



PTA 4000 SZÁMÍTÓGÉP KA 160 NYOMTATÓ KEZELÉSI UTASÍTÁS és BASIC PROGRAMOZÁS

**WWW.
PC-1500
.INFO**

HÍRADÁSTECHNIKA SZÖVETKEZET

Licensed by SHARP





PTA 4000 SZÁMÍTÓGÉP KA 160 NYOMTATÓ

**KEZELÉSI UTASÍTÁS
és
BASIC PROGRAMOZÁS**

HÍRADÁSTECHNIKA SZÖVETKEZET

HÍRADÁSTECHNIKA SZÖVETKEZET
H-1519 BUDAPEST PF. 268
TELEX: 226151 htszh

Készítette: Kovács Miklós
Dr. Sarkantyús István
Lektorálta: Kállay László

ISBN: 963 592 372 4

Készült: Az LSI Alkalmazástechnika Tanácsadó Szolgálat
1428 Budapest PF. 12. Gondozásában

Tartalomjegyzék

1.	<u>Üzemeltetési útmutató</u>	7
1.1.	A gép üzembehelyezése és kikapcsolása	7
1.2.	A számítógép áramellátása	9
1.3.	A számítógép billentyűzete	10
1.4.	A billentyűk szerepe	11
1.5.	A kijelző	16
1.6.	Az üzemmód megválasztása	17
1.7.	A PTA 4000 használata számológépként	19
1.8.	Beszúrás, törlés	21
1.9.	A közbenső eredmények tárolása	22
1.10.	A PTA 4000 beépített függvényei	24
1.11.	Zárójel használati szabályok	24
1.12.	Hexadecimális számok	25
1.13.	Szöveges string kifejezések	26
1.14.	Relációk (összehasonlító műveletek)	27
1.15.	Szögmértékegység beállítása	29
1.16.	Konverzió	29
2.	<u>A gép programozása</u>	30
3.	<u>A BASIC nyelv elemei</u>	34
3.1.	Numerikus állandók	34
3.2.	Karakteres állandók	34
3.3.	Numerikus változók	34
3.4.	Karakteres változók	35
3.5.	Numerikus kifejezés	35
3.6.	Karakteres kifejezés	36
3.7.	Logikai kifejezés	36
3.8.	A standard változók indirekt címzése	37
3.9.	BASIC parancsok, utasítások	38
3.9.1.	BASIC parancsok	38
3.9.2.	BASIC utasítások	40

4.	<u>Beépített BASIC függvények</u>	73
4.1.	Aritmetikai függvények	73
4.2.	Trigonometrikus függvények	79
4.3.	Karakter kezelő, string függvények	80
4.4.	Logikai függvények	85
4.5.	Egyéb függvények	88
5.	<u>A KA 160 bővítő egység</u>	92
5.1.	A bővítő csatlakoztatása a számítógéphez	92
5.2.	Tápfeszültség ellátás	94
5.3.	A KA 160 csatlakoztatás magnetofonhoz	94
5.4.	A nyomtató használata	96
5.4.1.	A papírtekercs cseréje	97
5.4.2.	A színes tollak cseréje	99
5.4.3.	A kézi számítások eredményeinek kinyomtatása	102
5.4.4.	A nyomtatót vezérlő parancsok, utasítások	103
5.5.	A magnetofon illesztő használata	116
5.5.1.	A kazettás magnetofon csatlakoztatása a KA 160-hoz	117
5.5.2.	Magnetofon vezérlő utasítások	122
6.	Bővítési lehetőségek	130
7.	A PTA-4000 műszaki adatai	131
	<u>Függelék</u>	
	ASCII karakterkód táblázat a PTA-4000 géphez	135
	Megengedett rövidítések	136
	Programindítás módszerek összehasonlítása	139
	A PTA-4000 hibaüzenetei	141
	PTA-4000 memóriatérkép	150
	KA 160 csatlakozó bekötés	151
	Tartozékok	153
	PTA-4000 számítógép billentyűzet	155
	PTA-4000 számítógép fedőlemez	156

1. ÜZEMELTETÉSI ÚTMUTATÓ

1.1. A gép üzembehelyezése és kikapcsolása

A számítógépet a billentyűzet jobb felső részén találhatók **[ON]** billentyűvel lehet bekapcsolni.

Bekapcsolás után a kijelzőn a következőket láthatjuk:

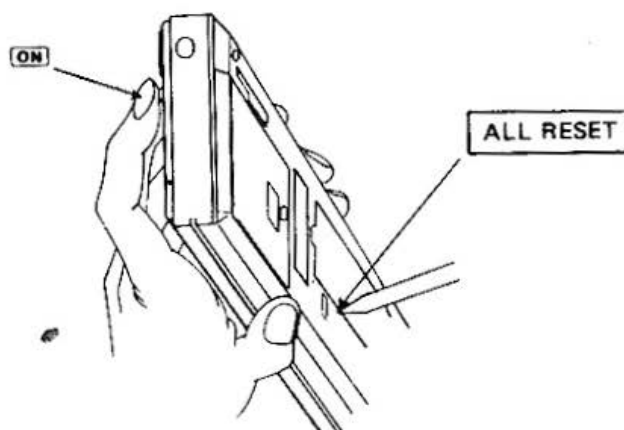
1. "☐" szimbólum a bal szélén, ami azt jelzi, hogy a gép működőképes,
2. DEG, RAD vagy GRAD, mely azt mutatja, hogy a számítógép milyen szög egységben dolgozik (fokban, radiánban, vagy új fokban, az új fok a teljes kör 1/400-ad része)
3. I, II vagy III, mely a RESERVE üzemmód szintjét jelzi,
4. RUN, PRO vagy RESERVE, ami az aktuális üzemmódot mutatja,
5. "." (egy pont), mely a gép áramellátását jelzi, ha a pont látszik, akkor az áramellátás megfelelő.

Elektrostatikus feltöltődés, vagy hibás kezelés esetén (pl. ha elemcsere előtt a gépet nem kapcsoltuk ki, stb.) a gép olyan állapotba kerülhet, hogy valamennyi billentyűje, beleértve az **[ON]** billentyűt is, hatástalanná válik. Ekkor a gép hátoldalán lévő **[ALL RESET]** feliratú gombot kell benyomni kb. 15 másodpercig, miközben az **[ON]** billentyűt is lenyomva tartjuk.

Ezután a kijelzőn

NEW : CHECK

felirat jelenik meg. Ha ez nem következne be, ismételjük meg a fenti műveletet.



1.1. ábra

Ha a felirat megjelenik, a gépet

CL **NEW** és **ENTER**

beadásával alapállapotba állíthatjuk.

Figyelem!

1. Az **ALL RESET** gomb lenyomásához valamilyen hegyes eszközt használjunk (pl. golyóstoll), de kerüljük olyan eszközök használatát, amely eltörhet (pl. grafitceruza) és beszennyezheti a készüléket!
2. Az **ALL RESET** és az **ON** együttes használatakor a gépben tárolt adatok, programok törlődnek, így csak akkor használjuk, ha feltétlenül szükséges!

A számítógép kétféleképpen kapcsolható ki:

1. Manuálisan, ha az **OFF** billentyűt lenyomjuk, ekkor az adatok, programok a gépben maradnak, de a kijelző tartalma törlődik.
A gép manuálisan nem kapcsolható ki, ha a kijelző bal felső sarkában a BUSY felirat látható!

2. A gép kikapcsolja önmagát, ha kb. 7 percig nem történik bebillentyűzés és nincs programvégrehajtás. Ekkor sem az adatok, programok, sem pedig a kijelző tár tartalma nem törlődik.

.2. A számítógép áramellátása

A gép áramellátására alkáli-mangán telepek vagy hálózati adapter használhatók.

Ha a gépet hálózati adatterről kívánjuk használni, akkor is tegyünk megfelelő állapotú elemet az elemtartóba. A telepek állapotát a kijelző jobb felső sarkában lévő pont jelzi. Ha nincs pont, az elemeket cserélni kell.

Ha hálózati adaptert használunk, a gép ADAPTOR-ral jelzett hüvelyéhez csatlakozzunk.

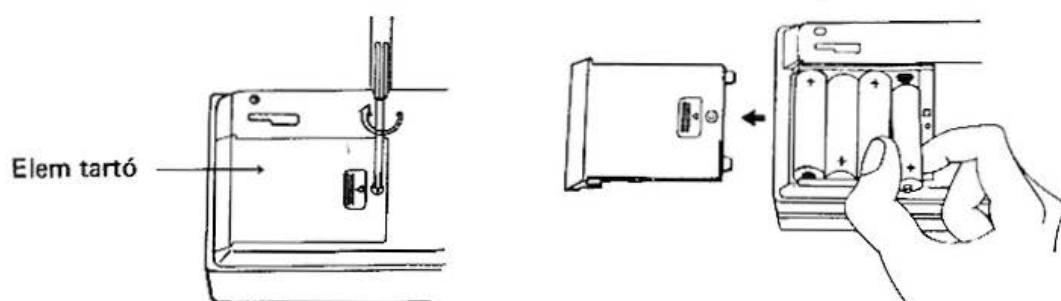
Az elemeket a következőképpen cseréljük: (1.2. ábra)

1. Kapcsoljuk ki a gépet.
2. Lazítsuk meg az elemház fedél rögzítő csavarját.
3. Vegyük le az elemház fedelét.
4. Cseréljük ki az elemeket (a helyes polaritásra ügyeljünk).
5. Tegyük vissza az elemház fedelet és rögzítsük a csavarral.
6. Kapcsoljuk be a gépet. Ha a kijelző üres, ellenőrizzük, helyesen raktuk-e be az elemeket.
7. **CL** NEW és **ENTER** beadásával állítsuk alapállapotba a gépet.

Figyelem!

1. Egyszerre mind a négy elemet cseréljük ki!
2. Ha a számítógépet hosszabb ideig nem használjuk, az elemeket vegyük ki!

3. A kimerült elemeket azonnal távolítsuk el!
4. Ha a gép a KA 160 bővítővel van összekötve, a hálózati adaptert a bővítőhöz kell csatlakoztatni



1.2. ábra

1.3. A számítógép billentyűzete

A gépnek összesen 65 billentyűje van.

Ezek között vannak alfabetikus (A-tól Z-ig), numerikus (0-tól 9-ig) és speciális billentyűk.

(Pl. **SPACE**, **ENTER**, **+**, **-**, **DEF** stb.)

Vannak olyan billentyűk is, amelyeknek két funkciójuk van, mint pl. a **[I]** -nek a **[<]** vagy **[*]** -nak a **[:]**. Ezek a funkciók a billentyűk fölött sárga színben vannak feltüntetve. A második funkciót a **SHIFT** billentyű segítségével hozzuk létre úgy, hogy először lenyomjuk a **SHIFT** billentyűt (ekkor a kijelző felső részén **SHIFT** kijelzés jelenik meg), majd lenyomjuk azt a billentyűt, melynek a második funkcióját akarjuk használni (pl. a ";" létrehozásához először lenyomjuk a **SHIFT** -et, majd a **+** billentyűt.) A **SHIFT** billentyű hatása csak a következő billentyű lenyomásáig érvényes!

[A] **[Q]** **[W]** **[E]** **[R]** **[T]** **[Y]** **[U]** **[I]** **[O]** és **[P]** billentyűk a gyakran használt BASIC kulcsszavakkal vannak kettős szerepben (pl. a **[Q]** az INPUT -tal, vagy az **[R]** a GOTO-val, stb.)

Ezeket a kulcsszavakat a **[DEF]** billentyűvel aktivizálhatjuk. (Pl. a **[DEF]** és **[Q]** -ra INPUT, stb.)

A **[DEF]** billentyű hatása - hasonlóan a **[SHIFT]** billentyűhöz - csak az öt követő billentyű (fent jelzettel) benyomásáig hatásos.

Ha a **[SHIFT]** billentyűt lenyomjuk egy alfabetikus billentyű előtt, akkor az adott betű kisbetűjét kapjuk meg (pl. **[SHIFT]** és **[A]** lenyomására a kijelzőn "a" látható stb.)

Az **[SML]** billentyű egyszeri lenyomása után, ha a továbbiakban betűket nyomunk le, akkor az annak megfelelő kisbetűt kaphatjuk meg. Az **[SML]** billentyű hatása az **[SML]** következő lenyomásáig tart.

4. A billentyűk szerepe

[CL] - törli a kijelző tároló és a kijelzőn megjelenő információt.

CA - **[SHIFT]** **[CL]** billentyűk lenyomása a kijelző törlésén túl alaphelyzetbe állítja a gépet.

[MODE] - üzemmódváltó billentyű. Lenyomására a kijelzőn az aktuális üzemmód (RUN vagy PRO) látható.

A RESERVE üzemmód létrehozásához **[SHIFT]** és **[MODE]** -ot kell lenyomni.

◀ - szintén két funkciója van:

1. a cursort balra mozgatja. ◀

2. DEL törlés, az adott sorból karaktert (vagy karaktereket) törölhetünk vele, **[SHIFT]** ◀ lenyomásával hozható létre.

Lásd az 1.8. Beszúrás, törlés c. fejezetet!

- ▶ - két funkciója van:
 1. a cursort jobbra mozgatja ▶
 2. INS Beszúrás, az adott sorba karaktert (vagy karaktereket) írhatunk be.
 SHIFT ▶ lenyomásával hozható létre.
 Lásd az 1.8. Beszúrás, törlés című fejezetet!
- * / - aritmetikai műveleti jelek (a szorzás jele a csillag: * ; az osztásé a per jel: / (és nem a kettőspont); az összeadásé a plusz jel + ; a kivonásé a mínusz jel - . Mind a négy billentyűnek van második funkciója is, rendre: "?", ":", ",", ";".
- - tizedespont.. A számítógépen nem tizedesvessző, hanem tizedespont van (pl. nem 1,3 hanem 1.3, stb.)
- 0....9 - numerikus jelek (számok).
- E - ez a betű a számok normál alakban való ábrázolására szolgál (pl. $179=1.79 \times 10^2$, ami 1.79E 2 alakban is írható).
- = - egyrészt az értékadás jele (pl.: A=2 értsd: A legyen egyenlő 2-vel), másrészt logikai műveleti jel (pl.: A=14, értsd: A egyenlő-e 14-gyel?).
- a - "at" jel, használatát lásd 1.6. Az üzemmód megválasztása és a 3.8. A standard változók indirekt címzése című fejezetben.
- A....Z - az angol abc 26 nagybetűje. A SHIFT vagy az SML billentyűvel hozhatók létre a megfelelő kisbetűk. A SHIFT a következő billentyű lenyomásáig, az SML a következő SML lenyomásáig hatásos.

- [<] - bal zárójel, második funkciója a "<" kisebb logikai műveleti jel.
- [>] - jobbzárójel, második funkciója a ">" nagyobb logikai műveleti jel.
- [DEF] - billentyű a manuális munka megkönnyítésére szolgál. Ha a [DEF] után lenyomjuk a [Q], [W], [E], [R], [T], [Y], [U], [I], [O], [P] billentyűk valamelyikét, akkor rendre a következő kulcsszavakat hívhatjuk le: (A [DEF] billentyű csak az öt követő billentyű lenyomásáig érvényes, hatásosságát a kijelző felső részén megjelenő DEF felirat jelzi) INPUT, PRINT, USING, GOTO, GOSUB, RETURN, CSAVE, CLOAD, MERGE, LIST.
A CSAVE, CLOAD és a MERGE kulcsszavak csak akkor használhatók, ha a géphez a KA 160 bővítőt csatlakoztattuk. Ha nincs bővítő és mi a [DEF] után [U], [I], vagy [O] billentyűt nyomunk le, a kijelzőn "~" jel látható, jelezve, hogy az adott kulcsszavak nem használhatók.
A fenti billentyűk használatát könnyíti meg a géphez mellékelt műanyag lemezke, melyet a billentyűzetre kell ráhelyezni. A [DEF] billentyű másik felhasználási lehetősége a programindítással kapcsolatos. Általában egy BASIC programot vagy RUN-nal, RUN sorszámmal (pl. RUN 10), vagy GOTO sorszámmal (pl. GOTO 0) indíthatunk. A PTA 4000 lehetővé teszi, hogy egy programot nem csak a fenti módon, hanem [DEF] és "cimke" lenyomásával is lehet indítani,

ahol a címke **A S D F G H J K L**
Z X C V B N M SPACE és = billen-
 tyűk valamelyike lehet.

Például legyen programunk a következő:

```
10 "M" : PRINT "A PROGRAM ELINDULT!"
20 END
```

Miután PRO üzemmódban bebillentyűztük a fenti két sort, váltsuk át a **MODE** -dal a gépet RUN üzemmódba. Nyomjuk le a **DEF** , majd a **M** billentyűt, amire programunk elindul. A programban a címkét a sorszám után idézőjelek (") közé kell elhelyezni, és kettősponttal kell elválasztani a sor többi részétől. A címkek másfajta használatát lásd majd a GOTO utasítás leírásánál, a "BASIC utasítások, parancsok" című fejezetben.

[] , **[]** ,


, **\$** ,

% , **&** , - egyrészt különleges karakterek, másrészt a RESERVE üzemmódban három szinten összesen $3 \times 6 = 18$ kulcsszót programozhatunk be. Lásd az "Üzemmód megválasztása" című fejezetben.

SHIFT - bizonyos billentyűk második szerepének és a kisbetűknek létrehozására szolgál. Használatakor a kijelző felső részén a SHIFT felirat jelenik meg. Hatása csak a következő billentyű lenyomásáig tart. A tévesen lenyomott **SHIFT** billentyűt annak újbóli lenyomásával tudjuk hatástalanítani.

OFF - kikapcsoló billentyű.

- ON** - egyrészt bekapcsoló billentyű, másrészt futó programnál BREAK billentyűként működik, mely megszakítja az illető program futását. Ekkor a kijelzőn BREAK IN sor-szám jelenik meg (pl. BREAK IN 1700). A BREAK a géphez csatlakoztatott nyomtató illetve mágnesszalagos egység működését is megszakítja.
- SML** - a kisbetűk létrehozására szolgál, hatása az **SML** következő lenyomásáig tart.
- ↕** - a RESERVE szintek lehívására szolgál. Lásd az "Üzem mód megválasztása" című fejezetben.
- RCL** - lásd az "Üzem mód megválasztása" című fejezetben.
- SPACE** - üres karakter vagy szóköz, második funkciója a hatványozás, jele: \wedge .
Pl.: $2^3 = 2 \wedge 3$ stb.
- ↓** **↑** - ha a gépben program van, ezekkel a billentyűkkel pörgethetjük végig a kijelzőn a sorokat
 ↓ - lefelé, **↑** - felfelé
 második funkciójuk:
SHIFT és **↓** -re a π -t kapjuk meg
 kilenc tizedesre (≈ 3.141592654)
SHIFT és **↑** -re egy számból négyzetgyököt vonhatunk (jele: $\sqrt{}$) pl.: $\sqrt{2}$
 SHIFT **↑** 2 **ENTER**
 de ugyanígy $\sqrt{2} = 2 \wedge (1/2)$; kijelzés: 1.44




- RUN állásban, ha a program futása valamilyen hiba következtében leáll, a  billentyű hatására a kijelzőn megjelenik a hibás sor és villogó cursor jelzi a nem értelmezhető utasítás kezdetét.



1.5. A kijelző

A kijelző egy 7x156-os LCD pontmátrix, egy karakter 7x5-ös pontmátrixból áll. (Egy pont a karakterek közötti távolság.)

A PTA 4000 kijelzőjén egyszerre 26 karakter (betű, szám, speciális karakter) látható, azonban egy sorba maximálisan 80 karakter írható be. Valójában csak 25 karakter látható, mivel a 26. a cursor.

A cursor jelzi a kijelzőn azt a helyet, ahová a legközelebbi karaktert írhatjuk. Normál módon a cursor az első szabad helyen áll és egy rövid vonalként jelenik meg.

Ha a cursort a  vagy  billentyűvel balra vagy jobbra mozgatjuk, akkor annak pozícióját villogás teszi felismerhetővé, azon a karakteren amelyiken áll. A cursor a  billentyűvel is mozgatható, de csak balról jobbra. Ekkor a cursor alá került karakter kitörlődik.

A  vagy  billentyű tartós lenyomása a cursort a megfelelő irányba folyamatosan gördíti, egyszeri lenyomás a cursort egy karakterrel mozdítja el.

A cursor jobbra mozgathatásával lehívhatjuk az egy sorban lévő karaktereket. Ha egy sorba több mint 80 jelet akarunk írni, azt a gép nem veszi figyelembe, tehát a 81. karaktert nem lehet beadni, ilyenkor a cursor a 80. pozíción villog.

1.6. Az üzemmód megválasztása

A PTA 4000 három üzemmódot tesz lehetővé:

1. RUN üzemmód




Ebben az üzemmódban a gép számológépként használható, vagy a betöltött program futtatható.

2. PRO üzemmód

Ebben az üzemmódban BASIC program írható be, illetve a bentlévő javítható, módosítható.



3. RESERVE üzemmód


Ebben az üzemmódban $3 \times 6 = 18$ gyakran használt kulcsszót programozhatunk be, a következőképpen:

SHIFT és **MODE** -dal váltsunk RESERVE üzemmódba, a  billentyűvel válasszuk ki a megfelelő szintet (I, II vagy III), amelyiken programozni akarunk. Ha lenyomjuk a , , stb. billentyűket, rendre F1:, F2:, F3 stb. jelenik meg, amikor is az illető billentyű már programozható is. Például nyomjuk le a **!**-t, ekkor a kijelzőn F1: jelenik meg. Most írjuk be, hogy RUN és **ENTER**. Ekkor a **!**-t RUN-ra programoztuk. Váltsunk vissza RUN üzemmódba. Ha most lenyomjuk a **!**-t, a kijelzőn a RUN jelenik meg (a **!"** a **SHIFT** és **!** nyomógombokkal hozható létre). Ezt követően az **ENTER**-t lenyomva a gépben tárolt program futása elindul.

A RESERVE üzemmódban beírt utasítások az adott szinten beírt NEW bevitelével törölhetők. Ne felejtsük el, hogy a beírt utasítás csak azon a szinten hatásos, amelyiken beírtuk.

Valamely beírt programot a következőképpen írhatunk felül illetve törölhetünk:

1. A **SHIFT** és **MODE** billentyűkkel RESERVE üzemmódba térünk át.
2. A kívánt (F1: F2: stb. függvény lehívásakor) a törölni illetve felülírni kívánt utasításra a  vagy  billentyűvel a cursort rávisszük. Ezután a **SHIFT** és DEL beadásával töröljük az utasítást, illetve felülírjuk azt.
3. Az **ENTER** lenyomásával az állapotot rögzítjük.
4. Visszatérünk RUN vagy PRO üzemmódba.

Itt kell megemlíteni az @ ("at") jel használatát. Ha az előző példában a RUN beadása után még beadjuk az @ -t, **SHIFT** és  jelet, a "!" billentyű RUN @ -ra lesz programozva. Ha átváltunk RUN üzemmódba, és lenyomjuk a "!" billentyűt, programunk rögtön elindul, nem kell még az **ENTER** -t is lenyomni.

Tehát ha egy kulcsszót a RESERVE üzemmódban @-tal együtt programozunk (pl. RUN @, NEW@ @, LIST a stb.), visszaváltva a megfelelő üzemmódba, benyomva a kívánt billentyűt a parancs végrehajtása azonnal bekövetkezik.

A gépen lehetőség van arra, hogy letiltssuk az üzemmódváltást. Erre szolgál a

LOCK és a letiltás feloldására az
UNLOCK parancs

A két parancsot programban utasításként is használhatjuk. A LOCK-ot érdemes használni, mert RUN üzemmódból nem tudunk PRO üzemmódba váltani, így nem tudunk és mások sem tudnak véletlenül belenyúlni a programba.

Itt kell még megemlítenünk a **[RCL]** billentyű használatát. A RESERVE üzemmódban valamelyik szinten (legyen ez pl. a III. szint) programozzuk be a megfelelő billentyűkkel a következő kulcsszavakat:

F1: RUN	F4: TO
F2: CONT	F5: STEP
F3: FOR	F6: NEXT

Tegyük fel, hogy a fenti kulcsszavakhoz rendre a RUN, CNT, FOR, TO, STP, NXT rövidítéseket kívánjuk hozzárendelni emlékeztetőül. RESERVE üzemmódban billentyűzzük be az alábbi szavakat:

[SHIFT] " **[SPACE]** RUN **[SPACE]** CNT **[SPACE]** FOR
[SPACE] TO **[SPACE]** **[SPACE]** STP **[SPACE]** NXT **[SHIFT]**
 " **[ENTER]**

Tehát a kívánt rövidítéseket egymástól szóköz -
 zel elválasztva. A gép az így tárolt jelsorozatot RUN vagy PRO üzemmódban megjeleníti az RCL billentyű lenyomására, pontosan a megfelelő billentyű felett. Így a kívánt billentyű jelentése munka közben bármikor lehívható.

Az **[RCL]** hatására a kijelző korábbi tartalma nem vész el, az **[RCL]** ismételt megnyomására újra előhívható.

