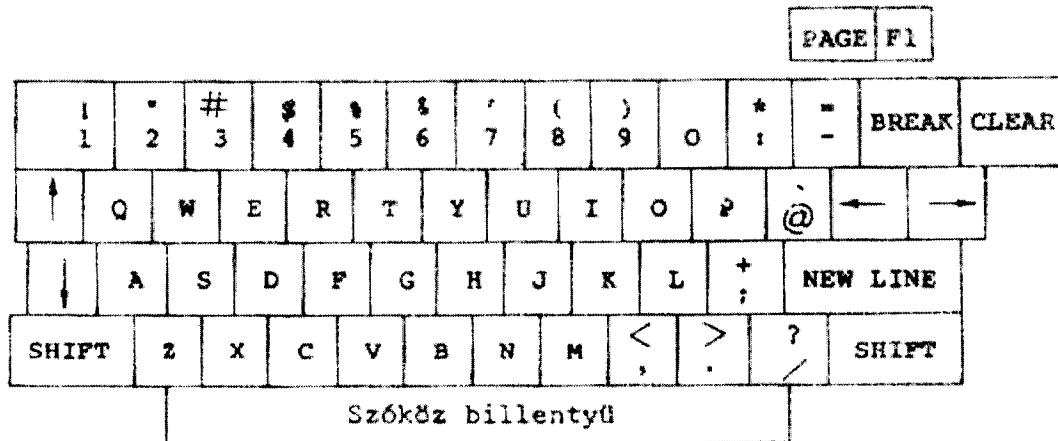


## BILLENTYŰZET



72. ábra

- BREAK** - programmegszakítást okoz  
- folytatás a CONT paranccsal lehetséges
- CLEAR** - képernyő törlése  
- 64 karakteres megjelenítés
- NEW LINE** - parancsok és utasítássorok lezárása
- SHIFT** - más billentyű lenyomásával együtt hatásos  
- felső karakterek előállítása
- SHIFT/@** - program leállítása, folytatás bármely billentyű lenyomásával
- SHIFT/→** - 32 karakteres megjelenítés
- SHIFT/·** - sor törlés
- szóköz billentyű** - a legalsó, hosszú billentyű (SPACE)  
- egy karakterpozíció kihagyása  
· - egy karakter törlése  
→ - 8., 16., 24. stb. karakterpozícióba állítja kurzort  
↑ - hatványozás jele  
↓ - következő sor elejére áll

## A BASIC NYELV ALAPELEMEI

<b>Sorszámok</b>	– pozitív egész számok az utasítássorok elején
	– Nem lehet nagyobb 65529-nél.
<b>Kulcsszavak</b>	– a BASIC rendszer számára lefoglalt betűkombinációk
<b>kezelési parancsok</b>	– 3. melléklet
<b>utasítások</b>	– 4. melléklet
<b>függvénynevek</b>	
aritmetikai	– 5. melléklet
karakterlánc	– 6. melléklet
egyéb	– 7. melléklet
<b>Konstansok</b>	
<b>numerikus</b>	decimális számok
<b>egész</b>	– 32768 és 32767 közötti szám
<b>egyszeres pontosságú</b>	
– 1,701411E+38-nál nem nagyobb abszolútértékű szám	
<b>kétszeres pontosságú</b>	
– 1,701411834244556D+38-nál nem nagyobb abszolútértékű szám	
<b>Karakterlánc</b>	– a billentyűzettel előállítható, és a képernyőn megjelenő karakterek sorozata
	– A programban csak akkor nem kell idézőjelek közé tenni, ha a DATA-ban szerepel.
<b>Változó</b>	– adatok tárolására, azok azonosítására szolgáló szimbólum
<b>egyszerű</b>	
<b>numerikus</b>	
egész	– A%      IK%      J5%
	– DEFINT A, I – K
egyszeres pontosságú	–
	– A X1 IJ P! XQ!
	– DEFSNG X, A – C
kétszeres pontosságú	
	– A#      XY#      K3#
	– DEFDBL C, X, P – S
<b>karakterlánc</b>	
	– A\$ X3\$ IK\$
	– DEFSTR A, X – Z
<b>indexes</b>	
<b>numerikus</b>	
egész	– A% (I)    B1% (15,2 *X)
egyszeres pontosságú	
	– A(50, N)    B3!(15)    IJ(2*I+K)

kétszeres pontosságú

– A#(K, J%)    B1#(X)

karakterlánc

– X\$(I)    XY\$(15, I+K)

– A DEFINT, DEFSGN, DEFDBL és a DEFSTR utasításokkal is kijelölhetjük a változó típusát

– A maximális index értékét a DIM utasítással kell megadni  
DIM A(N, M), B(100)

– Több indexet is használhatunk

### Műveleti jelek

aritmetikai

( ) zárójelek

↑ hatványozás

\* szorzás

/ osztás

+ összeadás

– kivonás

relációk

= egyenlő

< kisebb

<= kisebb, vagy egyenlő

> nagyobb

>= nagyobb vagy egyenlő

logikai

NOT tagadás

AND ÉS művelet

OR VAGY művelet

### Elválasztójelek

kettőspont    utasítások elválasztása

15 A+2: PRINT A

vessző    változók és adatlisták, indexek, függvényváltozók elválasztása

PRINT 150, "PONT"

READ A, B, C

DATA 15, BUKK, 137

DIM A(N, M)

A\$= LEFT\$(A\$, 2)

pontosvessző    kiíratási listaelemek elválasztása

PRINT "TERULET"; A; "CM ↑ 2"

INPUT "SUGAR"; R

mínusz jel

– DEF utasításokban a kezdőbetűk kijelölése

DEFINT A–D, X–Z

**KEZELÉSI PARANCSONK**

(81. oldal)

- AUTO — automatikus sorszámozás  
 AUTO — a sorszárok: 10, 20, ...  
 AUTO 2, 3 — a sorszárok: 2, 5, 8, ...  
 AUTO 3 — a sorszárok: 3, 13, 23, ...

