SZÁMÍTÁSTECHNIKAI FÜZETEK

DUSZA ÁRPÁD:

TV-COMPUTER BASIC NYELVE
(Szakköri füzet kezdőknek)

MPI BORSOD-ABAÚJ-ZEMPLÉN MEGYEI PEDAGÓGIAI INTÉZET

D U S Z A Á R P Á D:

TV-COMPUTER BASIC NYELVE
(Szakköri füzet kezdőknek)

M I S K O L C
1 9 8 8

A sorozatot szerkeszti:
B. Lázár István

Borító:
Tellinger István

Lektorálta:
Füle lajos és
Kosztolányi László

A kötetet szerkesztette:
B.Lázár István

Kiadja a Borsod-Abaúj-Zemplén Megyei Pedagógiai Intézet
Felelős kiadó: Götz György igazgató
Készült a Megyei Pedagógiai Intézet sokszorosításában 1988-ban
1.200 példányban
Megjelenik 3 (A/5) ív terjedelemben
ISSN 0238-1176
ISBN 963-01-8931-3

Előszó helyett

javaslatok, tanácsok a programozás tanításához,

a füzet és a programcsomag használatához

tanároknak:

- Mint ahogyan az általános- és középiskolai matematikatanításnak sem célja, hogy matematikusokat képezzen, a programozást sem azért kell(ene) tanítani, hogy mindenkiből programozó legyen. Programozással (programozgatással) kezdődhet az a tanulási folyamat, ami a számítógép megismeréséhez, a számítógép alkalmazásához vezethet. Természetesen nem csak ez az út járható. Én ehhez szeretnék segítséget nyujtani azzal,hogy bemutatom a TV-Computer BASIC programozási nyelvét, a TV-Computer lehetőségeit.
- Ne utasításkészletet tanítsunk, hanem számítógépes feladatmegoldást. Ennek csak egyik eszköze a BASIC nyelv.
- A programozás mindig valamilyen feladatmegoldást jelent. Egy-egy utasítás, függvény, vagy programozási fogás bemutatása mindig kapcsolódjon egy olyan problémához, amelyiknek a megoldásához szükség van az új ismeretre.
- Keressünk a feladatok megoldására több megoldást. A füzetben szereplő programok is csak a lehetséges megoldások egyikét jelentik.
- A számítógép 'adatokat kér', 'kiírja az eredményt', 'relációt vizsgál', 'valahányszor, vagy valameddig ismételten csinál valamit', stb. Ezeknek a műveleteknek a BASIC nyelvű, - szintaktikailag helyes - megfogalmazása csak gyakorlat kérdése. Egy korábbi kész program megtekintése vagy maga a számítógép is -azzal, hogy ellenőrzést végez- segitségünkre lehet a feladat megfogalmazásában, a program megírásában. Készíttessünk feljegyzéseket, szoktassuk tanulóinkat a segédanyagok használatához!

- A füzet anyagának feldolgozása többféleképpen is elképzelhető:

  - A Számítógép kezelése és a Parancsok c. fejezetek megismertetnek a számítógép kezelésével és sok olyan utasítással - parancsként használva -, amelyeket később alkalmazni tudunk a programok készítésenél.

  - A Programok c. fejezetet bármelyik témakörrel elkezdhetjük, és bármelyikkel folytathatjuk.

  - Kiválaszthatunk egy témakört és a parancsok után folytathatjuk a programokkal. Például a 2.2. fejezet után következhet a 3.2. fejezet.

- Minél előbb ültessük a tanulókat a számítógép mellé! Az a legjobb, ha rögtön ezzel kezdjük.

- Adjunk feladatot a tanulóknak és segítsünk nekik abban, hogy önállóan, a számítógép segítségére támaszkodva tudjanak dolgozni!

- Időnként redszerezzük a tanultakat! Ehhez nem szükséges a számítógép.

- A számítógépes munkát előzze meg a feladat megoldásához szükséges ismeretanyag feldolgozása, az algoritmus megfogalmazása.
- Az algoritmusokat adjuk meg szöveges formában, pontokba szedve, és (vagy) folyamatábra segítségével! A füzet korlátozott terjedelme miatt csak néhány esetben, példaként mutatom ezt be.

- Szoktassuk tanulóinkat a tervszerű munkára! Írassunk programot papíron is!

- Adjunk tanulóink kezébe szakkönyveket! Ezt a füzetet se zárjuk el előlük.

A programcsomagról:

- A füzet 'hosszabb' programjai, azok továbbfejlesztett változatai, valamint a programozás tanításához, oktatóprogramok írásához felhasználható programok, szubrutinok alkotják a programcsomagot.
A programok nevét a feladatok után adjuk meg. Például: 3130. Ez a 3.1.3. fejezet D feladatának a megoldása lesz.
A füzetben szereplő megoldás továbbfejlesztett változatainak és a feladathoz kapcsolódó programok nevében csak betűk szerepelnek. Például:RENDEZ.
A Mellékletben röviden ismertetjük azokat a programokat, amelyek már túllépnek a füzet keretein. Remélem, hogy lesz amelyik ötletet, segítséget ad a tehetséggondozáshoz, oktatóprogramok készítéséhez és segít a programozás tanításában. A füzetben ott is utalunk ezekre a programokra ahol ezek alkalmazása, bemutatása javasolt.

tanulóknak :

- Tervezd meg munkádat! Gondold végig, hogy mi a feladatod! Akkor ülj a számítógéphez, ha már tudod, hogy mit akarsz csinálni!

- Gépeld be a kész programokat, és próbáld ki azokat!

- Próbáld meg önállóan megoldani a feladatot! Miután megírtad a programod, hasonlítsd össze a megoldással!

- Értelmezd a kész programokat! Futtasd le a programcsomag programjait!

- Keress más megoldást!

- Gondold meg, hogy milyen feladatok megoldásához használhatnád fel a tanultakat!

- Készíts feljegyzéseket! Írd le tapasztalataidat, ötleteidet, kérdéseidet!

- Próbálkozz, kísérletezz!

Miskolc, 1987. december

Jó munkát kíván a Szerző

1. Számítógép kezelése

1.1. A billentyűzet

Miután bekapcsoltuk a TV-t és a számítógépet, a képernyőn megjelenik egy bejelentkező kép. A gyártó cég és a számítógépünk nevét látjuk a képernyőn. Ezek után le kell nyomni egy tesz̋leges billentyűt és elkezdhetjük a munkát.
A TV és a billentyűzet segítségével teremthetünk kapcsolatot a számítógéppel. A billentyűzeten begépelt szöveg megjelenik a képernyőn, és ott láthatjuk a számítógép üzeneteit is.
A képernyőre akármit írhatunk. A számítógép csak akkor próbálja meg értelmezni a begépelt szöveget, amikor lenyomjuk a RETURN billentyűt.
Ekkor a következők történhetnek:
1./ A számítógép nem érti, hogy mit akarunk. Ilyenkor valamilyen hibajelzést ír ki. Próbáljuk ki! Írjuk be hogy

MENNYI 5+4

és nyomjuk le a RETURN billentyűt!

2./ Valamilyen értelmes parancsot adtunk a számítógépnek és a számítógép valamit csinál.
Most gépeljük be azt, hogy

PRINT 5+4

és nyomjuk le a RETURN billentyűt!

A számítógép elvégzi az összeadást.

3./ A számítógép nem csinál semmit. Legalábbis nem látványosan. Nézzük meg, hogy mit gépeltünk be! Ha a sor elején szám volt, akkor a gépeljük be azt,hogy

LIST

A LIST parancs után is nyomjuk le a RETURN billentyűt!

A számítógép kiírja azt,amit begépeltünk. Most már nagybetűkkel. A számítógép tehát megőrizte ezt a sort.

- Milyen ismeretek szükségesek a billentyűzet használatához?

- A fehér billentyűk lenyomásával kisbetűket, számjegyeket, műveleti- és írásjeleket írhatunk a képernyőre.

- A SHIFT billentyűt lenyomva kell tartani, ha nagybetűt, vagy felső karaktert (például + jelet) szeretnénk begépelni.

- A botkormánnyal mozgathatjuk a kurzort oda, ahová írni szeretnénk.

- A DEL billentyű lenyomásával a kurzortól balra lévő karaktert lehet törölni.

- A SHIFT/DEL (Miközben a SHIFT billentyűt lenyomva tartjuk,

nyomjuk le a DEL billentyűt!) a kurzortól jobbra törli a karaktert.

- Az INS billentyűvel helyet csinálhatunk új karaktereknek, amelyeket be szeretnénk 'szúrni' valahová.

- A CTRL/Y törli azt a sort ahol a kurzor áll.

Ennyi ismerettel már megpróbálhatjuk a következő program begépelését. A nagybetűk helyett írhatunk kisbetűket is.
A sorok végén nyomjuk le a RETURN billentyűt!

```
   10 REM  --- billentyüzet ---
   20 CLS:PRINT AT 3,3,"Nyomd le a megfelel
ö":PRINT"  billentyüt!"
   30 RANDOMIZE:J=0:R=0
   40 K=RND(125)+33
   50 PRINT AT 12,16,CHR$(K)
   60 IF K=127 OR K=143 THEN 40
   70 GET Q$
   80 IF Q$=CHR$(K) THEN 100
   90 PRINT AT 16,12,"Tévedtél!":R=R+1:GOTO
  110
  100 J=J+1:PRINT AT 16,13,"Helyes!"
  110 FOR I=1 TO 300:NEXT
  120 PRINT AT 16,1,STRING$(20,32)
  130 IF J<20 THEN 40
  140 CLS:PRINT AT 12,1,"20-szor sikerült,"
;R;"-szer nem"
```

A programot úgy indíthatjuk el,hogy begépeljük a

RUN

parancsot. (A RETURN billentyű lenyomásáról ne feledkezzünk meg!)

Ezek után két dolog lehetséges:

1./ A program 'fut', és kéri a karaktereket. 20 jó válasz után értékeli teljesítményünket.

2./ Valamit rosszul gépeltünk be. A hibát ki kell javítani.

Mit kell ekkor tenni?
- A hibajelzés után kiírt hibás sorba be kell írni a kimaradt karakter(eke)t, vagy törölni kell, ami felesleges,(A javítás után ne feledkezzünk meg a RETURN billentyű lenyomásáról!)

- vagy újból beírjuk az egész sort.

a./ Állítsuk le a programot futás közben! Mit ír ki a számítógép?

b./ Jegyezd fel, hogy milyen hibák fordulhatnak elő!

c./ Mi történik, ha egy utasítássort újból begépelünk?

1.2. Programok kezelése

Két paranccsal már az előző fejezetben találkoztunk:

LIST - kiírja a képernyőre a programot,

RUN - elindítja a programot. Ez azt jelenti, hogy a számítógép elkezdi programban szereplő utasítások végrehajtását.

Akkor, ha előttünk, már dolgozott valaki a számítógépen, a

NEW - paranccsal törölhetjük az előző programot. (Előtte érdemes megnézni a LIST paranccsal a programot!)

Van amikor nem akarjuk a teljes programot törölni. Ilyenkor a

DELETE 20-50

paranccsal törölhetünk sorokat. Példánkban a 20-as sortól a 50-es sorig töröltük az utasítássorokat.

a./ Hogyan lehet egy programsort törölni?

b./ Mit kell tennünk, ha kifelejtettünk egy utasitássort a programból?

c./ Mi történik, ha kiadod a TRACE ON és a RUN parancsokat?

d./ Program futása alatt, miközben lenyomod a CTRL billentyűt, nyomd le az ESC billentyűt is!
Az előzőekben szó volt arról, hogyan kell begépelni egy programot és azt hogyan lehet javítani, módosítani.

- Mit kell csinálni akkor, ha a programot kazettáról akarjuk beolvasni?

Adjuk ki a

LOAD

parancsot, és indítsuk el a lejátszásra állított magnetofont! Arra kell ügyelni, hogy a program elején álljunk amikor a beolvasás elkezdődik.
Ha nem sikerült a programot beolvasni, állítsunk a hangerőn és a hangszínen, és próbáljuk meg újból a beolvasást.

e./ Mi történik a LOAD "iskola" parancs hatására?

Programunkat a

SAVE "billentyu"

paranccsal vihetjük ki kazettára. Mielőtt lenyomnánk a RETURN billentyűt, indítsuk el a felvételre állított magnetofont.

f./ Hogyan ellenőrizhetjük, hogy sikerült-e a felvétel?

g./ Hogyan olvashatjuk be a "billentyű" névvel ellátott programot?

2. Parancsok

2.1. Kiírás a képernyőre

2.1.1. Számok és karakterláncok

A. Gépeld be a következő parancsokat és nyomd le a RETURN billentyűt! Figyeld meg, hogy mi történik! Mit csinál a számitógép?

```
PRINT 1987

PRINT "Miskolc"

CLS
```

Amennyiben helyesen írtuk be a parancsokat, és a RETURN billentyű lenyomásakor a kurzor abban a sorban volt amelyikbe a parancsot írtuk, akkor

- az első parancs: kiírja a képernyőre azt, hogy 1987. Ezt a számot írtuk a PRINT szó után.
- a második parancs: kiírja azt a szót, hogy Miskolc. Ezt írtuk az idézőjelek közé.
- a harmadik parancs: törli a képernyőt.

A PRINT parancs oda írja a parancsban megadott számot, vagy az idézőjelek közé írt szöveget, ahol a kurzor áll. A RETURN billentyű lenyomásakor a kurzor a következő sor elejére kerül, tehát ide történik a kiírás.

B. Írassuk ki a számjegyeket! Milyen elválasztójeleket használunk? Miben különböznek a kiírások?

```
PRINT 0,1,2,3,4,5,6,7,8,9

PRINT 0;1;2;3;4;5;6;7;8;9

PRINT "0123456789"
```

A pontosvessző hatására egymás mellé kerülnek az adatok, csak egy üres hely marad ki (a pozitív számok esetén kettő, de ebből az egyik a pozitív előjelé).

Vessző esetén az 1.,9.,17.,stb. karakterpozíciótól kezdődik a kiírás, ezért a képernyőn egymás alá kerülnek a kiírt eredmények. A harmadik parancsban a számjegyeket karakterláncban adtuk meg. A kiírás úgy történik, ahogyan azt az idézőjelek közé írtuk.

a./ Milyen számokat írnak ki a parancsok?

```
PRINT -34.56,455.789,-0.012

PRINT 23.4E-3,1E10,-0.0012E2,6.67E23
```

b./ Írassuk ki a -6-nál nagyobb negatív egész számokat! Hány üres hely lesz a számok között, ha a számokat pontosvesszővel választjuk el egymástól a parancsban?

c./ Írassuk ki ábécé sorrendben a magánhangzókat! A magánhangzók között legyen egy szóköz karakter(üres hely)!

d./ Fogalmazd meg, hogy a PRINT parancs mit jelent a számítógép számára!

C. Gépeld be a következő parancsokat és figyeld meg, hogy mi történik! Mi a szerepe az első két számnak?

```
PRINT AT 9,15,"BASIC"

PRINT AT 9,20,1987
```

A PRINT AT paranccsal bárhová írhatunk a képernyőre. A parancsszó utáni első számmal a képernyő sorát, a másodikkal a soron belül a karakter helyét adjuk meg. A kurzor ezzel a két számmal megadott helyre kerül és itt kezdődik a kiírás.

2.1.2. Változók

A. Adjunk értéket a VÁROS$,TÁVOLSÁG és EGYSÉG$ változóknak!

A RETURN billentyű lenyomásáról ne feledkezz meg!

```
VÁROS$="Budapest"

TÁVOLSÁG=180

EGYSÉG$="km"
```

A parancsok olvasása:
- 'A VÁROS karakterlánc-változó legyen egyenlő Budapest-tel!'
- 'A TÁVOLSÁG változó értéke legyen 180!'
- 'Legyen km karakterlánc az EGYSÉG$ (dollár) karakterlánc-változó értéke!'

A parancs végrehajtását úgy ellenőrizhetjük, hogy kiíratjuk a VÁROS$,TÁVOLSÁG és az EGYSÉG$ változók értékeit.

```
PRINT VÁROS$;TÁVOLSÁG;EGYSÉG$
```

a./ Írjuk be kisbetűvel is a parancsokat!
b./ Hol különbözteti meg a számítógép a kis-és nagybetűt?

B. Nézzük meg mi lesz az értéke a DARAB numerikus- és a GYÜMÖLCS$ karakterlánc-változónak?

```
PRINT DARAB;GYÜMÖLCS$;" FA"

PRINT DARAB,GYÜMÖLCS$," FA"
```

A két változónak korábban nem adtunk értékeket, ezért 0 ill. 'üres karakter' lesz az értékük.

c./ Mi különbözteti meg a numerikus-változót a karakterlánc-változótól?

d./ Milyen neveket adhatunk a változóknak? Milyen karakter lehet az első karakter? Milyen hosszú lehet egy változó neve? Próbáljuk ki különböző változónevekkel!

C. Gépeljük be a következő parancsokat!

```
CLS

A$="Basic"

SOR=10

PRINT AT SOR,15,A$
```

Vigyük vissza a kurzort a harmadik parancsra és írjunk 1-gyel nagyobb számot az egyenlőség jobb oldalára! Miután beírtuk a számot, nyomjuk le kétszer a RETURN billentyűt! Ezzel módosítottuk a SOR változó értékét, és ismét végrehajtattuk a számítógéppel a PRINT AT parancsot. A BASIC szót újból kiírja, de már az eggyel nagyobb sorszámú sorba, az előző alá. Ismételjük meg az előzőeket!

e./ Írassuk ki a BASIC szót többször egymás mellé!

f./ Mi történik a

```
GRAPHICS 2
GRAPHICS 4
GRAPHICS 16
```

parancsok hatására? Milyen értékeket adhatunk az egyes GRAPHICS parancsok után, a PRINT AT utasításban a SOR és OSZLOP változóknak? Kísérletezz!

g./ Mi az azonosság és mi a különbség a CLS és a GRAPHICS parancsok között?

h./ Fogalmazd meg szavakkal az értékadó parancs jelentését!

D. Adjunk meg egyszerre több parancsot!

```
SOR=10:OSZLOP=12

CLS:PRINT AT SOR,OSZLOP,A$
```

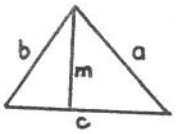
A parancsok közé kettőspontot kell tenni!

i./ Írasd ki bekeretezve, a képernyő közepére a MAGYARORSZÁG szót!

2.2. Számolás

2.2.1. Számok

A. Adott egy háromszög. Mérjük le a szükséges adatokat és számítsuk ki a háromszög kerületét és területét!
A mérés eredménye:
a=2,8 cm;b=2,3 cm ;c=3,1 cm;m=2,2 cm.



Gépeljük be a következő parancsokat és minden parancs után nyomjuk le a RETURN billentyűt!

```
PRINT 2.8+2.3+3.1
PRINT 3.1*2.2/2
```

A számítógép a PRINT parancsra kiszámítja és kiírja a képernyőre az eredményt. Miután tudjuk azt, hogy a háromszög kerületét és területét hogyan kell kiszámítani, nem nehéz kitalálni, hogy a parancsokban milyen műveleteket jelöltünk a + * és / karakterekkel.
A kivonást az oldalhosszak különbségének kiszámításával mutatjuk be:

```
PRINT 3.1-2.8,3.1-2.3,2.8-2.3

PRINT 3.1-2.8;3.1-2.3;2.8-2.3
```

a./ Milyen karakterekkel jelöljük a négy aritmetikai alapműveletet?

b./ Hogyan írjuk a tizedes számokat?

c./ Milyen karakterekkel választottuk el a PRINT parancsban az aritmetikai kifejezéseket?

