

KANDÓ KÁLMÁN VILLAMOSIPARI MŰSZAKI FŐISKOLA

A

HT~1080 Z

ISKOLASZÁMITÓGÉP

PROGRAMOZÁSA

BUDAPEST 1983

UTASSY SÁNDOR
MERSICH IVÁNNÉ - SZÉKELY JÁNOS

A

HT - 1080 Z
ISKOLASZÁMITÓGÉP
PROGRAMOZÁSA

I.

KANDÓ KÁLMÁN VILLAMOSIPARI MŰSZAKI FŐISKOLA

BUDAPEST, 1983.

Szerkesztő:

UTASSY SÁNDOR

Szerzők:

- 1. fejezet: SZÉKELY JÁNOS
- 2. fejezet: MERSICH IVÁNNÉ
- TI
- 3. fejezet: UTASSY SÁNDOR
ASZALIO

A kötet a HT - 1080 Z típusú iskolaszámítógép BASIC nyelven történő programozásához nyújt segítséget. Az 1. fejezet a számítógép kezelését, néhány alapparancsát és utasítását ismerteti.

A 2. fejezet a BASIC nyelvű programozást tárgyalja.

A 3. fejezet a gép parancskészletét, szövegszerkesztési lehetőségeit és a hangeffektusok programozását írja le.

A leírtak megértése, begyakorlása nem követel meg előzetes számítástechnikai ismereteket, feltételezi viszont azoknak a számítógépen történő kipróbálását, gyakorlását.

A gépi kódú programozással, perifériákkal, hibakezeléssel, speciális alkalmazási lehetőségekkel kapcsolatos ismereteket a könyv II. kötete ismerteti.

Budapest, 1983.

Készült a K.K.V.M.F. Soksorozító Üzemében 1983/408.

TARTALOMJEGYZÉK

	oldal
1. BEVEZETÉS	4
1.1. A gép jellemzői	4
1.2. Kezelési leírás	4
1.2.1. Bekapcsolás	4
1.2.2. Formátum választás	5
1.2.3. Billentyűzet, nagybetűk, kisbetűk, villogó kurzor	6
1.2.4. Beépített kazettás magnetofon, programok betöltése, felvétele	8
1.2.5. Rendszer RESET	12
1.3. Parancs és utasítás üzemmód	12
1.4. Parancsok	13
1.4.1. CLOAD	13
1.4.2. RUN	14
1.4.3. LIST	14
1.4.4. CSAVE	15
1.5. Utasítások	15
1.5.1. PRINT	15
1.5.2. INPUT	18
1.5.3. LET	19
1.5.4. GOTO	20
1.5.5. STOP	21
1.5.6. END	22
2. PROGRAMOZÁS	23
2.1. A program felépítése	23
2.2. Adattípusok, változók	25
2.2.1. Adatok deklarálása	26
2.2.2. Változók dimenzionálása	27
2.3. Belső aritmetikai függvények	28

	oldal
2.4. Vezérlésátadó utasítások	29
2.4.1. Feltétel nélküli vezérlés- átadás	29
2.4.2. Feltételes vezérlésátadás	30
2.4.3. Többirányú elágaztatás	31
2.4.4. Szubrutin hívás	31
2.5. Ciklusok	32
2.5.1. Ciklusképzés feltételes ve- zérlésátadással	32
2.5.2. Ciklusképző utasítások	33
2.5.3. Többszörös ciklusok /Tömbök kezelése/	34
2.6. Karakterfüzerek /stringek/ kezelése	35
2.6.1. Hozzárendelés, összehasonlítás, összekapcsolás	35
2.6.2. Füzetekre vonatkozó függvények	36
2.7. Értékadás	38
2.7.1. Belső értékadás	38
2.7.2. Külső értékadás	40
2.8. A PRINT utasítás alkalmazásai	41
2.8.1. PRINT@ mutató, tételsor	41
2.8.2. PRINT TAB /kifejezés/	42
2.8.3. PRINT USING karakteres kifeje- zés, nyomtatási lista	42
2.9. Adat file-ok kezelése programból	45
2.9.1. Adatok felírása kazettára	45
2.9.2. Adatok betöltése kazettáról	45
2.10. Képernyő grafika	46
2.11. Egyéb lehetőségek	46
3. EGYÉB FUNKCIÓK	48
3.1. Parancsok	48
3.1.1. AUTO sorszám, növekmény	48
3.1.2. CLEAR n	49
3.1.3. CLOAD #-egység száma, "file neve"	49

	oldal
3.1.4. CLOAD ?	49
3.1.5. CONT	50
3.1.6. CSAVE# -egység száma, "file neve"	50
3.1.7. DELETE sorszám -sorszám	50
3.1.8. EDIT sorszám	50
3.1.9. LIST sorszám - sorszám	51
3.1.10. LLIST sorszám - sorszám	51
3.1.11. NEW	51
3.1.12. RE sorszám, növekmény	51
3.1.13. RUN sorszám	52
3.1.14. SYSTEM	52
3.1.15. TRON	53
3.1.16. TROFF	53
3.1.17. SHIFT  P	53
3.2. Editálás /szövegszerkesztés/	53
3.2.1. A kurzor pozicionálása	55
3.2.2. Kiiratás	57
3.2.3. Törlési funkciók	57
3.2.4. Módosítások törlése	58
3.2.5. Visszatérés parancs üzemmódba	58
3.2.6. Beszúrás, kiegészítés	59
3.2.7. Karakterek kicserélése	60
3.3. Hangeffektusok programozása	61
3.3.1. Hangmagasság beállítása	62
3.3.2. Hangcsatornák és zajgenerátor engedélyezése /R7/	63
3.3.3. Hangerősség beállítása	64
3.4. Hibaüzenetek	66

1. BEVEZETÉS

1.1. A gép jellemzői

A HT-1080Z School-Computer - iskola-számítógép, vagy más néven személyi számítógép - egy központi egységből és ehhez közvetlenül, vagy bővítőn keresztül csatlakoztatható különböző egységekből áll. A központi egység - a számítógép -, amely 220V, 50Hz tápfeszültséggel működik, és 25 W fogyasztású, tartalmazza a működtetéshez szükséges billentyűzetet, kapcsolókat és az adattárolásra egy beépített kazettás magnetofont. A központi egységhez közvetlenül csatlakoztatható display, amely lehet monitor, vagy szabványos TV készülék. A TV készüléknél a csatlakozó dugaszt a VHF bemenethez csatlakoztatjuk /75 Ω -os lezárás/ és a készülék az OIRT 2. csatornán működtethető. A központi egységhez közvetlenül csatlakoztatható egy második, külső kazettás magnetofon is. Bővítőn keresztül különböző célok megvalósítására egyéb egységek is csatlakoztathatók, mint pl. nyomtató, floppy disk, stb.

A központi egységben található programozható hanggenerátor és hangszóró lehetőséget nyújt arra, hogy a programot hangeffektusokkal is kiegészítsük.

A tárolási kapacitás 16 Kbyte. 1 Kbyte = 2^{10} byte = 1024 byte, 1 byte egy 8 bites egységet jelent, egy bit pedig egyetlen bináris /kettes/ számjegy helyét jelenti.

1.2. Kezelési leírás

Ebben a fejezetben a központi egység kezelésével foglalkozunk; nem térünk ki a hozzá csatlakoztatható különböző egységek kezelésére.

1.2.1. Bekapcsolás

A bekapcsolás a központi egység hátoldalán jobboldalt elhelyezkedő hálózati kapcsolóval történik. A bekapcsolt állapotot a billentyűzet felett baloldalon elhelyezkedő

POWER feliratú piros LED /világító dióda/ jelzi.

Ügyelni kell arra, hogy a bekapcsolás sorrendje olyan legyen, hogy először a központi egységhez csatlakoztatott TV készüléket és utána a számítógépet kapcsoljuk be. A bekapcsolás után a képernyő bal felső sarkában

READY?-

felirat jelenik meg. A kérdőjel utáni kis vonal a kurzor /cursor/, az a jel, amelyet a számítógép minden beírt karakter /betű, számjegy vagy jel/ mögé kiír. Ezután megnyomjuk a **NEW LINE** feliratú gombot, erre a képernyő bal alsó sarkában

READY

> -

jelenik meg. A gép most PARANCS üzemmódban van, a kívánt működéshez szükséges parancsok, utasítások bebillentyűzhetők. Amennyiben a gépet bármely okból kikapcsoljuk, legalább 15 másodpercet várni kell az újbóli bekapcsolással!

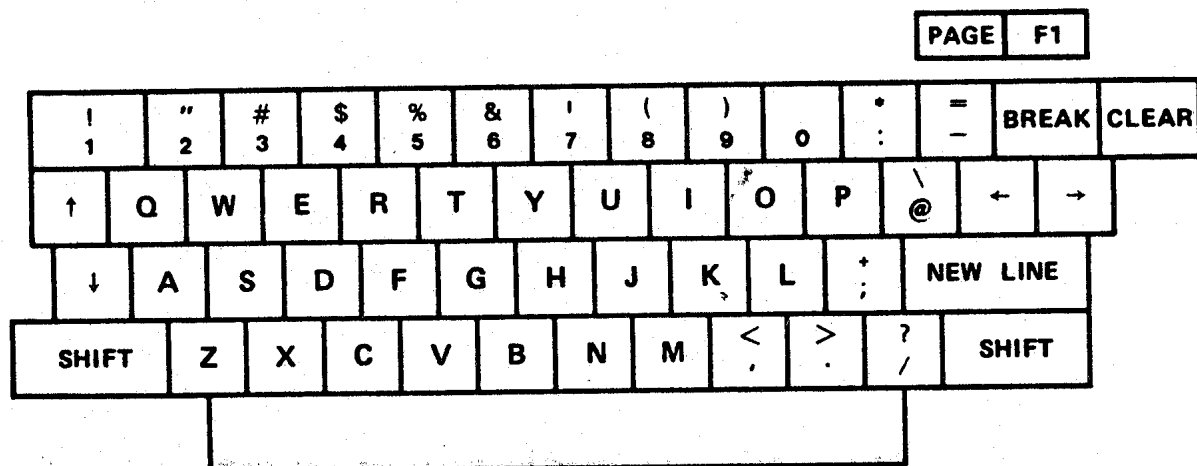
1.2.2. Formátum választás

A számítógép két kijelzési formátum választását teszi lehetővé; soronként 64, vagy 32 karakter. A 32 karakteres formátum esetén megvastagítottabbak és világosabbak a karakterek. Ez különösen TV készülék használata esetén előnyös.

A soronként 64 karakteres kijelzési formátum a készülék hátoldalán levő **VIDEO CUT** gomb kiugrasztott helyzetében, míg a soronként 32 karakteres kijelzési formátum a **VIDEO CUT** gomb benyomott állapotában jön létre. Az utóbbi esetben a billentyűzet felett elhelyezkedő **PAGE** lapválasztó gomb benyomása, ill. kiugrasztott helyzete szerint a teljes szöveg jobb-, vagy baloldali része látható a képernyőn.

1.2.3. Billentyűzet, nagybetűk, kisbetűk, villogó kurzor

A billentyűzet segítségével teremtünk kapcsolatot a géppel, parancsot, utasítást adunk, adatokat írunk be.



A billentyűzet a rajta levő betűk, jelek beírására szolgál. Egyes billentyűk speciális funkciókkal bírnak. Vizsgáljuk meg a legfontosabbakat:

BREAK

Megszakítja az éppen futó programot és a gép visszatér parancs üzemmódba.

NEW LINE

A parancs, utasítás, adat sorát lezárja.



Törli az utolsónak beírt karaktert.



Megjelenít egy [jelet, amely a rendszer által használt exponens jel.



A kurzort a következő sor elejére ugratja.



A kurzort 8 betűközzel jobbra mozgatja /tabulátor funkció/.

SHIFT →

Együttes megnyomás - valójában előbb az első /SHIFT/, majd azt nyomva tartva a második / → / billentyű lenyomása - esetén szóköz beszúrás; ekkor a karakterek közé, melyeket ezután írunk be, automatikusan szóköz iktatódik be. /Ugyanez történik akkor is, ha a PRINT CHR\$(23) parancsot billentyűzzük be./ Az alapállapot a NEW vagy CLS parancsok bebillentyűzésével áll vissza.

SHIFT ←

Együttes megnyomása sortörlést eredményez.

SZÓKÖZ

A billentyűzet legalsó sorában levő hosszú billentyű a SPACE - szóköz - billentyű. Minden lenyomásakor a gép a kurzort egy pozícióval jobbra mozgatja.

A billentyűzet egyéb funkcióiról, így a **CLEAR** billentyű szerepéről a későbbiek során beszélünk.

Alapállapotban a billentyűzet csak nagybetűk beírására alkalmas. Lehetőség van azonban kisbetűk használatára is. Ehhez az úgynevezett BASIC bővítésre van szükség, amely a kisbetűk használata mellett egyéb lehetőségeket is nyújt /pl. villogó kurzor, karakter bevitel automatikus ismétlése stb./. A BASIC bővítés használatbavétele a

READY

> _

megjelenítés után, tehát parancs üzemmódban végezhető, a következő módon:

Billentyűzzük be a SYSTEM szót és nyomjuk meg a **NEWLINE** billentyűt, ekkor a gép válasza a következő sorban *? jelek. Ezután billentyűzzük be /12288-at és nyomjuk meg a **NEWLINE** gombot. A képernyőn

NEW KEYBOARD ROUTINE ENABLE

felirat és villogó kurzor jelenik meg. A gép parancs üzem-

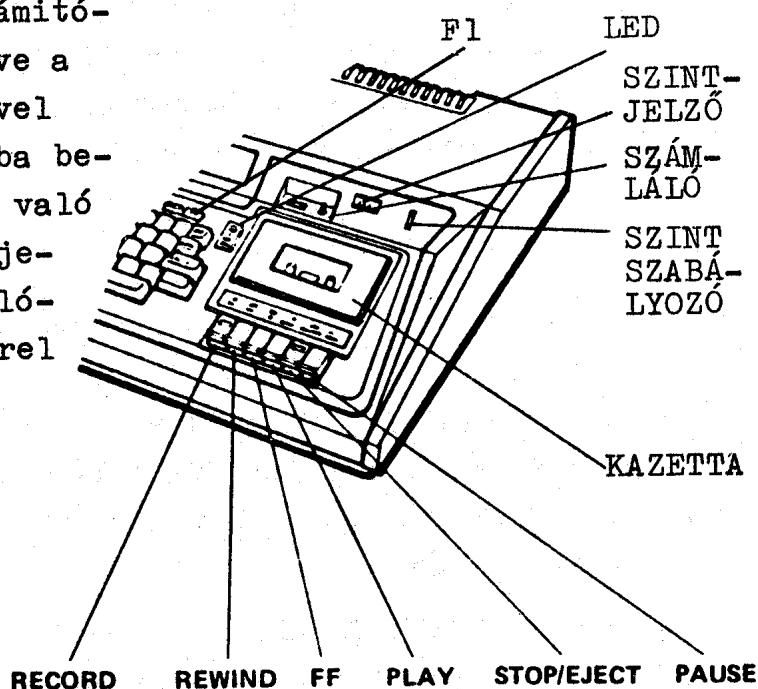
módban van. A 12288, melyet belépési címnek nevezünk, lehetővé teszi az összes ROM /Read Only Memory - csak olvasható memória/ lehetőség használatát. Amennyiben a 12299-es belépési címet billentyűzzük be, elmarad a villogó kurzor, a többi adottság marad, a 12294-es belépési cím pedig csak a kisbetűk használatát teszi lehetővé. Ez utóbbi esetben a képernyőn nem jelenik meg az angol nyelvű felirat.