A sorszárok után egy \* csillag jelzi, ha már van ilyen sorszáruk utasítás.

- CLOAD beolvasás magnetofonról  
 CLOAD — a beépített magnetofonról beolvassa a következő programot  
 CLOAD#-1, "JATEK" — a beépített magnetofonról beolvassa a J nevű programot  
 CLOAD#-2, "BETU" — a csatlakoztatott magnetofonról beolvassa a B nevű programot

A magnetofont lejátszásra kell kapcsolni, miután a szalagot a program elejére állítottuk. Helyes beolvasáskor a két \* jel közül a másodikkal villognia kell a képernyő jobb felső sarkában.

Ha két \* -ot jelez ki állandóan: a RESET gomb benyomásával állítsuk alaphelyzetbe a rendszert, és kezdjük előlről a beolvasást. Ugyanezt kell tennünk, ha REDO kiírással jelez hibát a rendszer.

- CLOAD?#-1, "X" ellenőrzi az X nevű tárolt programot a beépített magnetofonon  
 CLOAD? ellenőrzi a következő programot, amit a szalagon talál.

Kiírás után kell a vizsgálatot elvégezni, amikor a program még a memóriában van.

- CONT — megszakított program végrehajtásának folytatása  
 CONT — folytatódik a program futása, ha a STOP utasítással vagy a BREAK billentyű lenyomására áll le a program.

- CSAVE — program felvétel mágnesszalagra  
 CSAVE#-1, "EMSE" — Az E nevű program felvétele a beépített magnetofonon történik  
 CSAVE "E" — az előző utasítással azonos  
 CSAVE#-2, "ERA" — az E nevű program rögzítése a csatlakoztatott magnetofonon történik

— A magnetofont felvételre kell állítani.

- DELETE — utasítások törlése  
 DELETE 22-60 — A 22-es sorszámtól a 60-as sorszámmal bezárólag törli az utasítássorokat  
 DELETE-30 — A 30-as utasítássorig törli a programot  
 — Egy utasítássort a sorszáruk beírásával is törölhetünk.

- EDIT — utasítássor módosítása

Parancsmódban maradunk:

- kurzor mozgatása  
 SPACE — jobbra egy karakterpozícióval  
 5SPACE — jobbra öt karakterpozícióval

- ← – balra egy karakterpozícióval
- 6← – balra hat karakterpozícióval
- L – kiíratás
  - L – kiírja a sort
- S – keresés
  - SK – az első K betű elé áll a kurzor
  - 3SV – a harmadik V betű elé áll a kurzor
- D – törlés
  - D – törli a kurzor mögött álló karaktert
  - 3D – három karaktert töröl a kurzortól jobbra
- K – törlés adott karakterig
  - 3KZ – a kurzortól jobbra a harmadik Z karakterig törli a sort
  - KA – a kurzortól jobbra az A karakterig törli a sort (A-t már nem)
- C csere
  - CW – oda írja a W karakter, ahol a kurzor áll
  - 2CXY – oda, ahol a kurzor áll, beír két karaktert, X-et és Y-t.
- módosítás befejezése
  - A – a módosítást kezdhethetjük előlről
  - Q – a módosítás befejeződik és az utasítássor változatlan marad
  - NEWLINE – a változások lezárásával fejeződik be a módosítás
  - E – ugyanaz, mint a NEWLINE

Parancsmódból szövegmódba kerülünk:

- I – ahol a kurzor állt, ott módosíthatunk
- H – a kurzor helyétől kezdődően törli a sor végét
- X – a kurzor a sor végére áll

Módosítási lehetőségek:

- új karaktert írhatunk be
- a← billentyű lenyomásával karaktereket törölhetünk

SHIFT ↑ – átváltás szövegmódból parancsmódba

LIST – listázás képernyőre

LIST – kiírja a programot

LIST 35 – kiírja a 35-ös utasítássort

LIST 20–50 – a 20-as sorszámú utasítássortól az 50-esig kiírja a programot

LIST 60 – a 60-as utasítássorig kiírja a programot

A kiíratást a SHIFT és @ billentyűk együttes lenyomásával megállíthatjuk, és bármelyik billentyű lenyomásával folytathatjuk.

LLIST – program kiírása sornyomtatón

NEW – program törlése

Törli a programot és a képernyőt

RE – újrasorszámozás

Az eredeti sorszámokat átírja.

RE – az új sorszámok: 10, 20, 30, ...

RE2 – az új sorszámok: 2, 12, 22, ...

RE3, 5 – az új sorszámok: 3, 8, 11, ...

RE4 – az új sorszámok: 10, 14, 18, ...