Ugyanúgy, ahogyan papíron vagy fejben számolunk, a számítógép is két számot ad össze, von ki egymásból, szoroz ill. oszt el egymással. Nézzük, hogyan történik a parancsokban szereplő kifejezések értékeinek kiszámítása!

```
PRINT 3*2*5*6

PRINT 30+22+51+65

PRINT 10.8-2-1.5+25

PRINT 12*8/6*5/25
```

- Az első parancs végrehajtása: az első számot szorozd a második számmal, az így kapott értéket szorozd a harmadik számmal, és az így kapott értéket szorozd a negyedik számmal.
- Második parancs: az első számhoz add hozzá a második számot, az így kapott értékhez a harmadik számot, é.í.t.
- A harmadik és negyedik parancs végrehajtása csak annyiban tér el az első kettőtől, hogy itt már kivonás és osztás is szerepel a kifejezésben. Egy-egy kifejezésben azonos sorrendiségű műveletek szerepelnek. Ezek végrehajtása az előzőek szerint történik.

Különböző sorrendiségű műveletek esetén először a szorzás-osztás, majd az összeadás-kivonás végrehajtására kerül sor.

```
PRINT 3*3*3+2*2*2

PRINT 3+3*3-2*2+2

PRINT 3+3/3-2/2*2

PRINT 3/3*3-2+2*2
```

Amennyiben hatványozás is szerepel a kifejezésben, akkor először ezt végzi el a számítógép, ezután a szorzást-osztást (balról - jobbra haladva), végül az összeadást - kivonást (balról - jobbra haladva). Mielőtt kiszámoltatnánk valamelyik kifejezés értékét, számoljuk ki papíron is!

```
PRINT 2*8+6*3-25

PRINT 3^2+4^2

PRINT 2*3^2+4^2/32
```

2.2.2. Változók

A. Határozzuk meg az ABC derékszögű háromszög kerületét és területét!
Mért adatok:
a=2,7 cm;b=3,6 cm;c=4,5 cm.

A kerület és terület számértékének meghatározása:

```
A=2.7:B=3.6:C=4.5
PRINT A+B+C,A*B/2
```

Először értékeket adunk a változóknak. Az értékadó utasítások hatására az A,B és C változók felveszik az oldalhosszak számértékét. A PRINT parancsban ezeken a változókon jelöljük ki a műveleteket.

a./ Helyes eredményt kapunk, ha megváltoztatjuk A,B vagy C értékét?

B. Számítsuk ki az r=12,3 cm sugarú kör kerületét és területét!

```
EGYSéG$=" cm"

R=12.3

PRINT "A kör kerülete=";2*r*PI;EGYSéG$

PRINT "A kör területe=";R^2*PI;EGYSéG$"^2"
```

Most a kiírandó szöveg egy részét is karakterlánc-változókban adtuk meg. A pi változónak a számítógép ad értéket. Próbáljuk ki!

b./ Milyen neveket adhatunk a változóknak?

c./ Határozzuk meg az a=12,5 dm és b=137,5 cm oldalhosszúságú téglalap kerületét és területét!

2.2.3. Függvények

A. Számítsuk ki pitagorasz-tétel segítségével a 2.2.2./A feladatban megadott derékszögű háromszög átfogóját a két befogó ismeretében! Hasonlítsuk össze a mért értékkel!

```
PRINT SQR(2.7^2+3.6^2)
```

Az SQR betűhármassal a négyzetgyök függvényt jelöljük. A függvény a betűhármast követő zárójelben megadott kifejezés értékének a négyzetgyökét határozza meg.

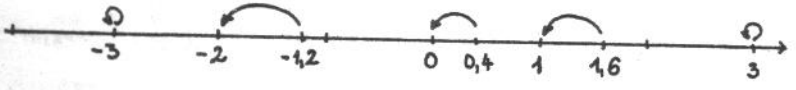
a./ X milyen értékeinél vannak értelmezve a megadott négyzetgyökfüggvények?

```
PRINT SQR(X)

PRINT SQR(X-2)

PRINT SQR(2-X)

PRINT SQR(2-X^2)

PRINT SQR((2-X)/(X+2))
```

B. Az X,Y és Z változók értékeinek változtatásával határozzuk meg, hogy mit csinál a három függvény!

```
PRINT ABS(X)

PRINT INT(Y)

PRINT SGN(Z)
```

Az ABS függvény az X változó értékének abszolutértékét, az INT függvény az Y változó értékénél nem nagyobb egészszámok közül a legnagyobb egész számot adja, az SGN függvény a Z változó előjelét jelzi -1,0 vagy 1 értékkel.
Az INT függvényről egy kicsit szemléletesebben: ha ábrázoljuk a számegyenesen Y érték, és Y értéke egész, akkor az INT függvény értéke Y. Ha Y tizedesszám, akkor a tőle balra eső legközelebbi egész szám lesz az INT függvény értéke.



b./ Határozzuk meg 12,87 és 6,32 egész ill. egytizedes pontosságú kerekített értékeit!

c./ Határozzuk meg az x1 és x2 változók értékeinek távolságát a számegyenesen!

2.3. Grafika

2.3.1. Pontok, szakaszok

A. Töröljük a képernyőt a CLS paranccsal, és gépeljük be a következő parancsokat!

```
PLOT 400,550

PLOT 300,400;800,600
```

A PLOT parancs után írt számpár tagjai a képernyő pontjainak koordinátáit jelentik. Egy számpárral egy pont jelenithető meg a képernyőn. Az első számmal a vízszintes, a másodikkal a függőleges koordinátát adjuk meg.
Szakaszt úgy rajzolunk, hogy a szakasz végpontjainak koordinátái közé pontosvesszőt teszünk.
A pontokat jelentő számpárok elé vagy mögé vesszőt vagy pontosvesszőt írhatunk. A számpár tagjai közé viszont mindig vesszőt kell tenni.
- vessző: 'felemeljük a ceruzát'
- pontosvessző: 'letesszük a ceruzát'
Nézzünk néhány példát:

```
PLOT 300,400;500,600

PLOT ;200,700

PLOT 330,750;
```

- első parancs: kirajzoljuk (300,400) koordinátájú pontot, a 'ceruza' a papíron marad és elmegyünk a (500,600) koordinátájú pontig. Röviden: rajzolunk egy szakaszt, amelynek a végpontjai a (300,400) és az (500,600) pontok.
- második parancs: 'letesszük a ceruzát és elmegyünk a (200,700) pontig. Röviden: húzunk egy szakaszt a megadott pontig.
- harmadik parancs: kijelöljük a (330,750) pontot és letesszük a ceruzát.
- negyedik parancs: rajzolunk egy háromszöget.
A harmadik parancs után a negyedik parancsban már felesleges volt az első pont koordinátáját megadni, mivel a harmadik parancsban ezt a pontot már kijelöltük, és a 'ceruzát' is letettük (a végére írt ;-vel).

a./ Nézd meg, hogy mi történik, ha a GRAPHICS 2, vagy a GRAPHICS 16 parancs után gépeled be az előző parancsokat!
b./ Keretezd be a képernyőt!
c./ Bekapcsolás után mit csinál PLOT ;400,600 parancs?

B. Rajzoljunk egy derékszögű háromszöget, és fessük be!

A 'ceruzát' a háromszög belsejébe visszük, és kiadjuk a PAINT parancsot.

```
PLOT 200,300;700,300;200,600;200,300
PLOT 250,350,PAINT
```

d./ Mi történik ha a 'ceruza' a háromszögön kívül van, amikor a PAINT parancs végrehajtására kerül sor?

C. Rajzoljunk szaggatott vonalakkal egy szakaszt!

```
SET STYLE 5
PLOT 250,350;800,620
```

A SET STYLE utasításban szereplő számmal a kirajzolandó vonaltípust határozzuk meg.

e./ Hány különböző vonaltípussal lehet rajzolni?
f./ Mi a különbség a CLS és a GRAPHICS utasítások között?
g./ Mi történik a PLOT 400,500:PRINT#0,"Basic" parancsok hatására?

2.4 Karakterláncok

2.4.1. Karakterlánc-műveletek

A. Az a$ karakterlánc-változónak értéket adunk és kiíratjuk a karakterlánc

- 10. karakterét,
- 5. karakterétől a karakterlánc 13. karakteréig a karaktereket,
- 3. karakterétől kezdődően a karaktereket,
- 15. karakteréig a karaktereket.

```
a$="abcdefghijklmnopqr"
print a$(10)
print a$(5:13)
print a$(3:)
print a$(:15)
```

a./ Milyen hosszú lehet egy karakterlánc?

B. Az a$ karakterlánc karaktereit 'összefűzzük' és kapunk egy szót. Ez lesz b$ értéke.
A 'láncolás' vagy 'összefűzés' jele: & ('és' jel).

```
b$=a$(4:5)&a$(2)&a$(18)&a$(5)&a$(3)&a$(5)&
a$(14)
print b$
```

C. Mi történik, ha az értékadó utasítás bal oldalán jelöljük ki a karakterlánc egy szeletét?

```
a$="abcdefghijk"
a$(3)="xyz"
print a$

a$="abcdefghijk"
a$(3:8)="xyz"
print a$

a$="abcdefghijk"
a$(2:)="xyz"
print a$

a$="abcdefghijk"
a$(:5)="xyz"
print a$
```

a$(3) - a harmadik karakter,
a$(3:8) - a 3. és 8. karakter közötti szelet
a$(2:) - a 2. karaktertől kezdődő szelet,
a$(:5) - az 5. karakterig tartó szelet helyére kerül az egyenlőség jobb oldalán álló karakterlánc.

2.4.2. Karakterlánc-függvények

A. A LEN függvény a karakterlánc hosszát adja meg. Nézzük meg, hogy hány karakterből áll az a$ karakterlánc!

```
print len(a$)
```

B. Minden karakterhez tartozik egy szám. Ez a szám a karakter kódja. Azt, hogy az egyes karaktereknek mi a kódja, megnézhetjük a Mellékletben, de meghatározhatjuk az ORD függvénnyel is.

```
print ord(a$)

print ord("a")

print ord("alma")
```

Mindhárom parancs ugyanazt az eredményt adja, mivel a karakterláncok első karektere az 'a' karakter. Az ORD függvény a karakterlánc első karakterének kódját adja meg.

C. Most a kódból állítjuk elő a karaktert a CHR$ függvénnyel. A két parancs ugyanazt az eredményt adja.

```
print chr$(65);crh$(66);chr$(67)

print "A";"B";"C"
```

D. Töröljük ki a képernyö 10 sorát a képernyő bal szélétől a képernyő közepéig!

A STRING$ függvénnyel előállítunk 16 db szóköz karaktert és azt kiíratjuk a 10.sor elejére.

```
print at 10,1,string$(16," ")

print at 10,1,string$(16,32)
```

a./ Mit csinál a következő parancs?

```
print at 1,1,string$(24,25)
```

E. Két számot akarunk összeadni, amiket karakterláncokként adtunk meg. Mi a teendő?

A karakterláncokat a VAL függvénnyel át kell alakítani numerikus adatokká, és ezekkel számolunk.

```
x$="1987":y$="12"

print val(x$)+val(y$)
```

b./ Mi lesz a VAL függvény értéke akkor, ha a karakterláncban nem csak számjegy karakterek szerepelnek?

```
print val("12A3")
print val("12e3")
print val("12*3")
print val("-123")
```

F. Határozzuk meg, hogy 2 tizenhatodik hatványa hány számjegyből áll!

Kiszámoljuk a hatványt, és átalakítjuk karakterlánccá az STR$ függvénnyel. Ennek hosszát a LEN függvénnyel határozzuk meg.
A feladat megoldható egyetlen paranccsal is, ha a függvényeket egymásba ágyazzuk.

```
x=2^16
x$=str$(x)
print ord("alma")

print len(str$(2^16))
```

c./ Milyen hosszú az str$(5-8) karakterlánc?

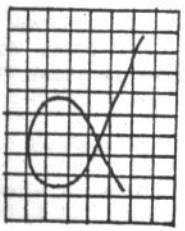
d./ Melyik parancs ad nagyobb értéket?

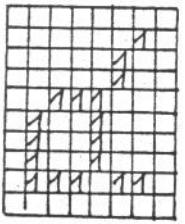
```
print len(str$(2e4))

print len(str$(2e24))
```

2.4.3. Karakterek definiálása

A. Írassuk ki a képernyőre a görög ábécé első betűjét, az alfát!

A 128-as kódszámtól kezdődően új karakterek adhatók meg a SET CHARACTER paranccsal. A karakterek egy 10 sorból, soronként 8 pontból álló téglalap pontjaiból alakíthatók ki.
Nézzük az alfa betű tervezésének lépéseit!





0
2
4
4
32+16+8
64+8
64+8
64+8
64+32+16+4+2
0

```
SET CHARACTER 129,0,2,4,4,32+16+8,64+8,64+
8,64+8,64+32+16+4+2,0

SET CHARACTER 129,0,2,4,4,56,72,72,72,118,
0
```

A sorokat egy nyolc számjegyből álló kettes számrendszerbeli számnak tekintjük. Amelyik pontnak világítania kell, oda 1-et, a többi helyre Ø-t írunk.
Az így kapott számokat át kell alakítani a tízes számrendszerbe. Ezek a számok szerepelnek a SET CHARACTER parancsban.
Az é billentyű lenyomására, vagy a PRINT CHR$(129) parancs hatására az alfa betű megjelenik a képernyőn.

a./ Melyik billentyű lenyomásával lehet kiíratni a 16Ø ill. a 177 kódszámú karaktert?

2.5. Színek

A./ Nézzük meg, hogy milyen színeket ismer a számítógépünk! Változtassuk a SET BORDER parancsban a számot!

Ez a parancs a keretet színezi be, a megadott kódszámú színnel. A kódszámok és szinek táblázatát a Mellékletben megtaláljuk.

```
SET BORDER 21
```

a./ Miben különbözik a SET BORDER 2Ø és a SET BORDER 84 parancs?

B. Vizsgáljuk meg, hogy a GRAPHICS 16 parancs után, azaz a 16 szín(képernyőn milyen színeket használhatunk?

```
GRAPHICS 16
SET PAPER 10:SET INK 15
PRINT AT 14,15,"BASIC"
PLOT 450,400;610,400
```

Változtassuk a SET INK és a SET PAPER parancsban a színek sorszámait. A sorszámokhoz tartozó színeket a Mellékletben megadjuk. (A színek sorszáma más lesz, mint a kódszámuk.)

C. Hogyan tudjuk a négyszínű képernyőn, azaz a GRAPHICS 4 parancs után megoldani az előző feladatot?

A SET PALETTE parancsban felsoroljuk azoknak a színeknek a kódszámait, amelyeket használni szeretnénk. A SET PAPER és a SET INK utasításban Ø,1,2 vagy 3 értékekkel adhatjuk meg, hogy hányadik színt választjuk a 'papir' ill. a 'ceruza' színének.

```
GRAPHICS 4
SET PALETTE Ø,8Ø,68,85
SET PAPER 2:SET INK 3
PRINT AT 14,15,"BASIC"
PLOT 450,400;610,400
```

b./ Hogyan kell megadni a GRAPHICS 2 parancs, azaz a kétszínű képernyő esetén a színeket.

c./ Hogyan lehet átszínezni a teljes képernyőt?

d./ Fessünk be egy zöld alapra, fekete vonalakkal rajzolt háromszöget pirosra!

2.6. Hangok

Hangkeltéshez a hang magasságát, a hangerőt és a hang időtartamát kell megadni.
A SOUND szó után a

PITCH - a hangmagasság,

DURATION - a időtartam,

VOLUME - a hangerő megadására szolgál.

Próbáljuk ki a következő parancsokat, és értelmezzük azokat!

```
sound volume 15

sound duration 100

sound pitch 3349
```

Első parancs: a legnagyobb hangerővel 1 másodpercig szól a középső C hang.

Második parancs: 2 másodpercig szól a középső C hang.

Harmadik parancs: közepes hangerővel 1 másodpercig szól a középső C hang.

Az előzőekből megállapíthatjuk, hogy alaphelyzetben egy közepes (7-es) erősségű, 2 másodperces, középső C hangra van beállítva a hanggenerátor. Ez a hang szól ha csak a SOUND parancsot adjuk ki.

```
sound volume 15,pitch 4000

sound volume 15;pitch 4000
```

Első parancs: egy magas hangot hallunk a legnagyobb hangerővel.

Második parancs: a C és a magas hang 1-1 másodpercig szól. Amikor vesszőt használunk az elválasztáshoz, akkor csak a paraméter változik. Pontosvesszőnél megszólal az alapállapotban adott C hang is.

```
sound pitch 4000,volume 1,duration 10

sound pitch 4000,volume 1,duration 10;dura
tion 100,pitch 3349
```

Első parancs: rövid ideig halkan egy magas hangot hallunk.

Második parancs: az előző hang után 2 másodpercig hallható a C hang. A pontosvessző után kezdődik a másik hang.

```
sound volume 3,duration 30,pitch 3349;pitc
h 3431;pitch 3503;pitch 3537;pitch 4095;pi
tch 3598;pitch 3652;pitch 3701
```

Ez a középső skála az egész hangokkal. A hangmagassághoz tartozó paramétereket a 3.6. fejezetben találjuk meg.
A SOUND PITCH 4095 paranccsal szünetet adhatunk meg.

3. Programok

3.1. Kiírás a képernyőre

3.1.1. Változók

A. Írjuk ki a képernyő közepére azt, hogy Miskolc-1987!

Parancsokkal a megoldás:

```
graphics 4
SOR=12:OSZLOP=10
szöveg$="Miskolc -"
print at sor,oszlop,szöveg$;1987
```

Írjunk sorszámokat a parancsok elé! A RETURN billentyű lenyomásakor azt tapasztaljuk hogy, a parancsokat nem hajtja végre a számítógép. A számítógép tárolja a sorszámokkal kezdődő sorokat.

```
10 graphics 4
20 sor=12:oszlop=10
30 szöveg$="Miskolc -"
40 print at sor,oszlop,szöveg$;1987
```

Arról, hogy mi került a számítógépbe, úgy győződhetünk meg, ha kiadjuk a LIST parancsot. Amennyiben sikerült a gépelés, akkor a következőket látjuk:

```
10 GRAPHICS 4
20 SOR=12:OSZLOP=10
30 SZöVEG$="Miskolc -"
40 PRINT AT SOR,OSZLOP,SZöVEG$;1987
```

Ez egy program. Minden program utasítások sorozata. Eddig parancsként használtuk a programban szereplő utasításokat.
A parancsokat a számítógép azonnal végrehajtja, amint lenyomjuk a RETURN billentyűt.
A program, vagyis a programban szereplő utasítások végrehajtására csak akkor kerül sor, ha kiadjuk a RUN parancsot.
Indítsuk el a programot a RUN paranccsal!

Programunk a következő utasításokból áll:

10-es sor:
A képernyőt töröljük, és közepes finomságúra állítjuk.

20-as sor:
A SOR és OSZLOP numerikus változóknak értékeket adunk. Ebben a sorban két utasítás van, amelyeket kettőspont választ el.

30-as sor:
A SZÖVEG$ karakterlánc változó értéke a Miskolc szó lesz.

40-es sor:
A képernyő 12. sorának 10. karakterhelyétől kezdődően kiírja a SZÖVEG$ változó értékét és az 1987 számot. A SOR és OSZLOP változók értékétől függ, hogy hová történik a kiírás. Azt, hogy mit írjon ki, utána adjuk meg.

a./ Próbáld ki a GRAPHICS 2 és a GRAPHICS 16 utasításokkal is a

programot! Írd felül a 10-es sort!

b./ Változtasd meg a SOR és OSZLOP változók értékeit úgy, hogy a kiírás a GRAPHICS 2 és a GRAPHICS 16 utasítások használatakor is a képernyő közepére kerüljön!

c./ Írasd ki a nevedet!

d./ Hány sor és soronként hány karakter fér el a képernyőn?

e./ Milyen neveket adhatunk a változóknak?

f./ Miért sorszámoztuk tízesével az utasítássorokat?