.7. A PTA 4000 használata számológépként

A PTA 4000 a felhasználó számára nagyszámú alkalmazási lehetőséget kínál. A gép normál számológépként is használható. Ehhez az összes használatos matematikai függvény rendelkezésre áll. A géphez csatlakoztatott nyomtatóval az összes betáplált

adatot és eredményt ki lehet iratni (ha a PRINT kapcsoló a nyomtatón P helyzetben áll).

A gép a tároló kialakításától függően kb. 700 szám tárolására alkalmas, figyelembe véve a közbenső eredmények tárolását is. Ezen kívül 25-ös mélységig használhatunk egymásba zárt zárójeleket.

A számoláshoz a gépet a **MODE** billentyűvel RUN üzemmódba állítjuk.

A számolás általános alakja:

CL numerikus kifejezés **ENTER**

Numerikus kifejezésnek nevezünk minden olyan matematikai formulát, amelynek eredménye számérték.

Példák:

<u>Betáplálás</u>	<u>Kijelzés</u>
CL 2 + 4 ENTER	6
CL 2 - 4 ENTER	-2
CL 2 * 4 ENTER	8
CL 2/4 ENTER	0.5
CL 4+2 ^4 ENTER	20

Megjegyzések:

1. A hatványozás jele a " \wedge ", **SHIFT** és **SPACE**-val hozható létre.
2. Az osztás jele a $/$, a szorzásé a $*$
3. A nullát áthúzva jelöljük: \emptyset , és nem 0 betűt kell beadni, hanem a \emptyset számot.
4. A törteknél nem tizedesvessző, hanem tizedes-pont van. Tehát nem 1,3, hanem 1.3.
5. Lehetőség van számok ábrázolására normál alakban.

Példák:

<u>Betáplálás</u>	<u>Kijelzés</u>
<u>CL</u> 5E - 5 <u>ENTER</u>	0.00005
<u>CL</u> 5E + 11 <u>ENTER</u>	5E 11

6. A normál alakban megadható számok alsó határa:
- 9.999999999E+99
felső határa: + 9.999999999E+99
7. A gép az eredményeket normál alakban közli, ha azt több mint 10 számjeggyel lehetne leírni.
8. A PTA 4000 a számokat 10 számjegyes pontossággal dolgozza fel, ez hosszabb számoknál kerekítési hibákat okozhat.
9. Ha hatványozásnál az alap negatív, akkor azt zárójelbe kell tenni (pl. $(-2)^3$.)
10. Egy már kiszámított eredmény újra felhasználható további számításhoz. Ekkor nem a CL -t kell lenyomni, hanem a következő műveleti jellel kell folytatni a számolást.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
<u>CL</u> 3 * 7 <u>ENTER</u>	21
/10 <u>ENTER</u>	2.1

1.8. Beszúrás, törlés

A gép lehetővé teszi, hogy a gépelési hibákat a betáplálás közben javítsuk, a már beadott formulákat ellenőrizzük és módosítsuk. Erre szolgál a

◀ és a ▶ billentyű. Ezekkel a cursort balra ill. jobbra tudjuk mozgatni, anélkül, hogy a betáplált információkat elvesztenénk.

Ha egy karaktert törölni akarunk, akkor a cursort a megfelelő irányba elmozgatva a törlendő karakter fölé visszük és [SHIFT] és ◀ lenyomásával töröljük.

A beszúrára nézzünk egy példát:

Legyen adott a "SZAMOGEP" szövegünk. Szúrjuk be az "M" és az "O" betűk közé az "I" és a "T" betűket. Mozgassuk el a cursort az "O" betű fölé, a ◀ vagy ▶ billentyűk valamelyikével. Ezután kétszer nyomjuk le a [SHIFT] és ▶ -t. Ekkor a beszúrandó helyeken □ □ látható példánkban:

SZAM□□OGEP

Erre a két helyre csak be kell billentyűzni a kívánt két karaktert ("I"-t és "T"-t). Így megkaptuk a kívánt SZAMITOGEP szöveget.

1.9. A közbelső eredmények tárolása

A változók a változóneveken keresztül érhetők el. Egy ilyen változónév egy vagy több alfanumerikus karaktersorozatból állhat, az első karakternek kötelezően betűnek kell lennie. (Pl. A A1, BETU, VA3, de nem helyes a szám és betű ill. szám és szám vagy betű karakter pl. 3A, A? stb.) A változónevek csak nagy betűket tartalmazhatnak (pl. Abl helytelen). A változónevek nem tartalmazhatnak BASIC kulcsszavakat vagy olyan neveket, amelyek intarziaként tartalmaznak BASIC utasítást. Pl. TO; AUTO22; helytelen, a TO BASIC utasítás. Pl. VALTOZO? SZAMITO", stb. sem lehet).

Ha a változó neve több karakter hosszú (pl. SUMMA), akkor a gép csak az első két karaktert veszi figyelembe, tehát pl. a SUM és SUMMA ugyanaz a változó lesz.

Fontos!

Az LF, IF, LN, PI, TO nevek tilosak, nehogy a gép összekeverje a BASIC kulcsszavakkal!

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
[CL] A=50 [ENTER]	50
[CL] A [ENTER]	50
[CL] ERT=1.2*2 [ENTER]	2.4
[CL] ERT+5 [ENTER]	7.4

Egy numerikus kifejezésben minden szám helyettesíthető egy változónévvel.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
A=3 [ENTER]	3
B=7 [ENTER]	7
C=A+B [ENTER]	10

Megjegyzések:

1. A változók értéke a gép ki- és bekapcsolása után is tárolva marad.
2. A CLEAR és [ENTER] hatására az összes változó törlődik.

1.10. A PTA 4000 beépített függvényei

A gépben számos karakterkezelő- (pl. MID, CHR stb.) matematikai (pl. SIN, COS, LOG, EXP. stb.) és egyéb beépített függvény (pl. BEEP, STATUS, PEEK stb.) van. Ezeket a függvényeket használhatjuk akkor is, ha számológépként működtetjük, vagy ha BASIC nyelven programozzuk a gépet.

A függvények működésének és alkalmazásának leírása "A beépített BASIC függvények" című fejezetben található.

1.11. Zárójel használati szabályok

Egy kifejezésben a végrehajtandó műveletek sorrendjét zárójelekkel megváltoztatjuk. Ennek eredményeként bizonyos műveletek hamarabb, mások pedig később kerülnek végrehajtásra. Például a $3+4*5$ kifejezésben a végrehajtási sorrend a következő: a gép először kiszámítja a $4*5$ értékét, majd hozzáadja a 3-hoz. Az eredmény 23 lesz. Ha a fenti kifejezés $(3+4)*5$ alakú, akkor először kiszámítja a zárójelben lévő kifejezést, majd ezt megszorozza 5-tel. Az eredmény 35 lesz.

A műveletek végrehajtási sorrendje:

1. Zárójeles kifejezés kiszámítása.
2. Függvényérték számítás, (pl. $\text{SIN } 30+4$ megfelel $(\text{SIN}30)+4$ -nek.
3. Hatványozás, pl.: $3*2^4$ megfelel $3*(2^4)$ -nek.
4. Előjel (+, -)
5. Szorzás, osztás (*, /)
6. Összeadás, kivonás (+, -)

7. Összehasonlító műveletek:

Jelentés	Szokásos matematikai jelölés mód	PTA 4000 jelölés	Bebillentyű- zés
kisebb mint	<	<	SHIFT (
nagyobb mint	>	>	SHIFT)
egyenlő	=	=	=
kisebb vagy egyenlő	< ≤ =	< =	SHIFT (=
nagyobb vagy egyenlő	> ≥ =	> =	SHIFT) =
nem egyenlő	≠	< >	SHIFT (SHIFT)

8. Logikai műveletek:

AND	-	logikai ÉS
OR	-	logikai VAGY
NOT	-	logikai NEM

Azonos szinten a balról jobbra szabály érvényes!
pl.: 3x4/5 megfelel (3x4)/5-nek.

1.12. Hexadecimális számok

A 0 és 65535 közötti természetes számok a PTA 4000-be hexadecimális (16-os számrendszerbeli) számként is betáplálhatók. Hexadecimális szám ábrázolására a 0 és 9 közötti számokat, valamint A és F közötti betűket (melyek számokat jelentenek) használjuk. A hexadecimális számokat az & ("et") jel jelzi.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
&A <u>ENTER</u>	10
&FF <u>ENTER</u>	255
&10 <u>ENTER</u>	16

Decimális szám átalakítása hexadecimális számmá közvetlenül nem lehetséges, ezt a következő program hajtja végre:

```

10 INPUT "DECIMALIS SZAM ? " ; D
20 H$="0123456789ABCDEF" : F$=""
30 I=INT (D/16)
40 IF I<>0 THEN LET M=D-I*16:
   F$=MID$ (H$, M+1, 1) + F$:D=I:GOTO 30
50 F$=MID$ (H$, D+1, 1) + F$
55 WAIT 150
60 PRINT "HEXA SZAM : " ; F$ : GOTO 10

```

1.13. Szöveges string kifejezések

A szöveges kifejezések a BASIC nyelv alkotórészei. Mint a numerikus kifejezéseknél, itt is vannak szövegállandók, szövegváltozók, szövegfüggvények és összetett szövegek. Egy szövegállandó tetszés szerinti karaktersorozat, melyet az elején és a végén egy-egy idézőjel határol ("). A szöveges kifejezés nem tartalmazhat idézőjelet. Ezek a határolójelek nem elemei a szövegnek.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
"PTA 4000" <u>ENTER</u>	PTA 4000

A szövegváltozók tulajdonképpen tárolóhelyek a gép memóriájában a szövegek számára. A szövegváltozók neveire ugyanaz vonatkozik, mint a numerikus változók neveire, azzal a kiegészítéssel, hogy a változónevek végén kötelezően még egy dollárjelnak ("§") kell állnia (pl.: A§, SZOVEG§ stb.).

Példa:

Betáplálás

Kijelzés

A§= "SZAMOL" ENTER

SZAMOL

B§= "PROGRAM" ENTER

PROGRAM

A szövegek között értelmezve van az összeadás művelete. A művelet jele az összeadásjel ("+").

Példa:

Betáplálás

Kijelzés

A§= "SZAMOLO" ENTER

SZAMOLO

A§+ "GEP" ENTER

SZAMOLOGEP

1.14. Relációk (összehasonlító műveletek)

A számítógép összesen 6 összehasonlító műveletet ismer. Ezek a következők:

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

A fenti műveletekkel két numerikus, vagy két karakteres kifejezés hasonlítható össze egymással. Ha a

reláció igaz, akkor a gép 1-et szolgáltat (igaz, logikai értéke 1), ha reláció hamis, akkor a gép 0-t szolgáltat (hamis, logikai értéke 0).

A relációk alkalmazása a BASIC utasítások, parancsok című fejezetben az IF utasítás leírásánál található.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
CL 2 = 1 ENTER	0
CL 1 < 2 ENTER	1
CL "JANI" < "JOZSI" ENTER	1

Szöveges változóknál az idézőjel fontos!

Megjegyzések:

1. Numerikus kifejezéseket értelemszerűen hasonlít össze a gép.
2. A karakteres kifejezéseket a gép karakterenként hasonlítja össze egészen addig, amíg az első eltérést meg nem találja. Az összehasonlítás alapja a karakterek ASCII kódja (lásd Függelékben). A megtalált eltérés logikai értékének megfelelően a reláció igaz (logikai 1) vagy hamis (logikai 0) lesz. Példánkban az első betűk megegyeznek (így ASCII kódjuk is), de a második betűk már eltérnek egymástól (az egyik "A", a másik "O"). Mivel az "A" kódja kisebb, mint az "O" kódja, ezért a reláció igaz lesz.
3. Mivel egy logikai kifejezés értéke egy szám, ezért az numerikus változóban tárolható.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
[CL] L="A" < "B" [ENTER]	1
[CL] L [ENTER]	1

1.15. Szög mértékegység beállítása

A trigonometrikus függvényeknél a szögek fokban, radiánban vagy új fokban értelmezhetők. (A teljes kör 1/400-ad része egy új fok.) A fokban való értelmezést a **DEGREE** **[ENTER]**, a radiánban való értelmezést a **RAD** **[ENTER]**, az új fokban való értelmezést **GRAD** **[ENTER]** bebillentyűzéssel állíthatjuk be. Az éppen aktuális beállítást a kijelző felső szélén látható DEG, RAD vagy GRAD felirat jelzi.

1.16. Konverzió

A trigonometrikus függvényeknél szükséges a fok-perc-másodpercben megadott értékek decimális, illetve a decimálisan megadott értékek fok-perc-másodperc alakra való átalakítása.

Pl. 1. 16.1932 fok, fok-perc-másodperc alakban a következőképpen nyerhető.

DMS 16.1932 **[ENTER]** bebillentyűzésre

16.113552 kijelzést kapunk, ami a következőképpen értelmezhető:

16 fok 11 perc 35.52 másodperc, azaz $16^{\circ} 11' 35,52''$.

2. $32^{\circ} 25' 13''$ érték decimális alakra a következőképpen hozható: **DEG 32.2513** **[ENTER]** bebillentyűzésre 32.42027778 kijelzést kapunk.

2. A GÉP PROGRAMOZÁSA

A PTA 4000 számítógép BASIC nyelven programozható. A BASIC program utasításokból áll, amelyeket kulcsszavak alkotnak megfelelő szintaxis (formai helyesség) szerint. Egy utasítás két részből áll:

1. utasítássorszám (a továbbiakban sorszám)
2. utasítás

A PTA 4000-nél a legnagyobb megengedett sorszám 65279. Ha a sorszám meghaladja ezt, de nem éri el a 65540-et, ERROR 1 hibaüzenet, ha meghaladja 65539-et, ERROR 3Ø hibaüzenet jelenik meg a kijelzőn. A gépet programozni, a begépelte programot javítani, módosítani PRO üzemmódban lehet.

Az elkészült program RUN üzemmódban futtatható.

Az üzemmódot a **MODE** billentyűvel váltjuk át.

A BASIC nyelvben meghatározott nyelvtani szabályok vannak.

A nyelv ábécéje az angol ábécé 26 kis- és nagybetűjét, a számjegyeket Ø-től 9-ig, valamint speciális jeleket tartalmaz. A teljes karakterkészlet a Függelékben található (ASCII, ejtsd eszki kódtáblázat a PTA 4000-hez).

Figyelem!

A programozásban különbséget teszünk nulla és az "O" betű között. A nullát áthúzva ábrázoljuk "Ø".

A gép is így ábrázolja és külön-külön billentyű van az "O"-ra és a nullára.

A BASIC programnyelvben a következő, a felhasználó által szabadon definiálható nyelvelemek vannak:

- numerikus állandók
- numerikus változók
- karakteres állandók
- karakteres változók
- dimenzionált numerikus vagy karakteres változók (tömbök)

A BASIC programnyelvben ezeken kívül u.n. kulcsszavak vannak a következők számára:

- rögzített függvények
- kijelölések
- input és output utasítások
- vezérlési utasítások
- parancsok
- műveleti jelek (+, -, *, /, ^, =, <, >)

Ezekből a nyelvelemekből utasítások vagy parancsok állíthatók össze, amelyeket a gép értelmez és végrehajt. Az utasítás és a parancs között az a különbség, hogy a parancsot a gép azonnal végrehajtja, amikor beadjuk, az utasítást pedig a program futtatása során akkor, amikor éppen sorrendben rá kerül a vezérlés.

A program sorokból áll, ezekben egy vagy több BASIC utasítás lehet. Ha egy soron belül több utasítás van, akkor azokat kettősponttal (:) kell elválasztani egymástól (pl. a $10\ L=2:K=3$ utasításban a 10 az utasítás sorszáma, $L=2$ és $K=3$ az utasítások és a közöttük lévő kettőspont az elválasztó jel).

A gép az utasításokat sorszámuk szerint növekvő sorrendben hajtja végre, hacsak vezérlő utasítással más sorrendet nem írunk elő. Az utasításokat legcélszerűbb tizesével sorszámozni, mert így le-

hetőség van később újabb utasítások beszúrására, minden különösebb nehézség nélkül.

Minden sort a sorszám begépelésével kezdjük, majd begépeljük az utasítást (vagy utasításokat, kettősponttal elválasztva), és végül az egészet az **ENTER** billentyű lenyomásával zárjuk. Az **ENTER** lenyomása után a gép egy kettőspontot tűz be a sorszám és az első utasítás közé, jelezve, hogy elfogadta azt.

Miután a programot begépeztük, PRO üzemmódban a **↑** vagy a **↓** billentyűvel előre vagy visszafelé le tudjuk hívni a program tetszőleges sorát.

Ha egy programban hibát vétettünk, akkor kétféleképpen javíthatjuk:

1. Újra gépeljük az egészet,
2. Lehívjuk a kívánt sort és az 1.8. Beszúrás, törlés című fejezetben leírtak alapján javítunk, módosítunk.

A programok végét, külön sorban az END utasítással zárjuk le. Ennek hiányában a programunk futtatásakor nem biztos, hogy a kívánt helyen áll meg a futás, s így esetleg hibás kijelzést kapunk.

Egy programot RUN üzemmódban úgy futtatunk, hogy beírjuk a gépbe a RUN parancsot és megnyomjuk az **ENTER** billentyűt. A gép elkezd a program végrehajtását a legkisebb sorszámtól, miközben a kijelző bal felső részén a BUSY felirat jelenik meg.

Ha a program futtatását nem a legkisebb sorszámtól kívánjuk, vagy a programunk nem itt kezdődik, akkor a RUN után a kívánt sorszámot írjuk be, és ezután nyomjuk le az **ENTER** billentyűt.

Ha a program sikeresen befejeződött, a kijelzőn újra az un. készenléti szimbólum (">") látható.

Ha a program valamilyen hiba miatt leállt, a kijelzőn

ERROR hibakód IN sorszám

jelenik meg. A hibakódok jelentését lásd a függékben.

Ha mi akarjuk megszakítani a program futását, akkor a **BREAK** billentyűt nyomjuk le. Ekkor a kijelzőn

BREAK IN sorszám

jelenik meg. Az így megszakított program változóit le tudjuk kérdezni. A programot **CONT** és **ENTER** beadásával indíthatjuk tovább. Ha **BREAK** után **PRO** üzemmódban megváltoztatunk valamit a programban, ezt az **ENTER** nyomógombbal érvényesítsük de ekkor a program **CONT**-tal nem folytatható. Ha mégis megpróbálnánk, **ERROR 26** hibaüzenetet kapunk. **CONT**-ot kiadni csak **RUN** üzemmódban lehet.

3. A BASIC NYELV ELEMEI

3.1. Numerikus állandók

Lehetnek egész vagy tört számok, normálalakban szereplő számok, vagy hexadecimális számok:

Például:

513
-25Ø
11.4
-5E-12
&AFØ2

3.2. Karakteres állandók (String)

A karakteres állandó (konstans) tetszőleges karaktersorozat, amelyiket idézőjelek (") közé kell bebillentyűzni. A határoló idézőjelek nem elemei a karakteres konstansnak és a karaktersorozatban nem szerepelhet idézőjel.

Például:

"KARAKTER"

" " - ez a szóköz (SPACE)
"" - ez a nulla hosszúságú karakter sorozat

3.3. Numerikus változó

Egy numerikus értéket tartalmazó memóriaterületre a változó nevével hivatkozhatunk. A változónév egy vagy több betű vagy szám lehet, de az első karakternek kötelezően betűnek kell lennie (pl. helyes a SUM, S1, V3, stb. de helytelen az 1A, V? stb.). Ha a változó neve több karakteres, a

névben nem szerepelhet BASIC kulcsszó (pl. LNV, VALTOZÓ stb., mert LN, VAL TO BASIC kulcsszavak, továbbá a gép csak az első két karaktert veszi figyelembe, ezért pl. az OSSZEG és az OSSZES ugyanaz a változó lesz.)

3.4. Karakteres változók (String változók)

A karakteres változó (alfanumerikus jelcsoport) a karaktereket tartalmazó memóriaterület neve. A karakteres változó nevére ugyanaz vonatkozik mint a numerikus változó nevére, azzal a kiegészítéssel, hogy a változónév végén kötelezően dollárjelnek (\$) kell állni.

Például:

Z\$ A5\$, FF\$ KARAKTERT\$ stb.

A numerikus állandókra illetve változókra az aritmetikai utasítások, függvények és a trigonometrikus függvények, míg a stringekre és a strigváltozókra a strigkezelő függvények értelmezettek.

3.5. Numerikus kifejezés

Egy numerikus kifejezés numerikus konstansból, numerikus változóból, numerikus függvényekből, vagy azok műveleti jelekkel (+, -, *, /,) logikai függvényekkel, valamint zárójelekkel való összekapcsolásából áll.

Például:

15

SIN (30)

A+B * (C-D) / (2* (A-D)) AND(B-A) / 3

A numerikus kifejezés eredménye mindig számérték.

3.6. Karakteres (String) kifejezés

Egy karakteres kifejezés karakteres konstansból, karakteres változóból vagy karakteres függvényből, vagy ezek összekapcsolással való összekapcsolásából áll. A karakteres kifejezés eredménye mindig karaktersorozat.

Például:

A\$ = "PTA - 4000"

B\$ = "COMPUTER"

C\$ = A\$ + B\$

3.7. Logikai kifejezés

Egy logikai kifejezés két numerikus, vagy két karakteres kifejezés összehasonlítása a következő műveletekkel:

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

vagy ilyen kifejezések összekapcsolása az AND, OR, NOT műveletek valamelyikével.

Ha az összehasonlítandó numerikus kifejezés értéke a relációnak megfelel, értéke 1, egyébként 0

Ha karakteres kifejezéseket hasonlítunk össze, a gép karakterenként, balról-jobbra végzi el. A kiértékelés alapja a karakterek ASCII kódja (lásd a Függelékben).

Ha két karaktersorozat különböző hosszúságú, akkor

a rövidebb jobbról annyi szóközzel egészül ki, hogy hossza megegyezzen a hosszabbikkal.

Példák:

<u>Betáplálás</u>	<u>Kijelzés</u>
CL 1 < 2 ENTER	1
CL 3 > 4 ENTER	Ø
CL "SAMU"="SAM" ENTER	Ø

3.8. A standard változók indirekt címzése

A PTA-4000 rendszerében az egybetűs változók tömb-változóként is hozzáférhetők. Az egybetűs változók így felfoghatók egy 26 elemű vektor elemeiként, a karakteres változók pedig egy 26 elemű karaktervektor elemeiként kezelhetők.

Ezen tömbök megadására az @ ("at") és az @ \$ ("at-dollár") jelek szolgálnak.

Igy az @ (1) változó az A változóval, az @ (3) változó a C változóval, az @ (26) változó a Z változóval azonos.

Karakteres változók esetén ugyanez a helyzet, például az @ \$ (4) a D \$ változóval azonos.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
CLEAR ENTER	
CL @ (1)=3 ENTER	3
CL A ENTER	3
CL @ \$ (14)="N" ENTER	N
CL N \$ ENTER	N

Figyelem!

Az @ és az @ \$ tömböket nem szabad deklarálni DIM utasításban!

3.9. BASIC parancsok, utasítások

Ez a fejezet betűrendben ismerteti azokat a BASIC utasításokat, parancsokat vagy beépített függvényeket, amelyek a PTA 4000 gépen használhatók. A legtöbb utasításhoz mintaprogram is van, ezek elemzése illetve kipróbálása az adott utasítás megértését segítheti elő.

3.9.1. BASIC parancsok

A most felsorolásra kerülő parancsokat programban utasításként nem használhatjuk!

CONT

Csak RUN üzemmódban használható!

szintaxisa: CONT

Már szó volt róla, hogy a program futását a BREAK lenyomásával vagy a programban elhelyezett STCP utasítással leállíthatjuk. A CONT parancs bevitelével a megszakított program futtatását folytathatjuk. Abban az esetben, ha a program valamelyik sorát megváltoztatjuk - mivel ehhez át kell térni PRO üzemmódra - a program CONT-tal nem folytatható. A CONT csak RUN üzemmódban vihető be.

LIST

Csak PRO üzemmódban használható!

szintaxisa: LIST sorszám
 vagy LIST "címke"
 vagy LIST

Ha egy programot ki akarunk listáztatni a kijelzőre (természetesen egyszerre csak egy sor jelenhet meg rajta), a LIST parancsot kell használnunk. Ha a LIST után megadunk egy sorszámot, vagy címkét (ezt idézőjelek között), akkor a kijelzőn az adott sorszámú illetve címkéjű utasítás jelenik meg.

Például ha van ilyen sorunk:

lØ "L": PRINT "PROGRAM"

akkor ez a sor megjeleníthető a kijelzőn mind a LIST lØ, mind a LIST "L" paranccsal. Ha a LIST után nem adunk meg semmit, akkor a program első sora jelenik meg a kijelzőn.

Ha nemlétező sorra hivatkoznánk, hibajelzést kapunk.

NEW

szintaxisa: NEW vagy NEWØ

Ezzel a paranccsal PRO üzemmódban a memória, RESERVE üzemmódban a programozott billentyű definíciók törölhetők. A parancsot RUN üzemmódban nem lehet be-
vinni!