Ezt az utasítást csak akkor használhatjuk, ha a BASIC rendszerünket a 12288-as, 12299-es vagy a 12294-es címről indítjuk a SYSTEM utasítással.

RUN program indítása

RUN — A legkisebb sorszámú utasítássor végrehajtásával kezdődik a program végrehajtása

RUN 30 — A 30-as utasítássortól indul a program

A program futását a SHIFT és @ billentyűk együttes lenyomásával megállíthatjuk és bármelyik billentyű lenyomásával folytathatjuk.

SYSTEM — rendszerprogramok indítása

\* ? kiírással kéri a rendszerprogramok kezdőcímét

/cim begépelésével indul a rendszerprogramunk

A címet decimális számmal kell megadni.

/20489 Gépi szintű programok részére lefoglalt memóriaterület legkisebb címének a beírásával vagy a NEWLINE billentyű lenyomásával indíthatjuk a BASIC rendszert.

A BASIC rendszer bővítése:

/12294 — kisbetűket írhatunk a SHIFT és a betűket jelző billentyűk egyidejű lenyomásával

— a RE utasítást használhatjuk

/12299 — az előző lehetőségek érvényesek, és ha lenyomva tartjuk a billentyűt, a karakter kiírása ismétlődik

/12288 — az előző lehetőségek érvényesek és megjelenik a villogó kurzor. A villogást a SHIFT és a BREAK billentyűk egyidejű lenyomásával tilthatjuk le, ill. engedélyezhetjük.

Monitor program

A központi egység regisztereinek és a memóriarekeszek tartalmának vizsgálatát, módosítását, gépi szintű programok írását, futtatását teszi lehetővé.

/12710 — monitor program indítása

Kiírja a központi egység regisztereinek értékét hexadecimális számokkal.

Lehetséges parancsok:

D — memóriatartalom vizsgálata

D234F — a 234F címtől kezdődően 16 memóriarekesz tartalmát kiírja

— a billentyű lenyomásával kisebb, a billentyűvel nagyobb címről írható ki a tartalom

— bármelyik billentyűt lenyomva befejeződik a memóriavizsgálat.

R — regiszterek tartalmának módosítása

— egymás után kiírja és kéri a regiszterek értékeit. Miután mind a 12 regiszternek értéket adtunk, visszatérünk monitor üzemmódba.

M — beírás a memóriába

M3D12 — a 3D12 címtől kezdődően megváltoztatjuk a memóriatartalmat, a legkisebb megadható cím 3C00.

Az X billentyű lenyomásával fejezhetjük be a beírást.

G — gépi szintű program indítása.

G700E, 71D2 — a 700E címtől kezdődik a gépi szintű utasítások végrehajtása.  
A 71D2 cím elérése után a monitor program folytatódik.

B — visszatérés a BASIC rendszerbe.

TROFF — nyomkövetés kikapcsolása

A TRON utasítást hatástalanítja

TRON — nyomkövetés

Program futásakor kiíródnak képernyőre a végrehajtott utasítássorok sorszámai.

— Rendszerünk megengedi, hogy a kezelési parancsokat a programba beírjuk.

## UTASÍTÁSOK

**CLEAR** kifejezés – memóriatörlés

A kifejezés egész értékének megfelelő számú rekeszt nulláz a memóriából, és törli a változók értékeit.

**CLEAR** – törli a változókat

**CLEAR 2000** – törli a változókat és 2000 bájtot

– karakterláncok helyét így biztosítjuk a memóriában

**CLS** – képernyő törlése

**DATA** adatlista – adatok elhelyezése programban

Az adatlistában számok és stringerek lehetnek. Az adatlista elemeit vesszővel választjuk el.

**DATA 23 456, "ADAT", 2356, "EVE", MISKOLC**

– Az adatok beolvasása a **READ** utasítással történik.

– A beolvasást a **RESTORE** utasítással lehet a **DATA** adatmező elejéről kezdeni.

**DEFDBL** betűlista – kétszeres pontosságú változók kijelölése

**DEFDBL A, B, E, H** – Az A, B, E, F, G, H betűkkel kezdődő változók kétszeres pontosságúak lesznek.

A kétszeres pontosságú változók jelölésére a # jelet is használhatjuk **A#, A1#**

**DEFINT** betűlista – egész változók kijelölése

**DEFINT I, N, T** – Az I, J, K, L, M, N és a T betűvel kezdődő változók egész változók lesznek.

Az egész változókat a % jellel is kijelölhetjük: **X1%, A% (I)**

**DEFSNG** betűlista – egyszeres pontosságú változók kijelölése

**DEFSNG X, Y, A, C** – Az X, Y, A, B és C betűkkel kezdődő változók egyszeres pontosságúak lesznek.

Egyszeres pontosságú változók kijelölésére a ! jelet is használhatjuk: **R!, C3! (15)**

– A változók alaphelyzetben egyszeres pontosságúak.

**DEFSTR** betűlista – string változók kijelölése

**DEFSTR R, T, A, G** – Az R, S, T, A és a G betűkkel kezdődő változók karakterlánc-változók lesznek.

A karakterlánc-változók jelölésére a \$ jelet is használhatjuk: **A\$, X\$ (I), LI\$**

**DIM** tömblista – tömbök dimenzionálása

A tömblista elemeinek indexeivel megadjuk a tömbök dimenzióját és az indexek maximális értékét.