B. Hogyan lehet a program módosítása nélkül értéket adni a változóknak?

Erre szolgál az INPUT utasítás. Írjuk felül, ill. 'szúrjunk be' újabb utasítássorokat az előző programba!

```
10 CLS:PRINT "Képernyö(2,4 vagy 16)";
20 INPUT K
30 PRINT "Szöveg";:INPUT Q$
40 PRINT "sor,oszlop";
50 INPUT SOR,OSZLOP
60 GRAPHICS K
70 PRINT AT SOR,OSZLOP,Q$
```

A képernyő törlése, és az útmutató szöveg kiírása után a program először a K változó értékét kéri. Ezt egy kérdőjel kiírásával jelzi. A lehetséges értékek közül valamelyiket beírjuk és lenyomjuk a RETURN billentyűt.
Ezek után azt a szöveget kell begépelni amit ki akarunk íratni. A szöveg beírása után is le kell nyomni a RETURN billentyűt.
A következő INPUT utasítás a SOR és az OSZLOP változók értékeit kéri. A két adat közé vesszőt kell tenni. Az adatok bevitele, most is a RETURN billentyű lenyomására fejeződik be.

g./ Milyen értékeket vehetnek fel a K, a SOR és az OSZLOP változók?

h./ Aláhúzással írasd ki a szöveget!

C. Oldjuk meg az előző feladatot az INPUT PROMPT utasítással is!

```
10 CLS
20 INPUT PROMPT"Képernyö(2,4 vagy 16)":K
30 INPUT PROMPT"Szöveg":Q$
40 INPUT PROMPT"sor,oszlop":SOR,OSZLOP
60 GRAPHICS K
70 PRINT AT SOR,OSZLOP,Q$
```

Amint látjuk a PRINT utasítások elmaradtak. A szöveget az INPUT PROMPT utasítás írja ki.

i./ Hol használjuk a kettőspont karaktert?

j./ Hasonlítsd össze az INPUT és az INPUT PROMPT utasításokat!

3.1.2. Vizsgálat

A. A kiírás kerüljön a képernyő közepére. Függetlenül attól, hogy milyen képernyőt használunk!.
A program neve: 312A

```
   5 CLS
  10 INPUT PROMPT "Képernyö(2vagy4vagy16)"
:K
  12 SOR=12
  15 IF K=2 THEN OSZLOP=23
  17 IF K=4 THEN OSZLOP=9
  20 IF K=16 THEN OSZLOP=3
  25 GRAPHICS K
  30 SZOVEG$="Miskolc"
  40 PRINT AT SOR,OSZLOP,SZOVEG$;1987
```

A sorok száma mindhárom betűméretnél 24, ezért csak az OSZLOP változó értéke függ K értékétől. Az IF utasításokkal megvizsgáljuk, hogy K változónak milyen értéket adtunk. Ennek megfelelően állítjuk be az OSZLOP változó értékét.
Ha az IF utasításban szereplő reláció értéke 'igaz', akkor a számítógép végrehajtja a THEN után álló utasításokat, majd a következő utasítássorral folytatódik a program végrehajtása.
Amennyiben a reláció értéke 'hamis' akkor a következő utasítássorral folytatódik a program.

B. Javitsunk az előző programon! Ellenőrizzük K értékét! Akkor, ha K értéke nem megfelelő, a program kérje újból!
A program neve: 3128

A megoldásban azt használjuk ki, hogy ha a relációk közül bármelyik is 'igaz', akkor több vizsgálatra nincs szükség. El lehet ugrani 25-ös sorra, és kezdődhet a kiírás.
Amennyiben egyik reláció sem teljesedik, vissza kell menni a 10-es sorra.

```
   5 CLS
   8 INPUT PROMPT"Név: ":NéV$
  10 INPUT PROMPT "Képernyö(2vagy4vagy16)"
:K
  12 SOR=12
  15 IF K=2 THEN KöZéP=32:GOTO 25
  17 IF K=4 THEN KöZéP=16:GOTO 25
  20 IF K=16 THEN KöZéP=8:GOTO 25
  22 GOTO 10
  25 GRAPHICS K
  30 OSZLOP=KöZéP-LEN(NéV$)/2
  40 PRINT AT SOR,OSZLOP,NéV$
```

a./ Mi az összefüggés a színek száma és a betűk mérete között!

b./ Mi a jelentése a GOTO utasításnak ?

c./ Oldd meg az előző feladatot GOTO utasítás nélkül!

d./ Milyen relációjeleket ismersz?
e./ Mi a jelentése a LEN függvénynek? Itt mi a szerepe?

C. Írjunk programot, amivel karakterekből ábrát rajzolhatunk a képernyőre!
A programok neve: 312C,RAJZKAR

A GET A$ utasítás egyetlen karaktert olvas be a billentyűzetről. Azt, amelyik billentyűt lenyomtuk.
Kiválasztunk négy karaktert, amivel a mozgás irányát megadjuk. Legyenek ezek az L,F,B és J karakterek. Azt kell megvizsgálnunk, hogy melyik billentyűt nyomtuk le, és ennek megfelelően kell változtatnunk a kiírás helyét.
A szóköz billentyűvel azt adhatjuk meg, hogy rajzolni, vagy törölni akarunk-e.

```
  5 SET CHARACTER 128,255,255,255,255,255
,255,255,255,255,255
 10 GRAPHICS 2
 20 I=12:J=30:T=1
 30 GET A$
 40 IF A$="j" THEN J=J+1
 50 IF A$="b" THEN J=J-1
 60 IF A$="l" THEN I=I+1
 70 IF A$="f" THEN I=I-1
 80 IF A$=" " THEN T=-T
 90 IF T=1 THEN PRINT AT I,J,CHR$(128)
100 IF T=-1 THEN PRINT AT I,J," ";
110 GOTO 30
```

f./ Hogyan működik a GET A$ utasítás?
g./ Mi a szerepe a T változónak?
h./ Mi a CHR$ függvény jelentése?
i./ Mi történik, ha a X billentyűt nyomjuk le?
j./ Mit tudsz a SET CHARACTER utasításról?

D. Készítsünk játékprogramot az előző programból!
Programok neve: 312D,UTKOZES

A botkormány mozgatásával úgy kell irányítani a kiírást, hogy az ne menjen le a képernyőről.
Amikor a program leáll, írassuk ki a PONT változó értékét. Értéke a 'befutott' helyek számát adja.

```
  5 SET CHARACTER 128,255,255,255,255,255
,255,255,255,255,255
 10 GRAPHICS 2
 20 I=12:J=30:T=1
 30 W$=INKEY$:IF W$<>"" THEN A$=W$
 40 IF A$=CHR$(4) THEN J=J+1
 50 IF A$=CHR$(19) THEN J=J-1
 60 IF A$=CHR$(24) THEN I=I+1
 70 IF A$=CHR$(5) THEN I=I-1
 80 IF A$=" " THEN T=-T
 90 IF T=1 THEN PRINT AT I,J,CHR$(128)
100 IF T=-1 THEN PRINT AT I,J," ";
105 PONT=PONT+1
110 GOTO 30
```

k./ A program működését akkor érted meg, ha válaszolni tudsz a következő kérdésekre:

1./ Mi a különbség a GET utasítás és az INKEY$ függvény között?

2./ Mi az az 'üres karakter' és hogyan jelöljük?

3./ Mi a CHR$ függvény jelentése?

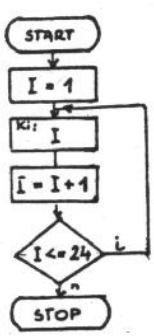
1./ Egészítsük ki a programot! Amikor lefutottunk a képernyőről akkor a program írja ki a PONT változó értékét!

3.1.3. Ciklus

A. Írassuk ki a Borsod szót egymás alá, amennyiszer csak lehet!

24 sor van, tehát 24-szer kell ismételten végrehajtani a programnak a PRINT utasítást, - amivel a kiírás történik - és megoldottuk a problémát. Próbáljuk ki!

```
10 CLS
20 I=1
30 PRINT I,"Borsod"
40 I=I+1
50 IF I<=24 THEN 30
```



A programban egy ciklust állítottunk elő, amelyik 24-szer ismétli a kiírást. Az I változó értékét minden kiírás után 1-el növeljük. Ezzel számláljuk, hogy hányadik kiírásnál tartunk.
Az IF utasítással azt vizsgáljuk, hogy I értéke nem lépi-e túl a megadott határt.
Jobb oldalt a program folyamatábrája.

a./ Oldd meg, hogy minden sorban látható legyen a kiírás!
b./ Kezdjük a kiírást a legalsó sorban!

c./ Töröljük a kiírás után az előző sort!
d./ Írasd ki egymás mellé a nevedet!

B. Oldjuk meg az előző feladatot a FOR/NEXT ciklusképző utasítással is!

```
10 CLS
20 FOR I=1 TO 24
30 PRINT I,"Borsod"
40 NEXT I
```

A FOR és a NEXT utasítások közé írt utasítások végrehajtása ismétlődik 24-szer. Az I értéke ismételten növekszik 1-gyel, és így jutunk el 1-től 24-ig.
Akkor, ha a lépésköz nem 1, meg kell adnunk az értékét a FOR utasításban a STEP szó után.

```
10 CLS
20 FOR I=24 TO 1 STEP -1
30 PRINT AT I,16,I;"Borsod";
40 NEXT I
50 GET
```

e./ Írjunk minden második sorba!

C. Haladjon végig a képernyőn egy 'autó'!
A programok neve: 313C,AUTO

Az 'autót' természetesen karakterekből alakítjuk ki, ahogyan ezt a 2.4. fejezetben leírtuk.

```
 1 REM   --- autó ---
 5 GRAPHICS 4
10 SET CHARACTER 128,0,0,31,16,112,64,12
7,20,8,0
20 SET CHARACTER 129,0,0,224,16,14,2,254
,40,16,0
28 !
29 !
30 FOR I=1 TO 30
40 PRINT AT 15,I," ";CHR$(128);CHR$(129)
45 ::::FOR J=1 TO 100:NEXT
50 NEXT
```

f./ Mi a szerepe a REM utsításnak és a felkiáltójeleknek ?
g./ Húzzon az 'autó' egy 'utánfutót'!
h./ Tegyük folyamatosabbá a mozgást a PLOT és a PRINT#0, utasításokkal!

D. 'Építsünk' különböző magasságú 'házakat' egymás mellé!
A programok neve: 31302,SOMBAZAS

Több 'házat' úgy tudunk 'építeni', ha a 'házépítő' programrészletet (belső ciklust) többször végrehajtjuk. Egy újabb ciklussal - a külső ciklussal - a 'ház' helyét változtatjuk.

```
 1 REM   --- Házak ---
 5 CLS
10 SET CHARACTER 128,255,129,129,129,129
,129,129,129,129,255
30 RANDOMIZE
34 !
35 FOR I=1 TO 30
40 :EMELET=RND(22)
50 : FOR J=23 TO 24-EMELET STEP -1
60 :  PRINT AT J,I,CHR$(128)
70 : NEXT
75 NEXT
76 !
80 PRINT AT 1,1
```

i./ Készítsünk oszlopdiagram-rajzoló programot!

3.2. Számolás

3.2.1. Változók

A. Számítsuk ki a 2,8 és 3,6 egység befogójú derékszögű háromszög kerületét és területét!

Gépeld be a következőket! A sorok végén nyomd le a RETURN billentyűt!

```
10 a=2.8:b=3.6
20 c=sqr(a^2+b^2)
30 print a+b+c,a*b/2
```

A programunkat a számítógép eltárolja. Erről meggyőződhetünk, ha a begépeljük a LIST parancsot. A program kiírása nagybetűvel történik.

```
10 A=2.8:B=3.6
20 C=SQR(A^2+B^2)
30 PRINT A+B+C,A*B/2
```

Ha a RUN paranccsal elindítjuk a programot, megkapjuk az eredményt.
A program három utasítássorból áll. Az utasítássorok sorszámára hivatkozva a program értelmezése:
10-es sor:
A és B változóknak értéket adunk. A két utasítást kettősponttal választjuk el.
20-as sor:
Kiszámítjuk az átfogót. A C változó értékét - pitagorasz-tétel segítségével - úgy kapjuk meg, hogy az A és B változók négyzetének összegéből négyzetgyököt vonunk.
30-as sor:
Kiszámítjuk és kiíratjuk a háromszög kerületét és területét.

a./ Számítsd ki a 3 és 4 ill. a 153,6 és 48,3 egység befogójú derékszögű háromszögek kerületét és területét!

b./ Hogyan számítjuk ki a derékszögű háromszög kerületét és területét, ha adott az átfogó és az egyik befogó?

3.2.2. Adatok bekérése

A. Írjuk át a 10-es sort és indítsuk el a programot!

```
10 INPUT A,B
20 C=SQR(A^2+B^2)
30 PRINT A+B+C,A*B/2
```

A számítógép kérdőjel kiírásával jelzi, hogy kéri az INPUT utasításban felsorolt változók értékeit. Gépeljük be a két befogó hosszát úgy, hogy a két szám közé tegyünk vesszőt és végül nyomjuk le a RETURN billentyűt.
Előző programunk nem ad semmiféle tájékoztatást arról, hogy mit kér és mi az amit kiszámított a számítógép. Most a PRINT utasítással, az adatok bekérésénél és az eredmény kiírásakor,

magyarázatokat is kiíratunk.

```
10 CLS:PRINT "Befogók=";:INPUT A,B
20 C=SQR(A^2+B^2)
30 PRINT "Kerülete=";A+B+C
40 PRINT "Területe=";A*B/2
```

Az INPUT PROMPT utasítással a változók értékeinek bekérésén kívül a szöveget is kiírathatjuk.

```
10 INPUT PROMPT "Befogók ":A,B
20 C=SQR(A^2+B^2)
30 PRINT "Kerülete=";A+B+C
40 PRINT "Területe=";A*B/2
```

A PRINT USING után #(kettőskereszt) karakterekkel az eredmény formátumát adjuk meg. A kettőskeresztek a kiírandó egész ill. a törtrész számjegyeinek számát határozzák meg.

```
10 CLS:INPUT PROMPT"Befogók=":A,B
20 C=SQR(A^2+B^2)
30 PRINT "Kerülete=";:PRINT USING"######.
##":A+B+C
40 PRINT "Területe=";:PRINT USING"######.
#":A*B/2
```

a./ Milyen különbséget látsz az INPUT és az INPUT PROMPT utasítások között?

b./ Adott egy kocka alakú test oldalélének hossza és sűrűsége. Számítsd ki a kocka tömegét!
Hogyan számítanád ki papíron?

c./ A PRINT AT utasítás alkalmazásával kérjük az adatokat a képernyő közepén, és az eredményt is oda írassuk ki!

3.2.3. Vizsgálat

A. Adott a derékszögű háromszög átfogója és befogója. Számítsuk ki a háromszög kerületét és területét!
Program neve: 323A

Gondoljuk végig a feladatot!

1./ A számítógépnek be kell kérnie két számot. Az egyik az átfogó, a másik a befogó hossza lesz.

2./ Ki kell számítani a másik befogót.

3./ A három oldal ismeretében már meghatározható a kerület és a terület.
A két beolvasott szám, azaz az átfogó és a befogó értéke nem lehet tetszőleges. Az átfogónak nagyobbnak kell lennie, mint a befogónak. A számítógéppel megvizsgáljuk,hogy tényleg ilyen értékeket adunk-e meg.
A 40-es sorban az IF utasítás biztosíja, hogy ha a C értéke kisebb vagy egyenlő, mint A értéke, akkor a PRINT utasítással kiírja azt, hogy 'rossz adatok', és a GOTO utasítással visszaugrunk a 20-as sorra, ahol a program újból kéri az adatokat.

```
10 CLS
20 INPUT PROMPT "Átfogó=":C
30 INPUT PROMPT "Befogó=":A
40 IF C<=A THEN PRINT "rossz adatok":GOT
O 20
50 B=SQR(C^2-A^2)
60 PRINT "Kerülete=";A+B+C
70 PRINT "Területe=";A*B/2
```

a./ Álljon meg a program, ha rossz adatokat adtunk meg!

b./ Az oldalak hosszát pozitív számokkal adjuk meg. Milyen vizsgálat szükséges az adatok beolvasásakor?

B. Írassuk ki szöveggel azt, hogy milyen értéket adtunk X változónak!
Program neve: 323B
A programban azt kell vizsgálni, hogy negatív, nulla vagy pozitív, valamint, hogy egész vagy tört számot adtunk-e meg.

```
10 CLS
20 INPUT PROMPT "A szám=":X
30 IF X=0 THEN PRINT "nulla":STOP
40 IF X<0 THEN PRINT "negativ";
50 IF X>0 THEN PRINT "pozitiv";
60 IF X=INT(X) THEN PRINT " egész":END
70 PRINT " tört"
```

c./ Milyen relációk, logikai műveletek szerepelhetnek az IF utasításban?

d./ Fogalmazd meg az IF utasítás jelentését!

e./ Mi a különbség a STOP és az END utasítások között?

C. Vizsgáljuk meg, hogy két szám távolsága a számegyenesen nagyobb-e, mint egy adott szakasz hossza!
Néhány megoldás az ABS függvény alkalmazásával:

```
10 CLS
20 INPUT PROMPT"A két szám:":A,B
30 INPUT PROMPT"A szakasz:":E
40 IF ABS(A-B)>E THEN PRINT"nagyobb":END
50 PRINT "nem nagyobb"
```

```
10 CLS
20 INPUT PROMPT"A két szám:":A,B
30 INPUT PROMPT"A szakasz:":E
40 IF ABS(A-B)<=E THEN PRINT"nem nagyobb
":END
50 PRINT "nagyobb"
```

```
10 CLS
20 INPUT PROMPT"A két szám:":A,B
30 INPUT PROMPT"A szakasz:":E
40 IF ABS(A-B)<=E THEN PRINT"nem ";
50 PRINT "nagyobb"
```

```
10 CLS
20 INPUT PROMPT"A két szám:":A,B
30 INPUT PROMPT"A szakasz:":E
35 PRINT "A két szám távolsága ";
40 IF ABS(A-B)>E THEN PRINT"nagyobb "
:ELSE PRINT "nem nagyobb"
45 PRINT " mint a szakasz hossza."
```

f./ Fogalmazd meg az utolsó megoldás jelentését!

g./ Oldjuk meg a feladatot az ABS függvény alkalmazása nélkül is!