RUN

szintaxisa: RUN sorszám
 vagy RUN "címke"
 vagy RUN

Ezzel a paranccsal a memóriában lévő program futását indíthatjuk el, vagy az elejétől (pl. RUN vagy RUN 10, ahol a 10 sorszámú utasítás a program első sora), vagy pedig egy tetszőleges sortól. Ekkor a kívánt utasítás sorszámát vagy címkéjét kell megadni a RUN kulcsszó után. Ha nemlétező utasítástól akarnánk elindítani a programot, hibajelzést kapunk. A RUN parancs csak RUN üzemmódban hatásos, egyébként ERROR 26 hibaüzenetet kapunk. A RUN hatását lásd a Függelékben (Programindítási módszerek összehasonlítása).

3.9.2. BASIC utasítások

AREAD

szintaxis: sorszám "cimke":AREAD változónév

Ez az utasítás lehetővé teszi, hogy a program indítása előtt a kijelzőre írt értéket a program valamely változójának adjuk át. A programot **DEF** és "cimke" típusú indítással kell indítani. Ez az utasítás csak címkével ellátott sorban, közvetlenül a cimke után szerepelhet.

Példa:

```
10 "C" : AREAD IDO
20 TIME=IDO
30 PRINT "KEZDOERTEK: ";IDO
40 END
```

Ha a program indítása előtt bebillentyűzzük azt az időt, ahonnan a gép belső óráját el akarjuk indítani, majd **DEF** és C lenyomásával elindítjuk, a kijelzőn

KEZDOERTEK : idő

jelenik meg és a gép órája erre áll be.

ARUN

szintaxisa: sorszám ARUN

Ha egy program első utasítása az ARUN, akkor a program a gép bekapcsolásakor rögtön elindul, feltéve, hogy a gépet RUN üzemmódban kapcsoltuk ki. A PRO üzemmódban történő kikapcsolás esetén bekapcsoláskor nem aktivizálódik az ARUN utasítás. Ugyanez a helyzet, ha a gép automatikusan kapcsolódik ki.

P1.:

```
1Ø   ARUN
2Ø   BEEP 5, 5Ø
3Ø   PAUSE "JO MUNKAT, EMBER!"
4Ø   CLS
5Ø   END
```

Az ARUN utasítás hatását lásd a Függelékben.
(Programindítási módszerek összehasonlítása).

BEEP

szintaxisa: sorszám BEEP n1, n2, n3,
ahol n1, n2, n3 numerikus konstans, változó vagy kifejezés lehet.

A gépben egy hanggenerátor van, a hang ismétlődésének számát, a hang magasságát és a hang időtartamát programozhatjuk a BEEP utasítással. A BEEP n1, n2, n3 szerepelhet parancsként vagy utasításként.

A paraméterek jelentése:

- n1: a hang ismétlődésének száma, értéke 0 és 65535 között lehet. Ezt a paramétert kötelező megadni!
- n2: a hang magasságát állítja be, értéke 0 (ami kb. 7000 Hz) és 255 (ami kb. 230 Hz) között lehet. Ezt a paramétert nem kötelező megadni. Alapértelmezése: n2=8.
- n3: a hangjelzés időtartamát szabhatjuk meg vele, értéke 0 és 65279 között lehet. Ezt a paramétert sem kötelező megadni, de ha n2-t elhagyjuk, akkor ezt is el kell hagyni. Alapértelmezése: n3=150.

A hanggenerátor működését a BEEP OFF paranccsal (vagy utasítással) letilthatjuk. A letiltást BEEP ON-nal oldhatjuk fel. Egy BEEP OFF hatása a következő BEEP ON beadásáig tart.

Példa:

Ez a program "elektronikus-orgona"-ként működik, minden billentyű (kivéve a BREAK-et) lenyomása más hangot ad.

```
10  A$=INKEY$
20  IF A$="" THEN GOTO 10
30  L=ASC(A$)
40  BEEP 1,L:GOTO 10
```

vagy az alábbi program:

```

10  FOR D=200 TO 0 STEP-40
20  DATA 245,1,245,1, 160,1,160,1
30  DATA 143,1, 143,1, 160,2
40  DATA 180,1,180,1,195,1,195,1
50  DATA 220,1, 220,1,245,2
60  FOR I=1 TO14
70  READ F,S
80  BEEP 1, F, (D*S)
90  NEXT I
100 RESTORE
110 NEXT D
120 END

```

CLEAR

szintaxisa: sorszám CLEAR

Ez a parancs (lehet utasítás is) a memóriában lévő összes változót törli, a standard változókat 0-ra állítja.

CLS

szintaxisa: sorszám CLS

Ez a parancs (vagy utasítás) törli a kijelző tartalmát és alapállapotba állítja a cursort.

CURSOR

szintaxisa: sorszám CURSOR n,

ahol n numerikus konstans, változó vagy kifejezés lehet.

Ez a parancs (vagy utasítás) a cursort az n által megadott pozícióba helyezi a kijelzőn. Az n értéke 0 és 25 között lehet. Ha n értékét elhagyjuk, akkor a cursort alapállapotba helyezzük (megfelel a CURSOR 0-nak). A CURSOR beállítás felhasználható a szöveg kijelzőn való elhelyezésére is.

Példa:

A következő program egy nyilat mozgat balról jobbra a kijelzőn:

```
10 I=0
20 CURSOR I:WAIT0
30 PRINT "=> ":BEEP 1, 1
40 I=I+1 : IF I=26 THEN GOTO 10
50 GOTO 20
```

DATA

szintaxisa: sorszám DATA 1. adat, 2. adat ...
n. adat,

ahol az adat numerikus vagy karakteres konstans, vagy aritmetikai kifejezés lehet.

Ez az utasítás egy adatmezőt határoz meg a memóriában, amihez konkrét értéket rendel hozzá. Ezeket az értékeket mi adhatjuk meg egymás után felsorolva, egymástól vesszővel elválasztva.

A DATA utasítások a READ utasításokkal dolgozhatók fel. A programban tetszőleges számú DATA utasítás állhat, tetszőleges helyen, mert a gép az összes DATA utasítást egy halmaznak veszi. A példát lásd READ utasításnál. Lásd még RESTORE utasítás.

DEGREE

szintaxisa: sorszám DEGREE

Ezzel a paranccsal (vagy utasítással) a gép által meghatározott trigonometrikus egységet adhatjuk meg fokban.

Példa:

A következő program a szög sinusát adja meg 0° -tól 360° -ig; 1° fokenként:

```
10  DEGREE
20  FOR X=0 TO 360 STEP 10
30  PAUSE "SIN(";X;")= "; SIN(X)
40  NEXT X
```

DIM

szintaxisa: sorszám DIM 1. név (dimenzió),...
n. név (dimenzió)

ahol a név a tömb neve, alfanumerikus karaktersorozat lehet (első karaktere kötelezően betű!), a dimenzió pedig egy vagy két szám, változó vagy kifejezés lehet, vesszővel elválasztva.

Ez a parancs(vagy utasítás) egy vagy kétdimenziós tömbök kezelését teszi lehetővé. Három vagy többdimenziós tömböt nem használhatunk! A tömb elemére a tömb nevével és az illető elem sorszámával (az indexével), illetve kétdimenziós tömbbel a sor és az oszlop számával hivatkozhatunk. Pl. az A nevű tömb 6. elemére A(5)-tel hivatkozhatunk. Azért így, mert A(0) a tömb 1. eleme, A (1) a tömb 2. eleme, stb.

Az index értéke 0 és 255 közé eshet. Ha egy tömböt használni akarunk, akkor azt előzőleg egy DIM utasításban (vagy parancsban) deklarálni kell.

A programban kijelölhetünk karakteres tömböket is, amelyek nevének utolsó karaktere kötelezően dollárjel ("\$").

A karakteres tömbök egy elemének hossza legalább 1, legfeljebb 80 karakter lehet. Azt, hogy a tömb elemei hány karakteresek legyenek, a következőképpen kell megadni:

DIM név (dimenzió) * hossz

ahol a hossz az elemek karaktereinek számát adja meg ($1 \leq \text{hossz} \leq 80$).

P1. DIM A\$(10)*25 egy olyan A\$ nevű, 11 elemű egydimenziós karakteres tömb, amelyben egy elemben maximum 25 karakter lehet. Ha a hosszt elhagyjuk, a gép 16 karakter számára foglal le helyet.

Ha egy numerikus- vagy karakteres tömböt DIM-mel deklarálunk, a gép a numerikus tömb elemeit 0-val, a karakteresét szóközzel tölti fel. Egy programban egy tömböt csak egyszer lehet deklarálni!

Példa:

Az alábbi program egy 10 elemű, A nevű tömb elemeit rendezzi növekvő sorba.

```

10 DIM A(9)
20 DATA 1,3,4,8,6,5,7,9,2,0
30 FOR I=0 TO 9
40 READ A(I)
50 NEXT I
60 FOR I=10 TO 1 STEP-1
70 FORJ=1 TO I-1

```

```

80 IF A(J-1) > A(J) THEN GOTO 200
90 NEXT J
100 NEXT I
110 FOR I=0 TO 9
120 PAUSE A(I)
130 NEXT I
140 END
200 A=A(J-1)
210 A(J-1)=A(J)
220 A(J)=A
230 GOTO 90

```

END

szintaxisa: sorszám END

Ez az utasítás a program logikai végén áll, hatására a program befejeződik. Az END utasítás el is hagyható, a program az (logikai) utolsó sor végrehajtása után automatikusan befejeződik. Az END utasítást célszerű a program végére kiténni, mert nélküle a program egy más címen lévő olyan utasításon áll meg, ami nem tartozik a programunkhoz, s így hibás kijelzést kapunk.

FOR...TO...STEP/NEXT

szintaxisa: sorszám FOR változó=n1 TO n2 STEP n3
és sorszám NEXT változó

ahol n1, n2 és n3 numerikus konstans, változó vagy kifejezés lehet.

Ez az ún. ciklusutasítás, amely lehetővé teszi, hogy egy programrészt többször végrehajtsunk (az ún. ciklusmagot).

A program először a változóhoz hozzárendeli az n_1 értéket, majd végrehajtja a FOR és a NEXT közötti részt (a ciklusmagot). Ezután a változó értékét megnöveli n_3 -mal, majd megvizsgálja, hogy a változó értéke meghaladta-e n_2 -t. Ha igen, akkor a NEXT utáni utasításon folytatódik a program. Ha a változó értéke kisebb mint n_2 , akkor még egyszer végrehajtja a ciklusmagot és így tovább, míg a változó értéke nagyobb lesz, mint n_2 . Az elmondottakból következik, hogy a ciklusmag egyszer mindenképpen végrehajtásra kerül, még akkor is, ha a ciklusváltozó már kezdetben is nagyobb n_2 -nél. A ciklus elhagyása után a ciklusváltozó értéke mindenképpen n_3 -mal nagyobb lesz, mint a ciklusmagban utoljára aktuális ciklusváltozó.

Ha a lépésköz 1 ($n_3=1$), akkor a STEP n_3 elhagyható. Az n_1 és n_2 értékei -32768 és $+32768$ közé eshetnek és n_3 értéke csak egész szám lehet.

Egy programban 16 FOR-NEXT hurok ágyazható egymásba

P1.	10	FOR X=1 TO 10 STEP 10	1. szint
	:		
	160	FOR Z=15 TO 100 STEP 3	16. szint
ciklus-	:		
mag	500	NEXT Z	
	600	NEXT X	

A hurokhatárokat nem szabad átfedni!

P1.	10	FOR X=1 TO 3	
	20	FOR B=1 TO 20	
	:		
	100	NEXT X.....	helytelen egymásba
	110	NEXT B.....	ágyazás!

helyesen:

100 NEXTB

110 NEXTX

A program végrehajtása közben nem szabad ciklusba beugrani csak akkor, ha előzőleg ugyanabból a ciklusból ugrottunk ki. A ciklusmag nem természetes úton való elhagyása megengedett (kiugrani szabad), ilyenkor a ciklusváltozó értéke a ciklusmagban utoljára aktuális értéke marad, valamint nem célszerű a ciklusváltozó értékét megváltoztatni. Ezek a ciklus helytelen végrehajtásához vezethetnek.

GCURSOR

szintaxisa: sorszám GCURSOR n

ahol n numerikus konstans, változó vagy kifejezés lehet.

Ezzel a paranccsal (vagy utasítással) a kijelzőn az írási pozíciót jelölhetjük ki (hasonlóan a CURSOR utasításhoz), azzal a különbséggel, hogy az egységet nem karakterekben, hanem pontraszterekben mérjük (lásd A kijelző című fejezetet). Így a kijelzés a 156 pozíció mindegyikére kijelölhető, a számozás balról jobbra növekszik, 0-tól 155-ig. Ha az n paramétert elhagyjuk, hatása a GCURSOR 0-val egyezik meg.

Példa:

A CURSOR-nál adott programot nézzük meg a GCURSOR használatával.

10 I=0

20 GCURSOR I:WAIT 0

```

3Ø    PRINT "== > "
4Ø    I=I+1 : IF I=156 THEN GOTO 1Ø
5Ø    GOTO 2Ø

```

Ha I értékét megváltoztatjuk, a nyíl az új pozíciótól indul.

GOSUB/RETURN

szintaxisa: sorszám GOSUB sorszám vagy
 sorszám GOSUB "cimke" és
 sorszám RETURN

A programozásban fontos szerepet játszanak a szubrutinok. Ezek olyan önálló programrészek, amelyeket egy programban esetleg többször is végre kell hajtani. A programban a megfelelő helyen (helyeken) el kell helyezni az ún. szubrutinhívó utasítást (ill. utasításokat). Ez (ezek), ha a vezérlés rá (rájuk) kerül, aktivizálja (aktivizálják) a megfelelő szubrutint, vagyis elvégzésre kerül a szubrutinban lévő utasítássorozat. Ezután a program a hívást követő utasításon folytatódik.

A BASIC-ben a szubrutinhívó utasítás a GOSUB, mely a programban tetszőleges számban és helyen állhat. Megengedett a szubrutinok egymásba ágyazása.

A szubrutinokat mindig RETURN utasításokkal kell zárni, amely megadja, hogy a szubrutinból mikor kell visszatérni a főprogramba.

Például:

```

:
1000 GOSUB 1000
110  REM FOLYTATAS 1
:
500  GOSUB "FELD"
510  REM FOLYTATAS 2
:
800  END:REM PROGRAM VEGE
1000 "FELD":REM SZUBRUTIN
:
2000 RETURN:REM - SZUBRUTIN VEGE

```

Lásd még ON GOSUB

GOTO

szintaxisa: sorszám GOTO sorszám vagy
sorszám GOTO "cimke"

A GOTO-val a program futását minden korlátozás nélkül az adott sorszámú (vagy címkéjű) utasításon folytathatjuk. Ez az un. feltétel nélküli vezérlés-átadás.

Példa:

```

:
100  IF A 0 THEN GOTO 550
:
200  IF X Y THEN GOTO "FOLYT"
:
300  "FOLYT": X=Y
:

```

```

5000 GOTO 1000
:
5500 PRINT X-A-Y
:
7000 END

```

Lehetőség van arra, hogy a programot GOTO-val elindítsuk. Ekkor a sorszám (vagy címke) az első végrehajtandó utasítás sorszáma (vagy címkéje) kell hogy legyen. A GOTO-val való programindítás hatását lásd a függelékben (Programindítási módszerek összehasonlítása).
Lásd még ON GOTO.

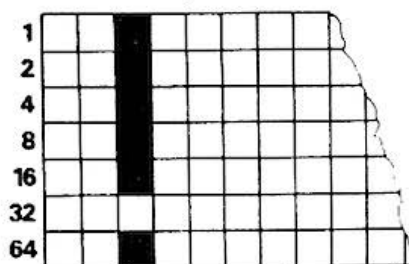
GPRINT

szintaxis: sorszám GPRINT m1, m2, ... mn

ahol m1, m2, ... mn az ún. mintakifejezések.

A PRINT-nél karakterek íródtak ki a kijelzőn, a GPRINT-nél pont mátrix-szal megadott ábrák jelennek meg.

A mintakifejezésekkel kapcsolatban tudni kell, hogy a gép a kijelző minden pontsorához egy számértéket (súlyfaktort) rendel hozzá. A hozzárendelés a következő:



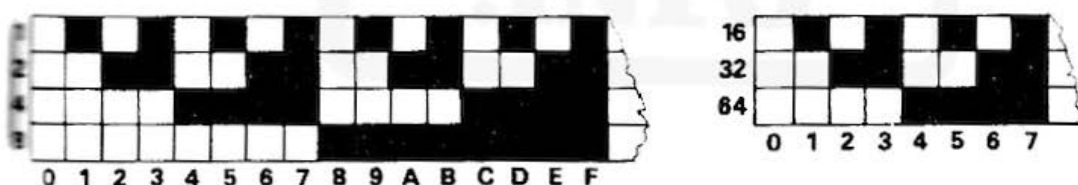
3.1. ábra

tehát az 1. sorhoz 1-et, a 2. sorhoz 2-t, ..., a 7. sorhoz 64-et rendel hozzá.

Egy adott pontmintát előállító mintakifejezés értéke azon sorok súlyfaktorainak összege, amelyeknek az illető minta létrehozása közben látszaniok kell. A 3.1. ábrán látható felkiáltójel esetében a megfelelő súlyfaktorok összege: $1+2+4+8+16+64=95$ tehát a GPRINT 95 megrajzolja a "!"-t.

Ha a GPRINT több mintakifejezést tartalmaz, akkor azokat pontosvesszővel (ekkor az ábrák közvetlenül egymás után lesznek kirajzolva), vagy vesszővel (ekkor az ábrák egy oszlop kihagyásával kerülnek egymás után) választhatjuk el egymástól.

Egy adott pontmintához tartozó mintakifejezéssel megadható hexadecimális számokkal is (a hexa-számokról lásd Hexadecimális számok című fejezetet!). Ekkor a kijelző két sorát két csoportra osztjuk. A felső négy sor az egyik, az alsó három sor a másik csoport (lásd a 3.2. ábrán):



3.2. ábra

Például az előbbi felkiáltójel GPRINT "5F"-fel rajzolható ki. A hexadecimális kód első jegyét mindig az alsó három sorból, a másodikat a felső négy sorból kell venni. A hexadecimális mintakifejezéseket idézőjelek közé elválasztás nélkül írhatjuk.

Példa:

Ez a program megadja az egy oszlopon belül kirajzolható összes ábrát.

```

10   FOR X=0 TO 127
20   WAIT 0
30   GPRINT X;
40   NEXT X
50   FOR Y=0 TO 100:NEXT Y
60   CLS
70   GOTO 10

```

vagy a következő példa:

```

5    CURSOR 10
8    PAUSE "FEKETE VONAT"
10   WAIT 0
20   FOR I=31 TO 170
30   GCURSOR I
35   ON ERROR GOTO 150
40   GPRINT "0038113F797F7F3838787E3F7E783810"
50   BEEP 3, I/16, 35
60   A=I
70   GOSUB 100
80   GOSUB 100
85   IF I=42 OR I=84 THEN BEEP 2, 75, 700
90   NEXT I
100  GCURSOR (A-15)
110  GPRINT "0038113F79793F39393F79793F1138"
120  BEEP 1, A/8, 45
130  A=A-15
140  RETURN
150  CLS
160  PAUSE "ELEMENT"

```

```
170 GOTO 10
    vagy
170 END
```

GRAD

szintaxisa: sorszám GRAD

Ez a parancs (vagy utasítás) a trigonometrikus egységet állítja be újfokban (ahol a kör 400 egy- ségre van bontva és nem 360-ra).

IF ... THEN

szintaxisa: sorszám IF U THEN utasítás 1:....:
utasítás n

ahol U numerikus változó, aritmetikai kifejezés, logikai változó vagy logikai kifejezés lehet (pl. $A < B$ vagy $X = Y$ OR $Z = D$ AND $A = B$ stb.)

Ha a program végrehajtása során a vezérlés egy IF utasításra kerül, a gép kiértékeli az IF utáni logikai kifejezést. Ha a logikai kifejezés értéke igaz, akkor a THEN utáni utasításon (vagy utasításokon) folytatódik a végrehajtás. Ha a logikai kifejezés értéke hamis, akkor a program nem megy a THEN utáni részre, hanem az IF utáni utasítást hajtja végre.

Ha THEN után közvetlenül értékadó utasítást akarunk végrehajtani, akkor alakja a következő:

pl.: IF $X = Y$ THEN LET $A = B : C = D : GOTO 100$.

Ha a THEN után GOTO utasítás van, akkor a GOTO kulcsszó elhagyható, tehát a

40 IF X=Y THEN GOTO 1000 és a
40 IF X=Y n THEN 1000 azonos hatású.

A PTA 4000 tulajdonsága, hogy ha az IF után relációjel nélküli egyetlen aritmetikai kifejezést írunk, akkor annak értéke szerint hajtódik végre a feltételes ugrás. Ha a kifejezés értéke pozitív, a THEN utáni utasítás következik, ha a kifejezés értéke 0 vagy negatív, a számítógép a THEN utáni utasítást átlépi. Utasítások láncolásánál ha valamelyik utasítás feltétel nélküli ugrás utasítást tartalmaz, az csak az utolsó lehet, vagyis a GOTO-t nem követheti kettőspont után újabb utasítás. Ha a THEN után kettősponttal elválasztott utasítás sorozatot írunk, az csak akkor kerül végrehajtásra, ha a feltétel teljesül, ellenkező esetben a következő sor számon folytatja a programot.

INPUT

szintaxisa: sorszám INPUT "szöveg 1";változó 1,
....., "szöveg n"; változó n

Ez az utasítás lehetővé teszi, hogy a programba a billentyűzetről adatokat adjunk be.

Ha erre az utasításra kerül a vezérlés, a program végrehajtása megszakad és a kijelzőn vagy kérdőjel (ha a "szöveg" nem volt megadva), vagy pedig a "szöveg" jelenik meg.

Ezután bebillentyűzzük az adatokat, s a beadott adatot **ENTER**-rel kell lezárni. Ha az összes adatot beadjuk, a program folytatódik. Több adatot

ENTER -rel, vagy vesszővel elválasztva adhatunk be. Ha rossz adatot adtunk be, de az ENTER -t még nem nyomtuk le, a hibás adatok **CL** -el törölhetők. Ha a "szöveg"-(ek)et megadjuk, azokat a változó(k)-tól elválasztjuk pontosvesszővel (pl.: 10 INPUT "A=n"; A), vagy vesszővel (pl.: 10 INPUT "A=n, A). Az előbbi esetben a beadott változó közvetlenül a "szöveg" után jelenik meg, az utóbbi esetben a 13. pozíción jelenik meg. A változók után csak vesszőnek szabad állni (pl.: 10 INPUT "A=n";A,"B=n";B helyes, 10 INPUT "A=n" A; "B=n";B helytelen).

Példa:

Az alábbi program a kör kerületét és területét számítja ki, adott sugárral.

```

10 INPUT "SUGAR=?";S
20 T=S*S*3.14
30 K=2*S*3.14:WAIT 100
40 PRINT "TERULET="; T
50 PRINT "KERULET=x"; K
60 END

```

LET

szintaxisa: sorszám LET változónév = n

ahol n lehet numerikus vagy karakteres konstans, változó vagy kifejezés.

A LET parancs (vagy utasítás) az un. értékadó utasítás. Hatására az egyenlőségjel bal oldalán lévő változó felveszi az egyenlőségjel jobb oldalán lévő értéket. Tehát a LET A=2 hatására az A nevű változó értéke 2 lesz. A LET kulcsszó el is hagyható,

tenát az A=2 és a LET A=2 értékadások egyenértékűek. A LET kulcsszót kötelezően a THEN utáni első értékadásban kell használni, azaz pl.: 10 IF X<10 THEN LET X1=X : Y1=Y : GOTO "A" a megfelelő használata.

LOCK

szintaxisa: sorszám LOCK

Ez a parancs (vagy utasítás) az üzemmód váltást tiltja le. Ha pl. RUN üzemmódban kiadjuk a LOCK-ot, nem válthatunk át sem PRO, sem RESERVE üzemmódba. Hatása az UNLOCK-kal oldható fel (lásd ott).

ON ERROR GOTO

szintaxisa: sorszám ON ERROR GOTO sorszám
vagy sorszám ON ERROR GOTO "cimke"

Ez az utasítás a program végrehajtására csak akkor van hatással, ha a program futásakor valamilyen hiba lép fel. Hatására (hiba esetén) a program nem szakad meg, hanem a GOTO után megadott sorszámú, vagy címkéjű utasításra adódik a vezérlés.

Mindig az utoljára kiadott ON ERROR GOTO utasítás érvényes. Ha a letiltandó hiba kódja 1, 7, vagy 32 (lásd Függelékben - a PTA 4000 hibaüzenetei), a program a hibára megakad.

Példa:

A programban a nevező=0 esetén 0-val való osztás lépne fel, nem történik hibajelzés, a program folytatódik.

```

100 "ELEJE" : INPUT X,B,C,E
150 A=X-B
160 D=C-E
170 ON ERROR GOTO "HIBA"
180 PRINT A/D
:
1000 END
1010 "HIBA":PAUSE "NULLAVAL VALO OSZTAS!":
GOTO "ELEJE"

```

ON...GOTO / ON ... GOSUB

szintaxisa: sorszám ON m GOTO sorszám 1, ...
sorszám n vagy sorszám ON m GOTO
"cimke 1", ..., "cimke n"

ahol m numerikus konstans, változó, vagy kifejezés lehet.

A GOTO kulcsszó helyett szerepelhet a GOSUB kulcsszó is.

