akkor a változók elé a REF szócskát is be kell írunk (l. még a Felhasználói kézikönyv, 72. oldalát).

Példák: 1. a CALL utasításnál és a Felhasználói kézikönyvben.

Hibajelzés: A DEF utasítás kiértékelése közben számos hibalehetőség van. Leggyakoribbak: nem adunk a változóknak értéket, kihagyjuk a REF utasítást, rossz változókat definiálunk.

DEFAULT CHANNEL parancs üzemmód, program üzemmód csatorna-kezelő utasítás

SET DEFAULT CHANNEL aritm. kif.

Az utasítással a rendszercsatorna számát lehet beállítani. Ezen a csatornán keresztül lehet a gép operációs rendszerével kommunikálni. Induláskor a rendszercsatornaszám 0. Ezen a csatornán keresztül programokat, utasításokat lehet beolvastatni, és a

gép üzeneteit vagy szövegeket kiíratatni. Választani a 0--99 csatornák között ajánlatos.

Példa:

```
SET DEFAULT CHANNEL 5
```

Ezzel az 5-ös csatornát nyitottuk meg rendszercsatornának.

Hibajelzés: Ha nem lépjük túl a 255 határt, nincs.

DEG parancs üzemmód, program üzemmód trigonometrikus függvény

DEG(aritm. kif.)

Az aritmetikai kifejezésben megadott radiánértéket átszámolja fokokká.

Példák:

a) PRINT DEG(PI/2) :REM 90

b) 100 CLEAR SCREEN

110 INPUT PROMPT "A oldal:":A

120 INPUT PROMPT "B oldal:":B

130 LET ALFA=ATN(A/B):LET BETA=PI/2-ALFA

140 PRINT :PRINT ,"ALFA=" DEG(ALFA)

150 PRINT ,"BETA=" DEG(BETA)

a) példa: önmagáért beszél, átszámítja a PI/2 szöget.

b) példa: egy derékszögű háromszög szögeit számítjuk ki fokokban, így a gép belső opcióját, a radiánban való számítást nem kell átállítani.

Hibajelzés: Ha túl nagy az érték, vagyis túllépi a gép által kiszámítható legnagyobb értékét. Forgásszögeket is kaphatunk eredményül.

DELETE csak parancs üzemmódban használható segédutasítás

DELETE programsorszám TO programsorszám

DELETE függvényblokknév

A megadott számú programsorokat törli. Az első programsorszám a már, a második a még törlendő sorok száma. A TO helyett - (kö-tőjel) is használható.

Példák:

- a) DELETE 100 TO 1000
- b) DELETE FIRST
- c) DELETE FIRST-500
- d) DELETE LAST
- e) DELETE 800-LAST
- f) DELETE GYOK

a) példa: 100-tól 1000-ig törli a programsorokat.

b) példa: az első sort törli.

c) példa: az első sortól 500-ig töröl.

d) példa: törli az utolsó sort.

e) példa: a 800-tól az utolsó sorig töröl.

f) példa: a GYOK-kel azonosított blokkot törli.

Ha csak önmagában a DELETE parancsot adjuk ki, akkor törli az egész programot.

Hibajelzés: Nincs. De óvatosan bánjunk vele, mert törölhet olyan programsorokat is, amire később szükségünk lenne!

DIM parancs üzemmód, program üzemmód segédutasítás  
DIM tömbváltozó (elemszám)

Az utasítás a megadott nevű, elemszámú és dimenziójú tömbnek foglal helyet a memóriában. A tömbbe számokat vagy karaktereket helyezhetünk el. Az alsó indexhatár 0 és maximum kétdimenziójú tömböket lehet definiálni. Ugyanazt a tömböt csak egyszer definiálhatjuk a programban.

Példák:

- a) DIM A%(10),GYOK(5 TO 25);NEV\$(50,25)
- b) 150 INPUT PROMPT " X értéke=":X
- 160 DIM FOM(X↑2+1,X)

a) példa: itt különböző típusú és elemszámú tömböket definiálunk.

b) példa: azt mutatja, hogy miként lehet az utasítást "dinamikusan" használni: a program végrehajtása közben definiálhatjuk a tömböt.

Hibajelzés: Indexhatárok kiértékelésénél, ha nincs elég hely a tömb elhelyezésére.

DISPLAY parancs üzemmód, program üzemmód képernyőkezelő utasítás

DISPLAY # vagy £ csatornaszám: AT konstans (a) FROM konstans (b) TO konstans (c)

A grafikus vagy szöveges lap megjelenítésére egy ablakot határoz meg a képernyőn. Az a konstans határozza meg, hogy az ablak hányadik sorban lesz. Ebben az ablakban a megadott lap a b-edik sorától a c-edikig lesz látható.

Példák:

a) DISPLAY # vagy £ 102: AT 10 FROM 5 TO 15

b) DISPLAY GRAPHICS

c) DISPLAY TEXT

a) példa: egy érdekes dolgot mutat be: hogyan nyissunk a szöveges lapon még egy szöveges lapot.

b) példa: a szöveges lapon egy 20 soros, a legutóbb használt grafikus lapot nyitja meg újra, de nem törli a szöveges lapot.

c) példa: ennek ellenkezője; a grafikus lapon nyit egy szöveges lapot (nem törli a grafikus képernyőt).

Hibajelzés: Ha a csatornát nem nyitottuk meg.

DO LOOP csak program üzemmódban használható ciklusszervező utasítás

DO { WHILE } UNTIL } logikai kifejezés

LOOP { WHILE } logikai kifejezés  
      { UNTIL }

Az utasítás a DO ciklus szervezésére alkalmas, lezárása a LOOP. Lehetővé teszi a DO és LOOP közötti rész (amelyet ciklustörzsnak hívunk) többszöri végrehajtását. A ciklusba kívülről nem szabad beugrani. A ciklusból feltétel nélküli kilépő utasítás az EXIT DO. A WHILE és az UNTIL feltételeket a ciklus elején a DO sornál vagy a ciklus végén a LOOP sornál egyaránt használhatjuk. A WHILE feltétel azt jelenti, hogy addig maradunk a ciklusban, amíg az igaz; az UNTIL pedig azt, hogy akkor lépünk ki a ciklusból, amikor a feltétel igazzá válik. (Részletesen l. Felhasználói kézikönyv.)

Példa:

```
100 CLEAR SCREEN
110 DO WHILE INKEY$=""
120 LOOP
130 PRINT AT 12,9:"Sikerült megállítani."
```

A példa a DO utasítás egyik használatát mutatja be, amíg le nem nyomunk egy billentyűt, a program végrehajtása áll.

Hibajelzés: Rosszul szervezett programnál, vagy ha kívülről lépünk be a ciklusba.

EDIT csak parancs üzemmódban használható segédutasítás  
EDIT programszám

Az EDIT utasítást követő programszámmal megadott programot jelöli ki az aktuális programmá. Ezután minden utasítás erre a programra vonatkozik.

Példa:

```
EDIT 1
```

Mivel a gép több programot tud tárolni, ezért így feldolgozható programmá tehetjük az 1-es számú programot.

Hibajelzés: Amíg van szabad tárterület az egyes programok részére, addig nincs.

EDITOR BUFFER parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET EDITOR BUFFER konstans

A képernyőszerkesztő méretét adja meg 256 bájtos lapokban. Lekérdezéséhez az ASK utasítás használható. A konstans maximális értéke 255 lehet.

Példák:

- a) ASK EDITOR BUFFER a  
PRINT a :REM 8-at ad
- b) SET EDITOR BUFFER 255

- a) példa: lekérdezi az alapértéket.
- b) példa: beállítja a maximálisan lefoglalható tárterületet.

Hibajelzés: Ha túllépjük a 255-ös határértéket.

EDITOR KEY parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET EDITOR KEY csatornaszám

Az utasítással beállított csatornán keresztül olvassa az EDITOR a billentyűzetet. Lekérdezni az ASK utasítással lehet.

Példák:

- a) ASK EDITOR KEY b  
PRINT b :REM 105-öt ad
- b) SET EDITOR KEY 255

- a) példa: az alapértelmezés szerinti 105-ös csatornát adja meg.
- b) példa: így a 255-ös csatornát használja.

Hibajelzés: Ha lekérdezésnél nem adjuk meg az azonosítót, vagy nem 0--255 közötti számot adunk meg.



EDITOR VIDEO parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET EDITOR VIDEO csatornaszám

Az utasítással kijelölt csatornán keresztül ír az EDITOR a képernyőre. Lekérdezni az ASK utasítással lehet. Alapértelmezésben a 102-es csatornát használja.

Példa:

```
SET EDITOR VIDEO 101
```

Az utasítás kiadása után az EDITOR a 101-es csatornán ír a képernyőre.

Hibajelzés: Ha nem 0--255 közötti számot írunk be.

ELSE IF...THEN...ELSE feltételes utasítás, 1.részletesebben ott.

END csak program üzemmódban használható utasítás.

END blokknév

Az utasítás végrehajtása a program futásának befejezését vagy a megadott blokk végét jelenti.

Példák:

```
END DEF
```

```
END SELECT
```

Ezen blokkok végét jelenti. A blokknév nem hagyható el. Ha az END csak önmagában áll, akkor a program logikai befejezését jelenti, és visszaadja a vezérlést a képernyőszerkesztőnek.

Hibajelzés: Nincs.

ENVELOPE \* parancs üzemmód, program üzemmód zenei utasítás  
ENVELOPE # vagy g csatornaszám :NUMBER aritm. kif.(1);aritm.  
kif.(2),aritm. kif.(3),aritm. kif.(4),aritm. kif.(5);....;  
RELEASE;.....;

Az utasítás a SOUND utasításban használható hangburkolót definiálja. Az (1) aritmetikai kifejezés az utasítás azonosítóját jelöli ki 0--254 között. A (2) azt határozza meg, hogy egy hangfázisban mennyit emelkedik vagy csökken a hang magassága az eredetihez képest, félhangokban mérve. A (3) és (4) kifejezés a bal és jobb oldali hangszóró hangerejét adja meg a SOUND utasításban megadott értékhez viszonyítva. Az (5) azt az időtartamot adja meg, amennyi ideig a hangfázis tartson. A RELEASE kifejezés utáni fázisok akkor is megszólalnak, ha a SOUND utasításban megadott időtartam lejárt, és a csatornán nem áll sorban másik hang. Részletesebben l. a Hanggenerálásnál.

EPS parancs üzemmód, program üzemmód numerikus függvény  
EPS(aritm. kif.)

Az a legkisebb értéket adja meg, amit hozzá kell adni vagy le kell vonni az aritmetika kifejezésből, hogy annak értéke megváltozzék.

Példák:

a) PRINT EPS(280) :REM 0.00000001  
b) 200 LET A=EPS(X)  
210 IF K<A THEN GOTO 300

a) példa: önmagáért beszél.

b) példa: függvényközelítésnél, ha az A nagyobb a K korrekciós tényezőnél, akkor nincs értelme a további függvénykorrigálásnak.

Hibajelzés: Ha az aritmetikai kifejezés túllépi a maximálisan ábrázolható karakterhosszúságot.

EXIT csak program üzemmódban használható ciklus-segédutasítás

EXIT { DO  
FOR  
DEF }

Feltétel nélküli kilépést jelent a megfelelő blokkból.

Hibajelzés: Ha az interpreter nem találja meg a megfelelő blokk végét (LOOP, NEXT, END DEF).

EXIT HANDLER csak program üzemmódban használható segédutasítás

EXIT HANDLER

A megszakításkezelő handlerekben megszakítást okoz. Ezzel újabb renkívüli esetet hozhatunk létre. A handlerblokkba kell beépíteni az utasítást.

EXLINE parancs üzemmód, program üzemmód belső függvény  
EXLINE

Az utolsó megszakítást okozó sor sorszámát adja meg.

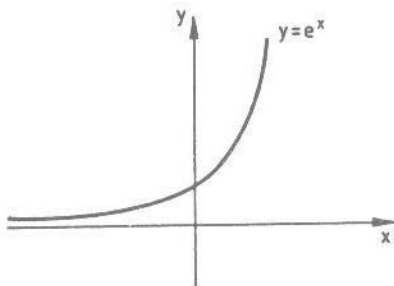
Példa:

PRINT EXLINE

Ha valamelyik programunk végrehajtása megszakadt, akkor lekérdezhetjük a sorszámát.

Hibajelzés: Nincs.

EXP parancs üzemmód, program üzemmód numerikus függvény  
EXP(X)



Exponenciális függvény. Az  $y = \text{EXP}(X) = e^X$  függvény értékét adja meg, ahol  $e$  a természetes alapú logaritmus alapja.

Példák:

```
a) PRINT EXP(1) :REM 2.7182281829
b) 100 CLEAR SCREEN
    110 FOR X=-5 TO 5 STEP .2
    120 LET Y=EXP(X):LET Y=ROUND(Y,2):PRINT TAB(3) "Y (" X;
        TAB(10) ")" TAB(12) "=" Y
    130 NEXT X
```

a) példa: kiszámítja az e értékét.

b) példa: két tizedesjegy pontossággal kiszámolja a függvény értékeit -5-től 5-ig, 0.2 léptetéssel.

EXSTRING\$ parancs üzemmód, program üzemmód hibakezelő függvény

EXSTRING\$(aritm. kif.)

Az aritmetikai kifejezésben megadott kódszámú megszakításra jellemző hibaüzenet karakteres szövegének kiíratására használható.

Példa:

```
100 FOR X=0 TO 40000
110 PRINT X; EXSTRING$(X)
120 NEXT X
```

A példaprogram kiíratja a gép összes, kódszámmal jelzett hibaüzenetét. Ha kétnyelvű gépünk van, először állítsk be a gép nyelvezetét.

Hibajelzés: Ha szintaktikailag helyes, nincs.

EXT parancs üzemmód, program üzemmód rendszerutasítás  
EXT "sztring kifejezés"

A megadott sztring kifejezést a külső programnak vagy az operációs rendszernek adja át. (L. még Felhasználói kézikönyv.)

Példák:

a) EXT "WP"

b) EXT "BASIC"

a) példa: a gép áttér szövegszerkesztő üzemmódba.

b) példa: hatása ugyanaz, mintha újraindítottuk volna a gépet.

Parancsként elhagyható az EXT utasítás, helyette írható:

:WP

:BASIC

Hibajelzés: Ha értelmezhetetlen a sztring kifejezés.

EXTYPE csak parancs üzemmódban használható függvény

EXTYPE

Az utasítással az utolsó megszakítás kódszámát lehet lekérdezni.

Példa:

Nyomjuk le a STOP billentyűt, majd gépeljük be:

```
PRINT EXTYPE
```

így a 9229-et adja a gép eredményül. Ez azt jelenti, hogy a "STOP billentyű lenyomva".

Hibajelzés: Nincs.

FAST SAVE program üzemmód, parancs üzemmód gépifeltétel-beállító utasítás

```
SET FAST SAVE ON/OFF
```

Magnetofonszalagra való programkimentés során a mentés sebességét lehet beállítani vele. A gyors kimentés (ON) sebessége 2400 baud, a lassú (OFF) kb. 1200 baud. Átkapcsolni a TOGGLE FAST SAVE utasítással lehet.

Hibajelzés: Ha szintaktikailag helyes, nincs.

FKEY program üzemmód, parancs üzemmód gépifeltétel-beállító utasítás

SET # vagy £ csatornaszám: FKEY billentyűszám "sztring kif."

Az utasítással a funkcióbillentyűk karaktersorozatait lehet beállítani. A billentyűszám 1...16. terjedhet, és bármilyen karaktersorozatot beállíthatunk vele. A csatornaszámot elhagyva a 105-ös csatornát használja. A funkcióbillentyűk eredeti jelentését a CLEAR FKEYS törlőutasítással állíthatjuk vissza. Ha a sor végén automatikus ENTER-t akarunk, akkor a CHR\$(13)-at is be kell gépelnünk.

Példák:

SET FKEY 2 "STOP"

SET FKEY 7 ":UK"

SET FKEY 15 ":BRD"

A példák egy-egy funkcióbillentyűt állítanak át a megadott sztring kifejezésre.

Hibajelzés: Ha szintaktikailag helyes, nincs.

FLUSH parancs üzemmód, program üzemmód utasítás

FLUSH # vagy £ csatornaszám

A megadott csatornára kiadja a pufferben lévő adatokat. A csatornát nem zárja le. Csak helyi hálózatra való kapcsolás esetén használható.

Hibajelzés: Ha csak parancsként írjuk be, és nem vagyunk hálózatra kötve, a 9034 kódszámú hibajelzést kapjuk. A parancs kiadása előtt a csatornát mindig meg kell nyitni.

FOR csak program üzemmódban használható ciklusutasítás

FOR ciklusváltozó = aritm. kif. (1) TO aritm. kif. (2) STEP aritm. kif. (3)

FOR... NEXT ciklusszervező utasítás. A változó értéke az 1.

aritmetikai kifejezéstől a 2. aritmetikai kifejezésig a 3. aritmetikai kifejezésben megadott lépésközökkel megy. A STEP és 3. aritmetikai kifejezés elhagyható, ekkor 1 lesz a lépésköz. A ciklusokat egymásba lehet ágyazni, de nem használhatják ugyanazt a ciklusváltozót.

A ciklusmagba kívülről beugrani nem szabad. A ciklusból feltétel nélkül az EXIT FOR utasítással lehet kilépni.

Példa:

```
100 FOR X=0 TO 100 STEP 5
110 PRINT TAB(10) "X=" X
120 NEXT X
```

A példa a ciklus legegyszerűbb használatát mutatja be: kiírja a képernyőre X értékét 0-tól 100-ig, 5-ösével.

Ciklusokat a következőképpen lehet egymásba ágyazni:

```
100 FOR X=A TO B
110 FOR Y=A TO B
120 FOR Z=A TO B
ciklusmag
180 NEXT Z
190 NEXT Y
200 NEXT X
```

Hibajelzés: Az aritmetikai kifejezések kiértékelése során sok hibalehetőség adódhat. Legjellemzőbbek: elmarad a NEXT sor, rossz ciklusváltozót adunk meg, az egymásba ágyazott ciklusok változóit felcseréljük.

FP parancs üzemmód, program üzemmód numerikus függvény  
FP(aritm. kif.)

Az utasítás az aritmetikai kifejezés tizedesjegyeit adja meg az egészérték nélkül.

Példák:

```
a) PRINT FP(8.324) :REM .324
b) PRINT FP(-8,324) :REM -0.324
```

Hibajelzés: Az aritmetikai kifejezés kiértékelése folyamán lehetőséges.

FREE parancs üzemmód, program üzemmód numerikus függvény

Kiszámítja az aktuális futtatott program részére szabadon felhasználható munkaterületet a tárban, bájtokban.

Példa:

```
PRINT FREE          : REM 44259 bájt
```

Ezt az értéket akkor kapjuk, ha bekapcsolás után a leírt utasítással lekérdezzük a 0-ás program részére biztosított szabad tárterületet.

Hibajelzés: Nincs.

GET csak program üzemmódban használható adatkezelő utasítás  
GET # vagy & csatornaszám: sztring változó

A megadott csatornáról egyetlen karaktert olvas be a sztring változóba. Ha nincs a csatornán adat, akkor üres karaktert ír ki. Ha a 105-ös csatornáról olvastatunk be, a csatornaszám elhagyható.

Példa:

```
100 CLEAR SCREEN  
110 GET A$  
120 IF A$="" THEN 110  
130 PRINT AT 19,12:A$
```

A példában a program futása addig áll, amíg le nem nyomunk egy billentyűt, majd a képernyő közepére kiírja ezt a karaktert. Hatása hasonló az INKEY\$-hoz.

GOSUB parancs üzemmód, program üzemmód alprogramkezelő utasítás

GOSUB programsorszám



Végrehajtja a megadott sorszámától kezdődő alprogramot. Feltétel nélküli vezérlésátadó utasítás. Az alprogramból a RETURN utasítással lehet a GOSUB utasítás utáni sorra visszatérni.

Példa:

```
500 GOSUB 1000
.
1000 REM alprogram
.
1500 RETURN
```

A példa szerint a gép végrehajtja az 1000-es sorral kezdődő alprogramot, és visszatér a GOSUB utasítás utáni sorra.

Hibajelzés: Ha a GOSUB utasításban megadott programrész nem létezik, vagy a 100 GOSUB 100 utasítást adjuk ki (önmagába nem térhet vissza), egy idő után "a tár betelt" hibajelzést kapjuk.

GOTO parancs üzemmód, program üzemmód ugróutasítás  
GOTO programsorszám

Feltétel nélküli vezérlésátadó utasítás. A program a megadott sorszámú sor végrehajtásával folytatódik.

Példa:

```
500 GOTO 2000
```

A 2000-es sortól folytatódik a program.

Hibajelzés: Ha nincs a megadott számú sor. A végtelenített ciklus nem ad hibaüzenetet.

(100 GOTO 100)

GRAPHICS parancs üzemmód, program üzemmód grafikus utasítás

```
GRAPHICS
GRAPHICS HIRES színszám
GRAPHICS LORES színszám
GRAPHICS ATTRIBUTE
```

Ezzel az utasítással léphetünk be a grafikus üzemmódba, és beállítja a képernyő kijelzési módját. Lezárja a normál képernyőcsatornát (a 101-est csak akkor, ha előzőleg is grafikus üzemmódban voltunk). Majd újra megnyitja a 101-es és a 102-es csatornákat úgy, hogy a képernyő első húsz sorát grafikus lapnak, az utolsó négyet szövegesnek nyitja meg. A grafikus lap pontjainak száma vízszintesen 1280, és függőlegesen 620.

A lehetséges módok: nagyfelbontású (HIRES 1280,620) és kislebontású (LORES 640,620) és attributum típusú grafikus lap (ATTRIBUTUM). A színszám 2, 4, 16 és 256 lehet. Az attributum típusú lapnál nem lehet színszámot beállítani.

#### Példák:

a) GRAPHICS :REM grafikus lapot nyit az első 20 sorban 4 szín-  
nel.

b) 100 GRAPHICS HIRES 256  
110 PLOT 0,0;1219,619

c) 100 GRAPHICS LORES 256  
110 PLOT 0,0;1219,619

a) példa: a parancs üzemmódban való használatot mutatja be. Alapértelmezésben négy szín van.

b) és c) példa: a grafikus képernyőre egy vonalat rajzol. Figyeljük meg a felbontásbeli különbséget!

Hibajelzés: Ha szintaktikailag helyes és csak a négyfajta színváltozatot használjuk, nincs. De használatával vigyázni kell mert törli a szöveges képernyőt!

GREEN parancs üzemmód, program üzemmód szín-segédfüggvény  
SET utasítás GREEN

Önmagában nem használható, a zöld színt állítja be (azonos RGB(0,1,0)).

Példa: 1. az RGB függvénynél.

Hibajelzés: Ha az utasítás többi része jó, nincs.

HANDLER csak program üzemmódban használható megszakításkezelő utasítás

HANDLER handlernév

hiba, megszakításkezelő rész

END HANDLER

A HANDLER blokk a program végrehajtása során általunk beírt, vagy a gép által okozott megszakítások és hibák kezelésére vonatkozik. Így tehát, ha előre felkészülünk hibára, akkor a futás nem áll meg, hanem a megfelelő hiba meglétekor meghívja ezt a hibakezelő rutint (1. Felhasználói kézikönyv).

HEX\$ parancs üzemmód, program üzemmód konverziós függvény

HEX\$ ("hexadec.szám, ..., hexadec.szám")

Az utasítás utáni sorban lévő sztringben felsorolt hexadecimális számokból képzett karaktersorozatot adja vissza. A megfelelő karakterek kódtáblázata a függelékben megtalálható. A hexadecimális 20--9F a szabvány ASCII kódoknak felel meg, 32--159-ig. Gépi kódú rutinokhoz szükséges a használata.

Példa:

```
PRINT HEX$("42,41,42,41") :REM BABA
```

Kiírja a BABA szót.

Hibajelzés: Ha a sztring kifejezésben értelmezhetetlen karaktereket adunk meg. Ügyeljünk nagyon a szintaktikájára!

IF parancs üzemmód, program üzemmód feltételutasítás

Többfajta IF struktúra létezik, ezek:

IF logikai kifejezés THEN utasítás vagy sorszám

IF logikai kifejezés THEN utasítás vagy sorszám

ELSE IF logikai kifejezés THEN

utasítás

ELSE sor :REM elmaradhat

utasítás

END IF

Feltételes vezérlésátadó utasításblokkok. Ha az IF szót követő kifejezés igaz, akkor a THEN sor teljesül. Ha nem igaz, akkor következő programsor, ill. az ELSE ág teljesül, az ELSE ágában még vizsgálhatunk feltételeket. Ha ilyen blokkot csinálunk, akkor END IF-fel kell lezárnunk. A blokkba nem szabad kívülről beugrani.

#### Példák:

```
a) 100 INPUT PROMPT "X=":X
    110 IF X>5 AND X<15 THEN 140
    120 PRINT "NEM IGAZ"
    130 END
    140 PRINT "IGAZ"

b) 100 CLEAR SCREEN
    110 INPUT PROMPT "X=":X
    120 INPUT PROMPT "Y=":Y
    130 IF X=Y THEN
    140   PRINT "X egyenlő Y"
    150 ELSE IF X>Y THEN
    160   PRINT "X nagyobb Y"
    170 ELSE
    180   PRINT "X kisebb Y"
    190 END IF
    200 PRINT
    210 GOTO 110
```

a) példa: az egyszerű használatot mutatja be. Ha X nem esik 5 és 15 közé, akkor kiírja, hogy "NEM IGAZ", ha közöttte van, akkor a logikai kifejezés "IGAZ".

b) példa: az ELSE ágak használatát mutatja be.

Hibajelzés: A blokk vagy a kifejezések kiértékelése sok hibát eredményezhet: pl. nem létező sorra mutat a THEN utáni sorszám, hibás blokkvégződés stb.

IMAGE csak program üzemmódban használható adatkezelő utasítás

IMAGE: formátumra vonatkozó adatok

A kiíratás formáját lehet meghatározni vele. A különböző formátumleírások a Felhasználói kézikönyvben megtalálhatók.

IN parancs üzemmód, program üzemmód adatkezelő függvény  
IN(n)

Az n bemeneti portról olvas be egy értéket. Az n értéke 0--255-ig terjedhet.

Példa:

```
PRINT IN(176)           :REM 248-at ad
```

Hibajelzés: Ha túllépjük a megengedett határokat.

INF parancs üzemmód, program üzemmód numerikus függvény

A gép által használható legnagyobb pozitív számot adja meg.

Példa:

```
PRINT INF              :REM 9.999999999E62
```

Ez az érték megfelel  $9.999999999 \times 10^{62}$ -nek.

Hibajelzés: Ha szintaktikailag jó, nincs.

INFO csak parancs üzemmódban használható utasítás

Kiírja a képernyőre a tárcapacitást, a szabad kapacitást, a gépben található programok számát, a lefoglalt bájtok mennyiségét és a tárolt programok első sorát. Az F8 billentyű alapértelmezése.

INK parancs üzemmód, program üzemmód grafikus utasítás  
SET # vagy ¢ csatornaszám: INK színszám

Az utasítással a színes, grafikus üzemmódoknál a tinta színét lehet beállítani. Ha 256 színt használunk, akkor a szín számmal, a nevével vagy az RGB függvénnyel adható meg a tinta szí-

ne. A többi színes mód esetében a paletta színszámával lehet megadni a választott színt. Alapértelmezésben a grafikus csatornára vonatkozik.

Példák:

```
SET # vagy £ 102:INK 2
```

A példa az ezentúl kiírandó betűknek a színét váltja át.

Hibajelzés: Ha elhagyjuk a csatornaszámot, és nem grafikus üzemmódot használunk.

INKEY\$ parancs üzemmód, program üzemmód sztring függvény  
INKEY\$="karakter"

Az utasítás a billentyűzetről olvas be egy karaktert. Ha nincs lenyomva billentyű, egy üres karaktert olvas be.

Példa:

```
200 DO UNTIL INKEY$="a"  
210 LOOP
```

A programban addig nem léphetünk ki a ciklusból, amíg le nem nyomjuk az a billentyűt.

Hibajelzés: Ha önálló utasításként használjuk.

INPUT csak program üzemmódban használható adatkezelő utasítás

INPUT # vagy £ csatornaszám, IFMISSING utasítás, AT sorszám, oszlopszám PROMPT sztring kifejezés: Változólista

A megadott csatornáról adatokat olvas be a változóba. Alapértelmezésben a 0 rendszercsatornáról olvas. Az IFMISSING utáni kifejezést akkor hajtja végre, ha a csatornáról hibás jel vagy fájl vége jelzés jön. Az AT sorban adhatjuk meg, hogy a ? jelet vagy sztringet hová írja ki a gép. A PROMPT után a ? jelet karaktersorozattal helyettesíthetjük. Ha mindezeket beírjuk, csak

Egy változót kérdezhetünk. A változók lehetnek egyszerű aritmetikai vagy sztring változók.

Példa:

```
100 INPUT A,B,A$,B$
110 PRINT A,B,A$,B$
```

A változókba beírt adatokat a gép kiírja. Az A és B változóba csak konstansokat, az A\$, B\$-be szövegeket is írhatunk.

Hibajelzés: A megadott adatok kiértékelése folyamán sok hibalehetőség adódhat.

