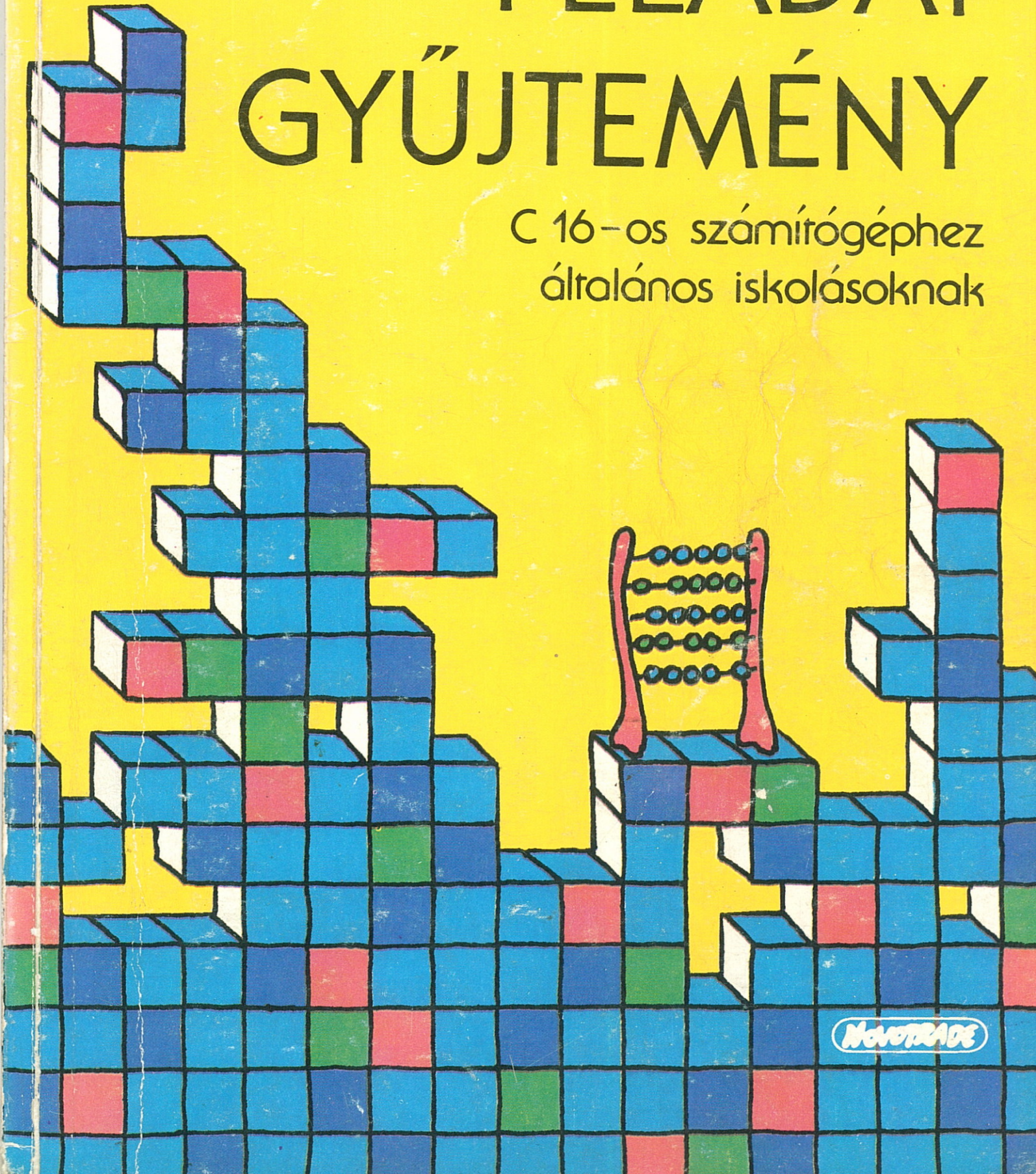


Bencsikné Takács Márta

# FELADAT- GYŰJTEMÉNY

C 16-os számítógéphez  
általános iskolásoknak



NOVOTRADE



**Feladatgyűjtemény**  
**COMMODORE 16 számítógépre**  
**általános iskolásoknak**

**Budapest, 1986**



**Készítette:**  
**BENCSIKNÉ TAKÁCS MÁRTA**

**Lektorok:**  
**DR. KÓROSNÉ MIKIS MÁRTA**  
**MAROSVÁRY ERIKA**

**Felelős szerkesztő:**  
**TARR KÁLMÁNNÉ**

**Köszönetet mondunk a budapesti Eötvös Gimnázium Számítástechnikai Szakkörének a könyvben szereplő programok ellenőrzéséért és kiprinteléséért.**

**Vezető tanár: VARGA BALÁZS**

**A munkában az alábbi tanulók vettek részt:**

**SZENTIVÁNYI ANDRÁS**

**PAÁL ZOLTÁN**

**GALAMBOS ZOLTÁN**

**SCHMIDER LÁSZLÓ**

**CSÁNYI CSABA**

Készült az OPI és a TII között létrejött állami szerződés keretében  
Kiadta a NOVOTRADE RT. Felelős kiadó: Rényi Gábor igazgató  
Kiadványmenedzser: Békés Tamás. Műszaki szerkesztő: Dévényi Erika  
A fedélterv R. Kovács Anna munkája  
ISBN NOVOTRADE RT.: 9630241099 ISBN OPI: 9636819602

---

Készült a Somogy Megyei Nyomdaipari Vállalat  
kaposvári üzemében – 85-6734 – 17 000 példányban (19,6 A/5 ív)  
Felelős vezető: Mike Ferenc igazgató



Kedves Olvasó!

Érdekel a számítástechnika, de gondot okoz számodra megfelelő és érdekes feladatok beszerzése, szeretnéd megtanulni a BASIC nyelvet, de még nem tudod, hogyan fogj hozzá, szeretnél programozni, de kifogytál az ötletekből?

Neked írtam ezt a könyvet!

Neked írtam, mert még nem tudsz programozni. Olvasd végig, és oldd meg sorban a feladatokat, és megismerkedsz a Commodore 16 érdekességeivel. Segítségére lehetnek a következő könyvek: HETEDHÉT sorozat, valamint a Commodore 16-os Felhasználói kézikönyv.

Már tudsz programozni, és szeretnél érdekes feladatokat megoldani; neked is írtam ezt a könyvet. Lapozgass a könyvet, és válogass kedved szerint, de vigyázz, mert a feladatok a témákon belül egyre nehezednek, és a „Néhány vegyes feladat” című csoport már komolyabb tudást feltételez.

A könyv tartalmazza a feladatok megoldását is. Ha úgy érzed, hogy nem tudsz megoldani egy feladatot, lapozz hátra, és mielőtt megnéznéd a megoldást, olvasd el, mit találsz a „Vegyük észre” megjegyzés után! Az ott olvasható ötletek remélem, segítségre lesznek.

A közölt programok nem feltétlenül a legegyszerűbbek, de találhatsz bennük olyan ötleteket is, melyeket később te is felhasználhatsz. Ezért akkor is érdemes elemezned a megoldást, ha a te programod is működik.

Mielőtt programozni kezdenél, javaslom, hogy tervezd el, mit akarsz csinálni, és erre mi lehet a legcélravezetőbb eljárás. A tervezésben hasznos segítő lehet egy jól megrajzolt folyamatábra.

Vigyázzat, ha egy program fut, az még nem jelenti azt, hogy jó is. A jó program a feladat által megengedett összes érték esetén működik. Ezért egy kész programot mindig tesztelni kell. Mivel a tesztelést csak véges számú elemmel végezhetjük el, meg kell állapítanunk a feladat szempontjából kritikus értékeket, és a próbafuttatásokat ezekre az értékekre kell elvégezni. Némi gyakorlattal ezek a kritikus értékek könnyen megtalálhatók.

A jó programozó nemcsak a programíráshoz, hanem a programolvasáshoz is ért. Ez azt jelenti, hogy idegen programokról is mindig meg tudja állapítani, hogy azok mit csinálnak. Ilyen típusú feladatokat tartalmaz a „Mit csinál a gép” című fejezet. Javaslom az áttanulmányozását mindenkinek.

Remélem, a feladatok között te is találsz kedvedre valót! A további munkához sok sikert és kellemes időtöltést kíván:

a szerző.







4. Gépeled be az összes karaktert a billentyűzetről!
  - a) Töröld a képernyőt jobbról balra!
5. Jelenítsd meg az összes grafikus jelet!
  - a) Töröld balról jobbra a képernyőt!
6. Töröld a képernyőt!
  - a) Mozgasd a kurzort a képernyő középső sorában és írd ki: COMMO. DORE 16!
  - b) Rajzolj a szöveg alá és fölé díszítő sorokat!
7. Írd tele a képernyőt úgy, hogy azon az összes betű és azok inverze felváltva szerepeljen!

8. Írj a képernyőre csupa kisbetűt!

9. Fejtsd meg az itt látható titkosírást a számítógép segítségével!

```

-♦|  ♣ ♠♦X|T|T  ♣ ♠ R-|  T|L|T|  T|  L|T|IN♣♦♦X♣L  ♠♦♦-|T|T|T|
\T|-4T\  ♣ H\N-T-T  T|N|  L|T|IN♣♦♦X♣L  L|T|
-4|T  ♠ -4  ♠ ♦-T|T|T|T|
  
```

10. Írd ki a képernyőre: Magyarország fővárosa Budapest!
11. Írd tele a képernyőt a neveddel úgy, hogy a kis- és nagybetűket értelem-szerűen használod!
12. Rajzolj az üres képernyőre különböző ábrákat! Pl. díszítő sorokat, házat, ajtót stb.

## 1.2 Játékok

13. Amőba: Rajzolj körül egy tetszőleges mezőt a képernyőn! Válassz két eltérő jelet (karaktert) a billentyűzetről! Már játszhattok is!
14. Malom: + és - jelekkel malomtáblát rajzolhatsz! Két különböző jel (karakter) felhasználásával már játszhattok is. Ügyelj arra, hogy miután elléptél egy mezőről, annak helyét újra kell rajzolni!
15. Ki húzza az utolsót?  
Rajzolj a képernyőre 100 tetszőleges karaktert! Társaddal felváltva 1–3 jelet törölhettek el. Az nyer, akié az utolsó húzás.

## 2. SZÁMOK ÉS MŰVELETEK

### 2.1 Példák

16. Próbáld ki, mit tapasztalsz!
  - a) 3+8
  - b) PRINT 3+8
  - c) ?3+8



17. Próbáld ki, és válaszolj a kérdésekre!

a/1. PRINT (31+2)\*5

a/2. PRINT 31+2\*5

b/1. PRINT 2\*(3+5)

b/2. PRINT 2\*3+5

c/1. PRINT (2+4)+6

c/2. PRINT 2+(4+6)

d/1. PRINT (3\*5)\*2

d/2. PRINT 3\*(5\*2)

Indokold meg:

1. Miért más az eredmény a/1.-a/2.

b/1.-b/2. esetben?

2. Miért azonos az eredmény c/1.-c/2.

d/1.-d/2. esetben?

3. Milyen sorrendben végzi a gép a műveleteket az egyes példákban?

18. Milyen sorrendben végeznéd a műveleteket?

a)  $(3+2)*15-8/9$     b)  $3 \uparrow 2+(5+6+9)*12-8$

19. Végeztesd el a géppel a következő műveleteket!

a)  $9-(6+5)*4$     b)  $(2+3)*(4+7)$     c)  $(9+14)/11$

d)  $3-5*8+3$     e)  $2-5/4+7$     f)  $2 \uparrow 2-7/6$

Mi lehet a műveletvégzés sorrendje az egyes esetekben?

20. Számítsd ki a géppel!

a) Mekkora annak a téglalapnak a kerülete és a területe, melynek oldalai 11 cm és 4,9 cm?

b) Mekkora annak a derékszögű háromszögnek a kerülete és a területe, melynek oldalai 3 cm, 4 cm, 5 cm?

c) Mekkora annak a körnek a kerülete és a területe, melynek sugara 12,4 cm? Használd a  $\pi$  billentyűt!

21. Írassz szöveget a géppel!

Próbáld ki, mit tapasztalsz!

a) PRINT "BARÁT"    b) PRINT "FÜLE"    c) PRINT "BARÁT", "FÜLE"    d) PRINT "BARÁT";"FÜLE"

22. Próbáld ki, hogy maximum hány karaktert írhatasz egy PRINT utasítás után!

## 2.2 A változók

23. Legyen  $A=123$ ,  $B=2,9$ . Írassd ki "A"-t, A-t, "B"-t, B-t!

24. Legyen  $A\%_0=22$ ,  $B\%_0=7$ . Írassd ki "A%\_0"-t, A%\_0-t, "B%\_0"-t, B%\_0-t!

25. Legyen  $A\$="BASIC"$ . Írassd ki "A\\$"-t, A\\$-t!

26. Legyen  $A\%_0=A$ . Mekkora  $A\%_0$  értéke, ha

a)  $A=45$

b)  $A=2,3$

c)  $A=-2$

d)  $A=-5,4$ ?