D. Adott három szakasznak a hossza. Milyen háromszög szerkeszthető a szakaszokból?
Program neve: 3230

Ahhoz, hogy egy háromszög derékszögű legyen, Pitagorasz- tétel értelmében az szükséges, hogy két oldal négyzetösszegének egyenlőnek kell lennie a harmadik oldal négyzetével. Amennyiben az egyenlőség nem teljesedik, és két oldal négyzetének összege nagyobb lesz, mint a harmadik oldal négyzete, a háromszög hegyesszögű, egyébként a háromszög tompaszögű.
Az átfogó csak a legnagyobb oldal lehet. Ez lesz Z változó értéke. Ha a legnagyobb érték a X vagy Y változóban van, akkor az értéküket kicseréljük Y változó értékével. Ez történik a THEN szó után, a 40-es és 50-es utasításrokban.
A 70-es sorban azt is megvizsgáljuk, hogy egyáltalán lehet-e ilyen háromszög.

```
10 CLS
20 INPUT PROMPT"A három szám:":X,Y,Z
30 D$="derékszögü "
32 H$="hegyesszögü "
34 T$="tompaszögü "
40 IF X>Z THEN T=X:X=Z:Z=T
50 IF Y>Z THEN T=Y:Y=Z:Z=T
60 IF X+Y<=Z THEN PRINT "nem háromszög":
END
70 V=X*X+Y*Y-Z*Z
80 IF V=0 THEN PRINT D$;
90 IF V>0 THEN PRINT H$;
100 IF V<0 THEN PRINT T$;
110 PRINT "háromszög"
```

h./ Írj programot a következő függvények behelyettesítési értékeinek kiszámítására:

$$\frac{1}{x};\ \frac{1}{x-2};\ \sqrt{x};\ \sqrt{x-2}$$

3.2.4. Ciklus

A. Írassuk ki 1-től N-ig a természetes számokat!

Az I változó értékét mindig 1-el növeljük és az IF-THEN utasítással azt vizsgáljuk, hogy I értéke nem nagyobb-e mint N értéke. Ameddig ez teljesedik, visszaugrunk és újból végrehajtjuk a PRINT utasítást.

```
10 CLS
20 INPUT PROMPT"n=":N
30 I=1
40 PRINT I
50 I=I+1
60 IF I<=N THEN 40
```

Miután ilyen ismétlődést, azaz ciklust gyakran használunk, ezért erre külön utasítás is van. Ez a FOR/NEXT ciklusképző utasítás.

```
10 CLS
20 INPUT PROMPT"n=":N
30 FOR I=1 TO N
40 PRINT I
50 NEXT I
```

a./ Írasd ki N-ig a természetes számok négyzeteit, köbeit, reciprokait és négyzetgyökeit!

B. Írassuk ki növekvő, majd csökkenő sorrendben az N-nél nem nagyobb pozitiv páratlan számokat!

A ciklusváltozó értékét most 2-vel kell növelnünk, miután minden második számot íratjuk csak ki. Ezt a STEP szó után adjuk meg.

```
10 CLS
20 INPUT PROMPT"n=":N
30 FOR I=1 TO N STEP 2
40 PRINT I
50 NEXT I
```

A kezdőérték meghatározásához az szükséges, hogy megállapítsuk : N páros, vagy páratlan.

```
10 CLS
20 INPUT PROMPT"n=":N
30 IF N/2=INT(N/2) THEN N=N-1
40 FOR I=N TO 1 STEP -2
50 PRINT I
60 NEXT
```

b./ Írasd ki az N-től kisebb páros számokat!
c./ Keresd meg a hárommal osztható kétjegyű számokat!

d./ Vizsgáld meg a következő programmal a FOR/NEXT utasítás működését!

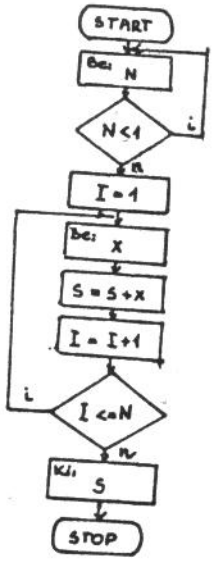
```
10 INPUT PROMPT"Kezdőérték:":K
20 INPUT PROMPT"Végérték:":V
30 INPUT PROMPT"Lépésköz:":L
40 FOR X=K TO V STEP L
50 PRINT X
60 NEXT X
```

Ábrázold számegyenesen a kezdő, a vég és a ciklusváltozó értékeit! Mikor fejeződik be a ciklus? Értelmezd a FOR/NEXT utasítást!

e./ Mikor nem kell a lépésközt megadni?
f./ Írj programot, amivel függvénytáblázatot készíthetsz!

C. Adjunk össze N darab számot!
Programok neve: 324C1,324C2

A FOR/NEXT ciklus vizsgálatakor észrevehettük, hogy a ciklust egyszer mindig végrehajtja a számítógép.
Ebben a feladatban a számok beolvasásához és azok összegzéséhez ciklust kell létrehoznunk. Amennyiben N értéke 0 akkor nincs szükség arra, hogy végrehajtsuk a ciklust. Erről külön gondoskodnunk kell.
A számok összegét az S változóban kapjuk meg, amit nullázunk kell mielőtt a ciklusba lépünk.
A program és a program folyamatábrája:




```
10 CLS
20 INPUT PROMPT"n=":N
30 S=0
40 IF N<1 THEN END
50 FOR I=1 TO N
60 INPUT PROMPTSTR$(I)&".szám:":X
70 S=S+X
80 NEXT
90 PRINT "összeg=";S
```

Talán egyszerűbb a következő megoldás, ahol a ciklusvizsgálatot a ciklus elején végezzük.

```
10 CLS
20 INPUT PROMPT"n=":N
30 S=0:I=1
40 IF N<I THEN 90
60 INPUT PROMPTSTR$(I)&".szám:":X
70 S=S+X:I=I+1
80 GOTO 40
81 !
90 PRINT "összeg=";S
```

g./ Készítsd el a program folyamatábráját!

h./ Szorozz össze N számot!

i./ Egy szám összes osztójának meghatározásához készültek a következő programok. Magyarázd meg a működésüket!

```
10 CLS
20 INPUT PROMPT"n=":N
30 FOR I=1 TO N
40 IF N/I=INT(N/I) THEN PRINT I
50 NEXT
```

```
10 CLS
20 INPUT PROMPT"n=":N
30 FOR I=1 TO N/2
40 IF N/I=INT(N/I) THEN PRINT I
50 NEXT
60 PRINT N
```

```
10 CLS
20 INPUT PROMPT"n=":N
30 FOR I=1 TO SQR(N)
40 IF N/I=INT(N/I) THEN PRINT I;N/I
50 NEXT
60 IF I*I=N THEN PRINT I
```

```
10 CLS
20 INPUT PROMPT"n=":N
30 FOR I=1 TO SQR(N)+.1
35 IF I*I=N THEN PRINT I:GOTO 50
40 IF N/I=INT(N/I) THEN PRINT  I;N/I
50 NEXT
```

D. Írjunk át egy tízes számrendszerben adott számot tetszőleges alapú számrendszerbe!
Programok neve: 324D,ATIRAS

Először nézzük meg a maradékos osztás programját!

```
10 REM  --- Maradékos osztás ---
20 CLS:INPUT PROMPT"Osztandó=":P
30 INPUT PROMPT"Osztó=":Q
40 H=INT(P/Q)
50 M=P-H*Q
60 PRINT P;"=";H;"*";Q;"+";M
```

Az átszámítás algoritmusa:

T - tízes számrendszerben adott szám
A - az új számrendszer alapja
H - hányados
M - maradék

1./ A T-t elosztjuk maradékosan A-val, és a maradékot leírjuk.
2./ A T felveszi a hányados értékét.
3./ A hányados 0?
        igen: folytasd a 4. ponttal!
        nem: folytasd az 1. ponttal!
4./ A maradékokat visszafelé leolvassuk!

A 80-as sorba írjuk be a PRINT M utasítást, és akkor a programunk

megegyezik a szövegesen megfogalmazott algoritmussal.

```
10 CLS:W$=""
20 PRINT"Tizes számrendszerben a szám:";
T
30 INPUT T
40 PRINT "A számrendszer alapja:";
50 INPUT A
60 H=INT(T/A)
70 M=T-H*A
80 W$=STR$(M)&W$
90 T=H
100 IF H>0 THEN 60
110 PRINT "A szám:";W$
```

F. Játszunk 'fej vagy írás' játékot a számítógéppel!
A 'pénzt ' a számítógép 'dobja fel'.

Az RND függvénnyel véletlenszámokat állítunk elő 0 és 1 között.
Az RND(2) függvény 0 vagy 1 értékeket ad véletlenszerűen.

```
10 REM  --- pénzdobálás ---
20 CLS:INPUT PROMPT"Hányszor?":N
30 RANDOMIZE
40 FOR I=1 TO N
50 V=RND
60 IF V<0.5 THEN PRINT "fej":ELSE PRINT
"irás"
70 NEXT
```

```
10 REM  --- pénzdobálás ---
20 CLS:INPUT PROMPT"Hányszor?":N
30 RANDOMIZE
40 FOR I=1 TO N
50 V=RND(2)
60 IF V<1 THEN PRINT "fej":ELSE PRINT "i
rás"
70 NEXT
```

k./ Mi történik, ha töröljük a RANDOMIZE utasítást?

l./ Egészítsük ki a programot úgy, hogy játszani lehessen a számítógéppel!

m./ Állítsunk elő TOTO tippeket!

3.2.5. Indexes változók

A. Olvassunk be N számot és írassuk ki azokat fordított sorrendben!
Programok neve: 325A,KARAKT
Az A indexes változók értékei lesznek a megadott számok. A változó neve után, a zárójelben szereplő index értékével különböztetjük meg a változókat

```
 10 REM ___ beolvasás-kiiras ___
 20 CLS:INPUT PROMPT"N=":N:DIM A(N)
 30 ! - beolvasás -
 40 FOR I=1 TO N
 50 INPUT PROMPT STR$(I)&".=":A(I)
 60 NEXT I
 70 ! - kiirás -
 80 FOR I=N TO 1 STEP -1
 90 PRINT I;".=";A(I)
100 NEXT I
```

a./ Mi a szerepe a DIM utasításnak?

B. Keressük ki N szám közül a legnagyobbat és a legkisebbet!
Program neve:3258

A beolvasott számokat sorban megvizsgáljuk és összehasonlítjuk az addig talált legnagyobb és legkisebb értékkel.
Úgy kezdjük, hogy a legnagyobb és a legkisebb érték is az első szám lesz. Az összehasonlítás során, amennyiben találunk nagyobb ill. kisebb számot, annak indexét megőrizzük a MAX ill. MIN változókban.

```
 10 REM --- maximum,minimum ---
 20 CLS:INPUT PROMPT"Adatok száma=":N
 30 DIM A(N)
 40 FOR I=1 TO N
 50 PRINT I;".=";
 60 INPUT A(I)
 70 NEXT
 75 !
 80 MAX=1:MIN=1
 90 FOR I=2 TO N
100 IF A(I)<A(MIN) THEN MIN=I
110 IF A(I)>A(MAX) THEN MAX=I
120 NEXT
130 PRINT "A legkisebb=";A(MIN)
140 PRINT "A legnagyobb=";A(MAX)
```

C. Egy versenyen minden versenyző maximum 10 pontot kaphat. A csapat átlagát úgy számítják ki, hogy a legjobb és a legrosszabb eredményt elérő versenyző pontszámát nem veszik figyelembe. Készítsünk programot, amivel ezt a számítást elvégezhetjük!
Program neve:325C

```
 10 REM --- átlag ---
 20 CLS:INPUT PROMPT"A versenyzök száma (
min.3):":N:DIM A(N)
 24 !
 25 REM --- Adatbeolvasás ---
 30 PRINT "Pontszámok (max. 10):"
 40 FOR I=1 TO N
 50 INPUT PROMPT STR$(I)&". vesenyzö:":A(
I)
 55 IF A(I)>10 OR A(I)<0 THEN 50
 60 NEXT
```

```
 70 REM  --- Legnagyobb,legkisebb ---
 80 MAX=1:MIN=1
 90 FOR I=2 TO N
100 IF A(I)<A(MIN) THEN MIN=I
110 IF A(I)>A(MAX) THEN MAX=I
120 NEXT
129 !
130 REM  --- Átlag ---
140 S=-A(MIN)-A(MAX)
150 S=0
160 FOR I=1 TO N
170 S=S+A(I)
180 NEXT
190 PRINT "Átlag=";
200 PRINT USING"##.##":S/(N-2)
```

D. Készítsünk programot, amivel LOTTO tippeket ad a számítógép!
Programok neve: 325D,LOTTO

A véletlenszerűen kihúzott számot meg kell jegyeznünk, hogy még egyszer ne húzhassuk ki. Az A indexes változók közül csak azoknak adok értéket, pl. 1-et, amelyiknek az indexe megegyezik a kihúzott számmal (70-es sor).
Az 50-es sorban állítjuk elő a LOTTO számokat, és ide ugrunk vissza a 60-as sorról, ha korábban már kihúztuk a számot, azaz ha A(T)=1.
A 90-es sortól kezdődik az a ciklus, amelyikkel kiíratjuk azokat az indexes változókat, amelyeknek értéke 1.

```
 10 REM --- LOTTO ---
 20 DIM A(90)
 30 CLS:RANDOMIZE
 40 FOR I=1 TO 5
 50 T=RND(90)+1
 60 IF A(T)=1 THEN 50
 70 A(T)=1
 80 NEXT
 90 FOR I=1 TO 90
100 IF A(I)=1 THEN PRINT I
110 NEXT
```

E. Adjunk össze két n számjegyből álló pozitív egész számot!
Programok neve: 325E,HATVANY

Nagyon nagy számokat csak úgy tudunk pontosan összeadni,ha a számok számjegyeit külön-külön adjuk meg és a papíron végzett összeadáshoz hasonlóan, helyiértékenként végezzük el az összeadást.
A számjegyek az A és B indexes változók értékei lesznek. A számjegyeket a DATA adatlistában adtuk meg, ahonnan a READ utasítással olvassuk be az indexes változókba. Az indexet a ciklussal változtatjuk.

Az azonos helyiértékű számok, azaz az azonos indexű számjegyek összegéhez még hozzáadjuk az átvitelt (V változó), amiből már meghatározható az összeg I. számjegye és az átvitel (130 és 140-es sorok).

```
   10 REM  --- nagy számok összeadása ---
   20 N=9:DIM A(N-1),B(N-1),C(N)
   30 REM --- számok beolvasása ---
   40 CLS:PRINT"    ";:FOR I=N-1 TO 0 STEP -
1
   50 READ A(I):PRINT A(I);
   60 NEXT:PRINT:PRINT "    ";
   70 FOR I=N-1 TO 0 STEP -1
   80 READ B(I):PRINT B(I);
   90 NEXT:PRINT
  100 REM --- összeadás ---
  110 V=0
  120 FOR I=0 TO N-1
  130 C(I)=A(I)+B(I)+V
  140 V=INT(C(I)/10):C(I)=C(I)-10*V
  150 NEXT
  160 C(N)=V
  170 REM  --- eredmény kiírása ---
  180 FOR I=N TO 0 STEP -1
  190 PRINT C(I);
  200 NEXT:PRINT
 1000 REM  --- a két szám ---
 1010 DATA 8,3,4,5,0,1,8,5,5
 1020 DATA 7,2,0,9,2,5,6,1,3
```

b./ Mit csinál a program? Írd le az algoritmusát!

```
   20 CLS:INPUT PROMPT "n:":N
   30 DIM A(N)
   40 FOR I=1 TO N
   50 INPUT PROMPT STR$(I)&". =":A(I)
   60 NEXT I
   70 !
   80 Q=0:I=1
   90 IF A(I)<=A(I+1) THEN 120
  100 W=A(I):A(I)=A(I+1):A(I+1)=W
  110 Q=1
  120 I=I+1:IF I<N THEN 90
  130 IF Q=1 THEN 80
  140 !
  150 FOR I=1 TO N
  160 PRINT I;".";A(I)
  170 NEXT I
```

3.3. Grafika

3.3.1. Változók

A. Keretezzük be a képernyőt!

Vonalakkal össze kell kötnünk a képernyő sarkaiban lévő pontokat. Gépeljük be a következő programot (a sorok végén nyomjuk le a RETURN billentyűt) és indítsuk el a RUN paranccsal!
Begépeléskor a program kisbetűkkel látható a képernyőn, de a LIST parancsra már nagybetűkkel írja ki a számítógép.

```
10 GRAPHICS 2
20 PLOT 0,0;1023,0;
30 PLOT 1023,958;0,958
40 PLOT ;0,0
50 GET
```

Az utasítások elején a számok az utasítássorok sorszámai. Az utasítások végrehajtása a sorszámok növekvő sorrendjében történik.

a./ Mi a szerepe a RETURN billentyűnek a programok begépelésekor!

b./ Miért van szükség a GET utasításra?

B. Adottak a téglalap bal felső és jobb alsó pontjainak koordinátái. A téglalap oldalai legyenek párhuzamosak a képernyő széleivel. Rajzoljuk meg a téglalapot!

Jelöljük a bal felső pont vízszintes koordinátáját A1-el, a függőleges koordinátáját A2-vel. A jobb alsó pont koordinátái legyenek B1 és B2.
Mindenek előtt meg kell határoznunk a másik két pont koordinátáit. Ezek: (A1,B2) és (B1,A2)
Az A1,A2,B1 és B2 változóknak az értékadó utasítással adunk értékeket. A PLOT utasítás már azokkal az értékekkel számol, amelyeket az egyenlőségek jobb oldalán megadtunk

```
10 REM  --- keret ---
20 GRAPHICS 4
30 A1=0:A2=958
40 B1=1023:B2=0
50 PLOT A1,A2;B1,A2;B1,B2;
60 PLOT A1,B2;A1,A2
70 GET
```

A pontok koordinátáit a RUN parancs után adhatjuk meg, -amikor már fut a program-,ha az adatokat az INPUT utasítással kérjük.
Az INPUT utasítás kéri a két pont koordinátáit. Ezeket vesszővel elválasztva begépeljük és befejezésül lenyomjuk a RETURN billentyűt.
Az INPUT elé PRINT utasításokat írtunk, amelyekkel kiíratjuk azt, hogy milyen adatokat kér a program. Ahhoz, hogy más is használhassa programunkat, erre mindig szükség van.

```
10 REM --- téglalap ---
20 CLS:PRINT "bal felso pont";
25 INPUT A1,A2
30 PRINT "jobb also pont";
35 INPUT B1,B2
40 GRAPHICS 16
50 PLOT A1,A2;B1,A2;B1,B2;
60 PLOT A1,B2;A1,A2
70 GET:GRAPHICS 4
```

Amihez az előzőekben két utasítást használtunk, azt egyedül az INPUT PROMPT utasítással is megoldhatjuk. Ez az utasítás kiírja a megadott szöveget és kéri a felsorolt változók értékét.

```
10 REM --- téglalap-2 ---
20 CLS: INPUT PROMPT"bal felso pont":A1,A
2
30 INPUT PROMPT"jobb also pont":B1,B2
40 GRAPHICS 16
50 PLOT A1,A2;B1,A2;B1,B2;
60 PLOT A1,B2;A1,A2
70 GET:GRAPHICS 4
```

c./ Hogyan jelzi a számítógép, hogy adatot vár tőlünk?

d./ Rajzolj téglalapot, ha adottak a bal felső pontjának koordinátái, továbbá az oldalak hossza!

e./ Hány adatot kell megadni
1./ a négyzet,
2./ a háromszög rajzolásához?

C. Toljunk el egy egyenlő oldalú háromszöget vízszintesen V1, függőlegesen V2 egységgel!
Program neve: 331C

A háromszöget úgy rajzoljuk meg, hogy megadjuk az egyik csúcspont koordinátáit és az oldal hosszát.
Ha a háromszög egyik oldala párhuzamos a képernyő valamelyik szélével, akkor ezekből az adatokból kiszámíthatjuk a háromszög másik két csúcsának koordinátáit.

```
10 REM  --- eltolás ---
20 GRAPHICS 4:INPUT PROMPT"a háromszög c
súcspontjának        koordinátái:":X,Y
30 INPUT PROMPT" oldalának a hossza:":A
40 GRAPHICS 2
50 PLOT X,Y;X+A,Y;X+A/2,Y+A*SQR(3)/2;X,Y
60 INPUT PROMPT "vizszintes eltolás=":V1
70 INPUT PROMPT "függöleges eltolás=":V2
80 X=X+V1:Y=Y+V2
90 PLOT X,Y;X+A,Y;X+A/2,Y+A*SQR(3)/2;X,Y
100 GET:GRAPHICS 4
```

f./ Mi történik, ha a 90-es sorba
1./ GOTO 50
2./ X1=X1+30:Y1=Y1+40:GOTO 70 utasításokat írjuk?

g./ Hogyan nagyíthatjuk a háromszöget?
h./ Tükrözzük a háromszöget a képernyőt felező egyenesre!