Ez az utasítás a program végrehajtását, a numerikus kifejezés értékétől függően ágaztatja el a céllistában felsorolt megfelelő sorszámú (vagy címkéjű) utasításra. Ha az m értéke nem egész szám, akkor a gép annak egész részét veszi.

Ha a numerikus kifejezés értéke 1, akkor a céllista első eleme által meghatározott utasításra, ha értéke 2, akkor a második eleme által meghatározott utasításra, ha értéke k, akkor a k-adik eleme által meghatározott utasításra adódik a vezérlés.

Ha a numerikus kifejezés értéke ≤ 0 vagy nagyobb, mint a céllistában szereplő ugrási címek száma, akkor a program végrehajtása az ON...GOTO (vagy

ON...GOSUB) utáni utasításra adódik. Ha ON...GOSUB-ot használunk, akkor ez a program valamelyik szubrutinját hívja. A szubrutin végrehajtása után a program az ON...GOSUB utáni utasításon folytatódik.

Példa:

```

      :
1000 ON T GOTO 1000, 2000, 3000
110  REM FOLYTATAS
      :
5000 ON V GOSUB "OLV", "FELD", "KKIR"
510  REM FOLYTATAS
      :
999  END
1000 REM ELSO ELAGAZAS
      :
1500 GOTO "EL"
2000 REM MASODIK ELAGAZAS
      :
2500 GOTO "VISSZA"
3000 REM HARMADIK ELAGAZAS
      :
3500 GOTO "UGR"
5000 "OLV":....: RETURN
6000 "FELD":....: RETURN
7000 "KIIR":....: RETURN

```

Ez a program először a T értékétől függően elugrik az 1000, 2000, vagy 3000 sorszámú utasításra, később V-től függően hívja az "OLV", "FELD" vagy

"KIIR" címkeű szubrutinokat.

PAUSE

szintaxis: sorszám PAUSE printlista

ahol a printlista lehet numerikus és/vagy karakteres konstans, változó és/vagy kifejezés, vesszővel és/vagy pontosvesszővel elválasztva.

A PRINT utasítás hatására a program megáll és a kívánt kiírás megtörténik. Újra indítás csak az ENTER beadásra folytatódik. A PAUSE a PRINT utasítás automatikusan tovább induló változata. A kijelzés abban különbözik a PRINT-től, hogy a program rövid időre leáll, a kívánt kiírás rövid ideig megjelenik, majd a program tovább indul.

E parancs (vagy utasítás) hatására kijelzésre kerülnek a printlista elemei, és a program futása kb. 0.85 másodperc múlva folytatódik. Ha a printlista elemeit pontosvesszővel választjuk el egymástól, akkor azok közvetlenül egymás után kerülnek kiírásra. Ha az elválasztásra vesszőt használunk, akkor a kijelző két részre lesz bontva, az első kiírandó elem a kijelző elejétől, a második a kijelző közepétől lesz kiírva.

Ha a kiírást nem vezéreljük (pl.: CURSOR utasítással vagy pontosvesszővel), akkor a karaktereket balra tömörítve, a numerikus értékeket jobbra tömörítve írja ki.

Példa:

Ez a program -15 -től +15 -ig kiírja a számok köbét, PAUSE-val.

```
1Ø    FOR X=-15 TO 15
2Ø    PAUSE X, X ^ 3
3Ø    NEXT X
```

Próbáljuk ki, hogy a 2Ø-as sorban vessző helyett pontosvesszőt használunk.

PRINT

szintaxisa: sorszám PRINT printlista

ahol a printlistára ugyanazok vonatkoznak, mint a PAUSE-nál.

Hatása azonos a PAUSE utasítással, csak az a különbség, hogy a program a kiírást követően az ENTER lenyomásával folytatódik, ha csak WAIT utasítással mást nem adunk meg (lásd majd ott!).

A kiírási formátum megszerkesztése azonos a PAUSE-nál leírtakkal (vessző, pontosvessző használata).

A CURSOR utasítással a pozicionálás vezérelhető.

Ha több karaktert íratnánk ki, mint amennyi a kijelzőn egyszerre látható lehetne, akkor csak az első 26 karakter látszik.

A kiírási formátumot megszerkeszthetjük az USING utasítással is (lásd ott!).

Példa:

A 2Ø-dik sorral megjelenik a 3.1415, majd ENTER lenyomása után a 4Ø-dik sor kiírja azt, hogy PTA-4000

```
1Ø    A=3.1415
2Ø    PRINT A
3Ø    A$="PTA-4000"
4Ø    PRINT A$
```

RADIAN

szintaxisa: sorszám RADIAN

Ez az utasítás (vagy parancs) a trigonometrikus mértékegységet állítja be radiánban.

RANDOM

szintaxisa: sorszám RANDOM

Az RND függvényt használva (ez az álvéletlenszámgenerátor, lásd a beépített BASIC függvényeknél) valahányszor újra bekapcsoljuk a gépet, mindig ugyanazt az álvéletlen számsorrendet kapjuk. A gép bekapcsolása után kiadjuk a RANDOM parancsot (vagy utasítást), akkor ha utána használjuk az RND-t, mindig más-más álvéletlenszámsorrendet kapunk.

Példa:

Az alábbi program teljesen véletlenszerű dobókockadobásokat generál.

```
1Ø    RANDOM
2Ø    PAUSE RND(6)
3Ø    GOTO 2Ø
```

Lehetőség van arra, hogy az 1 és a felső határ közé eső egész értéket adó RND függvénytől tetszés szerinti alsó és felső karakterek közé eső, tetszés szerinti sűrűségű véletlen számsort állítson elő. (A véletlen számsorozat jó közelítéssel egyenletes eloszlású.) MN minimális és MX maximális értékű, S sűrűségű véletlen számsorozat előállítását teszi lehetővé az alábbi program.

```

1Ø    RANDOM
2Ø    MN=2
3Ø    MX=12Ø
4Ø    S=1Ø
5Ø    A=MN+S*RND ((MX-MN)/S)
6Ø    PRINT A
7Ø    GOTO 2Ø
8Ø    END

```

READ

szintaxis: sorszám READ változó 1, ...,
 változó n

ahol n változók lehetnek numerikusak, vagy karakteresek.

Ez az utasítás a DATA-ban definiált adatokat rendeli hozzá a READ után megadott változókhoz.

Ha a READ utasítás több változót tartalmaz, ez azonos hatású, mint a szétválasztott READ utasítások egy sora. Tehát pl.: a 1Ø READ A,B,C azonos hatású, mint a

```

1Ø    READ A
2Ø    READ B
3Ø    READ C

```

Ha a DATA-ban egy numerikus kifejezés van megadva, akkor annak kiszámított értékét rendeli hozzá a READ a megfelelő változóhoz. A READ utáni változók és a DATA-ban felsorolt adatoknak illeszkedniök kell egymáshoz, azaz numerikus változóhoz numerikus kifejezést, karakteres változóhoz karakteres kifejezést rendelhetünk hozzá.

Ha a DATA-ban kevesebb adat van, mint a READ-ban szereplő változók száma, ez hibajelzéshez vezet, ha több, akkor a maradék nem kerül feldolgozásra. Ugyanaz a hibajelzés keletkezik, ha a program tartalmaz READ utasítást, de DATA utasítást nem. Lásd még a RESTORE utasítást.

Példa:

Ez a program a hét napjainak nevét írja ki, amit DATA-ban adtunk meg.

```
1Ø DATA "HETFO", "KEDD", "SZERDA", "CSUTORTOK",  
    "PENTEK", "SZOMBAT", "VASARNAP"  
2Ø FOR X=1 TO 7  
3Ø READ NAP Ø  
4Ø PAUSE NAP Ø  
5Ø NEXT X.
```

REM

szintaxisa: sorszám REM karaktersorozat

A REM-mel megjegyzéseket helyezhetünk el a programban. A REM után következő karaktersorozatot a gép teljes egészében megjegyzésnek tekinti.

Pl.: a 1Ø REM MEGJEGYZES:A=Ø utasításban az A=Ø-át nem műveletnek, hanem megjegyzésnek tekinti, így azt nem fogja végrehajtani.

A REM-ek használatának előnye, hogy általuk áttekinthetőbb lesz a program.

RESTORE

szintaxis: sorszám RESTORE

vagy sorszám RESTORE n

A DATA és READ utasításokkal kapcsolatban kell megemlíteni a RESTORE utasítást. Szerepe az, hogy végrehajtása után mindig a legalacsonyabb sorszámú DATA utasítás első adatát rendeli hozzá a READ első változójához, majd innen folytatja a további hozzárendeléseket.

Ha egy programot RUN-nal indítunk, az automatikusan elvégzi a RESTORE-nek megfelelő műveletet.

Továbbá a RESTORE lehetővé teszi, hogy ugyanazon adatsor egy programon belül többször is felhasználható.

Pl.: Háromszögek kerületét kiszámító programunk legyen a következő: 1, 2, 3, majd 4, 5, 6, és 7, 8, 9 cm oldalúak.

```

10 DATA 1, 2, 3, 4, 5, 6, 7, 8, 9
20 FOR I=1 TO 3
30 READ A
40 READ B
50 READ C
60 PRINT I; "HAROMSZOG KERULETE=", A+B+C
65 RESTORE
70 NEXT I
80 PRINT "NINCS TOBB"
90 END

```

A program a 65 RESTORE nélkül 1. 2. 3. háromszög kerületét, a 65 RESTORE-val mindháromszor az első háromszög kerületét számítja, azaz A-hoz, B-hez és C-hez az első három adatot rendeli.

A RESTORE után írt sorszám lehetővé teszi, hogy ne csak a legalacsonyabb sorszámú DATA utasítástól kezdjük beolvasni a data listát. Ez több adatcsoport szétválasztását teszi lehetővé úgy, hogy minden adatcsoportot külön sorszámon kezdődő DATA utasításba írunk, és egy RESTORE n utasítással (ahol n a kiválasztott DATA utasítás sorszáma) leválasztjuk az adatcsoportot.

STOP

szintaxisa: sorszám STOP

Ha a program végrehajtása során a vezérlés egy STOP utasításra kerül, futása megszakad és a kijelzőn

BREAK IN sorszám

jelenik meg. Ha a programot tovább akarjuk folytatni,

CONT és ENTER -t kell beírni.

A STOP kiválóan használható programhibák keresésénél. Egy programrész végén ha elhelyezünk egy STOP utasítást, hatására a program végrehajtása megszakad, le tudjuk kérdezni a program változóit.

Ha a vizsgálat után tovább akarunk dolgozni a programmal, nem kell előlről indítani, hanem CONT-tal folytathatjuk a STOP utáni utasítással.

Lekérdezésnél átmegyünk PRO üzemmódban és a ↓ billentyű lenyomására a sorszám:STOP íródik ki. Innét előre és hátra léphetünk a programban. Visszatérve a RUN üzemmódba CONT-tal folytatódik a program.

TRON/TROFF

szintaxisa: sorszám TRON
 illetve sorszám TROFF

A PTA-4000-nél programkövetésre van lehetőség a TRACE működésében. Ezt az állapotot a RUN üzemmódban kiadott TRON paranccsal (vagy utasítással) idézhetjük elő. Hatására a program végrehajtása lépésenként történik. A TRACE állapot TROFF paranccsal (vagy utasítással) megszüntethető.

A programkövetés a TRON paranccsal a következőképpen végezhető el:

1. RUN üzemmódban adjuk ki a TRON parancsot (TRON és **ENTER**), vagy helyezzük el a programban megfelelő helyre a TRON utasítást.
2. Indítsuk el a programot. Ekkor a gép végrehajtja az első utasítást, kijelzi a sorszámát és leáll. Ha STOP utasítás van a programban, ott BREAK IN sorszám kijelzést ad.
3. Ezután több lehetőségünk van:
 - a.) Ha lenyomjuk a **↓** billentyűt, végrehajtásra kerül a következő utasítás, megjelenik a sorszáma, majd a program leáll. Mindig egy teljes utasítássor kerül végrehajtásra, tehát, ha a sorban több utasítás van (kettősponttal elválasztva), mindegyik végrehajtásra kerül.
 - b.) Ha egy végrehajtott sor után lenyomjuk a **↑** billentyűt, megjelenik az éppen végrehajtott utasítássor. Ez addig látható, amíg a **↑** billentyűt lenyomva tartjuk.

- c.) A TRACE állapotban bármikor lekérdezhetjük (pl.: A és **ENTER** lenyomásával), vagy megváltoztathatjuk (pl.: A=3 és **ENTER**) egy (vagy több) változó tartalmát. A program utasításait ekkor azonban nem szabad megváltoztatni.

Ha a TRON-t egy programban utasításként helyezzük el, akkor utána következő programrész TRACE állapotban kerül végrehajtásra, amíg a TRACE-t fel nem oldjuk.

UNLOCK

szintaxisa: sorszám UNLOCK

Ez az utasítás (vagy parancs) a LOCK hatását oldja fel, tehát lehetővé teszi az üzemmódváltást.

USING

szintaxisa: sorszám USING formátum

vagy sorszám PAUSE USING formátum; n

vagy sorszám PRINT USING formátum; n

vagy sorszám LPRINT USING formátum; n

ahol n numerikus és/vagy karakteres konstans, változó vagy kifejezés lehet, a formátum pedig vezérlőkérdőjeleket tartalmaz.

Az USING-gal a kiíratási formátumot lehet megalkotni, hatása a következő USING kiadásáig tart. Az USING szerepelhet önmagában, vagy a kiírató utasítások után.

Az USING utáni szerkesztőkaraktereket idézőjelek (") között kell megadni.

Szerkesztőkarakterek:

- # - numerikus mezőt határoz meg, ahol a számjegyek jobbra igazítva jelennek meg. A nullák az első értékes jegyig szóközzel lesznek helyettesítve. Ha a kijelzendő számjegyek száma több, mint a lefoglalt hely, hibajelzést kapunk,
- * - a numerikus mező értékes jeggyel nem kitöltött jegyei "*" -gal kerülnek feltöltésre,
- . - (egy pont), a numerikus mezőben megadja a tizedespont helyét,
- , - (vessző) a tizedesponttól balra lévő minden harmadik számjegyet vesszővel választja el a könnyebb olvashatóság kedvéért,
- ^ - a numerikus kifejezés értéke normál alakban kerül kijelzésre (lebegő pontos kijelzés),
- + - a pozitív előjel is megjelenik a szám előtt,
- ' -
- & - karaktersorozat kijelzését teszi lehetővé, a karakterek balra tömörítve jelennek meg a kijelzőn. Ha több karakter van, mint ahány & jel, a többletkaraktereket jobbról levágja, és azok már nem kerülnek kijelzésre.

Megjegyzések:

1. A numerikus mező megállapításánál az előjelet is figyelembe kell venni! (A pozitív előjelet is)
2. A vessző (",") hatására az elválasztó vesszők csak az egész részben kerülnek kijelzésre!
(Pl.: 7123456.78952 esetén a megjelenítés 7,123,456.78952 lesz, tehát a "9" és "5" közé nem kerül vessző.) A vesszőt a karakterszámnál figyelembe kell venni.

Példák:

legyen 1Ø A=12.375 PRO üzemmódban
 2Ø B=11ØØ
 3Ø Z="ABCDEF"

RUN üzemmódban

PRINT USING " #### " ; A

12

PRINT USING " #### . " ; A

* 12.37

PRINT USING " + ### . ## ^ " ; B

+ 1.1ØE Ø3

PRINT USING " &&& " ; AZ

ABC

PRINT USING " &&&&& #### . ## ^ " ; AZ, A

ABCDE *** I.23E Ø1

WAIT

szintaxis: sorszám WAIT n

ahol n numerikus konstans, változó vagy kifejezés lehet.

A WAIT utasítás módosíthatja a PRINT vagy a GPRINT utasítás hatását. A WAIT utasítás az utána következő PRINT utasításokat olyan programozható PAUSE utasítássá változtatja, melynél a megállás időtartama meghatározható.

A WAIT utáni értéktől függően, meghatározott idő múlva folytatódik a program a PRINT, vagy GPRINT után, tehát nem kell az ENTER -t lenyomni.

Az n értéke 0 és 65535 között lehet. Ha a WAIT után nem áll ott az n, akkor a PRINT vagy GPRINT hatását nem változtatja meg.

A késleltetés idejét az n határozza meg, nagysága kb. $n * \frac{1}{64}$ másodperc, tehát ha $n=0$, ez gyakorlatilag leolvashatatlan, ha $n=65535$, ez kb. 17 percnyi, WAIT64 1 sec, WAIT3840 1 percnyi várakozást jelent.

A BEÉPÍTETT BASIC FÜGGVÉNYEK

Ebben a részben kerülnek ismertetésre a gépen alkalmazható aritmetikai, trigonometrikus, karakterkezelő, logikai és egyéb beépített függvények. Ezeket nemcsak a programokban használhatjuk, hanem akkor is, ha a gépet számológépként használjuk.

4.1. Aritmetikai függvények

ABS

szintaxisa: ABS (n)

ahol n numerikus konstans változó
vagy kifejezés lehet

Ezzel a függvénnyel az n paraméter abszolút értékét határozzuk meg.

$$\text{ABS } (n) = \begin{cases} n, & \text{ha } n \geq 0 \\ -n, & \text{ha } n < 0 \end{cases}$$

Példa:

A következő program egy beadott szám abszolút értékét írja ki.

```
10 INPUT "SZAM : " ; A
20 X=ABS (A)
30 PAUSE "ABS (" ; A ; ") = " ; X
40 GOTO 10
```

DEG

szintaxisa: DEG (n)

ahol n numerikus konstans, változó,
vagy kifejezés lehet.

Ezzel a függvénnyel a fok-perc-másodpercben megadott szög értékét számíthatjuk át tizedestörtben megadott szöggé. Az n értékét

FFFF . PPM

alakban kell megadni, ahol F a fok, P a perc, M a másodperc. (Pl. $15^{\circ} 10' 5''$ megadása: 15.1005).

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
DEG (45.30) <input type="text" value="ENTER"/>	45.5
DEG (12.2110) <input type="text" value="ENTER"/>	12.35277778

DMS

szintaxisa: DMS (n)

ahol n numerikus konstans, változó vagy kifejezés lehet.

A DMS, az n által tizedestörtben megadott fokot számítja át fok-perc-másodpercbe.

Példa:

<u>Betáplálás</u>	<u>Kijelzés</u>
DMS (30.25) <input type="text" value="ENTER"/>	30.15
DMS (12.4125) <input type="text" value="ENTER"/>	12.2445

Az utóbbi úgy értendő, hogy $12.4125^{\circ} = 12^{\circ} 24' 45''$.

EXP

szintaxisa: EXP (n)

ahol n numerikus konstans, változó, vagy kifejezés lehet.

Ez a függvény az e^n -t számítja ki.

Példa:BetáplálásEXP (2.5) **ENTER**EXP (-3.47) **ENTER**Kijelzés

12.18249396

3.111703066E-02

INT

szintaxisa: INT (n)

ahol n numerikus konstans, változó
vagy kifejezés lehet.

Az INT, az n paraméter egész részét számítja ki.
A szám egész része, a számnál nem nagyobb, legnagyobb egész szám például $\text{INT}(16.23) = 16$ és
 $\text{INT}(-8.2) = -9$.

Példa:

A következő program egy beadott szám egész részét számítja ki.

```
10 INPUT "SZAM : " ; A
20 X=INT (A)
30 PAUSE "INT("; A;") = " ; X
40 GOTO 10
```

LOG / LN

szintaxisa: LOG (n)

illetve: LN (n)

ahol n numerikus konstans, változó,
vagy kifejezés lehet.

A LOG az n paraméter 10-es alapú, az LN a termé-

szetes alapú logaritmusát számítja ki.

Példa:

Betáplálás

Kijelzés

LOG (1000)	ENTER	3
LN (3.7)	ENTER	1.30833282

PI

A **SHIFT** és **↓** lenyomásával lehetőség van lehívni a Ludolf-féle számot (a π -t), 9 tizedesjegy pontossággal.

RND

szintaxisa: RND (n)

ahol n numerikus konstans, változó vagy kifejezés lehet.

Az RND függvény egy adott intervallumba eső álvéletlen számot határoz meg.

Az RND(N) függvény által generált álvéletlen szám $n \geq 1$ esetén 1 és INT(n) közé eső egész szám, $0 \leq n < 1$ 0 és 1 közé eső maximum 10 értékes jegyet tartalmazó szám.

A számok igen hosszú periódusú ismétlődő számsorokat alkotnak, így az álvéletlen szám valódi véletlen számnak tűnik.

Ha a gépet bekapcsoljuk, mindig ugyanazt a véletlenszám sorrendet kapjuk. A RANDOM paranccsal (vagy utasítással) lehetőség van arra, hogy új sorrendet generáljunk, természetesen ugyanarra a paraméterre vonatkozólag.

Példa:

A következő program totó-tippeket generál.

```

10 RANDOM
20 TS = "12X"
30 A=RND(3) : WAIT 180
40 PRINT MIND$ (TS,A,1)
50 INPUT "KELL MEG TIPP? (I,N)";IS
60 IF IS="I" THEN GOTO 30
70 IF IS="N" THEN PAUSE "XXX VEGE XXX": END
80 GOTO 50

```

A következő program pedig lottó-számokat állít elő.

```

2 DIM B(10)
5 WAIT 0
7 B(1)=RND (90)
9 FOR K=2 TO 5
10 A=RND (90)
15 B(K)=A
20 FOR I=1 TO K-1
30 IF B(I)=A GOTO 10
40 NEXT I
45 NEXT K
60 FOR K=1 TO 4
70 FOR I=1 TO 4
80 IF B(I)< B(I+1) THEN 120
90 C=B(I)
100 B(I)=B(I+1)
110 B(I+1)=C
120 NEXT I
130 NEXT K :CLS
140 FOR I=1 TO 5
150 PRINT B(I);
160 NEXT I
170 FOR I=1 TO 300:NEXT I:GOTO 6

```

SGN

szintaxisa: SGN (n)

ahol n numerikus konstans, változó
vagy kifejezés lehet.

Az SGN az ugynevezett előjel (vagy signum) függvény. Értéke az n paramétertől függ.

ha $n > 0$, akkor $SGN(n) = 1$
 ha $n = 0$, akkor $SGN(n) = 0$
 ha $n < 0$, akkor $SGN(n) = -1$

Példa:BetáplálásKijelzés

SGN (17)	ENTER	1
SGN (0)	ENTER	0
SGN (-31)	ENTER	-1

SQR

szintaxisa: SQR (n)

ahol n numerikus konstans, változó
vagy kifejezés lehet.

Ez a függvény az n paraméter négyzetgyökét határozza meg. Hatása megegyezik $\sqrt{\text{SHIFT}} \text{ és } \text{f}$ hatásával. $n < 0$ esetén ERROR 39 hibaüzenetet kapunk.

4.2. Trigonometrikus függvények

Ebben a fejezetben a gépen használható trigonometrikus függvényeket soroljuk fel, a függvények jelentésével együtt.

ACS = arcus cosinus

ASN = arcus sinus

ATN = arcus tangens

COS = cosinus

SIN = sinus

TAN = tangens

A fenti függvények szintaxisa: függvéynév (n)
 ahol a függvéynév a fenti nevek valamelyike (pl. ACS vagy TAN stb.), az n pedig numerikus konstans vagy kifejezés lehet.

A fenti függvények a paraméter megfelelő szögfüggvény értékét számítják ki. A megfelelő trigonometrikus egységet (fok, radián, vagy újfok) természetesen a megfelelő paranccsal vagy utasítással be kell állítani.

Példa:

<u>Betáplálás</u>		<u>Kijelzés</u>
DEGREE	<u>ENTER</u>	>
SIN (90)	<u>ENTER</u>	1
RADIAN	<u>ENTER</u>	>
SIN ($\pi/2$)	<u>ENTER</u>	1
GRAD	<u>ENTER</u>	>
SIN (100)	<u>ENTER</u>	1

4.3. Karakterkezelő String függvények

ASC

szintaxisa: ASC (k)

ahol a k karakteres konstans, változó vagy kifejezés lehet.

A függvény a k paraméter ASC II kódját határozza meg. Ha a k paraméter több karakterből áll, akkor az ASC, balról az első karakter kódját adja meg.

Példa:

Ez a program egy beadott karakter ASC II kódját adja meg.

```
1Ø INPUT " KARAKTER : " K$
2Ø KOD = ASC (K$) : WAIT 32Ø
3Ø PRINT MID$ (K$, 1, 1) ; " KODJA " ; KOD
4Ø GOTO 1Ø
```

CHR\$

szintaxisa: CHR\$ (n)

ahol n numerikus konstans, változó vagy kifejezés lehet.

Ez a függvény az ASC (n) függvény inverze, az n paraméternek megfelelő ASCII karaktert határozza meg. Az n értéke 0 és 255 közé eshet.

Példa:

A következő program a számítógép teljes karakterkészletét írja ki.

```
10 FOR X = 0 TO 255
20 A$ = CHR$(X)
30 PAUSE X; "      "; A$
40 NEXT X
```

INKEY\$

Ez a függvény az adott pillanatban lenyomott billentyű beolvasására szolgál a program futása közben. A függvény szerepelhet egy kifejezés jobb oldalán (pl. A\$ = INKEY\$) vagy állhat önmagában is (pl. 10 IF INKEY\$ THEN ...). Az előbbi esetben tároljuk a lenyomott billentyűt (az A\$ változóban), az utóbbi esetben viszont nem. Ha a függvény végrehajtása közben nincs lenyomva billentyű, a függvény nullstringet "" generál.