27. Legyen  $A=8,5$ ,  $B=7$ . Írassd ki:

a)  $A+B$

b)  $A-B$

c)  $A*B$

d)  $A/B$

e)  $A \uparrow B$

f)  $A;B$

g)  $A,B$



28. Milyen típusú változókat célszerű használni a következő feladatok megoldásánál?
- Tetszőleges szövegrészek kiíratására.
  - Tetszőleges szövegek és a tízes szorzótábla elemeinek kiíratására.
  - Tetszőleges szám 70<sup>0</sup>/<sub>0</sub>-ának kiszámítására.
  - Tetszőleges oldalhosszúságú sokszög kerületének és területének kiszámítására.
  - Egész mérőszámú sugár esetén kör kerületének és területének kiszámítására.
29. Legyen egy háromszög alapja A, magassága M.
- Adj tetszőleges értéket A-nak és M-nek. Legyen  $X=(A*M)/2$ . Írasd ki X-et! Milyen adatot számoltál ki?
  - Így jelenjen meg az eredmény:  $T= \dots CM \uparrow 2!$
30. Legyen A\$="ALMA", B\$="KÖRTE", A=3, B=2.  
Írasd ki a változók segítségével: 3 ALMA 2 KÖRTE
- egymás alá
  - egymás mellé zónázva
  - egymás mellé tömörítve

### 3. A PROGRAMOZÁS

#### 3.1 A PRINT utasítás

31. Írj programot, amely letörli a képernyőt és a következőket jeleníti meg:  
BUDA PEST  
BUDAPEST  
BUDA PEST  
BUDAPEST  
Használj azonosítókat!
32. Írj programot, amely a 60. és 15,5 számokkal elvégzi a négy alpműveletet. Az eredményeket írd ki!
33. Bővítsd úgy a programot, hogy a gép az eredmények elé írja oda azt is, hogy éppen mit számol!  
Pl.: AZ ÖSSZEG=75.5.
34. Bővítsd úgy az előző programot, hogy a gép a műveletvégzés szempontjából azonos erősségű műveleteket egymás mellé zónázva írja ki!
35. Írj programot, amely az üres képernyő közepére kiírja: COMMODORE 16!  
Használd ki a kurzormozgató adta lehetőségeket!
36. Írjad ki kurzormozgató segítségével az üres képernyőre:
- |         |         |         |
|---------|---------|---------|
| a) ALMA | b) ALMA | c) ALMA |
| ALMA    | ALMA    | ALMA    |
| ALMA    | ALMA    | ALMA    |
37. Oldd meg az előző feladatot PRINT TAB(X) parancs segítségével!



38. Írj programot, amely a + jel segítségével házat rajzol! (Hasonló módszerrel más. ábrákat is készíthetsz!)

39. Írj programot, amely a hármasszorozótábla első három elemét táblázatba foglalja!

Ilyen fejléccet alkalmazsz:

A HÁRMOS SZORZÓTÁBLA ELEMEI	
SORSZÁM	SZORZAT

40. Írj programot, amely két díszítősor közé a képernyő közepére írja: BASIC!

### 3.2 A változók cseréje

41. Legyen A és B tetszőleges szám. Cseréld fel A és B értékét a memóriában! Írasd ki a kezdeti és a végállapotot!

42. Elemezd, majd ellenőrzésként futtasd le!  
Mit tapasztalsz? Magyarázd!

a)

```

10 A=7
20 B=5
30 A=B
40 C=A+B
50 PRINT C
    
```

b)

```

10 A=7
20 B=5
30 B=A
40 C=A+B
50 PRINT C
    
```

43. Elemezd a következő programokat! Mekkora C értéke az egyes esetekben?

a)

```

10 A=9
20 B=12
30 B=A
40 C=B-A
50 PRINT C
    
```

b)

```

10 A=7
20 B=3
30 A=A+1
40 B=A
50 C=A+B
60 PRINT C
    
```

c)

```

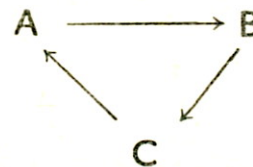
10 A=4
20 B=8
30 A=B
40 A=A+3
50 B=A
60 C=A+B
70 PRINT C
    
```

d)

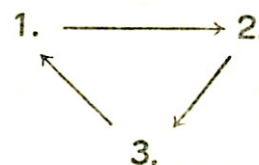
```

10 A=2
20 B=4
30 B=B/2
40 A=B-8
50 B=B+1
60 C=A-B
70 PRINT C
    
```

44. Legyen A=3, B=4, C=5. Cseréld fel a memóriában A, B, C változók értékét az ábra szerint! Írasd ki a kezdeti és a végállapotot!



45. Egy szekrényben ruhaneműk vannak. Az első fiókban zokni, a másodikban kesztyű, a harmadikban zsebkendő van. Cseréld fel a fiókok tartalmát, és írd ki a kezdeti és a végállapotot!



### 3.3 Az INPUT utasítás

46. Írj programot, amely két tetszőlegesen begépelte számot kiír a képernyőre!  
a) Azonos sorrendben zónázva  
b) Azonos sorrendben tömörítve  
c) Azonos sorrendben egymás alá!
47. Írj programot, amely két tetszőleges szám összegét kiszámítja és kiírja az eredményt!
48. Írj programot, amely kiírja egy tetszőlegesen begépelte szám négyzetét!
49. Írj programot, amely két nullától különböző tetszőleges szám hányadosát kiszámítja, és a képernyőn így jeleníti meg:
- AZ OSZTANDO= ...  
AZ OSZTO= ...  
A HANYADOS= ...
50. Írj programot, amely két nullától különböző tetszőleges számmal elvégzi a négy alpműveletet, és az eredmények elé azt is kiírja, hogy mit számolt!
51. Írj programot, amely kiszámítja két tetszőleges szám számtani közepét (átlagát). Írd meg a programot úgy, hogy az minden használó számára érthető legyen!
52. Írj programot, amely tetszőleges téglalap kerületét és területét kiszámítja, a cm-ben megadott oldalak segítségével!
53. Bővítsd úgy a programot, hogy az minden használó számára érthető legyen! Az eredmények után írja ki a gép a megfelelő mértékegységet is!
54. Készíts programot, amely a sugár és a  $\pi$  billentyű segítségével kiszámítja tetszőleges méretű kör kerületét és területét!
55. Írj programot, amely kiszámítja az  $Y=3*X+4$  függvény értékét tetszőleges X helyen!
56. Írj programot, amely kiszámítja az  $Y=-1/5*X+1$  függvény értékét tetszőleges X helyen!
57. Írj programot, amely tetszőlegesen meghatározott első fokú függvény értékét kiszámítja tetszőleges X helyen!
58. Írj programot, amely kiszámítja tetszőleges szám tetszőleges százalékát!
59. Ismert feladat: Három pók három nap alatt három hálót sző. Két pók két nap alatt hány hálót sző? Írj programot, amely tetszőleges számú pók és tetszőleges számú nap esetén is megadja a megfelelő választ!



60. Egy molylepke hernyója egy nap alatt tíz könyvlapon, ill. egy borítón tudja átrágni magát.  
Egy könyvespolc ugyanazon sorában azonos oldalszámú, pl. 100 oldalas könyvek vannak. Írj programot, amely a könyvek számának megadásával meghatározza az átrágásukhoz szükséges időt!
61. Készíts szövetárazó programot!  
Az egységár és a mennyiség alapján adja meg a gép egy vég szövet árát!
62. Írj programot, amely az órában, percben, másodpercben megadott időtartamot másodpercben írja ki!
63. Készíts mértékváltó programot!  
A gép a méterben megadott hosszúságot dm, cm, mm, km-re váltsa át!
64. Számkitaláló játék.  
Gondolj egy számot! Vegyél el belőle 28-at, szorozd meg 8-cal és adj hozzá 13-at! Az így kapott eredményt géped be, és a gép megmondja, mire gondoltál! Írj programot, amely ezt a játékot játssza!
65. Lakcímjegyzék.  
Kérje a gép három barátod nevét és lakcímét, majd táblázatba rendezve írja ki azokat!

### 3.4 Néhány gyakori függvény

#### Az egészrész függvény $\text{INT}(X)$

66. Írj programot, amely megadja tetszőleges szám egészrészét!
67. Készíts programot, amely kiírja tetszőleges szám egészekre kerekített értékét!
68. Egy dolgozat értékelésénél a következő ponthatárok vannak:
- |       |   |
|-------|---|
| 0– 4  | 1 |
| 5– 9  | 2 |
| 10–14 | 3 |
| 15–19 | 4 |
| 20–24 | 5 |
- A tanuló nevének és pontszámának begépelése után írja ki a gép, hogy a tanuló hányas dolgozatot írt!
69. Készíts címletező programot!  
Adja meg a gép, hogy 16 Ft-ot hogyan lehet a legkevesebb pénzdarabbal kifizetni!
70. Írj programot, amely a 11-es tízes számrendszerbeli számot hármasszámrendszerbeli számmá váltja át!
71. Írj programot, amely egy tanuló nevének és életkorának begépelése után

kiírja, hogy a tanuló hányadik osztályos! (Feltételezzük, hogy a gyerek általános iskolás és nem bukott, nem évvesztés!)

### A négyzetgyök függvény $SQR(X)$

72. Írj programot, amely kiszámítja tetszőleges szám négyzetgyökét!
73. Számoldtasd ki a géppel a derékszögű háromszög két befogójának ismeretében az átfogó hosszát, a háromszög területét és területét!

### A véletlenszámot előállító (generáló) függvény $RND(X)$

74. Írass ki a monitorra öt darab 1–15 közötti egész véletlenszámot!
75. Írass ki a monitorra öt darab 32–100 közötti egész véletlenszámot!
76. Lottó! Adjon a gép öt darab 1–90 közötti egész véletlenszámot! Te tippelhetsz. Játshatjátok többen is. Az nyer, akinek több találata van.
77. Kockajáték! A gép 1–6 között egész véletlenszámot ír ki. Az első három dobás eredménye a tiéd, a másik három a partneredé. Az nyer, akinek több pontja van.

### 3.5 Az $IF \dots THEN$ és a $GOTO$ kapcsolata

78. Írj programot, amely egy tetszőleges számról eldönti, hogy pozitív-e vagy nem!
79. Döntse el a gép két nullától különböző számról, hogy azok azonos előjelűek-e!
80. Készíts programot, amely meghatározza egy tetszőleges szám előjelét!
81. Írj programot, amely megadja a begépelte szám abszolútértékét!
82. Írj programot, amely két emberről a nevük és a testmagasságuk alapján eldönti, hogy ki a magasabb!
83. Írd át a programot úgy, hogy az akkor is jól működjön, ha a két ember egyforma magas!
84. Írj programot, amely megadja egy begépelte szám egészekre kerekített értékét  $IF \dots THEN$  és  $GOTO$  segítségével!
85. Írj programrészletet, amely lehetővé teszi, hogy csak te tudd futtatni a programodat!
86. Számítsa ki a gép két tetszőleges pozitív szám összegét! Rossz adat esetén jelezzen és kérjen új értékeket a gép!





102. Írjad ki a géppel az ALMA szót ilyen formában:

a) ALMA

b) ALMA

c) ALMA

d) ALMA

ALMA

ALMA

ALMA

ALMA

ALMA

ALMA

ALMA

ALMA

.

.

.

.

.

.

.

.

.

.

.

.

103. Írj programot, amely kiszámítja az első száz természetes szám összegét (Gauss feladata)!

a) Írd át úgy a feladatot, hogy az írja ki a részletösszegeket is!

b) Változtasd meg úgy a feladatot, hogy csak azt a részletösszeget írja ki, amit előzőleg begépettünk!

104. Írj programot, amely táblázatba foglalja az első 20 természetes számot és azok négyzetgyökét!

105. Írj programot, amely elkészíti tetszőleges egész szám első tíz hatványának táblázatát!

106. Ha pénzünket takarékbba tesszük, az évente kamatozik, átlag  $3\%$ -ot. Írj programot, amely megadja az évek számát és a betett összeg alapján a kamattal együtt kivehető összeget!

107. Egy dolgozat ponthatárai:

0– 4      1

5– 9      2

10–14     3

15–19     4

20–24     5

Írj programot, amely a tanuló neve és pontszámai alapján megállapítja XY érdemjegyét!

108. Írd át úgy az előző programot, hogy az annyiszor fusson, amennyit előre meghatározunk, és számítsa ki az osztály átlagát is!

109. Írj programot, amely táblázatba gyűjti az előre meghatározott területértékig az összes egész számmal megadható oldalhosszúságú négyzet oldalának mérőszámát, kerületét és területét!

110. Bővítsd úgy az előző programot, hogy az tetszőlegesen sokszor fusson!

111. Egy lombikban egysejtűek szaporodnak. Az első percben egy darab van a lombikban, de minden percben ketté osztódik minden egysejtű. Írj programot, amely akárhányadik percben megadja a lombikban levő egysejtűek számát! A program tetszőlegesen sokszor fusson!



**112.** A táptalaj kimerülésével az előbbi egysejtűek szaporodása is lassul, ugyanis minden ötödik perc végén elpusztul az állomány fele. Írnod ki táblázatosan egy óra időtartam alatt percenként hogyan alakul az egysejtűek száma! Úgy írod ki az értékeket, hogy azok olvasható gyorsasággal kövessék egymást!

### 3.7 A tömb

**113.** Írj programot, amely kiszámítja az első 30 pozitív szám összegét! Gyűjtsd össze a részletösszegeket is, és írod ki őket!

a) Módosítsd úgy a feladatot, hogy a gép csak minden második részletösszeget írjon ki!

b) Módosítsd úgy a programot, hogy a gép csak minden 1., 3. stb. számú részletösszeget írjon ki!

c) Módosítsd úgy az alapfeladatot, hogy az tetszőleges számú részletösszeget írjon ki!

**114.** Írj olyan programot, amely elhelyezi az első 100 pozitív szám négyzetét egy tömbben! Ezek közül bármely szám négyzetét kiíráthassuk! Fusson tetszőlegesen sokszor a program!

**115.** Egy kiskereskedő 20 féle zöldséget árul. Írj programot, amely a zöldség neve és mennyisége alapján elkészíti a napi leltárt!

**116.** Az előbbi feladatban szereplő kereskedő a hét minden napján folyamatosan rögzíti az adatokat. Készíts programot, amely megadja az egész heti leltárt napi bontásban!

**117.** Írass ki a géppel 100 darab  $-33$  és  $33$  közötti számot! Írass ki ezeket előjelük szerint csoportosítva is!