D. Rajzoljunk egy négyzetet és fessük be!
Program neve: 3310

A festést az átlók metszéspontjában kezdjük. A koordináták értékeit ki tudjuk számítani. Ez a pont biztos, hogy a négyzet belsejében van.

```
10 REM  --- négyzet ---
20 GRAPHICS 4
30 INPUT PROMPT"a négyzet csúcspontjának
     koordinátái:":X,Y
40 INPUT PROMPT"a négyzet oldalának hoss
za:":A
50 GRAPHICS 2
60 PLOT X,Y;X+A,Y;X+A,Y+A;X,Y+A;X,Y
70 PLOT X+A/2,Y+A/2,PAINT
80 GET:GRAPHICS 4
```

3.3.2. Vizsgálat

A. Készítsünk programot, amivel rajzolni lehet a képernyőre!
Programok neve 332A1,332A2,332A3,RAJZOL
Legegyszerűbben szakaszokból tudunk kialakítani valamilyen alakzatot. A szakaszok végpontjainak koordinátáit kell sorban megadni.
Mielőtt meghúznánk egy szakaszt, meg kell vizsgálnunk a begépelt adatokat. A pontnak a képernyőre kell kerülnie. Az X és Y változók értékei csak megadott határok között mozoghatnak.

```
10 REM    --- rajz ---
20 GRAPHICS 4
30 PRINT AT 23,8,CHR$(25);"koordináták";
40 INPUT X,Y
50 IF X<0 THEN 110
60 IF Y<0 THEN 110
70 IF X>1023 THEN 110
80 IF Y>958 THEN 110
90 PLOT X,Y;
100 GOTO 30
110 PRINT AT 23,8,CHR$(25);
```

a./ Hogyan lehetne felemelni a 'ceruzát'?

b./ Milyen relációjeleket ismersz?

Egy másik módja a rajzolásnak, amikor a billentyűk, vagy a botkormány segítségével irányítjuk a 'ceruza' mozgását.
A billentyűzetről GET utasítással beolvasott karaktert megvizsgáljuk és ha a karakter
j - jobbra,
B - balra,
f - fel,
l - lefelé húzunk vonalat;
i - 'letesszük a ceruzát',
n - 'felemeljük a ceruzát',
s - elindítjuk a festést.
A vizsgálatot az IF utasításokkal végezzük. Akkor, ha a reláció értéke 'igaz', a számítógép a THEN szó utáni utasítást hajtja végre, ezután megy a következő utasítássorra.

Amennyiben a reláció értéke 'hamis', akkor a következő utasítássor végrehajtására kerül sor.

```
10 REM --- rajz-1 ---
20 GRAPHICS 4:INPUT PROMPT"Kezdopont":X,
Y
30 SET RATE 1:GRAPHICS 2:T=1
40 PLOT X,Y
50 A$=INKEY$
60 IF A$="j" THEN X=X+10
70 IF A$="b" THEN X=X-10
80 IF A$="f" THEN Y=Y+10
90 IF A$="l" THEN Y=Y-10
100 IF A$="i" THEN T=1
110 IF A$="n" THEN T=0
120 IF A$="v" THEN 170
130 IF A$="s" THEN PLOT X,Y,PAINT
140 IF T=0 THEN PLOT  X,Y,
150 IF T=1 THEN PLOT ;X,Y
160 GOTO 50
170 PRINT AT 22,1
```

c./ Miben különbözik a GET utasítás az INKEY$ függvénytől?

d./ Értelmezd a CHR$ és az ORD függvényeket!

e./ Mi a szerepe az X1 és Y1 változóknak?

8. Rajzoljunk a képernyőre egy koordináta-rendszert, amelynek origója a képernyő közepére esik. A program kérje a megjelenítendő pontok koordinátáit.
Program neve: 3328

Mielőtt kirajzolnánk a pontot, megvizsgáljuk, hogy az a képernyőre kerülne-e (80-as és 90-es sorok).
A koordináta-rendszerünk és a képernyő (0,0) pontja nem esik egybe. A pont koordinátáit át kell számolnunk a képernyő koordináta-rendszerébe (100-as sor).
A program akkor fejeződik be, ha a pont első koordinátájának a 9999 értéket adjuk.

```
10 REM --- koordináta-rendszer-1 ---
20 GRAPHICS 4
30 PLOT 0,480;1023,480
40 PLOT 510,50;510,958
50 PRINT AT 23,2,CHR$(25);"Koordináták";
60 INPUT X,Y
70 IF X=9999 THEN 120
80 IF ABS(X)>510 THEN 50
90 IF ABS(Y)>480 THEN 50
100 PLOT X+510,Y+480
110 IF X<>9999 THEN 40
120 PRINT AT 22,1
```

f./ Mikor hajtja végre a számítógép a THEN szó utáni utasítást?

g./ Mit jelent az, ha a THEN szó után csak egy szám van ?

3.3.3. Ciklus

A. Rajzoljunk a képernyőre egy számegyenest! Az egység hossza legyen tetszőleges.
Program neve: 333A

```
10 REM --- számegyenes ---
20 GRAPHICS 4:INPUT PROMPT"Egység=":E
30 GRAPHICS 2
40 PLOT 1000,496:PRINT#0,">"
50 PLOT 0,480;1006,480,
60 FOR X=0 TO 500 STEP E
70 PLOT 500+X,485;500+X,475
80 PLOT 500-X,485;500-X,475
90 NEXT X
100 PLOT 493,460:PRINT#0,"0"
110 PLOT 493+E,460:PRINT#0,"1"
```

A FOR és a NEXT utasítások közé írtuk azokat az utasításokat, amelyekkel a beosztásokat rajzoljuk meg. Ezeknek az utasításoknak a végrehajtása ismétlődik mindaddig, ameddig E lépésközönként 0-tól el nem jutunk 500-ig, azaz a képernyő széléig.
a./ Számokkal is írjuk ki a beosztások értékét!

B. Egészítsük ki az előző programot! Ábrázoljuk a számegyenesen a ciklusváltozó értékének változását!
Program neve: 333B

```
10 REM --- ciklus-utasitás ---
20 GRAPHICS 4:INPUT PROMPT"Egység=":E
30 GRAPHICS 2:PLOT 1000,496:PRINT#0,">"
40 PLOT 0,480;1006,480,
50 FOR X=0 TO 500 STEP E
60 PLOT 500+X,485;500+X,475
70 PLOT 500-X,485;500-X,475
90 NEXT X
100 PLOT 493,460:PRINT#0,"0"
110 PLOT 493+E,460:PRINT#0,"1"
199 !
200 REM  --- adatok bekérése ---
201 !
210 PRINT AT 2,4,"kezdőérték";:INPUT K
220 PRINT AT 4,20,"kezdőérték=";K
230 PLOT 500+K*E,480;500+K*E,510
240 PRINT AT 2,4,STRING$(20,32)
250 PRINT AT 2,4,"végérték";:INPUT V
260 PLOT 500+V*E,480;500+V*E,510
270 PRINT AT 6,20,"végérték=";V
280 PRINT AT 2,4,STRING$(20,32)
290 PRINT AT 2,4,"lépésköz";:INPUT L
291 PRINT AT 8,20,"lépésköz=";L
292 PRINT AT 2,4,STRING$(20,32)
```

```
294 !
295 REM    --- ciklus-utasitás ---
296 !
300 FOR X=K TO V STEP L
310 GET
320 PLOT 500+X*E,480;500+X*E,500
330 SET INK 0:SET PAPER 1:PRINT AT 10,20,
"ciklusváltozó=";
340 PRINT USING"####.#":X:SET INK 1:SET P
APER 0
350 NEXT
360 PRINT AT 22,1
```

b./ Mi a szerepe a STRING$ függvénynek és a PRINT USING utasításnak?

c./ Próbáljuk ki a következő eseteket:
K<V és L>0
K<V és L<0
K>V és L>0
K>V és L<0
K=V és L tetszőleges.

C. Rajzoljunk kört!
Programok neve: 333C1, 333C2, 333C3, 333C4, 333C5, 333C6, ELLIPSZIS, SPIRÁLOK

Mielőtt hozzákezdenénk egy kört rajzoló program megírásához, rajzoljunk egy derékszögű háromszöget!
Az átfogó egyik végpontjának koordinátái (U;V), a háromszög átfogójának hossza R, vízszintes befogója X. A 60-as sorban a két befogó, a 70-es sorban az átfogó mentén jutunk el az átfogó egyik végpontjából a másikba.

```
10 REM  --- derékszögü háromszög ---
20 GRAPHICS 2
30 U=500:V=500:R=300
40 X=200
50 Y=SQR(R*R-X*X)
60 PLOT U,V;U+X,V;U+X,V+Y
70 PLOT U,V;U+X,V+Y
```

Változtassuk X értékét -R-től R-ig. A sok háromszög egy fél ellipszist fest a képernyőre.

```
10 REM  --- derékszögü háromszögek ---
20 GRAPHICS 2
30 U=500:V=500:R=300
40 FOR X=-R TO R
50 Y=SQR(R*R-X*X)
60 PLOT U,V;U+X,V;U+X,V+Y
70 PLOT U,V;U+X,V+Y
80 NEXT
90 GET
```

Azért rajzol az előző program egy fél ellipszist, mert a képernyő két pontjának távolsága függőlegesen mérve kisebb, mint vízszintesen. Szoroznunk kell Y-t 1.4-1.5-el (szemmérték és a képernyő beállításától függően), ha félkört akarunk rajzolni.
A képernyőn látszik, hogy a félkör két szélén a körív helyett

szakaszokat rajzol a program. Itt X értékének kis változásával Y sokat változik. A körívnek a középső része 'szép'. Ebből az ívből fogjuk rajzolni a kört úgy, hogy azt 90 fokkal elforgatjuk.
A 90 fokos elforgatást a következő programmal mutatjuk be. A programot úgy sorszámoztuk, hogy az előzőt ne írja felül ez a program. A RUN 200 paranccsal indítható, és valamelyik billentyűt le kell nyomni, hogy a következő merőleges szakasz megjelenjen.

```
200 REM --- merölegesek ---
210 GRAPHICS 2
220 U=500:V=500
230 X=100:Y=200:K=1.5
240 PLOT U,V;U+X,V+K*Y:GET
250 PLOT U,V;U-Y,V+K*X:GET
260 PLOT U,V;U-X,V-K*Y:GET
270 PLOT U,V;U+Y,V-K*X
```

Írjuk át a félkört rajzoló programunkat!

```
 10 REM  --- kor-1 ---
 20 GRAPHICS 2
 30 U=500:V=500:R=200
 34 X1=-3*R/4:Y1=SQR(7*R*R/16):K=1.5
 40 FOR X=-3*R/4 TO 3*R/4 STEP 30
 50 Y=SQR(R*R-X*X)
 60 PLOT U+X1,V+K*Y1;U+X,V+K*Y
 70 PLOT U-Y1,V+K*X1;U-Y,V+K*X
 75 PLOT U-X1,V-K*Y1;U-X,V-K*Y
 80 PLOT U+Y1,V-K*X1;U+Y,V-K*X
 90 X1=X:Y1=Y
100 NEXT
```

A K és L változók értékétől függ az ellipszis alakja. Az X1 és Y1 változókban az előző pont koordinátáit őrízzük meg. A lépésköz változtatásával gyorsíthatjuk vagy lassíthatjuk a rajzolást, és ettől függően durvább vagy finomabb görbét kapunk.

Ezek után nézzünk egy másik megoldást is!
Most úgy rajzolunk derékszögű háromszöget, hogy az átfogó egyik végpontjának koordinátáin (U;V) kívül, adott az átfogó hossza, és a vízszintes befogó melletti szög; FI.
A szöget radiánban kell megadni. Ez azt jelenti, hogy PI radián = 180 fok, tehát PI/4 radián = 45 fok.
Az átfogó és a SIN(szinusz) és COS(koszinusz) szögfüggvény segítségével kiszámíthatjuk a két befogót.
Most csak az átfogókat fogjuk kirajzolni, azaz egy kör sugarait.

```
 10 REM --- átfogók ---
 20 GRAPHICS 2
 30 U=500:V=500:R=200
 40 A=1:B=1.4
 50 FOR FI=PI/2 TO 5*PI/2 STEP PI/15
 60 X=R*COS(FI):Y=R*SIN(FI)
 70 X=A*X:Y=B*Y
 80 PLOT U,V;U+X,V+Y;
 90 NEXT
```

Az előző programból könnyű egy másodperceket mérő órát készíteni. A SET INK utasítással A 'ceruza' színét a háttér színére állítjuk,

ekkor törli az előző vonalat, vagy láthatóvá tesszük a 'ceruza' színét, ekkor húzzuk a vonalat.

```
10 REM --- óra ---
20 GRAPHICS 2
30 U=500:V=500:R=200
40 A=1:B=1.4
50 FOR FI=PI/2 TO -3*PI/2 STEP -PI/30
60 X=R*COS(FI):Y=R*SIN(FI)
70 X=A*X:Y=B*Y
75 SET INK 0
80 PLOT U,V;U+X1,V+Y1
85 SET INK 1
90 PLOT U,V;U+X,V+Y
100 X1=X:Y1=Y
105 FOR I=1 TO 500:NEXT
110 NEXT
```

Ezek után nézzük a programunkat, amivel kört rajzolhatunk:

```
10 REM --- kör ---
20 GRAPHICS 2
30 U=500:V=500:R=200
40 A=1:B=1.4
50 FOR FI=PI/2 TO 5*PI/2 STEP PI/15
60 X=R*COS(FI):Y=R*SIN(FI)
70 X=A*X:Y=B*Y
80 PLOT U+X,V+Y;
110 NEXT
```

d./ Rajzolj céltáblát a képernyőre!

3.3.4. Függvény, szubrutin.

A. Rajzoljuk fel az $x^2 - 15$ függvényt a képernyőre!
Programok neve: 334A,FVABR

A koordináta-rendszert kirajzoló részt a számegyenes-rajzoló program kiegészítésével oldottuk meg.
A függvény rajzolásakor végigmegyünk vízszintesen a képernyő minden lehetséges értékén, és a pontnak megfelelő X értékhez tartozó függvényértéket meghatározzuk. Legvégül csak azt kell kiszámítanunk, hogy az Y értékének a képernyő melyik pontja felel meg.
Akkor,ha a pont nem kerül a képernyőre, a képernyő szélén a 'papír' színével rajzoljuk ki a pontot.

```
10 REM  --- függvény rajzolás ---
20 REM  --- koordinata-rendszer ---
30 GRAPHICS 4:INPUT PROMPT"Egyseg=":E
40 GRAPHICS 2:PLOT 1000,496:PRINT#0,">"
50 PLOT 0,480;1006,480,
60 PLOT 493,958:PRINT#0,"^"
70 PLOT 500,0;500,940
80 PLOT 500+E,470;500+E,490
90 PLOT 493+E,460:PRINT#0,"1"
```

```
100 REM --- függvény -=-
110 DEF FNY(X)=X^2-15
120 FOR I=0 TO 1023 STEP 5
130 X=(I-500)/E
140 Y=FNY(X)
150 J=Y*E+500
160 T=1:SET INK T
170 PRINT AT 2,50,"x=";
180 PRINT USING"####.#":X
190 IF J<0 THEN T=0:J=0
200 IF J>958 THEN T=0:J=958
210 SET INK T:PLOT ;I,J
220 NEXT
230 GET:CLS
```

A függvényeket a DEF utasítással, az ún. függvénydefiniáló utasítással adtuk meg. A DEF szó után a függvény neve, és zárójelben a függvény változója. Az egyenlőség jobb oldalán a függvényünk. Más függvény ábrázolása esetén csak az egyenlőség jobb oldalát kell módosítani.

a./ Változtasd a lépésköz értékét a FOR ciklusban! Mit tapasztalsz?
b./ Magyarázd meg a 120-140-es sorokat!
c./ Mi történik, ha a T értéke 0? Miért van erre szükség?
d./ Ábrázold a következő függvényeket!

$y=x^2$; $y=x^2-5$; $y=2x^2$; $y=-x^2$; $y=(x-2)^2$

$y=x$; $y=2x$; $y=-3x$; $y=\frac{1}{2}x$; $y=-\frac{2}{3}x$; $y=2x+3$

e./ Oldd meg grafikusan a következő egyenleteket!

$x^2=2x+3$ $\qquad$ $(x-5)^2=-2x+5$ $\qquad$ $x^2=-2x-5$

$x^2-5=x$ $\qquad$ $-2x+3=6-x^2$ $\qquad$ $x^2-5=-5$

8. Készítsünk oszlopdiagram-rajzoló programot!
Program neve: 334B

Vegyük elő a téglalap-rajzoló programunkat (3.3.1./B)! Készítsünk ebből egy szubrutint, amit a GOSUB utasítással hívhatunk a ciklus belsejéből.
A szubrutin egy olyan programrészlet, ami valamilyen részfeladatot old meg. Jelen esetben egy téglalapot rajzol.
A szubrutin a RETURN utasítás végrehajtásával fejeződik be, és utána a GOSUB utasítást követő utasítással folytatódik a program végrehajtása.

```
 10 REM --- oszlop-diagram ---
 20 CLS:INPUT PROMPT"Adatok száma=":N
 30 D=1000/N:B2=0
 40 GRAPHICS 2
 41 !
 50 FOR I=0 TO N-1
 60 PRINT AT 2,50,STRING$(10,32)
 70 PRINT AT 2,50,"adat";:INPUT Y
 80 A1=I*D:A2=Y:B1=A1+D
 90 GOSUB 200
100 NEXT I
101 !
110 PRINT AT 2,50,STRING$(10,32)
120 GET:CLS:END
199 !
200 REM  --- téglalap ---
210 PLOT A1,A2;B1,A2;B1,B2;
220 PLOT A1,B2;A1,A2
230 RETURN
```

A FOR ciklussal N db adatot kérünk be és megrajzoljuk az adatokhoz tartozó oszlopokat.

f./ Fesd be a téglalapokat!

g./ Add meg az adatokat a programban!

h./ Rajzolj halmazkarikákat és azt keretezd be!

i./ Rajzold ki az olimpiai jelvényt!

3.4. Karakterláncok

3.4.1. Műveletek, függvények

A. Vizsgáljuk meg, hogy hogyan tudunk előállítani egy karakterlánc karaktereiből egy másik karakterláncot! Gépeljük be és értelmezzük a programot!
Program neve: 341A

```
 5 REM --- függvények,láncolás ---
10 CLS:K$="karakter"
20 INPUT PROMPTK$&"lánc:":Q$
30 INPUT PROMPT"Első "&K$&":   ":A
40 INPUT PROMPT"Utolsó "&K$&":   ":B
50 PRINT A;"."&K$&": ";Q$(A)
60 PRINT B;"."&K$&": ";Q$(B)
70 PRINT A;"."&K$&"töl: ";Q$(A:)
80 PRINT B;"."&K$&"ig: ";Q$(:B)
90 PRINT A;"."&K$&"töl a";B;". "&K$&"ig:
";Q$(A:B)
```

a./ Az INPUT PROMPT és a PRINT utasításban a & jel helyettesíthető ;-vel?