INT parancs üzemmód, program üzemmód numerikus függvény  
INT(aritm. kif.)

Az utasítás az aritmetikai kifejezésnél nem nagyobb egész értéket ad meg.

Példák:

```
PRINT INT(8.789)      :REM 8-at ad
PRINT INT(-8.23)     :REM -9-et ad
```

Hibajelzés: Nincs, de használatával vigyázni kell, mert meghamisíthatja számításainkat.

INTERRUPT parancs üzemmód, program üzemmód gépifeltétel-beállító utasítás

Az utasítással többfajta megszakítást lehet beállítani és lekérdézni:

1. ASK INTERRUPT CODE aritm. változó  
PRINT aritm. változó  
Ha volt megszakítás, akkor az utolsó megszakítás kódszámát írja ki.
2. SET INTERRUPT KEY ON/OFF  
Így minden billentyűre megszakítást lehet beállítani.
3. SET INTERRUPT STOP ON/OFF

A STOP billentyűre való megszakítást lehet így ki-, ill. bekapcsolni.

#### 4. SET INTERRUPT NET ON/OFF

Így állítható be, hogy hálózatra kötéskor a hálózatról érkező adatok megszakítást okozzanak-e vagy sem.

IP parancs üzemmód, program üzemmód numerikus függvény  
IP(aritm. kif.)

Az aritmetikai kifejezésben megadott értéknek az egészrészét adja meg. Törli a tizedesjegyeket.

Példák:

```
PRINT IP(89.945)      :REM 89-et ad  
PRINT IP(-89.945)   :REM -89-et ad
```

A példák önmagukért beszélnek.

Hibajelzés: Ha szintaktikailag helyes, nincs.

JOY parancs üzemmód, program üzemmód belső függvény  
JOY(N)

Az N-edik botkormány helyzetét lehet vele lekérdezni. A 0 jelű a beépített, az 1 és a 2 a külső botkormányok.

Az értékek: 1--jobbra, 2--balra, 4--le, 8--fel, 16--tűzgomb. A beépítettnél a szóközbillentyű a 16-os.

Példa:

```
100 LET A=JOY(0)  
110 PRINT A
```

Ha a START gomb lenyomásának pillanatában mozdítjuk a botkormányt, megkapjuk a leírt értékeket.

Hibajelzés: Ha szintaktikailag helyes, nincs.



KEY CLICK parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET KEY CLICK ON/OFF

A gép működése közben a billentyűk lenyomását jelző hang ki-, ill. bekapcsolását lehet vele állítani. A TOGGLE utasítással lehet átkapcsolni a másik állapotba. Az F7 funkcióbillentyű alapértelmezése ez.

Hibajelzés: Nincs.

KEY DELAY parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET KEY DELAY aritm. kif.

Az aritmetikai kifejezésben megadott értékkel lehet beállítani azt az időt, ameddig egy billentyűt lenyomva kell tartani ahhoz, hogy a hatása ismétlődjön. A mértékegység alapja 1/50 másodperc. Lekérdezni az ASK-kal lehet. Alapérték: 30/50 másodperc.

Példák:

- a) SET KEY DELAY 50 :REM 1 másodperc kell az ismétlődéshez
- b) SET KEY DELAY 1 :REM 1/50 másodperc után ismétlődik

Hibajelzés: Ha túllépjük a megengedett határokat.

KEY RATE parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET KEY RATE aritm. kif.

Az automatikus billentyűismétlési sebességet lehet vele beállítani 1/50 másodperces egységekkel. Alapérték 3/50 másodperc. Az ASK utasítással lehet lekérdezni.

Hibajelzés: Ha nem lépjük túl a megengedett értékeket, nincs.

LBOUND parancs üzemmód, program üzemmód sztring függvény  
LBOUND(tömbváltozó)

A tömbváltozóval jelölt egydimenziós tömb alsó határát adja meg. Parancs üzemmódban csak akkor használható, ha előtte definiálunk egy tömböt.

Példa:

```
100 DIM A(2 TO 20)
110 PRINT LBOUND(A)
```

A példa 2-t fog kiírni, mert ez az A-dimenziós tömb alsó határa. Használni akkor lehet, ha a programban túl sok tömböt használunk és elfelejtettük az alsó határt, vagy az alsó határtól akarunk értékeket beíratni.

Hibajelzés: Ha nincs deklarálnva a tömb, a 20092 számú hibaüzenetet kapjuk.

LBOUND(tömbváltozó, dimenzió N)

A tömbváltozóval jelölt többdimenziós tömb N-edik dimenziójának alsó határát adja meg.

LCASE\$ parancs üzemmód, program üzemmód sztring függvény  
LCASE\$(sztring kifejezés)

Az utasítás a sztring kifejezés nagybetűit kisbetűkké alakítja át.

Példa:

```
PRINT LCASE$("BABA") :REM a baba felírat jelenik meg
```

Hibajelzés: Ha nem sztring kifejezést adunk meg.

LEN parancs üzemmód, program üzemmód sztring függvény  
LEN(sztring kifejezés)

Kiszámítja, hogy a sztring kifejezésben hány darab karakter van.

Példák:

```
a) 100 A$="abcdef"
    110 PRINT LEN(A$)      :REM 7
b) PRINT LEN("ggg1")     :REM 4
```

a) példa: programban való használatot mutatja be.

b) példa: közvetlen parancs üzemmódban használva.

Hibajelzés: Ha a sztring kifejezés a megengedett hosszúságú (234 karakter), akkor hiba nélkül végrehajtódik.

LET parancs üzemmód, program üzemmód utasítás

LET sztring változó = sztring kif.

LET aritm. változó = aritm. kif.

Értékadó utasítás. A változók egyszerű vagy tömbváltozók is lehetnek. Ha a változó neve nem utasítás vagy alapszó, akkor a LET szócska elmaradhat. Ekkor a gép automatikusan LET-et tételez fel és listázáskor kiírja. Egyszerre több változó is kaphatja ugyanazt az értéket.

Példák:

```
a) LET x,y,z=0
b) 100 A$="CSOCSI":B$=A$
    110 PRINT "A$=" A$,"B$=" B$
c) LET LIST=145
```

a) példa: x,y,z egyaránt 0 értéket vesz fel.

b) példa: ha kilistáztatjuk, akkor látható, hogy a gép a LET szócskát az értékadás elé rakta.

c) példa: azt mutatja, hogy BASIC alapszó is lehet változó.

Hibajelzés: Akkor van, ha a változó és a kifejezés típusa nem egyezik, és ha a kifejezés értéke nem esik a megengedett határok közé.

LINE MODE parancs üzemmód, program üzemmód grafikus utasítás

SET # vagy £ csatornaszám: LINE MODE konstans

A rajzolósi módot lehet vele beállítani. A régi képernyőn levő sorok és az új képernyősorok egymásra hatását adja meg. A konstans értéke 0-tól 3-ig terjedhet. A 0 felülírja a régiéket, az 1 VAGY, a 2 és a 3 kizáró VAGY kapcsolatot ad meg. Ha a csatornaszámot elhagyjuk, 101-nek értelmezi.

LINE INPUT parancs üzemmód, program üzemmód adatkezelő utasítás

LINE INPUT sztring változó

Adatkérő utasítás, formailag hasonló az INPUT-hoz, de így a programsort egyetlen karakterláncba olvassa be, és csak sztring kifejezéseket tud beolvasni.

Példa:

```
100 LINE INPUT A$
110 PRINT"A$" A$
```

A példa szerint a gép a megadott karaktersorozatot olvassa be A\$-be.

Hibajelzés: Ha a változó nem sztring típusú, és 132-nél több karaktert akarunk beolvasatni.

LINE STYLE csak program üzemmódban használható grafikus utasítás

SET # vagy £ csatornaszám: LINE STYLE konstans

Szaggatott vonalakat lehet vele rajzoltatni a konstans értéktől függően. Ha a csatornaszámot elhagyjuk, akkor a 101-es grafikus csatornát használja.

Példa:

```
100 GRAPHICS HIRES 4
110 LET Y=1
120 FOR X=10 TO 250 STEP 20
130 SET LINE STYLE Y
140 LET Y=Y+1
```

```

150 PLOT 0,x;1200,X
160 NEXT X
170 SET VIDEO MODE 1
180 SET VIDEO COLOUR 2
190 SET VIDEO X 40
200 SET VIDEO Y 20
210 OPEN # vagy £ 50:"VIDEO:"
220 DISPLAY # vagy £ 50:AT 1 FROM 1 TO 11
230 LET Y=1
240 FOR Z=320 TO 600 STEP 20
250 SET # vagy £ 50:LINE STYLE Y
260 LET Y=Y+1
270 PLOT # vagy £ 50:0,Z;1200,Z
280 NEXT Z
290 END

```

A példa az utasítás különböző grafikus üzemmódokban való használatát mutatja be. Az alsó vonalsor a 4-színű mód, a felső sor a 16 színű módban húzható szaggatott vonalak fajtáit mutatja be.

Hibajelzés: Ha szintaktikailag helyes, nincs.

LIST parancs üzemmód, program üzemmód utasítás  
LIST # vagy £ csatornaszám: programsorszám TO programsorszám  
LIST függvényblokknév

A BASIC munkaterületen tárolt programot, vagy annak egy meghatározott részét listázza ki a megadott csatornára. Alapértelmezésben a rendszercsatornára, tehát a képernyőre listáz. A TO szócska kötőjellel helyettesíthető. A listázást a PAUSE (HOLD) billentyűvel lehet felfüggeszteni és a STOP-pal megállítani.

Példák:

- a) LIST 100 TO 500 :REM 100. sortól 500-ig listáz
- b) LIST FIRST TO 500 :REM az 1. sortól 500-ig
- c) LIST :REM az egész programot kilistázza
- d) LIST 100 TO LAST :REM a 100-adik sortól az utolsóig listáz
- e) LIST GYOK :REM a GYOK nevű blokkot írja ki.

Hibajelzés: Nincs.

LLIST ugyanaz, mint a LIST, de alapértelmezésben a nyomtatóra listáz ki.

LOAD parancs üzemmód, program üzemmód beolvastató utasítás  
LOAD # vagy £ csatornaszám: név  
LOAD eszköznév

Ez az utasítás a paramétereiben megadott programot, fájlt olvassa be a megadott csatornáról. Alapértelmezésben a 106-os csatornáról olvas. Ha eszköznevet adunk meg, akkor az első fájlt olvassa be. Ugyanez a hatása, ha elhagyjuk a program vagy fájl nevét. Ha programot töltünk be, akkor a beolvasott program törölheti az előző programot, és a beolvasott új program lesz az aktuális program. A fájl lehet adat és gépi kódú program is. A fájlnevet csak sztringként írhatjuk be. Ha megadtuk a nevet, akkor a magnetofonkazettát a program kezdete előtt bárhonnán indíthatjuk, a számítógép megkeresi a fájlt, és csak ezt olvasza be.

Példák:

- a) LOAD :REM a magnetofonról betölti az első programot
- b) LOAD "PELDA" :REM a magnetofonról a PELDA nevű programot olvassa be
- c) LOAD "TAPE" :REM a magnetofonról az első programot olvassa be, amelyiket megtalálja.

Hibajelzés: Ha nincs a megadott program, vagy fájlnev, vagy eszköz. A magnetofonról való olvasásnál ügyelni kell a hangerősség beállítására is.

LOG parancs üzemmód, program üzemmód numerikus függvény  
LOG(aritm. kif.)

Egyváltozós függvény, amely kiszámítja az aritmetikai kifejezés természetes e-alapú logaritmusát. Az EXP függvény inverze. Csak a pozitív számoknak van logaritmus.

Példák:

```
a) PRINT LOG(2.7182281829)           :REM 1-et kapunk
b) 100 INPUT X
   110 PRINT LOG(EXP(X))             :REM X-et kapjuk vissza
c) 100 INPUT X
   110 PRINT "LOG2 ="LOG(X)/LOG(2)   :REM 2-es alapú logaritmus
   120 PRINT "LOG10="LOG(X)/LOG(10)  :REM 10-es alapú logaritmus
   130 PRINT "LOG16"="LOG(X)/LOG(16):REM 16-os alapú logaritmus
```

A példák önmagukért beszélnek: A c) példa a más alapú logaritmusokba való átszámítást mutatja be. Az N szám logaritmusát a következő formával kaphatjuk meg:

$$\log_a N = \frac{\log N}{\log a}.$$

LOG10(aritm. kif.)

Ugyanaz, mint az előző, de az aritmetikai kifejezés 10-es alapú logaritmusát számítja ki.

LOG2(aritm. kif.)

Lásd az előzőeket, de az aritmetikai kifejezés 2-es alapú logaritmusát számítja ki.

Hibajelzés: Ha az aritmetikai kifejezés értéke negatív.

LOOK parancs üzemmód, program üzemmód grafikus utasítás

LOOK /# vagy £ csatornaszám AT x,y: változó

Az utasítás megadja egy x,y koordinátájú pont színének palettaszámát. Alapértelmezésben a grafikus csatornát használja. Ha a koordinátapontot elhagyjuk, akkor a rajzolósugár pillanatnyi helyzetét veszi aktuális pontnak. A megadott hely lesz a rajzolósugár aktuális helyzete.

Példa:

```
100 GRAPHICS HIRES 16
110 PLOT 0,0;1279,719
120 LOOK AT 640,360:AB
130 PRINT AB
```

A 640,360 koordinátájú pontnak megkapjuk a palettaszín-kódszámát.

LOOP 1. részletesen a DO ciklusnál.

LPRINT parancs üzemmód, program üzemmód adatkezelő utasítás

Részletesebben a PRINT utasításnál foglalkozunk vele. A PRINT-től abban tér el, hogy alapértelmezésben a 104-es PRINTER csatornára ír ki.

Példa:

```
500 LPRINT X,Y,Z
```

Programba beillesztve a változók értékeit a nyomtatóra küldi, ami ott kinyomtatódik.

Hibajelzés: Ha a géphez nincs nyomtató csatlakoztatva.

LTRIM\$ parancs üzemmód, program üzemmód sztring függvény  
LTRIM\$(sztring kifejezés)

A karaktersorozat elején álló szóközöket törli.

Példa:

```
100 PRINT "    LH"
110 PRINT LTRIM$("    LH")
```

A példa önmagáért beszél, az utasítás törölte a szóközöket.

Hibajelzés: Ha szintaktikailag helyes, nincs.



MAGENTA parancs üzemmód, program üzemmód szín-segédfüggvény  
SET utasítás MAGENTA

Ün magában nem használható; a bíborvörös színt állítja elő (azonos RGB(1,0,1)).

Példa: 1. az RGB függvénynél.

Hibajelzés: Ha az utasítás többi része jó, nincs.

MAX parancs üzemmód, program üzemmód numerikus függvény  
MAX (aritm. kif., aritm. kif.)

A két aritmetikai kifejezés közül kiválasztja a nagyobbikat.

Példa:

```
PRINT MAX(88,6)           :REM 88
```

A példához sok magyarázat nem kell.

Hibajelzés: Gyakorlatilag nem lehetséges.

MAXLEN parancs üzemmód, program üzemmód sztring függvény  
MAXLEN (sztring változó)

Az utasítás a sztring változó maximális hosszát adja meg karakterekben mérve.

Példa:

```
PRINT MAXLEN(a$)         :REM 132-t ad.
```

Hibajelzés: Ha nem sztring változót adunk meg.

MERGE parancs üzemmód, program üzemmód beolvastató utasítás  
MERGE # vagy £ csatornaszám: név

A megadott csatornáról a névvel jelzett programot vagy fájlt olvassa be. A LOAD-tól annyiban tér el, hogy nem törli az előző

programot, így hozzá lehet olvastatni a gépben lévő programhoz egy másik programot. Használatával vigyázni kell, mert törli a változókat, és a betöltött új program sorai felülírják a gépben lévő program azonos számú sorait. Alapértelmezésben a 106-os csatornát használja.

Példa:

```
MERGE "UJPROG"      :REM az UJPROG nevű fájlt beolvastatja.
```

Hibajelzés: Gyakorlatilag nincs, de a felülírás gondot okozhat.

MIN parancs üzemmód, program üzemmód numerikus függvény  
MIN(aritm. kif., aritm. kif.)

Az aritmetikai kifejezésekből kiválasztja a kisebbet.

Példa:

```
PRINT MIN(1,18)     :REM 1
```

A példa önmagáért beszél.

Hibajelzés: Gyakorlatilag nem lehetséges.

MOD parancs üzemmód, program üzemmód numerikus függvény  
MOD(1) aritm. kif. (2) aritm. kif.

Az utasítás az első aritmetikai kifejezés másodikkal való osztásának egészmaradékát adja meg, ha a kifejezések pozitívak. Ha a második aritmetikai kifejezés negatív, 0-t ad. Abban az esetben, ha az első kifejezés negatív, akkor azt az egész számot adja meg, amelyet az első kifejezés abszolút értékéhez kell hozzáadni, hogy a második kifejezéssel maradék nélkül osztható legyen.

Példa:

```
PRINT MOD(3,8)      :REM 3  
PRINT MOD(-1,15)    :REM 14  
PRINT MOD(9,2)      :REM 1  
PRINT MOD(-9,15)    :REM 6  
PRINT MOD(3,-2)     :REM 0
```

Hibajelzés: Ha az aritmetikai kifejezés értéke nem lépi túl a legnagyobb számot, akkor nincs.

NET CHANNEL parancs üzemmód, program üzemmód csatornakezelő utasítás

SET NET CHANNEL(aritm. kif.)

ASK NET CHANNEL aritm. változó

Hálózatra kapcsolás esetén a 2+ csatorna számát állítja be, amelyen a beérkező adatok a pufferbe kerülnek. Lekérdezni azt a csatornaszámot lehet, amelyiken a soros vonalon keresztül az olvasásra váró adat érkezett.

Ha nincs ilyen csatorna, akkor automatikusan a 255-ös értéket veszi fel.

Példa:

```
ASK NET CHANNEL A
```

```
PRINT A          : REM 255-öt kapunk
```

Mivel a gépünk a soros vonalon nem kommunikál, így a 255-ös csatornán áll.

Hibajelzés: Ha egyetlen csatornán sincs adat, akkor a 255-ön áll, így gyakorlatilag nincs.

NET MACHINE parancs üzemmód, program üzemmód csatornakezelő utasítás

ASK NET MACHINE aritm. változó

Szintén hálózatban használatos, és a NET CHANNEL-lal kap értéket. A változóba annak a gépnek a hálózati számát teszi, amelyik az adatokat küldte. Ez akkor lehet fontos, ha valamelyik gép az általános adás csatornáján küldött szét adatokat, és szeretnénk megtudni, hogy melyik volt a küldő. Ha nincs hálózatba kapcsolva a gép, akkor 255-öt ír ki. Használata olyan, mint a NET CHANNEL-é.

SET NUMBER parancs üzemmód, program üzemmód hálózatkezelő utasítás

SET NET NUMBER aritm. konstans

Lokális hálózatba kapcsolás esetén a saját gépünk hálózatbeli számát állíthatjuk be vele. Alapállapotban ez az érték 0, de ez hibát okoz a hálózat használatában. A beállítandó érték 1--32 terjedhet, a még szabad értékekre. Lekérdezni az ASK utasítással lehet.

Példa:

```
SET NET NUMBER 2 :REM 2 számú gép leszünk a hálózatban.
```

Hibajelzés: A beállításnál akármilyen értéket beállíthatunk, de ha a hálózat használatánál nem megfelelő értéket állítunk be, hibát okoz.

NEW csak parancs üzemmódban használható utasítás

NEW

Törli a memóriában tárolt aktuális programot és változókat.

NEW ALL

A NEW ALL az összes programot törli a gépből. Nincsenek paramétere, a szintaxisának megfelelően töröl.

Példa:

```
NEW :REM törölte az éppen használt programot
```

```
NEW ALL :REM az összes programot és változókat törölte.
```

Hibajelzés: Nincs, de a programunk elvész, ezért használatával ügyelnünk kell!

NEXT A FOR ciklus záróutasítása, részletesen 1. ott.

NUMERIC csak program üzemmódban használható utasítás

NUMERIC aritm. változó (tömbelemszám)

Az utasítás aritmetikai változók és tömbök létrehozására alkalmas. Kétdimenziós tömböket is létrehozhatunk vele. A tömb elemeire az aritmetikai változó (elemszám) kifejezéssel hivatkozhatunk. A mátrixok többelemeire is ugyanígy hivatkozhatunk. Alapértelmezésben az alsó indexhatár 0.

A tömb 2, 4 eleme a rajzon bevonalkázott.

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

Példák:

- a) 100 NUMERIC a(10) :REM egy 10-elemes tömb létrehozása
- b) 100 NUMERIC b(1 TO 20, 1 TO 5) :REM kétdimenziós tömb létrehozása

Hibajelzés: Ha a tömb elemeit nem számokkal, hanem sztring kifejezésekkel adjuk meg.

- ON csak program üzemmódban használható feltételes utasítás
- ON aritm. kif. GOTO sorszámlista
- ON aritm. kif. GOSUB sorszámlista

Az utasítás többirányú elágazást tesz lehetővé. Kiszámítja az aritmetikai kifejezés értékét, és ezt egészszé alakítja át, legyen ez X érték. Ezután a sorszámlistából kiválasztja az X-edik elemű sorszámot, és ettől a programsortól folytatja a program végrehajtását. Ha a listában nem szerepel X-edik elem, akkor az utasítás hatástalan.

Példák:

- a) 200 ON SGN(A)+2 GOTO 200,300,400
- b) 300 ON RND(30)+1 GOTO 100,200,300,400,  
500,600,...,2900,4000

a) példa: az előjelfüggvény használatára is példa. Ha A értéke negatív, a 200-as sortól, ha 0, a 300-as sortól és ha pozitív, a 400-as sortól folytatja a program végrehajtását.

b) példa: a véletlen programelágazást mutatja, ami különösen a játékprogramoknál hasznosítható.

Hibajelzés: Ha az aritmetikai kifejezés egészértéke nem esik a megengedett intervallumba, és ha a sorszám nem létező sorra mutat, a 20004-es kódszámú hibaüzenetet kapjuk.

OPEN parancs üzemmód, program üzemmód adatkezelő utasítás  
OPEN # vagy £ csatornaszám:"NÉV" "ESZKÖZ" /fájlnev ACCESS mód

A csatornaszámban megadott csatornát megnyitja, és hozzárendel egy eszközt és egy fájlt.

Részletes leírása megtalálható a Felhasználói kézikönyvben.

Példa:

```
OPEN # vagy £ 200:"EDITOR:"
```

A 200. csatornát nyitja meg az EDITOR számára.

Hibajelzés: A megadott paraméterek kiértékelése közben sok hiba előfordulhat: pl. nem létezik az eszköz stb.

OPTION parancs üzemmód, program üzemmód segédutasítás  
OPTION ANGLE DEGREES vagy RADIANS

Az utasítással szögek kiszámítása esetén a fokban vagy radiánban való számítási módot lehet kiválasztani. A gép alapértelmezésben radiánban számol. A DEGREES fokban számol, a RADIANS radiánban.

Példák:

a) 100 INPUT X

```
110 PRINT SIN(X)
```

b) 200 INPUT X

```
210 OPTION ANGLE DEGREES
```

```
220 PRINT SIN(X)
```

Hibajelzés: Ha formailag helyes, gyakorlatilag nincs.

OR parancs üzemmód, program üzemmód logikai kifejezés  
aritm. kif. OR aritm. kif.

A két aritmetikai kifejezésnek a logikai VAGY kapcsolatát adja meg.

Példa:

```
100 INPUT PROMPT "X" = ":X
110 IF X < 0 OR X > 100 THEN
120 PRINT "IGAZ"
130 ELSE
140 PRINT "HAMIS"
150 END IF
```

A példában akkor kapunk igaz üzenetet, ha az X kisebb, mint 0, vagy nagyobb, mint 100.

Hibajelzés: Ha a műveletek hibásak, vagy az aritmetikai kifejezések nem megfelelőek.

ORD parancs üzemmód, program üzemmód sztring függvény  
ORD (sztring kif.)

Az utasítás a sztring kifejezésben a szöveg első karakterének ASCII kódját adja meg.

Példa:

```
PRINT ORD("dad") : REM 100-at ad
```

Hibajelzés: Nincs.

OUT parancs üzemmód, program üzemmód gépi kódú utasítás  
OUT aritm. kif.(N), aritm. kif.(X)

Az N című INPUT/OUTPUT pontra kiír X darab bájtot.

Hibajelzés: Ha az aritmetikai kifejezések értékei nem esnek a 0--255 intervallumba.

PALETTE parancs üzemmód, program üzemmód grafikus utasítás  
SET vagy csatornaszám: PALETTE 0, 1, 2, 3, 4, 5, 6, 7

Grafikus utasításoknál a használni kívánt színeket mi választhatjuk ki. Ha kiadjuk a GRAPHICS HIRES 16 utasítást, akkor a rajzoláshoz 16 színt használhatunk, így változatos színű rajzok készíthetők. A használni kívánt paletta első nyolc színét (0--7) mi választhatjuk meg tetszőlegesen. A 0-ás kódban megadott szín lesz az első és így tovább. A kevesebb színt használó üzemmódokban (HIRES 2, HIRES 4) csak az első kettő, ill. 4 szín használható. Az alaplap (papír) és a rajzoló sugar (tinta) színét a palettaszín sorszámaival adhatjuk meg. A palettában a színek megadásához használhatjuk a szín számát, a nevét vagy az RGB függvényt. Alapértelmezésben a 101-es grafikus csatornát használja.

Példa:

```
100 GRAPHICS HIRES 16
```

```
110 SET PALETTE 0, 255, RGB(0.5,0.7, 1), YELLOW,54,67, BLUE,  
RGB(0.1,0.9,0.9)
```

A példa alapján beállítottuk a palettánk első nyolc színét. A második nyolcat a BIAS utasítással lehet beállítani. Ezután erről a palettáról válogathatjuk a színeket.

Hibajelzés: Ha nincs megnyitva a megadott csatorna, vagy nem létezik a megadott szín.

PAPER parancs üzemmód, program üzemmód grafikus utasítás  
SET # vagy £ csatornaszám: PAPER színszám

Az utasítás a megadott csatornára beállítja az alaplap (papír) színét. Alapértelmezésben a grafikus (101) csatornát használja. A "COLOUR MODE 3"-nál a szín számával, a többi esetben a paletta számával lehet a színt beállítani. 80 karakteresszöveges lapokon a palettáról csak az első négy színt, 40 karaktereseken csak az első két színt lehet kiválasztani papírszínnek.



Példa:

```
100 GRAPHICS HIRES 16
110 SET PAPER 2
120 SET INK 3
130 CLEAR SCREEN
```

A gyárilag beállított palettáról a 2-es színkódú lesz a papír, a 3-as pedig a tinta színe. Az új színű lapokat az előző törlése állítja be.

Hibajelzés: Ha a csatorna nincs megnyitva, vagy esetleg nem létező színt adunk meg.

PEEK parancs üzemmód, program üzemmód memóriakezelő függvény  
PEEK(aritm. kif.)

Az aritmetikai kifejezésben megadott sorszámú memóriacím tartalmát adja vissza. Az értéknek 0 és 65535 közé kell esnie. Ha nagyobb értéket adunk meg, akkor a 65535 után előlről kezdi el kiírni a tárcímeket.

Példa:

```
100 CLEAR SCREEN
110 FOR X=0 TO 65535
120 PRINT ,X " M=" PEEK(X)
130 NEXT X
```

A példa az összes memóriacímen lévő értéket kiírja a képernyőre.

Hibajelzés: Csak akkor van, ha nem létező memóriacímet adunk meg.

PI parancs üzemmód, program üzemmód numerikus függvény  
PI

A függvénynek nincs változója. Megadja a PI nevű szám (TT) értékét.

Példák:

a) PRINT "PI="PI

b) PRINT "SIN(PI/4)="SIN(PI/4) :REM SIN(PI/4)=.7071867812

A példák önmagukért beszélnek. A b) példából látható, hogy a PI-t nem kell számként beírni, hanem elég függvényként meghívni.

Hibajelzés: Gyakorlatilag nincs.

PING parancs üzemmód, program üzemmód zenei utasítás

Az utasítás hatására a gép kiad egy csilingelő hangot.

Akkor lehet használni, ha például a programban valahol (indulásnál, vagy ha gépi kódot hívunk, vagy ha rajzolni fog a gép) valamire fel szeretnénk hívni a figyelmet.

Hibajelzés: Nincs.

PLOT parancs üzemmód, program üzemmód grafikus utasítás

PLOT vagy csatornaszám: X1,Y1;X2,Y2;X3,Y3;...

PLOT vagy csatornaszám: ANGLE aritm. kif.

PLOT vagy csatornaszám: FORWARD vagy BACK aritm. kif.

PLOT vagy csatornaszám: LEFT vagy RIGHT aritm. kif.

PLOT vagy csatornaszám: ELLIPSE aritm. kif. aritm. kif.