**118.** Írj programot, amely pénzfeldobást szimulál! Írass ki 100 dobás eredményét (fej, írás) olvasható gyorsasággal! Írass ki azt is, hogy 100 dobásból összesen hány volt írás, hány volt fej!

**119.** Pénzdobáló játék.

A gép hatszor dob. Ebből az első három a tied, a második három a gépé. Az írás 1, a fej 2 pontot ér. Írja ki a gép a dobásokat olvasható gyorsasággal, azt, hogy kinek hány pontja van és azt is, hogy ki nyert! Fusson tetszőlegesen sokszor a program!

**120.** Anna almát, Kati körtét, Szilvi szilvát hozott uzsonnára. Mivel egyik lánynak sem az volt a kedvence, amit hozott, hát elcserélték egymás között. De még így sem volt az igazi, ezért még egyszer továbbadták a náluk levő gyümölcsöt. Írj programot, amely tetszőleges számú csere után megadja, hogy kinél milyen gyümölcs van éppen!

**121.** Írj programot, amely meghatározza tetszőleges szám összes osztóját!



### 3.8 FOR...TO...STEP és NEXT ciklusutasítások

- 122.** Készíts programot, amely kiírja az egész számokat 1-től 100-ig!
- Változtasd meg úgy a programot, hogy az minden második számot írjon csak ki!
  - Írd át úgy a programot, hogy az az 1., 3., 5. ... számokat írja csak ki!
  - Módosítsd a programot, hogy az a számokat most 100-tól 1-ig írja ki!
- 123.** Írj programot, amely kiszámítja az első harminc pozitív egész szám szorzatát!
- Módosítsd az alapprogramot úgy, hogy az a részletsorzatokat is kiírja!
  - Írjon ki a gép minden második részletsorzatot!
  - Írjon ki a gép minden 1., 3. ... részletsorzatot!
  - Változtasd meg úgy a programot, hogy a használó akárhányadik részletsorzatot kiíráthassa a géppel!
- 124.** Írj programot, amely telerajzolja a képernyőt csillagokkal!
- 125.** Írj programot, amely úgy rajzolja tele a képernyőt csillagokkal, hogy minden csillag után üres helyet hagy!
- 126.** Írass ki 100 darab egy és száz közötti egész számot és a véletlenszámok átlagát! Futtasd többször a programot! Mit tapasztalsz?
- 127.** Szimuláltass a géppel kockadobálást! 200 dobás után írasd ki azt is, hogy melyik szám hányszor szerepelt! Számítsd ki a dobások átlagát is! Futtasd többször a programot! Mit tapasztalsz?
- 128.** Írj TORPEDÓ játékot!
- Játékszabály: A gép egy 10-szer 10-es négyzetben eldug egy darab egy karakter nagyságú hajót. A hajó egy helyben áll. A hajó helyére koordináták begépelésével lehet tippelni. A gép számolja a tippek számát, és megjegyzi a tippet. Találat után kiírja a tippek számát és a lövések koordinátáit, hogy a játékos ellenőrizni tudja magát!
- 129.** Készíts LOTTÓ játékot!
- Játékszabály: A gép 1 és 90 közötti egész számokat generál. Ez az öt darab szám a nyerő. A játékos begépel szintén öt darab számot. Ahány egyező szám van a két számsor között, annyi a találat.
- 130.** Írj programot, amely egy tetszőleges tízes számrendszerbeli számot átír kettes számrendszerbeli számmá!
- 131.** Általánosítsuk az előző feladatot. Írj olyan programot, amely egy tetszőleges tízes számrendszerbeli számot átír tetszőleges számrendszerbeli számmá!
- 132.** Írj programot, amely átír egy tetszőleges számrendszerbeli számot tízes számrendszerbeli számmá!
- 133.** Készíts olyan programot, amely akárhány darab számot növekvő sorrendbe rendez!



134. Készíts olyan programot, amely akárhány nevet névsorba rendez!
135. A könyvtárakban a cédulák vagy az írók neve, vagy a könyvcímek alapján rendezettek. Írj olyan programot, amely lehetővé teszi, hogy az író neve alapján kérjük az általa írt könyvet, ill. a könyv címe alapján az íróját. Természetesen a program csak az általad nyilvántartott könyvek alapján működik majd.
136. Az ismerősök telefonszámát és címét általában kis noteszben szokták rögzíteni.  
Készíts olyan programot, amely a rögzített adatok alapján lehetővé teszi, hogy a név alapján kérjük a lakáscímet és a telefonszámot, a telefonszám alapján a nevet és a címet, a cím alapján a nevet és a telefonszámot. Feltételezzük, hogy csak egy név, egy telefonszám és egy lakcím tartozik össze:

### 3.9 READ, DATA, RESTORE

137. Írj programot, amely az első húsz természetes számot DATA-ban tárolja!  
a) Olvasd be ebből az első tíz számot, majd írasd ki azokat!  
b) Tölts fel egy 30 elemű tömböt ezekből az értékekből!  
c) Tölts fel egy 15 elemű tömböt ezekből az értékekből úgy, hogy az első öt szám háromszor szerepeljen benne!
138. Írj programot, amely bármely hónapról megmondja, hogy hány napos. A szökőévet nem vesszük figyelembe!
139. Az osztályokban félévenként értékelni szokták a tanulók eredményeit. A tanulmányi verseny értékeléséhez szükség van a tanulók tanulmányi átlagára. Írj programot, amely kéri a tanuló félévi jegyeit tantárgyanként, és kiszámítja a tanulmányi átlagot!
140. Írj programot, amely megállapítja, hogy egy adott pénzösszeget milyen címletű pénzekből a legcélszerűbb kifizetni!
141. Feladatként már az előzőekben is szerepelt egy telefonregiszter készítése. Hátránya volt, hogy a begépelte adatok a gép kikapcsolásával elvesztek. Írd meg úgy a telefonregisztert, hogy az adatokat DATA-ban tárolva, onnan kiolvassva használjuk!
142. Írd át úgy a „Könyvkatalógus” programot, hogy az adatokat itt is DATA-ban tárold!
143. Ha a feladatgyűjteményben szereplő feladatokat még kidolgoztad, meglehetősen sok programmal rendelkezel már ahhoz, hogy elkészítsd azok katalógusát. Írj programot, amely rögzíti, hogy az adott program melyik kazetta hányadik fordulataánál található. Tedd lehetővé a kazetta neve, ill. a program címe szerinti lekérdezést is!



### 3.10 GET A\$

144. Készíts olyan programrészletet, amely lehetővé teszi, hogy a program csak a megfelelő billentyű lenyomására induljon!
145. Írj olyan programrészletet, amely lehetővé teszi, hogy a program tetszőleges billentyű lenyomására induljon!
146. Írj programot, amely elkészíti tetszőleges számra a szorzótábla első száz elemét. Készítsd el úgy a programot, hogy a képernyő húszsoronként „lapozható” legyen!

### 3.11 Időmérés számítógéppel (TI és TI\$)

147. Készíts olyan programot, amely a képernyő közepén folyamatosan jelzi a pillanatnyi gépidőt órában, percben, másodpercben!
148. Írj programot, amelynek hatására a képernyő mint stopperóra üzemel!
149. Készíts programot, amelynek hatására a képernyő mint óra működik!
150. A feladatgyűjtemény elején találsz egy programot, amely segít a billentyűzet használatának gyakorlásában. Készíts te is billentyűgyakorló programot, hasonló szabályokkal. A gyakorlásra szánt karaktereket DATA-ban tárolhatod. Mérje a gép a karakter megjelenése és a megfelelő billentyű leütése között eltelt időt is!

### 3.12 Képernyőkezelés, grafika

151. Írj programot, amely a képernyőre véletlenszerűen pontokat helyez! A program az I leütésére induljon és a V leütésére álljon meg!
152. a) Készíts programot, amely a következő koordinátájú pontokat jeleníti meg a képernyőn: (40, 110), (80, 110), (140, 110), (260, 110), (300, 110), (80, 130), (260, 130), (140, 20), (140, 60), (140, 130)!
- b) Mit ábrázolnak ezek a pontok?  
Kösd össze a megfelelőket!
153. Tervezz te is hasonló ábrákat!
154. Elemezd a két programot! Hogyan működnek, mi a különbség?

a)

```
10 GRAPHIC 1,1
20 FOR J=0 TO 200 STEP 10
30 FOR I=0 TO 320
40 DRAW 1,I,J
50 NEXT I
60 NEXT J
```

b)

```
10 GRAPHIC 1,1
20 FOR I=0 TO 320
30 FOR J=0 TO 200 STEP 10
40 DRAW 1,I,J
50 NEXT J
60 NEXT I
```



155. Írj programot, amely véletlenszerűen vonalakat húz a képernyőre!
156. Készíts olyan programot, amely a képernyőre összefüggő megtört vonalakat rajzol! A rajzolást a bal felső sarokban kezdje a gép, és a vonalat véletlenszerűen húzza!
157. Rajzolj a képernyőre egy 40 egység oldalú négyzetet!
158. Színezd ezt a négyzetet feketére!
159. Színezd a hátteret fehérre, a papírt feketére, és írasd tele a képernyőt piros csillagokkal!
160. a) Rajzolj egyenlő szárú háromszöget a képernyőre!  
b) Színezd a háromszöget feketére!
161. Rajzoltass a géppel egy 40 egység sugarú kört! Rajzold meg a vízszintes és a függőleges átmérőt is!
162. Rajzolj a képernyőre öt koncentrikus kört! Rajzold meg a közös középpontot is!
163. a) Rajzolj a CIRCLE utasítás segítségével szabályos hatszöget!  
b) Rajzold meg a köré írható kört is!
164. Rajzolj téglalapot! Húzd meg a két átlóját!
165. Rajzolj a képernyőre egy térbeli kockát!
166. Rajzold meg a képernyőre lakásokok alaprajzát! Írd rá, hogy melyik helyiség milyen célt szolgál!

### 3.13 Zene

167. Szólaltasd meg a normál A hangot 2 másodperces időtartamban!
168. Írj programot, amely hatására a gép 1-1 másodpercig megszólaltatja az alsó és a felső C hangot! A program a V billentyű lenyomásával álljon meg!
169. Szólaltasd meg a számítógép teljes hangterjedelmét úgy, hogy minden hang fél-fél másodpercig szóljon!
170. Szólaltassa meg az egyik hanggenerátor a normál A hangot! A második hanggenerátor szóljon egy oktávval feljebb és másodpercenként kétszer kapcsoljon ki és be!
171. Rajzoljon a gép véletlenszerűen vízszintes és függőleges vonalakat a képernyőre! Ha a gép vízszintes vonalat húz, adjon a gép magas, ha függőleges vonalat húz, akkor mély hangot!

172. Írj olyan programot, amelynek hatására a számítógép mint elektromos orgona használható!
173. Írj programot, amelynek hatására a gép eljátssza a „Pál, Kata, Péter jó reggelt” című dalt!

### 3.14 Karakterfüzér függvények

174. Írass ki a géppel 100 betűből álló véletlenszerű karaktersort!
175. Írassd ki a géppel a következő szöveget:  
„A tudás gyökere keserű, gyümölcse édes”  
(Apáczai)
176. Jelenítsd meg a képernyőn a gép teljes karakterkészletét!
177. Írj programot, amely elkészíti egy tetszőleges szöveg karakterkódját!
178. Írj programot, amely a begépelte név alapján megadja az illető monogramját!
179. Írja ki a gép egy begépelte szóról, hogy hány betűt tartalmaz!
180. Készítsd programot, amely megállapítja egy adott szövegről, hogy hány karaktert tartalmaz!
181. Írj programot, amely a begépelte ige-ről megállapítja, hogy ikes ige-e!
182. Írj programot, amely kiírja, hogy a begépelte szó hány „e” betűt tartalmaz!
183. Írj programot, amely megállapítja, hogy a begépelte mondat hány ly-t tartalmaz!
184. Készíts programot, amely egy begépelte mondatot betűnként lassítva ír ki!
185. Írj programot, amely megállapítja, hogy egy véletlenszerűen megadott betűhalmazban hány-szor szerepel egy adott szó!
186. Javítsd ki úgy a monogram-készítő programot, hogy az akkor is helyesen készítse el a monogramot, ha a név kettős betűvel kezdődik!
187. Írj programot, amely sztringváltzóban helyezi el az első húsz természetes számot!
188. Rajzoltass a géppel egy borítékot!  
Írd meg úgy a programot, hogy a címzés tetszés szerinti legyen!

### 3.15 Logikai műveletek (AND, OR, NOT) és ON ... GOTO

189. Írass ki 50 db 20 és 200 közötti egész számot!  
a) Írj programot, amely ezek közül kiválasztja a párosokat!



- b) Írj programot, amely kiválasztja a 110-nél nagyobbakat!
- c) Írj programot, amely kiválasztja a páratlan és öttel osztható számokat!
- d) Írj programot, amely kiválasztja a 80-nál nagyobbakat vagy a négyel oszthatókat!
- e) Írj programot, amely kiválasztja a 100-nál nagyobb és 150-nél nem nagyobb számokat!
- f) Írj programot, amely kiválasztja azokat a számokat, amelyekre nem igaz, hogy nagyobbak 90-nél és páratlanok!

190. Fűzd össze egy programmá az 189. feladat programjait!

191. Írj programot, amely összegyűjti, hogy az őrőd tagjai milyen szakkörökre járnak! Egy tanuló maximum három szakkörre járhat.
- a) Írj programot, amely megnevezi azokat a pajtásokat, akikből célszerű összeállítani egy sportversenyre induló csapatot!
  - b) Írj programot, amely megnevezi a KRESZ vetélkedőn indítható csapattagokat!
  - c) Írj programot, amely megnevezi a népdaléneklési versenyen indítható pajtásokat!
  - d) Írj programot, amely javaslatot tesz arra, kiket érdemes indítani egy egyéni művészeti vetélkedőn!
  - e) Írj programot, amely kijelöli azokat a pajtásokat, akik indíthatók egy természettudományos csapatversenyen!
  - f) Írj programot, amely megnevezi a matematika-versenyen indítható pajtásokat!