Példa:

Az alábbi program kiírja a lenyomott billentyű jelét és kódját.

```
10 CLS
20 A$ = INKEY$
30 IF A$ = "" THEN GOTO 20
40 PAUSE A$ ; "      " ASC (A$)
50 GOTO 10
```

LEFT\$

szintaxis: LEFT\$ (k, n)

ahol k karakteres, n numerikus konstans, változó vagy kifejezés lehet.

Ezzel a függvénnyel a k karaktersorozat értékének, balról számított első n karakterét kaphatjuk meg.

Példa:

```
1Ø INPUT " SZO : " ; A$
2Ø FOR X = 1 TO LEN (A$)
3Ø PAUSE LEFT$ (A$, X) : CLS
4Ø NEXT X
```

Ez a program egy beadott szó szeleteit jeleníti meg a kijelzőn, amely szelet egy karakterrel mindig nő, (pl. "SAMU" esetén rendre S, SA, SAM, SAMU látható).

LEN

szintaxis: LEN (k)

ahol k karakteres konstans, változó vagy kifejezés lehet.

Ez a függvény az k paraméter hosszát adja meg.

Az üres karakter, nulstring(nem szóköz!!) hossza Ø, jele: "".

Példát lásd a LEFT\$ függvénynél, ahol a szeletelés a karaktersorozat hosszáig ment 2Ø FOR X + 1 TO LEN(A\$).

Példa:

```

1Ø   A$ = "EZ EGY STRING"
2Ø   H = LEN(A$)
3Ø   PRINT H,A$
4Ø   H = LEN("SZOVEG")
5Ø   PRINT H, "SZOVEG"

```

A programrészlet az első leállásnál 13, a másodikonál 6 kijelzést ad, ez a stringben lévő karakterek száma.

MID\$

szintaxis: MID\$ (k, n, m)

ahol k karakteres, n és m numerikus konstans, változó vagy kifejezés lehet.

A MID\$ függvény a k karakter sorozat n-edik karakterétől kezdve kiemel m darab karaktert és azt adja vissza. A függvény kiválóan alkalmas karaktersorozatok szeletelésére.

Például, ha A\$ = "KARAKTER", ekkor a MID\$ (A\$, 1,3) értéke "KAR", a MID\$ (A\$, 4, 3) értéke "AKT" lesz.

RIGHT\$

szintaxis: RIGHT\$ (k,n)

Ez a függvény annyiban különbözik a LEFT\$ függvénytől, hogy nem balról, hanem jobbról veszi a k karaktersorozat első n karakterét.

Használata megegyezik a LEFT\$ függvénnyel.

STR\$

Szintaxisa: STR\$ (n)

ahol n numerikus konstans, változó, vagy kifejezés lehet.

Az STR\$ függvény arra szolgál, hogy az n numerikus érték számjegyeiből karaktersorozatot állít elő.

A karakterré alakított számmal nem lehet aritmetikai műveletet végezni.

Példa:

```
10 FOR X = 1 TO 15
20 A$ = STR$ (X+5) : PAUSE A$ : CLS
30 NEXT X
```

A fenti program az A\$ karakter változóba rendre 1+5=6, 2+5=7, ... 15+5=20 értéket tesz, ami már karakter és nem numerikus érték.

VAL

szintaxisa: VAL (k)

ahol k karakteres konstans, változó vagy kifejezés lehet.

A VAL függvény az STR\$ függvény inverze, egy számjegyeket és tizedespontot tartalmazó karaktersorozatból numerikus értéket állít elő. Ha a karakterek között a számjegyeken és a tizedesponton kívül más karakter is van, az átalakítás az első tiltott karakternél befejeződik.

Példa:

```

1Ø    WAIT Ø
2Ø    AØ=INKEYØ:IFAØ="."THEN 2Ø
3Ø    PRINT VAL(AØ)+5
4Ø    GOTO 2Ø

```

A program minden lenyomott számbillentyű értékhez 5-öt hozzáad és kijelzi.

4.4. Logikai függvények

A gép két szám (-32768 és +32767 közötti) között logikai AND, logikai OR kapcsolatot és egy szám negációját (NOT) is elvégzi. A műveletet a gép bitenként végzi.

AND

szintaxis: n AND m

ahol n és m numerikus konstans, változó vagy kifejezés lehet.

Az (ÉS) AND táblázata

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

az eredmény 1, ha mindkét bit értéke 1

Például 5 AND 2 eredménye 0, mert

5 értéke binárisan: 101

2 értéke : 010

$$\begin{array}{r} \text{így} \quad 101 \\ \quad \quad 010 \\ \hline \quad \quad 000 \end{array} \quad \text{vagy}$$

& A AND &3F eredménye 10 decimálisan, mert & A értéke binárisan: 1010, & 3F értéke: 0011 1111,

$$\begin{array}{r} \text{így} \quad 00001010 \\ \quad \quad 00111111 \\ \hline \quad \quad 00001010 \end{array} \quad \text{stb.}$$

OR

szintaxis: n OR m

ahol n és m numerikus konstans, változó vagy kifejezés lehet.

Az (VAGY) OR táblázata

A	B	A+B	
0	0	0	
0	1	1	az eredmény 1, ha legalább az egyik
1	0	1	bit 1 volt.
1	1	1	

Például:

7 OR 14 eredménye 15, mert 7 értéke binárisan: 111
14 értéke: 1110

$$\begin{array}{r} \text{így} \quad 0111 \\ \quad \quad 1110 \\ \hline \quad \quad 1111 \end{array} \quad \text{vagy}$$

3 OR 15 eredménye 15 mert 3 értéke binárisan: 0011
 15 értéke : 1111

így 0011
 1111

 1111 stb.

NOT

szintaxisa: NOT n

ahol n numerikus konstans, változó vagy kifejezés lehet.

A (NEM) NOT művelet táblázata

A \bar{A}

1 0 az eredmény 1, ha a bit 0 volt
 0 1 0, ha 1 volt

Példa:

Eredmény

NOT 0	-1
NOT & F 0	-241
NOT 43	- 44

mert 43 binárisan 00101011
 ennek a negáltja 11010100 = -44

Megjegyzés:

A számítógép aritmetikája negatív számként értelmezi a szám kettes komplementjét,

tehát $00101100 = 44$, ennek kettes komplemente

$$\begin{array}{r}
 11010011 \\
 + \quad \quad 1 \\
 \hline
 11010100 = -44
 \end{array}$$

4.5. Egyéb függvények

MEM

Szintaxisa: MEM

Ezzel a beépített függvénnyel a memóriában lévő szabad byte-ok számát kérdezhetjük le.

PEEK

Szintaxisa: PEEK (n)

ahol n numerikus konstans változó vagy kifejezés lehet.

A PEEK függvénnyel a memória egy byte-jának tartalmát kérdezhetjük le. A byte tartalmát decimálisan kapjuk meg.

POINT

Szintaxisa: POINT (n)

ahol n numerikusan konstan, változó, vagy kifejezés lehet.

A POINT függvény értéke abban a kijelzőoszlopban lévő pontmintához tartozó súlyfaktor értékével egyenlő, ahová az n kifejezés mutat.

Az n értéke \emptyset és 155 közé eshet.

Pl. ha a pontraszter 27-ik oszlopában egy felkiáltó jel van kirajzoltatva (lásd GPRINT utasításnál), akkor a POINT 27 értéke 95 lesz.

POKE

Szintaxisa: POKE (n , m)

vagy : POKE n , m_1 , m_2 ... m_k

ahol n , m , m_1 ..., numerikus konstans, változó vagy kifejezés lehet.

A POKE n , m függvénnnyel az n című byte-ba beírhatjuk az m értéket, mely \emptyset és 255 közé eshet. Ha POKE n , m_1 , m_n ,-t használjuk, akkor az az n című byte-ba m_1 -et, $n+1$ című byte-ba m_2 -t, ..., $n+(k-1)$ byte-ba m_n -t ír be. Az m_1 ... m_k értéke \emptyset és 255 közé eshet. Célszerűen $\text{STATUS2} < n < \text{STATUS3}$.

STATUS

Szintaxisa: STATUS n

ahol n \emptyset és 255 közé eső egész szám lehet.

Ezzel a függvénnnyel a memóriáról szerezhethetünk információkat.

STATUS \emptyset - a memóriában lévő szabad byte számát adja meg (úgy mint a MEM),

STATUS 1 - a memóriában lefoglalt byte-ok számát adja meg,

STATUS 2 - az első szabad byte címe,

STATUS 3 - az utolsó byte címe

STATUS 4-től

STATUS 255-ig - az utoljára feldolgozott sor sorszám egy STOP vagy egy BREAK után.

TIME

Szintaxis: TIME

vagy: TIME = n

ahol n numerikus konstans, változó vagy kifejezés lehet.

A PTA-4000-ben belső óra van, mely a gép kikapcsolt állapotában is jár, amelyet be lehet állítani és bármikor le lehet kérdezni.

Az n paramétert a következőképpen kell megadni:

Pl.: ha az órát január 4-e, 12 óra 25 perc, 43 másodpercre akarjuk beállítani, akkor

TIME = 01 04 12. 2543 és **ENTER** -t kell
hó nap óra p.mp.

beadni.

Ha az időt le akarjuk kérdezni, TIME és **ENTER** -t kell lenyomni.

Példa:

Ez a program egy ébresztőóra lesz.

Megjegyzés:

A program futása közben az E betű megnyomására megjelenik a kijelzőn az ALARM szöveg, ami után óra (kétszámjeggyel) és perc (két számjeggyel), majd **ENTER** billentyűzés után a beírt időben az ébreszt-

tés megtörténik.

Bármely nyomógommbal a jelzés leállítható, ami 5 perc haladékot ad és újból ébreszt. Ha a Q lenyomásával történik a leállítás, akkor csak új beírásra ébreszt.

```

5      REM ORA PROGRAM
10     WAIT 0
20     E$="9999"
30     T$=STR$ (TIME *10000)
40     IF LEN T$=9LET T$="0"+T$
50     CURSOR 0
60     PRINT LEFT$ (T$,2);":";MID$ (T$,3,2);" ";
      MID$ (T$,5,2);":";MID$ (T$,7,2);":";
70     PRINT RIGHT$ (T$,2);:IF E$ < "9999"PRINT " E ";
      LEFT$ (E$,2);":";RIGHT$ (E$,2);":"00"
80     IF INKEY$="E" INPUT "ALARM";E$
90     IF MID$ (T$,5,2)=LEFT$ (E$,2)AND RIGHT$
      (E$,2) <=MID$ (T$,7,2)BEEP 3:GOTO 110
100    GOTO 30
110    IF INKEY$ = "Q"LET E$="9999":CLS:GOTO 90
120    IF INKEY$ < > ""LET E$=STR$ (VAL (E$)+5)
130    IF LEN (E$) < 4LET E$="0"+E$: GOTO 130
140    IF E$ 2359 LET E$="0000":GOTO 30
150    END

```

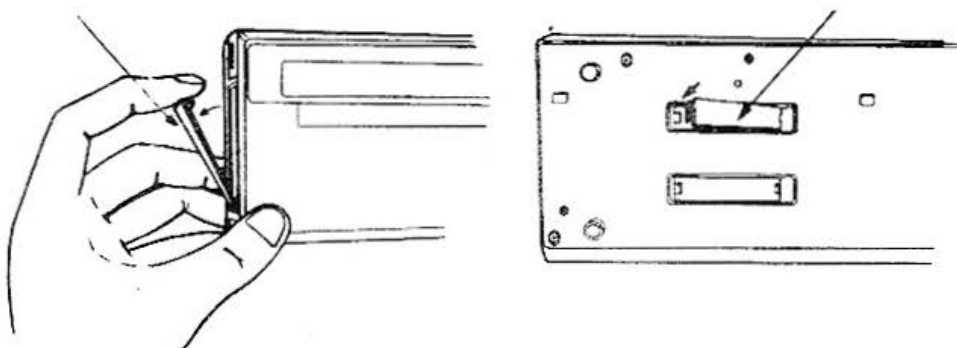
Természetesen első bekapcsoláskor a pontos időt meg kell adni.

5. A KA 160 BŐVÍTŐ EGYSÉG

A PTA-4000 -hez illeszthető a KA 160 bővítő egység, amely egy négyszínű grafikus egységet és egy magnetofonvezérlő egységet tartalmaz, melyen keresztül egyidejűleg két magnetofon csatlakoztatható a géphez.

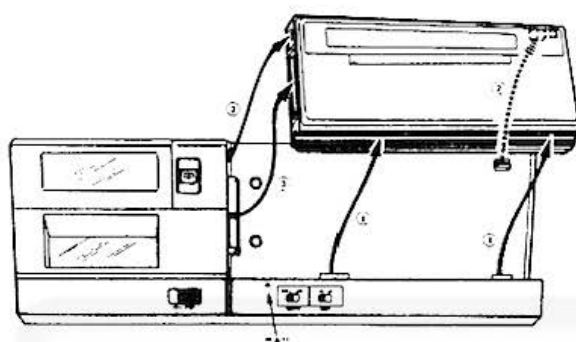
5.1. A bővítő csatlakoztatása a számítógéphez

1. Kapcsoljuk ki a gépet az **OFF** billentyűvel.
Ha a két egységet bekapcsolt állapotban illesztünk össze, a számítógép vezérelhetetlen állapotba kerülhet. Ekkor a gépet az **ALL RESET** gombbal alapállapotba lehet helyezni.
(Lásd: Üzemeltetési útmutató című fejezetben)
2. Vegyük le a gép oldalán lévő csatlakozót védő fedőlapot és pattintsuk be a bővítő alján lévő alkalmas helyre, így az nem veszt el.
Az így szabaddá váló elektromos csatlakozási pontokat NE ÉRINTSÜK MEG !!!, s még rövid időre se hagyjuk szabadon!



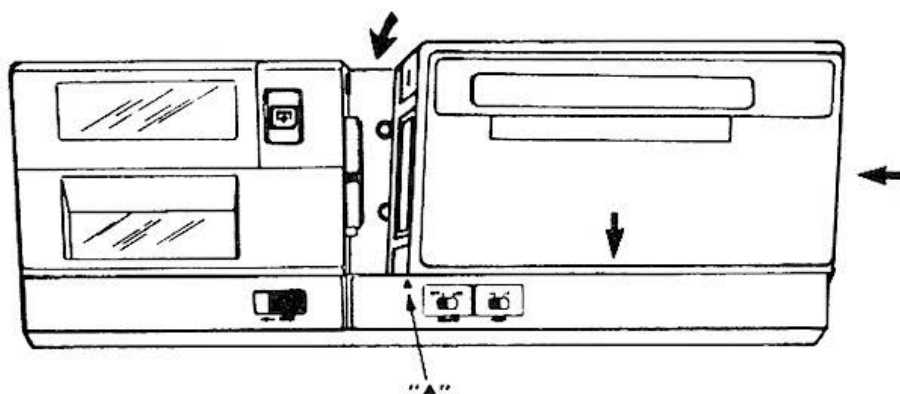
5.1. ábra

3. Helyezzük a számítógép alsó felét a bővítőn lévő sínekbe úgy, hogy a bővítőn lévő vezetőcsapok a gép alsó oldalán és bal oldalán lévő vezetőnyílásba kerüljenek. Az illesztést megkönnyíti a bővítőn lévő ▲ jel, melyhez a gép bal alsó 'sarkát kell illeszteni.



5.2. ábra

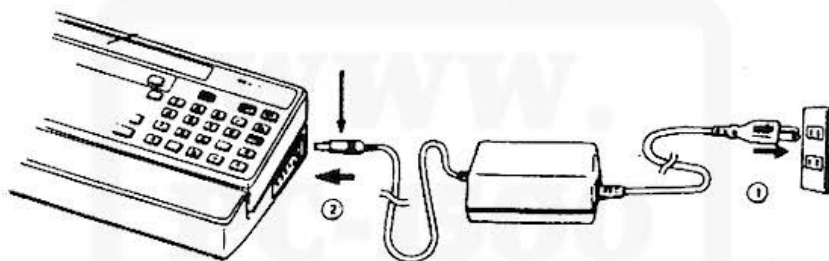
4. Mozgassuk meg a gépet, hogy teljesen feküdjön a sínekre, majd toljuk balra mindaddig, míg a bővítő csatlakozópontjával az összekapcsolás létre nem jön. Ha a csatlakoztatás nehezebben menne, NE ERŐLTESSÜK !, próbáljuk meg újra, mert a gép vagy a csatlakozó károsodhat.



5.3. ábra

5.2. Tápfeszültség ellátás

A KA 160 bővítőt tölthető nikkell- kadmium akkumulátorok látják el energiával. Ha a gép kijelzőjén ERROR 80, ERROR 78, :CHECK 6, vagy NEW:CHECK 6 üzenet látszik, az akkumulátort fel kell tölteni. A bővítő és a gép áramellátása lehetséges E09D hálózati adatterről is, mely egyben akkumulátortöltő is. Az adaptert a PTA-4000, illetve a KA 160 bővítő oldalán lévő ADAPTOR feliratú csatlakozóba kell bedugni. Ha a gépet a bővítőre csatlakoztattuk, az áramellátást a KA 160-ról kapja, ekkor az adaptert a bővítőre kell kötni.



5.4. ábra

5.3. A KA 160 csatlakoztatása magnetofonhoz

A bővítő lehetőséget nyújt arra, hogy a számítógéphez egyszerre két magnetofont csatlakoztassunk, így kazettára programot vagy adatot mentünk ki, illetve arról beolvassunk.

Csak a következő tulajdonságú magnetofonokat használjuk:

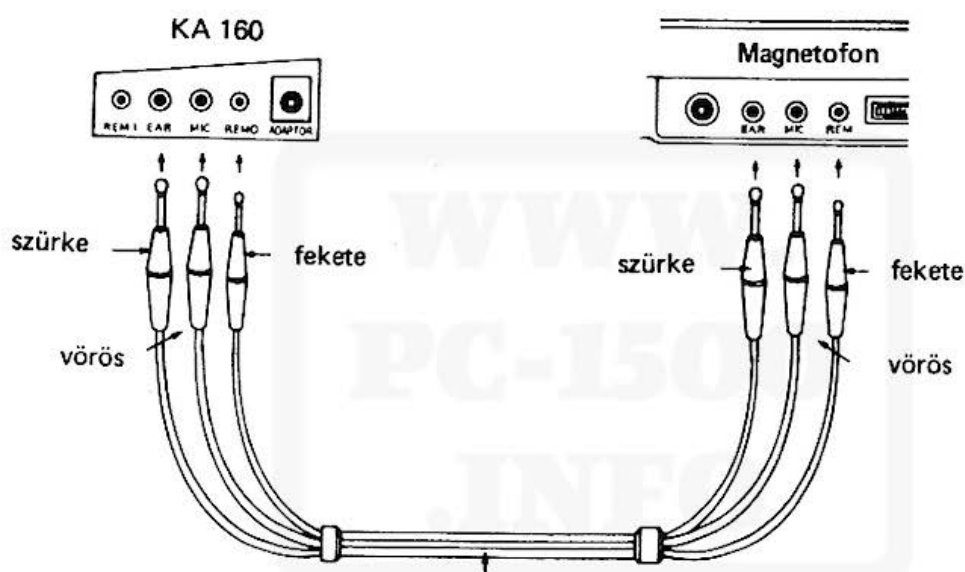
Bemeneti ellenállás: 200 - 1000 ohm

Minimális bemeneti szint: kisebb, mint 3 mV

Kimeneti ellenállás: kisebb, mint 10 ohm

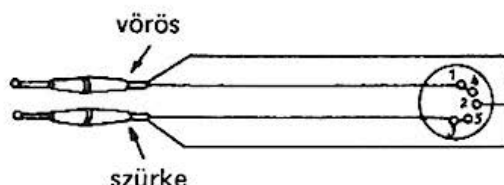
Kimeneti szint: nagyobb, mint 1 V
Minimális kimenő teljesítmény: 100 mV
Torzítás 2 kHz és 4 kHz között: kisebb, mint 15 %
Szalagsebesség ingadozás: kisebb, mint 0,3 %
Jó, ha a magnetofon rendelkezik távvezérelhető "pillanat - állj" funkcióval.

A magnetofon csatlakoztatása:



5.5. ábra

Ha a vezetéken Tuchel (DIN) csatlakozó van, azt a következőképpen kell bekötni.



5.6. ábra

FONTOS MEGJEGYZÉSEK:

1. Sokat használt magnetofonnál felléphetnek olyan paraméterromlások, amelyek az adatrögzítés biztonságát veszélyeztethetik.
2. Ha a magnetofonfej szennyezett, ez növelheti a torzítást és csökkentheti a lejátszási szintet.
3. Kerüljük az olyan szalag használatát, amelynek nagyon alacsony a határfrekvenciája, sérült, gyűrött vagy szennyezett.

5.4. A nyomtató használata

A nyomtató jellemzői:

- a karakterek száma soronként 4, 5, 6, 7, 9, 12, 18 vagy 36, a választott mérettől függően,
- a karakterméret: 1,2x0,8 mm és 1,8x7,2 mm között
- nyomtatási sebesség: a legkisebb karakterméret esetén maximum 11 karakter/másodperc,
- a karakterek a papír hosszirányára merőlegesen és azzal párhuzamosan is felírhatók, a karakterek elforgathatók,
- használható színek: fekete, kék, zöld, piros,
- lehetőség van grafikus ábrák megjelenítésére derékszögű koordinátarendszerben,

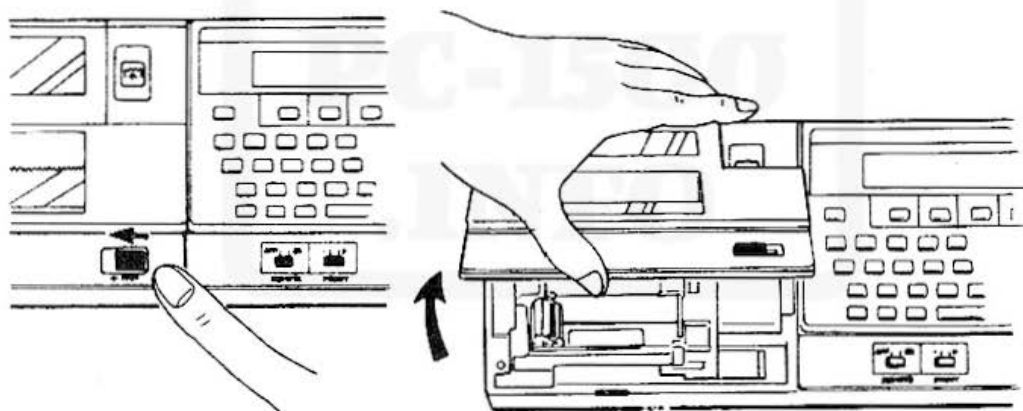
- a papírmozgatás lehet kézi vagy programozott.

A nyomtatóval számos hasznos információ jeleníthető meg:

- programlisták,
- adatlisták,
- programok számolási eredményei,
- manuális számolási eredmények,
- különböző grafikák
- függvénygörbék

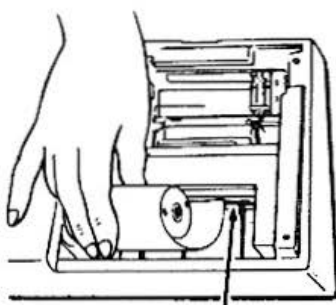
5.4.1. A papírtekercs cseréje

1. Reteszeljük ki és távolítsuk el a nyomtató fedelét. Ehhez a nyomtató fedelét lévő OPEN feliratú rögzítőt toljuk el balra ütközésig. Ekkor a fedőlap levehető.




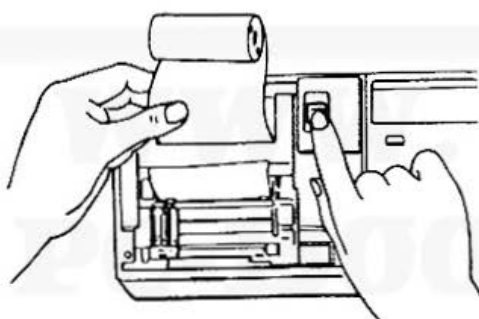
5.7. ábra

2. Egyengessük ki a papírtekercs elejét és helyezzük be a vezetőnyílásba. Ha a papír szakadt, a sérült részt távolítsuk el, mert a nyomtató károsodásához vezethet. Helyezzük a papírszalag végét a bevezetőnyílásba.



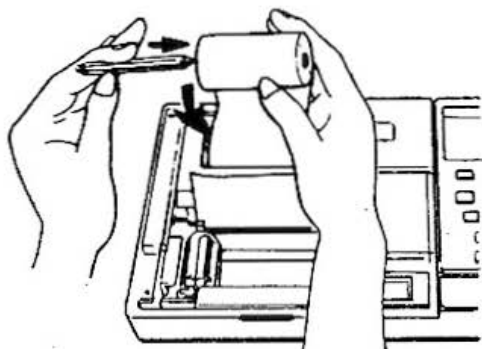
5.8. ábra

3. Ha a PTA-4000 nem volt bekapcsolva, kapcsoljuk be és nyomjuk le a papírtovábbító  billentyűt, míg a papír kb. 6 cm-t tovább nem gördül.