PLOT vagy csatornaszám: PAINT

A PLOT utasítással (és a megadott feltételekkel) meghatározott vonalakat rajzolhatunk. Az utasítás alapértelmezésben a 101-es grafikus csatornát használja. A grafikus lap felosztása a koordináta-rendszerben 1280, ill. 720 pont, így a képernyőn a bal alsó pont a 0,0 és a jobb felső pont az 1279, 719. Ha az X,Y pontokat adjuk meg, akkor a megadott koordináták közé rajzol vonalat. Ha folytonos vonalat akarunk, a ; jelet kell beírni a koordinátaértékek közé. Az ANGLE utasítás az aritmetikai kifejezésben megadott szöggel elforgatja a rajzolás irányát. A FORWARD azt jelenti, hogy a megadott irányba előre, a BACK, hogy hátra annyit mozdul el a rajzoló sugar, amennyit az aritmé-

titikai kifejezésben megadtunk. A EFT és a RIGHT a sugarat az aritmetikai kifejezésben megadott fokkal jobbra vagy balra forgatja a haladási irányhoz képest. A ELLIPSE utasítással ellipsziseket és köröket rajzolhatunk a képernyőre. Az első kifejezésben az X irányú tengely félhosszát, a másodikban az Y irányú tengely félhosszát adhatjuk meg. Ha a két aritmetikai kifejezés egyenlő, akkor kört rajzolhatunk. A PAINT utasítással területeket lehet beszínezni. A rajzolásugár mindig az INK parancsban megadott színnel dolgozik.

#### Példa:

```
100 OPTION ANGLE DEGREES
110 GRAPHICS HIRES 4
120 PLOT 0,0;10,100;100,50;1200,300;200,640;
130 PLOT ANGLE 34;
140 PLOT 100,23;
150 PLOT LEFT 20;
160 FOR A=1 TO 200
170   LET Y=RND(600):LET X=RND(1200)
180   LET Z=RND(200)
190   PLOT X,Y,
200   SET INK 4
210   PLOT ELLIPSE Y,Z,
220   LET C=RND(1000)
230   PLOT C,300,
240   SET INK 2
250   PLOT PAINT
260   PLOT 400,Z,
270   SET INK 3
280   PLOT PAINT
290 NEXT A
```

Hibajelzés: Ha túllépjük az 1279,719-es határokat.

POKE parancs üzemmód, program üzemmód gépi kódú programutasítás

POKE memóriacím, érték

Az utasítással egy tetszőleges memóriacímre beírhatunk egy 0 és 255 közötti számot. Az értékrészben adjuk meg a tárolandó bájtot. A memóriacím 0--65535, az érték 0--255. Lekérdezni a PEEK utasítással lehet.

Példa:

```
POKE 123,67
```

A 123-as memóriacímre (007C) elhelyezi a 67-es számú bájtot.

Hibajelzés: A paraméterek kiértékelésében sok hibalehetőség van. A használatához jól kell ismerni a memóriakiosztást és a gépi kódú programozást.

POS parancs üzemmód, program üzemmód sztring függvény  
POS(A\$,B\$,n)

A B\$ előfordulási helyét keresi meg az A\$-ben az n-edik pozíciótól kezdve, és a pozícióértéket adja meg. Az n elhagyható, akkor viszont a szövegrész első előfordulási helyét keresi meg a szövegben. Ha nincs meg a szövegrészlet, 0 értéket vesz fel.

Példa:

```
PRINT POS ("VALAHOL","HOL"):REM 5
```

A HOL előfordulását keresi meg.

Hibajelzés: Ha szintaktikailag helyes, nincs.

PRINT parancs üzemmód, program üzemmód kiírató utasítás  
PRINT # vagy £ csatornaszám, AT sorkifejezés, oszlopkifejezés, utasításkifejezés  
PRINT # vagy £ csatornaszám, USING sorszám: utasításkifejezés  
PRINT # vagy £ csatornaszám, USING sztring kifejezés, utasításkifejezés

Kiszámítja az utasításkifejezés értékét, majd a megadott formában kiírja a csatornára. Alapértelmezésben a képernyő a 102-es

csatornát használja. Az utasításkifejezésben csak a TAB és a kiíratandó szöveg lesz. A kiíratás formátumát a következőképpen lehet megadni.

TAB(y) a megadott oszloptól ír ki.

AT x,y az x-edik sortól és y-odik oszloptól kezdődik a kiíratás. Ha vesszőt teszünk a kiíratandó szöveg elé, akkor az előre meghatározott első tabulátorpozícióra ugrik és onnan ír ki.

Ha az első kiírt kifejezés után mást is akarunk kiíratni ugyanabban a sorban, akkor pontosvesszőt kell a kifejezés végére tennünk, és a kiíratás mindig a következő karakteren kezdődik.

A kiíratás formázása vagy a USING-ot követő IMAGE kifejezés sorszámával, vagy a USING-ot követő formátumadatok megadásával lehetséges. Ha a képernyőre írunk, a 24x40 karakteres felosztást használjuk.

Példák:

a) PRINT TAB(10)" INNEN KEZD KIIRNI" :REM 10-edik oszloptól ír ki.

b) PRINT AT 10,10:"IDE" :REM a képernyő közepére ír.

c) 100 FOR X=0 TO 24

110 PRINT ,X;

120 NEXT X

a) és b) példa: a képernyőn tetszőlegesen elhelyezett szöveg kiírása.

c) példa: az X-eket folyamatosan kiírja, de mindig a tabulátorpozícióktól kezdve.

Hibajelzés: Ha nincs szintaktikai hiba, akkor végrehajtódik.

PROGRAM csak parancs üzemmódban használható

PROGRAM név(változók)

Ha több programot használunk, akkor a programokat el kell neveznünk. Ezzel az utasítással ezt megtehetjük. A programjaink első sorába a PROGRAM utasítást beírva, az utána álló karaktereket a gép gyakorlatilag nem veszi figyelembe. Az INTO utasítást kiadva mindig megkaphatjuk a programjaink nevét.

Példák:

- a) 110 PROGRAM ELSO
- b) EDIT 1  
100 PROGRAM MASODIK
- c) EDIT 2  
100 PROGRAM HARMADIK

Így lehet a különböző programokat aktuálissá téve megjegyezni. Az EDIT-nél már beszéltünk róla.

Hibajelzés: Gyakorlatilag nincs.

RAD parancs üzemmód, program üzemmód trigonometrikus függvény  
RAD(aritm. kif.)

Az aritmetikai kifejezésben megadott fok értéket számolja át radiánba.

Példák:

- a) PRINT RAD(180) :REM 3.141592654
- b) PRINT RAD(90) :REM 1.570796327

- a) példa: PI értékét adja meg.
- b) példa: PI/2 értékét adja meg.

Hibajelzés: A kiírátható legnagyobb számig nincs.

RANDOMIZE csak program üzemmódban használható segédfüggvény

Véletlenszám-generálásánál az alapértéket 0-ra állítja, így változik a véletlenszám-generálás.

READ parancs üzemmód, program üzemmód adatkezelő utasítás  
READ változólista  
READ IT MISSING sorszám: változólista

A programban elhelyezett DATA utasításokkal tárolt adatokat olvassa be és rendeli hozzá az utasításban megadott változóhoz. A változók tetszőleges típusú, egymástól elválasztott egyszerű és tömbváltozók lehetnek. Ha az adatok elfogytak, akkor az IF MISSING részben meghatározott tevékenységet hajtja végre a gép.

#### Példa:

```
100 DO
110 READ A$,B,C
120 LET K=B+C
130 PRINT ,A$ ,K
140 LOOP UNTIL K=10
150 DATA első ... ,2,3, második,3,3, harmadik,4,3, negyedik,
    3,5
160 DATA ötödik...,4,3, hatodik,3,7
170 END
```

A program a READ használatát mutatja be. Az a AS-ba beolvassa az "első ... " sztringet, a B-be a 2-t, a C-be 3-at és így tovább, amíg az összeset ki nem írja és befejezi a program futását.

Hibajelzés: Ha a változó és a DATA utasításban megadott adatok típusa eltér egymástól.

REDIRECT parancs üzemmód, program üzemmód adatkezelő utasítás

REDIRECT FROM # vagy £ csatornaszám 1 TO # vagy £ csatornaszám 2

Az első számú csatornáról beolvasott adatokat átadja és kiírja a második csatornára.

REM parancs üzemmód, program üzemmód utasítás

REM karaktorsor

Az utasítással a program szövegébe megjegyzéssorokat tehetünk. A gép a REM észlelésétől kezdve a sort a továbbiakban nem veszi

figyelembe, és a következő utasítással folytatja. A REM után bármilyen karakterek állhatnak. A leggyakrabban program megnevezésére és utasítások hatásának leírására használható. Listázáskor a REM sort is kiírja a számítógép. A REM helyettesíthető a felkiáltójellel.

Példák:

- a) 100 REM szamlalas  
110 ! ide kerülhet a program
- b) PRINT 3:REM 4

Mint látható, parancs üzemmódban is használható. A 4-est nem veszi figyelembe a gép.

Hibajelzés: Nincs.

REM 1 parancs üzemmód, program üzemmód gépkezelő utasítás  
REM 2  
SET REM 1 ON/OFF  
SET REM 2 ON/OFF

A magnó távvezérlő-kimenetét lehet vele be-, ill. kikapcsolni. A TOGGLE utasítással lehet átkapcsolni, ez az alapértelmezése a 4-es és a 12-es funkcióbillentyűnek.

Példa:

SET REM 1 ON

Ha bekapcsoljuk a magnetofont, de nem akarunk betölteni vele, akkor elindíthatjuk az utasítással. A magnetofont a vezérlőutasítások (LOAD, SAVE, MERGE) automatikusan állítják.

REM(A,B) parancs üzemmód, program üzemmód numerikus függvény  
REM(aritm. kif.,aritm. kif.)

Az első aritmetikai kifejezésnek a második aritmetikai kifejezéssel való osztásának maradéka.



Példák:

a) PRINT REM(8,5) : REM 3-at ad

b) PRINT REM(4,5) : REM 4-et ad

A példák önmagukért beszélnek: az osztás utáni egészmaradékot kapjuk.

Hibajelzés: Gyakorlatilag nincs.

RENUMBER csak parancs üzemmódban használható segédutasítás

RENUMBER sorszám TO sorszám AT első sorszám STEP lépésköz.

RENUMBER blokknév AT első sorszám STEP lépésköz

Az aktuális programot, vagy annak egy részét lehet újraszámolni vagy átszámolni az utasítással. A paramétereket elhagyva a teljes program átszámozódik a 100-as sorszámmal indulva, 10-es lépésközzel. Ha a program utolsó sorszáma meghaladja a 9999-es értéket, a program nem számozódik át. A DEF-ben vagy a handler-ben megadott blokknévvel is lehet hivatkozni az átszámozandó részekre. Az utasítás a vezérlésátadó utasításoknál található sorszámokat is átírja.

Példa:

```
RENUMBER 1 TO 1000 AT 5 STEP 10
```

A program 1. sorától az 1000.-ig átírja a sorszámokat, 10-es lépésközökkel. Az új kezdősorszám 5 lesz.

Hibajelzés: Ha túllépjük 9999-es határt.

RESTORE parancs üzemmód, program üzemmód adatkezelő utasítás

RESTORE sorszám

Az utasítás hatása a következő: a READ utasítás a megadott sorszámú DATA sortól kezdi el beolvasni a tárolt adatokat. Ha elhagyjuk a sorszámot, akkor az első DATA sortól kezdi a beolvasást.

Élda:

```
100 DATA 1,2,3,4,5
200 RESTORE 300
210 DO
220 READ A
230 PRINT ,A
240 LOOP UNTIL A=10
300 DATA 6,7,8,9,10
310 END
```

A példa 6--10. beolvassa és kiírja a számokat. Ha elhagyjuk a RESTORE-ban a 300-at, akkor a 100 DATA sorral kezdi a beolvást és 1--10. írja ki a számokat.

Hibajelzés: Ha a RESTORE nem DATA sorra mutat, a READ utasításban kaphatunk hibát.

RETRY csak program üzemmódban használható megszakításkezelő utasítás

Megszakításkor ezzel az utasítással léphetünk vissza a megszakítást okozó sort követő sorra, a hiba lekezelése után.

RETURN csak program üzemmódban használható utasítás

Hatására a vezérlés a GOSUB utasítást követő első sorra ugrik vissza a GOSUB-ban meghívott programrészletből.

Hibajelzés: Ha nem találja meg a visszatérési címet, vagyis nincs GOSUB utasítás.

RGB parancs üzemmód, program üzemmód szín-segédfüggvény  
RGB(R,G,B)

A függvénnyel színf adhatunk meg a színkód használata helyett. A színeket a piros (RED), a zöld (GREEN) és a kék (BLUE) alapszínekből keverhetjük ki az arányok megadásával. A gép 0--255

Kódolja a színeket, 0 a fekete, 255 a fehér. Az alapszíneket a nevükkel is megadhatjuk.

Függvény	Színkód	Név	
RGB(0,0,0)	0	BLACK	fekete
RGB(1,0,0)		RED	piros
RGB(1,1,0)		YELLOW	sárga
RGB(0,1,0)		GREEN	zöld
RGB(0,1,1)		CYAN	lila
RGB(0,0,1)		BLUE	kék
RGB(1,0,1)		MAGENTA	bíbor
RGB(1,1,1)	255	WHITE	fehér

Példa:

```
100 FOR R=0 TO 1 STEP.1
110 FOR G=0 TO 1 STEP.1
120 FOR B=0 TO 1 STEP.1
130 SET BORDER RGB(R,G,B)
140 NEXT B
150 NEXT G
160 NEXT R
```

A példa a keret színét változtatja a megadott a feketétől a fehérig.

Hibajelzés: Ha a különböző alapszíneknél túllépjük a 0--1 intervallumot.

RND parancs üzemmód, program üzemmód numerikus függvény

A 0 és 1 közötti intervallumba eső véletlenszámot generál. A véletlenszám sosem éri el a 0-t és az 1-et. Ha nagyobb számot akarunk kapni, szorzással növelhetjük.

Példa:

```
100 FOR X=0 TO 10
110 PRINT RND
120 NEXT X
```

A példa 10 db 0 és 1 közötti véletlenszámot állít elő. Változtassuk át a 110-es sort PRINT INT(RND\*10)-re, és indítsuk el, így 10 db egész számot kapunk a 0--10 intervallumban.

Hibajelzés: Nincs.

RND(x) parancs üzemmód, program üzemmód numerikus függvény  
RND(aritm. kif.)

Ugyanaz, mint az RND, de itt 0 és az aritmetikai kifejezés közötti egész számot kapunk véletlenszámként. Az aritmetikai kifejezés maximális értéke 32767 lehet.

Példa:

```
PRINT RND(100)
```

0--100 között kapunk egész véletlenszámot. Gyakorlatilag ugyanezt csinálja a

```
PRINT INT(RND*100)
```

is, de ez az utóbbi módszer hosszabb, ezért célszerűbb az egészértéket az RND(X) utasítással generáltatni.

Hibajelzés: Ha túllépjük a 32767-es értékhatárt.

ROUND parancs üzemmód, program üzemmód numerikus függvény  
ROUND(aritm. kif.,konstans)

Az utasítás az aritmetikai kifejezést kerekíti a konstansban megadott tizedes értékre. A kifejezés a közelebbi értékre kerekít.

Példák:

```
PRINT ROUND(1.2350,2) :REM 1.23  
PRINT ROUND(1.2351,2) :REM 1.24  
PRINT ROUND(-1.2351,2) :REM -1.24  
PRINT ROUND(12.351,0) :REM 12
```

Hibajelzés: Az aritmetikai kifejezés kiértékelése közben, ha az nem konstans.

RTRIM\$ parancs üzemmód, program üzemmód sztring függvény  
RTRIM\$(sztring kif.)

A sztring kifejezésben megadott karakterek végén lévő szóközöket levágja.

Példa:

```
100 LET A$="BABA  "
110 PRINT A$ "ITT A VEGE"
120 PRINT RTRIM$(A$):"MOST ITT.A VEGE"
```

A példa az utasítás hatását mutatja be.

Hibajelzés: Csak akkor van, ha szintaktikailag hibás, vagy nem sztring kifejezést használunk.

RUN parancs üzemmód, program üzemmód alaputasítás  
RUN sorszám

Hatására a számítógép az aktuális tárolt programot, az adott sorszámától kezdve végrehajtja. Ha elhagyjuk a sorszámot, akkor a programot az elejétől kezdi el végrehajtani. A változókat törli.

Példák:

```
a) RUN :REM az első sortól futtatja a programot
b) RUN 200 :REM a 200-as sortól kezdi a végrehajtást.
```

Hibajelzés: Ha a megadott sorszámú sor nem létezik.

SAVE csak parancs üzemmódban használható utasítás  
SAVE "fájlnev"  
SAVE # vagy £ csatornaszám:"fájlnev"  
SAVE ALL # vagy £ csatornaszám:"fájlnev"

A megadott nevű fájlba átmásolja (ementi) az aktuális programot. Ha nem adunk meg csatornát, akkor a 106-os csatornára ment. Ha az ALL utasítással együtt adjuk meg, akkor az összes

programot elmenti, de a programok beolvasáskor felülírják egymást. A SAVE után eszköznevet is megadhatunk.

Példák: Gépeljük be akármelyik példaprogramot, és csatlakoztasuk a magnetofont. Bármelyik utasítással elmenthetjük rá a kiválasztott programot.

- a) SAVE "PELDA"
- b) SAVE "TAPE:"
- c) SAVE # vagy £ 106:"PELDA"

Hibajelzés: Ha az utasításban megadott eszköz nincs.

SCROLL parancs üzemmód, program üzemmód utasítás  
SET # vagy £ csatornaszám:SCROLL ON/OFF

Az automatikus képernyőgörgetést kapcsolja ki-, ill. be. Alapértelmezésben a 102-es csatornát használja.

SEC parancs üzemmód, program üzemmód trigonometrikus függvény  
SEC(aritm. kif.)

Az aritmetikai kifejezésben megadott értéknek kiszámítja a szekánsát. Értelmezése:

$$\text{SEC(ALFA)} = \frac{1}{\text{COS(ALFA)}} = \frac{C}{B}$$

Alapértelmezésben radiánban kell megadni ALFA értékét.

Példa:

```
100 OPTION ANGLE DEGREES
110 PRINT SEC(0)
```

A példában 1-et kapunk eredményül, de SEC(180)=-1.

Hibajelzés: Mivel páros függvény, így gyakorlatilag nincs.

SELECT csak program üzemmódban használható feltételes utasítás

```
SELECT CASE aritm. kif.  
CASE struktúrák  
END SELECT
```

Az utasításblokk többirányú elágazást tesz lehetővé a programban. Részletesen l. a Felhasználói kézikönyben.

SERIAL BAUD parancs üzemmód, program üzemmód soroscsatorna-kezelő utasítás

```
SET SERIAL BAUD aritm. kif.
```

Az utasítással a soros kimenet és a hálózat átviteli sebességét lehet megadni. Az aritmetikai kifejezés 0--15. kódolt, de 255-ig terjedhet. A 0 kód megfelel 50 baudnak, a 15-ös 9600-nak. A kódok értékei a Felhasználói kézikönyvben megtalálhatók. Lekérdezni az ASK utasítással lehet. Alapérték 15.

Példa:

```
ASK SERIAL BAUD A  
PRINT A :REM 15-öt kapunk.
```

Hibajelzés: Ha 255-nél nagyobb értéket állítunk be.

SERIAL FORMAT parancs üzemmód, program üzemmód soroscsatorna-kezelő utasítás

```
SET SERIAL FORMAT aritm. kif.
```

A soros csatornán használt formátumot állíthatjuk be vele az egyes biteken keresztül. Részletesen l. a Felhasználói kézikönyvben.

SET parancs üzemmód, program üzemmód gépifeltétel-beállító utasítás

```
SET utasítás
```

Ha különböző gépi feltételeket akarunk beállítani, az utasítások előtt használjuk ezt a szócskát.

Példa:

```
SET BORDER 27
```

Hibajelzés: Ha az utasítás többi része jó, nincs.

SGN parancs üzemmód, program üzemmód numerikus függvény  
SGN(aritm. kif.)

Előjelfüggvény. Az aritmetikai kifejezés értékétől függően a függvény három értéket vehet fel:

ha  $X > 0$ , akkor  $SGN(X) = +1$

$X = 0$ , akkor  $SGN(X) = 0$

$X < 0$ , akkor  $SGN(X) = -1$

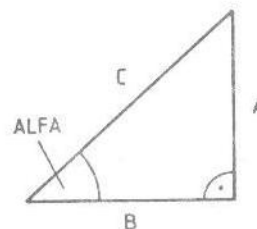
Példa:

```
PRINT SGN(-15) :REM -1-et ad
```

Hibajelzés: Ha az aritmetikai kifejezés túl nagy, vagy a konstanson kívül mást is tartalmaz.

SIN parancs üzemmód, program üzemmód trigonometrikus függvény  
SIN(aritm. kif.)

Az utasítás a radiánban megadott szög szinuszát számítja ki.



Példák:

```
a) PRINT SIN(PI/4) :REM 0.7071067812
```

```
b) 100 CLEAR SCREEN
```



```

110 FOR X=0 TO 10
120 LET A=12-INT(SIN(X*PI/9)*10
130 LET C=X*2
140 PRINT AT A,C:"*"
150 NEXT X
160 PRINT AT 23,1:"vege"

```

A példák önmagukat mutatják be. A második példa egy szinuszgörbét rajzol a képernyőre.

Hibajelzés: Az aritmetikai kifejezés kiértékelésében lehet.

SINH parancs üzemmód, program üzemmód trigonometrikus függvény  
 SINH(aritm. kif.)

Az utasítással kiszámítjuk az aritmetikai kifejezésben megadott érték szinusz hiperbolikusát. Ez a függvény a - végtelentől a + végtelenig monoton növekvő függvény.

Példa:

```
PRINT SINH(12) :REM 81377.39571
```

Hibajelzés: Ha a kiszámításhoz túl nagy értéket adunk meg.

SIZE parancs üzemmód, program üzemmód sztring függvény  
 SIZE(tömbazonosító)

Az azonosítóval megadott tömb számát adja meg.

Példa:

```

100 DIM a(10)
110 REM a program további része
200 PRINT SIZE(a)

```

HA begépelünk egy programot és ott tömböket használunk, lekérdezhetjük, hogy hány eleme van a tömbnek. Példánkban 11 tagú a tömb.

SIZE(tömbazonosító, dimenziószám)

Ha így adjuk meg, akkor a tömb megadott dimenziójának elemszámát kapjuk meg.

Hibajelzés: Ha a tömböt nem definiáljuk, a 20032 sz. hibaüzenetet kapjuk.

SOUND parancs üzemmód, program üzemmód zenei utasítás

Önmagában álló SOUND utasítás hatására a gép egy normál C hangot ad ki.

Ha zenei utasításként kívánjuk használni, akkor a hang paramétereit is meg kell adni. Ezt a Felhasználói kézikönyv részletesen közli, itt még a 4. fejezetben szólnunk róla.

Hibajelzés: Nincs.

SOUND BUFFER parancs üzemmód, program üzemmód zeneifeltétel-beállító utasítás

SET SOUND BUFFER aritm. kif.

A "SOUND:" eszköz hangpufferének méretét lehet beállítani vele. Az aritmikai kifejezés értéke 1--255.

Példa:

```
100 ASK SOUND BUFFER a
```

```
110 PRINT a
```

Ekkor az a értékére 20-at kapunk. Ez az alapértelmezés.

Hibajelzés: Nincs.

SOUND STYLE parancs üzemmód, program üzemmód zeneifeltétel-beállító utasítás.

SOUND STYLE aritm. kif.

AZ utasítás beállítja, hogy a hang milyen módon szólaljon meg,

ez az aritmetikai kifejezés értékétől függ. A különböző paraméterek a Felhasználói kézikönyvben részletesen megtalálhatók (4. fejezet).

SPEAKER parancs üzemmód, program üzemmód feltételbeállító utasítás

SET SPEAKER ON/OFF

Az utasítással a beépített hangszórót kapcsolhatjuk be-, ill. ki. Átkapcsolni a TOGGLE utasítással lehet, ez az alapértelmezése a SHIFT+F7 billentyűnek.

Példa:

```
SET SPEAKER OFF
```

Ezzel kikapcsoltuk a belső hangszórót. Ha most kiadunk egy SOUND utasítást, semmit sem fogunk hallani.

Hibajelzés: Ha szintaktikailag helyes, nincs.

SPEEK parancs üzemmód, program üzemmód függvény  
SPEEK(szegmencím,aritm. kif.)

A számítógép Z80 A processzora a belső tárból csak 64 K-t tud címezni és olvasni. Mivel a gép memóriája 128 K-s, ezt úgy tudja olvasni, hogy 16 K-s lapokra osztja fel és lapozza. Az ezen a tárcímen levő információkat olvashatjuk ezzel az utasítással. A szegmensek alapkiépítettségben 0--7. terjednek, a nagyobb címeken 255-öt ad vissza. Ha a kifejezés nagyobb 16383-nál, akkor újra kezdi a számolást, tehát a 16384 a 0-ásnak felel meg.

Példa:

```
100 LET Y=4
110 LET N=13757
120 X=SPEEK(Y,N)
130 FOR I=0 TO X
140 LET N=N+1
150 A=SPEEK(Y,N)
160 PRINT CHR$(A),
```

```
170 NEXT I
180 LET N=N+1
190 PRINT
200 GOTO 120
```

A példaprogram kiíratja a memóriából a BASIC utasításokat.

Hibajelzés: Ha az aritmetikai kifejezésben túl nagy értéket adunk meg.

SPOKE parancs üzemmód, program üzemmód gépi kódú utasítás  
SPOKE szegmenscím, aritm. kif.(1), aritm. kif.(2).

Gyakorlatilag ugyanaz, mint a POKE utasítás. A megadott szegmens memóriacímére helyezi el a (2) aritmetikai kifejezésben lévő értéket. Az aritmetikai kifejezés értéke 0--255.

Példa:

```
SPOKE 0,25,65
```

A 65-ös értéket a 0 szegmens 25-ös címére teszi.

Hibajelzés: Gyakorlatilag nincs, de csak akkor használjuk, ha tisztában vagyunk a számítógép tárkiosztásával.

SQR parancs üzemmód, program üzemmód numerikus függvény  
SQR(aritm. kif.)

Az utasítás kiszámítja az aritmetikai kifejezés négyzetgyökét.

Példák:

```
a) PRINT SQR(2) : REM 1.414213562
```

```
b) 100 CLEAR SCREEN
```

```
110 INPUT PROMPT "A=":A
```

```
120 INPUT PROMPT "B=":B
```

```
130 C=SQR(A*A+B*B)
```

```
140 PRINT "C="C
```

```
150 GOTO 110
```

a) példa: kiszámítjuk a négyzetgyök 2-t.

b) példa: egy derékszögű háromszög c oldalát (átfogót) számítsuk ki a befogók ismeretében. Ha be akarjuk fejezni a programot, nyomjuk le a STOP billentyűt.

Hibajelzés: Az aritmetikai kifejezés kiértékelése közben. Az argumentum értéke nem lehet negatív.

START csak parancs üzemmódban használható vezérlőutasítás

Ha van a tárban program, akkor azt indítja el, ha nincs, akkor betölti a 106-os csatornáról az első programot és azt indítja el. Az F1 funkcióbillentyű alapértelmezése is ez.

STATUS parancs üzemmód, program üzemmód gépifeltétel-beállító utasítás

SET STATUS ON/OFF

Az utasítással be, ill. ki lehet kapcsolni a képernyőn lévő állapotjelző sort.

Példa:

SET STATUS OFF

Hatására el fog tűnni a képernyő legfelső sora.

Hibajelzés: Nincs.

STOP parancs üzemmód, program üzemmód programutasítás

Az utasítás az aktuális program futását megállítja, és kiírja, hogy hányadik sorban állt meg. A programot a CONTINUE parancssal lehet folytatni. Programfejlesztés közben érdemes használni, így tördelhetjük a programot. A változók értékét a PRINT utasítással lehet lekérdezni. Ha javítunk vagy szerkesztünk a programban, akkor már nem futtathatjuk a CONTINUE parancssal a programot.

Hibajelzés: Nincs.

STRING csak program üzemmódban használható utasítás  
STRING\*konstans sztring változó (tömbszám)\*hossz

Az utasítással sztring változókat és tömböket hozhatunk létre. Az első konstanssal az összes itt létrehozott változó hosszát adhatjuk meg, a hosszkonstanssal csak annak a változónak a hosszát, amelyik után áll. Minden elhagyható a változó nevének kívül, és akkor az alapértelmezés 132 karakter hosszúságot ad meg.

Példa:

```
100 STRING*12 bab$,KUK$(15),FEME*100
```

A bab\$ 1 darab 12 elemes változót, a KUK\$ 15 darab 12 elemes, a FEME 1 darab 100-elemes változót hozott létre.

Hibajelzés: Ha szintaktikailag helyes, nincs.

STR\$ parancs üzemmód, program üzemmód sztring függvény  
STR\$(aritm. kif.)

A megadott aritmetikai kifejezés értékének sztring alakját adja meg.

Példa:

```
100 LET A$=STR$(0.3)  
110 PRINT A$
```

Hibajelzés: Ha az aritmetikai kifejezés sztringeket is tartalmaz.

TAB parancs üzemmód, program üzemmód pozícionáló függvény  
TAB(aritm. kif.)

Az utasítás csak a PRINT utasításban szerepelhet. Az aritmetikai kifejezésben megadott értékre helyezi a kurzort és onnan kezdi a kiíratást. Vigyázzunk, ha a kiíratott szó hosszabb,

mint a felhasználható hely, akkor a számítógép szerkesztője a következő sorba írja ki a szöveget!