192. Fűzd össze egy programmá az előző feladat programjait!

### 3.16 A sornyomtató használata

193. Nyomtass ki tetszőleges programot sornyomtatón!

194. Nyomtass tele 5 sort A betűkkel!

195. Nyomtass felváltva + és \* sorokat egymás után négyszer úgy, hogy a különböző sorokat üres sorok válasszák el!

196. Oldd meg úgy az előző feladatot, hogy a gép dupla szélességű karakter módban működjön!

197. Írasd ki egymás alá hatszor: COMMODORE 16! Minden páratlan sor normál, minden páros sor dupla szélességű karakterekkel íródjon!

198. Írasd ki sornyomtatóra a gép teljes karakterkészletét és a hozzájuk tartozó CHR kódokat!

199. Készíts tetszőleges számra szorzótáblát, és írasd ki az első húsz elemét a következő módon:

## 9 SZORZOTABLAJA

SZORZANDO

SZORZAT

1	9
2	18
3	27
4	36
5	45
6	54
7	63
8	72
9	81
10	90
11	99
12	108
13	117
14	126
15	135
16	144
17	153
18	162
19	171
20	180

200. Írj programot, amely sornyomatón megjeleníti az őröd tagjainak nevét, címét, telefonszámát a következő módon:

### MOKA MIKI ÖRS TAGJA INAK ADATAI

NEV	LAKC IM	TEL. SZAM
BALOGH ESZTER	BP. V. VACI U. 21	372456
BODA CSABA	BP. V. HARI S KOZ S	381442
CSOKA ILONA	BP. V. KIGYO U. 5	370350

201. Készíts pénztárgépet, amely a begépel t pénzösszegeket és a végösszeget sornyomatón, a tizedespontra illesztve jeleníti meg!

#### 3.17 A kazettás egység

202. Írj programot, amely 100 darab egy és kilencven közé eső véletlenszámot kazettán rögzít!



- 203.** Írj programot, amely az előző feladatban elkészített file-ból tetszőleges ( $X < 100$ ) elemet olvas be a memóriába, és jelenít meg a képernyőn!
- 204.** Írj programot, amely az őröd névsorát kazettán tárolja! Az őr általában háromnál több, de tizenötnél kevesebb tagot számlál.
- 205.** Kétszíts programot, amely az előző feladatban elkészített file elemeit visszaolvassa a memóriába, és a kapott adatokat képernyőn jeleníti meg!
- 206.** Készíts programot, amely az osztályod tanulóinak nevét és tanulmányi átlagát magnószalagon rögzíti!
- 207.** Készítsd el azt a programot, amely az előző program által rögzített adatokat beolvassa a memóriába, és azokat képernyőn is megjeleníti!
- 208.** A 206. feladatban létrehozott file elemeiben gyakran év közben, de félévenként biztosan beáll változás. Írj olyan programot, amely lehetővé teszi a file folyamatos gondozását, hogy az adatok naprakész információval szolgáljanak! Előforduló változások: új tanuló érkezik, tanuló távozik, félévkor, ill. évvégén a tanulmányi átlag változik!

### 3.18 Mit csinál a gép? Elemezd a programokat!

**209.**

```

10 INPUT A,B
20 IF A<B THEN GOTO 60
30 X=A
40 A=B
50 B=X
60 PRINT A,B

```

**210.**

```

10 INPUT A,B,C
20 IF A=B OR B>C THEN GOTO 60
30 X=C
40 C=B
50 A=X
60 PRINT "A=";A,"B=";B,"C=";C
70 END

```

211. Mit ír ki a gép?

a)

```
10 A=9
20 B=7
30 C=-1
40 A=B+C
50 B=A+C
60 C=A+B
70 PRINT A/B/C
```

c)

```
10 A=4
20 B=6
30 C=2
40 A=A+B-C
50 B=B+C-A
60 C=C+A-B
70 PRINT A/B/C
```

e)

```
10 A=3
20 B=9
30 C=4
40 A=-A
50 B=B+A
60 C=A+B+C
70 PRINT A/B/C
```

g)

```
10 A=3
20 B=4
30 C=A*B
40 D=A+B
50 C=C*D
60 PRINT C
```

b)

```
10 A=2
20 B=5
30 C=8
40 X=A
50 B=X
60 C=A+B
70 PRINT A/B/C
```

d)

```
10 A=5
20 B=6
30 C=7
40 A=B
50 B=C
60 C=A
70 PRINT A/B/C
```

f)

```
10 A=-1
20 B=-4
30 C=-2
40 D=(A-B)*C
50 D=D-H
60 D=D+B
70 D=D/C
80 PRINT D
```

h)

```
10 A=5
20 B=9
30 B=B*B
40 B=B/A
50 A=A+B
60 PRINT A/B
```

i)

```
10 A=2
20 B=3
30 C=4
40 A=A+B
50 B=B-C
60 C=C*A
70 PRINT A/B/C
```



212.

```
10 PRINT "Q"  
20 FOR I=1 TO 40  
30 PRINT TAB(I); "+"  
40 NEXT I  
50 GOTO 20
```

213.

```
10 PRINT "Q"  
20 FOR K=1 TO 5  
30 READ A$(K)  
40 NEXT K  
50 FOR K=1 TO 5  
60 FOR J=1 TO 40  
70 PRINT A$(K);  
80 NEXT J  
90 NEXT K  
100 GOTO 50  
110 DATA *,+,/,?,E
```

214.

```
10 INPUT A$  
20 PRINT "Q"  
30 PRINT A$;  
40 GOTO 30
```

215.

```
10 PRINT "Q"  
30 FOR J=1 TO 10  
40 FOR K=1 TO 10  
50 PRINT "*" ;  
60 NEXT K  
70 NEXT J  
90 END
```

216.

```
10 PRINT "Q"  
20 FOR I=1 TO 100  
30 A=(INT(RND(1)*100)+1)*2  
40 PRINT A;  
50 NEXT I  
60 END
```

217.

```
10 PRINT "Q"  
20 FOR I=1 TO 100  
30 A=(INT(RND(1)*101)*2)+1  
40 PRINT A;  
50 NEXT I  
60 END
```

218.

```
10 INPUT X
20 FOR I=1 TO 20
30 H=INT(I/X)
40 M=I-H*X
50 PRINT I,M
60 NEXT I
70 END
```

219.

```
10 INPUT X
20 IF X=INT(X/8)*8 THEN PRINT "**":GOTO 40
30 PRINT "*"
40 END
```

220.

```
10 INPUT A,B
20 IF A>0 AND B<0 THEN GOTO 50
30 PRINT "□"
40 PRINT TAB(20); "XXXXXXXXXX**"
50 END
```

221.

```
10 GET A#
20 PRINT "□"
30 IF A#="*" OR A#="+" THEN GOTO 50
40 GOTO 10
50 FOR I=1 TO 25
60 FOR J=1 TO 40
70 PRINT A#;
80 NEXT J
90 NEXT I
100 END
```

222.

```
10 A=INT(RND(1)*10)+1
20 IF A>5 AND A<>INT(A/3)*3 THEN PRINT "ALMA":GOTO 40
30 PRINT "KORTE"
40 END
```

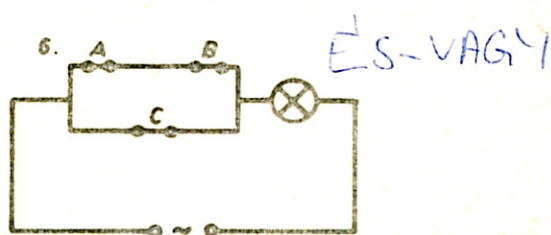
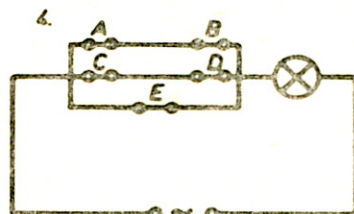
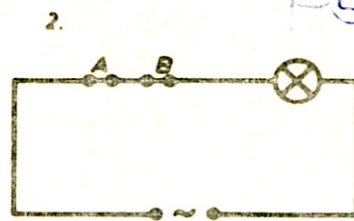
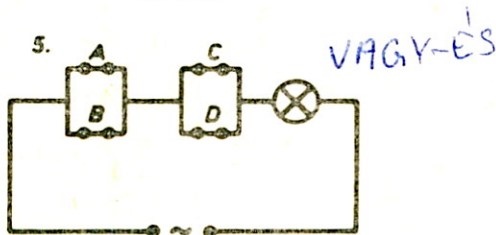
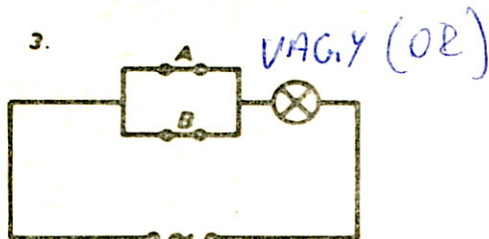
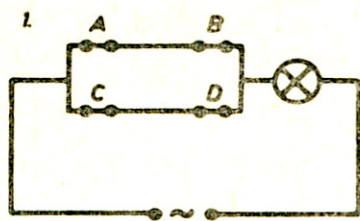


223. Melyik programhoz melyik logikai áramkör tartozik, ha Z jelentése, hogy a kapcsolók zártak, az áram akadálytalanul áramlik?

Programok:

- a)
- ```
10 INPUT A$,B$
20 IF A$="Z" AND B$="Z" THEN PRINT"EG":GOTO 40
30 PRINT"NEM EG"
40 END
```
- b)
- ```
10 INPUT A$,B$
20 IF A$="Z" OR B$="Z" THEN PRINT"EG":GOTO 40
30 PRINT"NEM EG"
40 END
```
- c)
- ```
10 INPUT A$,B$,C$
20 IF (A$="Z" AND B$="Z") OR C$="Z" THEN PRINT"EG":
   GOTO 40
30 PRINT"NEM EG"
40 END
```
- d)
- ```
10 INPUT A$,B$
20 INPUT C$,D$
30 IF (A$="Z" OR B$="Z") AND (C$="Z" OR D$="Z")
   THEN PRINT"EG":GOTO 50
40 PRINT"NEM EG"
50 END
```
- e)
- ```
10 INPUT A$,B$
20 INPUT C$,D$
30 IF (A$="Z" AND B$="Z") OR (C$="Z" AND D$="Z")
   THEN PRINT"EG":GOTO 50
40 PRINT"NEM EG"
50 END
```
- f)
- ```
10 INPUT A$,B$
20 INPUT C$,D$,E$
30 IF (A$="Z" AND B$="Z") OR (C$="Z" AND D$="Z")
   OR E$="Z" THEN PRINT"EG":GOTO 50
40 PRINT"NEM EG"
50 END
```

Logikai áramkörök:



224.

```

10 PRINT "□"
20 FOR I=1 TO 9
30 READ A
40 X=X+A
50 PRINT X
60 NEXT I
70 DATA 1,2,3,4,5,6,7,8,9
    
```

225.

```

10 PRINT "□"
20 FOR I=1 TO 9
30 READ A$
40 X$=X$+A$
50 PRINT X$
60 NEXT I
70 DATA 1,2,3,4,5,6,7,8,9
    
```

226.

```

10 X=1
20 A$="*****"
30 B$=" "
40 FOR I=0 TO 25
50 FOR J=0 TO 20
60 PRINT LEFT$(A$,X);B$;
70 NEXT J
80 NEXT I
90 X=X+1
100 IF X<=10 THEN GOTO 40
110 END
    
```

227.

```

10 PRINT "□"
20 FOR I=1 TO 9
30 H$=STR$(I)
40 B$=B$+H$
50 PRINT B$
60 NEXT I
    
```



228.

```
10 PRINT"□"  
20 TI$="000000"  
30 FOR I=1 TO 100 : NEXT I  
40 PRINT TI/60  
50 END  
60 NEXT I
```

229.

```
10 PRINT"□"  
20 A=INT(RND(1)*62)+33  
30 PRINT A  
40 TI$="000000"  
50 GET A$  
60 IF A$=CHR$(A) THEN PRINT TI/60;"S":GOTO 80  
70 GOTO 50  
80 END
```

230.

```
10 PRINT"□"  
20 FOR J=1 TO 16  
30 FOR K=1 TO 7  
40 COLOR O,J,K  
50 FOR L= 1 TO 200 : NEXT L  
60 NEXT K  
70 NEXT J  
80 END
```

231.

```
10 GRAPHIC 1,1  
20 SCNCLR  
30 FOR J=0 TO 200  
40 FOR I=0 TO 320 STEP 10  
50 DRAW 1,I,J  
60 NEXT I  
70 NEXT J  
80 END
```

232.

```
10 GRAPHIC 1,1
20 SCNCLR
30 FOR J=0 TO 320 STEP 10
40 FOR I=0 TO 200
50 DRAW 1,J,I
60 NEXT I
70 NEXT J
80 END
```

233.

```
10 PRINT "□"
20 COLOR 0,1,7
30 COLOR 4,2,1
40 COLOR 1,3,7
50 FOR I=1 TO 25
60 FOR J=1 TO 40
70 PRINT "+";
80 NEXT J
90 NEXT I
100 END
```

234.

```
10 GRAPHIC 1,1
20 SCNCLR
30 I=0 : J=0
40 DRAW 1,I,J
50 IF I=200 THEN GOTO 20
60 I=I+1
70 J=J+1
80 GOTO 40
```

235.

```
10 GRAPHIC 1,1
20 SCNCLR
30 FOR I=1 TO 280 STEP 40
40 CIRCLE 1,I,100,40
50 NEXT I
60 END
```

236.

```
10 GRAPHIC 1,1
20 SCNCLR
30 BOX 1,120,60,200,140
40 CIRCLE 1,160,100,40,60
50 CIRCLE 1,160,100,60,40
60 END
```