Kisbetűket úgy lehet beírni, hogy a karakter gombjának lenyomásával egyidejűleg lenyomjuk a **SHIFT** gombot is. Ha egy billentyűt egy másodpercnél hosszabb ideig lenyomva tartunk, a gép automatikusan megismétli a karakter bevitelét mindaddig, amíg a billentyűt lenyomva tartjuk.

Amennyiben a 12288 belépési címmel a kurzor villogását időszakosan meg akarjuk szüntetni, ez a **SHIFT** és **BREAK** gombok egyidejű lenyomásával érhető el. Az újbóli villogó kurzor ugyanezen gombok újbóli lenyomásával kapható meg. Ha a villogó kurzorra már kezdetben sincs szükségünk, használjuk a 12299 belépési címet.

1.2.4. Beépített kazettás magnetofon, programok betöltése, felvétele

A központi egységbe beépített kazettás magnetofon az adat-rögzítésre szolgál. Lehetőséget nyújt a kazettán tárolt programok betöltésére a számítógép memóriájába, illetve a billentyűzet segítségével a számítógép memóriájába beírt programok kazettán való rögzítésére. Háromszámjegyes nullázható számlálóval, szintjelző műszerrel és szintszabályozóval van ellátva.



A magnetofon a központi egység bekapcsolt állapota esetén működtethető.

Vizsgáljuk meg a magnetofont működtető gombokat:

[F1]

Az előlapon a magnetofon mellett helyezkedik el, benyomása kézi előre-, vagy hátratekerceseléskor leválasztja a kazettát a számítógépről. Ezt az állapotot a gomb mellett elhelyezkedő piros LED jelzi. Ekkor végezhető a magnószalag előre-, vagy hátratekerceselése.

[REW]

Gyors hátratekerceselés.

[FF]

Gyors előretekerceselés.

[RECORD]

és

[PLAY]

Együttes lenyomása esetén programfelvétel és egyúttal az előzőleg felvett adatok törlése.

[PLAY]

Program betöltés /lejátszás/, ellenőrzés.

[STOP/EJECT]

A magnó leállítását és a kazetta kiemelését teszi lehetővé.

[PAUSE]

Pillanat állj.

Kazettán levő program betöltése a számítógép memóriájába a következő módon történik: /Használjuk fel a demonstrációs kazettát, melyen számozott BASIC és gépi kódú programok vannak!/
/

I. BASIC programok betöltése

a./ Amennyiben tudjuk, hol kezdődik a program, a magnószalagot, a számlálón ellenőrizve, a program elejére csévéljük, majd kikapcsoljuk az **[F1]** -et, így a mellette levő piros LED kialszik. A képernyőn

READY

> _

megjelenése után, tehát parancs üzemmódban bebillen-

tyüzzük a CLOAD parancsot, és megnyomjuk a magnó **PLAY** gombját. Ezután a **NEWLINE** gombot megnyomva megindul a program betöltése. Ezt a képernyő jobb felső sarkában megjelenő két csillag jelzi, melyek közül a jobboldali időről-időre felvillan. Ez alatt az **F1** gomb melletti piros LED világít és a magnó szintjelző műszere jelez. A program végén a betöltés leáll, az **F1** gomb melletti piros LED kialszik, szintjelző mutatója nullán áll. Ezután RUN parancs bebillentyűzésével és **NEWLINE** gomb benyomásával indíthatjuk a programot.

- b./ Amennyiben nem tudjuk, hol kezdődik a program, a magnószalagot a szalag elejére csévéljük és ugyanúgy járunk el, mint az előző esetben. Most azonban a CLOAD #-1, "N" parancsot billentyűzzük be, ahol N a program neve vagy száma. A **NEWLINE** billentyű megnyomásával indíthatjuk el ezt a parancsot is, természetesen a **PLAY** gomb lenyomása után. Ekkor a magnetofon elindul és a gép elkezd keresni az N programot. A továbbiakban minden úgy folytatódik, mint az előző esetben.

II. Gépi kódú /"object code"/ programok betöltése

A magnószalagot, a számlálón ellenőrizve a program elejére csévéljük, majd kikapcsoljuk **F1** -et. A

READY

> _

megjelenése után, tehát parancs üzemmódban bebillentyűzzük a SYSTEM parancsot és megnyomjuk a magnó **PLAY** gombját, majd a **NEWLINE** gombot. Ekkor a képernyőn a következő sorban

* ?

jelek jelennek meg. Bebillentyűzzük a program nevét, pl. OTELLO és újból megnyomjuk a **NEWLINE** gombot. Ekkor megindul a program betöltése és minden úgy történik, mint BASIC programok esetén. Amikor a betöltés befejeződött

* ?

jelenik meg újból a képernyőn. A program indításához a / /per/ jel és az indító cím beírása szükséges. /Például /22222, mely a demonstrációs kazetta OTELLO programjának indító címe/ Ezután nyomjuk meg a **NEWLINE** gombot.

Amennyiben az I, II típusú program betöltésekor a leírt jelzések nem jelennek meg /pl. a képernyő jobb felső részében két csillag közül a jobboldali nem villog/, akkor a program betöltése nem történt meg, meg kell ismételni. A hibás programbetöltés gyakori oka a nem megfelelő minőségű programkazetta, vagy felvétel, illetve a kimeneti szint nem megfelelő értéke. Az ismételt programbetöltés előtt a szintmérő műszert figyelve, a szintszabályozó gomb segítségével a szintet a kívánt mértékűre lehet állítani. Ügyeljünk arra, hogy a program betöltése alatt a szintet nem szabad állítani, hanem előtte az **F1** és **PLAY** gombok lenyomásakor állítsuk be a szintet, és utána tekercseljük vissza a szalagot az eredeti helyzetébe és ezután kezdjük a korábbiakban leírt módon a program betöltését.

A betöltéshez ajánlott legjobb szint a közepes nagyságú /2-3 körül mutatja a műszer/. Ha a betöltés sikertelen, próbálkozunk magasabb, vagy alacsonyabb szinttel. Ha egyáltalán nem sikerül, valószínűleg sérült a programunk.

A program ellenőrzés a BASIC programok esetén a következőképpen történik:

Csévéljük a szalagot a program elejére, majd engedjük fel az **F1** gombot. Billentyűzzük be a CLOAD? parancsot. Nyomjuk meg a magnó **PLAY** gombját és utána a **NEWLINE** billentyűt. Ekkor megindul a programellenőrzés, melynek befejeztével a magnó automatikusan leáll. Hiba esetén a

BAD

jelzés jelenik meg. Ha bebillentyűztünk egy programot a számítógép memóriájába és azt a későbbiek számára meg kívánjuk őrizni, akkor azt magnószalagra fel kell vennünk.

A program felvétel a következő módon történik:

Parancs üzemmódban bebillentyűzzük a CSAVE # -1, N parancsot,

ahol N a program nevét vagy számát jelenti. Az **[F1]** gomb kiugratott állásában nyomjuk be egyszerre a magnetofon **[RECORD]** és **[PLAY]** gombjait, ezután nyomjuk be a **[NEW LINE]** gombot. Ekkor megindul a programfelvétel. A felvétel befejeztével a magnetofon automatikusan leáll. A felvétel hangjelzés kíséri. Felvételkor a szintet nem kell állítani.

A magnószalag indulási és programvéghelyzetét célszerű a számlálóról leolvasni, így tudjuk, mettől-meddig tart a programunk. A programfelvétel után igen fontos a program betöltésekor megismert programellenőrzés elvégzése. Erre elsősorban itt van szükség, mivel így bizonyosodhatunk meg arról, hogy a felvétel sikeres volt.

1.2.5. Rendszer RESET

Előfordul, hogy a rendszert alapállapotba kell állítani, pl. akkor, amikor a program futtatáskor végtelen ciklusba kerül. A központi egység hátoldalán baloldalt található **[RESET]** gomb benyomására a gép parancs üzemmódra tér át, a tár tartalma nem változik meg.

1.3. Parancs és utasítás üzemmód

Amikor

READY

> _

üzenet jelenik meg a képernyőn, a gép parancs üzemmódban van, és várja a bebillentyűzendő parancsot, vagy utasítást. Ha egy parancsot bebillentyűztünk, a **[NEW LINE]** gomb lenyomása után a gép rögtön végrehajtja azt. Több művelet egymásutáni végrehajtására utasításokból álló programot kell a gépbe beírni. Az utasítások - a parancsoktól eltérően - egy decimális számmal, címkével kezdődnek. Az utasítások tulajdonképpen fölfoghatók úgy is, mint a programon belüli parancsok. Az utasítás végrehajtása a program futása során történik csak meg. Természetesen az utasításokat is nekünk kell a gépbe

előzetesen bebillentyűzni, a programírás során. /1. 2. fejezet/

Mind a parancsok, mind az utasítások kulcsszavakból állnak. Egyes kulcsszavak mind parancs, mind pedig utasítás formájában is alkalmazhatók.

1.4. Parancsok

A teljes parancskészletből ebben a részben csak az alábbiakkal foglalkozunk:

CLOAD

RUN

LIST

CSAVE

A parancsokat gyakoroljuk a demonstrációs kazetta segítségével.

1.4.1. CLOAD

Amint erről már korábban beszéltünk, ez a parancs a magnószalagon tárolt programot tölti be a számítógépbe, ill. a programellenőrzést indítja. A CLOAD parancs következő alakjai használatosak:

CLOAD

Csak a magnószalagnak a program kezdetéhez tekercselt állapotában használható.

CLOAD # - kazettás magnó száma, "program név" pl.

CLOAD # -1, "N"

Jelentése: az 1-es magnetofonról - ez a beépített magnó - az N nevű filet kell betölteni.

/file: rendezett adathalmazt, pl. a programot jelenti/

A gép a file névnek csak az első karakterét használja, a különböző file-ok megkülönböztetéséhez; pl. az N és NÉNI ugyanannak a programnévnek számít.

E parancs használata esetén a magnószalag elején lehetünk, mert a gép megkeresi a /pl. N nevű/ programot.

CLOAD?

Ez a parancs összehasonlítja a kázzettán levő file-t a tárban levő programmal.

CLOAD? "program-név"

A számítógép a megadott nevű file-t keresi és hasonlítja össze a tárban lévővel.

1.4.2. RUN

Amint erről a korábbiakban szó volt, ez a parancs elindítja a tárban levő felhasználói programot. Ekkor a gép programvégrehajtás üzemmódba kerül. A RUN parancs következő alakjai használatosak:

RUN

A programot a legalacsonyabb sorszámmú sorról indítja.

RUN sorszám pl. RUN 20 A programot a 20 sorszámról indítja. Ha olyan sortól akarjuk a programot indítani, ahonnan a program szerkezete nem teszi lehetővé, vagy érvénytelen, nem létező sorszámot adunk meg, akkor hibajelzést kapunk.

1.4.3. LIST

Utasítja a számítógépet, hogy egy, vagy több programsort írjon ki a képernyőre. A LIST parancs következő alakjai használatosak:

LIST

Kiírja az egész, tárban levő programot. Ha meg akarjuk állítani a listázást, nyomjuk meg egyszerre a **SHIFT** és **Ⓢ** billentyűket. A listázás bármelyik billentyű lenyomásakor továbbindul.

LIST sorszám pl. LIST 10 A tízes sort írja ki a gép.

LIST sorszám-sorszám pl. LIST 10-20

A tizes, huszas és a közöttük levő sorokat írja ki a gép.

LIST - sorszám pl. LIST-20

Az elsőtől a huszas sorig az összes sort kiírja a gép.

LIST sorszám - pl. LIST 20-

A 20 sortól a program végéig a gép kiírja a sorokat.

1.4.4. CSAVE

Ez a parancs teszi lehetővé, hogy a gép memóriájában lévő programot magnókazettán rögzítsük. A CSAVE parancs alakja a következő:

CSAVE # - kazettás magnó száma, "program név"

pl. a CSAVE #-1, "C" parancs a memóriában lévő C nevű programot az 1. számú - a beépített - magnóra veszi fel. A gép az idézőjeleken kívül minden alfanumerikus karaktert file-névnek értelmez.

1.5. Utasítások

A BASIC programnyelv utasításkészletéből ebben a részben csak az alábbiakkal foglalkozunk részletesebben:

PRINT

LET

INPUT

GOTO

STOP

END

1.5.1. PRINT

Ez az utasítás, melyet egy tétel, vagy tételsor követ, a tételt, vagy tételsort megjeleníti a képernyőn.

Tételsornak tekintjük a következőket:

- | | |
|----------------------|---|
| a./ számkonstansok | - pl. 5, 10, -67 |
| b./ számváltozók | - olyan nevek, amelyek számokat reprezentálnak
pl. A; B; XX; Z3 |
| c./ karakterállandók | - idézőjelben lévő karakterek pl. "HT-1080Z" |
| d./ karakterváltozók | - olyan nevek, amelyek karakter-fűzért reprezentálnak
pl. A\$; X\$; PR\$ |
| e./ kifejezések | - a fenti tételek kombinációi pl. X + 10/Y |

Ügyelni kell arra, hogy a BASIC kulcsszavait pl. a parancsok, utasítások stb. neveit a változónevekben nem használhatjuk. A változónevek betűk és számok lehetnek, de csak az első két jelet használja a gép, tehát PE és PENZ ugyanazt jelenti. Az egyes tételeket vesszővel vagy pontosvesszővel választjuk el egymástól. Ha vesszőt használunk, akkor a kurzor automatikusan a következő nyomtatási zónába ugrik, mielőtt a következő tételt kiírja. Ekkor a gép soronként 4, egyenként 16 karaktert tartalmazó zónába ír. Ha két, vagy több vesszőt írunk be, akkor vesszőnként 16 szóköz jelenik meg.