Példák:

- a) PRINT TAB(38) "BABA"
- b) PRINT TAB(35) "BABA"
- c) PRINT TAB(18) "BABA"

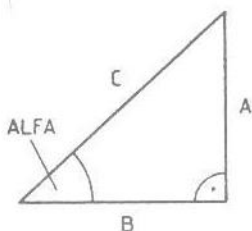
A példák a TAB hatását szemléltetik.

Hibajelzés: Ha az aritmetikai kifejezés értéke nem esik a 0--40 intervallumba.

TAN parancs üzemmód, program üzemmód trigonometrikus függvény  
TAN(aritm. kif.)

Alapértelmezésben a radiánban megadott aritmetikai kifejezés tangensét adja meg. Értelmezése az ábrán látható.

$$\text{TAN}(\text{ALFA}) = A/B$$



Példák:

- a) PRINT TAN(PI/2) :REM 1E11
- b) 100 OPTION ANGLE DEGREES  
110 INPUT PROMPT "ALFA=":ALFA  
120 INPUT PROMPT "B oldal=":B  
130 LET A=B\*TAN(ALFA)  
140 PRINT "A oldal=" A  
150 GOTO 110

a) példa: a TAN hatását láthatjuk. A számítógép PI/2-nél sem ad hibaüzenetet, habár itt a függvény a végtelenhez tart.

b) példa: kiszámítja egy derékszögű háromszög A oldalát az ALFA szög és a B oldal ismeretében.

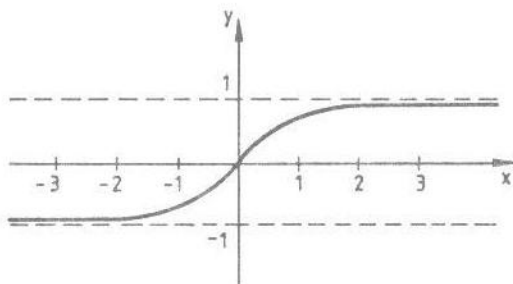
Hibajelzés: Az aritmetikai kifejezés kiértékelésekor lehetséges, ha túl nagy szögértéket adunk meg. A használható intervallum  $-3283$ -tól  $+3193$  fokig terjedhet, de  $-90$  foknál a  $3001$  számú hibajelzést kapjuk a  $TAN(x)=\sin(x)/\cos(x)$  miatt. A többi pontban a gép számítási hibája miatt a  $\cos(x)$  sohasem nulla.

TANH parancs üzemmód, program üzemmód trigonometrikus függvény

TANH(aritm. kif.)

Az utasítás kiszámítja az aritmetikai kifejezés tangens hiperbolikusát. Az értelmezése az ábrán látható.

$Y=TANH(x)$



Példa:

```
100 CLEAR SCREEN
110 FOR X=-3 TO 3 STEP .25
120 LET Y=TANH(X)
130 PRINT "X=" X TAB(10) "Y=" ROUND(Y,4)
140 NEXT X
```

A példa kiszámítja X tangens hiperbolikusát a  $-3...+3$  intervallumban.

Hibajelzés: Ha az aritmetikai kifejezés nem megfelelő (túl nagy, vagy nincs értéke).



TAPE LEVEL parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET TAPE LEVEL konstans

Az utasítással a magnetofon kimeneti jelszintjét lehet beállítani. A konstans értéke 1--255-ig terjedhet (6--255-ig a jelszint már nem változik). Az alapértelmezése 2 (ez 40 mV-nak felel meg). Az ASK utasítással lehet lekérdezni.

Példa:

```
SET TAPE LEVEL 6      :REM 6-os erősségű lesz a jelszint
ASK TAPE LEVEL K      :REM A beállított értéket kapjuk
PRINT K
```

Hibajelzés: Ha a konstans értéke negatív, vagy túl nagy.

TAPE SOUND parancs üzemmód, program üzemmód gépifeltétel-  
beállító utasítás

SET TAPE SOUND ON/OFF

Az IN bemenetre érkező jelek beépített hangszóróra való vezérlését kapcsolja ki-, ill. be. A TOGGLE utasítás használható az átkapcsolásra.

Példa:

```
TOGGLE TAPE SOUND
```

Ha a jel a hangszóróra ment, akkor ezt kikapcsolja.

TEXT parancs üzemmód, program üzemmód képernyőkezelő utasítás

```
TEXT
TEXT 40
TEXT 80
```

Az önálló TEXT utasítás a teljes képernyőre nyit meg egy szöveges lapot az előző 40 vagy 80 karakteres állapotnak megfelelően. Törli a képernyőt, lezárja a grafikus és az előző szöveges

lapokat. A 40 vagy a 80 paraméterrel az egy sorban használható karakterek számát adhatjuk meg.

Hibajelzés: Nincs, de elveszhet a képernyő tartalma.

### THEN

Csak az IF utasításban használható utasítás.

TIME parancs üzemmód, program üzemmód TIME "ÓÓ:PP:MM"

A gép belső óráját lehet beállítani vele. A megadás módja óra:perc:másodperc. Ha a dátumot is beállítottuk, éjféltkor automatikusan lépteti.

### Példa:

```
TIME"15:00:00" :REM 15 órát állítottunk be
```

Hibajelzés: Ha nem megfelelő időt állítunk be (pl. 36 óra 45 perc), akkor a 9235 számú hibaüzenetet kapjuk.

TIMER parancs üzemmód, program üzemmód megszakításgeneráló utasítás

SET TIMER konstans

A konstans értékének megfelelő másodperc múlva a program megszakítást hoz létre. A konstans értéke 0--255. A megszakítás olyan, mintha a STOP billentyűt nyomtuk volna meg.

### Példa:

```
100 LET X=0
110 SET TIMER 50
120 LET X=X+1:GOTO 120
```

A példában 50 másodperc múlva áll le a végtelenített ciklus. Írassuk ki az X értékét, és látható, hogy 9916-ig számolt el a számítógép. Különböző számítások gyorsaságának lemérésére is használható.

Hibajelzés: Ha a konstans értéke nem lépi túl a 255-öt, nincs.

Az aktuális időt lehet lekérdezni vele. A lekérdezett idő formája óra:perc:másodperc.

Példa:

```
PRINT TIME$ :REM az aktuális időt kapjuk meg
```

Hibajelzés: Gyakorlatilag nincs.

TRACE parancs üzemmód, program üzemmód segédutasítás

TRACE ON TO # vagy £ csatornaszám

TRACE OFF

Az utasítással az aktuális programban a nyomkövetést lehet bekapcsolni. Az éppen végrehajtott program sorszámát írja ki a megadott csatornára. Ha elhagyjuk a csatornaszámot, akkor a rendszercsatornára (0) írja ki a sorszámot.

Példa:

```
100 TRACE ON
```

```
110 REM bármelyik példaprogram
```

A példával nyomon követhetjük a végrehajtás sorrendjét. A példaprogramoknál és a tanulásakor lehet hasznos, hogy követhessük a végrehajtás folyamatát.

Hibajelzés: Gyakorlatilag nincs.

TRUNCATE parancs üzemmód, program üzemmód numerikus függvény

TRUNCATE(aritm. kif.,konstans)

Az utasítás az aritmetikai kifejezésből a konstansban megadott számú decimális értéket vág le. Gyakorlatilag csonkítja a kifejezést.

Példák:

```
a) 100 A=TRUNCATE(98.384,2)
```

```
110 PRINT ,A :REM 98.3
```

```
b) PRINT TRUNCATE(384568,5) :REM 3
```

A példák önmagukért beszélnek, csontkítják a megadott értékeket.

Hibajelzés: Nincs, de az értékünk el is veszhet, ha több pozíciót vágatunk le, mint ahány pozíció az aritmetikai kifejezés értéke.

TYPE parancs üzemmód, program üzemmód segédprogram-aktívizáló utasítás

Az utasítással a BASIC-ből a szövegszerkesztőt hívhatjuk meg, de ahhoz, hogy kilépjünk a BASIC-ből, meg kell erősíteni a parancsunkat. Törli az összes programot és a változókat a tárból.

Példa:

```
TYPE
```

Az ENTER leütése után belépünk a szövegszerkesztőbe.

Hibajelzés: Gyakorlatilag nincs, de ha visszatérünk a BASIC-be, olyan, mintha újraindítottuk volna a gépet.

UBOUND parancs üzemmód, program üzemmód numerikus függvény  
UBOUND(tömbvált.)

UBOUND(tömbvált.,dimenziószám)

A tömbváltozóval definiált egydimenziós tömb felső indexhatárát lehet lekérdezni vele. Ha a tömb többdimenziós, akkor a dimenziószámában megadott dimenzió felső indexhatárát lehet lekérdezni.

Példa:

```
100 DIM a(10)
```

```
110 REM a program további része
```

Ha lefuttatjuk a programot, akkor a

```
PRINT UBOUND(a)
```

 utasítással lekérdezhethetjük a felső indexhatárt.

Hibajelzés: Ha a tömb nincs definiálva.

UCASE\$ parancs üzemmód, program üzemmód sztring függvény  
UCASE\$(sztring kif.)

A sztring kifejezésben levő kisbetűket nagybetűkké alakítja át.

Példa:

```
110 A$="mama,papa,baba"
```

```
120 PRINT UCASE$(A$)
```

A példa önmagát mutatja be.

Hibajelzés: Ha nem sztring kifejezést adunk meg az argumentumban.

USR parancs üzemmód, program üzemmód hívóutasítás  
USR(rutinnév, konstans)

Az utasítás a megadott nevű gépi kódú programot (rutint) hívja meg, és a konstans értékét átadja a rutinnak feldolgozásra. Ez a fajta függvényhívás akkor használható, ha paramétert akarunk átadni és visszakapni a gépi kódú rutintól.

Példa: 1. a Felhasználói kézikönyvben a gépi kód használatánál.

Hibajelzés: Ha nem létezik a meghívott gépi kódú rutin.

VAL parancs üzemmód, program üzemmód sztring függvény  
VAL(sztring kif.)

Az utasításban megadott sztring kifejezést alakítja át aritmetikai kifejezéssé.

Példa:

```
PRINT VAL("12BABA")
```

A példában a használatát mutatjuk be. Csak az első alfanumerikus karakterig dolgozza fel a sztringet.

Hibajelzés: Gyakorlatilag nincs.

VERIFY csak parancs üzemmódban használható ellenőrző utasítás

VERIFY,# vagy  $\dot{\epsilon}$  csatornaszám: fájlnev

Az utasítás ellenőrzi, hogy a valamelyik perifériára elmentett program megegyezik-e a memóriában tárolt aktuális programmal. Alapértelmezésben a 106-os csatornát használja. Ha a két program nem egyezik, hibajelzést kapunk.

Példa:

```
SAVE "proba"           :REM visszatekerceselni  
VERIFY "proba"
```

Ha elmentjük a programot és visszatekerceseljük a szalagot a VERIFY kiadása előtt, akkor hatására a számítógép beolvassa a "proba" nevű programot, és összehasonlítja az aktuális programmal.

Hibajelzés: Nincs.

VIDEO COLOUR parancs üzemmód, program üzemmód grafikus utasítás

SET VIDEO COLOUR aritm. kif.

Az utasítással az ezután megnyitandó videólapok színelbontását lehet megadni. Az aritmetikai kifejezés értéke 0--3-ig terjedhet. Jelentésük a GRAPHICS HIRES utasításnál a következő:

- 0 - 2 szín , vízszintesen 640 képpont,
- 1 - 4 szín , vízszintesen 320 képpont,
- 2 - 16 szín , vízszintesen 160 képpont,
- 3 -256 szín , vízszintesen 80 képpont.

Ez azt jelenti, hogy vízszintesen ennyi képpont és szín használható. A LORES üzemmódban a színszámok nem változnak, de a képpontok száma felére csökken. A függőlegesen használható képpontok száma nem változik a különböző módokban. Használata akkor szükséges, ha több videólapot akarunk használni.

Példa: 1. a LINE STYLE utasításnál.

Hibajelzés: Nincs.

VIDEO MODE parancs üzemmód, program üzemmód grafikus utasítás

SET VIDEO MODE aritm. kif.

Az utasítással az ezután megnyitandó videólapokon lehet beállítani a videó üzemmódokat. Az aritmetikai kifejezések lehetséges értékei a Felhasználói kézikönyvben megtalálhatók.

Példa: 1. a LINE STYLE utasításnál.

Hibajelzés: Ha nincs értelmezve az aritmetikai kifejezés.

VIDEO parancs üzemmód, program üzemmód grafikus utasítás

VIDEO X

VIDEO Y

Az utasítással az ezután megnyitandó videólapokon lehet a vízszintes (x) és a függőleges (y) méretet meghatározni. A méretet karakterekben adhatjuk meg.

Példa: 1. a LINE STYLE utasításnál.

Hibajelzés: Ha vízszintesen a 42-es és függőlegesen a 255-ös határt túllépjük.

WAIT DELAY csak program üzemmódban használható segédutasítás

WAIT DELAY aritm. kif.

AZ utasítás az aritmetikai kifejezésben megadott másodpercig felfüggeszti a program végrehajtását. A maximális idő 32767 másodperc lehet.

Példa:

```
100 CLEAR SCREEN
110 WAIT DELAY 60
120 PRINT , "Most járt le egy perc!"
```

A példa egy percre felfüggeszti a program végrehajtását.

Hibajelzés: Ha az aritmetikai kifejezés negatív vagy kiértékelhetetlen.

```
WHEN csak program üzemmódban használható megszakításkezelő
utasítás
WHEN EXCEPTION USE handlernév
.
.
END WHEN
```

A blokkon belüli hibákat, megszakításokat kezeli le az utasítás.

Példa: 1. Felhasználói kézikönyv 109. old.

Hibajelzés: Ha nem létezik a megadott nevű hibakezelő rutin.

```
WHITE parancs üzemmód, program üzemmód szín-segédfüggvény
SET utasítás WHITE
```

Önmagában nem használható utasítás, a fehér színt állítja be (azonos RGB(1,1,1)).

Példa: 1. az RGB függvénynél.

Hibajelzés: Ha az utasítás többi része jó, akkor nincs.

```
WORD$ parancs üzemmód, program üzemmód csatorna-kezelő utasítás
WORD$(aritm. kif.)
```

Az utasítás az aritmetikai kifejezésnek megfelelő kétkaraktert ad vissza. Az aritmetikai kifejezésnek 2 bájt értékűnek kell lennie.



Példa:

```
100 LET A$=WORD$(20561)
110 REM 20561=BIN(01010000001010001)
120 PRINT A$
```

A példa a QP karaktereket írja ki a képernyőre.

Hibajelzés: Ha az aritmetikai kifejezés értéke nem 2 bájt.

YELLOW parancs üzemmód, program üzemmód szín-segédfüggvény  
SET utasítás YELLOW

Önmagában nem használható, a sárga színt állítja be (azonos RGB(1,1,0)).

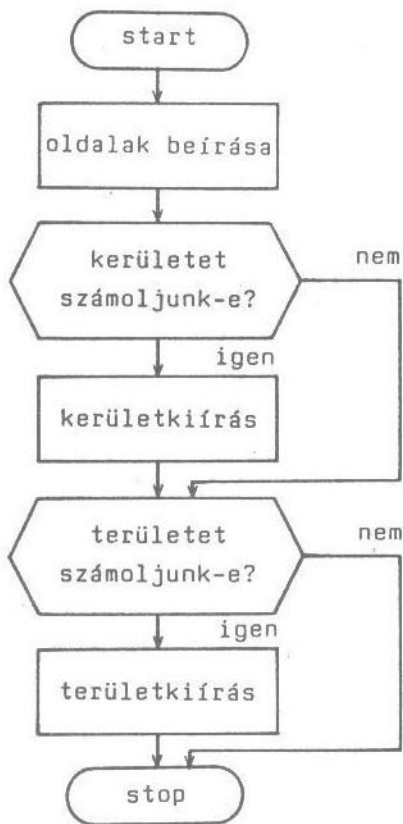
Példa: 1. az RGB függvénynél.

Hibajelzés: Ha az utasítás többi része jó, akkor nincs.

### BASIC programok készítése

A vásárolt vagy kazettán kapott programokat már tudjuk használni (a jól beállított magnóról az F1 lenyomására indul a program). Ismerjük a BASIC utasításait, és a gépben lévő BASIC programot -- az F2 lenyomásával -- ki is tudjuk listázni. Ezek után írjunk magunk programot! Az egyszerűség kedvéért a már ismert, a téglalap kerületét és területét kiszámító programot fogjuk BASIC nyelven elkészíteni.

Először is rajzoljuk meg az utolsó programváltozat folyamatábráját, majd egészítsük ki úgy, hogy a képernyő előtt ülő felhasználót kérjük meg, válasszon: a program a kerületet, a területet vagy mindkettőt számolja-e. A feladat megfogalmazása -- az ún. programszűkítés -- így még nem egyértelmű, mi az egyik lehetséges megoldást rajzoljuk fel (8. ábra).



8. ábra. Folyamatábra

A folyamatábrában két eldöntendő kérdést teszünk fel:

1. számoljunk-e kerületet?
2. számoljunk-e területet?

Ha valaki elindítja a programot, akkor előfordulhat, hogy csak a területet akarja kiszámolni -- amit a második választással tud csak kérni. Tegyük lehetővé, hogy a programot már az első válasszal is a területszámításra irányíthassuk! Programunk például legyen a következő:

- 1 Input a,b
- 2  $K=2 \times (a+b)$

```

3 INPUT C$
4 IF C$<>"k" THEN GOTO 6
5 PRINT K
6 IF C$ = "t" THEN 9
7 INPUT C$
8 IF C$<>"t" THEN 11
9 t=a*b
10 PRINT T
11 END

```

A bebillentyűzést az automatikus sorszámozás (F3 funkcióbillentyű) bekapcsolása segíti, így az első utasításunk a 100-as sorszámot kapja. Az utasítás sorszámára való hivatkozásoknál ezt figyelembe kell venni (például GOTO 6 helyett GOTO 150). Az 5. sor beírása után a LIST paranccsal kírátva az addigiakat, a következőket látjuk:

```

100 INPUT A,B
110 LET K=2*(A+B)
120 INPUT C$
130 IF C$<>"k" THEN GOTO 150
140 PRINT K

```

Feltűnő, hogy minden változó nagybetűs, de a feltételben az idézőjelek közötti betű olyan maradt, mint amilyen volt. A gép mindig kiírja a LET (legyen) szót is, akkor is, ha nem írtuk be. Ezután kipróbálhatunk néhány, az ENTERPRISE BASIC rendszerében is megengedett fogást.

-- Utasítás ismétlése. Tudjuk, hogy a 170-es sor meg fog egyezni a 120-as sorral, ezért írjuk át a sorszámot 160-ra és az ENTER billentyűvel zárjuk le a sort. Eredménye: a 120-as sor utasítása megismétlődik 160-as sorszámmal (erről az F2 leütésével elindított ismételt listázással győződhetünk meg).

-- Utasítások módosítása, javítása. A program 130., 150. és 170. utasítása csak egy-két betűvel tér el egymástól. A 130. utasítást a sorszám átírásával először ismételjük meg 150-es és 170-es sorszámmal, majd újra listázva már az új sorokat írjuk át.

-- Átsorszámzás. A programok sorszámozása -- a bebillentyűzésnél elkövetett tévedések és javításaik következtében -- rendszerint már nem egyenletes. Egyenletessé tehetjük a RENUMBER felíratú (SHIFT F3) funkcióbillentyűvel. A RENUMBER paranccsal a programunkban helyet tudunk csinálni újabb utasításoknak. Például, ha programunk első 5 sora után az utolsó hármát is beírjuk (automatikus sorszámozással), akkor a következő paranccsal tehetjük a helyére a befejező utasításokat:

```
RENUMBER 150 TO 170 AT 190
```

Ezt követően az új sorokat a végleges sorszámukat kiszámolva azonnal helyükre tehetjük. Ha csak az egymást követő sorrendjükre ügyelünk és a sorszámukra nem, a befejező RENUMBER paranccsal csinálhatunk rendet. Ez azt jelenti, hogy 100-tól kezdődően tízesével folyamatosan növekvő sorszámmal látjuk el a programsorokat.

Programunkkal még nem lehetünk elégedettek. Az adatok bekérése és kiírása még nem elég kényelmes (az sem lehet mentségünk, hogy néhány "szoftvertermék" hasonló színvonalú "szolgáltatást" nyújt). Mi most csupán az ENTERPRISE programírást segítő tulajdonságaival ismerkedünk, és nem kényelmesen használható programot akarunk írni.

-- Programok tördelése. Az F2 funkcióbillentyűvel vagy a LIST paranccsal kiírt programok érdekessége, hogy az összetartozó utasításpárok a sorok azonos oszlopában kezdődnek, míg a közrefogott utasítások 2 betűhellyel beljebb jelennek meg a képernyőn. A tördelés kitűnő lehetőséget ad arra, hogy a programírás egyik jellegzetes és gyakran előforduló hibáját kiszűrjük. Ugyanis egy utasításpár egyikének kihagyása a kiírásban szembe-tűnő változást okoz. A tördelés bemutatásaként a példaprogramunkat felkészítjük az egymás után többszöri számítás elvégzésére is.

```
90 DO
100 INPUT A,B
110 LET K=2*(A+B)
120 INPUT C$
```

```

130 IF C$<>"k" THEN GOTO 150
140 PRINT K
200 LOOP WHILE K<>0
210 END

```

Az ismétlődő futtatások során a képernyőn megjelenő számok közötti eligazodást mér segíteni kell. Ezt a célt szolgálják a következő programban az üzenetek.

```

100 DO
110 PRINT "a oldal",
120 INPUT A
130 PRINT "b oldal",
140 INPUT B
150 LET K=2*(a+b)
160 PRINT "(k)erület vagy (t)erület",
170 INPUT C$
180 IF C$<>"k" THEN GOTO 200
190 PRINT "kerület" K
200 IF C$ = "t" THEN GOTO 240
210 PRINT "(t)erület",
220 INPUT C$
230 IF C$<>"t" THEN GOTO 160
240 LET T=A*B
250 PRINT "terület:" T
260 LOOP WHILE K<>0
270 END

```

(Az ENTERPRISE BASIC nyelvében az üzenetkiíró PRINT és a választ beolvasó INPUT utasításokat az INPUT utasítás PROMT kiegészítésével össze is vonhatjuk egy utasítássá.)

Programunk a tördelésen kívül a ciklusok egyik veszélyes alkalmazását is bemutatja. A  $K \neq 0$  feltétel ugyanis minden értelmes esetben igaz, így kerületszámítási programunk minden értelmes esetben újraindul. Most ez nem baj, a STOP billentyűvel vagy az

Egyik szám hibás megadásával a ciklusból kiléphetünk (sőt, ha mindkét oldal értékére 0-t írunk, akkor leáll a ciklus is).

A programozók egyik réme, a végtelen ciklus, esetünkben jó szolgálatot tesz, de ha nem szándékosan állítjuk elő, akkor legszebb terveinket is kíméletlenül elrontja.

Következő példánkban két egymást követő ciklusban egy vektor (egydimenziós tömb) elemeinek értékeit adunk, majd kiíratjuk a vektor elemeinek értékét. A LIST parancs tördelésével -- ha jól írjuk be az utasítárpárokat -- programunk szinte ketté válik. A program listája és futási eredménye a következő:

```
100 DIM A(1 TO 20)
110 LET J=0
120 FOR I=1 TO 6 STEP .33
130   LET J=J+1
140   LET A(I)=J
150   LET A(J)=0
160 NEXT I
170 LET J=0
180 FOR I=1 TO 6 STEP .33
190   LET J=J+1
200   PRINT I,A(I),J,A(J)
210 NEXT I
```

Ebből a programból a 160-as sor kihagyása látványos változást okoz a listán. Az utána következő sorok két betűhellyel beljebb kezdődnek, így a 210-es sorban lévő NEXT I sem az első helyen kezdődik.

Programhibák elkerülése érdekében szólnunk kell a futtatás eredményeiről is.

```
1      4      1      4
1 . 33  4      2      7
1 . 66  4      3     10
1 . 99  4      4     13
2 . 32  7      5     16
2 . 65  7      6      0
```

2 . 98	7	7	0
3 . 31	10	8	0
3 . 64	10	9	0
3 . 97	10	10	0
4 . 3	13	11	0
4 . 63	13	12	0
4 . 96	13	13	0
5 . 29	16	14	0
5 . 62	16	15	0
5 . 95	16	16	0

OK

Az első oszlop egy indexként használt változó (az I) értékeit, a második az első számmal mint indexeszel megadott A tömbelem értékét mutatja. Az utolsó két oszlop pedig az A tömb elemeinek indexét és értékét mutatja. Láthatjuk, hogy az index értéke az indexben lévő változó egészrészre, ezért például az A(1), A(1,33),...,A(1,99) egyaránt az első tömbelemet jelenti, amelynek értéke 4. A kiírásból az is látszik, hogy a FOR ciklus az első, az utasításban megadott határnál nagyobb értékre már nem hajtja végre a cikluson belüli (190-es és 200-as) utasításokat. Az ilyen "apróságok" elsősorban más gépekről áthozott programok futásánál okozhatnak bajt, vannak ugyanis olyan BASIC értelmezők, amelyek például az indexben lévő számértéket a legközelebbi egészértékkel helyettesítik.

-- Programok összeállítása. Ha kész programjainkat vagy a használható programrészeket megőrzésre kazettára írtuk, akkor ezeket új programok készítésénél is felhasználhatjuk. Egy-egy program kazettára írása a SAVE paranccsal, beolvasása pedig a LOAD paranccsal könnyen megoldható. Ha több programot akarunk egymás után a tárba olvasni, akkor a MERGE parancsot kell használnunk. Így a LOAD paranccsal ellentétben a tárban lévő programot nem töröljük, kivéve azokat a sorokat, amelyek sorszáma a kazettáról olvasott programban is előfordul. Ezeket az új sorral átírjuk. Ha a kazettára írás előtt a programjainkat a SHIFT F3 billentyűvel újrasorszámozuk, akkor számíthatunk arra, hogy

a kazettáról érkező programrésznél mindig 100-as sorszámú sorral kezdődik. Ha a tárban lévőt az újabb programrész betöltése előtt átsorszámozzuk, például 7000-től egyesével, akkor kb. 700 soros, általunk felírt programot biztonságosan olvashatunk be a kazettáról anélkül, hogy a tárban lévőt megváltoztatnánk. BASIC nyelvű programjaink készítését, futtatását a géppel kapott lehetőségekkel már meg tudjuk oldani. Később ezeket a tapasztalataink alapján magunk is bővíthetjük, kényelmesíthetjük. A következőkben a BASIC nyelv utasításai segítségével megvalósítható gépi kódú programozással és az assembly nyelvvel foglalkozunk.

### Assembly nyelvű programozás

Az ENTERPRISE jelenleg kapható programjai az assembly nyelvű programozást kevés segédeszközzel támogatják. Csupán a gép BASIC nyelvének néhány utasítása teszi lehetővé az assembly nyelvű, pontosabban a gépi kódú programozást. Az assembly nyelvet igazán használhatóvá azonban egy fordítóprogram, egy assembler teszi. Várható, hogy elkészül és kapható lesz ilyen program az ENTERPRISE-hoz is, amivel számos, más gépre kidolgozott programot lehet majd a gépre átírni. A Z80 A ugyanis elterjedt, sok más gépbe is beépített processzor, amelyet a korábban közkedvelt Intel 8080 mikroprocesszor utasításai nagyon kevés változtatással működtethetnek. Az évek során összegyűlt programok elég értékesek ahhoz, hogy már most gondoljunk a felhasználásukra.

Az assembly nyelv utasításai megfelelnek a gépi kódú utasításoknak, csupán a programírást könnyíti meg a programozók számára. Az utasítások elnevezése az első könnyítés, ugyanis az utasításkód nem számjegy, hanem ún. mnemonik. Ez angol nyelvből származó rövidítés, tehát jobban emlékeztet az utasítás hatására. A másik jelentős könnyítés, hogy a gép regisztereit és tárcímeit a programban névvel lehet megadni. Az assembler az ezekkel a könnyítésekkel megírt programot alakítja át a géppel végrehajtható számsorozattá. Amíg nincs fordítóprogramunk, addig ezt az átalakítást nekünk kell elvégeznünk.



Az assembly programok készítése során az utasítások gépi kódá alakítását a függelékében található, ábécésorrendben felsorolt utasítások táblázata segíti. Ha a programlistát egy assembler már kiírta, akkor rendszerint ezt az átkódolást is megtaláljuk a programlistán.

A tárban lévő programok felismerését és értelmezését a kódszámok növekvő sorrendjében felsorolt utasítások listája könnyíti meg. A gépi kód assembly utasításokká való alakítására is készítették már programokat. Ezeket a visszafordító programokat disassemblereknek nevezik.