237.

```
10 GRAPHIC 1,1
20 SCNCLR
30 FOR I=1 TO 5
40 READ X(I),Y(I)
50 NEXT I
60 FOR I=1 TO 5
70 CIRCLE 1,X(I),Y(I),40
80 NEXT I
90 DATA 160,100,120,100,160,60,200,100,160,140
```



### 3.19 Néhány vegyes feladat

- 238.** Készíts programot, amely megállapítja, hogy egy adott lineáris függvénynek milyen a menete!
- 239.** Egy baráti társaság összejövetelt rendez. Már mindent megvásároltak, ami a szendvicsek elkészítéséhez szükséges volt. A maradék pénzen üdítőt kívánnak vásárolni. Írj programot, amely segít kiszámítani, hogy a rendelkezésre álló összegből hány darab adott egységáru termék vásárolható!
- 240.** A gyerekek nagy részének gondot okoz a szorzótábla. Mivel a szorzótábla csak ismétléssel tanulható meg, és ezt sokan unalmasnak tartják, nem is sajátítják el megfelelő módon. Írj olyan programot, amely a tanuló által megadott szorzótáblát segít begyakorolni!
- 241.** Félévente az osztályok között tanulmányi versenyek értékeléséhez szükség van az osztály tárgyakénti tanulmányi átlagára. Írj programot, amely megkönnyíti ezt a fárasztó és időigényes munkát!
- 242.** Írj programot, amely a használó által megadott számkörben segít gyakorolni a négy alapműveletet!
- 243.** A hatodik osztályos matematika könyv foglalkozik már a prímek kikeresésének egy érdekes módszerével, amelyet a feltalálója után Erathosztenész szitája néven emlegetnek. A módszerrel már dolgoztatok. Írj programot, amely tetszőleges számig megkeresi a prímszámokat! Vigyázz, mert az oszthatóság csak a természetes számok körében értelmezett!
- 244.** Készítettünk már olyan programot, amely egy tízes számrendszerbeli számot tetszőleges alapú számrendszerbeli számmá ír át, és megírtuk ennek a fordítottját végző programot is. Készíts most olyan programot, amely tetszőleges számrendszerbeli számot tetszőleges számrendszerbeli számmá ír át!
- 245.** Írtunk már olyan programot, amely Torpedó játékot játszik. Készíts most olyan programot, amely megrajzolja a játékteret is, és a lövéseket a megfelelő koordinátájú négyzet beszínezésével jelzi. Ha a játékos kilőtte a hajót, mutassa meg a gép újra a lövéseket a megfelelő sorrendben!
- 246.** Matematika órán sok időt elvesz, amikor a valószínűségszámításos feladatok során el kell végezni a kísérleteket, például száz kockadobást. Írj programot, amely szimulálja a dobásokat, és az egyes értékek előfordulását grafikonon, oszlopdia grammal ábrázolja!
- 247.** A háborúban igen nagy mennyiségű kódolt üzenetet kellett megfejteni a kémelhárítóknak. Ennek a hatalmas adatmennyiségnek a feldolgozása sok gondot okozott. Ez a probléma is elősegítette a számítástechnika kialakulását, fejlődését. Írj olyan programot, amely lehetővé teszi szövegek titkosítását, titkosítás szöveggé való alakítását! A kódot Te választod meg!
- 248.** Nagyobb cukrászdákban már meglehetősen elterjedt a számlázógépek hasz-



nálata. Az étel kódjának begépelése és az asztalszám megadása után a géptől kérni lehet a számlát. Írj programot, amely elvégzi ennek a gépnek a feladatát, és ilyen formátumú számlát ad:

## 1 . ASZTAL SZÁMLAJA

MALNA	5.30
NARANCS	5.50
PUNCSTORTA	4.50
MANGO	5.50
TONIK	5.50
ÖSSZESEN	26.30

## 5 . ASZTAL SZÁMLAJA

DOBOSTORTA	5.20
NARANCS	5.50
CITROMTORTA	4.50
ÖSSZESEN	15.20

- 249.** A különböző grafikus utasítások megadásával szép ábrák rajzolhatók a képernyőre. Ezek azonban mind előre megtervezett ábrák. Írj olyan programot, amely lehetővé teszi, hogy a billentyűk segítségével tetszőleges ábrákat rajzolhassunk a képernyőre! Tegyük lehetővé, hogy az elrontott részeket törölje a használó! Vonalat balra a B; jobbra a J; felfelé az F; lefelé az L billentyűvel rajzolhassunk!
- 250.** Magyar órán gyakran kell egy verset vagy szövegrészt hangulata alapján elemezni. Egy szöveg hangulatát nagy mértékben befolyásolhatja a benne szereplő betűk százalékos aránya. Írj olyan programot, amely egy tetszőleges szövegben szereplő betűket ebből a szempontból vizsgálja!
- 251.** Biztosan ismeri mindenki az „akasztófa” játékot. Írj olyan programot, amely az ismert játékszabályok alapján egy ötbetűs szó kitalálását várja el a játékostól!
- 252.** Már kis korotok óta foglalkoztak halmazok közötti kapcsolatokkal. Írj programot, amely két halmaz közt fennálló legismertebb halmazkapcsolatot szemléltet! Ezek legyenek: metszet, különbség, komplementer és maga a két halmaz.
- 253.** Az összeadások, kivonások gyakorlására gyakran oldottatok meg bővös-négyzet feladatokat. Írj olyan programot, amely megrajzol egy háromszor hármas bővös-négyzetet, megadja az egyik átló elemeit. Az eltalált további elemeket a gép írja a négyzet megfelelő helyére!



- 254.** Készíts programot, amely egy tantárgy, pl. fizika egy témájának gyakorlását teszi lehetővé tesztkérdések alapján! A kérdéseket és a helyes válaszokat célszerű file-ban rögzíteni.
- 255.** Egy török szultán a börtönben 100 foglyot őriztetett. Minden cellán kétállású zár van, amely egy fordításra nyílik, másik fordításra záródik. Mivel a rabok élelmezése sokba kerül, a szultán úgy döntött, hogy a rabok egy részét elengedi. Így elküldte az első szolgáját, hogy fordítson minden záron egyet; a másodikat, hogy fordítson minden második; a harmadikat, hogy fordítson minden harmadik...; a századikat, hogy fordítson minden századik záron egyet. Amelyik cella ajtaja az utolsó fordítás után nyitva marad, az a fogoly elmehet. Írj programot, amely a mese alapján tippet kér a játékosától, hogy ha ő lenne a rab, melyik cellát választaná, majd a gép írja ki, hogy az illető kiszabadulna-e vagy sem!

## II. MEGOLDÁSOK

### 1. A BILLENTYŰZET HASZNÁLATA

#### 1.1 Példák

1. Minden programsor után RETURN gombot kell lenyomni.
2. Képernyőtörlés SHIFT+HOME/CLEAR együttes lenyomásával történik.
3. A bezárás a SHIFT+INST/DEL lenyomásával történik.
4. Jobbról balra törlés az INST/DEL lenyomásával történik.
5. Balról jobbra törölni a SPACE billentyűvel lehet.
6. Kurzormozgató billentyűkön nyilak jelzik a megfelelő irányt.
7. Inverz karakter a CTRL (ConTRol) gomb lenyomásával jön létre.
8. Kisbetű a C+=SHIFT együttes lenyomásával írható.
9. Megfejtés: Ezt a szöveget a SHIFT billentyű lenyomásával készítettem. Megfejtteni a C= gomb lenyomásával lehet.

### 2. SZÁMOK ÉS MŰVELETEK

#### 2.1 Példák

16. a) A gép nem ír ki semmit. Hiányzik a kiíró utasítás.  
b)–c) Megjelenik az eredmény: 11. Tehát a PRINT utasítás helyettesíthető a ? leütésével!

#### 17. Indokolás:

1. Zárójel nélkül más a műveletvégzés sorrendje, így az eredmény is.
2. Zárójel nélkül is ugyanaz a műveletvégzés sorrendje, így az eredmény is. Ezért természetesen a zárójel helyének megváltoztatása nem jelenti az eredmény változását.
3. a/1. 1. a zárójelben levő művelet  $3+2$   
2. az összeg szorzása  $33*5$   
a/2. 1. a szorzás  $2*5$   
2. az összeadás  $31+10$   
b/1. 1. a zárójelben levő művelet  $3+5$   
2. az összeg szorzása  $2*8$   
b/2. 1. a szorzás  $2*3$   
2. az összeadás  $6+5$   
c/1. 1. a zárójelben levő művelet  $2+4$   
2. az összeadás  $6+6$   
c/2. 1. a zárójelben levő művelet  $4+6$   
2. az összeadás  $2+10$   
d/1. 1. a zárójelben levő művelet  $3*5$   
2. a szorzás  $15*2$   
d/2. 1. a zárójelben levő művelet  $5*2$   
2. a szorzás  $3*10$

(A műveletek prioritása (sorrendje, erőssége):

1. zárójelben levő bármilyen művelet,
2. hatványozás-gyökvonás,



3. szorzás-osztás,
4. összeadás-kivonás.

Azonos erősségű műveletek esetén balról jobbra végzi a gép a műveleteket.)

- 18. a)**
1. zárójelben levő összeadás  $3+2$
  2. az összeg szorzása  $(3+3)*15$  vagy az osztás  $8/9$
  3. a kivonás  $(3+2)*15-8/9$
- b)**
1. hatványozás  $3 \uparrow 2$
  2. a zárójelben levő összeadás  $5+6+9$
  3. az összeg szorzása  $(5+6+9)*12$
  4. vagy az összeadás  $3 \uparrow 2+(5+6+9)*12$  vagy a kivonás  $(5+6+9)*12-8$ . (Itt a balról jobbra szabály dönt.)
- 19. a)**
1. a zárójelben levő művelet
  2. a szorzás
  3. a kivonás
- b)**
- 1., 2. a két zárójelben levő művelet balról jobbra
  3. a szorzás
- c)**
1. a zárójelben levő művelet
  2. az osztás
- d)**
1. a szorzás
  2. a kivonás
  3. az összeadás
- e)**
1. az osztás
  2. a kivonás
  3. az összeadás
- f)**
1. a hatványozás
  2. az osztás
  3. a kivonás

- 20. a)** PRINT  $(11+4.9)*2$                       PRINT  $11*4.9$   
**b)** PRINT  $3+4+5$                               PRINT  $3*4/2$   
**c)** PRINT  $2*12.4 * \pi$                          PRINT  $12.4 \uparrow 2 * \pi$

**22.** Maximum 85 karakter írható egy PRINT után az idézőjelek közé.

## 2.2 A változók

Változótípusok

Valós

Jelentésük

$10^{-38} \dots 10^{+38}$  közé eső tetszőleges valós szám

Jelölése: A B C A1 B2 C3 AB CD GJ

Egész

A begépelt számból csak az egészeket tartja meg. Ha a szám nem egész, úgy azt csonkolja.

Jelölése: A<sup>0</sup>/<sub>0</sub> A1<sup>0</sup>/<sub>0</sub> AB<sup>0</sup>/<sub>0</sub>





### 3.1 A PRINT utasítás

#### Szintaxis

```
10 PRINT A
20 PRINT A$
30 PRINT A%

40 PRINT "A"
50 PRINT "A$"
60 PRINT "A%"

70 PRINT " "
80 PRINT " "
90 PRINT " "
100 PRINT " "
110 PRINT " "
120 PRINT " "

130 PRINT
140 PRINT A;B
150 PRINT A,B
160 PRINT TAB(X);A
```

#### Jelentés

A PRINT után írt változó értéke jelenik meg

A PRINT után idézőjelben levő szöveg jelenik meg

Képernyőtörlés, kurzor a bal felső sarokba kerül

Kurzor-  
mozgatás

Jobbra egy egységgel

Balra egy egységgel

Lefelé egy egységgel

Felfelé egy egységgel

Bal felső sarokba kerül, törlés nélkül

Egy üres sort húz

A és B értékét tömörítve írja ki

A és B értékét zónázva írja ki

A sor X. helyére írja A értékét

#### 31. Vegyük észre:

- Az első lépés a képernyő törlése.
- Változóként két füzér (sztring) típusút kell alkalmazni.
- Az első és a harmadik sor zónázott forma, ezért a változók között vesszőt, a második és a negyedik sorban a tömörített forma miatt pedig pontosvesszőt lehet alkalmazni.

A program:

```
10 PRINT " "           40 PRINT A$,B$       70 PRINT A$;B$
20 A$="BUDA"          50 PRINT A$;B$       80 END
30 B$="PEST"          60 PRINT A$,B$
```

#### 32. Vegyük észre:

- A változóknak valós típusúakat kell választani.
- Vigyáznunk kell, hogy definiáló egyenlőségekről van szó, így az csak balról jobbra olvasva igaz.

A program:

```
10 PRINT " "           70 F=A/B
20 A=50                80 PRINT C
30 B=15.5              90 PRINT D
40 C=A+B              100 PRINT E
50 D=F-B              110 PRINT F
60 E=A*B              120 END
```





**36. Vegyük észre:**

- A c) feladat megoldható a kurzor mozgatásával és üres sorok íratásával is.

A programok:

**a)**

```
10 PRINT "□"  
20 PRINT "ALMA"  
30 PRINT " ALMA"  
40 PRINT "  ALMA"  
50 END
```

**b)**

```
10 PRINT "□"  
20 PRINT "ALMA"  
30 PRINT "  ALMA"  
40 PRINT "    ALMA"  
50 END
```

**c)**

```
10 PRINT "□"  
20 PRINT "ALMA"  
30 PRINT "  ALMA"  
40 PRINT "    ALMA"  
50 END
```

**d) És egy megoldás  
a PRINT utasítás  
segítségével**

```
10 PRINT "□"  
20 PRINT "ALMA"  
30 PRINT  
40 PRINT "  ALMA"  
50 PRINT  
60 PRINT "    ALMA"  
70 END
```

**37. Vegyük észre:**

- A PRINT TAB(X) parancs az utolsó kurzorhelytől számítja az X. helyet.
- Ha a PRINT utasítás után nincs pontosvessző vagy vessző, a következő PRINT utasítás mindig új sort jelent. Így ilyenkor a PRINT TAB(X) utasítás a sor elejéről elkezdi a számolást.