B. Írjunk programot, amivel megvizsgálhatjuk a karakterlánc-függvényeket!

```
10 REM --- karakterlánc-függvények-1 ---
20 CLS:INPUTPROMPT"A karakterlánc:":Q$:P
RINT
30 PRINT "hossza=";LEN(Q$);" karakter":P
RINT
40 PRINT "első karakterének kódja: ";ORD
(Q$):PRINT
50 PRINT "numerikus értéke: ";VAL(Q$)

10 REM --- karakterlánc-függvények-2 ---
20 CLS:INPUTPROMPT"A karakter kódja:":K:
PRINT
30 INPUTPROMPT"A karakterek száma:":N:PR
INT
40 PRINT " a karakter: ";CHR$(K):PRINT
50 PRINT N;"db. karakter: ";STRING$(N,K)

10 REM --- karakterlánc-függvények-3 ---
20 CLS:INPUTPROMPT"Numerikus adat: ":Q
30 PRINT "karakterlánc: ";STR$(Q)
```

b./ Mi a jelentésük ezeknek a függvényeknek?
c./ Csoportosítsuk a karakterlánc-függvényeket!

3.4.2. Vizsgálat

A. Döntse el a számítógép, hogy számjegyet, kisbetűt, vagy nagybetűt gépeltünk-e be!
Program neve: 342A
A beolvasott karaktert kódja alapján vizsgáljuk.

```
  10 REM  --- karakterek vizsgálata ---
  20 CLS:PRINT AT 3,2,"Nyomj le egy billen
tyüt!":PRINT
  30 GET A$:A=ORD(A$)
  40 IF A<48 THEN STOP
  50 IF A<=57 THEN PRINT "számjegy":STOP
  70 IF A<65 THEN STOP
  80 IF A<=90 THEN PRINT "nagybetü":STOP
 100 IF A<97 THEN STOP
 110 IF A<=122 THEN PRINT "kisbetü":STOP
 120 IF A<128 THEN STOP
 130 IF A<=136 THEN PRINT"nagybetü":STOP
 140 IF A<144 THEN STOP
 150 IF A<=152 THEN PRINT "kisbetü"
```

a./ Mi a jelentése az IF utasításnak?
b./ Logikai műveletek alkalmazásával egyszerüsítsd a vizsgálatokat!

3.4.3. Ciklus

A. Írassuk ki egy karakterlánc karaktereit egymás alá!

A karakterlánc betüit egyenként kiírjuk. A LEN függvény megadja, hogy a karakterlánc hány karakterből áll. A FOR/NEXT utasításpárral I értékét 1-től kezdődően egyesével növeljük mindaddig, ameddig I értéke nagyobb nem lesz,mint a karakterlánc hossza.
A karakterláncok maximális hosszát a DIM utasításban adjuk meg.

```
  10 REM  --- betük ---
  20 CLS:DIM W$*32
  30 INPUT PROMPT"Szó: ":W$
  40 FOR I=1 TO LEN(W$)
  50 PRINT W$(I)
  60 NEXT I
```

a./ Írassuk ki visszafelé a szó karaktereit!
b./ Írassuk ki a szó magánhangzóit!
c./ Számoljuk meg a mássalhangzókat!

B. Vizsgáljuk meg a karakterek kódját!
Program neve: 3438
A botkormány előre-hátra mozgatásával lehet a kódokat növelni, ill. csökkenteni(50-es és 60-as sorok).
A 90-es sorban azt vizsgáljuk, hogy lenyomtuk-e a RETURN billentyüt. Ezzel jelezhetjük a vizsgálat végét.

```
  10 REM --- karakterkódok ---
  20 CLS: I=32
  30 PRINT AT 12,12,CHR$(I);I
  40 GET A$
  50 IF ORD(A$)=5 THEN I=I+1
  60 IF ORD(A$)=24 THEN I=I-1
  70 IF I=160 THEN I=32
  80 IF I=31 THEN I=159
  90 IF ORD(A$)<>13 THEN 30
```

d./ Mi történik a 70-es és a 80-as sorokban?

C. Oldjuk meg, hogy egy adat begépelésekor csak számjegyeket olvasson be a programunk!
Programok neve: 343C,TIZEDES
A számjegyeket egyenként olvassuk be, megvizsgáljuk és összefűzzük egy karakterlánccá. A SZÁM$ változó kezdőértéke az 'üres karakter' lesz.

```
10 REM --- számok beolvasása ---
20 CLS:SZÁM$=""
50 PRINT AT 3,5,"Kérem a számot:"
60 PRINT AT 3,20+I,X$;
70 GET X$
80 IF ORD(X$)=13 THEN 110
90 IF X$<"0" OR X$>"9" THEN 70
100 SZÁM$=SZÁM$&X$:I=I+1:GOTO 60
110 SZÁM=VAL(SZÁM$)
120 PRINT AT 6,10,"Négyzete=";SZÁM^2
```

e./ Hogyan lehetne a tévesen beírt számjegyet törölni?

D. Készítsünk programot, amivel egy tizenhatos számrendszerben adott számot (hexadecimális szám) tízes számrenszerbeli számmá alakítjuk (decimális szám).
Program neve: 343D

A hexadecimális számok 9-nél nagyobb számjegyeit A,B,C,D,E és F betűkkel jelöljük.

```
10 REM --- h-d átalakitás ---
20 CLS:INPUT PROMPT"Hexadecimális szám:"
:X$
30 D=0
40 FOR I=1 TO LEN(X$)
50 X=ORD(X$(I)):V=0
60 IF X>=65 AND X<=70 THEN X=X-55:V=1
70 IF X>=48 AND X<=57 THEN X=X-48:V=1
80 IF V=0 THEN V=2:I=LEN(X$)
90 D=16*D+X
100 NEXT
110 IF V=2 THEN PRINT "rossz számjegy!":E
ND
120 PRINT "Decimális számrendszerben:";D
```

E. Írjunk fel egy decimális számot hexadecimális alakban!
Programok neve: 343E,ATIRAS

Az átszámítás algoritmusát a 2. fejezetben megtaláljuk. Az osztások maradékai adják a számjegyeket. A 9-nél nagyobb számjegyeket kell átalakítani A,B,C,D,E vagy F karakterré, ezután a számjegyeket összefűzzük egy karakterlánccá.

```
10 REM --- d-h átalakitás ---
20 CLS:INPUT PROMPT"Decimális szám:":D
30 H$=""
40 Q=INT(D/16):M=D-16*Q
50 IF M>=10 THEN M=M+55:ELSE M=M+48
60 H$=CHR$(M)&H$
70 D=Q:IF D<>0 THEN 40
80 PRINT "Hexadecimális alak:";H$
```

F. Hogyan lehet kis helyre hosszabb szöveget írni?
Program neve: 343F

A szöveg, amit kiíratunk a programmal, maximálisan 32 karakterből állhat. Ezt a DIM utasításban megadtuk. Az ablak hossza legyen kisebb a karakterek számánál.
Az első L karaktert kiíratjuk (60-as sor), és az első karaktert a karakterlánc végére tesszük (70-es sor). Ezek után kiíratjuk a karakterlánc első L karakterét, é.i.t.

```
10 REM  --- ablak ---
20 CLS:DIM W$*32
30 INPUT PROMPT"Szöveg:":W$
40 INPUT PROMPT"Ablak hossza:":L
50 CLS:W$=W$&" "
60 PRINT AT 12,16-L/2,W$(:L)
70 W$=W$(2:)&W$(1)
80 FOR I=1 TO 200:NEXT
90 Q$=INKEY$:IF Q$="" THEN 60
```

3.4.4. Indexes változók

A. Számoljuk meg, hogy egy karakterláncban hányszor ismétlődnek a magyar ábécé betűi!
Program neve: 344A

```
20 CLS:DIM M$*100,T(56)
30 INPUT PROMPT"szöveg:":M$
40 FOR I=1 TO LEN(M$)
50 L=ORD(M$(I))-97
60 T(L)=T(L)+1
70 NEXT
80 REM --- kiiratás ---
90 CLS:L=0
100 FOR I=0 TO 34
110 IF I=26 THEN L=L+21
120 L=L+1:J=INT(I/5):K=I-5*J
130 PRINT AT J+1,6*K+1,CHR$(L+96);"/";T(L
-1);
140 NEXT
```

B. Gépeljük be a következő játékprogramot!
Programok neve: 344B,KEVER

```
 10 REM  --- betük keverése ---
 20 CLS:DIM W$*32,A$(32)
 30 INPUT PROMPT"Szó(max.32 betü)":W$
 39 !
 40 REM  --- betük ---
 50 FOR I=1 TO LEN(W$)
 60 A$(I)=W$(I)
 70 NEXT I
 79 !
 80 REM  --- keverés ---
 90 RANDOMIZE
100 FOR I=1 TO 50
110 R=RND(LEN(W$))+1:S=RND(LEN(W$))+1
120 T$=A$(R):A$(R)=A$(S):A$(S)=T$
130 NEXT
139 !
140 REM --- összefüzés ---
150 S$=""
160 FOR I=1 TO LEN(W$)
170 S$=S$&A$(I)
180 NEXT
189 !
190 REM  --- kiirás ---
200 PRINT S$
```

C. Rendezzük ábécé sorrendbe az angol ábécé betűivel megadott szavakat!
Program neve: 344C,MAGYAR

Két karakterlánc összehasonlítása úgy történik, hogy az első karakterrel kezdődően egymás után összehasonlítja a számítógép a karakterláncok karaktereit. Ha az összehasonlítás során a két karakter kódszáma nem azonos, akkor azt a karakterláncot tekinti 'kisebbnek' amelyikben a kisebb kódszámú karaktert találta. Akkor, ha az összehasonlítás során valamelyik karakterlánc végére ért, akkor a rövidebb lesz a kisebb. Ezt elkerülhetjük azzal, hogy a szavakat kiegészítjük szóköz karakterekkel (70-es sor).

```
 10 REM --- sorbarendezés ---
 20 CLS:INPUT PROMPT"A szavak száma:":N
 30 DIM A$(N)*20,T$*20
 40 REM --- beolvasás ---
 50 FOR I=1 TO N
 60 INPUT PROMPT STR$(I)&".:":A$(I)
 70 A$(I)=A$(I)&STRING$(20-LEN(A$(I)),32)
 80 NEXT
 90 REM --- rendezés ---
100 FOR I= 1 TO N-1
110 : FOR J=I+1 TO N
120 :   IF A$(I)<A$(J) THEN 140
130 :   T$=A$(I):A$(I)=A$(J):A$(J)=T$
140 : NEXT
150 NEXT
160 REM --- kiirás ---
170 CLS:FOR I=1 TO N
180 PRINT A$(I)
190 NEXT
```

Tizenhatszínű képernyő esetén a SET INK utasítással a rajz, a betű színének sorszámát adja meg. Lásd a Mellékletet!

a./ Töröld az első programból 110-es és a 120-as sorokat! Hogyan oldanád meg a feladatot a megmaradt adatokkal?

Négyszínű képernyőn a SET PALETTE utasítással adjuk meg a színeket. Itt soroljuk fel azoknak a színeknek a kódját, amelyeket használni szeretnénk. A SET INK és a SET PAPER utasításokban a kódszámok sorszámára hivatkozunk.
Új színkódok megadásával a színek az egész képernyőn megváltoznak.

b./ Próbáld ki a második programot a GRAPHICS 2 utasítással is! Mit tapasztalsz?

c./ Milyen utasítások vonatkoznak a színekre?

B. Rajzoljunk nemzetiszínű zászlót!
Program neve: 35B

```
  10 REM --- zászló ---
  20 GRAPHICS 4
  30 SET PALETTE 68,85,80,21
  35 REM piros,fehér,zöld,szürke
  40 SET PAPER 3:SET INK 0:CLS
  50 PLOT 300,500;700,500;700,800;300,800;
300,500
  60 PLOT 300,600;700,600
  70 PLOT 300,700;700,700
  80 PLOT 500,750,PAINT
  90 SET INK 1:PLOT 500,650,PAINT
 100 SET INK 2:PLOT 500,550,PAINT
 110 GET
 120 SET PALETTE 80,85,68,21
```

C. Írassunk ki villogó szöveget a képernyőre!
Program neve: 35C

```
  10 REM --- villogó kiírás ---
  15 GRAPHICS 4
  20 INPUT PROMPT"A szöveg:":Q$
  30 SET PALETTE 68,85,80,21:CLS
  40 I=12:GOSUB 200
  50 GRAPHICS 4:END
 199 !
 200 REM - villogtató szubrutin -
 210 SET PAPER 1-V:SET INK V
 220 PRINT AT I,17-LEN(Q$)/2,Q$
 230 FOR T=1 TO 300:NEXT
 240 V=1-V:W$=INKEY$:IF W$="" THEN 210
 250 RETURN
```

3.6. Hangok

A. Szólaltassuk meg a zenei hangokat!

Program neve: 36A

A hangmagasságot a PITCH utasításban adjuk meg egy paraméterrel. A paraméter ismeretében kiszámíthatjuk a hang frekvenciáját (60-as sor).
A következő hang frekvenciáját úgy kapjuk meg, hogy 2^(1/12)-nel (tizenkettedikgyök kettő) szorozzuk az előző frekvencia érteket. Ebből a paramétert a 120-as sorban számítjuk ki.

```
  10 REM      --- skála ---
  20 CLS:S=5:DIM H$(12)*2
  30 FOR I=0 TO 11:READ H$(I):NEXT
  40 PRINT AT 3,3,"PITCH paraméter  frekve
ncia"
  50 I=0:P=110:REM PITCH paraméter
  60 F=195312.5/(4096-P):REM frekvencia
  70 IF S>22 THEN PRINT AT 5,1,STRING$(2,2
5):S=21
  80 P1=INT(P+0.5):F1=INT(10*F+0.5)/10:REM
 kerekítés
  90 IN=I-12*INT(I/12):REM maradék
 100 PRINT AT S,6,P1;TAB(16);H$(IN);TAB(22
);F1:S=S+2
 110 SOUND PITCH P,VOLUME 1
 120 F=F*2^(1/12):REM következö hang
 130 P=4096-195312.5/F
 140 IF P<4050 THEN GET:I=I+1:GOTO 70
 200 REM  --- hangok ---
 210 DATA G,#G,A,#A,H,C,#C,D,#D,E,F,#F
```

Hang	P	F	P	F	P	F
C	2603	130.8	3350	261.6	3723	523.3
#C	2687	138.6	3391	277.2	3744	554.4
D	2766	146.8	3431	293.7	3763	587.4
#D	2841	155.6	3468	311.1	3782	622.3
E	2911	164.8	3504	329.6	3800	659.3
F	2978	174.6	3537	349.2	3816	698.5
#F	3040	185.0	3568	370.0	3832	740.0
G	3100	196.0	3598	392.0	3847	784.0
#G	3155	207.7	3626	415.3	3861	830.6
A	3208	220.0	3652	440.0	3874	880.0
#A	3258	233.1	3677	466.2	3887	932.4
H	3305	247.0	3701	493.9	3898	987.8

8. 'Zongorázzunk' a billentyűzeten!

Program neve: 368

A megfelelő billentyűkhöz hozzá kell rendelni a megfelelő hangot. A billentyűket úgy választottuk ki, hogy azok elhelyezkedése hasonlítson a zongora billentyűinek elrendezéséhez.

```
  10 REM       --- zongora ---
  20 CLS:DIM B(255),K$(255)*2,Z(200),T(200
),H$(200)*2
  30 FOR I=0 TO 12
  40 PRINT AT 3,6,"Hangok és billentyük":P
RINT
  50 FOR I=0 TO 12
  60 READ W$,P,B$:PRINT " ";W$;"-";B$;
  70 B(ORD(B$))=P:K$(ORD(B$))=W$
  80 NEXT
  84 !
  85 REM  --- lejátszás ---
  90 PRINT AT 9,3,"Lehet játszani!":PRINT"
A RETURN billentyü lenyomásával jelezzük
a végét.":PRINT
 100 P=4095:Z(0)=P:I=0
 110 Q$=INKEY$:IF Q$<>"" THEN 140
 120 SOUND VOLUME 1,DURATION 2,PITCH P
 130 TI=TI+1:IF TI<256 THEN 110 :ELSE END
 140 IF B(ORD(Q$))=0 AND ORD(Q$)<>13 THEN
110
 150 T(I)=TI:TI=0:P=B(ORD(Q$)):Z(I+1)=P:I=
I+1
 160 PRINT K$(ORD(Q$));"/";:P=B(ORD(Q$))
 170 IF ORD(Q$)<>13 THEN 110
 179 !
 180 REM  --- visszajátszás ---
 190 FOR J=0 TO I+1
 200 SOUND ;VOLUME 1,DURATION T(J),PITCH Z
(J);DURATION 5,PITCH 4095
 220 NEXT:PRINT
 299 !
 300 REM  --- hangok ---
 310 DATA C,3349,e,C#,3391,0,D,3431,;,D#,3
468,1,E,3503,q,F,3537,w,F#,3568,3
 320 DATA g,3598,e,g#,3626,4,a,3652,r,a#,3
677,5,h,3701,t,c,3723,z
```

C. Befejezésül a miskolci avasi templom harangjátékát szólaltatjuk meg.

Programok neve: 36C,AVAS

A hangokat olyan formában kell megadnunk, hogy ne okozzon gondot a kotta alapján a hangokat meghatározó kódok beírása a DATA-ba, egy karakterláncba, vagy indexes változókba.
Mi most a DATA mellett döntöttünk. Ide írjuk a hangokat meghatározó kódokat.

időtartam: T1 - egész hang,
T2 - fél hang,
T4 - negyed hang, stb.

hangerő: V7 - 7-es hangerő.

hangmagasság: C0,C#0,D0, stb. -alsó skála
C1,C#1,D1, stb. -középső skála
C2,C#2,D2, stb. -felső skála

A kódolást rábízzuk a számítógépre.

```
 10 REM --- avasi harangjáték ---
 20 CLS:DIM P(13),H$(13)*2
 30 REM  --- hangok ---
 40 F=246.94:FOR I=1 TO 13
 50 F=F*2^(1/12)
 60 P(I)=4096-195312.5/F
 70 READ H$(I):NEXT
 80 REM  --- zene ---
 90 FOR I=1 TO 16
100 READ W$
110 J=1
120 IF H$(J)<>W$ THEN J=J+1:GOTO 120
130 T=25
140 IF I=8 OR I=16 THEN T=50
150 SOUND ;VOLUME 1,DURATION T,PITCH P(J)
160 NEXT
200 REM  --- skála ---
210 DATA C,#C,D,#D,E,F,#F,G,#G,A,#A,H,C'
220 REM  --- dallam ---
230 DATA C',A,F,D,H,G,E,C
240 DATA D,E,F,A,G,H,C',C
```

4. Feladatok megoldása

1. Számítógép kezelése

1.1. A billentyűzet

a./ A program leállítása a CTRL és a ESC billentyűk egyidejű lenyomásával, vagy A RESET nyomógomb (jobb oldalt alul) egyszeri benyomásával történhet.
A RESET nyomógomb kétszeri benyomásával, a számítógép alapállapotba kerül. Ekkor a program törlődik.

b./ - Ha a LIST parancsra nem ír ki minden begépelt sort, a RETURN billentyűt nem nyomtad le.

- Ha két sort ír ki a számítógép hibajelzéskor, akkor két sort egybe gépeltél. A számítógép automatikusan sort emel, amikor a sor végére ér. Ilyenkor elfelejtkezünk a RETURN lenyomásáról.

- Nem írtál, vagy felesleges írtál 'szóköz' karaktert valahová.