Ha pontosvesszőt használunk, akkor a tételek között nem marad szabad hely, egymáshoz fűzve jelennek meg, a numerikus tételek között pedig egy hely marad ki /pl. számoknál/.

A PRINT utasítás végrehajtása után a gép automatikusan sor-emelést hajt végre, kivéve azt az esetet, ha a tételsort valamelyik elválasztójellel - vessző, pontosvessző - fejezzük be.

Amennyiben a PRINT utasítást parancs üzemmódban használjuk, akkor a képernyőre való kiírás a NEW LINE gomb megnyomása után rögtön megtörténik.

A továbbiakban a programpéldáinknál a következő jelöléseket használjuk: az aláhúzás nélküli karaktereket a gép automatikusan megjeleníti, az aláhúzott karaktereket be kell billentyűzni. A parancs,- vagy utasítássorok lezárását jelentő

NEW LINE billentyű lenyomását a továbbiakban nem jelöljük!!

Parancs üzemmódban való használat

READY

>PRINT 6, 7, 8; 9

6

7

8 9

READY

>

A parancs végrehajtása után a gép újból parancs állapotba kerül.

Használat programon belül:

a./ READY

>10 PRINT 6,,7

>20 END

>RUN

6

7

b./ READY

>10 A = 2

>20 PRINT A

>30 END

>RUN

2

c./ READY

>10 PRINT "HT1080-Z"

>20 END

>RUN

HT1080-Z

d./ READY

>10 M\$ = "KANDO"

>20 C\$ = " KALMAN"

>30 PRINT M\$;C\$

>40 END

>RUN

KANDO KALMAN

Ügyeljünk arra, hogy a 10-es sorban a KANDO és a záró idézőjel, vagy a 20-as sorban a kezdő idézőjel és a KALMAN között maradjon ki egy karakter, mert egyébként KANDOKALMAN-t kapunk.

e./ READY

```
>10 N = 25 + 7
>20 PRINT "25 + 7 EGYENLO" ; N
>30 END
>RUN
25 + 7 EGYENLO 32
```

A PRINT utasítás további lehetőségeit a 2.9. fejezetben ismertetjük.

1.5.2. INPUT

Ezzel az utasítással az utasítást követő numerikus, vagy karakteres változóknak a program végrehajtásakor adhatunk értéket a billentyűzetről.

A beírandó adatok numerikus értékek és alfanumerikus karakterek lehetnek, a változó típusától függően. Minden tételt vesszővel kell elválasztani.

A következő példaprogramnál két karakterfüzért és két számértéket lehet beírni. A beírások sorrendjére ügyelni kell. A beírt adatoknak a változó típusának megfelelőnek kell lenniük. Számváltozóba pl. nem írhatunk be karakterfüzért. Ha ezt eltévesztjük, akkor erre a gép

?REDO

?

sorokkal válaszol, és új adatokat kér abba a változóba, amelynek típusa nem egyezett meg az előzőleg beírtakkal.

Az INPUT utasítás végrehajtásakor a gép egy kérdőjelet ír a képernyőre és várja az adatokat. Egy sorba írhatjuk be mind a négy értéket, vesszővel elválasztva azokat, vagy mind a négy értéket külön sorba írhatjuk.

READY

>10 INPUT A\$, B\$, A, B

>20 PRINT A\$, B\$, A, B

>30 END

>RUN

?MAMA, PAPA, 35, 36

MAMA PAPA 35
-- -- -- -- --
READY

>RUN

?MAMA

??PAPA

??35

??36

MAMA PAPA 35
-- -- -- -- --
READY

>RUN

?MAMA, PAPA, GYEREKEK, 36

?REDO

?MAMA

??PAPA

??35

??36

MAMA PAPA 35 36
READY
>

ebben a sorban várja a gép a válaszokat, pl. a beírtakat

36 A vessző használat miatt 16 karakter van a kiírások kezdete között

ebben a sorban várja a gép újból a válaszokat, amelyeket most külön sorba írunk

36

újból elkezdjük a programot, de most hibát fogunk csinálni, az egyik számváltozó helyett karaktereket írunk be

Természetesen a MAMA, PAPA helyett bármilyen más karaktersor, a 35, 36 helyett bármilyen más szám beírható.

Az INPUT utasítás másik változata az INPUT # - kazettás magnó száma, tételsorozat alakú, mely hatására a gép kiolvassa a szalagon lévő adatokat és beírja azokat a tételsorozat változóiba /lásd 2.10. fejezet/.

1.5.3. LET

A LET programutasítás egy változóhoz egy értéket rendel.

Formája LET változó = kifejezés pl. LET A = 10. A mi rendsze-

rünkben a LET szó elhagyható, tehát A = 10 ugyanazt jelenti, mint a LET A = 10.

Az alábbi példaprogramnál az egyenlőség baloldalán álló változóba a jobb oldalon levő kifejezés értéke íródik be.

READY

>10 LET A = 10

>20 LET B = 20

>30 LET AS = "SOK"

>40 PRINT A, B, AS

>50 END

READY

>RUN

10 20 SOK

1.5.4. GOTO

A GOTO sorszám utasítás hatására a program végrehajtása a megadott soron folytatódik /ugró utasítás/. Ha önmagában használjuk, akkor a gép feltétlen ugrást hajt végre.

Példaprogram:

READY

>10 A = 10

>20 B = 20

>30 C = A + B

>40 GOTO 80

>50 B = 0

>60 A = 0

>70 C = 0

>80 PRINT A, B, C

>90 END

READY

>RUN

10 20 30

Ha a vezérlésátadást /ugrást/ valamilyen feltételtől akarjuk függővé tenni, a GOTO elé IF utasítást írhatunk. Az IF arra

utasítja a gépet, hogy vizsgáljon meg egy relációt vagy egy logikai kifejezést. Ha a kifejezés "igaz" akkor a gép a műveletet végrehajtja, ha hamis akkor a műveletet nem veszi figyelembe, és folytatja a programvégrehajtást a következő utasítással. Numerikusan az "igaz" azt jelenti, hogy a kifejezés nem nulla.

Példaprogram:

READY

>10 INPUT A

>20 B = 20

>30 C = A + B

>40 IF C > 30 GOTO 60

>50 PRINT "A ERTEKE ELEG KEVES, NEM UGROM"

>60 PRINT A, B, C

>70 END

READY

>RUN

?20

20

READY

>RUN

?5

A ERTEKE ELEG KEVES, NEM UGROM

5

20

25

Az IF feltételes vezérlésátadást a 2.4.2 fejezetben részletesen ismertetjük.

1.5.5. STOP

Ez az utasítás a program hibáinak a megkereséséhez, vagy a program egyéb okból történő megszakításához nyújt segítséget. Töréspontot helyez el a program végrehajtásában és lehetővé teszi értékek megváltoztatását, vagy kiírását. STOP utasítás végrehajtásakor a gép kiírja a BREAK IN sorszám
- Megszakítás a-sorban - üzenetet.

A CONT paranccsal a program végrehajtása folytatható, attól a ponttól, ahol korábban meg lett szakítva.

Példaprogram:

READY

>10 INPUT B, C

>20 A = B + C

>30 STOP

>40 INPUT X, Y

>50 Z = X + Y

>60 IF Z 10 GOTO 90

>70 PRINT A, B, C

>80 PRINT X, Y, Z

>90 END

READY

>RUN

?10,20

BREAK In 30

READY

>PRINT A

30

READY

>CONT

?240,50

ha ebbe a sorba pl. 2,4-et
írunk be, akkor a program foly-
tatódik és a 70,80 sor utasi-
tásai szerint kiírja A, B, C,
X, Y, Z értékét.

A STOP utasítással lehetővé vált, hogy A értékét a 40. sor előtt megnézzük.

1.5.6. END

Ez az utasítás a program logikai végét jelenti. Az END utasítás utáni sorokat csak szubrutin /részprogram, alprogram/ hívással lehet végrehajtani. /2.4.4. fejezet/

2. PROGRAMOZÁS

2.1. A program felépítése

A BASIC nyelvű program utasítások sorozata. Minden utasítássor sorszámmal kezdődik. A sorszámoknak kettős szerepe van:

- rendszerint meghatározzák az utasítások végrehajtási sorrendjét;
- lehetővé teszik az utasítások azonosítását /a vezérlés-átadó utasításokban azokra az utasításokra, amelyekre a vezérlést át kívánjuk adni, sorszámuk alapján hivatko-zunk/.

A sorszám legfeljebb ötjegyű egész szám /max. 65529/, amelyet közvetlenül a sor elejére kell írni. A sor végét NEW LINE billentyűvel jelezzük. A programban nem fordulhat elő két azonos sorszámú utasítás. Ha egy utasítást olyan sorszámmal gépelünk, amely a programban már szerepelt, akkor a később leírt utasítás "felülírja" a korábbi.

A program utasításait tetszőleges sorrendben írhatjuk, a programot a gép növekvő sorszámok szerint hajtja végre. Ez lehetővé teszi, hogy a programba utólag írjunk be utasításokat. Ezért célszerű az utasításokat "hézagosan" /pl. tízé-sével/ sorszámozni. Így utólag bárhová beszúrhatunk új sorokat, anélkül, hogy ehhez a program meglevő utasításait át kellene számozni.

Nézzünk meg egy egyszerű programot.

```

10 PRINT "KET SZAM OSSZEADASA"
20 INPUT "KEREM AZ ELSO SZAMOT:"; A
30 INPUT "KEREM A MASODIK SZAMOT:"; B
40 C = A + B
50 PRINT "AZ EREDMENY:"; C
60 END

```

Ez a program beolvas két számot, összeadja azokat és az eredményt kiírja. A programban értékadást is láthatunk. Az értékadó utasítás jobb oldalán mindig valamilyen aritmetikai kifejezés áll. Az aritmetikai kifejezés előírás az aritmetikai

műveletek elvégzésére.

A BASIC-ben használatos aritmetikai műveletek a következők:

- + összeadás
- kivonás
- * szorzás
- / osztás
- [hatványozás /a [↑] billentyű lenyomásakor jelenik meg/

Pl. $5 \cdot 12^{1/3}$ -t a programban a következőképpen írhatjuk le
 $5 * 12 [(1/3)]$

Az aritmetikai kifejezések aktuális értékének összehasonlítására relációs jeleket használunk. Bár ezek értékadó utasításban alig fordulnak elő, a teljesség kedvéért itt soroljuk fel őket:

- = egyenlő
- < kisebb
- > nagyobb
- <= kisebb, vagy egyenlő
- >= nagyobb, vagy egyenlő
- <> nem egyenlő

A relációkkal részletesen a feltételes vezérlésátadásnál fogunk foglalkozni.

A számítógép az AND, OR és NOT logikai műveleteket ismeri még.

Pl. 70 A = NOT (F > 5)

Ha F kisebb vagy egyenlő 5-tel, A értéke /-1/, különben nulla lesz. Tehát a /-1/ az "igaz", a 0 pedig a "nem igaz" feltételnek felel meg.

Az aritmetikai kifejezések kiértékelését a gép bizonyos szabályok szerint végzi. Ezek a szabályok határozzák meg többek között a műveletek elvégzésének sorrendjét. A műveleti jeleket, a matematikai képletek értelmezéséhez hasonlóan, itt is csoportosíthatjuk elsőbbségük /prioritásuk/ szerint.

1. hatványozás $A \uparrow B$
2. előjelváltás /negáció/ $\neg C$
3. szorzás, osztás $A * B, C / D$
4. összeadás, kivonás $C + D, E - F$
5. relációk $A < B, C = D, 15 > 16$
6. logikai műveletek NOT, AND, OR

Az aritmetikai kifejezések kiértékelése a BASIC-ben az alábbi szabályok szerint megy végbe:

- Az aritmetikai kifejezésekben a műveleteket a gép a műveleti jelek prioritása szerinti sorrendben végzi el. Ettől eltérő sorrendet zárójelek alkalmazásával írhatunk elő.
- Az egyenlő prioritású műveleteket a gép balról jobbra haladva hajtja végre. Zárójelek alkalmazásával a végrehajtás sorrendjét itt is megváltoztathatjuk.

A zárójeleket a BASIC nyelv aritmetikai kifejezéseiben ugyanúgy használhatjuk, mint azt a matematikai képletekben megszoktuk.

2.2. Adattípusok, változók

Az algebrai változókhoz hasonlóan, a BASIC-ben is használhatunk szimbolikus neveket az egyes mennyiségek megjelölésére. Minden változónak van egy "neve". Ezt a nevet azonosítónak hívjuk. Az azonosító lehet az angol ábécé bármely nagybetűje, valamint betű és ezt követő számjegy kombinációja. A változónévnek mindig betűvel kell kezdődnie és nem tartalmazhatja a BASIC kulcsszavait. /pl. a CIF változónévben benne van az IF/ A számítógép elfogad olyan változókat is, amelyek két jelnél hosszabbak, a változók megkülönböztetésére azonban csak az első két jelet használja.

Pl. az "ER" és az "EREDMENY" változókat azonosnak tekinti. A számítógép négyfajta adattípussal dolgozik. Az első három típus numerikus adatok, a negyedik pedig karaktersorozatok tárolására, műveletvégzésére használatos.

1. %: egésztípusú /integer/ /-32 768/ - /+32 767/
 Pl. A% = -30
 BB% = 8000
2. !: egyszeres pontosságú /max. 6 számjegyes/
 Pl. A! = 40.33
 D4! = 0.1241
3. #: dupla pontosságú /max. 16 számjegyes/
 Pl. A# = 3.141592653589
 A2# = 456.6582178
4. %: karaktersorozat /string/ /max. 255 jel hosszú/
 Pl. A% = "A KOR KERULETE"
 M2% = "A + B EREDMENYE"

Az A%, A!, A és A% változónevek mindegyikében szerepel az "A", a számítógép számára ezek mégis különböző változók, különböző értékek.

2.2.1. Adatok deklarálása

Az egyes változónevek típusát előre definiálhatjuk, deklarálhatjuk a következő utasításokkal:

- DEFINT változók kezdőbetűi

Az ezen betűkkel kezdődő változókat a gép egésztípusú változóként kezeli és tárolja. Pl.

DEFINT X, Y, Z

minden X, Y, Z-vel kezdődő változót egészként kezel /XX, X1, Y2, Z1 stb./

DEFINT A - D

minden A, B, C, D-vel kezdődő változót egésznek nyilvánít.