A programok:

**a)**

```
10 PRINT "□"  
20 PRINT "ALMA"  
40 PRINT TAB(1); "ALMA"  
60 PRINT TAB(2); "ALMA"  
70 END
```

**b)**

```
10 PRINT "□"  
20 PRINT "ALMA"  
30 PRINT  
40 PRINT TAB(2); "ALMA"  
50 PRINT  
60 PRINT TAB(4); "ALMA"  
70 END
```

**c)**

```
10 PRINT "□"  
20 PRINT "ALMA"  
40 PRINT TAB(2); "ALMA"  
60 PRINT TAB(4); "ALMA"  
70 END
```

**38. Vegyük észre:**

- A megtervezett ábrát a PRINT TAB( ), ill. a kurzormozgatás segítségével soronként lehet programozni.

A házrajzoló program:

```
10 PRINT "□"
15 C=4.5
20 PRINT "□"
25 B=A+C
26 B%=B
30 PRINT TAB(21); "++"
40 PRINT TAB(21); "++"
50 PRINTTAB(11); "++++++++++"
60 PRINT TAB(10); "+          +"
70 PRINT TAB(9); "+          +"
80 PRINT TAB(8); "+          +"
90 PRINT TAB(7); "+          +"
100 PRINT TAB(6); "+          +"
110 PRINT TAB(5); "++++++++++"
120 PRINT TAB(7); "+          +"
130 PRINT TAB(7); "+          +"
140 PRINT TAB(7); "+          +"
150 PRINT TAB(7); "+      ++++++      +++      +"
160 PRINT TAB(7); "+      +      +      +      +"
170 PRINT TAB(7); "+      +      +      +      +"
180 PRINT TAB(7); "+      ++++++      +      +"
190 PRINT TAB(7); "+          +      +      +"
200 PRINT TAB(7); "+          +      +      +"
210 PRINT TAB(7); "+          +      +      +"
220 PRINT "++++++++++"

```

**39. Vegyük észre:**

- A TAB( ) segítségével szép táblázat készíthető.
- Ha a PRINT parancs után vagy ; szerepel, a tabulálás ugyanabban a sorban folytatódik.

A program:

```
10 PRINT "□"
20 PRINT "A HARMAS SZORZOTABLA ELEMEI"
30 PRINT
40 PRINT "SORSZAM", "SZORZAT"
50 A=1
60 B=A*3
70 PRINT TAB(3); A;
80 PRINT TAB(12); B
90 A=2

```





42. Mivel az  $A=B$  és a  $B=A$  egyenlőségek definiáló egyenlőségek, így azoknál csak a bal oldalon álló változó egyenlő a jobb oldalon állóval, visszafelé ez nem igaz.

43. a)  $C=0$                       b)  $C=16$                       c)  $C=22$                       d)  $C=-9$

44. Vegyük észre:

- A rendezés előtti sorrendet még a rendezés előtt ki kell íratni, hiszen a mozgatással a változók értéke folyamatosan változik.
- Fel kell vennünk egy új változót, hogy a csere létrejöhessen.

A program:

```
10 PRINT "▼"  
20 PRINT TAB(11); "VÁLTOZOK CSEREJE▼"  
30 A=3  
40 B=4  
50 C=5  
60 PRINT "CSERE ELŐTT: "  
70 PRINT "A=";A,"B=";B,"C=";C  
80 X=A  
90 A=B  
100 B=C  
110 C=X  
120 PRINT "CSERE UTÁN: "  
130 PRINT "A=";A,"B=";B,"C=";C  
140 END
```

45. Vegyük észre:

- Az előző feladattól annyiban tér el, hogy itt füzér változókkal kell dolgoznunk, így az újonnan bevezetett változó is füzér lesz.

A program:

```
10 PRINT "▼"  
20 PRINT TAB(6); "HAROM TÁRGY CSEREJE"  
30 PRINT  
40 A$="ZOKNI"  
50 B$="KESZTYU"  
60 C$="ZSEBKENDO"  
70 PRINT "CSERE ELŐTT: "  
80 PRINT "1. FIOK",A$  
90 PRINT "2. FIOK",B$  
100 PRINT "3. FIOK",C$  
110 X$=A$  
120 A$=C$  
130 C$=B$  
140 B$=X$  
150 PRINT
```



```

160 PRINT"CSERE UTAN:"
170 PRINT"1. FIOK",A$
180 PRINT"2. FIOK",B$
190 PRINT"3. FIOK",C$
200 END

```

### 3.3 Az INPUT utasítás

#### Szintaxis

10. sor 10 INPUT A

20. sor 20 INPUT A\$

30. sor 30 INPUT A%

40. sor 40 INPUT"SZOVEG";A

50. sor 50 INPUT"SZOVEG";A\$

60. sor 60 INPUT"SZOVEG";A%

#### Tartalom

Begépett valós számot. A változónévvel a memóriába visz.

Begépett füzért A\$ változónévvel a memóriába visz.

Begépett számot egész típusúként viszi a memóriába, A% változónévvel.

Az INPUT utasításban idézőjelben szereplő szöveg a képernyőn megjelenik. A memóriába kerülő értéket közvetlenül ez után kéri.

#### 46. Vegyük észre:

- Mivel két tetszőleges számról van szó, valós típusú változókkal kell dolgozni.

A program:

a)

```

10 PRINT"□"
20 INPUT A
30 INPUT B
40 PRINT A,B
50 END

```

b)

```

10 PRINT"□"
20 INPUT A
30 INPUT B
40 PRINT A
50 PRINT B
60 END

```

c)

```

10 PRINT"□"
20 INPUT A
30 INPUT B
40 PRINT A;B
50 END

```

47.

```

10 PRINT"□"
20 INPUT A
30 INPUT B
40 C=A+B
50 PRINT C
60 END

```

48.

```

10 PRINT"□"
20 INPUT A
30 B=A+2
40 PRINT B
50 END

```

49. Vegyük észre:

- A szöveg kiíratására több lehetőség van:
  - a) A számolás után a hányadossal együtt.
  - b) Az adatok bekérése előtt.
  - c) Az adatok bekérésével egyszerre az INPUT utasítás speciális tulajdonságát kihasználva.

A programok:

a)

```
10 PRINT "□"  
20 INPUT A  
30 INPUT B  
40 C=A/B  
50 PRINT "AZ OSZTANDO=" ; A  
60 PRINT "AZ OSZTO=" ; B  
70 PRINT "A HANYADOS=" ; C  
80 END
```

b)

```
10 PRINT "□"  
20 PRINT "AZ OSZTANDO=" ;  
30 INPUT A  
40 PRINT "AZ OSZTO=" ;  
50 INPUT B  
60 C=A/B  
70 PRINT "A HANYADOS=" ; C  
80 END
```

c)

```
10 PRINT "□"  
20 INPUT "AZ OSZTANDO=" ; A  
30 INPUT "AZ OSZTO=" ; B  
40 C=A/B  
50 PRINT "A HANYADOS=" ; C  
60 END
```

50.

```
10 PRINT "□"  
20 PRINT "NEGY ALAPMŰVELET"  
30 PRINT  
40 INPUT A  
50 INPUT B  
60 C=A+B  
70 D=A-B  
80 E=A*B  
90 F=A/B  
100 PRINT "AZ OSSZEG=" ; C  
110 PRINT "A KÜLÖNBSEG=" ; D  
120 PRINT "A SZORZAT=" ; E  
130 PRINT "A HANYADOS=" ; F  
140 END
```



**51. Vegyük észre:**

- A program indulása előtt közölni kell a használókkal, hogy milyen feladat elvégzésére írtuk. Ezért célszerű címet adni.
- Meg kell magyarázni, hogy milyen adatokat vár a gép.
- Az eredmény elé oda kell írni, hogy mit számolt a gép.

A program:

```
10 PRINT"□"  
20 PRINT"SZAMTANI KOZEP (ATLAG) SZAMITASA"  
30 PRINT  
40 PRINT"AZ ELSO SZAM"  
50 INPUT A  
60 PRINT"A MASODIK SZAM"  
70 INPUT B  
80 C=(A+B)/2  
90 PRINT"A KET SZAM SZAMTANI KOZEPE"=C  
100 END
```

**52. Vegyük észre:**

- Közismert képletek használatakor célszerű a hagyományos jelölésmódot alkalmazni.

A program:

```
10 PRINT"□"  
20 PRINT  
30 INPUT A  
40 INPUT B  
50 K=(A+B)*2  
60 T=A*B  
70 PRINT K  
80 PRINT T  
90 END
```

**53. Vegyük észre:**

- Két állandó szövegrész közé változókat is írhatunk, ha azt az idézőjelben levő szövegrészek közé, tőlük vesszővel vagy pontosvesszővel elválasztva írjuk.

A program:

```
10 PRINT"□"  
20 PRINT"TEGLALAP KERULETE ES TERULETE"  
30 PRINT  
40 PRINT"ELSO OLDAL (CM)= "  
50 INPUT A  
60 PRINT"MASODIK OLDAL (CM)= "  
70 INPUT B  
80 K=(A+B)*2  
90 T=A*B
```

```

100 PRINT "A KERULET" ; K ; "CM"
110 PRINT "A TERULET" ; T ; "CM^2"
120 END

```

#### 54.

```

10 PRINT "□"
20 PRINT "KOR KERULETE ES TERULETE"
30 PRINT
40 PRINT "A KOR SUGARA (CM) " ;
50 INPUT R
60 K=2*R*π
70 T=R^2*π
80 PRINT "A KOR KERULETE" ; K ; "CM"
90 PRINT "A KOR TERULETE" ; T ; "CM^2"
100 END

```

#### 55.

```

10 PRINT "□"
20 PRINT "AZ Y=3*X+4 FV ERTEKE X HELYEN"
30 PRINT
40 PRINT "X ERTEKE " ;
50 INPUT X
60 Y=3*X+4
70 PRINT "A FV ERTEKE AZ X=" ; X ; "HELYEN" ; Y
80 END

```

#### 56.

```

10 PRINT "□"
20 PRINT "AZ Y=-1/5*X+1 FV ERTEKE X HELYEN"
30 PRINT
40 PRINT "X ERTEKE " ;
50 INPUT X
60 Y=-1/5*X+1
70 PRINT "A FV ERTEKE AZ X=" ; X ; "HELYEN" ; Y
80 END

```

#### 57. Vegyük észre:

- Az általános elsőfokú függvény  $Y=A*X+B$  alakú.
- Az X mellett meg kell kérdezni az A és a B értékét is.

A program:

```

10 PRINT "□"
20 PRINT "TETSZOLEGES Y=A*X+B ALAKU FV ERTEKEINEK"
30 PRINT "SZAMITASA TETSZOLEGES X HELYEN"

```



```

40 PRINT "A=" ;
50 INPUT A
60 PRINT "B=" ;
70 INPUT B
80 PRINT "X=" ;
90 INPUT X
100 Y=A*X+B
110 PRINT
120 PRINT "AZ Y =" ; A ; " * X +" ; B ; " EV ERTEKE" ; Y
130 END

```

**58.**

```

10 PRINT "☐"
20 PRINT "TETSZOLEGES SZAM TETSZOLEGES SZAZALAKA"
30 PRINT
40 INPUT "AZ ALAP" ; A
50 INPUT "A SZAZALEKLAB" ; SL
60 SE=A*SL/100
70 PRINT "A SZAM" ; SL ; "SZAZALAKA" ; SE
80 END

```

**59. Vegyük észre:**

– 3 pók 3 nap 3 háló → 1 pók 3 nap 1 háló → 1 pók 1 nap 1/3 háló.

A program:

```

10 PRINT "☐"
20 PRINT TAB(5) ; "3 POK 3 NAP ALATT 3 HALOT SZO"
30 PRINT
40 INPUT "HANY POK" ; P
50 INPUT "HANY NAP ALATT" ; N
60 H=P*N*1/3
70 PRINT
80 PRINT P ; "POK" ; N ; "NAP ALATT" ; H ; "HALOT SZO"
90 END

```

**60. Vegyük észre:**

– Egy könyv 100 oldalból és két borítóból áll. Ez 12 napi rágcsálásnak felel meg. Tehát amennyi a könyvek száma, annyiszor 12 nap!

A program:

```

10 PRINT "☐"
20 PRINT TAB(14) ; "MOLYLEPKE"
30 PRINT
40 INPUT "A KONYVEK SZAMA" ; K
50 N=K*12
60 PRINT
70 PRINT "A MOLY" ; N ; "NAP ALATT RAGJA AT MAGAT"

```

61.

```
10 PRINT"□"  
20 PRINT TAB(14); "SZOVETARAZO"  
30 PRINT  
40 PRINT "AZ EGYSEGAR (FT/M)";  
50 INPUT E  
60 PRINT "A HOSSZUSAG (M)";  
70 INPUT H  
80 A=E*H  
90 PRINT "A VEG SZOVET ARA"; A; "FT"  
100 END
```

62.

```
10 PRINT"□"  
20 PRINT TAB(11); "MASODPERC SZAMLALAS"  
30 PRINT  
40 PRINT "KEREM AZ IDOTARTAMOT"  
50 PRINT "ORA";  
60 INPUT H  
70 PRINT "PREC";  
80 INPUT M  
90 PRINT "MASODPERC";  
100 INPUT S  
110 I=H*3600+M*60+S  
120 PRINT "AZ IDOTARTAM MASODPERCEKBEN"; I  
130 END
```

63.

```
10 PRINT"□"  
20 PRINT TAB(13); "MERTEKATVALTAS"  
30 PRINT  
40 PRINT "A HOSSZUSAG METERBEN";  
50 INPUT H  
60 D=H*10  
70 C=H*100  
80 M=H*1000  
90 K=H/1000  
100 PRINT "A HOSSZUSAG"; D; "DM"  
110 PRINT "A HOSSZUSAG"; C; "CM"  
120 PRINT "A HOSSZUSAG"; M; "MM"  
130 PRINT "A HOSSZUSAG"; K; "KM"  
140 END
```



**64. Vegyük észre:**

- A játékos által begépelte értékkel ugyanazokat a műveleteket kell elvégezni, amelyeket a szabályban megadtunk, csak fordított sorrendben.