**DIM A(10), X(5,10,4,12)** – Az A vektor 10 elemű lehet. Az X többdimenziós tömb, indexeinek értéke legfeljebb 5, 10, 4, ill. 12 lehet.

**DIM Y(2\*N+1,50)** – Az Y mátrix elemeinek első indexe nem lehet nagyobb 2\*N+1 értékénél, a második indexe 50-nél. Az N változónak korábban értéket kell adni.

– A tömbök elemeinek indexe lehet 0, de nem lehet negatív szám.

**END** – a program logikai vége

Vége a programnak. Ezt a **READY** szó kiírásával jelzi a rendszer.

**ERROR** hibakód – hiba szimulálás



A rendszer úgy reagál az utasításra, mintha a megadott kódszámú hiba bekövetkezett volna.

A hibák kódja a mellékletben található.

– Az ON ERROR GOTO utasítás tesztelésénél használjuk.

FOR TO - STEP/NEXT ciklus utasítás

FOR változó = kifejezés TO kifejezés STEP kifejezés

NEXT változó

A FOR és a NEXT utasítások közé írt utasításokat ismételten végrehajtja.

Az ismétlések számát, a ciklusváltozó értékét a kifejezések értékei határozzák meg.

FOR X = 4 TO 26 STEP 3      A ciklusmag végrehajtása X = 4, 7, 10, ... 25 értékekkel történik.

NEXT X

FOR I%=N TO 1 STEP -1      Az I változó N, N-1, ... , 2, 1 értékeket veszi fel a ciklusmag végrehajtásakor.

NEXT I%

FOR I = 1 TO M      M számú adatot olvas be. A Ciklusváltozó értékének változását nem kell megadni, ha az 1.

INPUT A (I)

NEXT I

FOR I = 1 TO M: INPUT A(I) : NEXT I      A ciklust egy sorban is elhelyezhetjük.

FOR J = 1 TO 200 : NEXT J      Üres ciklus. Várakozás.

A ciklusváltozó nem lehet kétszeres pontosságú.

– Minden FOR utasításhoz egy NEXT utasítás.

– A NEXT utasításban ugyanannak a változónak kell szerepelni, mint a FOR utasításban.

– A NEXT utasítással kell befejezni a ciklusmag végrehajtását.

– A ciklusba nem lehet kívülről „beugrani”. A FOR utasítással kezdődik a ciklus.

– A ciklusból kiugorhatunk.

– A ciklusmag a változó kezdőértékével egyszer mindig végrehajtódik.

– Több ciklust is egymásba építhetünk, de azok nem keresztezhetik egymást.

– A NEXT utasításban nem kell a ciklusváltozót kijelölni.

GOSUB sorszám – szubrutin hívás

A megadott sorszámú utasítással folytatódik a program. A szubrutin végét jelző RETURN utasítás hatására a GOSUB utasítást követő utasítás végrehajtására kerül sor.

GOSUB 300      – A 300-as sortól kezdődő szubrutin végrehajtása következik.

– A szubrutinból is hívhatunk szubrutinokat.

GOTO sorszám – feltétlen ugrás

Az utasításban megadott sorszámú utasítással folytatódik a program.

GOTO 125 – A 125-ös sorszámú utasítással folytatódik a program.

IF THEN ELSE – Feltételes vezérlés

IF állítás THEN utasítások v. sorszám ELSE utasítások v. sorszám.

Állítás lehet egy reláció vagy ezekre kijelölt logikai művelet.

Az állítás igaz: a THEN szó után szereplő utasítások végrehajtására kerül sor, vagy az adott sorszámú utasítással folytatódik a program.

Az állítás hamis: az ELSE után írt utasítások végrehajtására kerül sor, vagy az adott sorszámú utasítással folytatódik a program.

IF  $A < 0$  THEN  $B = -A$  ELSE  $B = A$  – Ha A értéke negatív, akkor  $B = -A$ , ha  $A = 0$  vagy pozitív,  $B = A$  utasítás végrehajtása történik.

IF  $A = 3$  OR  $B = 4$  THEN 50 – Ha az A értéke 3 vagy a B értéke 4, az 50-es utasítással folytatódik a program, egyébként az IF után következő sorszámú utasítással.

IF  $A = 1$  THEN IF  $B = 1$  THEN 30 ELSE 40 – Ha az A értéke 1, és a B értéke is egy, akkor a 30-as utasítás következik. A 40-es utasításra akkor jutunk, ha A egyenlő 1-gyel, és B nem egyenlő 1-gyel. Egyébként a következő sorral folytatódik a program.

IF  $X = 0$  THEN  $A = 1$  : GOTO 50 – X egyenlő nulla esetén A értéke 1 lesz és az 50-es sorral folytatódik a program.

IF POINT(X, Y) THEN RESET(X, Y) ELSE SET(X, Y) – Akkor, ha az X, Y koordinátájú pont világít a képernyőn törli azt, egyébként megjeleníti.

50 IF INKEY\$ = " " THEN 50 – Mindaddig várakozik a program, ameddig le nem ütünk egy billentyűt.

INPUT beolvasási lista – adatbeolvasás

– Listaelem lehet: magyarázó szöveg, változók

– Kiírja a magyarázó szöveget és kéri a változók értékeit a billentyűzetről, vagy magnetofonról. Ezt ? kiírással jelzi.