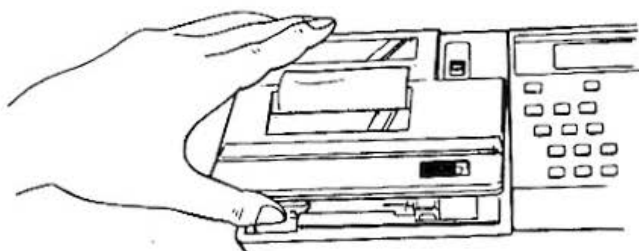
5.9. ábra

4. Dugjuk a papírtartó tengelyt a tekercsbe és helyezzük az egészet a papírtartóba.



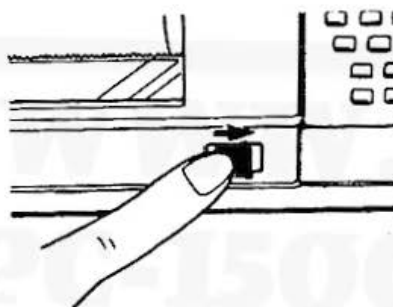
5.10. ábra

5. A papírtekerces elejét vezessük át a nyomtató fedelén lévő résen és tegyük vissza a fedelet.



5.11. ábra

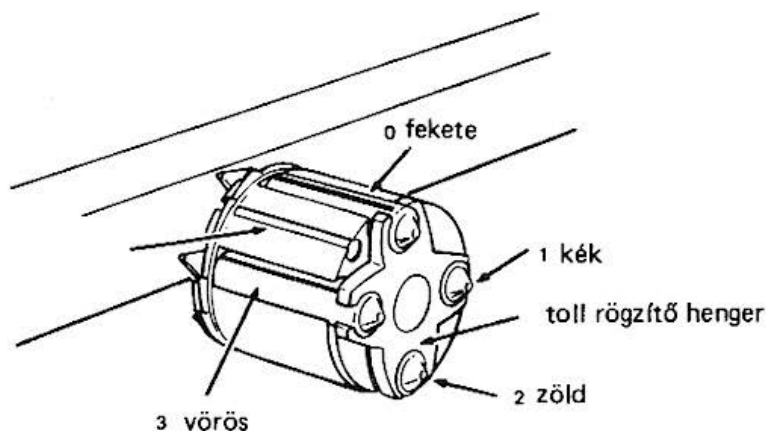
6. Rögzítsük a nyomtató fedelét.



5.12. ábra



5.4.2. A színes tollak cseréje

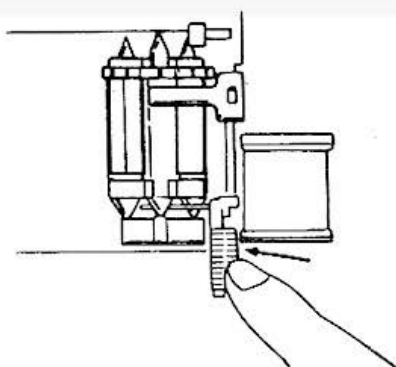
A nyomtató négy féle színű tollal működtethető.
A tollakat a nyomtató fedelének eltávolítása után lehet cserélni.



5.13. ábra

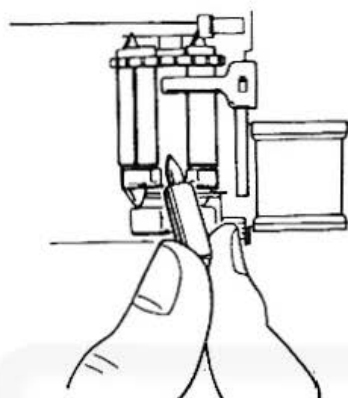
A tollcsere lépései:

1. Távolítsuk el a nyomtató fedelét.
2. Nyomjuk le a papírtovábbító  billentyűt, miközben a nulla  billentyűt is lenyomva tartjuk. Ekkor a tollrögzítő henger egy negyedfordulatot elfordul és a nyomtató jobb szélére áll. Azt a tollat hozza felső helyzetbe, amelynek cseréje soron következik.
3. A felül lévő tollat az eltávolító karral lehet kimozdítani a helyéről.





5.14. ábra

4. Az így kivett toll helyére új toll helyezhető. Ügyeljünk arra, hogy a toll hegye a lyuktárcsa felső lyukába kerüljön. A toll végén kifejtett könnyű nyomásra a toll biztonságosan elhelyezhető a helyén.



5.15. ábra

5. Nyomjuk be a papírtovábbító billentyűt. A toll ekkor balra egy negyedfordulatot elfordul, majd visszafut a nyomtató jobb szélére. Ezután végezzük el a 3. pontnál leírtakat, amíg minden tollat ki nem cseréltünk.
6. A tollcsere befejezése után nyomjuk le a papírtovábbító  billentyűt, miközben a  billentyűt lenyomva tartjuk. Erre a tollrögzítő henger a nyomtató bal szélére fut és alaphelyzetbe áll.
7. Végül tegyük vissza a nyomtató fedelét.

Figyelem !

1. A nyomtatót ne használjuk úgy, hogy nincs minden toll behelyezve, mert ez téves tollváltást eredményezhet!
2. Ügyeljünk arra, hogy minden toll a saját helyére kerüljön, ha a tollat beleejtenénk a nyomtatóba, ne nyúljunk bele kézzel, hanem a gép megfordításával távolítsuk el.
3. Ha a nyomtatót nem használjuk, vegyük ki belőle a tollakat és zárókupakkal lezárva helyezzük vissza azokat a tárolócsőbe, mert a tollak könnyen beszáradhatnak.

5.4.3. A kézi számítások eredményeinek kinyomtatása

A KA 160 bővítővel lehetőség van arra, hogy a számítógépen végzett számítási eredményeket kinyomtassuk. Ehhez a bővítőn lévő . és P kapcsolót kell P állásba állítani. Ekkor a nyomtató minden beadott információt rögzít, kivéve az ENTER-t.

Ha az ily módon való nyomtatást le akarjuk tiltani, a fenti kapcsolót . pozícióba kell állítani. Ekkor a nyomtatóra csak LPRINT paranccsal (vagy utasítással) lehet íratni. A nyomtatás mindig az utoljára beállított színnel történik. A gép bekapcsolása után a fekete szín kerül írási pozícióba. Színt választani a COLOR paranccsal (vagy utasítással) lehet (lásd ott!).

A nyomtatónak két üzemmódja van:

1. TEXT, ami "írás" üzemmódot jelent. A kapcsoló P állásba állítása automatikusan TEXT üzemmódba helyezi a nyomtatót. Lásd még TEXT utasításnál.
2. GRAPH, ami a "rajzolás" üzemmódot jelenti. Ha a kapcsoló P-be való állítása GRAPH üzemmódban történt, a . -ba való visszakapcsolás nem állítja vissza a GRAPH üzemmódot. Erre külön GRAPH parancsot kell kiadni.

5.4.4. A nyomtatót vezérlő parancsok, utasítások

COLOR

szintaxisa: sorszám COLOR n

ahol n numerikus konstans változó, vagy kifejezés lehet.

Ez a parancs (vagy utasítás) a következő híváshoz beállítja a kívánt színű tollat. Az n értéke 0, 1, 2, 3 lehet, amely rendre a fekete, kék, zöld, piros tollat választja ki.

A gép bekapcsolása után a fekete tollal lehet írni. TEXT üzemmódban a COLOR parancs (vagy utasítás) után a toll a papír bal szélére áll, GRAPH üzemmódban a színváltás után a toll a váltást megelőző rajzolási pozícióba tér vissza.

Példa:

Ez a program a négy színnel kiírja a megfelelő színek neveit:

```

1Ø DATA "FEKETE", "KEK", "ZOLD", "PIROS"
2Ø FOR X=Ø TO 3
3Ø COLOR X
4Ø READ SZØ
5Ø LPRINT SZØ
6Ø NEXT X

```

CSIZE

szintaxisa: sorszám CSIZE n

ahol n numerikus konstans, változó vagy kifejezés lehet.

Ez a parancs (vagy utasítás) a jelméretet határozza meg. Az értéke 1 és 9 közé eshet.

A CSIZE egyaránt hatásos TEST és GRAPH üzemmódban is. Az n értéke és a méretek közötti összefüggést a következő táblázat mutatja:

CSIZE	1	2	3	4	5	6	7	8	9
Karakter- szám soronként	36	18	12	9	7	6	5	4	4
Karakter magasság (mm)	1,2	2,4	3,6	4,8	6,0	7,2	8,4	9,6	10,8
Karakter szélesség (mm)	0,8	1,6	2,4	3,2	4,0	4,8	5,6	6,4	7,2

Példa:

Ez a program a 9-féle jelméretben kiírja a BUBU szót.

```

1Ø    FOR X=1 TO 9
2Ø    CSIZE X
3Ø    LF 2
4Ø    LPRINT "BUBU"
5Ø    NEXT X

```

GLCURSOR

szintaxis: sorszám GLCURSOR (n1, n2)

ahol n1 és n2 numerikus konstans, változó vagy kifejezés lehet.

Csak GRAPH üzemmódban használható.

Ez a parancs (vagy utasítás) tetszőleges (X, Y) koordinátájú pontra állítja a tollat, de nem rajzol vonalat. Az n1, az X, az n2 az Y koordinátára utal.

Ha a kívánt pont kívül esik a megengedett határon ($-2048 > n$ vagy $n > 2047$), akkor hibajelzést kapunk. Lásd még a GRAPH utasításnál.

Példa:

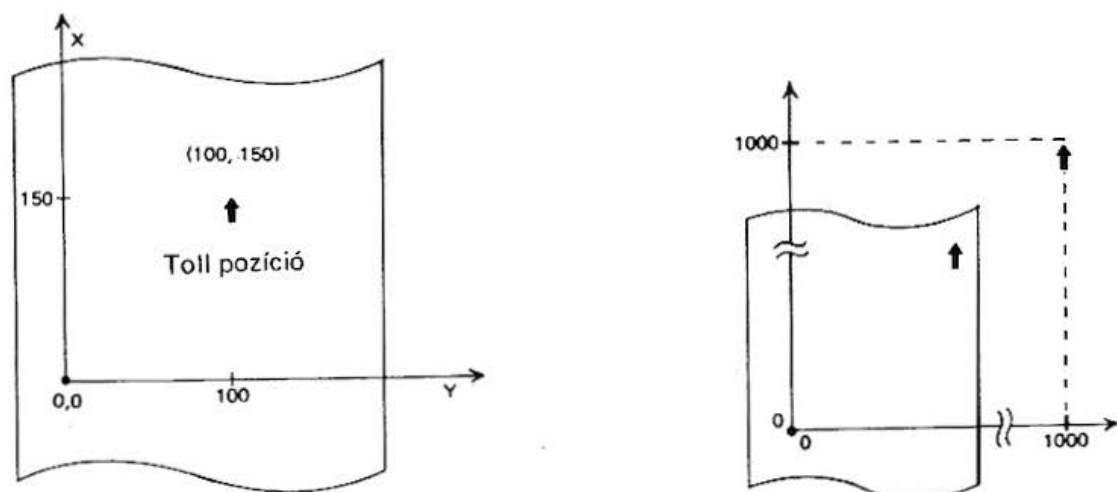
Ez a program $0^\circ - 180^\circ$ intervallumban megrajzolja a sinus függvényt.

```

1Ø    GRAD
2Ø    GRAPH
3Ø    FOR X=1 TO 200 STEP 2
4Ø    GLCURSOR (X, SIN(X)*100)
5Ø    LPRINT " ."
6Ø    NEXT X

```

A toll pozíció beállítása



5.16. ábra

GRAPH

szintaxis: **sorszám** GRAPH

Ez a parancs (vagy utasítás) a "rajzolás" üzemmód beállítására szolgál. Hatására a toll a papír bal szélére megy.

Ebben az üzemmódban a nyomtató X-Y derékszögű koordináta rendszert használ. A csak GRAPH üzemmódban használható utasítások:

SORGN
GLCURSOR
LINE
RLINE
ROTATE

GRAPH üzemmódban a tollmozgató utasítások mindkét argumentumában -2048-tól 2047-ig terjedő számok adhatók meg, azonban a toll ilyen tartományban nem tud mozogni. Vízszintesen 217, függőlegesen -510

és 2047 tartományban mozog a toll. Ha a rajzolni kívánt pont kívül esik ezen a tartományon, a toll a papír szélénél megáll, a gép azonban a tollvezérlés szempontjából \pm 2047 tartományon belül számon tartja a képzeletbeli tollpozíciót.

LCURSOR

szintaxisa: sorszám LCURSOR n

ahol n numerikus konstans, változó, vagy kifejezés lehet.

Csak TEXT üzemmódban használható.

Ezzel a paranccsal (vagy utasítással) a tollat a sor egy meghatározott helyzetébe állíthatjuk.

A tollhelyzet, ahová a toll kerül, az adott karaktermérettől függ. Értéke \emptyset és (K-1) közé eshet, ahol K az adott méretnél, soronként megengedett maximális karakterszám. (Lásd a CSIZE utasításnál lévő táblázatot)

LF

szintaxisa: sorszám LF n

ahol n numerikus konstans, változó vagy kifejezés lehet.

Csak TEXT üzemmódban használható.

Az LF parancs (vagy utasítás) vezérli a papírmozgatást. Az n numerikus érték határozza meg a mozgatás mértékét (hogyan hány sorral mozgatjuk) és irányát (ha negatív, akkor vissza, ha pozitív, akkor előre). A mozgatás mértéke természetesen a CSIZE-vel beállított karaktermérettől is függ.

A vissza történő mozgatásnál maximum kb. 10 cm tekerhető vissza. Ha ennél nagyobb mértékben mozgatnánk, akkor az hibajelzést eredményez.

LINE

szintaxisa: sorszám LINE (X1, Y1)-(X2, Y2), n, m
ahol X1, Y1, X2, Y2 n, m numerikus konstans, változó, vagy kifejezés lehet.

Csak GRAPH üzemmódban használható.

Ez a parancs (vagy utasítás) az (X1, Y1) koordinátájú pontból, az (X2, Y2) koordinátájú pontba, az n vonaltípussal és az m színnel vonalat rajzol.

A pontok koordinátáinak a megadott intervallumon belül kell esniük ($-2048 \leq X, Y \leq 2047$).

A vonaltípus és a szín megadása nem kötelező.

Ha a kiindulási koordináták nem voltak megadva, akkor a toll pillanatnyi helyzetétől kezdve történik a rajzolás.

Az n értéke 0 és 9 közé, az m értéke 0 és 3 közé eshet.

A vonaltípus a következő lehet:

0	folytonos vonal	0	_____
1	0,4 mm szaggatott vonal	1
2	0,6 mm szaggatott vonal	2
3	0,8 mm szaggatott vonal	3
4	1,0 mm szaggatott vonal	4
5	1,2 mm szaggatott vonal	5
6	1,4 mm szaggatott vonal	6
7	1,6 mm szaggatott vonal	7
8	1,8 mm szaggatott vonal	8
9	"toll fel" (nincs vonal)	9	

A LINE-nak még további alakjai vannak:

1. LINE (X1,Y1)-(X2,Y2) n, m, B
ahol B az utasítás végén azt jelenti, hogy az (X1,Y1)-(X2,Y2) egyenes egy derékszögű négyszög átlója, és ezzel az utasítással egy (X1,Y1), (X2,Y1), (X2,Y2), (X1,Y2) koordinátájú csúcsokkal rendelkező derékszögű négyszöget rajzol a gép.
2. LINE (X1,Y1)-(X2,Y2)- ... - (X5,Y5)-(X6,Y6), n,m
Ezzel lehetőség van több egyenes szakasz (maximum 5) megrajzolására egyetlen LINE-val.
(Lásd 5.17. ábra)

Példák:

1. Ez az utasítás egy derékszögű négyszöget rajzol:
10 LINE (10, 10)-(50,50), 0, 0, B
2. Ez a program egy félkört rajzol a koordináta-rendszerben:
10 J=1 : R=100 : GRAPH
20 GLCURSOR (110, 0) : SORGN
30 FOR I=0 TO 2*R STEP J
40 X=R-I
50 Y=SQR (R*R-X*X)
60 LINE - (X,Y), 0
70 NEXT I
80 LINE -(0,0)
90 LINE (0,0)-(0,80), 1
100 LINE (0,0)-(0,-30), 1

LLIST

Ez az utasítás (vagy parancs) a memóriában lévő program egészének, vagy egy részének kinyomtatására

szolgál.

A LLIST után meg lehet adni a kinyomtatni kívánt utasítás (vagy utasítások) sorszámát (sorszámait) vagy címkéjét (címkéit).

Az LLIST lehetséges alakjai:

- LLIST - a teljes program kinyomtatása,
- LLIST n - az n sorszámú (vagy címkéjű) sor kinyomtatása,
- LLIST n, - az n sorszámú (vagy címkéjű) sortól a végéig kinyomtatja a programot,
- LLIST, n - az első sortól az n címkéjű (sorszámú) sorig kinyomtatja a programot,
- LLIST n1, n2 - az n1 sorszámú (címkéjű) sortól az n2 sorszámú (címkéjű) sorig kinyomtatja a programot.

Az LLIST a nyomtatót automatikusan TEXT üzemmódba állítja. Ha az LLIST kiadása GRAPH üzemmódban történt, a GRAPH-ba való visszahíváshoz külön GRAPH parancsra van szükség.

LPRINT

szintaxisa: sorszám LPRINT n1, n2, ...

ahol n1, n2, ... numerikus és/vagy karakteres konstans, változó, és/vagy kifejezés lehet.

Ezzel a paranccsal (vagy utasítással) tetszőleges numerikus és/vagy karakteres érték nyomtatható ki. Az LPRINT-re ugyanaz vonatkozik, mint a PRINT-re, csak a program futását nem állítja le. Az LPRINT önmagában használható TEXT és GRAPH üzemmódban.

Ha az LPRINT-et TEXT üzemmódban használjuk, az egy sort emel és a tollat a papír bal szélére viszi. Ha GRAPH-ban használjuk, akkor a koordinátarendszer origóját az új tollpozícióba helyezi. Az LPRINT-tel használható a PRINT-nél megismert USING szerkezet is, de csak GRAPH üzemmódban.

RLINE

Csak GRAPH üzemmódban használható.

szintaxisa: megegyezik a LINE szintaxisával:

RLINE (X1,Y1)-(X2,Y2) n, m
X, Y, m, n mint LINE utasításnál.

Ezzel a paranccsal (vagy utasítással) olyan vonalak rajzolhatók, amelyek olyan koordináta rendszerre vonatkoznak, melyek kezdete a pillanatnyi toll pozíció. Így a SORGN-nal (lásd ott!) definiált abszolút koordinátarendszer érintetlen marad.

Példa:

A most következő három program ugyanazt az ábrát rajzolja, különböző módszerrel.

```
1Ø    GRAPH
2Ø    LINE-(5Ø,5Ø)-(1ØØ,-5Ø)-(5Ø,-1ØØ)
3Ø    END

1Ø    GRAPH
2Ø    SORGN
3Ø    LINE-(5Ø,5Ø)
4Ø    SORGN
5Ø    LINE-(5Ø,-1ØØ)
6Ø    SORGN
```

```

7Ø    LINE-(-5Ø,-5Ø)
8Ø    END

1Ø    GRAPH
2Ø    RLINE-(5Ø,5Ø)-(5Ø,-1ØØ)-(-5Ø,-5Ø)
3Ø    END

```

ROTATE

szintaxis: sorszám ROTATE n

ahol n numerikus konstans, változó, vagy kifejezés lehet.

Csak GRAPH üzemmódban használható.

A ROTATE utasítással lehetőség van megváltoztatni a nyomtatás irányát. Az n lehetséges értékei: Ø, 1, 2, 3. Az 5.19. ábrán az n értékéhez tartozó nyomtatási irányokat láthatjuk.

Példa:

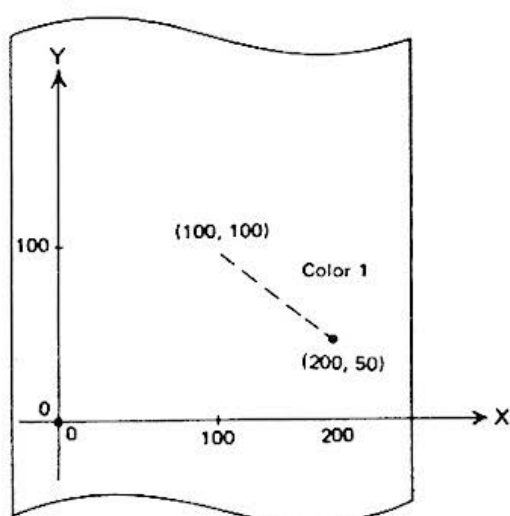
Ez a program bemutatja a ROTATE hatását:

```

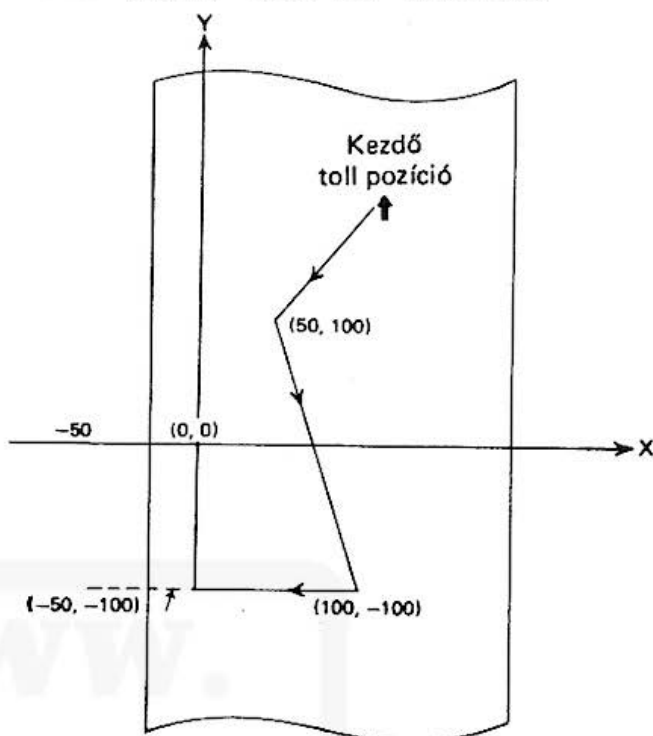
1Ø    GRAPH
2Ø    GLCURSOR (1ØØ,Ø)
3Ø    FOR X=Ø TO 3
4Ø    ROTATE X
5Ø    LPRINT "PTA-4000"
6Ø    NEXT X

```

LINE (100, 100) - (200, 50), 2, 1

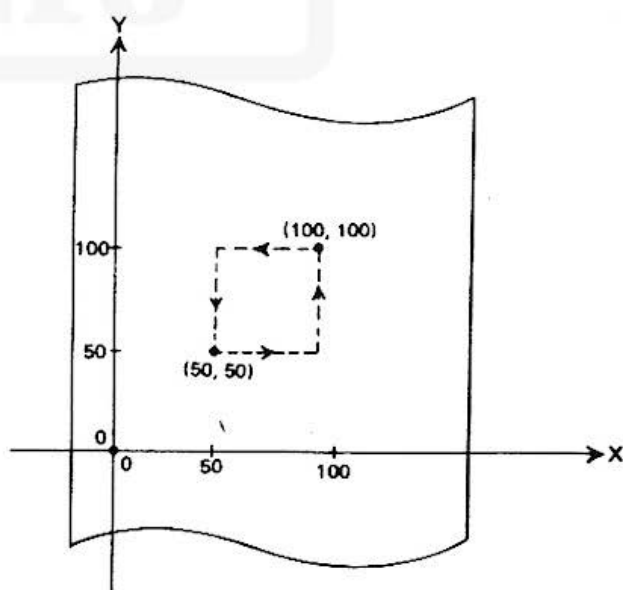
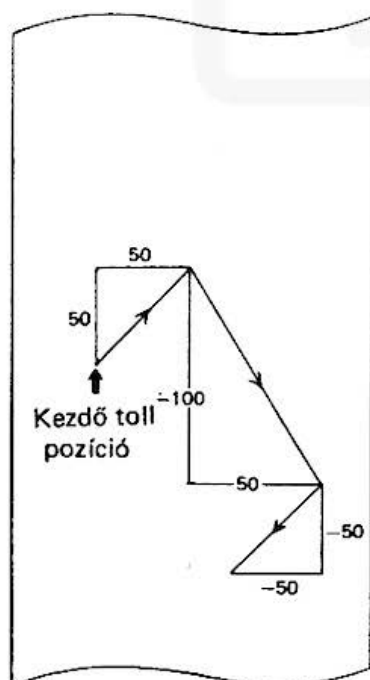


LINE - (50, 100) - (100, -100) - (-50, -100)



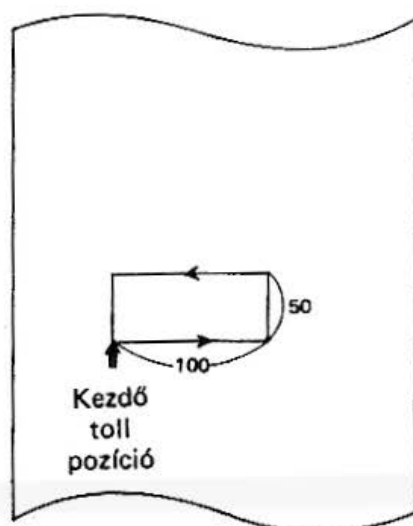
RLINE - (50, 50) - (50, -100) - (-50, -50)

LINE (50, 50) - (100, 100), 2, , B

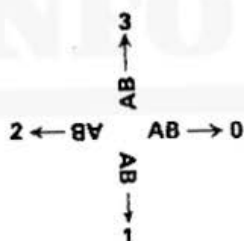


5.17. ábra

RLINE – (100, 50), , , B



5.18. ábra



5.19. ábra

SORGN

szintaxisa: sorszám SORGN

Csak GRAPH üzemmódban használható.