- Ø (nulla) helyett O betűt írtál.

c./ Az előzőleg beírt, azonos sorszámú sor helyére ez az utasítássor kerül.
Akkor, ha csak a sorszámot írjuk be, ez a programsor törlődik.

1.2. Programok kezelése

a./ Lásd 1.1.a feladat megoldását

b./ Utólag be kell gépelni a programsort olyan sorszámmal, amilyen nem szerepel a programban. Ezért célszerű 'szellősen'sorszámozni a programsorokat.

c./ ún. 'nyomkövetés' történik. A számítógép kiírja azoknak az utasításoroknak a sorszámát, amelyeket végrehajt.
A 'nyomkövetést' a TRACE OFF paranccsal lehet megszüntetni.

d./ A program leáll.

e./ A számítógép a kazettáról beolvassa az ISKOLA nevű programot, ha megtalálta a szalagon.

f./ A szalagot visszatekercseljük a program elejére, kiadjuk a VERIFY parancsot és elindítjuk a lejátszásra állított magnetofont. Ez a parancs a számítógépben lévő programot összehasonlítja a szalagra kiírt programmal.
g./ A program nevében nem szerepelhet hosszú magánhangzó.

2.1. Kiírás képernyőre

2.1.1. Számok és karakterláncok kiírása

a./ Az első paranccsal 'tizedespontos' számokat írunk ki. A vessző helyett pontot használunk a tizedesszámok írásakor.A számok tehát:
-34,56;455,789;-0.012
A második parancsban 'lebegőpontos' alakban adtuk meg a számokat.

$23{,}4 \cdot 10^{-3}$; 10^{10}; $-0{,}0012 \cdot 10^{2}$; $6{,}67 \cdot 10^{23}$

A képernyőn csak azok a számok jelennek meg lebegőpontos alakban, amelyek leírásához tíz számjegynél több szükséges.
b./

```
PRINT -5,-4,-3,-2,-1

PRINT -5;-4;-3;-2;-1
```

Csak egy szóköz karakter választja el a számokat egymástól. Pozitív számok esetén, miután nem írja ki az előjelet, két szóköz lesz közöttük.
c./

```
PRINT "a á e é i í o ó ö ő ü ű"

PRINT "A Á E É I Í O Ó Ö Ő Ü Ű"
```

d./ 'Írd ki a PRINT szó után felsorolt számokat! Pontosvessző esetén egy szóközt hagyjál ki és úgy írd ki a következő számot. Ha vesszőt találsz, akkor a 9.,17., stb. karakterpozíción folytasd a kiírást, attól függően, hogy melyik következik!'

2.1.2. Változók

a./ Kisbetűkkel is elfogadja a parancsokat a számítógép.

b./ A karakterláncban, az idézőjelek közé írt karakterek esetében.

c./ A karakterlánc-változók nevének végén $(dollár) jel van. A karakterláncokat stringeknek (ejtsd:sztring) is nevezik.

d./ - Az első karakter csak az angol ábécé valamelyik betűje lehet. Máshol lehet számjegy és hosszú magánhangzó is.
- A változó neve nem kezdődhet olyan szóval, amelyet a BASIC nyelvben is használunk. Ezeket a szavakat kulcsszavaknak nevezzük. Például: PRINT, AT, PLOT,OR,END,stb.
(Van amikor az értékadásnál nem jelez hibát a gép, csak akkor ha ki akarjuk íratni a változó értékét.)
Program: TOKEN
e./

```
OSZLOP=2

PRINT AT 10,OSZLOP,"BASIC"
```

Az OSZLOP változó értékét kell növelni.
A BASIC szó elé írj egy szóköz karaktert. Nézd meg mi történik!

f./ GRAPHICS 2 : 1<= OSZLOP <=64
GRAPHICS 4 : 1<= OSZLOP <=32
GRAPHICS 16 : 1<= OSZLOP <=16

A sorok száma mindig 24.
g./ Mindkettő törli a képernyőt. A CLS nem állítja át a betűk méretét.

h./ 'Az egyenlőség baloldalán álló változó vegye fel a jobb oldalon megadott értéket!'

i./

```
PRINT AT 12,10,"***************":
PRINT AT 13,10,"*MAGYARORSZÁG*":
PRINT AT 14,10,"***************"
```

Készítsük el a keretet grafikus karakterekből is. Miközben lenyomod a számjegy billentyűket, tartsd lenyomva a CTRL billentyűt.

2.2. Számolás

2.2.1. Számok

a./ szorzás: * (csillag)
osztás: / (per)
összeadás: + (plusz)
kivonás: - (minusz.

b./ Lásd a 2.1.1/a feladatot

c./ Lásd A 2.1.1/d feladatot

2.2.2. Változók

A./ Nem. Ha csak az egyik oldalt változtatjuk, biztos, hogy a háromszög nem lesz derékszögű. A terület értéke hibás lesz.

b./ Lásd 2.1.2/b feladat megoldását!

c./

```
A=125:B=137.5

PRINT 2*(A+B),A*B

PRINT 2*A+2*B,A*B
```

2.2.3. Függvények

a./

$$x \geq 0\ ;\ x \geq 2\ ;\ x \leq 2\ ;\ -2 \leq x\ ;\ -2 \leq x \leq 2$$

b./

```
PRINT INT(12.87+0.5),INT(6.32+0.5)

PRINT INT(10*12.87+0.5)/10

PRINT INT(10*6.32+0.5)/10
```

c./

```
X1=34.67:X2=-4.56

PRINT ABS(X1-X2)
```

2.3. Grafika

2.3.1. Pontok, szakaszok

a./ A pontokat, szakaszokat ugyanott látjuk a képernyőn mint a GRAPHICS 4 parancs után. A pontok kisebbek ill. nagyobbak, a vonalak vékonyabb, ill. vastagabbak lesznek. GRAPHICS 2 parancs után az ábrák fínomabbak, GRAPHICS 16 után durvábbak lesznek.

b./ A képernyő négy sarkában elhelyezkedő pontokat kell összekötni. Ezek koordinátái:
bal felső: 0,959 jobb felső: 1023,959

BAl alsó: 0,0 jobb alsó: 1023,0
c./ (0,0) kezdőponttal, (400,600) véponttal, egy szakaszt rajzol.

Bekapcsoláskor a 'ceruza' azaz a grafikus kurzor a (0,0) pontban van.

d./ Csak a háromszög belsejét nem festi be.

e./ 14 féle vonal lehet. A SET STYLE utasítás paraméterét kell megváltoztatni. Legyen az értéke 1,2,...,14!
f./ A CLS az eredeti háttérszínnel, vagy azzal a színnel törli a képernyőt amilyenre a SET PALETTE utasítással beállítottuk a hátteret.
A GRAPHICS paranccsal a megfelelő betűméretet, a grafika finomságát és a színek számát is megadjuk, miközben a képernyő törlődik.
A színekről a 2.5. fejezetben lesz szó.
g./ A kiírás oda kerül, ahová a PLOT paranccsal a ceruzát helyeztük.

2.4. Karakterláncok

2.4.1. Karakterlánc-műveletek

a./ Alaphelyzetben csak 10 karakterből álló karakterláncokat használhatunk. Adjuk ki a DIM B$ * 30 parancsot!
Ekkor a változó hossza 30 karakter lehet.

2.4.2. Karakterlánc-függvények

a./ Törli a képernyőt! A soronkénti törlés így azt a látszatot kelti, mintha a képernyőtartalom felfelé mozogna és 'lemenne' a képernyőről.
Program: MOZGAT

b./ Az első nem számjegy karakterig értelmezi a karakterláncot.
Kivételek:
- lebegőpontos alakban adott szám (második parancs)
- előjel (negyedik parancs)

c./ Az eredmény -3, tehát a karakterlánc hossza 2. Az előjelet is figyelembe veszi.

d./ A két parancs ugyanazt az értéket adja! Írassuk ki a két számot!

2.4.3. Karakterek definiálása

a./ ALT és a 0, ill. az ALT és az 'a' billentyűket kell lenyomni.

2.5. Színek

a./ A 64-el nagyobb kódszámú színek világosabbak.

b./ A SET INK paranccsal a 'ceruza' azaz a pontok és vonalak, valamint a betűk színét, a SET PAPER a 'papír', vagyis a háttér színét állítja be.
A SET PALETTE parancsban adott színek közül az első kettő lesz érvényben.

c./ A CLS paraccsal.

d./
```
GRAPHICS 16
SET PAPER 13:SET INK 0:CLS
PLOT 380,400;700,400;500,600;380,400
SET INK 2:PLOT 400,450,PAINT

GRAPHICS 4
SET PALETTE 81,0,68,7
SET PAPER 0:SET INK 1:cls
PLOT 380,400;700,400;500,600;380,400
SET INK 2:PLOT 400,450,PAINT
```

3. Programok

3.1. Kiírás a képernyőre

3.1.1. Változók

a./ Vagy újból beírod a sort, vagy csak a képernyőre kiírt program 10-es sorát változtatod.
A RETURN billentyű lenyomásáról ne feledkezz meg!

b./ Az OSZLOP változó értékét kell módosítani. A sorok száma mindig 24.

c./ A 30-as sorba, az idézőjelek közé kell írni a nevet, és a 40-es sorból az 1987-t törölni kell.
Írasd ki a programot és a kurzor mozgatásával,a DEL és INS billentyűk segítségével, írd át ezeket a sorokat. A módosítások után nyomd le a RETURN billentyűt. A kurzor lehet a sor közepén is.

d./ 24 sor és soronként
GRAPHICS 2-nél /két szín - kis karakteres/ 64 karakter
GRAPHICS 4-nél /négy szín - normál karakteres/ 32 karakter
GRAPHICS 16-nál /tizenhat szín - nagy karakter/ 16 karakter

e./ Lásd 2.1.2.d feladat megoldását.

f./ Azért jó, ha 'szellősen' programozunk, mert így utólag is beírhatunk újabb utasítássorokat a meglévő utasítássorok közé.

g./ K - 2,4 vagy 16
SOR - 0<SOR<=24
OSZLOP - nem vehet fel nagyobb értéket, mint amennyi karakter egy sorban elfér és nem lehet negatív. (0-t vizsgáld meg!)

Lásd a 3.1.1/d feladatot!

h./ Írd be a programba! Lásd 2.4. fejezetet!

```
80 PRINT AT SOR+1,OSZLOP,STRING$(LEN(Q$)
,45)
```

i./ - utasítások elválasztásakor
- INPUT PROMPT utasításban a kiírandó szöveg után
- PRINT USING utasításban a formátum után
- ELSE utasításszó előtt

j./ Az INPUT utasítás ?(kérdőjel) kiírással jelzi, hogy adatra vár. Az INPUT PROMPT utasítással szöveg is kiíratható.

3.1.2. Vizsgálat

a./ Több szín esetén nagyobbak a betűk. Lásd a 3.1.1/D feladatot!

b./ 'Ugorj a megadott sorszámú sorra, és folytasd ott a program végrehajtását!'

c./ A GOTO utasítások elhagyhatók és át kell írni a 22-es sort.

```
22 IF K=0 THEN 10
```

Oldd meg a 3.2.3/d feladatot!

d./ Lásd a 3.2.3/c feladat megoldását!

e./ Nézd meg a 2.3. fejezetben!

f./ Megáll a program futása és vár egy billentyű lenyomására. Az A$ változó értéke az a karakter lesz, amelyik a billentyűt lenyomtuk.

g./ Értékétől függően, rajzol, vagy töröl a program.

h./ Előállítja azt a karaktert, amelyiknek a kódszámát a zárójelben megadjuk.
Lásd a 2.4. fejezetet.

i./ Semmi. Egyik reláció sem teljesedik, a program újabb karaktert vár.
j./ Új karaktert alakíthatunk ki (definiálunk.)
A SET CHARACTER utasítás után az első szám a karakter kódja, aminek 127-nél nagyobbnak kell lennie.
A többi számmal a karakter pontjait adjuk meg. Írd át a számokat kettes számrendszerbe, és írd egymás alá. Látni fogod, hogy milyen lesz a karakter.

k./ 1.- GET - addig vár, ameddig le nem nyomunk egy billentyűt.
INKEY$ függvény értéke az a karakter lesz, amelyik billentyű le van nyomva a függvény végrehajtásakor.
Akkor, ha nincs lenyomva egy billentyű sem, az INKEY$ függvény értéke az 'üres karakter', és a program fut tovább.

2.- Olyan karakterlánc, amelyik nem tartalmaz karaktert (üres halmaz).
Jelölése: két idézőjel egymás mellett
3.- Lásd 3.1.2/h feladatot!

1./ Egészítsük ki a programot a következő vizsgálatokkal!

```
 82 IF I<1 THEN 120
 84 IF J<1 THEN 120
 86 IF I>24 THEN 120
 88 IF J>64 THEN 120
120 CLS:PRINT "Pontok száma=";PONT
```

3.1.3 Ciklus

a./ A GET utasítás egy billentyű lenyomását várja, így nem jelenik meg az a program befejezését jelző 'ok' kiírás.
A cikluson kívüli PRINT utasítás után pontosvesszőt teszünk, hogy ne legyen soremelés.

```
10 CLS
20 I=1
30 PRINT I;"Borsod"
40 I=I+1
50 IF I<24 THEN 30
60 PRINT I;"Borsod";
70 GET
```

b./

```
10 CLS
20 I=24
30 PRINT AT I,10,25-I;"Borsod";
40 I=I-1
50 IF I>0 THEN 30
70 GET
```

c./

```
10 CLS
20 I=23
30 PRINT AT I,10,"Borsod";
35 PRINT AT I+1,10,"       ";
40 I=I-1
50 IF I>0 THEN 30
70 GET
```

d./Annyival kell növelni a ciklusváltozó értékét, amennyi karakterből áll a neved. A LEN függvény a karakterlánc hosszát adja meg.

```
10 GRAPHICS 4
15 INPUT PROMPT"Nev=":NÉV$
20 CLS:I=1
30 PRINT AT 12,I," ";NÉV$
40 I=I+LEN(NÉV$)+1
50 IF I<32 THEN 30
```

e./ A lépésköz 2, vagy -2.

f./ A REM után bármit írhatunk, a számítógép nem veszi figyelembe. Ugyanez történik a felkiáltójel esetében is. Az utóbbi csak ezen a gépen igaz.

Így írhatunk a programokba magyarázatokat, és áttekinthetővé tehetjük a programot.

g./ Egészítsük ki a programot!

```
    5 GRAPHICS 16

   25 SET CHARACTER 130,0,0,0,0,0,0,126,67,12
  6,40,16,0

   40 PRINT AT 15,I," ";CHR$(130);CHR$(128)
  ;CHR$(129);
```

Írd be ezt a programot is! Mit Látsz?
Program: F313G

```
    5 GRAPHICS 4
   25 SET CHARACTER 130,0,0,0,0,0,0,126,67,12
  6,40,16,0
   26 SET CHARACTER 131,28,28,28,8,12,62,62
  ,28,54,99
   27 SET CHARACTER 132,28,28,28,8,62,93,93
  ,28,28,28,28
   30 FOR I=1 TO 31
   40 PRINT AT 15,I," ";CHR$(131);
   45 FOR T=1 TO 100:NEXT
   47 PRINT AT 15,I," ";CHR$(132);
   49 FOR T=1 TO 100:NEXT
   50 NEXT
```

h./ A PLOT utasítással azt a pontot jelöljük ki, ahová a karaktert ki akarjuk írni. Ilyenkor a PRINT#0 utasítást kell használnunk. Így jelezzük, hogy a képernyőre akarunk írni.

```
    5 GRAPHICS 16
   10 SET CHARACTER 128,0,0,31,16,112,64,12
  7,20,8,0
   20 SET CHARACTER 129,0,0,224,16,14,2,254
  ,40,16,0
   30 FOR X=0 TO 1000 STEP 8
   40 PLOT X,400,:PRINT#0," ";CHR$(128);CHR
  $(129)
   50 NEXT
```

i./ Az adatokat valahol meg kell őrizni, ha azt akarjuk, hogy a képernyőn csak a diagram legyen látható.
Indexes változókat használunk. A változó neve után zárójelben adjuk meg az indexet. Az indexként használt változó értékével jelölhetjük ki, hogy hányadik elemről van szó.
Először beolvassuk az adatokat, majd kirajzoljuk a diagramot.
Program: F313I

```
   5 GRAPHICS 2
  10 SET CHARACTER 128,129,129,129,129,129
,129,129,129,129,129
  20 SET CHARACTER 129,255,129,129,129,129
,129,129,129,129,129
  24 !
  25 REM --- adatok beolvasása ---
  26 CLS:INPUT PROMPT"Adatok száma(max.64)
:":N
  27 DIM A(N)
  29 !
  30 FOR I=1 TO N
  35 INPUT PROMPT STR$(I)&".adat:":A(I)
  36 IF A(I)>24 THEN 35
  40 NEXT
  42 !
  43 REM --- diagram ---
  44 CLS
  45 FOR I=1 TO N
  50 FOR J=23 TO 24-A(I)+1 STEP -1
  60 PRINT AT J,I,CHR$(128)
  70 NEXT
  72 PRINT AT J,I,CHR$(129)
  75 NEXT
  80 PRINT AT 1,1
  90 GET
```

- Tíz karaktert kell definiálnunk, ha tizedenként akarunk ábrázolni. Ez megoldható tíz SET CHARACTER utasítással, vagy egy ciklussal.
Program: DIAGRAM

3.2. Számolás

3.2.1. Értékadás

a./ Írd át a 10-es sort, ahol az A és B változóknak értéket adtunk! Vagy az egész sort átírod, vagy csak az értékeket változtatod. A sor beírása ill. módosítása után nyomd le a RETURN billentyűt!

b./ Pitagorasz-tétel segítségével a másik befogót kell először kiszámítani.

3.2.2. Adatok bekérése

a./ Az INPUT utasítás kérdőjel kiírásával jelzi, hogy adatot vár. Jegyezd meg, hogy az INPUT PROMPT utasításban a kiírandó szöveg után kettőspontot kell tenni.

b./

```
10 CLS
20 INPUT PROMPT"Kocka oldala=":A
30 INPUT PROMPT"Sürüség=":D
40 V=A^3
50 M=V*D
60 PRINT "Tömeg=";M
```

c./

```
10 CLS
20 PRINT AT 3,6,"Kocka oldala";
30 INPUT A
40 PRINT AT 5,10,;
50 INPUT PROMPT"Sürüsége=":D
60 M=A^3*D
70 PRINT AT 10,12,"Tömege=";M
```

3.2.3. Vizsgálat

a./ Akkor fejeződjön be a program, amikor az átfogó nem nagyobb, mint a befogó, mivel ez nem lehetséges.
A GOTO utasítás helyett írj END, vagy STOP utasítást!

b./

```
34 IF A<=0 THEN 20
36 IF C<=0 THEN 20
```

```
34 IF A<=0 OR C<=0 THEN 20
```

A második megoldásban az OR szó a VAGY logikai műveletet jelenti. A VAGY logikai művelet eredménye akkor lesz 'igaz', ha legalább az egyik 'igaz'.

c./ Relációk: = egyenlő
) nagyobb
(kisebb
<> nem egyenlő
<= kisebb vagy egyenlő
(= nagyobb vagy egyenlő

Logikai műveletek:

NOT
művelet eredménye akkor 'igaz', ha a reláció értéke 'hamis', és akkor 'hamis', ha a reláció értéke 'igaz'.

```
10 CLS:PRINT "igaz-hamis";
20 I$="igaz":H$="hamis"
30 INPUT X$
40 IF NOT(X$="igaz") THEN PRINT I$ :ELSE
PRINT H$
```

AND
művelet eredménye akkor 'igaz', ha mindkét reláció értéke 'igaz' egyébként 'hamis'

OR
művelet eredménye akkor 'hamis', ha mindkét reláció értéke 'hamis'.