A program:

```
10 PRINT " "
20 PRINT TAB(8); "SZAMKITALALO"
30 PRINT
40 PRINT "GONDOLJ EGY SZAMOT! VEGYEL EL BELOLE"
50 PRINT "28-AT, SZOROZD 8-CAL ES ADJ HOZZA 13-AT"
60 PRINT
80 PRINT "HA BEGEPELED AZ EREDMENYT, MEGMONDOM,"
90 PRINT "MILYEN SZAMRA GONDOLTAL !"
100 PRINT
110 PRINT "MI AZ EREDMENY";
120 INPUT A
130 B=((A-13)/8)+28
140 PRINT "A GONDOLT SZAM";B
150 END
```

**65. Vegyük észre:**

- A feladatban fűzér változók szerepelnek.
- A táblázatot célszerű PRINT TAB(X) segítségével megszerkeszteni.

A program:

```
10 PRINT " "
20 PRINT TAB(12); "LAKCIMJEGYZEK"
30 PRINT
40 INPUT "NEV";A1#
50 INPUT "LAKCIM";B1#
60 INPUT "NEV";A2#
70 INPUT "LAKCIM";B2#
80 INPUT "NEV";A3#
90 INPUT "LAKCIM";B3#
100 PRINT " "
110 PRINT "NEV";
120 PRINT TAB(22); "LAKCIM"
130 PRINT A1#;
140 PRINT TAB(20);B1#
150 PRINT A2#;
160 PRINT TAB(20);B2#
170 PRINT A3#;
180 PRINT TAB(20);B3#
190 END
```

### 3.4 Néhány gyakori függvény

#### Az egészrész függvény INT(X)

**66. Vegyük észre:**

- Tetszőleges A szám egészrésze INT(A) alakban számítható.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(8); "EGESZRESZ MEGHATAROZO"  
30 PRINT  
40 INPUT "A SZAM"; A  
50 B=INT(A)  
60 PRINT A; "EGESZRESZE "; B  
70 END
```

**67. Vegyük észre:**

- Az egészrész függvény a számnál nem nagyobb egészek közül a legnagyobbat adja.
- Felfelé kerekítünk, ha a törtrész nagyobb vagy egyenlő 0,5-del.
- Ha a kerekítendő számhoz 0,5-et adunk, annak egészrésze pont a kerekített értéket adja.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(15); "KEREKITES"  
30 PRINT  
40 INPUT "A KEREKITENDO SZAM"; A  
50 B=A+.5  
60 B=INT(B)  
70 PRINT  
80 PRINT A; "EGESZEKRE KERKITETT ERTEKE "; B  
90 END
```

**68. Vegyük észre:**

- A pontszámokat öttel osztva és a hányados egészrészét számítva, az osztályzatnál eggyel kisebb számot kapunk. Így a pontszám egyszerűen számítható.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(15); "OSZTALYOZO"  
30 PRINT  
40 INPUT "A TANULO NEVE"; N$  
50 INPUT "A PONTSZAMA"; P  
60 J=INT(P/5)+1  
70 PRINT  
80 PRINT N$; " ERDEMJEJYE "; J  
90 END
```



### 69. Vegyük észre:

A címletezés a következő lépésekkel oldható meg:

- Vesszük a szám és az érme értékének hányadosát.
- Vesszük ennek egész részét. Ez megadja, hogy hány darab ilyen értékű pénzt kell felhasználni.
- Vesszük a maradékot. Ha az nem nulla, ezt a műveletsort végezzük a következő pénzérmével.
- Ha a maradék nulla, kiíratjuk a kapott értékeket.
- A feladat megoldását mindig a legnagyobb pénzérmével kell kezdeni.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(15); "CÍMLETEZŐ☐"  
30 A=16  
40 B=A/10  
50 B=INT(B)  
60 C=A-B*10  
70 D=C/5  
80 D=INT(D)  
90 E=C-D*5  
100 F=E/2  
110 F=INT(F)  
120 G=E-F*2  
130 PRINT "A 16 FT LELEGYSZERÜBB FELBONTÁSA:"  
140 PRINT  
150 PRINT B; "DB 10 FT-OS"  
160 PRINT D; "DB 5 FT-OS"  
170 PRINT F; "DB 2 FT-OS"  
180 PRINT G; "DB 1 FT-OS"  
190 END
```

### 70. Vegyük észre:

A feladat a következő lépésekkel oldható meg:

- Vesszük a szám és a három hányadosának egész részét.
- Vesszük a szám és egészrészének különbségét. Így megkaptuk a maradékot.
- Ha ez nem nulla, az előző műveletsor ismétlődik.
- Ha nulla, kiíratjuk az eredményeket, fordított sorrendben.

A program:

```
10 PRINT "☐"  
20 PRINT "A 11 A 3-AS SZÁMRENDSZERBEN:"  
30 A=11  
40 B=INT(A/3)  
50 C=A-B*3  
60 D=INT(B/3)  
70 E=B-D*3
```

```

80 F=INT(D/3)
90 G=D-F*3
100 PRINT G/E/C
110 END

```

**71. Vegyük észre:**

– A természetes esetben 6–14 éves kor között az életkor és az osztály közötti összefüggés:

Az osztály=életkor–5.

A program:

```

10 PRINT "□"
20 PRINT TAB(12); "OSZTALYBA SOROLO"
30 PRINT
40 INPUT "A TANULO NEVE"; N#
50 INPUT "ELETKORA (EV)"; E
60 X=E-5
70 PRINT N#; X# ". OSZTALYOS"
80 END

```

A négyzetgyök függvény SQR(X)

**72. Vegyük észre:**

– Tetszőleges A szám négyzetgyökét SQR(A) alakban számíthatjuk ki.

A program:

```

10 PRINT "□"
20 PRINT TAB(11); "NEGYZETGYOK SZAMOLO"
30 PRINT
40 INPUT "A SZAM"; X
50 Y=SQR(X)
60 PRINT X# "NEGYZETGYOKE" # Y
70 END

```

**73. Vegyük észre:**

– Az átfogó a Pitagorasz-tétel alapján:  $C = \sqrt{A^2 + B^2}$

– A kerület és a terület mérőszáma után írhatunk mértékegységet is.

A program:

```

10 PRINT "□"
20 PRINT TAB(10); "DEREKSZOGU HAROMSZOG"
30 PRINT
40 INPUT "EGYIK BEFOGO (CM)"; A
50 INPUT "MASIK BEFOGO (CM)"; B
60 C=SQR(A^2+B^2)
70 PRINT "AZ ATFOGO" # C# "CM"
80 K=A+B+C
90 T=A*B/2

```



```

100 PRINT "A HÁROMSZÖG KERÜLETE" ; K ; "CM"
110 PRINT "A HÁROMSZÖG TERÜLETE" ; T ; "CM2"
120 END

```

### A véletlenszámot előállító (generáló) függvény RND(X)

#### 74. Vegyük észre:

- Véletlenszámok az RND(X) függvény segítségével hozhatók létre.
- Ezek mindig 0–1 közé esnek, de az 1-et nem érik el.
- Egész számok generálásához a véletlenszámot egy egész számmal szorozni kell.
- 0–14 közötti számot kapok, ha a véletlenszámot 15-tel szorzom és annak egészrészét veszem.
- 1–15 közötti véletlenszámot kapok, ha az így kapott értékhez még 1-et adok.

A program:

```

10 PRINT "□"
20 PRINT TAB(10) ; "VELETLEN SZÁMOK (1-15)"
30 PRINT
40 A=INT(RND(1)*15)+1
50 B=INT(RND(1)*15)+1
60 C=INT(RND(1)*15)+1
70 D=INT(RND(1)*15)+1
80 E=INT(RND(1)*15)+1
90 PRINT A ; B ; C ; D ; E
100 END

```

#### 75. Vegyük észre:

- 0–68 közötti véletlenszámokat kapok, ha a gép által generált véletlenszámot 69-cel szorzom és az eredmény egészrészét veszem. A számok között így a 68 is szerepelhet.
- 32–100 közöttiek a véletlenszámok, ha az eredményhez még 32-t is adok.

A program:

```

10 PRINT "□"
20 PRINT TAB(10) ; "VELETLEN SZÁMOK (32-100)"
30 PRINT
40 A=INT(RND(1)*69)+32
50 B=INT(RND(1)*69)+32
60 C=INT(RND(1)*69)+32
70 D=INT(RND(1)*69)+32
80 E=INT(RND(1)*69)+32
90 PRINT A ; B ; C ; D ; E
100 END

```

76.

```
10 PRINT "□"  
20 PRINT TAB(14); "LOTTO JATEK"  
30 PRINT  
40 A=INT(RND(1)*90)+1  
50 B=INT(RND(1)*90)+1  
60 C=INT(RND(1)*90)+1  
70 D=INT(RND(1)*90)+1  
80 E=INT(RND(1)*90)+1  
90 PRINT "A NYERO SZAMOK:"  
95 PRINT  
100 PRINT A;B;C;D;E  
110 END
```

77.

```
10 PRINT "□"  
20 PRINT TAB(14); "KOCKADOBALO"  
30 PRINT  
40 PRINT "A TE DOBASAI:"  
50 A=INT(RND(1)*6)+1  
60 B=INT(RND(1)*6)+1  
70 C=INT(RND(1)*6)+1  
80 PRINT A,B,C  
90 PRINT  
95 PRINT  
100 PRINT "A PARTNER DOBASAI:"  
110 D=INT(RND(1)*6)+1  
120 E=INT(RND(1)*6)+1  
130 F=INT(RND(1)*6)+1  
140 PRINT D,E,F  
150 PRINT  
160 A=A+B+C  
170 D=D+E+F  
180 PRINT "A TE PONTJAI:";A  
190 PRINT  
200 PRINT "A PARTNERED PONTJAI:";D  
210 END
```

### 3.5 Az IF...THEN és a GOTO kapcsolata

– IF (valamilyen reláció), THEN (valamilyen utasítás).

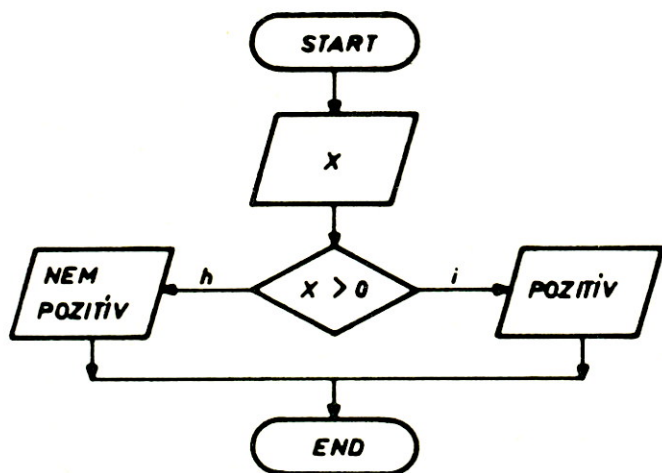


- Ha az IF után írt reláció igaz, akkor a THEN után írt utasítást hajtja végre a gép.
- Ha az IF után írt reláció nem teljesül, akkor a program futása a következő soron folytatódik.
- A THEN után leggyakrabban a GOTO (sorszám) utasítás található, aminek hatására a program futása a GOTO után írt sorszámnál folytatódik.
- A THEN után több utasítást is írhatunk, ezeket kettősponttal kell elválasztani egymástól. Ekkor ezek egy programsorként értelmeződnek.
- IF THEN GOTO-val ciklus is képezhető. A ciklus addig ismétlődik, amíg az IF THEN közé zárt feltétel teljesül.
- Csak GOTO utasítással is képezhető ciklus. Ekkor feltételtől függetlenül, ahányszor a GOTO utasításhoz ér a program, a ciklus újra indul. Ez a végtelen ciklus.

### 78. Vegyük észre:

- Ha a szám  $>0$ , akkor pozitív.
- Célszerű folyamatábrát készíteni, amelyen a program főbb lépéseit ábrázoljuk.
- Célszerű a folyamatábra elemeit a „nem” ág felé számozni, mert így ugró utasítás takarítható meg.

A folyamatábra:



A program:

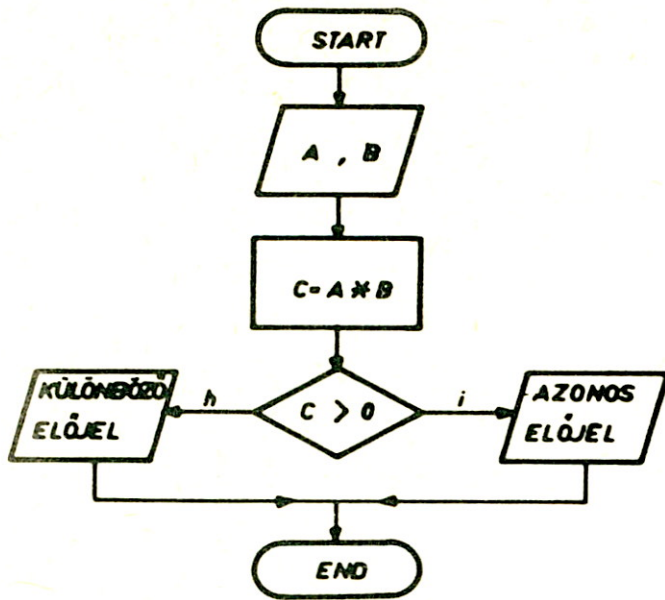
```

10 PRINT " "
20 PRINT TAB(8); "POZITIV VAGY NEM?"
30 PRINT
40 INPUT "A SZAM="; X
50 IF X > 0 THEN GOTO 80
60 PRINT X; "NEM POZITIV"
70 GOTO 90
80 PRINT X; "POZITIV"
90 END
  
```

79. Vegyük észre:

- Két nullától különböző szám azonos előjelű, ha a szorzatuk pozitív; különböző előjelűek, ha a szorzat negatív.