Mint tudjuk, a nyomtató GRAPH üzemmódban, derékszögű koordinátarendszerben rajzol. A SORGN utasítás hatására a toll pillanatnyi helyzete vagy képletbeli pozíciója válik a koordinátarendszer kezdőpontjává (az origóvá). A koordinátarendszer kezdőpontjává olyan pont is választható, mely nincs rajta a papíron (lásd a GRAPH üzemmódnál).

TAB

szintaxisa: sorszám LPRINT TAB n, n1, n2 ...

ahol n numerikus konstans, változó, vagy kifejezés lehet, n1, n2, ... pedig amit ki akarunk írni (numerikus vagy karakteres változó, konstans, vagy kifejezés).

Csak TEXT üzemmódban használható.

A TAB hatása azonos az LCURSOR-éval, azzal a különbséggel, hogy a TAB-ot az LPRINT-en belül lehet használni. Az n mondja meg, hogy hányadik karakterpozíciótól kell kezdeni a nyomtatást. Értéke a karaktermérettől függ $0 \leq n \leq (k-1)$, ahol k = a karaktermérettől függő soronkénti maximális karakterszám. (Lásd a CSIZE utasításnál lévő táblázatot). Figyeljünk arra, hogy az LPRINT GRAPH és TEXT üzemmódban is hatásos, az LPRINT TAB utasítás csak TEXT üzemmódban hatásos.

TEST

szintaxisa: sorszám TEST

Ez a parancs (vagy utasítás) a printer tesztelésére szolgál. Hatására 4 db különböző színű négyzetet rajzol a nyomtató.

TEXT

szintaxisa: sorszám TEXT

Ez a parancs (vagy utasítás) a nyomtatót "írás" üzemmódba állítja.

5.5. A magnetofonillesztő használata

A KA160 bővítőn keresztül a PTA 4000-hez egyidejűleg két kazettás magnetofon csatlakoztatható, mint külső program- vagy adattároló.

A kazettás magnetofonnal a következő tevékenységek valósíthatók meg:

- programok manuális tárolása,
- programok manuális és ellenőrzött visszatöltése,
- adatok manuális és ellenőrzött tárolása illetve visszatöltése,
- két vagy több program manuális visszatöltése.

Az adatok illetve programok problémamentes rögzítésére és visszatöltésére egy új magnetofon beszerzése ajánlatos. Erre a célra kiválóan alkalmasak a mono magnetofonok.

A magnetofon lehetőleg rendelkezzen a következőkkel:

- számlálóval
- mikrofon be- és fülhallgató kimenettel

- távvezérelhető pillanat Stop csatlakozóval
A magnetofon jellemzőit lásd az 5.3. fejezetben!

Megjegyzés:

1. A magnetofont lehetőleg csak a számítógéphez használjuk!
2. Csak minőségi kazettákat használjunk!
(Pl.: BASF compusitte, VERBATIM Data Casette, SONY C60 vagy C90, stb.)

5.5.1. A kazettás magnetofon csatlakoztatása a KA160-hoz

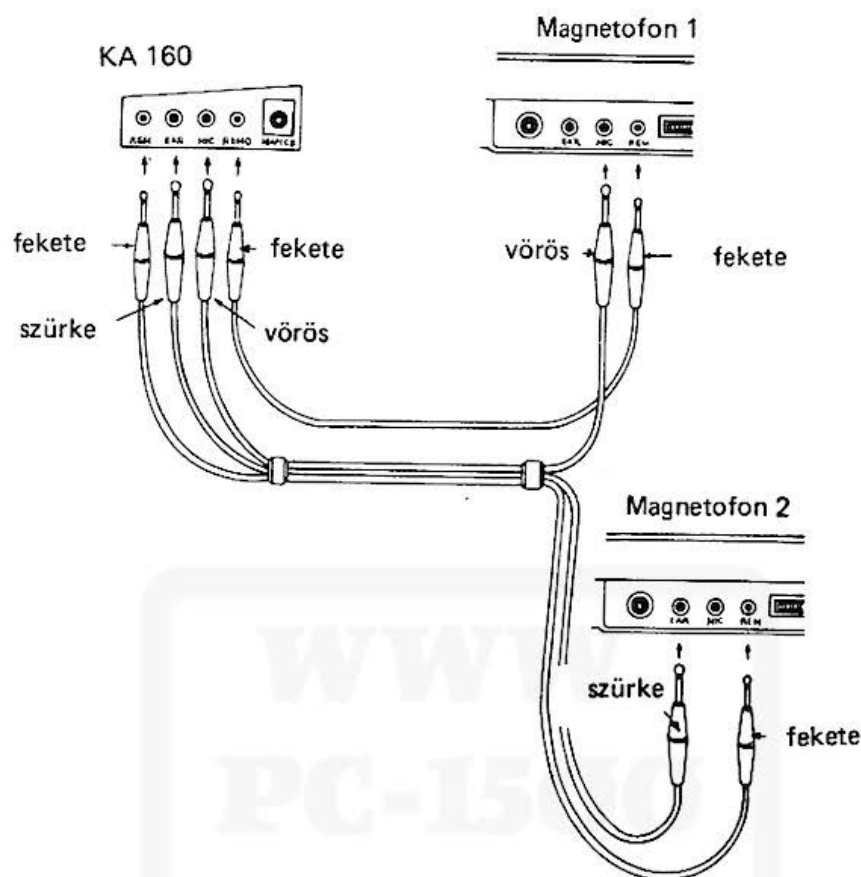
1. Egy magnetofon csatlakoztatása

1. Kapcsoljuk ki a PTA-4000-et.
2. A géphez adott háromágú jack-dugós vezetékkel értelemszerűen kössük össze a KA160 jobb oldalán lévő EAR, MIC illetve RMØ feliratú kimeneteket, a magnetofonon lévő megfelelő bemenetekkel. Lásd az 5.5. ábrát.
3. Ha ez megtörtént, állítsuk a KA 160 REMOTE feliratú kapcsolóját ON-ra, amely letiltja a magnetofon összes szalagmozgató funkcióját.
4. Ha a szalagot mozgatni akarjuk, a REMOTE kapcsolót állítsuk OFF-ra.
(Az ábrát lásd a KA 160 csatlakoztatása magnetofonhoz c. fejezetben.)

2. Két magnetofon csatlakoztatása

Ekkor az első magnetofont felvételre, a másodikat lejátszásra állítsuk. A vezetékek dugaszol-

lását az ábrán láthatjuk.



5.20. ábra

3. Adattárolás mágnesszalagon

A szalagon adatok, programok tárolhatók és visszanyerhetők. A kazetta típusától függően, egy-egy kazettán nagyon sok információ tárolható. A számítógép a szalagra a következőket viszi fel:

Szignál hang	Programnév vagy adatnév	Program vagy adat	File vége jel
-----------------	-------------------------------	-------------------	------------------

A program vagy adat neve max. 16 karakteres, tetszőleges jelsorozat lehet. A szalagon tárolt információk csak szekvenciálisan érhetők el. A könnyebb visszakeresés érdekében érdemes használni a magnetofon számlálóját.

4. A mágnesszalag használata

Ha a gépet összekapcsoltuk a magnetofonnal, a felvételt, az ellenőrzést vagy a betöltést a következőképpen kell elvégezni:

A program felvétele:

- a program a számítógép memóriájában van, és a magnetofont az 5.5. ábra szerint csatlakoztatuk a KA 160-hoz.
- Ha a magnetofon rendelkezik távvezérelhető pillanat állj lehetőséggel, kapcsoljuk a REMOTE kapcsolót OFF állásba.
- Keressünk a szalagon üres helyet. Ügyeljünk arra, hogy a szalagot túltekercseljük a befűzőn. Jegyezzük fel a számláló állását.
- Ha a magnetofon automatikus felvételi szintszabályozási lehetőséggel rendelkezik, úgy használjuk azt, ha nem, úgy készítsünk próbafelvételt, állítsuk be a felvételi szintszabályozót a magnó kezelési utasításának megfelelően.
- Kapcsoljuk a REMOTE kapcsolót ON állásba.
- Kapcsoljuk a magnót felvétel állásba. Ha nincs távvezérelhető pillanat állj, a pillanat állj kapcsolót kapcsoljuk be (ne indítsuk el a magnót).

- Billentyűzzük be a következő utasítást:

CSAVE **SHIFT** "PROG1 **SHIFT** "

A CSAVE a rögzítsd utasítás, a PROG1 a program neve. A program nevét idézőjelek közé kell tenni. Ha a magnón nincs távvezérelhető pillanat állj, kapcsoljuk ki a pillanat állj kapcsolót (indítsuk el a magnót), majd nyomjuk be az **ENTER** gombot. Távvezérelhető magnónál az **ENTER** megnyomása után azonnal indul a felvétel. Működés közben a kijelző felső részén a BUSY felirat látható, és a felvételre jellemző hang hallható. Ha a kijelzőn újra a > jel látszik, a felvétel véget ért, ekkor a magnó megáll vagy megállítható.

A kazettán lévő program ellenőrzéséhez a CLOAD? **ENTER** parancsot használjuk. Ha a felvétel jól sikerült, a program neve, ha hiba van, ERROR 43 jelenik meg a kijelzőn. Ekkor a memóriában tárolt és a szalagon lévő program eltér. Próbáljuk meg az ellenőrzést kisebb vagy nagyobb hangerővel. Ha ez sem vezet eredményre, a felvételt meg kell ismételni. Az új felvételnél a hangerőt változtassuk meg (növeljük). Ne feledkezzünk el a felvétel ismétlésekor a szalag előzetes visszatekeréséről. Távvezérelt pillanat állj esetén a REMOTE kapcsoló OFF állásában tudjuk ezt elvégezni.

A program szalagról való visszatöltésekor a következőképpen járunk el:

- a REMOTE kapcsolót kapcsoljuk OFF állásba.
- tekerjük a kazettát a program elejére, ehhez használjuk a magnetofon számlálóját.

- állítsuk le a visszatekereslést a program kezdeténél.
- a REMOTE kapcsolót állítsuk ON állásba.
- a "lejátszás" billentyűjét nyomjuk le, majd billentyűzzünk:

CLOAD **SHIFT** " PROG1 **SHIFT** " **ENTER**

A PROG1 a felvett program azonosító neve.

A kézi "pillanat állj" kapcsoló esetén az **ENTER** benyomása után indítsuk a magnetofont, a távvezérelt "pillanat állj" esetén az **ENTER** benyomására a magnetofon elindul.

A beolvasás alatt a BUSY felirat látható, befejezésekor eltűnik. Ezután ellenőrizzük RUN vagy LIST paranccsal, hogy valóban betöltődött-e a program.

Ha ERROR43 vagy ERROR44 jelzést kapunk, a program betöltését ismételjük meg.

Az eddigiekben CSAVE és CLOAD utasításokat a program rögzítésére illetve betöltésére használtuk. Egyetlen változó vagy egy változó csoport értékét a PRINT # és INPUT # utasítással rögzíthetjük. Ez szükséges lehet akkor, ha egy program eredményeit egy másik programban akarjuk felhasználni kézi bevitel nélkül. Használatukat lásd az 5.5.2. fejezetben.

5.5.2. Magnetofon vezérlő utasítások

CHAIN

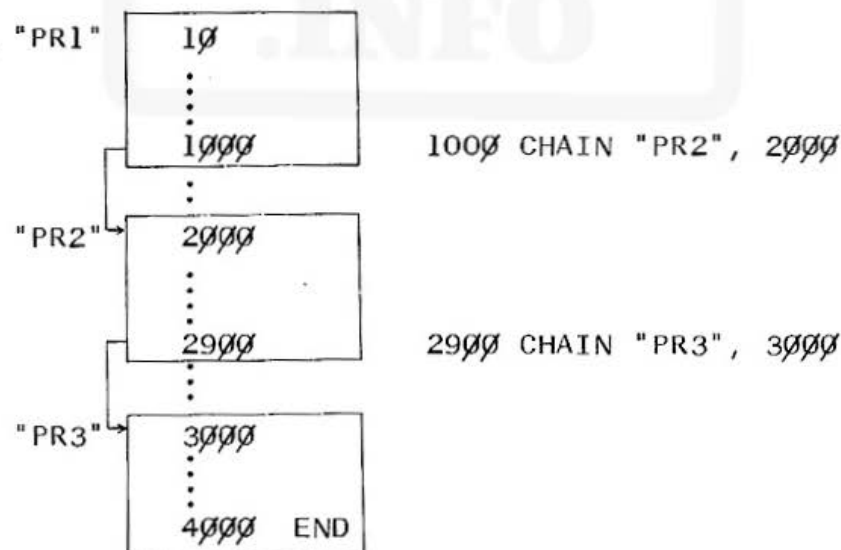
szintaxisa: sorszám CHAIN "programnév", sorszám 1, ahol a programnév max. 16 karakteres jelsorozat, a sorszám 1 pedig annak az utasításnak a sorszáma, ahol a beolvasott programrészt folytatni kell.

A CHAIN utasítás lehetővé teszi, hogy a memóriához képest túl nagy programokat részenként végrehajtsunk. Emiatt programunkat önállóan is működőképessé kell tenni, és külön-külön, CSAVE paranccsal (lásd ott!) fel kell venni a szalagra, egymás után, a végrehajtás sorrendjében.

A CHAIN csak utasítás lehet, parancs nem!

A CHAIN használatához a magnetofont a mágnesszalag használata című fejezetben leírtaknak megfelelően kell előkészíteni.

Példa:



Tegyük fel, hogy programunk három működőképes részből áll, melyek nevei rendre PR1, PR2 és PR3.

Először CLOAD "PR1" paranccsal beolvassuk az első programot, amelyet elindítunk. Mikor a program elér az 1000 sorszámú utasításra, a CHAIN utasítás törli a programtárat és elindítja a "PR2" nevű program betöltését, majd a betöltés után elindítja a programot a 2000 sorszámú utasítással. Mikor a vezérlés a 2900 sorszámú utasításra kerül, a CHAIN utasítás törli a programtárat, betölti a "PR3" nevű programot, majd elindítja azt a 3000 sorszámú utasítással. A programrész, s így az egész program a 4000 END utasítással fejeződik be.

CLOAD

szintaxisa: CLOAD "programnév"

ahol a programnév max. 16 karakteres jelsorozat lehet, mely elhagyható.

Ha ezt a parancsot RUN vagy PRO üzemmódban kiadjuk, a kazettán tárolt programot olvashatjuk be a memóriába. Ha a programnevet nem adjuk meg, akkor az első megtalált program kerül beolvasásra. Ha megadjuk a program nevét, akkor a gép megkeresi az adott nevű programot a szalagon, és ha megtalálja, beolvassa. Ha a programot nem találja meg, a kazetta a végére ér, s ekkor a műveletet BREAK -kel kell megszakítani.

Ha a parancsot RESERVE üzemmódban adjuk ki, akkor lehetőség van a korábban kimentett, programozott RESERVE kifejezések beolvasására.

Miközben a betöltés folyik, a kijelzőn az éppen beolvasás alatt álló program, vagy RESERVE kifejezés

neve látható. (Lásd A mágnesszalag használata című fejezetben).

CLOAD?

szintaxisa: CLOAD? "programnév"

ahol a programnév max. 16 karakteres jelsorozat, mely elhagyható.

A CLOAD? paranccsal a kazettára kimentett programok, vagy RESERVE kifejezések hasonlíthatók össze a memóriában lévő eredeti programmal, vagy RESERVE kifejezéssel.

A CLOAD? lefutása azonos a CLOAD-éval. Ha az összehasonlítás során hibát talál, a kijelzőn ERROR 43 hibajelzés lesz látható, különben megjelenik a " > " készenléti szimbólum.

CSAVE

szintaxisa: sorszám CSAVE n

ahol n karakteres konstans, változó vagy kifejezés lehet.

Ezzel a paranccsal (vagy utasítással) lehet kazettára felvenni a programokat, vagy RESERVE kifejezéseket. Ez utóbbi esetben RESERVE üzemmódba kell helyezni a számítógépet. Ha programot veszünk fel, az üzemmód lehet PRO vagy RUN. Ha egy új programot a szalagra úgy mentünk ki, hogy ezzel közben letöröljük egy, már szalagon lévő program egy részét, akkor ezt az utóbbit már nem tudjuk többé betölteni. Ha két felvett program neve megegyezik, akkor min-

dig azt fogja beolvasni, amelyiket előbb megtalálja.

Példa:

```

:
5000 B$="PELDA PROGRAM"
10000 INPUT "FELVETEL MEHET? (I,N)"; I$
1010 IF I$="I" THEN GOTO 10000
1020 CSAVE B$
:

```

INPUT

szintaxis: sorszám INPUT # "név"; adatlista

ahol a "név" a kimentendő adatok közös neve, mely elhagyható, az adatlista numerikus és/vagy karakteres konstans, változó lehet, egymástól vesszővel elválasztva.

Ezzel a paranccsal (vagy utasítással) lehetőség van a PRINT # -tel a szalagra kimentett adatokat (lásd ott!) a megfelelő változókba beolvasni.

Ha az adatokat közös változónévvel láttuk el a PRINT # utasításban (pl.: PRINT # "ADAT"; A,B,C), akkor ezek így kerülnek föl a szalagra, és visszaolvasáskor az ilyen névvel ellátott adatok kerülnek beolvasásra. Ekkor az adatok közös neve a kijelzőn látszik. Ha a nevet nem adjuk meg, akkor az első talált adatcsoport kerül beolvasásra.

Ha a változók száma az INPUT # és a megfelelő PRINT # között különböző, akkor

- ha az INPUT # -ban több változó volt megadva, mint a PRINT # -ben, akkor a maradék numerikus

változók nullával, a karakteres változók space-vel lesznek feltöltve.

- ha a PRINT # -ben több változó volt megadva, mint az INPUT # -ban, akkor a többletváltozók nem kerülnek beolvasásra.

Ha a szalagra tömböt akarunk kimenteni, illetve szalagról beolvasni, azt a következőképpen kell megadni:

kimentés: PRINT # "név"; tömbnév (*)

beolvasás: INPUT # "név"; tömbnév (*)

ahol a "név" az adatmező neve.

Pl.: az INPUT "DATA"; A,B,C(*),D\$(*) esetén az adatmező neve "DATA", beolvasásra kerülnek az A, B numerikus változók, a C összes változata, tehát C, C(0), C(1), C(2), ... stb., valamint a D\$ összes változata, tehát D\$, D\$(0), D\$(1), D\$(2), ... stb. A magnót a beolvasáshoz elő kell készíteni! (Lásd a magnetofon használata című fejezetben leírtakat.)

MERGE

szintaxisa: MERGE "programnév"

ahol a programnév a beolvasandó program neve.

Ez a parancs lehetővé teszi, hogy a memóriába egyidejűleg több program legyen betöltve. A MERGE paranccsal kazettáról úgy tölthető be a program, hogy az nem törli a memóriában lévő programsorokat még akkor sem, ha így azonos sorszámmal, több utasítássor is van egyidőben a memóriában. Ez lehetőséget ad arra, hogy a szalagról betöltött program bizonyos részeit felhasználjuk egy új program

létrehozására. Az önálló programrészek első sorait címkével kell ellátni. A program végrehajtásánál ugrások GOTO-val, csak ugyanabban a programban lehetségesek. Egyik programból a másikba csak a GOTO "címke"-vel lehet átugrani, ahol a címke az önálló program első sorának címkéje. Ha a végrehajtás során elérjük a program végét, a program futtatása automatikusan befejeződik.

A MERGE használata előtt a magnót elő kell készíteni a betöltésre (lásd a mágnesszalag használata című fejezetben).

Példa:

Tegyük fel, hogy adott a következő három programunk:

```
10 "PR1":FOR X=0 TO 255
20 PAUSE X; " CHR$(X)
30 NEXT X
```

```
10 "PR2":DIM A(100)
20 FOR Y=0 TO 100
30 A(Y)+Y
```

```
10 "PR3":FOR T=0 TO 100
200 PAUSE A(T)
300 NEXT T
```

Töltsük be az első programot CLOAD "PR1" paranccsal, majd töltsük mellé MERGE "PR2" és MERGE "PR3" parancsokkal a másik két programot.

Az első program RUN-nal elindítható. Végrehajtása után a második programot vagy GOTO "PR2"-vel,

vagy az első programba előzetesen elhelyezett 40 sorszámú GOTO "PR2" utasítással lehet elindítani. Ha ez a program is végrehajtásra került, a harmadik programot vagy GOTO "PR3" paranccsal, vagy 50 GOTO "PR3" utasítással lehet elindítani, amit előzetesen el kell helyezni a második programban. Ha betöltöttünk n db programot a memóriába a MERGE paranccsal, akkor az utasítást törölni, javítani, új utasítást beszúrni mindig a legutoljára betöltött programban lehet (tehát az n-edikben).

PRINT

szintaxisa: sorszám PRINT # "név"; adatlista
ahol a "név" és az adatlista megegyezik az INPUT# - nál leírtakkal.

Ez a parancs (vagy utasítás) lehetővé teszi, hogy szalagra adatokat mentsünk ki. Ha megadjuk a "név"-et, akkor az adatoknak ez lesz a közös neve. Használata megegyezik az INPUT # -nál leírtakkal (lásd ott!). A magnetofont az adatkivitelre készítjük elő!

Példa:

Ez a program az X változó utolsó értékét és az A tömb elemeit menti ki:

```
10 DIM A(20)
20 FOR X=0 TO 20
30 A(X)=X
40 NEXT X
50 PRINT # "MEZO"; X,A(X)
```

RMTON/RMTOFF

szintaxisa: sorszám RMTON
 illetve sorszám RMTOFF

Mint már szó volt róla, a PTA-4000-hez egyidejűleg két magnetofon csatlakoztatható. Ez a két utasítás pótolja a második magnetofon REMOTE kapcsolóját.

A RMTON-nal kapcsolhatjuk be a második magnetofon távvezérelt állapotát. Ha a fenti állapotot meg akarjuk szüntetni, a RMTOFF-ot kell kiadni.

Ha a második magnetofont használjuk, akkor a felvevő, ellenőrző és betöltő, stb. utasítások kulcszavai a következőképpen változnak meg:

kimentés: CSAVE-1
 betöltés: CLOAD-1
 laptechnika: CHAIN-1
 adatkimentés: PRINT # -1
 adatbetöltés: INPUT # -1
 szerkesztés: MERGE-1
 ellenőrzés: CLOAD?-1

Az utasítások többi része és a használatuk nem változik meg. A két magnetofon csatlakoztatását az 5.4.6. fejezet 2. pontja írja le.

A második magnetofont megfelelően elő kell készíteni!

6. BŐVÍTÉSI LEHETŐSÉGEK

A memóriamodulok használata

Az alapgép memóriája 1850 byte, ez az, amihez a felhasználó közvetlenül hozzáférhet. Ez a tárterület csak kisebb programokra elegendő. Lehetőség van arra, hogy a PTA-4000 memóriáját maximálisan 24 kbyte-ra kibővítsük.

A modulokat a következőképpen kell beszerezni:

1. Kapcsoljuk ki a PTA-4000-et és válasszuk le a KA.160-ról!
2. Távolítsuk el a gép aljának közepén levő fedelelet! Ide kerül a memóriamodul.
3. Helyezzük bele a modult és toljuk be a csatlakozórésbe!
4. Zárjuk vissza a fedelet!
5. Kapcsoljuk be a gépet! a gép kijelzőjén NEW=CHECK lesz látható.
6. Adjuk be a gépnek a **CL** NEW és **ENTER** -t.
7. A MEM vagy STATUS 0-val meggyőződhetünk arról, hogy mekkora memóriaterületünk van.

Ha a modult el akarjuk távolítani, értelemszerűen a fentiek fordítottját végezzük el.

Használható modulok: pl.: CE151, CE155, CE159, stb.