XOR
művelet eredménye akkor 'igaz', ha valamelyik reláció értéke 'igaz'.

Próbáld ki a következő programot! A 40-es sorba az OR művelet helyett írj be AND vagy az XOR műveletet!

```
10 CLS:PRINT "i-h";
20 INPUT X$,Y$
30 I$="igaz":H$="hamis"
40 IF X$="i" OR Y$="i" THEN PRINT I$ :EL
SE PRINT H$
```

d./ 'Ha a reláció, vagy a logikai művelet értéke 'igaz', akkor a THEN szó után szereplő utasítást hajtsd végre!
Akkor, ha a vizsgált logikai kifejezés értéke 'hamis', folytasd a program végrehajtását a következő utasítássorral!'
Az IF-THEN-ELSE változatot használva:
'ha a vizsgált reláció, vagy logikai kifejezés értéke 'hamis', akkor az ELSE után írt utasításokat hajtsd végre!'
(Az ELSE szó elé kettőspontot kell tenni!)

e./ Mindkét utasítás hatására a program megáll. A STOP utasítás esetében kiíródik a képernyőre, hogy melyik soron állt meg.

f./ HA (IF) A-B abszolutértéke nagyobb mint E értéke, AKKOR (THEN) írd ki azt, hogy 'nagyobb', EGYÉBKÉNT (:ELSE) azt, hogy 'nem nagyobb'!

g./

```
10 CLS
20 INPUT PROMPT"A két szám:":A,B
30 INPUT PROMPT"A szakasz:":E
40 IF A<B THEN T=A:A=B:B=T
50 IF A-B<=E THEN PRINT "nem ";
60 PRINT "nagyobb"
```

```
10 CLS
20 INPUT PROMPT"A két szám:":A,B
30 INPUT PROMPT"A szakasz:":E
35 REM segédváltozóra nincs szükség
40 IF A>B THEN 60
50 A=A+B:B=A-B:A=A-B
60 IF A-B<=E THEN PRINT "nem ";
70 PRINT "nagyobb"
```

```
10 CLS
20 INPUT PROMPT"A két szám:":A,B
30 INPUT PROMPT"A szakasz:":E
40 IF A>B THEN K=A-B :ELSE K=B-A
50 IF K<=E THEN PRINT "nem ";
60 PRINT "nagyobb"
```

Az első megoldásnál a két szám különbségét úgy képezzük, hogy mindig a nagyobból vonjuk ki a kisebbet.
A második megoldásnál kicseréljük a két változó értékét, ha A kisebb mint B. A cseréhez egy segédváltozót is használunk.
Harmadik megoldás: kiszámítjuk a különbséget.

h./ Határozzuk meg a függvények értelmezési tartományát! Használjuk fel a számítógép segítségét!

```
10 CLS: INPUT PROMPT"x=": X
20 IF X=0 THEN PRINT "nincs értelme": END
30 PRINT "1/x="; 1/X
```

A többi függvénynél a relációt és a kiírást kell átírni: X=2, X<0, X>2

3.2.4. Ciklus

a./ A ciklus belsejébe (a FOR és NEXT közé) írd be:

```
PRINT X%X;X*X*X,1/X,SQR(X)
```

b./

```
10 CLS: INPUT PROMPT"N=": N
20 PRINT "növekvö(i/n)";: GET Q$
25 PRINT
30 K=2: V=2*INT(N/2): L=1
40 IF Q$="i" THEN 60
50 K=V: V=2: L=-1
60 FOR I=K TO V STEP L
70 PRINT I
80 NEXT
```

c./

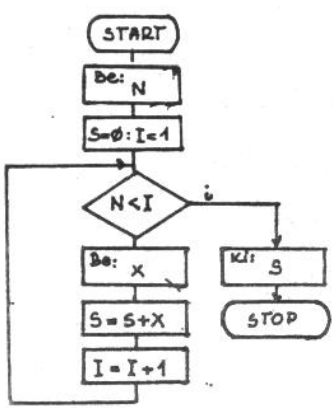
```
10 CLS
20 FOR I=12 TO 99 STEP 3
30 PRINT I
40 NEXT
```

d./ Figyeld meg, hogy a ciklus akkor fejeződik be, ha X értéke már nem esik a K és V értékekkel adott szakaszra.

e./ Amikor a lépésköz 1.

f./ Csak a függvényértékek kiíratását kell a d. feladat programjába beírni.

g./



h./ Annak a változónak a kezdőértéke 1, amelyikben a szorzatot előállítjuk.

i./ 1./ Megvizsgálunk minden egész számot 1-től n-ig.
2./ Azt használjuk ki, hogy a szám felénél nincs nagyobb osztó, csak a szám.
3./ Ha I osztója N-nek, akkor N/I is osztója.
A ciklus négyzetszámok esetén nem úgy működik, ahogyan azt elvárnánk. Próbáld ki a következő programot.

```
10 FOR I=1 TO SQR(25)
20 PRINT I
30 NEXT
```

4./ A számítógép számolási hibáját most másképpen korrigáljuk.

j./ Az STR$ függvénnyel a maradékot átalakítjuk karakterlánccá és azokat fodított sorrendben összefűzzük.
Lásd a 2.4. és a 3.4. fejezeteket.

k./ Ugyanazt a véletlenszerű számsorozatot kapjuk minden futtatásnál. A RANDOMIZE hatására futtatásonként változik.

l./ Találjuk ki, hogy hányszor dob 'írást' a számítógép. A dobások számát úgy számoljuk,hogy egy változó értékét mindig eggyel növeljük. Például ha 'fej' volt a dobás, akkor végrehajtjuk a F=F+1 utasítást.

m./

```
10 CLS:RANDOMIZE
20 FOR I=1 TO 13
30 T= RND
40 IF T<1/3 THEN PRINT "1"
50 IF T>1/3 AND T<2/3 THEN PRINT "x"
60 IF T>2/3 THEN PRINT "2"
70 NEXT
```

```
10 CLS:RANDOMIZE
20 FOR I=1 TO 13
30 T= RND(3)
40 IF T=0 THEN PRINT "1"
50 IF T=1 THEN PRINT "x"
60 IF T=2 THEN PRINT "2"
70 NEXT
```

3.2.5. Indexes változók

a./ A DIM utasítással az index maximális értékét adjuk meg, azaz az azonos nevű indexes változók számát.

b./ N adatot sorba rendez.
A szomszédos elemeket összehasonlítjuk és ha nem jó a sorrend, akkor az értékeiket kicseréljük és a cserét megjegyezzük:Q=1.
Az összehasonlítások végén Q-t vizsgálva eldönthető, hogy történt-e csere. Ha volt csere (Q=1), akkor újból megvizsgáljuk a sorrendet. Ha nem volt (Q=0), akkor a sorbarendezés kész.

3.3. Grafika

3.3.1. Változók

a./ A RETURN billentyű lenyomásával zárjuk az utasítássort. Az így lezárt sorokat a számítógép a memóriájában tárolja, ahonnét a LIST parancsal kilistáztathatjuk.
b./ A GET utasítás azt várja, hogy egy billentyűt leüssünk. A program csak valamelyik billentyű lenyomása után fejeződik be, így az ábrát nem rontja el az 'ok' szó.

c./ Az INPUT utasítás egy kérdőjel kiírásával, az INPUT PROMPT utasítás a kiírt szöveggel jelzi, hogy adatot vár.

d./ Program: F331D

```
   10 REM  --- téglalap ---
   20 CLS:INPUT PROMPT"A téglalap bal felsö
  ponjának   koordinátái":A1,A2
   30 INPUT PROMPT"Oldalak hossza:":A,B
   40 GRAPHICS 16
   50 PLOT A1,A2;A1+A,A2;A1+A,A2+B;
   60 PLOT A1,A2+B;A1,A2
   70 GET:GRAPHICS 4
```

e./ Tetszőleges elhelyezés esetén
1./ négy
2./ hat adat
f./ 1./ Ismételten megadhatjuk, hogy hová toljuk el a háromszöget.
2./ Többször elvégzi a program az eltolást.

g./ Egészítsük ki a 70-es sort az A=2*A utasítással!

h./ Program: F331H

```
   10 REM --- tükrözés ---
   20 CLS:PRINT "a háromszög csúcspontjaina
 k      koordinátái"
   30 INPUT PROMPT"A=":A1,A2
   40 INPUT PROMPT"B=":B1,B2
   50 INPUT PROMPT"C=":C1,C2
   60 GRAPHICS 2
   70 PLOT A1,A2;B1,B2;C1,C2;A1,A2
   80 PLOT 500,0;500,958:REM tengely
   90 P1=500+(500-A1):Q1=500+500-B1:R1=1000
 -C1
  100 PLOT P1,A2;Q1,B2;R1,C2;P1,A2
```

3.3.2. Vizsgálat

a./ Például egy PLOT ,0,0 utasítással.

b./ Lásd a 3.2.3/c feladat megoldását!

c./ A GET utasítás várja, hogy leüssünk egy billentyűt. Az INKEY$ függvény végrehajtásakor nem várakozik. Azt a karaktert olvassa be, amelyik éppen le volt nyomva. Ha nem nyomtunk le billentyűt, akkor az értéke az 'üres karakter' lesz.

d./ A CHR$ függvény a karakter kódjából állítja elő a karaktert.
Az ORD függvény a karakterlánc első karakterének kódját adja meg.

e./ Ezek a változók őrzik meg az előző pont koordinátáit.

f./ Akkor, ha a reláció értéke 'igaz'.

g./ Az a cím, ahol a program futása folytatódik, ha a reláció értéke 'igaz'. Ugyanazt jelenti, mint a THEN szó után írt GOTO utasítás.

3.3.3. Ciklus

a./ A PLOT utasítással kijelöljük a helyet, és a PRINT#0, utasítással kiíratjuk a beosztáshoz tartozó számot.

```
 10 REM --- számegyenes-2 ---
 20 CLS:INPUT PROMPT"Egység=":E
 30 GRAPHICS 2:PLOT 1000,496:PRINT#0,">"
 40 PLOT 0,480;1006,480,
 50 FOR X=0 TO 470 STEP E
 60 PLOT 500+X,485;500+X,475
 70 PLOT 500+X-23,460:PRINT#0,X/E
 80 PLOT 500-X,485;500-X,475
 90 PLOT 500-X-23,460:PRINT#0,-X/E
100 NEXT X
```

b./ A STRING$ függvénnyel előállított szóköz karakterekkel töröljük az előző kiírást.
A PRINT USING utasítással egy tizedes pontossággal írjuk ki a ciklusváltozó értékét.

c./ A ciklus mindig akkor fejeződik be, amikor a ciklusváltozó értéke már nem esik a kezdő- és végértékkel adott szakaszra.

d./

```
 10 REM --- céltábla ---
 20 GRAPHICS 2
 30 U=500:V=500
 40 FOR I=1 TO 10
 50 A=20*I:B=A:GOSUB 1000
 60 IF I/2<>INT(I/2) THEN 80
 70 PLOT U+A-10,V,PAINT
 80 NEXT
 90 END
```

3.3.4. Függvény, szubrutin

a./ A rajz durvább lesz.

b./ Kiszámítjuk
120: a képernyőre eső X értékeket,
130: X-hez tartozó Y értékeket,
140: az Y értéknek megfelelő képernyőpont függőleges koordinátáját. (Az 1.4-es szorzó a torzítás miatt kell.)

c./ Így nem látjuk a képernyő szélére helyezett pontokat.

d./ A 110-es sort kell átírni.

e./ Program:F334E

```
300 REM  --- metszéspont ---
310 IF SGN(FNY(X)-FNZ(X))=SGN(FNY(X1)-FNZ
(X1)) THEN 370
350 SET INK 1:PLOT I,J;I,480:PLOT ,I,J
360 G=G+1:PRINT AT 18+2*G,50,"X"&STR$(G)&
"=";:PRINT USING"####.#":X
370 SET INK 0:RETURN
```

f./ Lásd a 3.3.1./D feladatot !

g./ Használd a READ és a DATA utasításokat!

h./ Program: HALMAZMUV
Mutasd be programmal a halmazműveleteket!

i./ Az előzőek alapján már nem lesz nehéz elkészíteni a programot. Színezéssel a következő fejezet foglalkozik.

3.4. Karakterláncok

3.4.1. Műveletek, függvények

a./ A PRINT utasításban igen. Az INPUT PROMPT utasítással csak egy karakterlánc írható ki, ezért itt össze kell fűzni a két karakterláncot.

b./ LEN - a karakterlánc hosszát,
ORD - a karakterlánc első karakterének kódját adja meg.
CHR$ - a karakterkódból előállítja a karaktert.
VAL - a számjegyekből álló karakterláncból számot képez.
STR$ - aritmetikai kifejezés értékéből karakterláncot állít elő.

c./ LEN,ORD,VAL - a függvény argumentuma karakterlánc, értéke numerikus adat
CHR$ STR$ - a függvény argumentuma numerikus adat, értéke karakterlánc.

3.4.3. Vizsgálat

a./ Lásd a 3.2.3./d feladat megoldását!

b./ Program: F342B

```
10 REM  --- karakterek vizsgálata-2 ---
20 CLS:PRINT AT 3,2,"Nyomj le egy billen
tyüt!":PRINT
30 GET A$
40 IF A$>="0" AND A$<="9" THEN PRINT "sz
ámjegy":STOP
50 IF A$>="A" AND A$<="Z" OR A$>="Á" AND
A$<="Ö" THEN PRINT "nagybetü":STOP
60 IF A$>="a" AND A$<="z" OR A$>="á" AND
A$<="ü" THEN PRINT "kisbetü":STOP
70 PRINT "egyik sem"
```

3.4.3. Ciklus

a./ 30 FOR I=LEN(W$) TO 1 STEP -1

b./ Program: F3438

```
   10 REM  --- a szó magánhangzói ---
   20 DIM S$*100
   30 CLS:INPUT PROMPT"A szó:":S$
   40 FOR I=1 TO LEN(S$)
   50 Q$=S$(I)
   60 IF Q$="a" OR Q$="á" OR Q$="e" OR Q$="
é" OR Q$="o" OR Q$="ó" OR Q$="ö" OR Q$="ő"
 OR Q$="ö" OR Q$="ő" OR Q$="u" OR Q$="ú" O
R Q$="ü" OR Q$="ű" THEN S$(I)="e"
   70 NEXT
   80 PRINT S$
```

c./ Egyszerűbb számolni a magánhangzókat. Ezt ismerve már kiszámíthatjuk a mássalhangzók számát.

d./ 'Végtelenítjük' a számlálást.

e./ A CHR$(8) kiírása a DEL billentyű lenyomásával egyenértékű. A DEL billentyű lenyomásának figyelése az ORD(X$)=8 reláció vizsgálatával történhet.

3.5. Színek

a./

```
   10 REM --- szinek-1 ---
   20 GRAPHICS 16
   30 FOR I=0 TO 15
   35 IF I=8 THEN RESTORE
   40 READ SZiN$
   45 IF I<8 THEN SZiN$="sötét"&SZiN$
   47 IF I=7 THEN SZiN$="szürke"
   50 SET INK I
   60 PRINT AT I+5,1,I;" - ";SZiN$;
   70 NEXT
   80 GET:GRAPHICS 4
  100 REM --- szinek ---
  130 DATA fekete,kék,vörös
  140 DATA bibor,zöld,cián,sárga
  150 DATA fehér
```

b./ Az első két színt tudod használni.

c./ SET BORDER - a keret színének,
SET INK - a 'ceruza' színének,
SET PAPER - a 'papír' színének kiválasztása.
SET PALETTE - a színek kijelölése.
SET MODE - a színek keverése. (Ezzel az utasítással nem foglalkoztunk.)

5. Melléklet

5.1. Definiált karakterek

	32	A	65	a	97	Á	128
!	33	B	66	b	98	É	129
"	34	C	67	c	99	Í	130
#	35	D	68	d	100	Ó	131
$	36	E	69	e	101	Ö	132
%	37	F	70	f	102	Ő	133
&	38	G	71	g	103	Ú	134
'	39	H	72	h	104	Ü	135
(	40	I	73	i	105	Ű	136
)	41	J	74	j	106	┌	137
*	42	K	75	k	107	┐	138
+	43	L	76	l	108	│	139
,	44	M	77	m	109	├	140
-	45	N	78	n	110	┬	141
.	46	O	79	o	111	┼	142
/	47	P	80	p	112	▒	143
0	48	Q	81	q	113	á	144
1	49	R	82	r	114	é	145
2	50	S	83	s	115	í	146
3	51	T	84	t	116	ó	147
4	52	U	85	u	117	ö	148
5	53	V	86	v	118	ő	149
6	54	W	87	w	119	ú	150
7	55	X	88	x	120	ü	151
8	56	Y	89	y	121	ű	152
9	57	Z	90	z	122	┐	153
:	58	[	91	{	123	┘	154
;	59	\	92	¦	124	─	155
<	60	]	93	}	125	┤	156
=	61	^	94	~	126	┴	157
>	62	_	95	■	127	▒	158
?	63	`	96			▒	159
@	64						

32-ig: vezérlő karakterek

160-tól: definiálható karakterek

5.2. Színek

Sorszám	Színkód	Sorszám	Színkód
0 - fekete	0	8 - fekete	64
1 - sötétkék	1	9 - kék	65
2 - sötétvörös	4	10 - vörös	68
3 - sötétlila	5	11 - lila	69
4 - sötétzöld	16	12 - zöld	80
5 - sötét kékeszöld	17	13 - kékeszöld	81
6 - sötétsárga	20	14 - sárga	84
7 - szürke	21	15 - fehér	85

5.3. Programok a Programcsomagból

BASIC kulcsszavak - Program neve: KULCS

A BASIC parancs és utasításszavak, függvénynevek, műveleti és relációjelek jelentésének tanulását segíti.

Pontok koordinátái - Program neve: PONTOK

A koordináta-rendszerben megjelenő pont koordinátáit kell megadni ill. a koordinátáival adott pontot kell megkeresni.

Egyszerűsítés - Program neve: EGYSZERU

A törtek egyszerűsítésének gyakorlása. Segítségül kérhető a számláló és a nevező primtényezős felbontása.

Egyenletek grafikus megoldása - Program neve: EGYGRAF

Input adatként megadható két - első ill. másodfokú - függvényt ábrázol, és megkeresi a metszéspontok abszcisszáját.

Sorbarendezés - Program neve: SORREND

A sorbarendezés algoritmusát szemlélteti

Névsor a magyar ábécé szerint - Program neve: MAGYAR

Karakterláncok összehasonlítása az angol ábécé, azaz a karakterek kódszáma szerint történik. A hosszú magánhangzók kódja nem illeszkedik ebbe a kódszám-rendszerbe, ezért a magyar ábécé szerint a rendezést nehezebb megoldani.

Négyszínű képernyő - Program neve: NEGYSZIN

A négyszínű képernyő színezésére szolgáló utasítások tanítását segíti.

Spirálok, görbék - Program neve: SPIRALOK

A különböző spirálok, görbék bemutatásán kívül, a menü rendszerű programszerkezet tanításához is felhasználható.

Karakter nagyítás - Program neve: NAGYKAR

A grafikus képernyőpontok vizsgálatával rajzolja ki a karaktert.

Kikerülés - Program neve: UTKOZES

Az 'ütközéses' játék egyik egyszerű változata

Számlétra - Program neve: SZAMLET

Bombázás - Program neve: BOMBAZ

A demonstrációs programcsomag játékprogramja tanítható formában.

LY vagy J - Program neve: HELYES

Közismert oktatóprogram 'nyitott' változata.

Tartalom