- DEFSNG változók kezdőbetűi

Minden, a megadott betűkkel kezdődő változót egyszeres pontosságúnak deklarál. Pl.

DEFSNG A - D, Y

- DEFDBL változók kezdőbetűi - duplapontosságú változók deklarálása

- DEFSTR változók kezdőbetűi - karakteres /string/ változók deklarálása /a fűzér maximális hossza 255 karakter lehet/

Pl. 10 DEFSTR A-D sor után a gép pl. elfogadja

B3 = "EGY FUZER" kifejezést, máskülönben hibát jelez. A rendszer bekapcsolásakor a gép 50 byte-ot tesz szabaddá a füzérek tárolására. Ha ez nem elegendő, akkor a CLEAR n utasítással a számítógép "n" helyet szabadit fel. Ez az utasítás minden változót nulláz.

2.2.2. Változók dimenzionálása

Ha egy változóhoz csak egyetlen érték tartozik, egyszerű változónak nevezzük.

Gyakran szükség van arra is, hogy összetartozó értékekre egyetlen névvel hivatkozassunk. Ez a helyzet pl. amikor vektorokkal, vagy mátrixokkal dolgozunk.

Vektoron a továbbiakban n db számot értünk, aminek megőriztük a sorrendjét. Az egyes számokat a vektor elemeinek nevezzük és egy meghatározott számra a hozzárendelt indexszel tudunk hivatkozni /0, 1, 2.../. Kétdimenziós mátrixon egy n sorból és m oszlopból álló táblázatot értünk. A mátrix egy elemét a sor és oszlop indexszel adhatjuk meg. Tehát ha vektorral, vagy mátrix-szal dolgozunk, ezeknek helyet kell lefoglalnunk a tárban. Ezt a DIM utasítással valósíthatjuk meg.

DIM név(dim1, dim2,...dimn)

Pl. 10 DIM A(5), B(3, 4), C(3, 4, 4)

Ez az utasítás egy 6 elemű A vektort /0...5/, egy B két dimenziós mátrixot /4x5/, és egy C három dimenziós mátrixot definiál.

A DIM utasítást a program tetszőleges részén helyezhetjük el. A dimenzió megadása történhet egész számmal, vagy kifejezéssel. Egy tömb újra dimenzionálása előtt CLEAR utasítást kell adni, különben a gép hibát jelez.

AZ UTASÍTÁSOKNÁL, FÜGGVÉNYEKNÉL MINDIG GÖMBÖLYÜ ZÁRÓJELET KELL HASZNÁLNI! A SZÖVEGBEN, GÉPELÉSI PROBLÉMÁK MIATT, ZÁRÓJELKÉNT / JELET ALKALMAZTUNK.

2.3. Belső aritmetikai függvények

A matematikai képletekhez hasonlóan a BASIC-ben is megengedett, hogy az aritmetikai kifejezésekbe ún. függvényhivatkozást építsünk be. Legtöbb esetben a függvényérték kiszámítása előtt meg kell adni a függvény argumentumát. Ez az argumentum lehet konstans, numerikus változó, vagy kifejezés. Az általános alak:

Azonosító = függvény (argumentum)

Az aritmetikai kifejezésekben használható belső függvények a következők:

- | | |
|----------|---|
| ABS (X) | - Kiszámítja X abszolút értékét. |
| ATN (X) | - Kiszámítja X radiánban vett arcus tangens főértékét. |
| CDBL (X) | - X-et kétszeres pontossággal ábrázolja. |
| CINT (X) | - Kiszámolja az egészrészt /argumentumnál kisebb, ahhoz legközelebb eső egész számot/. Az értéknek a -32768 és +32767 között kell lennie. |
| COS (X) | - Kiszámolja a radiánban megadott X koszinuszát. |
| CSNG (X) | - X-et egyszeres pontossággal ábrázolja. |
| EXP (X) | - Kiszámítja az exponenciális függvény értékét az X helyen $/e^X/$. |
| FIX (X) | - Leválasztja az X argumentum tizedes pont utáni részét. |
| INT (X) | - X-et egész számként ábrázolja, kiszámítja az X-nél nem nagyobb egész számot /lefelé kerekít/. Ennek a számnak nem kell a -32768 és +32767 intervallumba esni. |
| LOG (X) | - Kiszámítja X természetes alapú logaritmusát $/\ln X/$. $/X > 0! /$ |
| RANDOM | - A gép véletlen számokat generál. A függvényhez nem kell argumentumot megadni. |

RND (X)	- Egy álvéletlen számot állít elő a /0 - X+1 / intervallumban.
SGN (X)	- Az előjelfüggvény értéke: -1, ha X negatív; 0, ha X nulla; +1, ha X pozitív.
SIN (X)	- Kiszámolja a radiánban megadott X szinuszt.
SQR (X)	- Kiszámolja X négyzetgyökét. /X > 0! /
TAN (X)	- Kiszámolja a radiánban megadott X tangensét.

2.4. Vezérlésátadó utasítások

Az eddigiekben láttuk, hogy az utasítások általában növekvő sorszámozás szerinti sorrendben kerülnek végrehajtásra. Ennek a sorrendnek a megváltoztatására az ún. vezérlő utasításokat használjuk.

2.4.1. Feltétel nélküli vezérlésátadás

Ha azt akarjuk előírni, hogy egy utasítás után ne a "soron következő" utasítás kerüljön végrehajtásra, a GOTO utasítást használhatjuk. Az utasítás alakja:

GOTO sorszám

ahol a sorszám a következő végrehajtandó utasítás sorszámát jelenti.

```
Pl.      5   INPUT  Y
          10   X = 2 * Y
          20   GOTO 40
          30   X = X ^ 2
          40   PRINT X
```

A programban a 10-es értékadó utasítás végrehajtása után, a 20 GOTO 40 utasítás hatására a vezérlés a 40-es utasításra adódik át, a 30-as utasítás nem hajtódik végre.

2.4.2. Feltételes vezérlésátadás

Ha a program végrehajtásának természetes sorrendjét valamely feltétel teljesülésétől, vagy nem teljesülésétől függően akarjuk megváltoztatni, erre az IF utasítást használjuk. Az utasítás formái:

IF A1 rel A2 GOTO sorszám

ahol A1, A2 - aritmetikai kifejezések
rel - egy relációjel.

Ha a reláció igaz, a számítógép a GOTO-val megadott sorra ugrik, ha nem igaz, a soron következő utasítást hajtja végre.

IF A1 rel A2 THEN sorszám

ahol A1, A2, rel ugyanaz, mint az előző alaknál.

Mindkét utasításnak ugyanaz a hatása.

Pl. nézzünk egy program részletet, amely két számot hasonlít össze.

```

:
40 INPUT X, Y
50 IF X <= Y THEN 80
60 PRINT "X NAGYOBB MINT Y"
70 GOTO 90
80 PRINT "X KISEBB VAGY EGYENLO Y"
:

```

IF A1 rel A2 THEN sorszám ELSE sorszám

Az IF utasításban, ha a kifejezés "hamis", akkor a gép az ELSE után következő sorszámra ugrik, ill. az ELSE után álló utasítást hajtja végre.

Egy NEW parancs után írjuk be a következő programot!

```

10 INPUT "ADJ HAROM SZAMOT"; X, Y, Z
20 PRINT "A LEGNAGYOBB SZAM:"
30 IF X < Y OR X < Z THEN
   IF Y < Z THEN PRINT Z ELSE
   PRINT Y ELSE PRINT X
40 END

```

A program három számot kér és kiírja a legnagyobbat.

READY

>RUN

ADJ HAROM SZAMOT ? 30, 75, 73

A LEGNAGYOBB SZAM: 75

2.4.3. Többirányú elágaztatás

A BASIC nyelv lehetővé teszi a programok többirányú elágaztatását is az ON utasítással. Az ON utasítás egy aritmetikai kifejezés aktuális értéke szerint választja ki több címke közül a megfelelőt.

Az ON utasítás alakja:

ON aritmetikai kifejezés GOTO 1.sorszám, 2.sorszám...

Az aritmetikai kifejezés értékének 0 és 255 közöttinek kell lenni. Az ON GOTO utasítás végrehajtásakor a gép először az aritmetikai kifejezés egész értékét /n/ számolja ki, majd megkeresi a sorszámlista n-dik elemét és erre a sorra ugrik. Ha n nagyobb, mint a megadott sorszámszámok száma, akkor a gép az ON GOTO utasítást követő utasítást hajtja végre.

ON utasítás segítségével egyszerűen megvalósíthatunk pl. előjel szerinti elágaztatást. Az

50 ON 2 + SGN(X) GOTO 100, 200, 300

utasítás negatív X esetén a 100-as, X = 0 esetén a 200-as, pozitív X esetén a 300-as utasításra adja a vezérlést.

2.4.4. Szubrutin hívás

A vezérlésátadó utasítások között kell megemlítenünk a szubrutin hívó utasításokat is. Gyakran előforduló eset, hogy ugyanazt az utasításcsoportot kell a program különböző helyein végrehajtani. A végrehajtandó utasítássorozat adott esetben igen hosszú is lehet, és ezért célszerűtlen volna azt mindig újra és újra leírni. A kézenfekvő megoldás az, hogy írjuk le a kívánt utasítássorozatot egyetlen egyszer,

és speciális vezérlésátadó utasítással ugrassuk rá erre a sorozatra minden olyan helyről, ahol e sorozat végrehajtását el kívánjuk végeztetni. Ez a speciális utasítás:

GOSUB sorszám

Ez az utasítás arra a sorra adja át a vezérlést, amelyiken a megadott szubrutin kezdődik. Ha a számítógép a szubrutinban RETURN utasítást talál, akkor visszatér a GOSUB utasítást követő utasításra. A GOTO-hoz hasonlóan itt is megadhatunk különböző feltételeket:

IF A1 rel A2 GOSUB sorszám
ON n GOSUB sorszám

2.5. Ciklusok

2.5.1. Ciklusképzés feltételes vezérlésátadással

Ciklusnak nevezünk egy olyan programrészletet, amelyet a futás során többször egymás után végre kell hajtani. Az alábbi programrészlet egy vektor elemeit írja ki:

```

:
100 I = 0
110 IF I>N THEN 150
120 PRINT A (I)
130 I = I + 1
140 GOTO 110
150 PRINT "VEGE A CIKLUSNAK"
:

```

A 120-as PRINT utasítást a ciklus magjának, az I változót ciklus változónak nevezzük. Ebben az esetben feltételes vezérlésátadó és GOTO utasítással szerveztük meg a ciklust.

2.5.2. Ciklusképző utasítások

Mivel a programozásnál a ciklusszervezés igen gyakori feladat, a BASIC külön utasításokat tartalmaz a ciklusok szervezésére.

```
FOR I = A1 TO A2 STEP A3
:
NEXT I
```

ahol:

- I - a ciklus változó
- A1 - aritmetikai kifejezés, amelynek aktuális értéke a ciklus változó kezdőértéke
- A2 - aritmetikai kifejezés, amelynek aktuális értéke a ciklus változó végértéke /TO=ig/
- A3 - növekményt meghatározó aritmetikai kifejezés /STEP = lépésköz/

A ciklus magját a FOR cikluskezdő és NEXT cikluszáró utasítás között elhelyezkedő utasítások alkotják.

Nézzük az előző programot ciklusképző utasítással

```
:
100 FOR I = 0 TO N STEP 1
110 PRINT A(I)
120 NEXT I
130 PRINT "VEGE A CIKLUSNAK"
```

A FOR utasítás első végrehajtásakor a gép a ciklus változót a kezdőértékre állítja be. A programot a következő NEXT utasításig végrehajtja. Itt a ciklus változót a lépésköz értékével megnöveli. Ha a lépésközt nem adtuk meg, automatikusan 1-et állít be. Ha a lépésköz negatív, akkor visszafelé számol. Ezután a gép a ciklus változót összehasonlítja a ciklusváltozó végértékével. Ha a ciklus változó értéke túllépte a végértéket, akkor a gép a NEXT utasítás utáni, ha nem, akkor a FOR utasítás utáni első utasításnál folytatja a program végrehajtását.

A NEXT I helyett egyszerűen NEXT-et is írhatunk. Egymásba skatulyázott ciklusok esetén azonban célszerű megadni azt, hogy a konkrét NEXT-el éppen melyik ciklust zártuk le.

2.5.3. Többszörös ciklusok /Tömbök kezelése/

Gyakran előfordul, hogy egy ciklus magjában újabb ciklusutasítást használunk. Ilyenkor egymásba skatulyázott ciklusról beszélünk. Például legyen a feladat az, hogy egy tömb elemei közül keressük ki a páros számokat és nyomtassuk ki őket.
/Tömb alatt általában vektorokat és mátrixokat értünk/
Egy lehetséges megoldást ad az alábbi programrészlet.

```

:
100 DIM A (8, 12)
110 FOR J = 0 TO 8
120 FOR K = 0 TO 12
130 IF A(J,K) > 2 * INT ( A(J,K) / 2 ) THEN 150
140 PRINT A(J,K)
150 NEXT K
160 NEXT J
:

```

A programrészlet 120-150-ig tartó /K/ ciklusa az A tömb egy sorából nyomtatja ki a páros értékű elemeket. Hogy a tömb melyik sorát vizsgálja éppen, azt J pillanatnyi értéke határozza meg.

Az egymásba skatulyázott ciklusokkal kapcsolatban a következő szabályokra szeretnénk felhívni a figyelmet.

- Valamennyi belső ciklus minden utasítása, beleértve a belső ciklus kezdő és záróutasítását /FOR-NEXT/ is, a külső ciklus ciklus-magjához kell, hogy tartozzon. Átlapolás tilos.
- A ciklus magjában a ciklus változónak nem adhatunk értéket.
- Tilos egy ciklus magjára kívülről ráugrani, vagy a ciklus magjából a cikluskezdő FOR utasításra adni a vezérlést.

Láthatjuk, hogy a ciklus szervezéssel egy tömb minden eleméhez hozzáférhetünk. Így az egyes elemeknek értéket adhatunk, műveleteket végezhetünk vele, stb. A számítógép szám-és karakter tömböket egyaránt tud kezelni.

2.6. Karakterfüzerek /stringek/ kezelése

Az adatfeldolgozásban a karakterfüzereknek nagy jelentőségük van.