INPUT "OLDALAK" : A, B – Kiírja a képernyőre azt, hogy OLDALAK és kéri az A és B változók értékeit a billentyűzetről.

INPUT # 1, X, Y, Z – Az X, Y és a Z változók értékeit olvassa be az 1-es számú magnetofonról.

Adatbevitel:

– A változók értékei közé vesszőket gépeljünk és legvégül a NEWLINE billentyűt nyomjuk le.

– A változók értékei után a NEWLINE billentyűt nyomjuk le.

– Csak közvetlenül az INPUT utasítást követően helyezhető el magyarázó szöveg. Utána : t kell írni.

LET változó = kifejezés – értékadás

Az egyenlőség bal oldalán álló változó felveszi a jobb oldalon álló kifejezés kiszámí-

tott értékét. A változó és a kifejezés lehet aritmetikai vagy karakterlánc is, de a típusnak meg kell egyeznie.

A LET utasításszó elhagyható.

$A = 4 * B + C$  – A B változó értékének 4-szereséhez hozzáadjuk C változó értékét, és ez lesz az A változó értéke.

$R\$ = "ABC" + K\$$  – Ha a K\$ értéke XY, akkor R karakterlánc-változó értéke ABCXY lesz.

– Akkor, ha az aritmetikai kifejezés nagyobb pontosságú, mint a változó, értékes számjegyeket veszíthetünk.

– A karakterlánc-változóknak maximum 255 karaktert helyezhetünk el.

LPRINT kiíratási lista – kiíratás sornyomtatóra

A PRINT utasításnál leírtak érvényesek.

ON kifejezés GOSUB sorszámlista – feltételes szubrutinhívás

A kifejezés egész értéke meghatározza, hogy a sorszámlista hányadik elemével jelölt szubrutin hívására kerül sor.

$ON X GOSUB 20, 30, 60, 45$  – Ha az [X] értéke 1, akkor a 20-as sorszámmal kezdődő szubrutint hívja, ha 2, akkor a 30-as, ha 3, akkor a 60-ast, ha 4, akkor a 45-öt.

A szubrutin végrehajtása után, vagy ha [X] értéke 0, vagy 4-nél nagyobb az ON GOSUB utasítást követő utasítással folytatódik a program.

A kifejezés egészértéke nem lehet negatív és 255-nél nagyobb.

On kifejezés GOTO sorszámlista – feltételesugrás

A kifejezés egészértéke határozza meg, hogy a sorszámlista hányadik elemével jelölt sorszámon folytatódjon a program.

$ON X GOTO 100, 60, 85, 35$  – Ha az [X] értéke 1, akkor a 100-as, ha 2, akkor a 60-as, ha 3, akkor a 85-ös, és ha 4, akkor a 35-ös utasítással folytatódik a program. Akkor, ha az [X] értéke 0, vagy nagyobb mint 4, az ON GOTO-t követő utasítást hajtja végre a program.

– A kifejezés egész értéke nem lehet negatív és 255-nél nagyobb.

OUT kódszám, adat – adatkivitel regiszterekbe

A kódszámmal adott regiszterbe ír egy bájtot, amit decimálisan kell megadni.

$OUT 31,6$  – A 31-es regiszterbe ír 6-ot. Ezzel kiválasztotta a hanggenerátor 6-os regiszterét.

$OUT 30,15$  – A 31-es regiszter értékével meghatározott alregiszterbe ír 15-öt. Ezzel beállítja a zajgenerátor periódusidejét.

POKE cím, adat – beírás a memóriába

A megadott memóriacímre beírt egy bájtot. A címet és az adatot decimálisan kell megadni.

$POKE 15000,B$  – B változó értékét beírja a 15000-es című memóriarekeszbe. A változó 0-tól 255-ig vehet fel értékeket.

PRINT kiíratási lista – kiíró utasítás

Kiírja a kiíratási lista elemeinek értékét.

Listaelem lehet: konstans, kifejezés, TAB (kifejezés)

TAB (kifejezés) – hatására a kifejezés értékének megfelelő karakterpozícióba állítja a kurzort az adott sorban.

A listaelemeket vesszővel vagy pontosvesszővel választjuk el.

Pontosvessző hatása: folyamatos a kiírás

Vessző hatása: következő zónába ír

PRINT "KERÜLET"; KE; TAB (40), "TERULET" ; TE – Kiírja azt, hogy KERULET és a KE változó értékét. A 40-es karakterpozíciótól pedig azt, hogy TERULET és a TE változó értékét.

PRINT @ kifejezés, kiíratási lista – kurzor elhelyezése

A kifejezés értéke megadja, hogy a kiírás a képernyő hányadik karakterpozíciójában kezdődjön. A kifejezés 0 és 1024 közötti értékeket vehet fel, mivel 16 sor van, egy sorban pedig 64 karakter.

PRINT @I\*64, "FOLDES" – Az I. sor elejére írja, hogy FOLDES

PRINT USING karakterlánc kiíratási lista – kiíratási formátum megadása.

A karakterlánc értékével megadott formátummal írja ki a lista elemeit. A listában csak numerikus kifejezések szerepelhetnek, ha a karakterláncban # jelek vannak és egy pont. A # jelekkel adjuk meg, hogy hány számjeggyel történjen a kiírás. Ponttal a tizedes pont helyét jelöltük ki. A \$ és a \* jelekkel azt határoztuk meg, hogy mi legyen a szám elején.