Az assembly utasításokat a következőkben a végrehajtott művelet típusa szerint csoportosítva mutatjuk be. Leírjuk az utasítás nevét, az utasítás végrehajtása után kialakult állapotot leíró állapotbitek megváltozását és az utasítás szöveges értelmezését. Az állapotbitek és jelölésük a következő:

- CY átvitel (carry); értéke 1, ha a művelet eredményében a legnagyobb helyértékű bitnél nagyobb helyértékű is keletkezett.
- N kivonás (subtract); értéke 1, ha a művelet kivonás volt.
- P vagy V paritás vagy túlcscordulás (parity vagy overflow); értéke 1, ha az eredmény páratlan vagy túlcscordult. A állapotbit jelentése a végrehajtott utasítástól függ, logikai műveletek után paritást, aritmetikai műveletek után túlcscordulást jelöl.
- H segédátvitel (half carry); értéke 1, ha a két félbájt között az eredményben átvitel van.
- Z zérus (zero); értéke 1, ha a művelet eredménye nulla.
- S előjel (sign); értéke 1, ha a legnagyobb helyértékű bit értéke 1.

A feltételes utasításokban akkor tekintjük a feltételt igaznak, azaz a feltétel akkor teljesül, ha a megfelelő állapotbit értéke 1.

Az utasítások ismertetésében az állapotbitek leírására a következő megállapodásokat alkalmaztuk:

- Csak az állapotbit nevét adjuk meg, ha a bit értéke az utasítás eredményétől függ.

Kérdőjellel adjuk meg az állapotbitet, ha a megváltozása nem függ az utasítástól.

-- Értékével együtt adjuk meg az állapotbitet, ha a megváltozása nem függ az utasítás eredményétől.

-- Nem írjuk ki az állapotbit nevét, ha az utasítás nem változtatja meg a bit értékét.

Az ismertetést Krizsán György: ZILOG mikroprocesszor-családok című könyvében közölt leírás alapján készítettük. További jelölések:

b : bithely (0--7)

cc: feltételek (állapotbit-értékek)

d : címmódosítás (-128-- +127)

dd: regiszterpárok (BC, DE, HL, SP)

e : a relatív vezérlésátadás mértéke

m : a regiszterek és HL, IX+d, IY+d tartalma

n : konstans (0--255)

nn: konstans (0--65535)

pp: regiszterpárok (BC, DE, IX, SP)

qq: regiszterpárok (BC, DE, HL, AF)

rr: regiszterpárok (BC, DE, IY, SP)

s : m és n

ss: a regiszterpárok (BC, DE, HL, SP) tartalma

r : regiszterek (A, B, C, D, E, H, L)

i : az indexregiszterek bájtjai (XH, XL, YH, YL)

ih: az indexregiszterek magas helyértékű bájtjai

ii: indexregiszterek (IX, IY)

il: az indexregiszterek alacsony helyértékű bájtjai

## Z80 assembly utasítások

### Aritmetikai és logikai utasítások

ADC A,s

A:=A+s+CY

Az akkumulátor tartalmához hozzáadja az "s" operandust és a

CY állapotbit értékét, majd az eredményt az akkumulátorban helyezi el.

Állapotbitek: CY, N=0, V, H, Z, S.

ADC HL, ss

HL:=HL+ss+CY

A HL regiszterpár tartalmához hozzáadja az "ss" regiszterpár tartalmát és hozzáadja a CY állapotbit értékét, majd az eredményt a HL regiszterpárban tárolja.

Az állapotbitek a 16 bites eredmény legmagasabb helyértékű 8 bitje állítja be.

Állapotbitek: CY, N=0, V, H, Z, S.

ADD A,s

A:=A+s

Az akkumulátor tartalmához hozzáadja az "s"-sel definiált bájttartalmát, az eredményt az akkumulátorban helyezi el.

Állapotbitek: CY, N=0, V, H, Z, S.

ADD HL, ss

HL:=HL+ss

A HL regiszterpár tartalmához hozzáadja az "ss" regiszterpár tartalmát, majd az eredményt a HL regiszterpárban tárolja.

Az állapotbitek a 16 bites eredmény legmagasabb helyértékű 8 bitje állítja be.

ADD IX, pp

IX:=IX+pp

Az IX regiszterpár tartalmához hozzáadja az "pp" regiszterpár tartalmát, majd az eredményt az IX regiszterpárban tárolja.

Az állapotbitek a 16 bites eredmény legmagasabb helyértékű 8 bitje állítja be.

ADD IY, rr

IY:=IY+rr

Az IY regiszterpár tartalmához hozzáadja az "rr" regiszterpár tartalmát, majd az eredményt az IY regiszterpárban tárolja. Az állapotbiteket a 16 bites eredmény legmagasabb helyértékű 8 bitje állítja be.

CPL

A:=-A

Az akkumulátor tartalmát invertálja.

Állapotbitek: N=1, H=1.

DAA

A BCD számokkal végzett aritmetikai műveletek után az eredmény nem BCD szám. Ez az utasítás az eredményt a megfelelő szám hozzáadásával a helyes BCD eredménnyé alakítja.

Állapotbitek: CY, V, H, Z, S.

DEC IX

IX:=IX-1

Az IX regiszterpár tartalmát eggyel csökkenti. Az állapotbitek változatlanok.

DEC IY

IY:=IY-1

Az IY regiszterpár tartalmát eggyel csökkenti. Az állapotbitek változatlanok.

DEC m

m:=m-1

Az "m"-mel definiált bájtt értékét eggyel csökkenti.

Állapotbitek: V, H, Z, S.

DEC ss

ss:=ss-1

Az "ss" regiszterpár tartalmát eggyel csökkenti. Az állapotbitek változatlanok.

```
INC IX
  IX:=IX+1
```

Az IX regiszterpár tartalmát eggyel növeli. Az állapotbitek nem változnak.

```
INC IY
  IY:=IY+1
```

Az IY regiszterpár tartalmát eggyel növeli. Az állapotbitek nem változnak.

```
INC m
  m:=m+1
```

Az "m"-mel definiált bájt értékét eggyel növeli. A V bit értéke 1, ha az m legmagasabb helyértékű bitje 1.

Állapotbitek: N=0, V, H, Z, S.

```
INC ss
  ss:=ss+1
```

Az "ss" regiszterpár tartalmát eggyel növeli. Az állapotbitek nem változnak.

```
NEG
  A:=0-A
```

Az akkumulátor tartalmát negálja (kettes komplementessel), azaz kivonja nullából.

Állapotbitek: CY, N=1, V, H, Z, S.

```
OR s
  A:=A logikai VAGY s
```

Az akkumulátor és az "s" között logikai VAGY kapcsolatot hoz létre. Az eredményt az akkumulátorban tárolja.

Állapotbitek: CY=0, N=0, P, H=1, Z, S.

SBC A,s

A:=A-s-CY

Az akkumulátor tartalmából kivonja az "s" operandust és a CY állapotbit értékét, majd az eredményt az akkumulátorba tölti. CY értéke alulcsordulás esetén lesz 1.

Állapotbitek: CY, N=1, V, H, Z, S.

SBC HL, ss

HL:=HL-ss-CY

HL regiszterpár tartalmából kivonja az "ss"-sel jelzett regiszterpár tartalmát, majd kivonja az átvitel-állapotbit tartalmát. Az eredményt a HL-be tölti be. CY értéke alulcsordulás esetén lesz 1.

Állapotbitek: CY, N=1, V, H, Z, S.

SUB s

A:=A-s

Az akkumulátor tartalmából kivonja az "s" operandust, és az eredményt az akkumulátorba írja be. CY értéke alulcsordulás esetén lesz 1.

Állapotbitek: CY, N=1, V, H, Z, S.

AND s

A:=A logikai ÉS s

Az akkumulátor és az "s" között logikai ÉS kapcsolatot hoz létre. Az eredményt az akkumulátorban tárolja.

Állapotbitek: CY=0, N=0, V, H, Z, S.

BIT b, m

Z:=-mb

Az "m"-mel definiált bájt b-edik bitjének tartalmát invertálva a Z állapotbit helyére írja be.

Állapotbitek: CY, N=0, V=?, H, Z, S=?.

CCF

CY:=-CY

A CY állapotbitet invertálja. A H bit az előző CY értéket kapja.

Állapotbitek: CY, N=0, H.

CP s

A - s

Az akkumulátor tartalmából kivonja az "s" operandust. Az akkumulátor tartalma változatlan marad, az eredmény az állapotbitek beállításában jelentkezik.

Állapotbitek: CY, N=1, V, H, Z, S.

CPD

A - (HL)

BC:=BC-1

HL:=HL-1

A (HL)-lel címzett tárrekesz tartalmát kivonja az akkumulátor tartalmából, és ennek megfelelően állítja be az állapotbitekét. Az akkumulátor tartalma változatlan marad. A BC és HL regiszterpárok tartalmát eggyel csökkenti.

Állapotbitek: N=1, V, H, Z, S.

CPDR

A - (HL)

HL:=HL-1

BC:=BC-1

A (HL)-lel címzett tárrekesz tartalmát kivonja az akkumulátor tartalmából és ennek megfelelően állítja be az állapotbitekét.

Az akkumulátor tartalma változatlan marad. A BC és HL regiszterpárok tartalmát eggyel csökkenti. Ha a BC tartalma nullára csökken, vagy  $A = (HL)$ , akkor CPU a következő utasításra lép. Ellenkező esetben a PC tartalmát kettővel csökkenti, és ismétli az utasítást. Minden utasításismétlés utasításbeolvasást is jelent, vagyis ekkor van frissítőkiklus. Ha a megszakításkérés engedélyezett, akkor érvényesülhet is.

A V állapotbit alulcsordulást jelent.

Állapotbitek: N=1, V, H, Z, S.

#### CPI

A - (HL)

HL:=HL+1

BC:=BC-1

A (HL)-lel címzett tárrekesz tartalmát kivonja az akkumulátor tartalmából, és ennek megfelelően állítja be az állapotbiteket. Az akkumulátor tartalma változatlan marad. A BC regiszterpár tartalmát eggyel csökkenti, a HL regiszterpár tartalmát eggyel növeli.

A V állapotbit alulcsordulást jelent.

Állapotbitek: N=1, V, H, Z, S.

#### CPIR

A - (HL)

HL:=HL+1

BC:=BC-1

A (HL)-lel címzett tárrekesz tartalmát kivonja az akkumulátor tartalmából, és ennek megfelelően állítja be az állapotbiteket. Az akkumulátor tartalma változatlan marad. A BC regiszterpár tartalmát eggyel csökkenti, a HL regiszterpár tartalmát eggyel növeli.

Ha a BC tartalma nullára csökken, vagy  $A = (HL)$ , akkor a CPU a következő utasításra lép, ellenkező esetben a PC tartalmát csökkenti kettővel, és ismétli az utasítást. Minden utasításismétlés utasításbeolvasást is jelent, vagyis ekkor van frissítő-



ciklus. Ha a megszakításkérés engedélyezett, akkor érvényesülhet is.

A V állapotbit alulcsordulást jelent.

Állapotbitek: N=1, V, H, Z, S.

RL m

A CPU az "m"-mel definiált bájtot úgy tolja balra, hogy a 7-es bithely tartalma a C állapotbit helyére, és annak megelőző tartalma a 0-ás bithelyre kerül. A többi bithely tartalma eggyel balra tolódik.

Állapotbitek: CY, N=0, P, H, Z, S.

RLA

A CPU az akkumulátor tartalmát úgy tolja balra, hogy a 7-es bithely tartalma a C állapotbit helyére, és annak megelőző tartalma a 0-ás bithelyre kerül. A többi bithely tartalma eggyel balra tolódik.

Állapotbitek: CY, N=0, H=0.

RLCA

A CPU az akkumulátor tartalmát úgy tolja balra, hogy a 7-es bithely tartalma a C állapotbit helyére, valamint a 0-ás bithelyre kerül. A többi bithely tartalma eggyel balra tolódik.

Állapotbitek: CY, N=0, H=0.

RLC m

A CPU az "m"-mel definiált bájtot úgy tolja balra, hogy a 7-es bithely tartalma a C állapotbit helyére, valamint a 0-ás bithelyre kerül. A többi bithely tartalma eggyel balra tolódik.

Állapotbitek: CY, N=0, P, H=0, Z, S.

RLD

A HL regiszterpárral címzett tárrekesz alsó félbájtját átmásolja a felső félbájtba. Annak megelőző tartalma az akkumulátor

alsó félbájtjába kerül. Az akkumulátor megelőző tartalma a címzett tárrekesz alsó félbájtjába kerül.

Állapotbitek: CY, N=0, P, H=0, Z, S.

RR m

A CPU az "m"-mel definiált bájtot úgy tolja jobbra, hogy a 0-ás bithely tartalma a C állapotbit helyére kerül, annak megelőző tartalma pedig a 7-es bithelyre. A többi bithely tartalma egygyel jobbra tolódik.

Állapotbitek: CY, N=0, P, H=0, Z, S.

RRA

A CPU az akkumulátor tartalmát úgy tolja jobbra, hogy a 0-ás bithely tartalma a C állapotbit helyére, annak megelőző tartalma pedig a 7-es bithelyre kerül. A többi bithely tartalma egygyel jobbra tolódik.

Állapotbitek: CY, N=0, H=0.

RRCA

A CPU az akkumulátor tartalmát úgy tolja jobbra, hogy a 0-ás bithely tartalma a C állapotbit helyére és a 7-es bithelyre kerül. A többi bithely tartalma egygyel jobbra tolódik.

Állapotbitek: CY, N=0, H=0.

RRC m

A CPU az "m"-mel definiált bájtot úgy tolja jobbra, hogy a 0-ás bithely tartalma a C állapotbit helyére és a 7-es bithelyre kerül. A többi bithely tartalma egygyel jobbra tolódik.

Állapotbitek: CY, N=0, P, H=0, Z, S.

RRD

A HL regiszterpárral címzett tárrekesz alsó félbájtjának tar-

tartalmát az akkumulátor alsó félbájttjába tölti, annak megelőző tartalmát a címzett tárrekesz felső félbájttjába, ennek megelőző tartalmát az alsó félbájttba tölti át.

Állapotbitek: N=0, P, H=0, Z, S.

SCF

CY:=1

Az átvitel-állapotbit helyére 1-et ír.

Állapotbitek: CY=1, N=0, H=0.

SET b,m

mb:=1

Az "m"-mel definiált bájt b-edik bitjét 1-re állítja be.

Az állapotbitek tartalma változatlan.

SLA m

A CPU az "m"-mel definiált bájtot úgy tolja balra, hogy a 0-ás bithely tartalma 0 lesz. A többi bithely tartalma eggyel balra tolódik.

Állapotbitek: CY, N=0, P, H=0, Z, S.

SRA m

A CPU az "m"-mel definiált bájtot úgy tolja jobbra, hogy a 7-es bithely tartalma nem változik, a 0-ás bithely a C állapotbitbe kerül. A többi bithely tartalma eggyel jobbra tolódik.

Állapotbitek: CY, N=0, P, H=0, Z, S.

SRL m

A CPU az "m"-mel definiált bájtot úgy tolja jobbra, hogy a 0-ás bithely tartalma a C állapotbitbe kerül, a 7-es bithely tartalma 0 lesz. A többi bithely tartalma eggyel jobbra tolódik.

Állapotbitek: CY, N=0, P, H=0, Z, S.

XGR s

A:=A logikai kizáró VAGY s

Az akkumulátor és az "s" között kizáró VAGY kapcsolatot hoz létre. Az eredményt az akkumulátorba írja be.

Állapotbitek: CY=0, N=0, P, H=1, Z, S.

Adatmozgató utasítások

EX AF, AF'

AF csere AF'

Az AF regiszterpár tartalmát felcseréli a háttér AF regiszterpár-tartalmával. Az állapotbitek a háttér AF tartalmának megfelelően állítódnak be.

EX DE, HL

DE csere HL

A DE regiszterpár tartalmát felcseréli a HL regiszterpár tartalmával.

Az állapotbitek nem változnak.

EX (SP), HL

H csere (SP+1)

L csere (SP)

A HL regiszterpár tartalmát kicseréli a veremtár (STACK) legfelső rekeszeiben levő két bájt tartalmával. Az SP veremtármutató (stack pointer) értéke nem változik. Az állapotbitek nem változnak.

EX (SP), IX

XH csere (SP+1)

XL csere (SP)

Az IX regiszterpár tartalmát kicseréli a STACK legfelső rekeszeiben levő két bájt tartalmával. Az SP értéke nem változik. Az állapotbitek változatlanok.

EX (SP), IY

YH csere (SP+1)

YL csere (SP)

Az IY regiszterpár tartalmát kicseréli a STACK legfelső rekeszeiben levő két bájt tartalmával. Az SP értéke nem változik. Az állapotbitek nem változnak.

EXX

(BC) csere (BC')

(DE) csere (DE')

(HL) csere (HL')

Az általános célú regiszterek (BC, DE, HL) tartalmát a megfelelő háttérregiszterek (BC', DE', HL') tartalmával cseréli ki. Az állapotbitek nem változnak.

INDR

(HL):=(C)

B:=B-1

HL:=HL-1

Az IND utasítást ismétli mindaddig, amíg a B regiszter tartalma nullára nem csökken.

Állapotbitek: N=1, P=?, H=?, Z=1, S=?.

INIR

(HL):=(C)

B:=B-1

HL:=HL-1

Az INI utasítást ismétli mindaddig, amíg a B regiszter tartalma nullára nem csökken.

Állapotbitek: N=1, P=?, H=?, Z=1, S=?.

LD A, (BC)

A:=(BC)

A BC regiszterpárral címzett tárrekesz tartalmát A-ba tölti. Az állapotbitek nem változnak.

```
LD A, (DE)
A:=(DE)
```

A DE regiszterpárral címzett tárrekesz tartalmát A-ba tölti. Az állapotbitek nem változnak.

```
LD A; (nn)
A:=(nn)
```

Az "nn" operandussal címzett tárrekesz tartalmát A-ba tölti. Az állapotbitek nem változnak.

```
LD A, I
A:=I
```

Az A-ba tölti az I regiszter tartalmát.  
Állapotbitek: N=0, P, H=0, Z, S.

```
LD A, R
A:=R
```

Az A-ba tölti az R regiszter tartalmát.  
Állapotbitek: N=0, P, H=0, Z, S.

```
LD (BC), A
(BC):=A
```

BC-vel címzett tárrekeszbe tölti az A tartalmát. Az állapotbitek nem változnak.

```
LD dd, nn
dd:=nn
```

A dd regiszterpárt "nn"-nel feltölti. Az állapotbitek nem változnak.

LD dd, (nn)

ddH:=(nn+1)

ddL:=(nn)

A dd regiszterpárt az "nn" és az "nn+1"-gyel címzett tárrekeszek tartalmával tölti fel. Az állapotbitek nem változnak.

LD (DE), A

(DE):=A

A DE regiszterpárral címzett tárrekeszbe tölti az A tartalmát. Az állapotbitek nem változnak.

LD HL, (nn)

H:=(nn+1)

L:=(nn)

A HL regiszterpárba tölti az "nn"-nel címzett tárrekeszek két egymás után következő bájtját. Az állapotbitek nem változnak.

LD I, A

I:=A

Az I (megszakításvektor) feltöltése A tartalmával. Az állapotbitek nem változnak.

LD IX, (nn) (

XH:=(nn+1)

XL:=(nn)

Az IX regiszterpárt az "nn" és az "nn+1"-gyel címzett tárrekeszek tartalmával tölti fel. Az állapotbitek nem változnak.

LD IX, nn

IX:=nn

Az IX regiszterpárba az "nn"-t tölti be. Az állapotbitek nem változnak.

LD IY, (nn)

YH:=(nn+1)

YL:=(nn)

Az IY regiszterpárt az "nn" és az "nn+1"-gyel címzett tárrekeszek tartalmával tölti fel. Az állapotbitek nem változnak.

LD IY, nn

IY:=nn

Az IY regiszterpárba tölti be az "nn"-t. Az állapotbitek nem változnak.

LD m,s

Az utasítás hatására az "s"-sel definiált bájt tartalmát az "m"-mel definiált helyre tölti. Csak az egyik operandus lehet (HL), (IX+d), (IY+d). Az állapotbitek nem változnak.

LD (nn), A

(nn):=A

Az utasítás hatására az "nn"-nel címzett tárrekeszt feltölti az A tartalmával. Az állapotbitek nem változnak.

LD (nn), dd

(nn+1):=ddH

(nn):=ddL

A dd regiszterpár tartalmával tölti fel az "nn" és az "nn+1" című tárrekeszeket. Az állapotbitek nem változnak.

LD (nn), IX

(nn+1):=XH

(nn):=XL

Az IX regiszterpár tartalmával tölti fel az "nn" és az "nn+1" című tárrekeszeket. Az állapotbitek nem változnak.



LD (nn), IY

(nn+1):=YH

(nn):=YL

Az IY regiszterpár tartalmával tölti fel az "nn" és az "nn+1" című tárrekeszeket. Az állapotbitek nem változnak.

LD R, A

R:=A

A tárfrissítő regisztert feltölti az A tartalmával. Az állapotbitek nem változnak.

LD SP, HL

SP:=HL

A veremtármutatóba tölti be a HL regiszterpár tartalmát. Az állapotbitek nem változnak.

LD SP, IX

SP:=IX

A veremtármutatóba tölti az IX regiszterpár tartalmát. Az állapotbitek nem változnak.

LD SP, IY

SP:=IY

A veremtármutatóba tölti az IY regiszterpár tartalmát. Az állapotbitek nem változnak.

LDD

(DE):=(HL)

DE:=DE-1

HL:=HL-1

BC:=BC-1

A HL regiszterpárral megcímezett tárrekesz tartalmát tölti a DE regiszterpár által címzett tárrekeszbe. A HL, DE és BC regiszterpárok tartalmát eggyel csökkenti. P bit értéke 1, ha a BC regiszterpár tartalma nulla.

Állapotbitek: N=0, P, H=0.

LDDR

(DE):=(HL)

DE:=DE-1

HL:=HL-1

BC:=BC-1

A HL regiszterpárral megcímezett tárrekesz tartalmát tölti a DE regiszterpár által címzett tárrekeszbe. A HL, DE és BC regiszterpárok tartalmát eggyel csökkenti. Az utasítást mindaddig ismétli, amíg a BC tartalma nullára nem csökken.

Állapotbitek: N=0, P=0, H=0.

LDI

(DE):=(HL)

DE:=DE+1

HL:=HL+1

BC:=BC-1

A HL regiszterpárral megcímezett tárrekesz tartalmát tölti a DE regiszterpár által címzett tárrekeszbe. A HL és DE regiszterpárok tartalmát eggyel növeli, a BC tartalmát eggyel csökkenti.

Állapotbitek: N=0, P, H=0.

LDIR

(DE):=(HL)

DE:=DE+1

HL:=HL+1

BC:=BC-1

A HL regiszterpárral megcímezett tárrekesz tartalmát tölti a DE regiszterpár által címzett tárrekeszbe. A HL és DE regiszterpá-

ok tartalmát eggyel növeli. A BC blokkszámláló regiszterpár tartalmát eggyel csökkenti. Az utasítást mindaddig ismétli, amíg a BC tartalma nullára nem csökken.

Állapotbitek: N=0, P=0, H=0.

POP IX

XH:=(SP+1)

XL:=(SP)

SP:=SP+2

Az IX regiszterpárba betölti a veremtár legfelső rekeszeiben levő két bájtot. Az SP tartalmát növeli kettővel. Az állapotbitek változatlanok.

POP IY

YH:=(SP+1)

YL:=(SP)

SP:=SP+2

Az IY regiszterpárba betölti a veremtár legfelső rekeszeiben levő két bájtot. Az SP tartalmát kettővel növeli. Az állapotbitek változatlanok.

POP qq

qqH:=(SP+1)

qqL:=(SP)

SP:=SP+2

A qq regiszterpárba betölti a veremtár legfelső rekeszeiben levő két bájtot. Az SP tartalmát kettővel növeli. Az állapotbitek változatlanok.

PUSH IX

(SP-2):=XL

(SP-1):=XH

SP:=SP-2

Az IX regiszterpár tartalmát betölti a veremtármutató által címzett rekeszbe. A veremtármutató tartalmát kettővel csökkenti. Az állapotbitek nem változnak.

PUSH IY

(SP-2):=YL

(SP-1):=YH

SP:=SP-2

Az IY regiszterpár tartalmát betölti a veremtármutató által címzett rekeszbe. A veremtármutató tartalmát kettővel csökkenti. Az állapotbitek nem változnak.

PUSH qq

(SP-2):=qqL

(SP-1):=qqH

SP:=SP-2

A qq regiszterpár tartalmát betölti a veremtármutató által címzett rekeszbe. A veremtármutató tartalmát kettővel csökkenti. Az állapotbitek nem változnak.

Be- és kiviteli utasítások

IN A, (n)

A:=(An)

A CPU az A0--A7 jelű címsínekre az "n" operandust, az A8--A15-re az akkumulátor tartalmát helyezi. Az így megcímezett perifériáról olvassa be az adatot az akkumulátorba. Az állapotbitek nem változnak.

IN r, (C)

r:=(C)

A BC regiszterpárral címzett perifériáról olvassa be az adatot az "r" regiszterbe. Ha r=F, akkor nem az adatot olvassa, hanem

csak az állapotbitekét állítja be (az utasítás kódja: ED 70).  
Állapotbitek: N=0, P, H, Z, S.

IND

(HL):=(c)

B:= B-1

HL:=HL-1

A BC regiszterpárral címzett perifériáról tölt adatot a HL-lel címzett tárrekeszbe, majd a B regiszter tartalmát eggyel csökkenti. A HL regiszterpár tartalmát is eggyel csökkenti.

Állapotbitek: N=1, P=?, H=?, Z, S=?.

INI

(HL):=(C)

B:=B-1

HL:=HL+1

A BC regiszterpárral címzett perifériáról tölt adatot a HL-lel címzett tárrekeszbe, majd a B regiszter tartalmát eggyel csökkenti. A HL regiszterpár tartalmát eggyel növeli.

Állapotbitek: N=1, P=?, H=?, Z, S=?.

OTDR

(C):=(HL)

B:=B-1

HL:=HL-1

A HL regiszterpárral címzett tárrekesz tartalmát a "C" regiszterrel (BC regiszterpárral) címzett perifériába juttatja. A HL regiszterpár és a B regiszter tartalmát eggyel csökkenti. Az utasítást addig ismétli, amíg a BC tartalma nullára nem csökken.

Állapotbitek: N=1, P=?, H=?, Z=1, S=?.

OTIR

(C):=(HL)

B:=B-1

HL:=HL+1

A HL regiszterpárral címzett tárrekesz tartalmát a C regiszterrel (BC regiszterpárral) címzett perifériába juttatja. A HL regiszterpár tartalmát eggyel növeli, a B regiszter tartalmát eggyel csökkenti. Az utasítást addig ismétli, amíg a B tartalma nullára nem csökken.

Állapotbitek: N=1, P=?, H=?, Z=1, S=?.

```
OUT(C), r
(C):=r
```

Az "r" regiszter tartalmát a C regiszterrel (BC regiszterpárral) címzett perifériának kiküldi. Az állapotbitek nem változnak.

```
OUT(n), A
(n):=A
```

Az A regiszter tartalmát az "n"-nel címzett perifériának kiküldi. Az állapotbitek nem változnak.

```
OUTD
(C):=(HL)
B:=B-1
HL:=HL-1
```

A C regiszterrel (BC regiszterpárral) címzett perifériára kiviszi a HL regiszterpárral címzett tárrekesz tartalmát. A HL regiszterpár és a B regiszter tartalmát eggyel csökkenti.

Állapotbitek: N=1, V, H=?, Z, S=?.

```
OUTI
(C):=(HL)
B:=B-1
HL:=HL+1
```

A C regiszterrel (BC regiszterpárral) címzett perifériára kiviszi a HL regiszterpárral címzett tárrekesz tartalmát. A HL

regiszterpár tartalmát eggyel növeli. A B regiszter tartalmát eggyel csökkenti.

Állapotbitek: N=1, V, H=?, Z, S=?.

Programszervező utasítások

CALL cc, nn

Ha a cc igaz: (SP-1):=PCH

(SP-2):=PCL

SP:=SP-2

PC:=nn

A feltétel teljesülése esetén szubrutinhívást hajt végre, azaz a következő utasításra mutató PC tartalmát a veremtárba helyezi el, majd az "nn"-t tölti a PC-be. Az állapotbitek nem változnak.

CALL nn

(SP-1):=PCH

(SP-2):=PCL

SP:=SP-2

PC:=nn

A CPU feltétel nélküli szubrutinhívást hajt végre. Ez úgy történik, hogy a következő utasításra mutató PC tartalmát a veremtárba helyezi el, majd az "nn"-t tölti a PC-be. Az állapotbitek nem változnak.

DI

IFF:=0

Az utasítás tiltja a maszkolható megszakításkérés érvényesülését. Az állapotbitek nem változnak.

DJNZ e

B:=B-1

ha B nem 0, akkor PC:=PC+e

A B regiszter tartalmát eggyel csökkenti. Ha a B új tartalma nem nulla, akkor a PC-hez "e"-t hozzáadja, azaz a PC+e címre adja a vezérlést. Ha a B regiszter 0-ra csökken, a következő utasításra lép. Tipikus ciklusszervező utasítás. Az állapotbitek nem változnak.

EI

IFF:=1

Az utasítás a maszkolható megszakításkérés érvényesülését engedélyezi, az EI utasítást követő utasítás végrehajtása után. Megjegyzendő, hogy ezt az utasítást még engedélyezett maszkolható megszakításkérés sem szakíthatja meg. Az állapotbitek nem változnak.