A karakterek egy része nyomtatható /pl. betűk, számok/, más részük nem nyomtatható /betűköz, kocsi-vissza, vezérlőkarakterek, stb/. Minden karakterhez tartozik egy számérték, a karakter kódja. A kód teszi lehetővé a karakterek számítógépben való ábrázolását, ill. a velük való műveletvégzést. A BASIC nyelv az ASCII kódrendszert használja. Az ASCII kód-táblázatot a függelékben közöljük. /F 1/

A karakterek tárolására karakteres változókat és tömböket használunk. Jelölésük a változónév után tett \$ jellel /Pl. A\$, B\$(1)/ történik. A változó értékén az adott karakter-sorozatot értjük. Egy karakteres változó hosszúsága maximum 255 karakter lehet.

2.6.1. Hozzárendelés, összehasonlítás, összekapcsolás

Karakterláncot, vagy karakteres konstanst általában idézőjelek közé zárva írunk fel, így rendelünk értéket egy karakteres változóhoz.

Pl. A\$ = "IGEN"

Az aritmetikai kifejezésekhez hasonlóan, két fűzér értékét is összehasonlíthatjuk az =, <, >, <=, >=, <> relációs jelek segítségével. Két fűzért akkor tekintünk egyenlőnek, ha karakterről karakterre megegyeznek. Az összehasonlítást a karakterek ASCII kódja alapján értelmezzük. A betűkre az ABC sorrendje érvényes "A" < "Z".

Ha különböző hosszúságú fűzereket hasonlítunk össze, akkor a hosszabbnak az értéke nagyobb /ha különben azonosak/.

Pl. "B" < "B "

Az IF utasításra fűzér relációk esetén ugyanazok a szabályok vonatkoznak, mint aritmetikai relációknál, csupán itt aritmetikai kifejezések helyett karakter kifejezéseket hasonlítunk össze.

Pl. 50 IF A\$ = "IGAZ" THEN 200

Két karakterfüzért a + művelet segítségével kapcsolhatunk össze.

```
Pl.  10  A$ = "SZEPTEMBER"
      20  B$ = "1983 " + A$
      30  C$ = B$ + " 15"
      40  PRINT C$
```

Az utasítások végrehajtása után a gép kiírja

1983 SZEPTEMBER 15

2.6.2. Füzőkre vonatkozó függvények

- | | |
|---------------------|---|
| ASC (füzér) | - Megadja a füzér első karakterének ASCII kódját. A füzérnek idézőjelben kell lenni. |
| CHR\$ (kifejezés) | - Az előző függvény inverze, azaz megadja a kifejezés értékéhez tartozó karaktert. |
| LEFT\$ (füzér, n) | - Kiválasztja a füzér első n karakterét. A füzér lehet egy kifejezés vagy egy konstans, az n pedig numerikus kifejezés vagy állandó. |
| RIGHT\$ (füzér, n) | - Kiválasztja a füzér utolsó n karakterét. A füzér lehet karakteres változó vagy konstans, n pedig numerikus változó vagy konstans. Ha a füzér hossza kisebb, vagy egyenlő n-nel, akkor az egész füzért adja értékként. |
| LEN (füzér) | - Megadja a füzér hosszát. |
| MID\$ (füzér, p, n) | - Megadja a füzér egy részét a p pozíciótól kezdve. A rész-füzér hossza n karakter. |
| STR\$ (kifejezés) | - Egy szám-konstanst, vagy kifejezést karakterfüzérre alakít át. |

STRING\$ (n,karakter vagy szám) - n hosszúságú, az adott karakterből álló stringet, fűzért állít elő. A számot a gép ASCII kódként kezeli és az utasítás az ennek a kódnak megfelelő jelet vagy grafikus kódot állítja elő.

VAL (fűzér)

- Numerikus értéket állít elő a fűzér karaktereiből. /Az STR\$ inverze/

Példa: Nézzünk egy játékprogramot. A program generál egy négyjegyű számot, amit ki kell találnunk. Segítségképpen minden egyes tippnél kiírja a gép, hogy hány számot találtunk el, ami helyre és értékre is megfelelő és hányat, ami csak értékre jó.

```

10 CLS
20 Y = RND (9999)
30 IF Y < 1000 THEN 20
40 B$ = STR$ (Y)
50 INPUT "KEREK EGY NEGYJEGYU SZAMOT:" ;X
60 A$ = STR$ (X)
70 HJ = 0 : EJ = 0
80 FOR I = 2 TO 5
90 FOR J = 2 TO 5
100 IF MID$ (B$, I, 1) <> MID$ (A$, J, 1) THEN 130
110 EJ = EJ + 1
120 IF I = J THEN HJ = HJ + 1
130 NEXT J
140 NEXT I
150 PRINT "HELYRE ES ERTEKRE JO" ; HJ
160 PRINT "ERTEKRE JO"; EJ
170 IF HJ < 4 THEN 50
180 END

```

Az STR\$ (X) utasítás a számot következőképpen alakítja át karakterfűzérré.

Pl. X = 12
 STR\$ (X) = " 12"

2.7. Értékadás

2.7.1. Belső értékadás

A BASIC nyelvben az értékadó utasítás általános alakja

LET változó = kifejezés

Az értékadó utasítás végrehajtása során a gép először kiszámítja az egyenlőség jobb oldalán álló kifejezés aktuális értékét, majd hozzárendeli az egyenlőséggel bal oldalán álló változóhoz.

Ha indexes változónak adunk értéket, a gép először az indexkifejezést értékeli, és az így meghatározott tömbelemnek adja a jobboldali kifejezés értékét.

A mi rendszerünkben a LET szót nem feltétlenül szükséges kitenni.

A futás során minden programnak szüksége van adatokra. Az adatokat magában a programban is elhelyezhetjük DATA utasítás segítségével és READ utasítással olvassuk be.

DATA adat, adat

Az adatokat a gép szekvenciálisan veszi elő. Minden adat karakteres és numerikus konstans is lehet. Azonban a tételek rendezésekor ügyelni kell arra, hogy az egyes adatok típusa a DATA és a változók típusa a hozzá tartozó READ utasításban azonos legyen. A DATA utasítás a program tetszőleges helyén lehet.

Az adatmezőben elhelyezett adatok beolvasása a READ utasítással történik.

READ változó, változó

A DATA utasítás értékeit a READ utasítás sorrendben beolvassa. Ha az első DATA utasítás adatait már kiolvastuk, akkor a következő READ utasítás a következő DATA utasítás első adatát

olvassa ki. Ha az összes adatot kiolvastuk és READ utasítással további értékeket akarunk elérni, a gép hibát jelez./OD/

```
Pl.      :
          :
10 DATA 2, 2, 3, 5, 8
20 DATA 4, 7, 11, 19, 5
30 READ X1
40 FOR K = 1 TO 7
50 READ W ( K )
60 NEXT K
70 READ X2
          :
```

Nézzük meg milyen értékeket vesznek fel az egyes változók.

```
X1 = 2;
W(K) = 2, 3, 5, 8, 4, 7, 11;
X2 = 19
```

Előfordulhat, hogy a program futása során ismét előről akarjuk kezdeni az adatmező beolvasását. Erre a célra egy külön utasítás szolgál, a

RESTORE

Az utasítás lehetővé teszi, hogy a következő READ utasítással az első DATA lista első adatát olvashassuk ki akkor is, ha már voltak ezelőtt READ utasítások.

```
Pl.      10 READ X
          20 RESTORE
          30 READ Y
          40 PRINT X, Y
          50 DATA 50, 60
          60 END
```

RUN

```
50          50
READY
>
```

2.7.2. Külső értékátadás

A program futása során szükség lehet olyan adatokra is, amelyeket a program írásakor még nem ismerünk. Ezeket futás közben kell beolvasni. Erre szolgál az INPUT utasítás.

INPUT változónév, változónév.....

Az utasítás hatására a számítógép bizonyos típusú és számú adatot vár. A beírások sorrendjére figyelni kell. A számítógép egy kérdőjelet `/?` ír a képernyőre és várja az adatot/ka/t.

Az adatokat két módon írhatjuk be:

- Megfelelő számú és típusú adatot vesszővel választunk el egymástól és utána megnyomjuk a `NEW LINE` billentyűt.
- Minden adatot külön sorba írunk

? adat `NEW LINE`
 ?? adat `NEW LINE`
 ??

amíg az összes adatot be nem irtuk, addig a képernyőn két kérdőjel látható.

Annak érdekében, hogy egyértelmű legyen a felhasználónak, hogy melyik adatot várja a gép, az INPUT utasításhoz megjegyzést írhatunk, amelyet a gép kinyomtat.

INPUT "megjegyzés" ;változónév,változónév.....

A megjegyzésnek - idézőjelek között -, közvetlenül az INPUT kulcsszó mögött kell következnie, lezárását pontosvesszővel jelöljük.

```
Pl.  :
      :
      100 INPUT "NEVE ES SZULETESI EVE"; N$, SE
      :
      :
      RUN 100
      NEVE ES SZULETESI EVE ?
```

Programozás közben előfordulhat, hogy egy karaktert akarunk beolvasni a billentyűzetről, ilyenkor az INKEY\$ utasítást használjuk. Ha az utasítás végrehajtásával egyidőben nem

ütünk le billentyűt, akkor "" karaktert állít elő a gép. Az INKEY\$ utasítással beolvasott adatok a képernyőn nem jelennek meg.

Ezzel az utasítással karaktereket olvashatunk be. Pl.

```
90 PRINT "HÁROM BILLENTYŰ BEOLVASASA"
100 A$ = INKEY$ : IFA$ = " " THEN 100 ELSE PRINT A$ ;
110 B$ = INKEY$ : IFB$ = " " THEN 110 ELSE PRINT B$ ;
120 C$ = INKEY$ : IFC$ = " " THEN 120 ELSE PRINT C$ ;
130 D$ = A$ + B$ + C$
```

A három billentyű kódja a D\$ -ben megtalálható.

2.8. A PRINT utasítás alkalmazásai

Az I. fejezet 5. pontjában /1.5/ megismerkedhettünk az egyszerű nyomtatási utasítással /PRINT/. Egyszerűbb alkalmazásokra ez az utasítás kielégítő, bonyolultabb számítások, vagy speciális programok esetén azonban szükségessé válhat egy "formatervezettebb" nyomtatási kép kialakítása. Erre szolgálnak a következő speciális PRINT utasítások:

2.8.1. PRINT@ mutató, tételsor

Ez az utasítás a tételsort a képernyő "mutató" által meghatározott helyére nyomtatja. A mutató értéke 0 és 1023 között lehet /ld. F3/.

Pl. 100 PRINT@ 550, "550-ES HELY"

Ha ezzel az utasítással az utolsó sorba akarunk kiírni, akkor az utasítás végére egy pontosvesszőt kell tennünk, hogy az automatikus soretolódást megakadályozzuk.

Pl. 150 PRINT@ 1000, 1000;

2.8.2. PRINT TAB /kifejezés/

Ez az utasítás lehetővé teszi, hogy a kurzort egy adott sorban tetszőleges helyre vigyük. A kifejezés értékének 0 és 255 között kell lennie, ha az értéke 63-nál nagyobb, akkor a kurzor a következő sorra ugrik.

```
Pl. 10 PRINT TAB ( 5 ) "5-OS POZICIO"; TAB ( 25 )
      "25-OS POZICIO"
```

A TAB-ot nagyon gyakran használjuk pl. táblázatok készítésére, matematikai függvények grafikus ábrázolására stb.

```
Pl. 10 X = 3
      20 PRINT TAB(X)X
      30 PRINT TAB(X/2)X/2
      40 PRINT TAB(X/3)X/3
```

2.8.3. PRINT USING karakteres kifejezés, nyomtatási lista

A karakteres kifejezés aktuális értéke egy ún. formátum vezérlő füzért ad meg. Ez az utasítás a kifejezésben megadott formátumban nyomtatja ki az adatokat.

A formátum vezérlő kifejezések felépítésében bizonyos karaktereknek megkülönböztető szerep jut, ezeket a karaktereket formátum vezérlő karaktereknek nevezzük.

A legegyszerűbb formátum vezérlő karakterek:

#

Számjegy - ezzel határozhatjuk meg a nyomtatási lista elemeinél a számjegyek számát, pl. 12.58-at akarunk ki-nyomtatni, akkor a formátum

##.##

Ha a megadott mező kisebb, mint a szám, akkor a gép % jelet ír ki a szám elé.

**

Ha két csillagot írunk a formátum-mező elejére, akkor a tizedesponttól balra lévő be nem töltött helyekre csillagokat ír a gép.

- \$\$** Ha a formátummező elején két dollár-jelet írunk, akkor a gép a legnagyobb helyiértékű szám elé \$ jelet ír.
- **\$\$** Ebben az esetben minden üres helyre csillag, a legnagyobb helyiértékű szám elé \$ jel kerül.
- +** Ha ez a jel szerepel a formátummező elején, vagy végén, a gép kiírja a számok előjeleit.
- Ha - jel van a formátummező végén, akkor a gép minden pozitív szám után kihagy egy karaktert és - jelet ír minden negatív szám elé.
- % betűköz %** Karakter mező definiálása. A karakter mező hossza: betűközök száma + 2.
- !** Ha felkiáltójelet írunk, akkor a gép csak az aktuális füzér első karakterét használja. A felkiáltójel segítségével összefűzhetők a füzérek. Ha több felkiáltójelet használunk, akkor a gép a füzér első betűjét annyi karakterrel később nyomtatja ki, ahány üres karakter a felkiáltójelek között van.

A következő program szemlélteti a fentiek használatát.

1. példa

```
10 INPUT A$, A
20 PRINT USING A$; A
30 GOTO 10
```

RUN

? ##.## , 12.12

12.12

? ###.## , 12.12
12.12

? ##.## , 121.21

% 121.21

? ##.## , 12.127
12.13

? +##.## , 12.12
+ 12.12

? +##.## , -12.12
-12.12

? ##.## + , 12.12
12.12 +

? ##.## + , -12.12
12.12 -

? ##.## - , 12.12
12.12

? ##.## - , -12.12
12.12 -

? \$\$\$##.## , 12.12
\$ 12.12

Az utasítás karakterfüzérek kiíratásánál is alkalmazható:

2. példa

```
10 INPUT A$, B$
20 PRINT USING A$; B$
30 GOTO 10
RUN
? !, ABCD
A
? %%, ABCD
```

```

AB
?%    %, ABCD
      ABCD

```

3. példa

```

10 INPUT A$, B$, C$
20 PRINT USING "!"; A$, B$, C$
RUN
? ABC, DEF, GHI
  ADG

```

4. példa

```

10 INPUT A$, B$, C$
20 PRINT USING "!  !  !"; A$, B$, C$
RUN
? ABC, DEF, GHI
  A  D  G

```

2.9. Adat file-ok kezelése programból2.9.1. Adatok felírása kazettára

PRINT# - kazettaszám, tételsorozat

A rendszer két magnetofont tud kezelni, a kazettaszámmal adjuk meg, hogy melyikre akarjuk kivinni a tételsort. A tételsor elemeinek össz karakter száma nem lehet több 255-nél. Az ezen felül lévő karaktereket a gép nem veszi fel, elvesznek.