PRINT USING "##.#"; A,B – A és B változó két egész és egy tizedes értékét írja ki.

PRINT USING AS;A,B – Ugyanaz történik, mint az előző példánkban, ha AS = "##.#"

A\$ " \* ##.#" – \* jeleket ír a szám elé

A\$ " \$ ##.#" – \$ jelet ír a szám elé

AS " \$ \$ \* ##.#" – \* jeleket és \$ jelet ír a szám elé

PRINT USING BS;CS – Ha CS="ABCD" és BS="!" a kiírás A

BS="%" a kiírás AB

BS="%%" a kiírás ABC

A kiíratási lista elé pontosvesszőt kell tenni.

**RANDOM** – véletlenszám generátor

Előállítja az első véletlenszámot. A véletlenszámot az RND függvénnyel hívhatjuk.

Írjuk a programunk elejére RANDOM utasítást, ha az RND függvényt használjuk!

**READ** változólista – adatbeolvasás a DATA adatmezőből

A változólista elemeinek értékét a DATA adatmezőből olvassa be.

READ A %,X,A(I) – Az A%, X és az A(I) változók értékeit a DATA soron következő adatai adják.

– RESTORE utasítással kezdhetjük előlről az adatbeolvasást.

**REM** karakterek – megjegyzés

A REM utasításban tetszőleges karaktersorozatot, ún. magyarázó szöveget elhelyezhetünk. Ez csak a programlistában szerepel, a rendszer nem értelmezi. REM EZ EGY SZOVEG

**RESET** (kifejezés, kifejezés) – grafikus pont törlés

A kifejezések egész értéke a képernyő grafikus pontjainak koordinátáit adják. Az utasítás hatására az adott koordinátájú pont törlődik (sötét lesz).

RESET (15,I) – A (15,I) koordinátájú pont törlődik.

– A koordináták lehetséges értékei:

$0 \leq X \leq 127$

$0 \leq Y \leq 47$

RESTORE – beolvasás a DATA adatmező elejéről

A RESTORE utasítást követő READ utasítás a legkisebb sorszámú DATA adatmezőből olvassa be az adatokat.

RESUME sorszám – hibakezelő rutin vége

Az ON ERROR GOTO utasítás hatására történő ugrás után csak a RESUME utasítás végrehajtásával folytatódhat a program. Előtte meg kell szüntetni a hiba okát

RESUME – Ott folytatódik a program, ahol a hiba történt.

RESUME 40 – A 40-es sorszámú utasítással folytatódik a program

RESUME NEXT – A hibás sort követő utasításoron folytatódik a program

RETURN – szubrutin vége

Hatására a GOSUB vagy az ONGOSUB után következő utasítás végrehajtásával folytatódik a program.

SET (kifejezés, kifejezés) – képpont megjelenítés

A kifejezések egész értékei a képernyő grafikus pontjainak koordinátáit adják meg

Az utasítás hatására adott koordinátájú pont világos lesz.

SET(X,Y) – Az X,Y koordinátájú pontot megjeleníti.

A koordináták lehetséges értékei:

$$0 \leq X < 127 \quad 0 \leq Y \leq 47$$

– A (0,0) koordinátájú pont a képernyő bal felső sarkában van.

STOP – a program leállítása

Megáll a program. A rendszer ezt a BREAK IN és az utoljára végrehajtott sorszám kiírással jelzi.

– A CONT utasítás hatására folytatódik a program végrehajtása.

## Aritmetikai függvények

A függvények argumentumát  $X$ -szel jelöltük, ami tetszőleges aritmetikai kifejezés lehet.

**ABS(X)** – abszolútérték függvény

A függvény értéke:  $X$ , ha  $X \geq 0$  és  $-X$ , ha  $X < 0$

**ATN(X)** – arkusztangens függvény

A függvény értéke az a szög radiánban, amelyiknek a tangense  $X$ .

**CDBL(X)** kétszeres pontosságú ábrázolás

A függvény értéke:  $X$  értéke kétszeres pontossággal.

**CINT(X)** – egészrész függvény

A függvény értéke:  $[X]$ , azaz  $X$  értékénél nem nagyobb legnagyobb egész szám.

Értelmezési tartomány:  $-32768 \leq X \leq 32767$

**COS(X)** koszinusz függvény

A függvény értéke:  $\cos X$

$X$  értékeit radiánban kell megadni.

**CSNG(X)** – egyszeres pontosságú ábrázolás

A függvény értéke:  $X$  értéke egyszeres pontossággal.

**EXP(X)** exponenciális függvény

A függvény értéke:  $e^X$ , ahol  $e = 2,71828$

Értelmezési tartomány:  $X \geq 0$

**FIX(X)** egészrész függvény

A függvény értéke: a szám egész része

**INT(X)** egészérték függvény

A függvény értéke  $[X]$ , azaz  $X$  értékénél nem nagyobb legnagyobb egész szám.