HALT

Megállás

Az utasítás hatására a CPU nem növeli az utasítás számlálóját, a címről beolvasott utasítást figyelmen kívül hagyva "NOP" utasítást hajt végre, amíg megszakításkérés vagy RESET nem érkezik. Az állapotbitek nem változnak.

IM 0

A nulla típusú megszakításmód beállítása

Az utasítás a CPU-t a nulla típusú megszakításmódba (I8080 megszakítási mód) állítja. Az állapotbitek nem változnak.

IM 1

Az egy típusú megszakításmód beállítása

Az utasítás a CPU-t az egy típusú megszakításmódba állítja. Megszakításkéréskor a CPU egy szubrútint hív a hexa 38 címről. Az állapotbitek nem változnak.

IM 2

A kettő típusú megszakításmód beállítása



Az utasítás a CPU-t a kettő típusú megszakításmódba állítja. Ebben az üzemmódban a megszakításkérés vektoros. Az állapotbitek nem változnak.

NOP

Nincs műveletvégzés.

RET

PCL:=(SP)

PCH:=(SP+1)

SP:=SP+2

Visszatérés a szubrutinból. A veremtármutató által kijelölt két bájt tartalmát betölti a programszámlálóba. A veremtármutató tartalmát kettővel növeli. Az állapotbitek változatlanok.

RET cc

Ha cc igaz: PCL:=(SP)

PCH:=(SP+1)

SP:=SP+2

Feltételes visszatérés a szubrutinból. Ha a feltétel igaz, a veremtármutató által kijelölt két bájt tartalmát betölti a programszámlálóba. A veremtármutató tartalmát kettővel növeli. Az állapotbitek változatlanok.

RETI

PCL:=(SP)

PCH:=(SP+1)

SP:=SP+2

Visszatérés a megszakításból. A veremtármutató által kijelölt két bájt tartalmát betölti a programszámlálóba. A veremtármutató tartalmát kettővel növeli. Az állapotbitek változatlanok.

RETN

PCL:=(SP)

PCH:=(SP+1)

SP:=SP+2

Visszatérés a nemmaszkolható megszakitásból. A veremtármutató által kijelölt két bájt tartalmát betölti a programszámlálóba. A veremtármutató tartalmát kettővel növeli. A CPU IFF2 flip-flopjának tartalmát áttölti az IFF1 flip-flopba. Az állapotbitek változatlanok.

```
RST p
  (SP-1):=PCH
  (SP-2):=PCL
  SP:=SP-2
  PCH:=0
  PCL:=P
```

Szubrutinhívás az első lapon lévő fix címekre. A futó program utasításszámlálójának tartalmát a veremtárba tölti. Az utasításszámlálóba (PC) a "p"-nek megfelelő értéket tölti be. A lehetséges értékek: 0, 8, 10H, 18H, 20H, 28H, 30H, 38H. Az állapotbitek nem változnak.

Vezérlésátadó utasítások

```
JP cc, nn
  ha cc igaz, PC:=nn
```

A feltétel (cc) teljesülése esetén megvalósul a vezérlésátadás. Az állapotbitek nem változnak.

```
JP (HL)
  PC:=HL
```

A HL regiszterpár tartalma megcímezi a következő programutasítást. Az állapotbitek nem változnak.

```
JP (IX)
  PC:=IX
```

Az IX regiszterpár tartalma megcímezi a következő programutasítást. Az állapotbitek nem változnak.

JP (IY)

PC:=IY

Az IY regiszterpár tartalma megcímezi a következő programutasítást. Az állapotbitek nem változnak.

JP nn

PC:=nn

Feltétel nélküli vezérlésátadás az "nn" címre. Az állapotbitek nem változnak.

JR e

PC:=PC+e

Feltétel nélküli relatív vezérlésátadás a PC+e címre. Az állapotbitek nem változnak.

JR C,e

ha C=1, PC:=PC+e

Feltételes relatív vezérlésátadás a PC+e címre. Ha C=0, akkor a következő utasításra lép. Az állapotbitek nem változnak.

JR NC,e

ha C=0, PC:=PC+e

Feltételes relatív vezérlésátadás a PC+e címre. Ha C=1, akkor a következő utasításra lép. Az állapotbitek nem változnak.

JR NZ, e

ha Z=0, PC:=PC+e

Feltételes relatív vezérlésátadás a PC+e címre. Ha Z=1, akkor a következő utasításra lép. Az állapotbitek nem változnak.

JR Z, e

ha Z=1, PC:=PC+e

Feltételes relatív vezérlésátadás a PC+e címre. Ha Z=0, akkor a következő utasításra lép. Az állapotbitek nem változnak.

## A programok beírása

Az assembly nyelvű programot átkódolás után gépi kódú utasítások sorozataként tudjuk a tárba beírni. A beírást a BASIC nyelvű CODE és a POKE utasításokkal végezhetjük el. Kezdeti próbálkozásainknál előnyösebb a CODE használata, ugyanis ezzel a tárban való elhelyezésről nem nekünk kell gondoskodnunk, míg a POKE utasításban meg kell adnunk azt a tárcímet, ahová a kódot írni akarjuk. Ha ezt a címet eltévesztjük, akkor a következmények beláthatatlanok. A CODE utasítással mindig az első szabad helyre írunk, amelynek megadása nem a mi dolgunk. Így egy hibalehetőséggel kevesebb lesz ugyan, de ennek ára van: a programba nem írhatjuk be a címek abszolút számértékét. A kis programokban a Z80 assembly relatív címmegadási lehetősége még segíthet. Természetesen a tár címezését, pontosabban a programban egy-egy utasítás vagy adat helyének meghatározását elkerülni nem lehet. A következő példában bemutatotthoz hasonló megoldás segíthet (a hexadecimális számokat a függelékben található assembly utasítások táblázatából olvashatjuk ki):

```
CODE CIM1=HEX$("78,BC,CA")
CODE WORD$(CIM1+5)
CODE HEX$("C9,79,8D,C9")
```

Ez a HL és BC regiszterpárok értékét összehasonlító kis eljárás assembly nyelven így írható:

```
CIM1: LD  A,B
      CP  H
      JP  Z, CIM2
      RET
CIM2: LD  A,C
      CP  L
      RET
```

Ha a relatív vezérlésátadó utasítást használjuk, a WORD\$ függvény elmarad:

```
CIM1: LD A,B
      CP H
      JR Z,1
      RET
      LD A,C
      CP L
      RET
```

Beírása:

```
CODE CIM1=HEX$("78,BC,28,01,C9")
CODE HEX$("79,8D,C9")
```

Az eljárást a feltételes RET utasítás használatával egyszerűsíthetjük :

```
CIM1: LD A,B
      CP M
      RET NZ
      LD A,C
      CP L
      RET
```

Beírása:

```
CODE CIM1=HEX$("78,BC,C0,79,8D,C9")
```

A gépi kódú, ill. assembly nyelvű programozás során az ilyen jellegű apró változtatásokkal tehetjük a programokat valamilyen szempontból kedvezőbbé. Az utolsó változatot a legegyszerűbb beírni -- és a relatív címet sem hibázhatjuk el --, de elveszítjük azt az információt, hogy már a B és H regiszterek értéke különbözik. Az első két esetben ezt az eljáráson belül könnyebben kihasználhatjuk, hiszen a feltételes vezérlésátadás és a RET között megvan a helye a nemzérus ágak is. A nemzérus ágba

Újabb utasításokat könnyebb beírni az első változatban, mivel a másodikban a relatív címet minden módosítás után újra ki kell számolni, ezt viszont az assembler sokszor nem végzi el. A fordítók ugyanis általában a fordítás során az utasítások címrészében lévő, a programban később megadott utasításcímeket is értelmezni tudják, míg a relatív ugrás értékét meghatározó számításokat gyakran a programozónak kell elvégeznie. Ha nincs assembler programunk, mindkét esetben ki kell számolnunk a címeket. Amíg az az első esetben csak a számolás pontosságára kell ügyelnünk, addig a másodikban a relatív címzési szabályok betartására is.

### Gépi kódú eljárások használata

A tárban lévő gépi kódú eljárásokat az ENTERPRISE BASIC nyelvben kétféleképpen futtathatjuk:

1. `USR` függvényhívással és
2. `CALL` utasítással.

Az `USR` függvényhívás egyben azt is jelenti, hogy a zárójelben megadott paramétert betöltjük a HL regiszterpárba. Például az

```
USR (CIM1,16810)
```

függvényhívással az eljárásban a BC regiszterpár pillanatnyi értékét 16810-zel hasonlítjuk össze, és a függvény értéke -- mivel az eljárás HL értékét nem változtatja -- 16810 lesz. A `CALL` utasítással, amelynek formája esetünkben a következő:

```
CALL USR(CIM1,16810)
```

egyszerűen csak elindítjuk az eljárást; értéket nem adunk át és nem is kapunk vissza. Az utasításban lévő második operandus hatástalan. Ezzel az utasítással az eljárás például a BC és HL regiszterpárok pillanatnyi értékét hasonlítja össze, és a feltételbitekét állítja csak be. Az eljárás futtatása előtt tehát gondoskodni kell a regiszterpárok értékének beállításáról, azután pedig az állapotbitek használatáról. Ezek a "gondoskodások"

lehetnek újabb eljárás-hívások vagy az eljárás beépítése egy nagyobb programba, amelyik ezeket a feladatokat megoldja. Figyelni kell arra, hogy az eljáráson belül a hívó programba a visszatérést feltétlenül biztosítsuk! (Leegyszerűsítve: az eljárásokban a RET utasítások valamelyik változatának szerepelnie kell!)

A gépi kódú programrészek és a hívó BASIC programok közötti adatcserét nem csak a paraméterekkel oldhatjuk meg. A tár közvetlen címzésével -- a POKE és PEEK, illetve SPOKE és SPEEK -- is beállíthatjuk vagy kiolvashatjuk a tárterület egy részét. A közvetlen tárkezelés azonban a cím meghatározásában elkövethető hibák miatt a paraméterek alkalmazásához képest rendkívül sok hibalehetőséggel jár. Ezért kezdetben inkább használjuk a biztonságosabb megoldást!

# Beépített segédprogramok

## Szövegszerkesztés ENTERPRISE számítógéppel

A számítógépek sajátos alkalmazási területe a levelek, üzenetek, feljegyzések (például napló) készítése. Ennek során nem számolni akarunk, hanem olyan műveleteket elvégezni, amelyekkel a karaktereket -- betűket, számjegyeket, írásjeleket stb. -- jelenítünk meg az általunk elképzelt sorrendben szavakká, mondatokká, bekezdésekké stb. rendezve. Az így kialakított karaktersorozatot, a szöveget, célszerű a későbbi felhasználásra, ellenőrzésre, javításra a magunk és mások számára is tárolni. Az ilyen szövegkezelést együttesen szövegszerkesztésnek nevezük, a feladat elvégzését segítő programot pedig szövegszerkesztőnek.

A szövegszerkesztők a házi számítógépek megjelenésével kezdtek elterjedni. Korábban az ilyen feladatok elvégzésére külön berendezéseket, az ún. íróautomatákat használták. Ezek -- méretüknél és értéküknél fogva -- csak irodákban működtek, így legfontosabb alkalmazási területük a levelezés és dokumentálás volt. A mai szövegszerkesztők, így az ENTERPRISE szövegszerkesztője is, lényegesen eltér elődjeitől, hiszen a számítógépnek csak az egyik, és nem egyetlen futó programja. Az előállított szövegek sem csak egy-két meghatározott célra, hanem többféle felhasználásra készülnek.

A szövegszerkesztők a következő alapvető műveleteket teszik lehetővé: szövegek írása (begépelés), megjelenítése (kiírás), tárolása. Ezeket a műveleteket géptípusonként és összeállításonként eltérő módon vehetjük igénybe. Most a magnóval, fekete-fehér tv-vel összeállított, nyomtató nélküli ENTERPRISE számítógép alpműveleteit mutatjuk be.



Az ENTERPRISE számítógépek bekapcsolása után a szövegszerkesztő program indul el. Minden más programot -- a BASIC értelmezőt is -- külső programként, csak a programtár csatlakoztatása után használhatunk. A szövegszerkesztő program készítői tehát csak a képernyőt, a billentyűzetet és a botkormányt tekintették adott-nak, a gép egyéb kiegészítő részeit csupán lehetőségként vették számításba (például a nyomtatót, a magnót és a lemezegységet). A szöveg beírása során tehát ezekre támaszkodhatunk.

A képernyőn -- a szövegszerkesztő funkcióbillentyűk értelmezését megjelenítő sorokon kívül, amelyek főleg a használat kezdetén jelentenek segítséget -- 19 sorba, soronként 38 karaktert írhatunk.

A szövegszerkesztő használható méretét a következő példánk mutatja be:

```

12345678901234567890123456789012345678
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

```

Pontosabban ennyi a látható karakter. A továbbiakban ezt a 19 sort nevezzük képernyőnek. A 20. sorba való áttéréskor a képernyő első sora eltűnik a szemünk elől. Visszahozásakor az alsó, a 20. sor tűnik el a képernyőről. Igaz, az sem vész el, a kurzor pozicionálásával visszahozható. A sorokba írt karakterek

száma is növelhető, amit az INS billentyű lenyomását követő karakterbeszúrással érhetünk el.

Az első sorból karakterbeszúrással kialakítható hosszú sor például a következő:

```
123456789012345678901234567890
12345
```

A képernyőről kilépő karakterek nem láthatók, csak egy > jel mutatja jelenlétüket. Vagy az ERASE billentyű ismételt leütésével, vagy az F1 funkcióbillentyű és az ALT vagy a CTRL váltó együttes leütésével visszahozhatók. Az ERASE billentyű használata esetén a sorból annyi karaktert törölünk, hogy már beférjen a képernyőre.

Az első sort ERASE billentyűvel átalakítva a következő sort is kaphatjuk:

```
12ab345678901234567890123456789012345678
```

A látható karakterek számát így kettővel növelhetjük.

Az F1 funkcióbillentyűvel nem hagyunk ki egyetlen karaktert sem, csupán a hosszú sort több db 38 karakteres sorrá rendezzük át. Példánkban az első sor átrendezése után a következő képernyőt kapjuk:

```
12345678901234567890123456701234
5678901234567882
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

Az első hosszú srt két sorra alakítottuk és a második sorban lévő 2-essel kiegészítettük.

A begépeléskor a margókat és a tabulátorokat kezelő funkcióbillentyűk hatása megegyezik az írógépeken található hasonló funkciójú billentyűk hatásával. Két, az írógépekről ismert billentyűnek: a szóköznek és a sorváltó (ENTER) billentyűnek a szokásostól eltérő jelentése is van.

A szóköz a szavakat határolja. Beírásnál a sorok végén a másik sorba átnyúló szavakat a szövegszerkesztővel automatikusan a következő sor elejére írjuk. Kivétel ez alól az a "szó", amelyik túl hosszú, mint a példánkban írt első sor, amelyikben egyetlen szóközkarakter sem volt.

Az ENTER a bekezdéseket határolja. Az F1 funkcióbillentyű a beírt szöveget bekezdésenkénti egységekre osztva kezeli. Megtehetjük, hogy minden sort bekezdésként értelmezzünk, azaz minden sor végén ENTER leütésével lépünk át a következő sorba. A szövegszerkesztő programba beépített automatikus sorváltás eltér az ENTER-rel való sorváltástól. A képernyőn a kettő közötti különbség nem látható. Az ENTER billentyű segítségével azonban kideríthetjük az eltérést. Leütésére ugyanis a kurzor a következő ENTER-leütési hely utáni pozícióra áll, azaz a következő bekezdés elejére. A legutóbb bemutatott képernyőn az első sorban leütött ENTER hatására a kurzor a 3. sor elején lévő 3-as szám után villog.

## Szövegek módosítása, javítása

A begépelés során észrevett hibák javítását, vagy a begépelés után adódó módosításokat a szövegszerkesztő az írógépeknél nem található sajátos szolgáltatásokkal segíti. Ezeket a HELP (pl. F4 funkcióbillentyű) billentyű leütésével áttekinthetjük. A lehetőségek közös alapja, hogy a tárban lévő karakterek újrabil-lentyűzés nélkül áthelyezhetők.

A HELP listában nem található meg az F6 funkcióbillentyűvel kiváltható áthelyezési lehetőség. CTRL kiegészítéssel felfelé, ALT kiegészítéssel lefelé mozgathatunk egy bekezedést, azaz egy ENTER billentyűvel lezárt szövegrészt. A mozgatóst azonban nem

bekezdésként, hanem soronként végezhetjük el. Az utolsó példánkban bemutatott képernyőt ennek a műveletnek az ábrázolása érdekében kibővítettük. A 4. sort kétsoros bekezdéssé alakítottuk:

```
123456789012345678901234
567890123456782
3
444444
444444
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```

Az 1. sorra állított kurzor mellett az ALT és F6 billentyű hatására a következő képernyőt kapjuk:

```
3
123456789012345678901234
567890123456782
4
44444
5
6
7
8
9
10
11
12
13
14
15
16
17
18
```



Az összeállított vagy csak begépelte szöveggel való foglalatokódást későbbre halaszthatjuk, ha a szöveget kazettára -- esetleg lemezre -- másoljuk. Az ENTERPRISE kikapcsolásával elvesző munkánk eredményét így újra beolvastva ismét felhasználhatjuk. A szövegek szalagra írását az F2 -- SAVE feliratú -- funkcióbillentyű leütésével kezdhetjük el. Ezután a szöveg azonosítására szánt név beírását lezáró ENTER billentyű leütését várja a szövegszerkesztő. Az ENTER leütése előtt azonban jól be kell állítanunk a magnót, ugyanis a gép azonnal megkezd az "adást", függetlenül attól, hogy van-e egyáltalán magnónk. Ha elkészünk a magnó indításával, esetleg csak egy részét vesszük fel a szövegnek. Ez természetesen hiba, és a felvétel használhatatlan lesz. Vigyáznunk kell a kazetta beállítására is, ugyanis a SAVE utasítással nem ellenőrizzük, hogy a szalagon már van-e valami, így értékes részéket felülírva azokat akaratlanul is elronthatjuk.

A kazettára felírt szöveget beolvastva -- a LOAD feliratú F1 funkcióbillentyűvel -- a tárban lévő szöveget töröljük. Ez megnehezíti a szalagra írt szövegrészek összemásolását. A teljes szövegek megjelenítését viszont változtathatjuk a sorszélesség előzetes beállításával. Így elérhetjük, hogy a 40 karakteres képernyőn összeállított szövegrészeket az esetleg 80 karakteres nyomtatónak megfelelően is láthatjuk képernyőnkön.

A beépített szövegszerkesztő programot -- programozási tapasztalataink gyarapodásával -- kitűnő alapnak tekinthetjük sajátos igényünknek megfelelő szövegkezelő rendszer kidolgozásához.

### Grafika a képernyőn

Az ENTERPRISE grafikus lehetőségeinek megismerését a GRAPHICS feliratú funkcióbillentyű használatával kezdjük. Leütése a GRAPHICS parancs kiadását jelenti, amely alapértelmezésben a képernyőt két részre osztja: a felső 20 sorra, amit üresen hagy, és az alsó 4 sorra, ahol ezt követően a gép üzenetei és az általunk beírt parancsok megjelennek.

A képernyő felső része a grafikus lap, az alsó a szöveges lap. A grafikus lapra vonatkozó parancsokat, programokat a BASIC rendszerben megszokott módon adhatjuk meg, de a beírtakból és a gép üzeneteiből egyszerre csak 4 sort nyit láthatunk.

## A grafikus lap

Grafika készítéséhez a grafikus lapot pontokra osztották. Minden pontnak meg lehet adni a színét. Mivel a pontok adatait a tárában kell elhelyezni, és a tárterület adott nagyságú, a pontok lehetséges színeinek számát növelve a grafikus lap pontjainak a számát csökkenteni kell. Így a 2-színű lapon lévő pontok száma  $180 \times 640$ , a 4-színű lapon  $180 \times 320$ , a 16 színű lapon  $180 \times 160$  és a 256 színű lapon  $180 \times 80$ . A különböző számú pontból álló lapok egységes kezelhetőségét úgy biztosították, hogy a lapok pontjait a színek számától függetlenül vízszintes irányban 0-tól 1279-ig, függőleges irányban 0-tól 719-ig sorszámozott pontoknak tekintik. A képernyő egy-egy pontját így több képeletbeli pont is kijelöli.

A funkcióbillentyűvel kiválasztott, alapértelmezés szerint 4-színű képernyőn egy vonalat rajzoló PLOT program futtatásával megnézhetjük a rajzolás menetét. A következő programmal a pontok koordinátáit (vízszintes és függőleges sorszámaikat) egyesével növelve rajzolunk a grafikus lapra:

```
100 GRAPHICS
110 LET J=0
120 FOR I=0 TO 1279
130   PLOT I,J
140   IF J 718 THEN
150     LET J=0
160   ELSE
180     LET J=J+1
190   END IF
200 NEXT I
210 END
```

A függőleges sorszám kialakítása során figyelni kell arra, hogy a túl nagy sorszám (pl. 720) hibát okoz. A hiba egy hibajelzés kiíratása után a programot megállítja. A feltételes utasításokkal ezt a programhibát kerültük el.

## Függvények rajzolása

A BASIC beépített függvényeinek ábráját a PLOT utasítással könnyen megjeleníthetjük a képernyőn. Ábrázolhatjuk például a négyzetgyökfüggvényt (SQR), ha a következő programmal sorra vesszük annak pontjait.

```
100 GRAPHICS
110 PLOT 0,0;1279,0;0,0;0,719;
120 FOR I=0 TO 1279
130   LET J=SQR(I)
140   PLOT I,J;
150 NEXT I
160 END
```

A program elég lassan rajzol. A rajzolást meggyorsíthatjuk, ha a grafikus lap képzeletbeli pontjait nem egyesével, hanem nagyobb lépésközökkel növeljük:

```
100 GRAPHICS
110 PLOT 0,0;1279,0;0,0;0,719;
120 FOR I=0 TO 1279 STEP 10
130   LET J=SQR(I)
140   PLOT I,J;
150 NEXT I
160 END
```

A képpontok függőleges koordinátáit a függvénynek a pont vízszintes sorszámához tartozó értékei határozzák meg. Mivel ez az érték kicsi, az ábra túl lapos. Jobban látható a függvény alakja, ha azt például a programnak a következő módosításával megnyújtjuk:



```

100 GRAPHICS
110 PLOT 0,0;1279,0,0,0;0,719;
120 FOR I=0 TO 1279 STEP 10
130 LET J=20*SQR(I)
140 PLOT I,J;
150 NEXT I
160 END

```

A függvényábrára legjobb méreteinek kialakítása a fentiekhez hasonló program módosításokkal egyszerűen elvégezhető.

A függvények ábrázolása során a görbe kezdőpontját a képernyő közepére célszerű helyezni. Ezt például a  $\sin(x)$  beépített függvényt ábrázoló, következő programban bemutatott módon végezhetjük el:

```

100 GRAPHICS
110 PLOT 0,320;1279,320,0,0;0,719,
120 FOR I=0 TO 2*PI STEP 2*PI/1279
130 LET J=300*SIN(I)+320
140 LET I1=1279*I/(2*PI)
150 PLOT I1,J;
160 NEXT I
170 END

```

### Rajzolás a képernyőn

A képernyő grafikus lapján egy, a kurzornak megfelelő pont mozgásával is rajzolhatunk. Ezt a rajzolást különleges mozgató utasítások segítik. A Felhasználói kézikönyvben ismertetett "teknőcutasítások" angol szavakkal írják le, hogy a pont milyen irányban induljon el és mennyit haladjon előre. Egy-egy egyszerű programmal megkönnyíthetjük magunknak a rajzolást. A következő például néhány billentyű lenyomására rajzol vonalat a grafikus lapon. (A program az angol ábécének megfelelő billentyűzetű gépre készült.)

100 OPTION ANGLE DEGREE\$

110 DISPLAY GRAPHICS

120 PLOT 640,360;

130 LET C\$=INKEY\$

140 IF C\$="" THEN GOTO 130

150 LET S=-1

160 IF C\$="d" THEN LET S=0

170 IF C\$="r" THEN LET S=45

180 IF C\$="e" THEN LET S=60

190 IF C\$="f" THEN LET S=90

200 IF C\$="l" THEN LET S=270

210 IF C\$="w" THEN LET S=120

220 IF C\$="q" THEN LET S=135

230 IF C\$="a" THEN LET S=180

240 IF C\$="c" THEN LET S=-45

250 IF C\$="x" THEN LET S=-60

260 IF C\$="z" THEN LET S=-120

270 IF C\$="\ " THEN LET S=-135

280 IF S=-1 THEN GOTO 130

290 PLOT ANGLE S;

300 PLOT FORWARD 4

310 GOTO 130

500 END

Az s billentyűt körülvevő billentyűk lenyomásának a hatása olyan, mint a teknőcutasításoké. A mozgás iránya körülbelül megegyezik az s és a lenyomott billentyű irányával: a q, w, \, e és r billentyűk felfelé, a \, z, x, és c billentyűk lefelé, az a és a d billentyűk vízszintesen balra és jobbra, az f és l billentyűk függőlegesen fel-, ill. lefelé mozgatja a pontot. (A program végtelen ciklus, leállítani csak a STOP billentyűvel lehet.)

A pont a grafikus lap közepéről indul, és a mozgása is lassú. Ez nem mindig jó, ezért a rajz egyes részleteinek kialakítása előtt célszerű a programot átalakítani. Ehhez a paramétereket a STOP billentyű lenyomása után, a grafikus lapot megőrizve kell módosítani. A változtatásokat a képernyő grafikus lap alatti

részét használhatjuk. Itt ugyan a programnak csak négy sora látható, de ez is elegendő egy-egy sor átírásához. A programot néhány soros kiegészítéssel az ábra színezésére is felhasználhatjuk. Például a 120., 290. és 300. utasítások végéről töröljük ki a pontosvesszőt, és a 300. utasításban növeljük meg a lépésméretet (4-ről 20-ra), majd írjuk hozzá a következő sorokat:

```
275 IF C$="s" THEN GOTO 400
400 INPUT K
410 SET INK K
420 PLOT PAINT
430 GOTO 130
```

Az új programváltozattal a pontot az előzőhöz hasonlóan, de gyorsabban vihetjük a színezni kívánt területbe, és az s leütése után állíthatjuk be a terület színének kódját.

Az ábrák kialakítása során nem kell különválasztani a függvényrajzolás és a pontmozgatás módszerét. Az összetett ábrák rajzolásához olyan programokat írhatunk, amelyekben mindkettőt használjuk. Hosszú programokat ne tervezzünk. Helyettük az utasításokat ismertető részben bemutatott csatornamódszer alkalmazásával, a rajzolandó ábra elemeiből másoljuk össze a bonyolultabbakat.

### Hanggenerálás

Az ENTERPRISE számítógép a hasonló kategóriájú gépek közül kitűnik kiváló hanggenerálási adottságaival. A számítógépbe egy speciális hang IC van beépítve. Az IC-ben három, külön programozható hanggenerátor és egy zajgenerátor található. Az átfogható hangtartomány 8 oktáv és az IC sztereó. A különleges felépítés lehetővé teszi, hogy a hangokat sztereó fejhallgatóval vagy erősítővel is hallgathassuk. A hanghatásokat a BASIC-ből közvetlenül is irányíthatjuk.

A hangkezelő IC egyszerre csak egy BASIC csatornát tud használni. Így a többi csatornán lévő hangkérések sorban állnak, és mindegyikük egymás után lejátszásra kerül. A hangsorok maximális hossza 25 hang lehet.

A sorban álló hangok mindegyikéhez tartozik egy 0--254 számú hangburkoló is. Ez a hangburkoló vezérli a megszólaló hang frekvenciamenetét, a bal és a jobb oldali csatorna amplitúdójának (erősségének) nagyságát és az egyes fázisok hosszúságát. A megszólaló hangok maximum 255, egy-egy hangburkoló pedig 40 fázisból áll.

A fázison belül a hang változásait előjeles viszonyszámokkal adhatjuk meg. Ez a tartomány -63...+63-ig terjedhet az előző értékhez képest. Az egyes fázisok időtartamát 1/50 másodpercekben lehet megadni.

A hangokat a BASIC-ből az ENVELOPE NUMBER és a SOUND utasításokkal irányíthatjuk. A hangtár méretét fázisokban mérve a SOUND BUFFER utasítással adhatjuk meg.

Az ENVELOPE utasításban az egyes hangokhoz használható hangburkolót, a SOUND utasításban pedig a ténylegesen megszólaló hangot határozhatjuk meg.

A SOUND utasítás a következő paramétereket tartalmazza:

- Az ENVELOPE részben a burkoló azonosítóját adjuk meg.
- A PITCH kifejezésben adhatjuk meg az induló hang magasságát félhangokban, az érték 0--127 lehet. A normál C hang a 37-esnek felel meg.
- LEFT és a RIGHT a bal és a jobb oldali hangszóró amplitúdóját (hangerejét) adja meg a maximális hangerőhöz viszonyítva. Értéke 0--255 lehet.
- A STYLE a különböző megszólalási módokat állítja be. A különböző értékek a Felhasználói kézikönyvben megtalálhatók.
- A SOURCE kijelöli, hogy melyik hangforrás szólaljon meg. Értéke 0--3 lehet.
- A DURATION-ban a hang időtartamát adhatjuk meg 1/50 másodpercekben mérve.
- A SYNC a különböző hanggenerátorokat szinkronizálja egymáshoz.
- Az INTERRUPT hatálytalanítja a hang hatását, vagyis megszakítja.