Pl. 10 PRINT#-1, A\$, B\$, C, "ITT A VEGE"

2.9.2 Adatok betöltése kazettáról

INPUT# - kazettaszám, tételsorozat

Az előző utasítás inverze.

Pl. 10 INPUT#-1, A\$, C\$, B, D\$

Ügyelnünk kell arra, hogy visszaolvasáskor az adatok típusa megegyezzen a PRINT utasításban megadottakkal. Természetesen a változóneveknek nem kell megegyezniük.

2.10. Képernyő grafika

A számítógép lehetővé teszi, hogy a felhasználó minden grafikus feladatot a képernyőre kirajzoltasson. A képernyő 128 vízszintes és 48 függőleges mezőre van felosztva /ld. F 3/. Az egyes mezőket X/O-127/ és Y/O-47/ koordinátákkal adhatjuk meg. X és Y lehet konstans, változó, vagy kifejezés. A grafikus utasítások X és Y egész részét használják fel. A számítógép a következő grafikus utasításokat ismeri:

SET(x,y)

- Bekapcsolja az /x, y/ koordinátákkal megadott mezőt.

RESET(x, y)

- Kikapcsolja az /x, y/ koordinátákkal megadott mezőt.

CLS

- Törli a képernyőt és a kurzort a képernyő bal felső sarkába viszi.

POINT(x, y)

- Megvizsgálja az /x, y/ koordinátákkal megadott grafikus mezőt. Ha be van töltve a blokk, a POINT értéke -1 /logikai igaz/, ha nincs, akkor 0 /logikai hamis/.

Példa: Vegyünk két 15x15-ös mezőt a képernyőn. Az egyik mező összes pontja legyen ki-, a másik mező összes koordinátája legyen bekapcsolva. Véletlenszerűen kapcsoljunk be 1 pontot az első mezőn és az ennek megfelelő pontot a másik mezőn kapcsoljuk ki./Lásd a következő oldalt!/

2.11. Egyéb lehetőségek

A REM utasítással megjegyzéseket írhatunk a programba, ezek a program végrehajtását nem befolyásolják, csak a programlistában jelennek meg.

Például:

```

:
110 REM  SZAMITAS KEZDETE
:
130 REM  SZAMITAS VEGE

```

A REM kulcsszó után tetszőleges karakterek írhatók.

Lehetőség van összetett utasítássorok használatára is. Egy utasítássorban kettősponttal `:/` elválasztva több utasítás is megadható, így csökkenthető az utasítássorok száma.

Például:

```

10 CLS : FOR I = 1 TO 127 : PRINT CHR$(I); : NEXT I
    irható
10 CLS
20 FOR I = 1 TO 127
30 PRINT CHR$(I);
40 NEXT I

```

helyett.

```

5 CLS
10 FOR I = 1 TO 15
20 FOR J = 1 TO 15
30 SET ( I + 61, J + 16 )
40 NEXT J
50 NEXT I
60 X = RND(15)
70 Y = RND(15)
80 SET ( X + 46, Y + 16 )
90 RESET ( X + 61, Y + 16 )
100 GOTO 60

```

A grafikus utasítások között említjük meg a következő függvényt:

POS(X) - Megadja a kurzor soron belüli pozícióját
 /0 - 61 között/, X értékeként bármi megadható.

3. EGYÉB FUNKCIÓK

3.1. Parancsok

A berendezés parancs üzemmódban a sorszám /cimke/ nélkül irt sorokat parancsként értelmezi, és a **NEW LINE** billentyű lenyomása után megkísérli végrehajtani.

A 2. fejezetben leírt programutasítások egy része parancsként is végrehajtható.

Az alábbiakban a berendezés vezérlésére szolgáló - üzemmód-váltó, törlő, beolvasó, stb. - parancsokat ismertetjük. A felsorolás tartalmazza az 1. fejezetben már ismertetett parancsokat is.

AUTO	CONT	LIST	RUN	SHIFT	↓	P
CLEAR	CSAVE	LLIST	SYSTEM			
CLOAD	DELETE	NEW	TRON			
CLOAD ?	EDIT	RE	TROFF			

A parancsok - a CONT kivételével - utasításokként programban is megadhatók!

3.1.1. AUTO sorszám, növekmény

Programíráskor automatikus cimke /sorszám/ kiírást és növelést eredményez. A "sorszám" a legkisebb cimke értékét, a "növekmény" az egyes utasítás sorszámok közötti értéket határozza meg. Például:

AUTO 25, 5 után

25- , majd a 25-ös utasítássor beírása, lezárása után
30- stb. következik.

A parancsot sorszám és növekmény nélkül megadva:

AUTO a kezdő cimke és a növekmény is 10-es értéket vesz fel /10, 20, 30,.../

Ha már létező sorszámú sorhoz érünk, a gép ezt "*" kiírásával jelzi.

Az AUTO üzemmód a BREAK billentyűvel szüntethető meg.

3.1.2. CLEAR n

A karakterfüzerek tárolására n helyet /byte-ot/ foglal le, és törli az előzőleg használt változókat.

CLEAR 80 Törli a változókat, és max. 80 karakteres füzerek használatát teszi lehetővé.

Bekapcsolás után CLEAR 50 érvényes.

Az n nélkül megadott CLEAR nem változtatja meg a füzerek max. hosszát.

3.1.3. CLOAD #-egység száma, "file neve"

Betölti a megadott sorszámú kazettás egységről a megadott nevű file-t. A névnek csak az első karakterét veszi figyelembe! Például:

CLOAD # -1, "PROGRAM"

a beépített egységről betölti az első "p"-vel kezdődő nevű file-t. /A külső kazettás egység száma: 2/

A sorszám és név nélkül megadott parancs a soronkövetkező file-t tölti be a beépített egységről:

CLOAD /Lásd még 1.2.4/

3.1.4. CLOAD ?

Használata hasonló a CLOAD parancséhoz.

Betöltés helyett a megadott file-t összehasonlítja a gépben lévő programmal. Hiba /eltérés/ esetén "BAD" /hibás/ üzenetet ír ki. Programok kazettára vételekor a felvétel ellenőrzésére használjuk! /Lásd 1.2.4./

3.1.5. CONT

A STOP utasítás, vagy a **BREAK** billentyű hatására megszakított program folytatását eredményezi.

CONT3.1.6. CSAVE#-egység száma, "file neve"

A gépben lévő programot a megadott sorszámmal kazettás egységre, a megadott névvel írja /veszi/ fel.
/Lásd 1.2.4./

3.1.7. DELETE sorszám - sorszám

Törli a megadott sorszámú - és a köztük lévő - utasításokat.

Például:

DELETE

Törli a legutoljára kezelt sort.

DELETE 25

Törli a 25 sorszámú sort.

DELETE 10 - 30

Törli a 10, 30 és a köztük lévő sorokat.

DELETE - 35

A 35 sorszámú sorig minden sort töröl.

3.1.8. EDIT sorszám

A számítógépet EDIT /szövegszerkesztés/ üzemmódba helyezi és lehetővé teszi a megadott sorszámú utasítássor módosítását, átszerkesztését.

EDIT 100

Az EDIT parancs részletes leírása a 3.2. fejezetben.

3.1.9. LIST sorszám -sorszám

A gépben lévő program megadott sorszámú - és a közöttük lévő - utasításait kiírja /"listázza"/ a képernyőre.

LIST

A teljes programot kiírja. A kiíratás a **[SHIFT]@** billentyűkombinációval állítható le, és bármely más billentyű lenyomásával folytatható.

LIST 25

A 25 sorszámú sort írja ki.

LIST 10-30

A 10, 50 és a közöttük lévő sorokat írja ki.

LIST - 50

Az 50 és az előtte levő összes sort írja ki.

LIST 50 -

Az 50 és az utána levő összes sort írja ki.

LIST .

Az utoljára kezelt /beírt, szerkesztett/ sort írja ki.

3.1.10. LLIST sorszám - sorszám

A megadott utasítássort, vagy sorokat nyomtatóra írja ki.

Használata megegyezik a LIST parancsával.

Csak nyomtatót is tartalmazó összeállításnál használható!

3.1.11. NEW

Törli a tárban lévő programot és a változókat. Új program beírása előtt alkalmazzuk.

3.1.12. RE sorszám, növekmény

A gépben lévő program sorainak újraszámolására szolgál. A "sorszám" a legkisebb új címkét, a "növekmény" a sorszámok közötti különbséget határozza meg. A teljes programot újra címkézi. Ha a sorszámot, vagy növekményt nem adjuk meg, akkor az 10-es értéket vesz fel.

RE 30, 2

Az utasítások 30, 32, 34... sorszámúak lesznek.

RE , 2

10, 12, 14... sorszámokat állít be.

RE 100,

100, 110, 120... sorszámokat határozz meg.

RE

A sorszámok 10, 20, 30... értéket vesznek fel.

Ez a parancs csak BŐVITETT BASIC-nél használható!
 /SYSTEM /12288 beírás után! Lásd 1.2.3. fejezet!/

3.1.13. RUN sorszám

A gépben lévő program végrehajtását indítja a megadott sorszámú utasítástól.

Sorszám nélküli megadásnál a program legkisebb sorszámú utasításánál kezdi a végrehajtást.

RUN 100RUN3.1.14. SYSTEM

Gépi kódú programok kazettáról való betöltésére és a tárban lévő gépi kódú programok elindítására szolgál. /Részletesebben lásd az 1.2.4. II.-ben/

> SYSTEM*? GALAXY

Betölti a GALAXY nevű gépi kódú programot.

*? /

Elindítja a betöltött programot.

> SYSTEM*? /12288

Elindítja a BŐVITETT BASIC opció /kisbetűk, villogó kurzor, stb./ használatát lehetővé tevő gépi kódú programot az 12288 címről.

3.1.15. TRON

Beállítja a "nyomkövetés" /TRACE/ üzemmódot. Hatására a program futásakor kiíródnak a végrehajtott utasítások sorszámai.

3.1.16. TROFF

Megszünteti a "nyomkövetés" üzemmódot.

3.1.17. [SHIFT] [↓] P

A gombokat az adott sorrendben megnyomva /és nyomva tartva/ a képernyő tartalmát kiírja a nyomtatóra. Csak BŐVITETT BASIC-nél, nyomtatóval használható!

3.2. Editálás /szövegszerkesztés/

A gépben lévő programot a parancs üzemmódból EDIT üzemmódba áttérve lehet javítani, módosítani, átszerkeszteni. EDIT /szerkesztés/ üzemmódba az EDIT parancs és a módosítandó utasítás-sor sorszámanak megadásával térhetünk át:

EDIT sorszám

Sorszám helyett pontot is beírhatunk, ebben az esetben az utoljára kezelt /beírt, listázott, szerkesztett, stb./ utasítássort lehet módosítani:

EDIT .

EDIT üzemmódban az alábbi alparancsok és billentyűk használhatóak:

NEW LINE

: visszatérés parancs üzemmódba az átírt sorral

n **SZÓKÖZ**

: kurzor mozgatás jobbra n hellyel

n **[←]**

: kurzor mozgatás balra n hellyel

SHIFT **[↑]**

: szövegbeíró alparancsok /X, I, H/ befejezése

L	: az utasítássor listázása
A	: a változtatások törlése
X	: sor végére állás és beírás /kiegészítés/
I	: beírás a kurzor pozícióján /beszúrás/
H	: törlés a sor elejétől a kurzorig és beírás
E	: visszatérés parancs üzemmódba az átírt sorral
Q	: visszatérés parancs üzemmódba az eredeti sorral
n D	: n karakter törlése a karaktertől jobbra
n K c	: a kurzortól az n. "c" karakterig teljes törlés
n S c	: a kurzor beállítása az n. "c" karakterre
n C c	: a kurzortól jobbra a karakter átírása a megadott "c" karakterekre /n db!/

/Az "n" decimális számot, a "c" egy, vagy több karaktert jelent./

A szerkesztő alparancsok beíráskor nem jelennek meg a képernyőn!

A továbbiakban a szerkesztés funkcióit, a kurzor mozgatását, a karakterek törlését, beszúrását egy rövid programon mutatjuk be.

Egy NEW parancs után írjuk be az alábbi programot:

```

10 PRINT SZERKESZTESI PELDA
20 FOR I = 1 TO 100 STEP 1
30 INPUT CHR$(I)
40 NEXT J

```

A fenti - hibás - utasításokat úgy szerkesztjük át, hogy a következő programot kapjuk:

```

10 CLS : PRINT "PELDA", "KARAKTERKESZLET"
20 FOR I = 32 TO 192
30 PRINT CHR$(I); " ";
40 NEXT I : END

```

3.2.1. A kurzor pozicionálása

Az eredeti utasítássor megváltoztatása nélkül a kurzort az alábbi alparancsokkal állíthatjuk a kívánt helyre:

n

n

n S c

Az előzőekben leírt /hibás/ program beírása után térjünk át a 10 sor editálására!

>EDIT 10

E parancs hatására a gép EDIT üzemmódba tér át, lehetővé téve a 10 címkéjű utasítássor szerkesztését, és kiírja a megadott sor címke részét:

10 _

A billentyű leütésére a kurzor egy pozícióval jobbra lép. Többször - pl. ötször - leütve, a kurzor öt hellyel lép jobbra, és kiíródik az első öt pozíción lévő karakter:

10 PRINT _

Több pozícióval való jobbra mozgathatáshoz írjuk be a kívánt lépésszámot, majd nyomjuk le a szököz billentyűt!

6

Ennek hatására a kurzor újabb 6 hellyel lép jobbra:

10 PRINT SZERK _

Ha a kurzortól jobbra lévő karakterek számánál nagyobb számot adunk meg, akkor a kurzor a sor végére áll:

30 SZOKOZ

10 PRINT SZERKESZTESI PELDA _

A ← billentyűvel a kurzort balra mozgathatjuk egy, vagy több hellyel. Például:

9 ←

10 PRINT SZERKESZT _

A kurzor balra mozgatásánál a karakterek csak a képernyőről tűnnek el, az utasítássorból nem törölődnek!

A kurzortól balra lévő karakterek számánál nagyobb számot megadva a kurzor a sor címke utáni pozíciójára áll be:

25 ←

10 _

Az n S c formájú /S: "search" - keresés/ paranccsal a kurzort a - pillanatnyi pozíciójától jobbra lévő - n. "c" karakterre lehet beállítani. Például:

2 S E megkeresi a második "E" karaktert a sorban:

10 PRINT SZERK _

A keresett karakter már nem jelenik meg a képernyőn!