**LOG(X)** – természetes alapú logaritmus függvény

A függvény értéke:  $\ln x$ , azaz a logaritmus alapszáma  $e = 2,71828$

Értelmezési tartomány:  $X > 0$

**RND(X)** – véletlenszám függvény

Értelmezési tartomány:  $0 \leq X \leq 32767$

A függvény értéke:

$[X] = 0$  esetén a  $[0, 1]$  intervallumból egyszeres pontosságú,

$X \neq 0$  esetén az  $[1, |X|]$  intervallumból választ ki egy véletlen egész számot:

$0 < |X| < 32768$

**SGN(X)** – előjelfüggvény

A függvény értéke:  $1$ , ha  $X > 0$

$0$ , ha  $X = 0$

$-1$ , ha  $X < 0$

**SIN(X)** – szinuszfüggvény

A függvény értéke:  $\sin X$

$X$  értékét radiánban kell megadni.

**SQR(X)** – négyzetgyökfüggvény

A függvény értéke:  $\sqrt{X}$

Értelmezési tartomány:  $X \geq 0$

---

TAN(X) – tangensfüggvény

A függvény értéke:  $\text{tg } X$

X értéke radiánban adható meg.

## Karakterlánc művelet és függvények

A példánkban szereplő változók értékei:

A\$ = "ABCDE"

B\$ = "XYZ"

V\$ = "1543"

K = 65 (65 az A karakter kódja)

karakterláncok összefűzése

C\$ = A\$ + B\$ – C\$ értéke: ABCDEXYZ

ASC (karakterlánc) – karakterlánc első karakterének kódja

L = ASC(A\$) – L értéke: 65

CHR\$ (kifejezés) – a kifejezés értékének megfelelő kódszámú karakter

C\$ = CHR\$(K) – C\$ értéke: A

LEN (karakterlánc) – a string hossza

N = LEN(A\$) – N értéke: 5

LEFT\$ (karakterlánc, kifejezés) – a kifejezés értékének megfelelő számú karakter a string elejéről

C\$ = LEFT\$(A\$,3) – C\$ értéke: ABC

MID\$ (karakterlánc, kifejezés, kifejezés) – a string adott elemétől adott számú karakter kiválasztása. Az első kifejezés értéke azt adja meg, hogy hányadik karaktertől, a második azt, hogy mennyit.

C\$ = MID\$(A\$,2,3) – C\$ értéke: BCD

RIGHT\$ (karakterlánc, kifejezés) – a kifejezés értékének megfelelő számú karakter a string végéről

C\$ = RIGHT\$(A\$,3) – C\$ értéke: CDE

STR\$ (kifejezés) – az aritmetikai kifejezés értékét stringként adja meg

C\$ = STR\$(654.65) – C\$ értéke: 654.65

STRING\$ (kifejezés, karakter vagy kódja) – a kifejezés értékének megfelelő számban előállítja a karaktert

C\$ = STRING\$(4,"A") – C\$ értéke: AAAA

C\$ = STRING\$(5,65) – C\$ értéke: AAAAA

VAL (numerikus karakterlánc) – a numerikus karakterláncot számmá alakítja

N = VAL(V\$) – N értéke: 1543

– A kifejezés alatt ebben a fejezetben tetszőleges aritmetikai kifejezést értsünk.



## Egyéb függvények

INKEY\$ — egy karakter beolvasása

V\$ = INKEY\$ — Az utasítás végrehajtásakor a begépelte karaktert elhelyezi a string változóba. A karakter nem jelenik meg a képernyőn.

INP (kódszám)- adatbeolvasás regiszterekből

A kódszámmal adott regiszter tartalmát olvassa be.

MEM — memória szabad területe

Értéke egyenlő a memória szabad rekeszeinek számával.

PRINT MEM — kiírja, hogy hány bájt a szabad memóriaterület.

PEEK (cím) — olvasás a memóriából

Értéke az adott című memóriarekesz értékét adja

POINT(kifejezés)— grafikus pont vizsgálat

A kifejezésekkel a képernyő koordinátáit adjuk meg. Ha az (X,Y) koordinátájú pontja a képernyőnek világít, az értéke -1, ha nem 0 lesz. Ez megfelel IGAZ, vagy HAMIS logikai ítéletnek.

IF POINT(X,Y) THEN RESET(X,Y) ELSE SET(X,Y)

– Ha világít az (X,Y) koordinátájú pont, törli, ha nem, megjeleníti a pontot.

– A koordináták lehetséges értékei:

$$0 \leq X \leq 127 \quad 0 \leq Y \leq 47$$

– A (0,0) koordinátájú pont a képernyő bal felső sarkában van.

USR (egész kifejezés) — gépszintű szubrutin hívása

Az egész kifejezéssel a gépszintű szubrutin bemenő paramétereit adjuk meg. Értéke a gépi szintű szubrutin kimenő paramétere lesz.

VARPTR (változó) — változó helye a memóriában

A változó értékét tartalmazó memóriarekeszek címeit határozhatjuk meg az utasítás segítségével.

## Karakterek és kódjaik

## Vezérlő karakterek

Kód	Funkció	Funkció
8	a kurzor egy pozícióval lép és ott törli a karaktert	
10 13	a következő sor elejére állítja a kurzort	
14	a kurzort megjeleníti	
15	a kurzort letiltja	
23	ritkített kiírás	
	Egy sorban 32 karakter írható. CLS utasítással vagy a CLEAR billentyű lenyomásával állítható vissza	
24	a kurzor egy pozícióval balra lép és ott törli a karaktert	
25	a kurzor egy pozícióval jobbra lép	
26	egy sorral lejjebb állítja a kurzort	
27	egy sorral feljebb állítja a kurzort	
28	a kurzort a képernyő bal felső sarkába állítja	
29	a kurzort a sor elejére állítja	
30	a kurzortól jobbra törli a sort	
31	a kurzortól kezdődően törli a képernyőt.	