Ezek után lássuk, hogy a különböző utasítások hogyan állítanak elő egy hangot.

```
100 ENVELOPE NUMBER 1;63,63,63,50;  
    63,63,63,50  
120 SOUND PITCH 0,DURATION 100,LEFT  
    200,RIGHT 200,SOURCE 0,ENVELOPE 1
```

A kis példaprogramunkkal végigskálázhatjuk az ENTERPRISE egész hangtartományát. Most pedig vizsgáljuk meg, hogy az egyes paraméterek mit jelentenek:

- PITCH 0. Azt jelenti, hogy a hang a nullás (30 Hz-es) hangról indul.
- DURATION 100. Azt jelenti, hogy két másodpercig szól a hang.
- LEFT 200 és RIGHT 200. Azt jelenti, hogy a sztereó csatornák amplitúdója 200-as értékű.
- SOURCE 0. Azt jelenti, hogy a nullás hanggenerátort jelöljük ki a hang megszólaltatásához.
- ENVELOPE 1. Azt jelenti, hogy az 1-es azonosítójú hangburkolót használjuk.

Az ENVELOPE NUMBER hangburkoló utasításban az 1-es szám az utasítás azonosítója. Az első 63-as azt jelenti, hogy az első fázisban 63 félhanggal emelkedjen a hang magassága. A második és a harmadik 63-as viszont azt, hogy a megadott hangerőre maximálisan erősítse fel a hangot az első fázisban. Az 50-es jelentése pedig az, hogy az első fázis egy másodpercig tart. A második fázisban a hangmagasság szintén 63 félhanggal emelkedik, és a hangerő nullára csökken, vagyis kikapcsoljuk a hangot. Ezt továbbfejlesztve írjuk be a következő programot is.

```
100 ENVELOPE NUMBER 1;63,63,63,50;  
    63,-63,-63,50  
110 ENVELOPE NUMBER 2;-63,63,63,50;  
    -63,-63,-63,50  
120 SOUND PITCH 0,DURATION 100,LEFT  
    200,RIGHT 200,SOURCE 0,SYNC 1  
130 SOUND PITCH 127,DURATION 100,  
    LEFT 200,RIGHT 200,SOURCE 1,  
    ENVELOPE 2,SYNC 1
```

Így két hanggenerátor egyszerre fog megszólalni, az egyik növelve, a másik csökkentve a hangfrekvenciát.

A hangokat a STYLE utasítással lehet módosítani. Például ha az előző programunk SOUND utasításaiba beillesztjük a STYLE 16 és a STYLE 32-t, akkor érdekes hanghatásokat érhetünk el. Amennyiben a programot tovább akarjuk fejleszteni, például a hangmagasságot véletlenszám-generátorral is megadhatjuk. Ha zeneszerzésre adjuk a fejünket, a hangmagasságot, a frekvenciamenetet DATA utasításokban is megadhatjuk, és a READ utasítással olvastathatjuk be. Sok gyakorlás és a különféle hanghatások pontos kiismerése után önmagunkat (esetleg barátainkat is) szórakoztató kis szintetizátorként is használhatjuk számítógépünket.

# Függelék

## Hibaüzenetek

### A hibakódok jelentése

Felhasználói hiba (0--999)  
Túlcordulás (1000--1999)  
Tömbindexhiba (2000--2999)  
Matematikai hiba (3000--3999)  
Paraméterhiba (4000--4999)  
Nem elegendő a tár (5000--5999)  
Mátrixhiba (6000--6999)  
Fájlkezelési hiba (7000--7999)  
Beolvasási/kiírási hiba (8000--8999)  
EXOS hiba (9000--9999)  
Vezérlési hiba (10000--10999)  
Grafikai hiba (11000--11999)  
Időzítési hiba (12000--12999)  
Szintaxishiba (20000--20999)  
Rendszerhiba (30000)

### Az ENTERPRISE angol nyelvű hibaüzenetei

Argument to CHRS out of range (4002)  
A CHRS argumentuma nem megfelelő  
Argument to TAB out of range (4005)  
A TAB argumentuma nem megfelelő  
Array identifier expected (20032)  
A tömbazonosító hiányzik  
Array subscript out of bounds (2001)  
Tartományon kívüli tömbindex  
BASIC data has been corrupted (30000)  
A rendszer változói megsérültek

Call not supported by this device (9231)

A hívás nem támogatott ezen az eszközön

Cannot do specified RENUMBER (20010)

Végrehajthatatlan RENUMBER

Cannot use both serial and network (9215)

Egyszerre nem lehet a soros vonalat és a hálózatot is hasz-  
nálni

Cassette CRC error (9208)

Kazetta-ellenőrzőkód hiba (CRC)

Channel already open (7003)

A csatorna már nyitva van

Channel does not exist (9251)

A csatorna nem létezik

Channel exist (9249)

A csatorna már létezik

Channel not open (7004)

A csatorna nincs nyitva

Command in program (20072)

Ez a parancs nem lehet a programban

CONTINUE not possible (20020)

Nem folytatható

Copyright INTELLIGENT SOFTWARE LTD (9463)

Védelem felirat

Device does not exist (9250)

Az eszköz nem létezik

Device in use (9233)

Az eszköz használtban van

Division by zero (3001)

Nullával való osztás

Editor load file error (9466)

Editor -- fájlbeolvasási hiba

Editor load file too big (9209)

Editor -- a betöltendő fájl túl nagy

Editor keyboard channel error (9210)

Editor -- billentyűzetcsatorna hiba

Editor video channel error (9212)

Editor - videócsatorna hiba



End of file (9228)  
Hibás fájl vége jel

Envelope storage full (9224)  
Az ENVELOPE tár megtelt

Envelope too big (9225)  
Az ENVELOPE túl nagy

Error in CHAIN parameters (4301)  
Hiba a CHAIN paramétereiben

Error in DEF parameter (4000)  
Hiba a DEF paramétereiben

EXOS stack overflow (9252)  
A rendszererem túlcsordult

EXOS function call not allowed (9254)  
Az EXOS funkcióhívás nem engedélyezett

Function key string too long (9226)  
A funkcióbillentyű szövege túl hosszú

Identifier declared twice (20041)  
Kétszer deklarált változó

Identifier expected (20030)  
Az azonosító hiányzik

Identifier too long (20042)  
Az azonosító túl hosszú

Index to LBOUND out of range (4008)  
Az LBOUND-beli index hibás

Index to SIZE out of range (4004)  
A SIZE-beli index a megengedett tartományon kívül van

Index to UBOUND out of range (4009)  
Az UBOUND-beli index hibás

Insufficient ALLOCATE space (5120)  
Az ALLOCATE tár betelt

Insufficient extension space (5110)  
A bővítőtár betelt

Insufficient memory (5000)  
A tár betelt

Insufficient memory (9247)  
A tár betelt

Insufficient stack space (5100)  
A veremtár betelt

Insufficient video memory (9246)

A videótár betelt

Invalid argument to ASIN or ACOS (3007)

Hibás ASIN vagy ACOS argumentum

Invalid argument to LOG (3004)

Hibás LOG argumentum

Invalid argument to ORD (4003)

Hibás ORD argumentum

Invalid argument to SQR (3005)

Hibás SQR argumentum

Invalid beam position (9216)

Hibás rajzolópont-pozíció

Invalid channel number (7001)

Hibás csatornaszám

Invalid cursor coordinates (9217)

Hibás kurzorpozíció

Invalid date or time value (9235)

Hibás dátum- vagy időmegadás

Invalid device descriptor (9241)

Hibás eszközeíró

Invalid display parameters (9220)

Hibás képernyőparaméter

Invalid end of block (20051)

Hibás blokk vége jel

Invalid escape sequence (9230)

Hibás kilépési sorozat

Invalid EXOS function code (9255)

Hibás EXOS funkciókód

Invalid EXOS string (9253)

Hibás EXOS szöveg

Invalid file format (20080)

Hibás fájlformátum

Invalid file header (9239)

Hibás fájlfejléc

Invalid line number (20001)

Hibás sorszám

Invalid line number range (20002)

Hibás sortartomány

Invalid machine option use (20060)  
Hibás gépifeltétel-használat

Invalid multi-statement line (20074)  
Hibás többutasításos sor

Invalid relocatable module (9237)  
Hibás áthelyezhető modul

Invalid row number to scroll (9218)  
Hibás sorszám a képernyőléptetéshez

Invalid special function call (9234)  
Hibás a speciális funkció hívása

Invalid unit number (9232)  
Hibás egységszám

Invalid USING string (8201)  
Hibás USING karakterlánc

Invalid video mode (9221)  
Hibás videó üzemmód

Invalid video page size (9222)  
Hibás videólapméret

Line number does not exist (20004)  
A sorszám nem létezik

Line too long (20075)  
A sor túl hosszú

Missing closing quotes (20043)  
Hiányzik a lezáró idézőjel

Missing end of block (20050)  
Hiányzik a blokk vége jel

Network address not set (9214)  
A hálózati cím nincs beállítva

Network link exists (9213)  
A hálózat összekapcsolása már létezik

No CASE selected (10004)  
Nincs CASE ág

No format item in USING string (8202)  
USING-ban nem csak formátumelem van

Not understood (20000)  
Nem érthető

Numeric data expected (8101)  
Számadat kell

Out of data in READ/INPUT (8001)  
Nincs elég adat a READ/INPUT-hoz

Overflow in numeric constant (1001)  
Túlcsordulás a számkonstansban

Overflow in numeric expression (1002)  
Túlcsordulás a számkifejezésben

Overflow in string assignment (ie. string too long) (1106)  
Túlcsordulás az értékadásban (pl. a karakterlánc túl hosszú)

Overflow in string expression (1051)  
Túlcsordulás a szövegkifejezésben

Program does not exist (10005)  
A program nem létezik

Programs do not VERIFY (20081)  
A program nem azonos

Protection violation (9227)  
A tárvédelem megsértése

Return without GOSUB (10002)  
RETURN GOSUB nélkül

Sound queue full (9223)  
A hangtár megtelt

Statement in immediate mode (20071)  
Ez az utasítás nem lehet parancs üzemmódban

Statement not allowed after THEN (20073)  
A THEN után nem állhat ilyen utasítás

STOP key pressed (9229)  
A STOP billentyű lenyomva

String identifier expected (20031)  
A karaktersorozat-azonosító hiányzik

Too many nested blocks (20052)  
Túl sok az egymásba ágyazott blokk

TRACE channel not open (7401)  
A TRACE csatorna nincs nyitva

Type mismatch (20034)  
Típuseltérés

Unexpected value given (1000)  
A megadott érték nem megfelelő

Unknown EXOS variable number (9242)  
Ismeretlen EXOS rendszerváltozó

Unknown module type (9238)

Ismeretlen modul típus

Unrecognised command string (9240)

Ismeretlen parancs

USING format item too short (8203)

A USING formátumelem túl rövid

Variable not initialised (20040)

A változónak nincs értéke

Az Enterprise német nyelvű hibaüzenetei

ALLOCATE-Speicher zu klein (5120)

Az ALLOCATE tár betelt

Anführungszeichen nicht geschlossen (20043)

Hiányzik a lezáró idézőjel

Aufruf dieses Gerates nicht möglich (9231)

A hívás nem támogatott ezen az eszközön

Basic-Interpreter Schadhaf (30000)

A rendszer változói megsérültek

Befehl nach THEN ungöltig (20073)

A THEN után nem állhat ilyen utasítás

Bildschirm-Modus auserhalb des erlaubten Bereichs (9221)

Hibás videó üzemmód

Bildschirm-Format auserhalb des erlaubten Bereichs (9222)

Hibás videólappméret

Blockende fehlt (20050)

Hiányzik a blokk vége jel

CHAIN-Parameter-Fehler (4301)

Hiba a CHAIN paraméterekben

CHRS-Argument auserhalb des erlaubten Bereichs (4002)

A CHRS argumentuma nem megfelelő

CONTINUE nicht möglich (20020)

Nem folytatható

Copyright INTELLIGENT SOFTWARE LTD (9207)

Védelem felirat

Cursor-Koordinaten auserhalb des erlaubten Bereichs (9217)

Hibás kurzorpozíció

Datei-Ende (9228)  
Hibás adat vége jel

Datum oder Uhrzeit ausserhalb des erlaubten Bereichs (9235)  
Hibás dátum- vagy időmegadás

Direktmodus-Befehl im Programm unzulässig (20072)  
Ez a parancs nem lehet a programban

Display-Parameter ausserhalb des erlaubten Bereichs (9220)  
Hibás képernyőparaméter

Division durch Null (3001)  
Nullával való osztás

Editor: Tastatur-Kanal-Fehler (9211)  
Editor -- billentyűzetcsatorna hiba

Editor: Video-Kanal-Fehler (9212)  
Editor -- videócsatorna-hiba

Editor: Überlange der zu ladenden Datei (9209)  
Editor -- a betöltendő fájl túl nagy

Ende der Modul-Datei (9236)  
Adatállomány vége

EXOS (betreibssystem) (9243)  
EXOS operációsrendszer hiba

EXOS-Stapel-Speicher Überlauf (9252)  
A rendszerverem túlcsoordult

Fehler in der zu ladenden Datei (9210)  
Fájlbeolvasási hiba

Fehler in der zu ladenden Datei (9466)  
Fájlbeolvasási hiba

Feld ausserhalb des erlaubten Bereichs (2001)  
Tartományon kívüli tömbindex

Feld-Identifikator wird erwartet (20032)  
A tömbazonosító hiányzik

Funktionstasten-String ausserhalb des erlaubten Bereichs (9226)  
A funkcióbillentyű szövege túl hosszú

Geräte-Nummer ausserhalb des erlaubten Bereichs (9232)  
Hibás egységszám

Gerät nicht vorhanden (9250)  
Az eszköz nem létezik

Gerät Wird schon benutzt (9233)  
Az eszköz már használatban van

Grafik-Cursor-Koordinaten ausserhalb des erlaubten Bereichs  
(9216)  
Hibás rajzolópont-pozíció  
Hüllkurve ausserhalb des erlaubten Bereichs (9225)  
Az ENVELOPE túl nagy  
Hüllkurvenstapel voll (9224)  
Az ENVELOPE tár megtelt  
Identifikator doppelt initialisiert (20041)  
Kétszer deklarált változó  
Identifikator zu lang (20042)  
Az azonosító túl hosszú  
Identifikator wird erwartet (20030)  
Azonosító hiányzik  
Kanal nicht offen (7004)  
A csatorna nincs nyitva  
Kanal nicht vorhanden (9251)  
A csatorna nem létezik  
Kanal schon offen (7003)  
A csatorna már nyitva van  
Kanal schon vorhanden (9249)  
A csatorna már létezik  
Kassetten-Lese-Fehler (9208)  
Kazetta-ellenőrzőkód hiba (CRC)  
Keine Daten mehr für READ/INPUT (8001)  
Nincs elég adat a READ/INPUT-hoz  
keine Format-Angabe im USING-String (8202)  
USING-ban nem csak formátumelem van  
kein CASE angegeben (10004)  
Nincs CASE ág  
LBOUND-Index ausserhalb des erlaubten Bereichs (4008)  
Az LBOUND-beli index hibás  
Modul nicht verlegbar (9237)  
Hibás áthelyezhető modul  
Netzwerk-Adresse nicht gesetzt (9214)  
A hálózati cím nincs beállítva  
Netzwerk-Verbindung schon vorhanden (9213)  
A hálózat összekapcsolása már létezik

Nicht verstanden (20000)

Nem érthető

numerische Daten erwartet (8101)

Számadat kell

Programm-Befehl im Direktmodus unzulässig (20071)

Ez az utasítás nem lehet parancs üzemmódban

Programme stimmen nicht überein (20081)

A program nem azonos

Program vorhanden nicht (10005)

A program nem létezik

RENUMBER kann nicht ausgeführt werden (20010)

Nem végrehajtható RENUMBER utasítás

RETURN ohne GOSUB (10002)

RETURN GOSUB nélkül

Schnittstelle schon belegt (9215)

Egyszerre nem lehet a soros vonalat és a hálózatot is használni

Schutz-Verletzung (9227)

A tárvédelem megsértése

SIZE-Index ausserhalb des erlaubten Bereichs (4004)

SIZE-beli index a megengedett tartományon kívül

Speichererweiterung zu klein (5110)

A bővítőtár betelt

Speicherplatz zu klein (5000)

A tár betelt

Speicherplatz zu klein (9247)

A tár betelt

Stapel-Speicher zu klein (5100)

A veremtár betelt

STOP-Taste gedrückt (9229)

A STOP billentyű lenyomva

String-Identifikator wird erwartet (20031)

A karaktersorozat-azonosító hiányzik

TAB-Argument ausserhalb des erlaubten Bereichs (4005)

A TAB argumentuma nem megfelelő

Ton-Stapel voll (9223)

A hangtár megtelt



TRACE-Kanal nicht offen (7401)  
A TRACE csatorna nincs nyitva  
Typen-Verwechslung (20034)  
Típuseltérés  
UBOUND-Index ausserhalb des erlaubten Bereichs (4009)  
Az UBOUND-beli index hibás  
unbekannte EXOS-Variablen-Number (9242)  
Ismeretlen EXOS rendszerváltozó  
unbekannte Modulart (9238)  
Ismeretlen modultípus  
unbekannter Befehls-String (9240)  
Ismeretlen parancs  
unerwarteter Wert eingegeben (1000)  
A megadott érték nem megfelelő  
ungültige Anweisung an das Betriebssystem (20060)  
Hibás gépifeltétel-használat  
ungültige Bildschirmseiten-Datei (9219)  
Hibás videólapfájl  
ungültige DEF-Parameter (4000)  
Hiba a DEF paraméterekben  
ungültige Gerat-Bezeichnung (9241)  
Hibás eszközeíró  
ungültige Kanal-Nummer (7001)  
Hibás csatornaszám  
ungültige Mehr-Befehls-Zeite (20074)  
Hibás többutasításos sor  
ungültiger Aufruf einer Spezialfunktion (9234)  
Hibás a speciális funkció hívása  
ungültiger Datei-Anfang (9239)  
Hibás fájlfejléc  
ungültiger EXOS Funktions-Code (9255)  
Hibás EXOS funkciókód  
ungültiger EXOS-String (9253)  
Hibás EXOS szöveg  
ungültiger USING-String (8201)  
Hibás USING karakterlánc  
ungültiger Zeilennummer-Bereich (20002)  
Hibás sortartomány

ungültiges Argument für ASIN oder ACOS (3007)

Hibás ASIN vagy ACOS argumentum

ungültiges Argument für LOG (3004)

Hibás LOG argumentum

ungültiges Argument für ORD (4003)

Hibás ORD argumentum

ungültiges Argument für SQR (3005)

Hibás SQR argumentum

ungültiges Blockende (20051)

Hibás blokk vége jel

ungültiges Datei-Format (20080)

Hibás fájlformátum

ungültiges Steuerzeichen (9230)

Hibás kilépési sorozat

USING-Format-Angabe zu kurz (8203)

A USING formátumelem túl hosszú

Überlauf der String-Zuweisung (1106)

Túlcsordulás az értékadásban (pl. a karakterlánc túl hosszú)

Überlauf des String-Ausdrucks (1051)

Túlcsordulás a szövegkifejezésben

Überlauf numerischen Ausdrucks (1002)

Túlcsordulás a számkifejezésben

Überlauf numerischer Konstante (1001)

Túlcsordulás a számkonstansban

Variable nicht initialisiert (20040)

A változónak nincs kezdeti értéke

verbotener EXOS-Funktionsaufruf (9254)

Az EXOS funkció hívás nem engedélyezett

Video-Speicher-Platz zu klein (9246)

A videótár betelt

Zeile zu lang (20075)

A sor túl hosszú

Zeilennummer ausserhalb des erlaubten Bereichs (20001)

Hibás sorszám

Zeilennummer ist nicht vorhanden (20004)

zu scrollender Zeilenbereich ausserhalb des erlaubten Bereichs (9218)

Hibás sorszám a képernyőléptetéshez

zu viele Blocks ineinander verschachtelt (20052)

Túl sok egymásba ágyazott blokk

BASIC alapszavak

ABS  
ACOS  
ANGLE  
ALLOCATE  
ASIN  
ASK  
ATN  
ATTRIBUTES  
AUTO  
  
BEAM  
BIAS  
BIN  
BLACK  
BLUE  
BORDER  
  
CALL  
CAPTURE  
CASE  
CAUSE EXCEPTION  
CEIL  
CHAIN  
CHARACTER  
CHR\$  
CLEAR  
CLOSE  
CODE  
COLOUR  
CONTINUE  
COPY  
COS  
COSH  
COT  
CSC  
CURSOR  
CYAN  
  
DATA  
DATE  
DATE\$  
DEF  
DEFAULT CHANNEL  
DEG  
DELETE  
DIM  
DISPLAY  
DO  
  
EDIT  
EDITOR BUFFER  
EDITOR KEY  
EDITOR VIDEO  
ELSE

END  
ENVELOPE  
EPS  
EXIT  
EXIT HANDLER  
EXLINE  
EXP  
EXSTRING\$  
EXT  
EXTYPE  
  
FAST SAVE  
FKEY  
FLUSH  
FOR  
FP  
FREE  
  
GET  
GOSUB  
GOTO  
GRAPHICS  
GREEN  
  
HANDLER  
HEX\$  
  
IF  
IMAGE  
IN  
INF  
INFO  
INK  
INKEY\$  
INPUT  
INT  
INTERRUPT  
IP  
  
JOY  
  
KEY CLICK  
KEY DELAY  
KEY RATE  
  
LBOUND(a)  
LBOUND(a,n)  
LCASE\$  
LEN  
LET  
LINE MODE  
LINE INPUT  
LINE STYLE  
LIST  
LLIST  
LOAD  
LOG(X)

LOG10(X)  
LOG2(X)  
LOOK  
LOOP  
LPRINT  
LTRIM\$  
  
MAGENTA  
MAX  
MAXLEN  
MERGE  
MIN  
MOD  
  
NET CHANNEL  
NET MACHINE  
NET NUMBER  
NEW  
NEW ALL  
NEXT  
NUMERIC  
  
ON  
OPEN  
OPTION  
OR  
ORD  
OUT  
  
PALETTE  
PAPER  
PEEK  
PI  
PING  
PLOT  
POKE  
POS  
PRINT  
PROGRAM  
  
RAD  
RANDOMIZE  
READ  
REDIRECT  
REM  
REM1  
REM2  
REM(A,B)  
RENUNBER  
RESTORE  
RETRY  
RETURN  
RGB  
RND  
RND(X)  
ROUND  
RTRIM\$  
RUN

SAVE  
SCROLL  
SEC  
SELECT  
SERIAL BAUD  
SERIAL FORMAT  
SET  
SGN  
SIN  
SINH  
SIZE(a)  
SIZE(a,n)  
SOUND  
SOUND BUFFER  
SOUND STYLE  
SPEAKER  
SPEEK  
SPOKE  
SQR  
START  
STATUS  
STOP  
STRING  
STR\$  
  
TAB  
TAN  
TANH  
TAPE LEVEL  
TAPE SOUND  
TEXT  
THEN  
TIME  
TIMER  
TIME\$  
TRACE  
TRUNCATE  
TYPE  
  
UBOUND(a)  
UBOUND(a,n)  
UCASE\$  
USR  
  
VAL  
VERIFY  
VIDEO COLOUR  
VIDEO MODE  
VIDEO X  
VIDEO Y  
  
WAIT DELAY  
WHEN  
WHITE  
WORD\$  
  
YELLOW

ASCII	Német	Angol	ASCII	Német	Angol	ASCII	Német	Angol
32	space	space	80	P	P	128	ø	ø
33	!	!	81	O	Q	129	Ï	Ï
34	"	"	82	R	R	130	Ä	Ä
35	#	£	83	S	S	131	Ö	Ö
36	S	S	84	T	T	132	Ú	Ú
37	%	%	85	U	U	133	Á	Á
38	&	&	86	V	V	134	σ	σ
39	'	'	87	W	W	135	Ft	Ft
40	(	(	88	X	X	136	é	é
41	)	)	89	Y	Y	137	£	£
42	*	*	90	Z	Z	138	à	à
43	+	+	91	Ä	[	139	†	†
44	,	,	92	Ö	\	140	→	→
45	-	-	93	Ü	]	141	Ω	Ω
46	.	.	94	^	^	142	■	■
47	/	/	95	-	-	143	∫	∫
48	0	0	96	,	,	144	π	π
49	1	1	97	a	a	145	Σ	Σ
50	2	2	98	b	b	146	ä	ä
51	3	3	99	c	c	147	ö	ö
52	4	4	100	d	d	148	ü	ü
53	5	5	101	e	e	149	à	à
54	6	6	102	f	f	150	σ	σ
55	7	7	103	g	g	151	2	2
56	8	8	104	h	h	152	e	e
57	9	9	105	i	i	153	§	§
58	:	:	106	j	j	154	‡	‡
59	;	;	107	k	k	155	↓	↓
60	<	<	108	l	l	156	←	←
61	=	=	109	m	m	157	μ	μ
62	>	>	110	n	n	158	←	←
63	?	?	111	o	o	159	■	■
64	§	e	112	p	p			
65	A	A	113	q	q			
66	B	B	114	r	r			
67	C	C	115	s	s			
68	D	D	116	t	t			
69	E	E	117	u	u			
70	F	F	118	v	v			
71	G	G	119	w	w			
72	H	H	120	x	x			
73	I	I	121	y	y			
74	J	J	122	z	z			
75	K	K	123	ä	{			
76	L	L	124	ö	i			
77	M	M	125	Ü	}			
78	N	N	126	ß	~			
79	O	O	127	€	€			

HEX.	DEC.	Z80	HEX.	DEC.	Z80
00	000	NOP	33	051	INC SP
01	001	LD BC,NN	34	052	INC (HL)
02	002	LD (BC),A	35	053	DEC (HL)
03	003	INC BC	36	054	LD (HL),N
04	004	INC B	37	055	SCF
05	005	DEC B	38	056	JR C,disp
06	006	LD B,N	39	057	ADD HL,SP
07	007	RLCA	3A	058	LD A,(NN)
08	008	EX AF,AF'	3B	059	DEC SP
09	009	ADD HL,BC	3C	060	INC A
0A	010	LD A,(BC)	3D	061	DEC A
0B	011	DEC BC	3E	062	LD A,N
0C	012	INC C	3F	063	CCF
0D	013	DEC C	40	064	LD B,B
0E	014	LD C,N	41	065	LD B,C
0F	015	RRCA	42	066	LD B,D
10	016	DJNZ disp	43	067	LD B,E
11	017	LD DE,NN	44	068	LD B,H
12	018	LD (DE),A	45	069	LD B,L
13	019	INC DE	46	070	LD B,(HL)
14	020	INC D	47	071	LD B,A
15	021	DEC D	48	072	LD C,B
16	022	LD D,N	49	073	LD C,C
17	023	RLA	4A	074	LD C,D
18	024	JR disp	4B	075	LD C,E
19	025	ADD HL,DE	4C	076	LD C,H
1A	026	LD A,(DE)	4D	077	LD C,L
1B	027	DEC DE	4E	078	LD C,(HL)
1C	028	INC E	4F	079	LD C,A
1D	029	DEC E	50	080	LD D,B
1E	030	LD E,N	51	081	LD D,C
1F	031	RRA	52	082	LD D,D
20	032	JR NZ,disp	53	083	LD D,E
21	033	LD HL,NN	54	084	LD D,H
22	034	LD (NN),HL	55	085	LD D,L
23	035	INC HL	56	086	LD D,(HL)
24	036	INC H	57	087	LD D,A
25	037	DEC H	58	088	LD E,B
26	038	LD H,N	59	089	LD E,C
27	039	DAA	5A	090	LD E,D
28	040	JR Z,disp	5B	091	LD E,E
29	041	ADD HL,HL	5C	092	LD E,H
2A	042	LD HL,(NN)	5D	093	LD E,L
2B	043	DEC HL	5E	094	LD E,(HL)
2C	044	INC L	5F	095	LD E,A
2D	045	DEC L	60	096	LD H,B
2E	046	LD L,N	61	097	LD H,C
2F	047	CPL	62	098	LD H,D
30	048	JR NC,disp	63	099	LD H,E
31	049	LD SP,NN	64	100	LD H,H
32	050	LD (NN),A	65	101	LD H,L