Amennyiben nem adunk meg számot, úgy az első, - a kurzortól jobbra lévő - keresett karakterre fog mutatni a kurzor:

S L

10 PRINT SZERKESZTESI PE _

Ha a megadott karakter a kurzortól jobbra nem található, akkor a kurzor a sor végére fog mutatni:

S P

10 PRINT SZERKESZTESI PELDA _

3.2.2. Kiiratás

Az L alparanccsal /listázás/ szerkesztés közben - ha nem I, X, vagy H műveletet hajtunk végre /lásd később/ - a sor kurzortól jobbra lévő részét irathatjuk ki. Utána új sorban újból a címke jelenik meg. Például:

```

12  beírása, és
10 PRINT SZERKE__ megjelenése után
L beírására:
10 PRINT SZERKESZTESI PELDA
10 __
kerül kijelzésre.
```

Az L alparanccsal a szerkesztett sor utolsó, módosított állapotát irathatjuk ki.

3.2.3. Törlési funkciók

Az utasítássor egyes karaktereit, vagy karaktercsoportjait az alábbi alparancsokkal törölhetjük:

```

n D
n K c
```

Az n D alparanccsal az utasítássor n darab karakterét törölhetjük a kurzortól kezdődően. A törölt karakterek a képernyőn felkiáltó jelek között jelennek meg.

Állítsuk a kurzort a 10 sor "S" karakterére,

```
10 PRINT __
```

majd írjuk be: 3 D

```
10 PRINT !SZE! __
```

Az n K c alparanccsal a kurzortól az n. "c" karakterig törölhetünk:

```
2 K E
```

```
10 PRINT !SZE!!RKESZT! __
```

Az n. "c" karakter /példánkban a második "E"/ már nem törlődik!

Kétszeri kiíratással meggyőződhetünk a kívánt karakterek törléséről:

```

L
10 PRINT !SZE!!RKESZT!ESI PELDA
10 _
L PRINT ESI PELDA
10 _

```

A fentieken kívül a H és X alparancsok /lásd később/ is rendelkeznek törlési funkcióval.

3.2.4. Módosítások törlése

Az A alparancs törli valamennyi változtatást, és visszaállítja az eredeti utasításort. A kurzort a sor elejére állítja:

```

10 PRINT ESI PELDA
A
10 _

```

Listázással győződjünk meg az eredeti állapot visszaállításáról:

```

L
10 PRINT SZERKESZTESI PELDA
10 _

```

Az A alparancs csak az EDIT üzemmódban hatásos!

Parancs üzemmódba való visszatérés után már nem állítható vissza az eredeti állapot!!!

3.2.5. Visszatérés parancs üzemmódba

Egy-egy utasításor szerkesztésének befejezésekor az EDIT üzemmódból vissza kell térni parancs üzemmódba. Ez az alábbi

alparancsokkal lehetséges:

NEW LINE

E

Q

A **NEW LINE** billentyű, vagy az E alparancs hatására a gép az átszerkesztett sorral visszatér parancs üzemmódba.

Az E alparancs az I, H, X alparancsok végrehajtása közben nem alkalmazható!

A Q alparancs az eredeti, szerkesztés előtti sorral állítja vissza a parancs üzemmódot. /Hatása azonos egy A és E alparancsával./

Az I, H, X alparancsoknál ez sem használható!

3.2.6. Beszúrás, kiegészítés

A szerkesztés alatt lévő utasítássorba újabb karaktereket, az alábbi alparancsok segítségével írhatunk be:

X, I, H

Ezek az alparancsok EDIT üzemmódban kiadva úgynevezett "Inzert" - beszúrás, beiktatás üzemmódot hoznak létre. Ekkor a leütött karakterek beíródnak a kurzor által meghatározott helyre. A  billentyű ezekben az üzemmódokban törli a kurzortól balra lévő karaktert:

Az inzertálást a

SHIFT 

billentyűk lenyomásával fejezhetjük be és térhetünk vissza normál EDIT üzemmódba.

Az X alparancs a kurzort a sor végére állítja, és lehetőséget ad a sor végének folytatására újabb karakterek beírásával.

NEW LINE

> EDIT 10

10 _

X

10 PRINT SZERKESZTESI PEIDA _

A most beírt karakterek a sor végére íródnak:

```
10 PRINT SZERKESZTESI PELDA , "KARAKTERKESZLET"
```

A SHIFT ↑ alparanccsal befejezve a kiegészítést L alparanccsal győződünk meg a változásról!

Az I alparancs lehetővé teszi a kurzor által meghatározott helyre karakterek beszúrását.

S, -vel állítsuk a kurzort a , -re:

```
10 PRINT SZERKESZTESI PELDA _ majd
```

I " SHIFT ↑ műveletekkel szúrjuk be az "-et!

L -el listázzuk ki a sort:

```
10 PRINT SZERKESZTESI PELDA", "KARAKTERKESZLET" _
```

```
10 _
```

A sor elejére szúrjuk be a CLS: utasítást:

```
I CLS: SHIFT ↓
```

```
10 CLS: _
```

majd a PELDA első karaktere elé állva, és I üzemmódot beállítva

```
2 S P I
```

```
10 CLS : PRINT SZERKESZTESI _
```

töröljük a ← billentyű ismételt lenyomásával a SZERKESZTESI szövegrészt, majd írjuk be a hiányzó idézőjelet:

" és lépünk ki a beszúrás üzemmódból: SHIFT ↑

A H alparancs törli a sor kurzor utáni részét, és lehetővé teszi ide más karakterek beírását.

3.2.7. Karakterek kicserélése

Az n C c alparanccsal a kurzortól jobbra lévő n darab karaktert cserélhetjük /írhatjuk/ át a megadott c karaktersorozatra. /A c most n darab tetszőleges karaktert jelent./

Térjünk vissza egy `NEW LINE` -nal parancs üzemmódba, majd

```
>EDIT 30
```

```
30 _
```

```
  L
```

```
30 INPUT CHR$ /I/
```

```
30 _
```

után az INPUT szövegrészt cseréljük át PRINT-re!

```
5C PRINT
```

```
  L
```

```
30 PRINT CHR$( I )
```

A fentiek alapján hajtsuk végre a példaprogram szükséges módosítását, majd futtassuk a programot!

3.3. Hangeffektusok programozása

A berendezés háromcsatornás programozható hanggenerátora sokféle hangeffektus előállítását teszi lehetővé.

Az egyes csatornák frekvenciája /hangmagassága/, amplitúdója /hangerőssége/ külön-külön programozható. Emellett lehetőség van egy programozható frekvenciájú zajgenerátor bármelyik csatornára kapcsolására, valamint a kimeneti szint /hangerősség/ különböző burkológörbék szerinti változtatására.

A három hangcsatorna paramétereinek /hangmagasság, hangerősség, zaj, változó hangerő/ tárolását 14 kimeneti regiszter /R0 - R13/ végzi.

Az egyes regiszterek kiválasztása az

`OUT 31, I` utasítással, a kiválasztott regiszter beírása az

`OUT 30, P` utasítással történik, ahol

I: a regiszter sorszáma /0 - 14 között/

P: a regiszterbe írandó paraméter értéke.

3.3.1. Hangmagasság beállítása

Egy-egy csatorna hangmagasságát két-két regiszter együttes tartalma határozza meg.

A kisebb /páros/ sorszámú /R0, R2, R4/ regiszterek a hangmagasság finom beállítására szolgálnak. Értékük 0 - 255 között változtatható.

A nagyobb /páratlan/ sorszámú /R1, R3, R5/ regiszterek a hangmagasság durva beállítását végzik. Értékük 0 - 15 között változtatható. Egy értékkel történő változtatásuk a finomszabályozó egy értékváltozásának 256-szorosát jelenti.

Ilymódon egy csatorna hangmagassága a nulla /0/ és a 4095 $/= 15 * 256 + 255/$ értékek között állítható be.

A nagyobb számértékekhez alacsonyabb frekvencia /hangmagasság/ tartozik!!!

- R0: A csatorna frekvencia finom: 0 - 255 között
- R1: A csatorna frekvencia durva: 0 - 15 között / 256! /
- R2: B csatorna frekvencia finom: 0 - 255 között
- R3: B csatorna frekvencia durva: 0 - 15 között / 256! /
- R4: C csatorna frekvencia finom: 0 - 255 között
- R5: C csatorna frekvencia durva: 0 - 15 között / 256! /
- R6: Zajgenerátor frekvencia : 0 - 31 között

A zajgenerátor hangmagasságát az R6 regiszter - 0 - 31 között beállítható - tartalma határozza meg.

Zenei hangok előállításához vegyük figyelembe az alábbiakat:

Egy adott /pl. a 100-as értéknek megfelelő/ hangmagasságnál egy oktávval magasabb hangot az érték felénél $/100/2=50\text{-nél}/$, egy oktávval alacsonyabb hangot az érték kétszeresénél $/100 * 2 = 200\text{-nál}/$ kapunk. Az oktáv 12 félhangos bontásából adódóan egy félhang emelkedés a frekvencia $12\sqrt{2}$ -szeres változását jelenti $/f * 1,05946/$.

Például, ha egy hangmagasságnak megfelelő /nem frekvencia!/ értéket D0, az ennél egy félhanggal magasabb hangnak megfelelő értéket D1 jelöli, akkor:

$DI = DO/2 \lceil (1/12)$ kifejezéssel kapjuk meg /94.3874/.

/Magasabb hanghoz alacsonyabb számérték tartozik!/
 A DO-nál öt félhanggal magasabb FA értéke:

$$FA = DO/2 \lceil ((1/12) * 5)$$

A félhangváltozásnak megfelelő tényezőt X-szel jelölve, a DO-nál N félhanggal magasabb hangnak megfelelő értéket /H/ így kaphatjuk:

$$X = 2 \lceil (1/12)$$

$$H = DO/X \lceil N$$

Ha ez az érték 255-nél nagyobb, akkor ki kell számolni a finom, illetve a durva beállító regiszterekbe töltendő értéket. Legyen D: durva, F: finom:

$$D = \text{INT} (H/256)$$

$$F = H - 256 * D$$

Igy például az A csatorna hangmagasságát a

⋮

100 OUT 31, 0 : OUT 30, F

110 OUT 31, 1 : OUT 30, D

⋮

utasításokkal állíthatjuk be.

3.3.2. Hangcsatornák és zajgenerátor engedélyezése /R7/

A három programozható hangmagasságú csatorna, valamint a zajgenerátor kimenetre kapcsolását az R7 regiszter vezérli.

A regiszter bitjeinek 0 /nulla/ állapota engedélyezi a hozzárendelt egységek kimenetre kapcsolódását.

0. bit $/2^0 = 1/$	A csatorna engedélyezése az I. kimenetre
1. bit $/2^1 = 2/$	B csatorna engedélyezése a II. kimenetre
2. bit $/2^2 = 4/$	C csatorna engedélyezése a III. kimenetre
3. bit $/2^3 = 8/$	zajgenerátor engedélyezése az I. kimenetre
4. bit $/2^4 = 16/$	zajgenerátor engedélyezése a II. kimenetre
5. bit $/2^5 = 32/$	zajgenerátor engedélyezése a III. kimenetre
6. bit $/2^6 = 64/$	/R14 vezérlése/
7. bit $/2^7 = 128/$	/R15 vezérlése/

A 6., és 7. bitek a perifériavezérlő pontok /R14, R15/ vezérlését végzik /1: kimenet, 0: bemenet/, kezelésüket a "HT - 1080Z programozási segédlet haladóknak" kötet tárgyalja. Értékük a hanggenerátor működését nem befolyásolja. /Célszerűen 1-nek választhatók./

Ha a regiszter minden bitjén /minden 2-es számrendszerbeli helyiértékén/ 1 van, akkor a hanggenerátor kimenetre egyetlen egység sem kapcsolódik. A regiszter értéke ebben az esetben 255.

Hogy valamelyik bitbe /helyiértékre/ 0 számjegyet írjunk be, ahhoz a 255-nél a megfelelő helyiértékkel kisebb számot kell az R7 regiszterbe beírni.

Például:

Az A csatorna kimenetre kapcsolásához:	255-1=254-et,
az A és C csatorna engedélyezéséhez :	255-1-4=250-et,
a zajgenerátor kimenetre kapcsolásához:	255-8=247-et,
vagy :	255-16=239-et,
vagy :	255-32=223-at

kell az R7 regiszterbe írunk!

Az A, B, C csatornák és a zajgenerátor egyidejűleg is engedélyezhetők!

3.3.3. Hangerősség beállítása

A három kimeneti csatorna hangerősségét az R8, R9, R10 regiszterekbe írt /0 - 15 közötti/ értékekkel lehet beállítani. A kisebb számokhoz kisebb, a nagyobbakhoz nagyobb hangerősség tartozik.

R8 : I. kimenet : 0 - 15 között
 R9 : II.kimenet : 0 - 15 között
 R10 : III.kimenet : 0 - 15 között

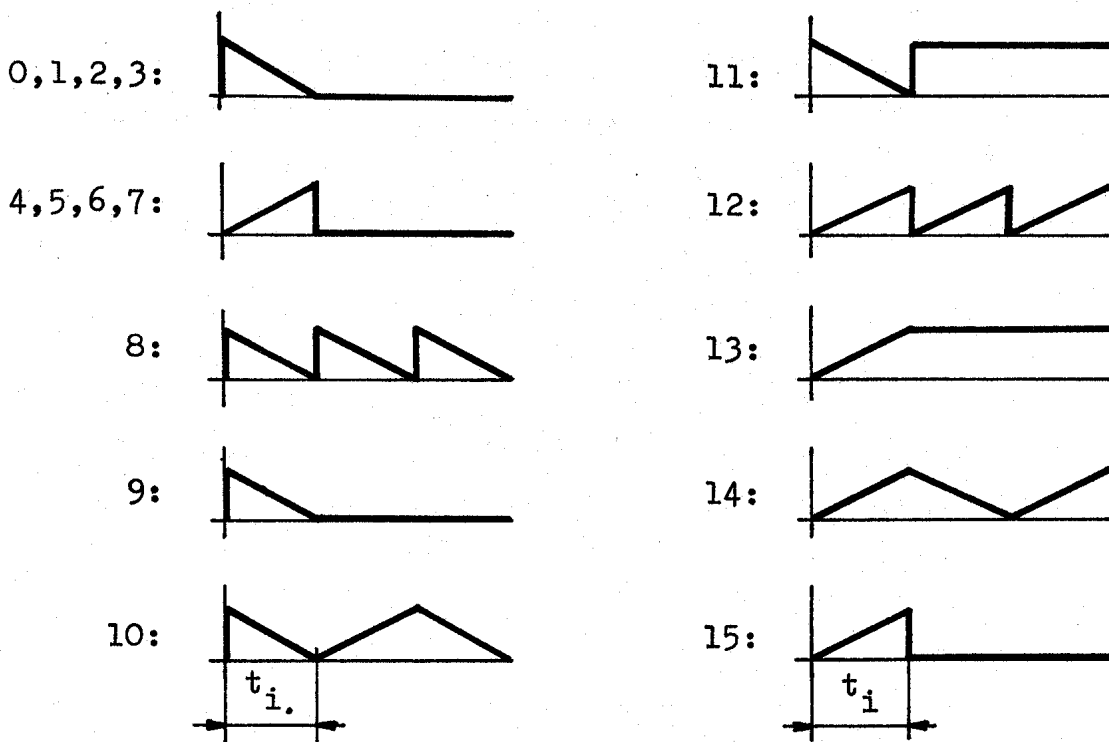
Bármelyik regiszterbe 16-ot írva az adott kimenet hangerős-
 ségét a programozható burkológörbe generátor fogja vezérelni.