7. A PTA-4000 MŰSZAKI ADATAI

Modell	PTA-4000
Számábrázolás	10 számjegy mantissza, 2 jegy kitevő
Kijelzés	Folyadékkristályos kijelző, 26 karakteres pontmátrixból, alfametrikus vagy 7x156 pont- mátrixból álló grafikus
Billentyűzet	65 billentyű (betű, szám, második funkció, programozható és előre programozott billen- tyűzet).
Programnyelv	BASIC
Központi egység	8 bites CMOS mikroprocesszor
Memória	ROM - 16 Kbyte RAM - 3,5 Kbyte, ebből a fel- használó 2,6 Kbyte-hoz férhet hozzá (változók, programok).
Memóriavédelem	A gépben lévő információk a kikapcsolás után nem törlődnek.
Áramellátás	6 V (4 dB szárazelem) vagy hálózati adapter
Áramfogyasztás	6 V-on 0,13 W
Működési idő	Egy elemkészlettel kb. 50 óra
Működési hőmér- séklet	0 ^o -tól 40 ^o C-ig
Méretetek	195 * 86 * 25,5 mm
Tömeg	kb. 375 g (elemekkel)

A nyomtató műszaki adatai

Jelméret	1,2*Ø,8 mm-től 1Ø,8*7,2 mm-ig
Jelek száma soron- ként	4, 5, 6, 7, 9, 12, 18 vagy 36 a jelmérettől függően
Nyomtatási sebesség	max. 11 karakter/másodperc a legkisebb karakterméret esetén
Nyomtatási irány	4 (balról jobbra és vissza, felülről lefelé és vissza)
Nyomtatási szín	4 (fekete, kék, zöld, piros)
Papírtovábbítás	manuális vagy programozott
Papírtípus	58*3Ø mm-es (famentes javasolt)
Írópatronok	EA-85Ø B vagy EA-85Ø L

WWW.
PC-1500
.INFO

FÜGGELÉK





1. sz. mellékletASCII karakterkód táblázat a PTA 4000 géphez

		b7, b6, b5 bitek							
		000	001	010	011	100	101	110	111
b4, b3, b2 b1, b0 bitek	Hexa decimal	0	1	2	3	4	5	6	7
0000	0			SPACE	0	@	P		p
0001	1			!	1	A	Q	a	q
0010	2			"	2	B	R	b	r
0011	3			#	3	C	S	c	s
0100	4			\$	4	D	T	d	t
0101	5			%	5	E	U	e	u
0110	6			&	6	F	V	f	v
0111	7			□	7	G	W	g	w
1000	8			(8	H	X	h	x
1001	9)	9	I	Y	i	y
1010	A			*	:	J	Z	j	z
1011	B			+	;	K	√	k	{
1100	C			,	<	L	¥	l	!
1101	D			-	=	M	π	m	}
1110	E			.	>	N	^	n	~
1111	F			/	?	O	_	o	■

Használati példa: a g bináris kódja: 110 0111 (balra
a legnagyobb helyiérték)
hexadecimális kódja:67

2. sz. melléklet

Megengedett rövidítések

A számítógépen és a bővítőn használható kulcsszavakat le lehet rövidíteni. Ez a rövidítés azt jelenti, hogy az illető kulcsszavaknak legalább azokat a karaktereket tartalmaznia kell, amelyeket a későbbiekben megadunk. A rövidítés végén kötelezően egy pontnak (.) kell lenni!

Ha egy kulcsszó nem rövidíthető, akkor utána hiányjel (-) áll. Például a RESTORE-t rövidítjük, legalább RES.-t kell megadnunk, de jó a REST., RESO., RESTOR. is.

Kulcsszavak

AREAD	A.	IF	-
RUN	ARU.	INPUT	I.
BEEP	B.	LET	LE.
CLEAR	CL.	LOCK	LOC.
CLS	-	NEXT	N.
CURSOR	CU.	ON	O.
DATA	DA.	PAUSE	PA.
DEGREE	DE.	PRINT	PR.
DIM	D.	RADIAN	RAD.
END	E.	RANDOM	RA.
ERROR	ER.	READ	REA.
FOR	F.	REM	-
GCURSOR	GCU.	RESTORE	RES.
GOSUB	GOS.	RETURN	RE.
GOTO	G.	STEP	STE.
GPRINT	GP.	STOP	S.
GRAD	GR.	THEN	T.

TRON	TR.	WAIT	W.
TROFF	TROF.	PEEK	-
UNLOCK	UN.	POKE	-
USING	U.		

Parancsok

CONT	C.
LIST	L.
NEW	-
RUN	R.

Függvények

ABS	AB.	MEM	M.
ACS	AC.	MID §	MI.
AND	AN.	NOT	NO.
ASC	-	OR	-
ASN	AS.	PI	-
ATN	AT.	POINT	POI.
CHR §	CH.	RIGHT §	RI.
COS	-	RND	RN.
DEG	-	SGN	SG.
DMS	DM.	SIN	SI.
EXP	EX.	SQR	SQ.
INKEY §	INK.	STATUS	STA.
INT	-	STR §	STR.
LEFT §	LEF.	TAN	TA.
LEN	-	TIME	TI.
LOG	LO.	VAL	V.
LN	-		

Printer parancsok

COLOR	COL.
CSIZE	CSI.
GLCURSOR	GL.
GRAPH	GRAP
LCURSOR	LCU.
LF	-
LINE	LIN.
LLIST	LL.
LPRINT	LP.
RLINE	RL.
ROTATE	RO.
SORGN	SO.
TAB	-
TEST	TE.
TEXT	TEX.

Kazettaillesztő parancsok

CHAIN	CHA.
CLOAD	CLO.
CLOAD?	CLO.?
CSAVE	CS.
INPUT	I.
MERGE	MER.
PRINT	P.
RMTOFF	RM.OF.
RMTON	RM.O.

3. sz. mellékletProgramindítási módszerek összehasonlítása

A PTA-4000 többféle programindítási módszert ismer. Programjainkat RUN, RUN sorszám vagy címke, ARUN, GOTO sorszám vagy címke, DEF címke billentyűzéssel indíthatjuk el vagy CONT paranccsal folytathatjuk. A különböző programindítási módszerek hatása is különbözik. Ebből a szempontból különböznek az egybetűs azonosítójú numerikus illetve karakteres változók a két vagy többjelből álló azonosítójú numerikus, karakteres ill. tömbváltozóktól. A két vagy több betűs változókat a RUN parancs törli (nullává illetve nullstring-gé teszi), az összes változót, az egybetűseket is csak a CLEAR parancs vagy utasítás törli (vagy a NEW parancs a programmal együtt). A programindítási módszerek közötti különbséget a következő táblázatban foglaljuk össze:

Hatás	Program indítási módszer		
	ARUN, RUN sorszám v. címke	GOTO sorszám v. címke	DEF címke
Törli a kijelzőt	igen	igen	nem
A cursor visszatér az első pozícióba	igen	nem	nem
A két vagy több betűs változók és a tömbváltozók törlődnek	igen	nem	nem
Az ON ERROR GOTO utasítás hatása törlődik	igen	nem	nem

Hatás	Program indítási módszer		
	ARUN, RUN sorszám v. címké	GOTO sorszám v.címke	DEF címké
A DATA utasítást tartalmazó program- ban automatikus RESTORE hajtódik végre	igen	nem	nem
Az érvényben lévő USING formátum tör- lődik	igen	nem	nem
Az érvényben lévő WAIT utasítás tör- lődik	igen	nem	nem
A DIM utasítás el- veszti hatását	igen	nem	nem

Például ha egy a 10 sorban DIM utasítást tartalmazó programot kétszer egymás után GOTO 10 paranccsal indítunk, a második indításnál hibajelzést kapunk.

4. sz. mellékletA PTA-4000 hibaüzenetei

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
1	ERROR 1 IN 1Ø	Szintaktikai hiba a; hiányzó információ miatt: 1Ø GOTO b; felismerhetetlen uta- sítás miatt: pl: 1Ø 5A=1 vagy: 1Ø NEW (a NEW nem írható programba)
2	ERROR 2 IN 3Ø	FOR nélküli NEXT vagy GOSUB nélküli RETURN pl: 1Ø FOR X=1 TO 1Ø 2Ø A(X)=X 3Ø NEXT Y
4	ERROR 4 IN 2Ø	A READ-hoz tartozó DATA túl rövid vagy nem is létezik pl: 1Ø DATA 1,2 2Ø READ X,Y,Z,
5	ERROR 5 IN 2Ø	Egynél több, azonos vál- tozónévre vonatkozó tömbdeklaráció. pl: 1Ø DIM A(2) 2Ø DIM A(5,2)

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
6	ERROR 6 IN 2Ø	A tömböt korábban nem deklaráltuk DIM utasításban. pl: 1Ø CLEAR 2Ø A(1)=2
7	ERROR 7 IN 1Ø	Nem megfelelő változót használtunk. pl: 1Ø AØ=1Ø vagy: 1Ø A="ROSSZ"
8	ERROR 8 IN 1ØØ	A tömbdeklarációban a dimenziószám több mint 2 pl: 1ØØ DIM A(2,3,4)
9	ERROR 9 IN 2Ø	Az index kifejezés értéke nagyobb a deklarált indexhatárnál. pl: 1Ø DIM A(3) 2Ø A(4)=1
1Ø	ERROR 1Ø	Az új változó felvételéhez már nem áll rendelkezésre elegendő tárolóhely.
11.	ERROR 11 IN 2Ø	Vezérlésátadás nem létező sorra. pl: 1Ø A=2 2Ø GOTO 15

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
12	ERROR 12 IN 1Ø	Nem létező USING speci- fikáció. p1: 1Ø PRINT USING "***# # A# .# # ",A
13	ERROR 13	Nincs elég memóriahely több program táro- lásához vagy új RESERVE definíciók felvételéhez.
14	ERROR 14	A BASIC - Stack (verem) kapacitását egymásba á- gyazott FOR-NEXT hurkok- kal vagy egy aritmetikus kifejezés kiszámítása közben túlléptük.
15	ERROR 15	A BASIC - Stack kapaci- tását egymásba ágyazott GOSUB-RETURN utasítások- kal, vagy egy aritmeti- kai kifejezés kiszámítá- sával túlléptük.
16	ERROR 16	A szám nagyobb vagy e- gyenlő, mint 1 E 1ØØ, vagy kisebb vagy egyenlő mint -1 E 1ØØ illetve a hexadecimális számnak megfelelő decimális szám nagyobb, mint 65535.

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
17	ERROR 17 IN 1Ø	A kifejezésben nem megengedett változótypus szerepel. pl: 1Ø L=1+"A"
18	ERROR 18 IN 1Ø	A függvény argumentuma nem megfelelően volt megadva. pl: 1Ø RIGHT Ø ("AB") vagy: 1Ø COS (1Ø; 2Ø)
19	ERROR 19 IN 1Ø	A numerikus érték a megengedett határon kívül van. pl: 1Ø DIM A(256)
2Ø	ERROR 2Ø IN 1Ø	A standard változóra nem megfelelően hivatkoztunk. pl: 1Ø @=3 vagy: 1Ø @="12"
21	ERROR 21 IN 1Ø	A kifejezésből hiányzik a változónév. pl: 1Ø FOR 1=Ø TO 1Ø
22	ERROR 22	A betöltendő program számára nincs elegendő memóriahely.

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
23	ERROR 23 IN 1Ø	A belső órát nem megfelelő értékre akarjuk beállítani. pl: 1Ø TIME=131ØØ5.1Ø
26	ERROR 26	Az aktuális üzemmódban végrehajthatatlan parancs vagy utasítás. pl: PRO üzemmódban RUN és ENTER
27	ERROR 27	Nem létező címkeire hivatkozunk. pl: DEF Ø
28	ERROR 28 IN 1Ø	INPUT vagy AREAD után egy változóhoz egy BASIC kulcsszót rendeltünk. pl: 1Ø INPUT AØ Indítsuk el ezt a programot és a ?-re adjuk be pl: DEF W
3Ø	ERROR 3Ø	Az utasítássorszám meghaladja a 65539-et (ha 6528Ø és 65539 közé esik, ERROR 1 jelzést kapunk).

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
32	ERROR 32 IN 2Ø	A CURSOR annyira jobbra haladt, hogy az INPUT-tal kért adatot már nem lehet kijelezni. pl: 1Ø GCURSOR 152 2Ø INPUT X
36	ERROR 36 IN 2Ø	Az USING specifikációt a megadott kifejezéshez kicsinek választottuk. pl: 1Ø USING "###.###" 2Ø PRINT 123456
37	ERROR 37	Az aritmetikai kifejezés abszolút értéke meghaladja a 9.999999999 E 99-et
38	ERROR 38 IN 1Ø	Nullával való osztás. pl: 1Ø PRINT 5/Ø
39	ERROR 39	A függvény argumentuma kívül esik az értelmezés tartományon. pl: LN (-1Ø) vagy: SQR (-3)
177-181	ERROR 177,..., ERROR 181	A programterület és az adatterület összeér.

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
0, 224-241	ERROR 0, ERROR 224, ..., ERROR 241	Az INPUT vagy az AREAD utasítások végrehajtásakor helytelen adatot adtunk meg.

A magnetofon használatával kapcsolatos hibajelzések

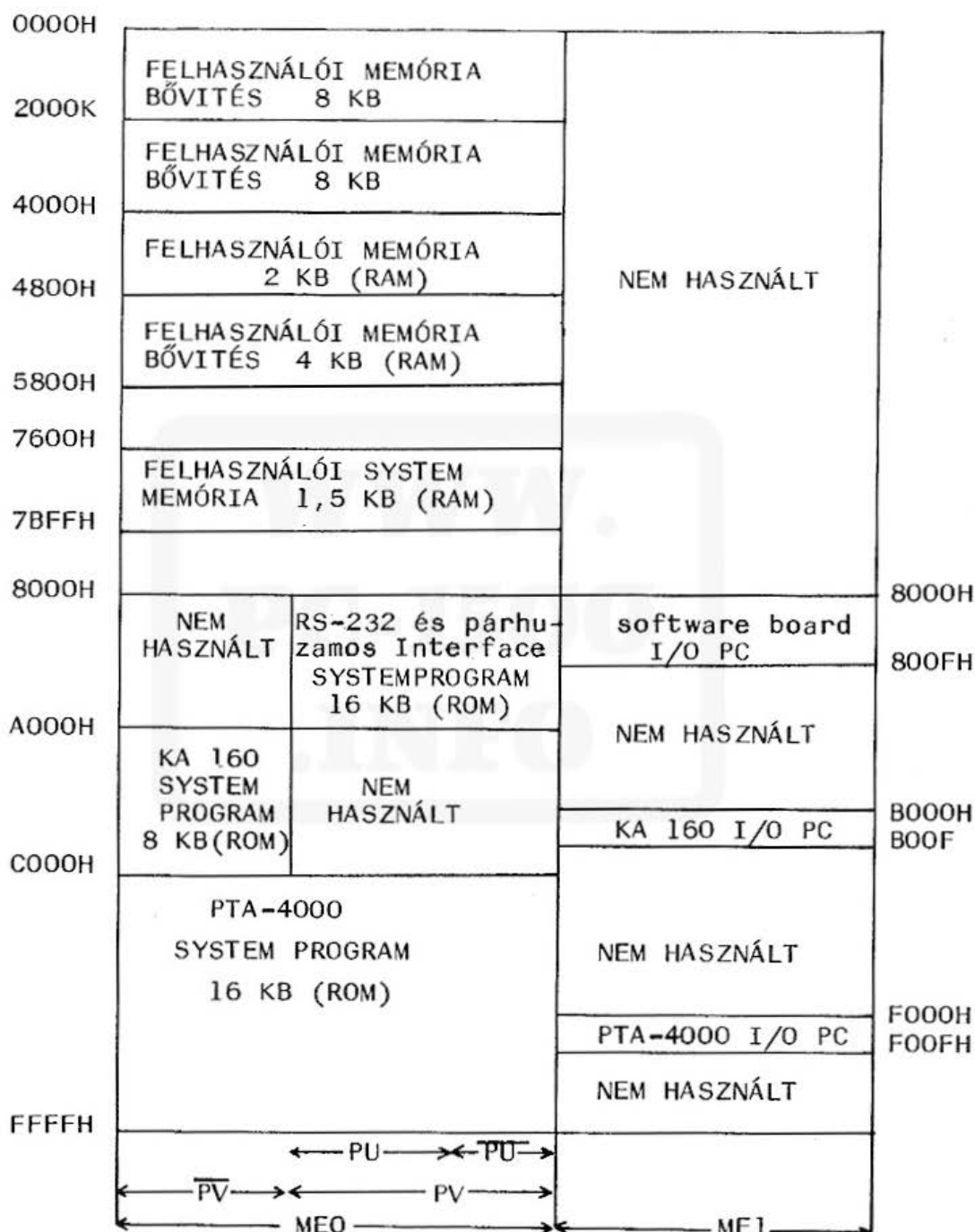
<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
40	ERROR 40	Helytelen specifikáció a kifejezésben.
42	ERROR 42	A kazettán lévő adatok számára nincs hely a memóriában.
43	ERROR 43	A kazettán tárolt információk nem egyeznek meg a memóriában lévő információkkal a CLOAD? alkalmazása során.
44	ERROR 44	Az ellenőrző összeg hibás. Ez a hiba csak hibás betöltés esetén léphet fel.

A nyomtató használatával kapcsolatos hibajelzések

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
70	ERROR 70	A toll kívül került vagy kívül kerülne a megengedett (-2048, 2047) tartományon.
71	ERROR 71	A TEXT üzemmódban a papírt több mint 10,24 cm-rel mozdítanánk vissza.
72	ERROR 72	A TAB vagy az LCURSOR utasításban megadott kifejezés nem megengedett.
73	ERROR 73	A kiadott parancs (vagy utasítás) az aktuális nyomtató üzemmódban nem hajtható végre.
74	ERROR 74	A LINE vagy RLINE utasításokban túl sok paramétert adtunk meg. pl: 10 LINE (0,0) - (10,0) - (0,2,3)
76	ERROR 76	A TEXT üzemmódban az LPRINT utasításnál a nyomtatási kép nem fér el egy sorban.

<u>Hibakód</u>	<u>Jelzés</u>	<u>Hiba oka</u>
78	ERROR 78	1./ A tollcserét nem fejeztük be. 2./ A LOW BATTERY állapotot nem töröltük. 3./ Az LPRINT vagy LINE utasítás nem hajtható végre.
79	ERROR 79	A szín pozíciójel nem érkezett meg (hardware hiba).
80	ERROR 80	Alacsony a tápfeszültség.

WWW.
PC-1500
.INFO

5. sz. mellékletPTA-4000 memóriatérkép

6. sz. mellékletKA 160 csatlakozó bekötés

pólusszám

1.	AD7	7. címvezeték
2.	AD6	6. címvezeték
3.	AD5	5. címvezeték
4.	AD4	4. címvezeték
5.	AD3	3. címvezeték
6.	AD2	2. címvezeték
7.	AD1	1. címvezeték
8.	AD \emptyset	\emptyset . címvezeték
9.	PB \emptyset	nem használt
10.	PC7	nem használt
11.	VCC	
12.	VCC	
13.	NC	
14.	NC	
15.	PV	Chip select
16.	PU	Chip select
17.	D7.	7. adatvezeték
18.	D6.	6. adatvezeték
19.	D5.	5. adatvezeték
20.	D4.	4. adatvezeték
21.	D3.	3. adatvezeték
22.	D2.	2. adatvezeték
23.	D1.	1. adatvezeték
24.	D \emptyset	\emptyset . adatvezeték
25.	<u>INHIBIT</u>	Megakadályozza ROM kiválasztását a PTA 4000-nek, ha földre kerül
26.	<u>WEX</u>	külső WAIT jel
27.	<u>CMT IN</u>	kazettás egység adat bemenet

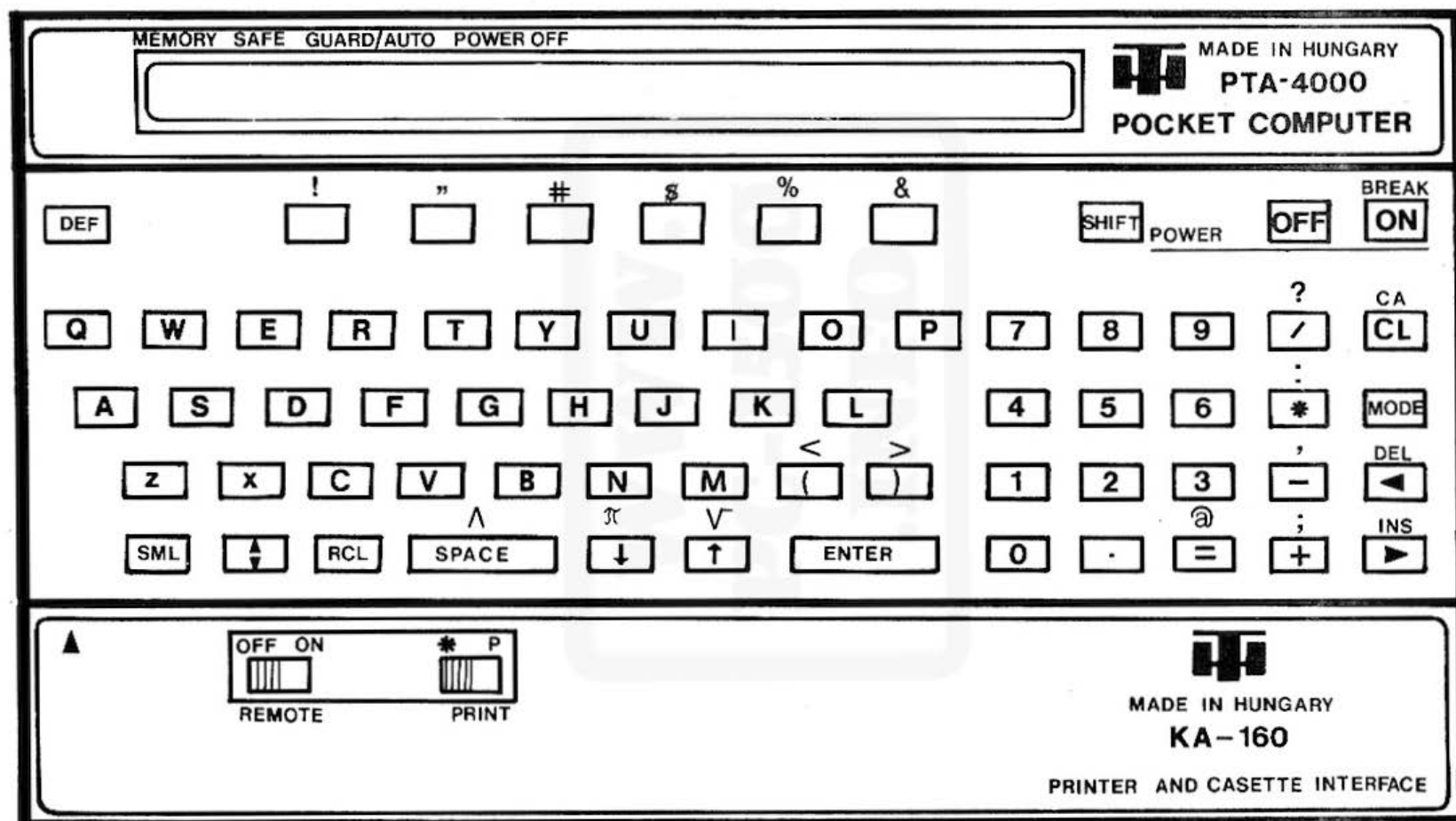
pólusszám

28.	W1	WAIT állapot bemenet
29.	CMT OUT	kazettás egység adat kimenet
30.	INT	megszakítás kérés a CPU-hoz
31.	AD8	8. címvezeték
32.	AD9	9. címvezeték
33.	AD10	10. címvezeték
34.	AD11	11. címvezeték
35.	AD12	12. címvezeték
36.	AD13	13. címvezeték
37.	AD14	14. címvezeték
38.	AD15	15. címvezeték
39.	PB1	nem használt
40.	NC	
41.	VCC	
42.	VCC	
43.	F.GND	Ház föld
44.	VBAT	akkumulátor feszültség
45.	VBAT	akkumulátor feszültség
46.	VBAT	akkumulátor feszültség
47.	VBAT	akkumulátor feszültség
48.	VBAT	akkumulátor feszültség
49.	NC	
50.	BFO	VCC output
51.	POS	órajel
52.	GND	
53.	GND	
54.	GND	
55.	GND	
56.	DME	CHIP select a ME 0 területen
57.	R/W	olvasás/írás jel
58.	DME1.	CHIP select az ME1 területen
59.	ME1	ME1 memóriaterület kiválasztás
60.	OD	kimenet tiltás

7. sz. melléklet

TARTOZÉKOK

KA 160 Nyomtató és magnetofon illesztő	1 db
EØ9D Hálózati adapter	1 db
Magnetofon csatlakozó kábel (3xJack- 3xJack)	1 db
Magnetofon távkapcsoló kábel (1xJack- 1xJack)	1 db
Tuchel (DIN) csatlakozó	1 db
Billentyű-kód lemez	1 db
Nyomtató-írótoll készlet	2 db
Nyomtató-tartalék papír	2 db
Gépkönyv	1 db
PTA 4000 védőtok	1 db
Tárolódoboz	1 db



9. sz. melléklet

INPUT	PRINT	USING	GOTO	GOSUB	RETURN	CSAVE	CLOAD	MERGE	LIST
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



A könyv összeállítása során igyekeztünk a legkörültekintőbben eljárni, mégis előfordul, hogy egyes helyeken a fogalmazás nem elég érthető vagy hibát is tartalmaz.

Kérjük, további munkánk segítésére írja meg észrevételeit, tapasztalatait.

Segítségét köszönjük.

HIRADÁSTECHNIKA SZÖVETKEZET

Feladó:

HIRADÁSTECHNIKA SZÖVETKEZET

Számítástechnikai Vevőszolgálat

Budapest

Pt. 268.

H - 1419