## Megjelenő karakterek kódja

Kód	Karakter	Kód	Karakter
32	szóköz		
33	!	61	=
34	"	62	>
35	#	63	?
36	\$	64	@
37	%	65	A
38	&	66	B
39	'	67	C
40	(	68	D
41	)	69	E
42	*	70	F
43	+	71	G
44	,	72	H
45	-	73	I
46	.	74	J
47	/	75	K
48	0	76	L
49	1	77	M
50	2	78	N
51	3	79	O

52	4	80	P
53	5	81	Q
54	6	82	R
55	7	83	S
56	8	84	T
57	9	85	U
58	:	86	V
59	;	87	W
60	<	88	X
		89	Y
		90	Z

– módosított géptípuson 64 az É karakter kódja.

### Kisbetűk

Kód	Karakter	Kód	Karakter
97	a	110	n
98	b	111	o
99	c	112	p
100	d	113	q
101	e	114	r
102	f	115	s
103	g	116	t
104	h	117	u
105	i	118	v
106	j	119	w
107	k	120	x
108	l	121	y
109	m	122	z

– Kibővített rendszerben használhatjuk csak a kisbetűket. A bővített rendszer in-  
tása a SYSTEM utasítással történik.

Géptípusonként eltérő kódok

Első változat (jelszint kijelzős)

	alaphelyzet	bővített rendszer
91 vagy 123	[	91 [ 123 {
92 vagy 124	\	92 \ 124
93 vagy 125	]	93 ] 125 }
94 vagy 126	~	94 ~ 126 ~
95 vagy 127	–	95 – 127 ☼
96	@	96 @ 64 @

második változat

	alaphelyzet	bővített rendszer
01 vagy 123	↑	91 ↑ 123 [
02 vagy 124	Ö	92 Ö 124 ö
03 vagy 125	Á	93 Á 125 á
04 vagy 126	Û	94 Û 126 ü
05 vagy 127	~	95 ~ 127 ~
06	é	96 é

A bővített rendszer indítása a SYSTEM utasítással történik.

### Grafikus kódok

Grafikus kódok a CHR\$(kód) karakterlánc-függvénnyel állíthatók elő.

### Tabulátor kódok

CHR\$(N+192) függvény értéke N darab szóköz karakter.

$$0 < N < 63$$

### Grafikus kódok

128		144		160		176	
129		145		161		177	
130		146		162		178	
131		147		163		179	
132		148		164		180	
133		149		165		181	
134		150		166		182	
135		151		167		183	

136		8	152		24	168		56	184		56
137		7	153		25	169		47	185		57
138		6	154		26	170		42	186		58
139		5	155		27	171		33	187		59
140		4	156		28	172		26	188		60
141		3	157		29	173		21	189		61
142		2	158		30	174		16	190		62
143		1	159		31	175		11	191		63

73. ábra

## 9. HIBAÜZENETEK ÉS HIBAKÓDOK

Jel	Kód	A hiba jellege, forrása
BS	9	Az index nagyobb, mint amit a DIM utasításban megadtunk
CN	17	CONT utasítással nem folytatható a program végrehajtása
DD	10	Kétszer szerepel ugyanaz a változó a DIM utasításban
FC	5	A változó az értelmezési tartománynak nem eleme (negatív index, negatív szám négyzetgyöke, a SET utasítás argumentuma nem megengedett)
FD	22	Hibás adatbeolvasás magnóról
ID	12	Nem megengedett direkt utasítás
LS	15	A karakterlánc 255 karakternél hosszabb
MO	21	Operandus hiányzik
NE	1	FOR nélküli NEXT utasítás vagy nem megfelelő ciklusváltozó a NEXT utasításban
NR	18	Nincs RESUME utasítás az ON ERROR GOTO utasítás után
OD	4	Nincs adat. A READ vagy az INPUT utasítás számára nem adtuk meg az adatot, vagy nem eleget
OS	14	Nincs hely a karakterláncok részére Használjuk a CLEAR utasítást
OM	7	Nincs több tárolóhely: memória vagy betelt, vagy a CLEAR utasítással lefoglaltuk, sok az egymásba ágyazás (GOSUB, FOR-NEXT), túl hosszú a program
OV	6	Túl- vagy alulcsordulás. Egy szám vagy egy kifejezés értéke nagyobb vagy kisebb, mint a megengedett értékhatár
RG	3	GOSUB nélküli RETURN utasítás
RW	19	A RESUME utasítás nem az ON ERROR GOTO után következik
SN	2	Szintaktikus hiba: utasításszó hibás leírása, nem megengedett karakter használata, rossz zárójelezés stb.
ST	16	Túlságosan összetett karakterlánc kifejezés
TM	13	Karakterláncok és numerikus adatok vegyes használata
UE	20	Nincs olyan hibakód, amit az ERROR utasításban használunk
UL	8	Nem létező sorra hivatkoztunk
H/Ø	11	Kísérlet nullával való osztásra

Széchenyi Nyomda, Győr 85. K-434  
Felelős nyomdavezető: Nagy Iván igazgató