A burkológörbe ismétlődési idejét az R11, R12, a burkológör-
 be formáját az R13 regiszter tartalma határozza meg.

R11 : burkológörbe ismétlődési idő finom : 0 - 255 között

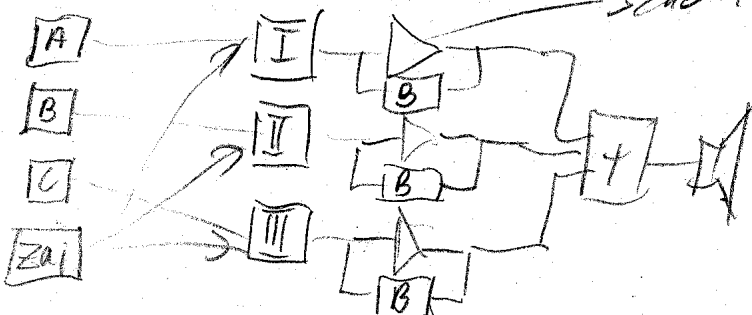
R12 : burkológörbe ismétlődési idő durva : 0 - 15 között
 /* 256!/*

R13 : /10 féle görbét választhatunk/ : 0 - 15 között



t_i : ismétlődési idő /R11, R12/

A burkológörbe generátorral sokféle hangeffektust /szélzú-
 gás, lövés, tengerzúgás, autó, stb./ állíthatunk elő, de
 használhatjuk zenei programoknál is.



Burkoló görbe generátor

Példaprogram:

```

10 DATA 0,0,0,0,0,0,0,0,255,15,15,15,0,0,0
20 FOR I = 0 TO 13 : READ A : OUT 31, I : OUT 30, A : NEXT I
30 X = 25(1/12)
40 FOR I = 1 TO 72
50 READ U : READ V : HU = 255/X U : HV = 255/X V
60 OUT 31,7 : OUT 30, 255 - 2 * SGN(V) - SGN(U)
70 GOSUB 100
80 NEXT I
90 RESTORE : GOTO 40
100 OUT 31, 0 : OUT 30, HU : OUT 31, 2 : OUT 30, HV
110 FOR T = 1 TO 20 : NEXT T
120 RETURN
130 DATA 6,6,6,6,6,9,6,9,9,13,9,13,9,18,9,18,13,17,13,18
140 DATA 13,20,13,21,14,23,14,21,14,20,14,18,5,20,5,13,5,20,
    5,23
150 DATA 0,21,0,20,13,18,13,17,13,15,13,17,12,18,12,21,12,20,
    12,18,
160 DATA 11,17,11,15,11,13,11,14,10,16,10,19,10,18,10,16,9,
    14,9,13
170 DATA 9,11,9,13,8,14,8,16,7,14,7,13,6,11,6,9,5,8,5,9
180 DATA 1,11,1,13,6,11,6,9,11,18,11,14,9,13,9,11,9,9,9,18
190 DATA 8,17,8,15,8,13,8,11,6,9,6,8,6,9,6,13,9,18,9,13
200 DATA 9,11,9,13,0,0

```

3.4. Hibaüzenetek

A végre nem hajtható parancsok, utasítások jelzésére a képernyőn hibaüzenetek íródnak ki.

A program utasításainak helyessége csak a program futtatásakor kerül ellenőrzésre!

A hibaüzeneteket az F 4. tartalmazza.

A hibakezeléssel kapcsolatos funkciókat a II.kötet ismerteti.

F 1.ASCII karakter kódok 32 - 128

KÓD	KARAKTER	KÓD	KARAKTER
32	space	76	L
33	!	77	M
34	"	78	N
35	#	79	O
36	\$	80	P
37	%	81	Q
38	&	82	R
39	'	83	S
40	(	84	T
41	)	85	U
42	*	86	V
43	+	87	W
44	,	88	X
45	-	89	Y
46	.	90	Z
47	/	91	↑
48	0	92	↓
49	1	93	←
50	2	94	→
51	3	95	-
52	4	96-127	Kisbetűk
53	5		Space
54	6	128	
55	7		
56	8		
57	9		
58	:		
59	;		
60	<		
61	=		
62	>		
63	?		
64	@		
65	A		
66	B		
67	C		
68	D		
69	E		
70	F		
71	G		
72	H		
73	I		
74	J		
75	K		

Grafikus kódok 129 -191

Lásd F 2.-nél!

Tabulátor kódok 192 -255

0 - 63 szóköz adható meg

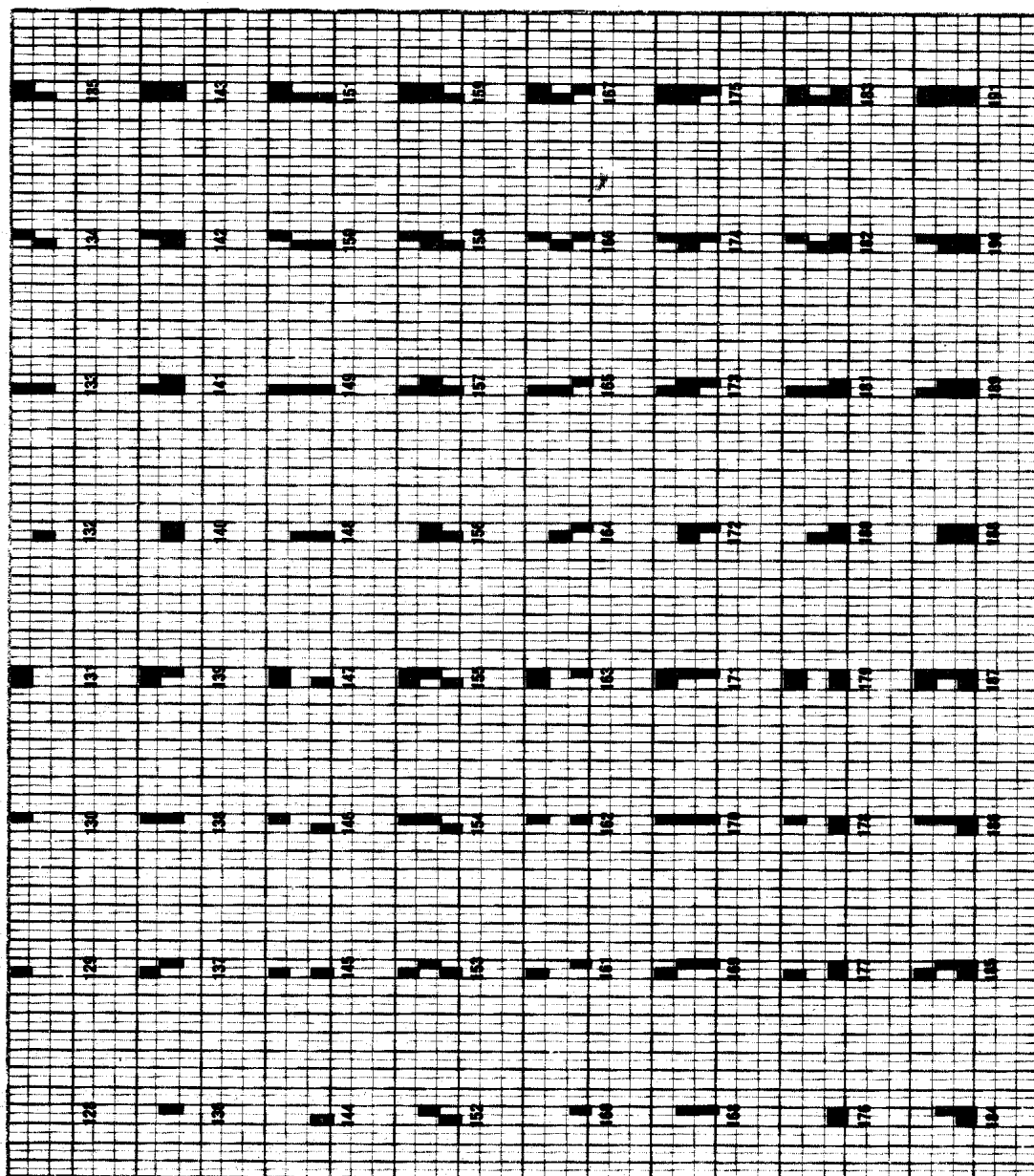
Például a

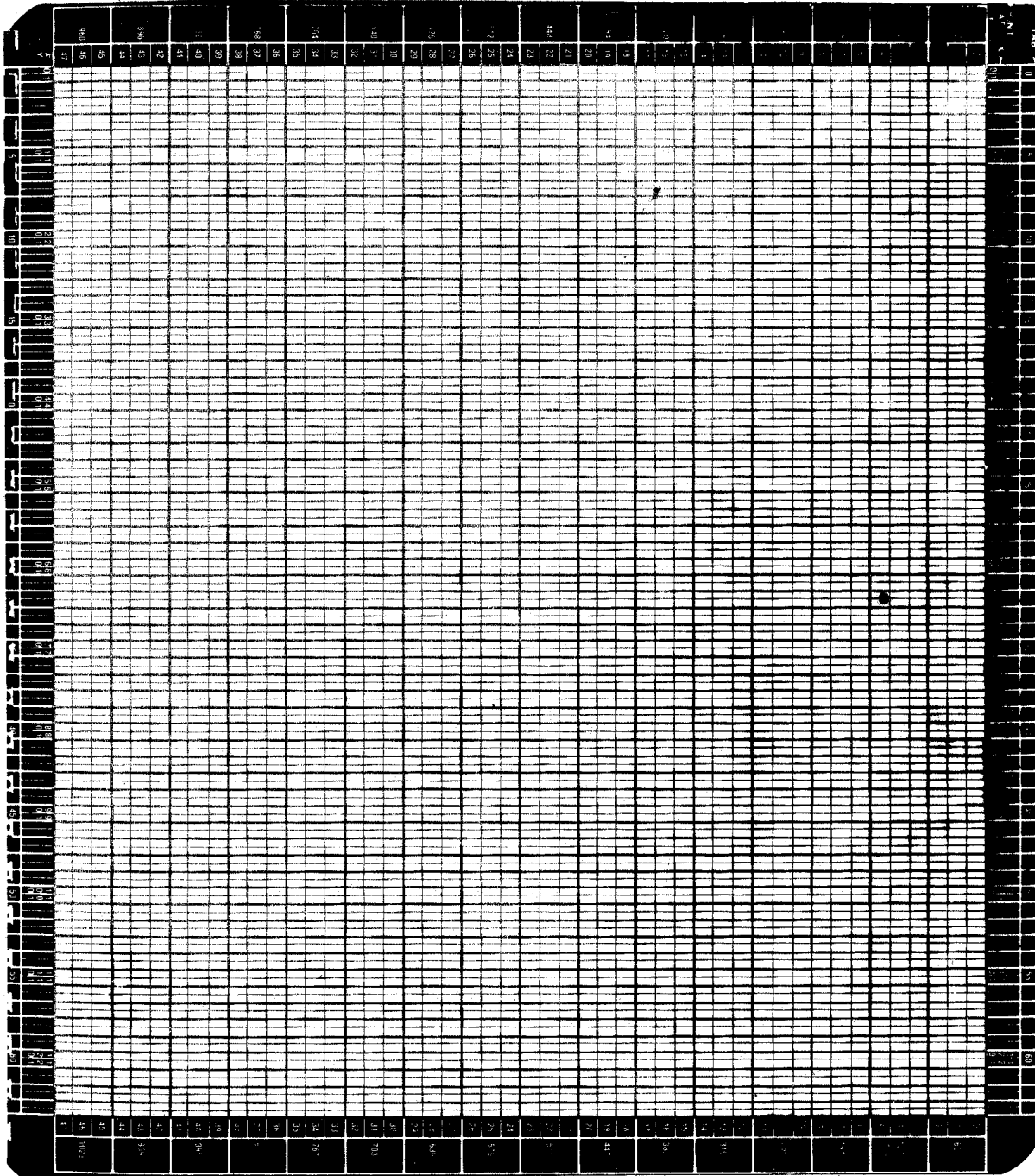
100 PRINT "*"; CHR\$(200/; "*"'

utasítás a két csillag között 200-192=8 szóközt eredményez.

Vezérlő kódok 0 - 31

Funkciójukat a II. kötet ismerteti.

F 2.ASCII grafikus kódok 129 - 191

F 3.A képernyő karakteres és grafikus blokkjai

F 4.HIBAÜZENETEK

BS	A dimenzionáltnál nagyobb index
CN	CONT utasítást nem lehet végrehajtani
DD	Ujradimenzionálás
FC	Nem megengedett függvény
FD	Hibás file
ID	Parancsként nem használható utasítás
LS	Túl hosszú fűzér
MO	Hiányzó operandus
NF	NEXT használata FOR nélkül
NR	Program vége RESUME nélkül /hibakezelésnél/
OD	Kevés adat /READ, vagy INPUT -nál/
OM	Nincs több hely a tárban
OS	Karakterfüzerek helye betelt
OV	Túl nagy szám /túlcsordulás/
RG	• RETURN utasítás GOSUB nélkül
RW	RESUME használata ON ERROR GOTO nélkül
SN	Szintaktikai /formai/ hiba
ST	Túl összetett karakteres műveletsor
TM	Numerikus és karakteres adatok keveredése
UE	Egyéb - nem kiírható - hiba
UL	Nem létező sorszámra hivatkozás
/O	Nullával való osztás