HEX.	DEC.	Z80	HEX.	DEC.	Z80
66	102	LD H,(HL)	9C	156	SBC A,H
67	103	LD H,A	9D	157	SBC A,L
68	104	LD L,B	9E	158	SBC A,(HL)
69	105	LD L,C	9F	159	SBC A,A
6A	106	LD L,D	A0	160	AND B
6B	107	LD L,E	A1	161	AND C
6C	108	LD L,H	A2	162	AND D
6D	109	LD L,L	A3	163	AND E
6E	110	LD L,(HL)	A4	164	AND H
6F	111	LD L,A	A5	165	AND L
70	112	LD (HL),B	A6	166	AND (HL)
71	113	LD (HL),C	A7	167	AND A
72	114	LD (HL),D	A8	168	XOR B
73	115	LD (HL),E	A9	169	XOR C
74	116	LD (HL),H	AA	170	XOR D
75	117	LD (HL),L	AB	171	XOR E
76	118	HALT	AC	172	XOR H
77	119	LD (HL),A	AD	173	XOR L
78	120	LD A,B	AE	174	XOR (HL)
79	121	LD A,C	AF	175	XOR A
7A	122	LD A,D	B0	176	OR B
7B	123	LD A,E	B1	177	OR C
7C	124	LD A,H	B2	178	OR D
7D	125	LD A,L	B3	179	OR E
7E	126	LD A,(HL)	B4	180	OR H
7F	127	LD A,A	B5	181	OR L
80	128	ADD A,B	B6	182	OR (HL)
81	129	ADD A,C	B7	183	OR A
82	130	ADD A,D	B8	184	CP B
83	131	ADD A,E	B9	185	CP C
84	132	ADD A,H	BA	186	CP D
85	133	ADD A,L	BB	187	CP E
86	134	ADD A,(HL)	BC	188	CP H
87	135	ADD A,A	BD	189	CP L
88	136	ADC A,B	BE	190	CP (HL)
89	137	ADC A,C	BF	191	CP A
8A	138	ADC A,D	C0	192	RET NZ
8B	139	ADC A,E	C1	193	POP BC
8C	140	ADC A,H	C2	194	JP NZ,NN
8D	141	ADC A,L	C3	195	JP NN
8E	142	ADC A,(HL)	C4	196	CALL NZ,NN
8F	143	ADC A,A	C5	197	PUSH BC
90	144	SUB B	C6	198	ADD A,N
91	145	SUB C	C7	199	RST 0
92	146	SUB D	C8	200	RET Z
93	147	SUB E	C9	201	RET
94	148	SUB H	CA	202	JP Z,NN
95	149	SUB L	CB	203	1. CB csoport
96	150	SUB (HL)	CC	204	CALL Z,NN
97	151	SUB A	CD	205	CALL NN
98	152	SBC A,B	CE	206	ADC A,N
99	153	SBC A,C	CF	207	RST 8
9A	154	SBC A,D	D0	208	RET NC
9B	155	SBC A,E	D1	209	POP DE

HEX.	DEC.	Z80	HEX.	DEC.	Z80
D2	210	JP NC,NN	05	005	RLC L
D3	211	OUT (N),A	06	006	RLC (HL)
D4	212	CALL NC,NN	07	007	RLC A
D5	213	PUSH DE	08	008	RRC B
D6	214	SUB N	09	009	RRC C
D7	215	RST 10H	0A	010	RRC D
D8	216	RET C	0B	011	RRC E
D9	217	EXX	0C	012	RRC H
DA	218	JP C,NN	0D	013	RRC L
DB	219	IN A,(N)	0E	014	RRC (HL)
DC	220	CALL C,NN	0F	015	RRC A
DD	221	1. DD csoport	10	016	RL B
DE	222	SBC A,N	11	017	RL C
DF	223	RST 18H	12	018	RL D
EO	224	RET PD	13	019	RL E
E1	225	POP HL	14	020	RL H
E2	226	JP PD,NN	15	021	RL L
E3	227	EX (SP),HL	16	022	RL (HL)
E4	228	CALL PD,NN	17	023	RL A
E5	229	PUSH HL	18	024	RR B
E6	230	AND N	19	025	RR C
E7	231	RST 20H	1A	026	RR D
E8	232	RET PE	1B	027	RR E
E9	233	JP (HL)	1C	028	RR H
EA	234	JP PE,NN	1D	029	RR L
EB	235	EX DE,HL	1E	030	RR (HL)
EC	236	CALL PE,NN	1F	031	RR A
ED	237	1. ED csoport			
EE	238	XOR N	20	032	SLA B
EF	239	RST 28H	21	033	SLA C
FO	240	RET P	22	034	SLA D
F1	241	POP AF	23	035	SLA E
F2	242	JP P,NN	24	036	SLA H
F3	243	DI	25	037	SLA L
F4	244	CALL P,NN	26	038	SLA (HL)
F5	245	PUSH AF	27	039	SLA A
F6	246	OR N	28	040	SRA B
F7	247	RST 30H	29	041	SRA C
F8	248	RET M	2A	042	SRA D
F9	249	LD SP,HL	2B	043	SRA E
FA	250	JP M,NN	2C	044	SRA H
FB	251	EI	2D	045	SRA L
FC	252	CALL M,NN	2E	046	SRA (HL)
FD	253	1. FD csoport	2F	047	SRA A
FE	254	CP N	30	048	SLL B
FF	255	RST 38H	31	049	SLL C
			32	050	SLL D
			33	051	SLL E
			34	052	SLL H
			35	053	SLL L
			36	054	SLL (HL)
			37	055	SLL A
			38	056	SRL B
			39	057	SRL C
CB csoport					
00	000	RLC B			
01	001	RLC C			
02	002	RLC D			
03	003	RLC E			
04	004	RLC H			

<u>HEX.</u>	<u>DEC.</u>	<u>Z80</u>	<u>HEX.</u>	<u>DEC.</u>	<u>Z80</u>
3A	058	SRL D	70	112	BIT 6,B
3B	059	SRL E	71	113	BIT 6,C
3C	060	SRL H	72	114	BIT 6,D
3D	061	SRL L	73	115	BIT 6,E
3E	062	SRL (HL)	76	116	BIT 6,H
3F	063	SRL A	75	117	BIT 6,L
40	064	BIT 0,B	77	118	BIT 6,(HL)
41	065	BIT 0,C	77	119	BIT 6,A
42	066	BIT 0,D	78	120	BIT 7,B
43	067	BIT 0,E	79	121	BIT 7,C
44	068	BIT 0,H	7A	122	BIT 7,D
45	069	BIT 0,L	7B	123	BIT 7,E
46	070	BIT 0,(HL)	7C	124	BIT 7,H
47	071	BIT 0,A	7D	125	BIT 7,L
48	072	BIT 1,B	7E	126	BIT 7,(HL)
49	073	BIT 1,C	7F	127	BIT 7,A
4A	074	BIT 1,D	80	128	RES 0,B
4B	075	BIT 1,E	81	129	RES 0,C
4C	076	BIT 1,H	82	130	RES 0,D
4D	077	BIT 1,L	83	131	RES 0,E
4E	078	BIT 1,(HL)	84	132	RES 0,H
4F	079	BIT 1,A	85	133	RES 0,L
50	080	BIT 2,B	86	134	RES 0,(HL)
51	081	BIT 2,C	87	135	RES 0,A
52	082	BIT 2,D	88	136	RES 1,B
53	083	BIT 2,E	89	137	RES 1,C
54	084	BIT 2,H	8A	138	RES 1,D
55	085	BIT 2,L	8B	139	RES 1,E
56	086	BIT 2,(HL)	8C	140	RES 1,H
57	087	BIT 2,A	8D	141	RES 1,L
58	088	BIT 3,B	8E	142	RES 1,(HL)
59	089	BIT 3,C	8F	143	RES 1,A
5A	090	BIT 3,D	90	144	RES 2,B
5B	091	BIT 3,E	91	145	RES 2,C
5C	092	BIT 3,H	92	146	RES 2,D
5D	093	BIT 3,L	93	147	RES 2,E
5E	094	BIT 3,(HL)	94	148	RES 2,H
5F	095	BIT 3,A	95	149	RES 2,L
60	096	BIT 4,B	96	150	RES 2,(HL)
61	097	BIT 4,C	97	151	RES 2,A
62	098	BIT 4,D	98	152	RES 3,B
63	099	BIT 4,E	99	153	RES 3,C
64	100	BIT 4,H	9A	154	RES 3,D
65	101	BIT 4,L	9B	155	RES 3,E
66	102	BIT 4,(HL)	9C	156	RES 3,H
67	103	BIT 4,A	9D	157	RES 3,L
68	104	BIT 5,B	9E	158	RES 3,(HL)
69	105	BIT 5,C	9F	159	RES 3,A
6A	106	BIT 5,D	A0	160	RES 4,B
6B	107	BIT 5,E	A1	161	RES 4,C
6C	108	BIT 5,H	A2	162	RES 4,D
6D	109	BIT 5,L	A3	163	RES 4,E
6E	110	BIT 5,(HL)	A4	164	RES 4,H
6F	111	BIT 5,A	A5	165	RES 4,L





HEX.	DEC.	Z80	HEX.	DEC.	Z80
50	080	IN D,(C)	45	69	RETN
51	081	OUT (C),D	46	70	IM 0
52	082	SBC HL,DE	47	71	LD I,A
53	083	LD (nn),DE	48	72	IN C,(C)
56	086	IM 1	49	73	OUT (C),C
57	087	LD A,I	4A	74	ADC HL,BC
58	088	IN E,(C)	4B	75	LD BC,(nn)
59	089	OUT (C),E	4D	77	RETI
5A	090	ADC HL,DE	4F	79	LD R,A
5B	091	LD DE,(nn)	50	80	IN D,(C)
5E	094	IM 2	51	81	OUT (C),D
5F	095	LD A,R	52	82	SBC HL,DE
60	096	IN H,(C)	53	83	LD (nn),DE
61	097	OUT (C),H	56	86	IM 1
62	098	SBC HL,HL	57	87	LD A,I
63	099	LD (nn),HL	58	88	IN E,(C)
67	103	RRD	59	89	OUT (C),E
68	104	IN L,(C)	5A	90	ADC HL,DE
69	105	OUT (C),L	5B	91	LD DE,(nn)
6A	106	ADC HL,HL	5E	94	IM 2
6B	107	LD HL,(nn)	5F	95	LD A,R
6F	111	RLD	60	96	IN H,(C)
70	112	IN F,(C)	61	97	OUT (C),H
71	113	OUT (C),F	62	98	SBC HL,HL
72	114	SBC HL,SP	63	99	LD (nn),HL
73	115	LD (nn),SP	67	103	RRD
78	120	IN A,(C)	68	104	IN L,(C)
79	121	OUT (C),A	69	105	OUT (C),L
7A	122	ADC HL,SP	6A	106	ADC HL,HL
7B	123	LD SP,(nn)	6B	107	LD HL,(nn)
A0	160	LDI	6F	111	RLD
A1	161	CPI	70	112	IN F,(C)
A2	162	INI	72	114	SBC HL,SP
A3	163	OUTI	73	115	LD (nn),SP
A8	168	LDD	78	120	IN A,(C)
A9	169	CPD	79	121	OUT (C),A
AA	170	IND	7A	122	ADC HL,SP
AB	171	OUTD	7B	123	LD SP,(nn)
B0	176	LDIR	A0	160	LDI
B1	177	CPIR	A1	161	CPI
B2	178	INIR	A2	162	INI
B3	179	OTIR	A3	163	OUTI
B8	184	LDDR	A8	168	LDD
B9	185	CPDR	A9	169	CPD
BA	186	INDR	AA	170	IND
BB	187	OTDR	AB	171	OUTD
			B0	176	LDIR
			B1	177	CPIR
			B2	178	INIR
			B3	179	OTIR
			B8	184	LDDR
			B9	185	CPDR
			BA	186	INDR
			BB	187	OTDR

EO csoport		
HEX.	DEC.	Z80
40	64	IN B,(C)
41	65	OUT (C),B
42	66	SBC HL,BC
43	67	LD (nn),BC
44	68	NEG

HEX.	DEC.	Z80	HEX.	DEC.	Z80
		FD csoport			
09	009	ADD i,BC	76	118	LD (i+d),A
19	025	ADD i,DE	7C	124	LD A,ih
21	033	LD i,nn	7D	125	LD A,il
22	034	ID (nn),i	7E	126	LD A,(i+d)
23	035	INC i	84	132	ADD A,ih
24	036	INC ih	85	133	ADD A,il
25	037	DEC ih	86	134	ADD A,(i+d)
26	038	LD ih,n	8C	140	ADC A,ih
29	041	ADD i,i	8D	141	ADC A,il
2A	042	LD i,(nn)	8E	142	ADC A,(i+d)
2B	043	DEC i	94	148	SUB ih
2C	044	INC il	95	149	SUB il
2D	045	DEC il	96	150	SUB (i+d)
2E	046	LD il,n	9C	156	SBC A,ih
34	052	INC (i+d)	9D	157	SBC A,il
35	053	DEC (i+d)	9E	158	SBC A,(i+d)
36	054	LD (i+d),n	A4	164	AND ih
39	057	ADD i,SP	A5	165	AND il
44	068	LD B,ih	A6	166	AND (i+d)
45	069	LD B,il	AC	172	XOR ih
46	070	LD B,(i+d)	AB	173	XOR il
4C	076	LD C,ih	AD	174	XOR (i+d)
4D	077	LD C,il	B4	180	OR ih
4E	078	LD C,(i+d)	B5	181	OR il
54	084	LD D,ih	B6	182	OR (i+d)
55	085	LD D,il	BC	188	CP ih
56	087	LD D,(i+d)	BD	189	CP il
5C	092	LD E,ih	BE	190	CP (i+d)
5D	093	LD E,il	E1	225	POP i
5E	094	LD E,(i+d)	E3	227	EX (SP),i
60	096	LD ih,B	E5	229	PUSH i
61	097	LD ih,C	E9	233	JP (i)
62	098	LD ih,D	F9	249	LD SP,i
63	099	LD ih,E			
64	100	LD ih,ih			
65	101	LD ih,il			
66	102	LD H,(i+d)			
67	103	LD ih,A			
68	104	LD il,B			
69	105	LD il,C			
6A	106	LD il,D			
6B	107	LD il,E			
6C	108	LD il,ih			
6D	109	LD il,il			
6E	110	LD L,(i+d)			
6F	111	LD il,A			
70	112	LD (i+d),B			
71	113	LD (i+d),C			
72	114	LD (i+d),D			
73	115	LD (i+d),E			
74	116	LD (i+d),H			
75	117	LD (i+d),L			

Z80	HEX.	DEC.	Z80	HEX.	DEC.
ADC A,A	8F	143	DAA	27	039
ADC A,B	88	136	DEC A	3D	061
ADC A,C	89	137	DEC B	05	005
ADC A,D	8A	138	DEC BC	08	011
ADC A,E	8B	139	DEC C	0D	013
AOC A,H	8C	140	DEC DE	1B	027
ADC A,L	8D	141	DEC D	15	021
ADC A,N	CE	206	DEC E	10	029
ADC A,(HL)	8E	142	DEC H	25	037
ADD A;A	87	135	DEC HL	28	043
ADD A,B	80	128	DEC (HL)	35	053
ADD A,C	81	129	DEC L	2D	045
ADD A,D	82	130	DEC SP	38	059
ADD A,E	83	131	DI	F3	243
ADD A,H	84	132	DJNZ disp	10	016
ADD A,L	85	133	EI	FB	251
ADD A,N	C6	198	EX AF,AF'	08	008
ADD A,(HL)	86	134	EX DE,HL	EB	235
ADD HL,BC	09	009	EX (SP),HL	E3	227
ADD HL,DE	19	025	EXX	D9	217
ADD HL,HL	29	041	HALT	76	118
ADD HL,SP	39	057	IN A,(N)	DB	219
AND A	A7	167	INC A	3C	060
AND B	A0	160	INC B	04	004
AND C	A1	161	INC BC	03	003
AND D	A2	162	INC C	0C	012
AND E	A3	163	INC D	14	020
AND H	A4	164	INC DE	13	019
AND L	A5	165	INC E	1C	028
AND N	E6	230	INC H	24	036
AND (HL)	A6	166	INC HL	23	035
CALL C,NN	DC	220	INC (HL)	34	052
CALL M,NN	FC	252	INC L	2C	044
CALL NC,NN	D4	212	INC SP	33	051
CALL NN	CD	205	JP C,NN	DA	218
CALL NZ,NN	C4	196	JP (HL)	E9	233
CALL P,NN	F4	244	JP M,NN	FA	250
CALL PE,NN	EC	236	JP NC,NN	D2	210
CALL PO,NN	E4	228	JP NN	C3	195
CALL Z,NN	CC	204	JP NZ,NN	C2	194
CCF	3F	063	JP P,NN	F2	242
CP A	BF	191	JP PE,NN	EA	234
CP B	B8	184	JP PO,NN	E2	226
CP C	B9	185	JP Z,NN	CA	202
CP D	BA	186	JR C,disp	38	056
CP E	BB	187	JR disp	18	024
CP H	BC	188	JR NC,disp	30	048
CP L	BD	189	JR NZ,disp	20	032
CP (HL)	BE	190	JR Z,disp	28	040
CP A,N	FE	254	LD A,	3E	062
CPL	2F	047	LD A,A	7F	127

Z80	HEX.	DEC.	Z80	HEX.	DEC.
LD A,B	78	120	LD H,(HL)	66	102
LD A,C	79	121	LD L,A	6F	111
LD A,D	7A	122	LD L,B	68	104
LD A,E	7B	123	LD L,C	69	105
LD A,H	7C	124	LD L,D	6A	106
LD A,L	7D	125	LD L,E	6B	107
LD A,(BC)	0A	010	LD L,H	6C	108
LD A,(DE)	1A	026	LD L,L	6D	109
LD A,(HL)	7E	126	LD L,N	2E	046
LD A,(NN)	3A	058	LD L,(HL)	6E	110
LD B,A	47	071	LD BC,NN	01	001
LD B,B	40	064	LD (BD),A	02	002
LD B,C	41	065	LD DE,NN	11	017
LD B,D	42	066	LD (DE),A	12	018
LD B,E	43	067	LD HL,NN	21	033
LD B,H	44	068	LD HL,(NN)	2A	042
LD B,L	45	069	LD (HL),A	77	119
LD B,N	06	006	LD (HL),B	70	112
LD B,(HL)	46	070	LD (HL),C	71	113
LD C,A	4F	079	LD (HL),D	72	114
LD C,B	48	072	LD (HL),E	73	115
LD C,C	49	073	LD (HL),H	74	116
LD C,D	4A	074	LD (HL),L	75	117
LD C,E	4B	075	LD (HL),N	36	054
LD C,H	4C	076	LD (NN),A	32	050
LD C,L	4D	077	LD (NN),HL	22	034
LD C,N	0E	014	LD SP,HL	F9	249
LD C,(HL)	4E	078	LD SP,NN	31	049
LD D,A	57	087	NOP	00	000
LD D,B	50	080	OR A	B7	183
LD D,C	51	081	OR B	B0	176
LD D,D	52	082	OR C	B1	177
LD D,E	53	083	OR D	B2	178
LD D,H	54	084	OR E	B3	179
LD D,L	55	085	OR H	B4	180
LD D,N	16	022	OR L	B5	181
LD D,(HL)	56	086	OR N	F6	246
LD E,A	5F	095	OR (HL)	B6	182
LD E,B	58	088	OUT (N),A	D3	211
LD E,C	59	089	POP AF	F1	241
LD E,D	5A	090	POP BC	C1	193
LD E,E	5B	091	POP DE	D1	209
LD E,H	5C	092	POP HL	E1	225
LD E,L	5D	093	PUSH BC	C5	197
LD E,N	1E	030	PUSH DE	D5	213
LD E,(HL)	5E	094	PUSH HL	E5	229
LD H,A	67	103	PUSH AF	F5	245
LD H,B	60	096	RET	C9	201
LD H,C	61	097	RET C	D8	216
LD H,D	62	098	RET NC	D0	208
LD H,E	63	099	RET NZ	C0	192
LD H,H	64	100	RET M	F8	248
LD H,L	65	101	RET P	F0	240
LD H,N	26	038	RET PE	E8	232

Z80	HEX.	DEC.	Z80	HEX.	DEC.
RET PO	E0	224	BIT 0,H	44	068
RET Z	C8	200	BIT 0,L	45	069
RLA	17	023	BIT 0,(HL)	46	070
RLCA	07	007	BIT 1,A	4F	079
RRA	1F	031	BIT 1,B	48	072
RRCA	0F	015	BIT 1,C	49	073
RST 0	C7	199	BIT 1,D	4A	074
RST 8	CF	207	BIT 1,E	4B	075
RST 10H	D7	215	BIT 1,H	4C	076
RST 18H	DF	223	BIT 1,L	4D	077
RST 20H	E7	231	BIT 1,(HL)	4E	078
RST 28H	EF	239	BIT 2,A	57	087
RST 30H	F7	247	BIT 2,B	50	080
RST 38H	FF	255	BIT 2,C	51	081
SBC A,A	9F	159	BIT 2,D	52	082
SBC A,B	98	152	BIT 2,E	53	083
SBC A,C	99	153	BIT 2,H	54	084
SBC A,D	9A	154	BIT 2,L	55	085
SBC A,E	9B	155	BIT 2,(HL)	56	086
SBC A,H	9C	156	BIT 3,A	5F	095
SBC A,L	9D	157	BIT 3,B	58	088
SBC A,(HL)	9E	158	BIT 3,C	59	089
SCF	37	055	BIT 3,D	5A	090
SUB A	97	151	BIT 3,E	5B	091
SUB B	90	144	BIT 3,H	5C	092
SUB C	91	145	BIT 3,L	5D	093
SUB D	92	146	BIT 3,(HL)	5E	094
SUB E	93	147	BIT 4,A	67	103
SUB H	94	148	BIT 4,B	60	096
SUB L	95	149	BIT 4,C	61	097
SUB (HL)	96	150	BIT 4,D	62	098
SUB A,N	D6	214	BIT 4,E	63	099
SBC N	DE	222	BIT 4,H	64	100
XOR A	AF	175	BIT 4,L	65	101
XOR B	A8	168	BIT 4,(HL)	66	102
XOR C	A9	169	BIT 5,A	6F	111
XOR D	AA	170	BIT 5,B	68	104
XOR E	AB	171	BIT 5,C	69	105
XOR H	AC	172	BIT 5,D	6A	106
XOR L	AD	173	BIT 5,E	6B	107
XOR N	EE	238	BIT 5,H	6C	108
XOR (HL)	AE	174	BIT 5,L	6D	109
CB csoport	CB	203	BIT 5,(HL)	6E	110
DD csoport	DD	221	BIT 6,A	77	119
ED csoport	ED	237	BIT 6,B	70	112
FD csoport	FD	253	BIT 6,C	71	113
			BIT 6,D	72	114
			BIT 6,E	73	115
			BIT 6,H	76	116
			BIT 6,L	75	117
			BIT 6,(HL)	77	118
			BIT 7,A	7F	127
			BIT 7,B	78	120
			BIT 7,C	79	121
BIT 0,A	47	071			
BIT 0,B	40	064			
BIT 0,C	41	065			
BIT 0,D	42	066			
BIT 0,E	43	067			

Z80	HEX.	DEC.	Z80	HEX.	DEC.
BIT 7,D	7A	122	RES 6,B	80	176
BIT 7,E	7B	123	RES 6,C	81	177
BIT 7,H	7C	124	RES 6,D	82	178
BIT 7,L	7D	125	RES 6,E	83	179
BIT 7,(HL)	7E	126	RES 6,H	84	180
RES 0,A	87	135	RES 6,L	85	181
RES 0,B	80	128	RES 6,(HL)	86	182
RES 0,C	81	129	RES 7,A	8F	191
RES 0,D	82	130	RES 7,B	88	184
RES 0,E	83	131	RES 7,C	89	185
RES 0,H	84	132	RES 7,D	8A	186
RES 0,L	85	133	RES 7,E	8B	187
RES 0,(HL)	86	134	RES 7,H	8C	188
RES 1,A	8F	143	RES 7,L	8D	189
RES 1,B	88	136	RES 7,(HL)	8E	190
RES 1,C	89	137	RL A	17	023
RES 1,D	8A	138	RL B	10	016
RES 1,E	8B	139	RL C	11	017
RES 1,H	8C	140	RL D	12	018
RES 1,L	8D	141	RL E	13	019
RES 1,(HL)	8E	142	RL H	14	020
RES 2,A	97	151	RL L	15	021
RES 2,B	90	144	RL (HL)	16	022
RES 2,C	91	145	RLC A	07	007
RES 2,D	92	146	RLC B	00	000
RES 2,E	93	147	RLC C	01	001
RES 2,H	94	148	RLC D	02	002
RES 2,L	95	149	RLC E	03	003
RES 2,(HL)	96	150	RLC H	04	004
RES 3,A	9F	159	RLC L	05	005
RES 3,B	98	152	RLC (HL)	06	006
RES 3,C	99	153	RR A	1F	031
RES 3,D	9A	154	RR B	18	024
RES 3,E	9B	155	RR C	19	025
RES 3,H	9C	156	RR D	1A	026
RES 3,L	9D	157	RR E	1B	027
RES 3,(HL)	9E	158	RR H	1C	028
RES 4,A	A7	167	RR L	1D	029
RES 4,B	A0	160	RR (HL)	1E	030
RES 4,C	A1	161	RRC A	0F	015
RES 4,D	A2	162	RRC B	08	008
RES 4,E	A3	163	RRC C	09	009
RES 4,H	A4	164	RRC D	0A	010
RES 4,L	A5	165	RRC E	0B	011
RES 4,(HL)	A6	166	RRC H	0C	012
RES 5,A	AF	175	RRC L	0D	013
RES 5,B	A8	168	RRC (HL)	0E	014
RES 5,C	A9	169	SET 0,A	C7	199
RES 5,D	AA	170	SET 0,B	C0	192
RES 5,E	AB	171	SET 0,C	C1	193
RES 5,H	AC	172	SET 0,D	C2	194
RES 5,L	AD	173	SET 0,E	C3	195
RES 5,(HL)	AE	174	SET 0,H	C4	196
RES 6,A	B7	183	SET 0,L	C5	197

<u>Z80</u>	<u>HEX.</u>	<u>DEC.</u>	<u>Z80</u>	<u>HEX.</u>	<u>DEC.</u>
SET 0,(HL)	C6	198	SET 7,H	FC	252
SET 1,A	CF	207	SET 7,L	FD	253
SET 1,B	C8	200	SET 7,(HL)	FE	254
SET 1,C	C9	201	SLA A	27	039
SET 1,D	CA	202	SLA B	20	032
SET 1,E	CB	203	SLA C	21	033
SET 1,H	CC	204	SLA D	22	034
SET 1,L	CD	205	SLA E	23	035
SET 1,(HL)	CE	206	SLA H	24	036
SET 2,A	D7	215	SLA L	25	037
SET 2,B	D0	208	SLA (HL)	26	038
SET 2,C	D1	209	SLL A	37	055
SET 2,D	D2	210	SLL B	30	048
SET 2,E	D3	211	SLL C	31	049
SET 2,H	D4	212	SLL D	32	050
SET 2,L	D5	213	SLL E	33	051
SET 2,(HL)	D6	214	SLL H	34	052
SET 3,A	DF	223	SLL L	35	053
SET 3,B	D8	216	SLL (HL)	36	054
SET 3,C	D9	217	SRA A	2F	047
SET 3,D	DA	218	SRA B	28	040
SET 3,E	DB	219	SRA C	29	041
SET 3,H	DC	220	SRA D	2A	042
SET 3,L	DD	221	SRA E	2B	043
SET 3,(HL)	DE	222	SRA H	2C	044
SET 4,A	E7	231	SRA L	2D	045
SET 4,B	E0	224	SRA (HL)	2E	046
SET 4,C	E1	225	SRL A	3F	063
SET 4,D	E2	226	SRL B	38	056
SET 4,E	E3	227	SRL C	39	057
SET 4,H	E4	228	SRL D	3A	058
SET 4,L	E5	229	SRL E	3B	059
SET 4,(HL)	E6	230	SRL H	3C	060
SET 5,A	EF	239	SRL L	3D	061
SET 5,B	E8	232	SRL (HL)	3E	062
SET 5,C	E9	233			
SET 5,D	EA	234			
SET 5,E	EB	235	<u>DD csoport</u>		
SET 5,H	EC	236	ADC HL,BC	4A	074
SET 5,L	ED	237	ADC HL,DE	5A	090
SET 5,(HL)	EE	238	ADC HL,HL	6A	106
SET 6,A	F7	247	ADC HL,SP	7A	122
SET 6,B	F0	240	CPD	A9	169
SET 6,C	F1	241	CPDR	B9	185
SET 6,D	F2	242	CPI	A1	161
SET 6,E	F3	243	CPIR	B1	177
SET 6,H	F4	244	IM 0	46	070
SET 6,L	F5	245	IM 1	56	086
SET 6,(HL)	F6	246	IM 2	5E	094
SET 7,A	FF	255	IN A,(C)	78	120
SET 7,B	F8	248	IN B,(C)	40	064
SET 7,C	F9	249	IN C,(C)	48	072
SET 7,D	FA	250	IN D,(C)	50	080
SET 7,E	FB	251	IN E,(C)	58	088









Kiadja a Műszaki Könyvkiadó  
Felelős kiadó: Szűcs Péter igazgató  
Felelős szerkesztő: Czere Károlyné

Műszaki vezető: Kőríz Károly  
Műszaki szerkesztő: Bagi Miklós  
A borítót és a kötést tervezte: Kovács Tibor  
A könyv formátuma: B5  
Ívterjedelme: 20,375 (A5)  
Ábrák száma: 8  
Azonosítási szám: 80 036  
MÚ: 4123-h-8790  
Készült az MSZ 5601 és 5602 szerint  
A kézirat lezárva: 1987. november

VTV Reprotechnika - 87.151



MŰSZAKI KÖNYVKIADÓ