A folyamatábra:



A program:

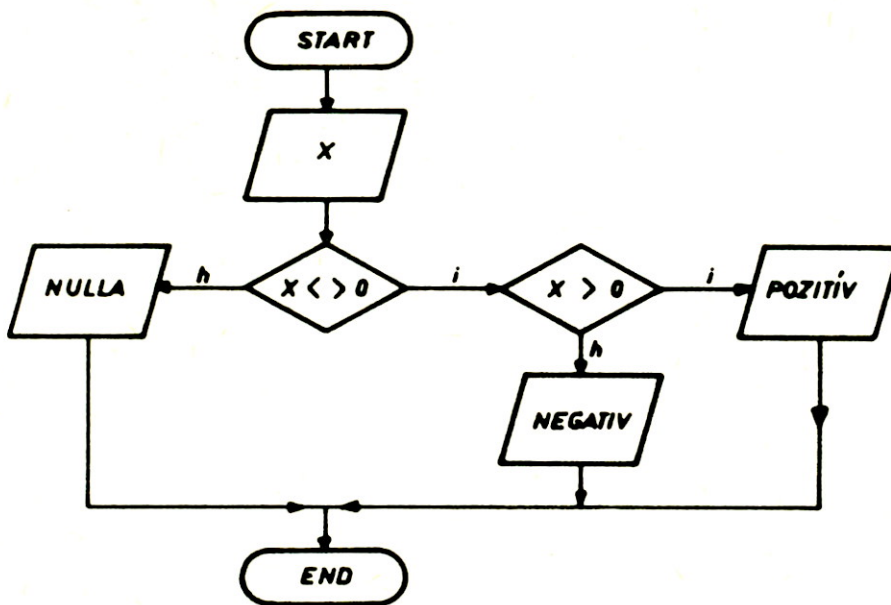
```
10 PRINT " "
20 PRINT TAB(8); "ELOJELVIZSGALO PROGRAM"
30 PRINT
40 INPUT "ELSO SZAM="; A
50 INPUT "MASODIK SZAM="; B
55 PRINT
60 C=A*B
70 IF C>0 THEN GOTO 100
80 PRINT A;B;"KULONBOZO ELOJELUEK"
90 GOTO 110
100 PRINT A;B;"AZONOS ELOJELUEK"
110 END
```

80. Vegyük észre:

- Három lehetséges eset van, így két kérdésre van szükség.
- Például így: 1. a szám nulla.  
2. A szám pozitív



A folyamatábra:



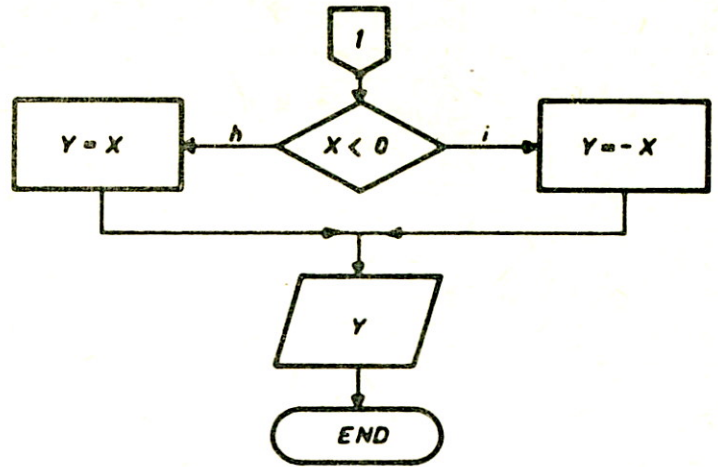
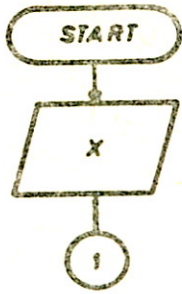
A program:

```
10 PRINT " "
20 PRINT TAB(8); "MILYEN ELOJELU A SZAM?"
30 PRINT
40 INPUT "A SZAM="; X
45 IF X <> 0 THEN GOTO 50
46 PRINT X; "NULLA"
47 GOTO 90
50 IF X > 0 THEN GOTO 80
60 PRINT X; "NEGATIV"
70 GOTO 90
80 PRINT X; "POZITIV"
90 END
```

81. Vegyük észre:

- Nem negatív számok abszolútértéke önmaga, negatív szám abszolútértéke a szám ellentettje.

A folyamatábra:



A program:

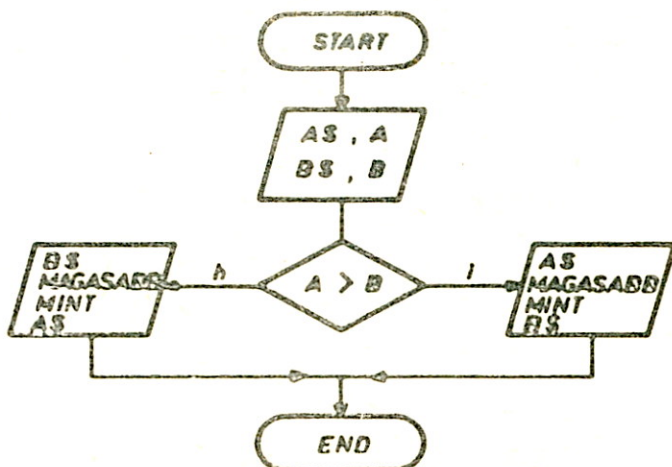
```

10 PRINT " "
20 PRINT "ABSZOLUTERTEK SZAMOLO"
30 PRINT
40 INPUT "A SZAM=" ; X
50 IF X < 0 THEN GOTO 80
60 Y=X
70 GOTO 90
80 Y=-X
90 PRINT X ; "ABSZOLUTERTEKE" ; Y
100 END
  
```

82. Vegyük észre:

- Sztring és valós típusú változókkal kell dolgozni.
- Összehasonlítani csak a testmagasságokat kell. A neveket a kiíratásnál használjuk.

A folyamatábra:





A program:

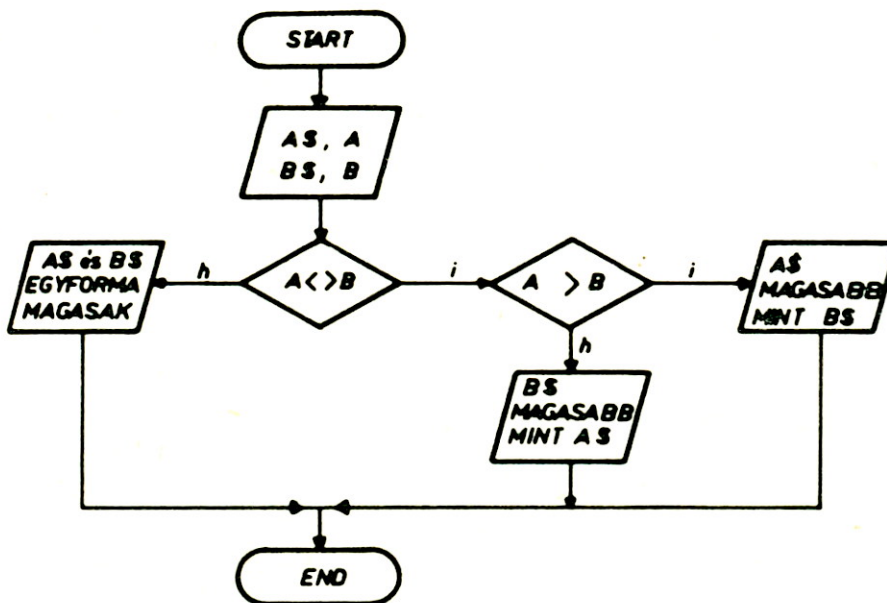
```
10 PRINT "❏"
20 PRINT TAB(4); "TESTMAGASSAGOK OSSZEHASONLITASA"
30 PRINT
40 INPUT "ELSO NEV: "; A$
50 INPUT "TESTMAGASSAGA: "; A
60 INPUT "MASODIK NEV: "; B$
70 INPUT "TESTMAGASSAGA: "; B
80 PRINT
90 IF A > B THEN GOTO 120
100 PRINT B$; " MAGASABB MINT "; A$
110 GOTO 130
120 PRINT A$; " MAGASABB MINT "; B$
130 END
```

83. Vegyük észre:

- Három lehetséges eset van, így két kérdésre van szükség. Pl.:

  1. A egyenlő B-vel?
  2. A nagyobb, mint B?

A folyamatábra:



A program:

```
10 PRINT "❏"
20 PRINT TAB(4); "TESTMAGASSAGOK OSSZEHASONLITASA"
30 PRINT
40 INPUT "ELSO NEV: "; A$
50 INPUT "TESTMAGASSAGA: "; A
60 INPUT "MASODIK NEV: "; B$
```

```

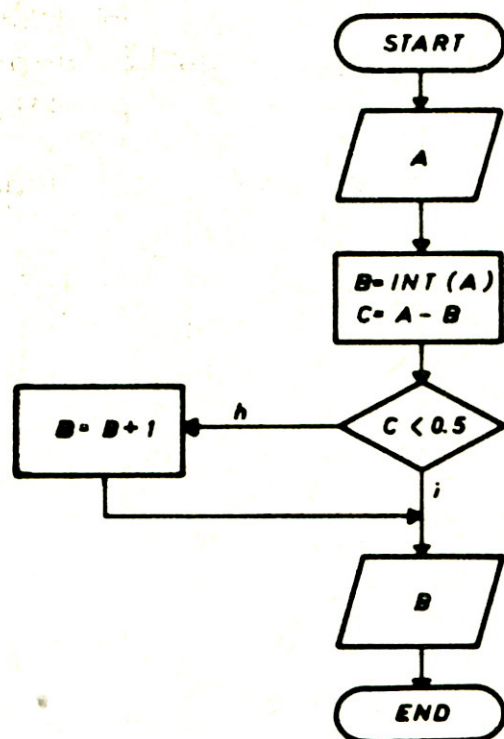
70 INPUT "TESTMAGASSAGA:";B
80 PRINT
81 IF A<>B THEN GOTO 90
82 PRINT A$;" ES ";B$;" EGYFORMA MAGASAK "
83 GOTO 130
90 IF A>B THEN GOTO 120
100 PRINT B$;" MAGASABB MINT ";A$
110 GOTO 130
120 PRINT A$;" MAGASABB MINT ";B$
130 END

```

84. Vegyük észre:

- Egy számot lefelé kerekítünk, ha a törtrésze kisebb 0,5-nél, különben felfelé kerekítünk.
- A törtrészt a szám és egészrészének különbségeként kapjuk.

A folyamatábra:



A program:

```

10 PRINT "□"
20 PRINT TAB(9); "FELTETELES KEREKITO"
30 PRINT
40 INPUT "A SZAM=";A
50 B=INT(A)
60 C=A-B
70 IF C<0.5 THEN GOTO 90
80 B=B+1
90 PRINT A;" EGESZEKRE KEREKITETT ERTEKE ";B
100 END

```



85. Vegyük észre:

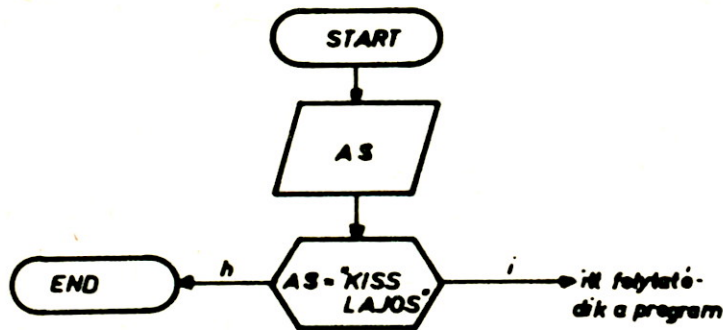
- Olyan programvédelem nincs, amit senki sem tud kijátszani. Ez a nagy szoftverházak legnagyobb gondja.
- Nem szakemberekkel szemben azonban védelmet ad, ha úgy szervezzük meg a programot, hogy a kitűzött feladatot megoldó rész elé egy olyan programrészletet írunk, amely csak akkor engedi tovább futni a programot, ha egy bekért érték megfelel egy bizonyos feltételnek.

Például a program csak akkor fut tovább, ha az elején begépelik az írójának nevét.

Kitalálhatsz jelszót, számsorozatot stb.

Ha nem a kívánt dolgot gépelik be, a gép megáll.

A folyamatábra:



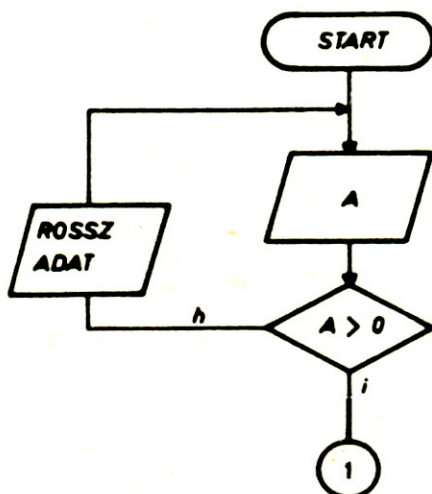
A program:

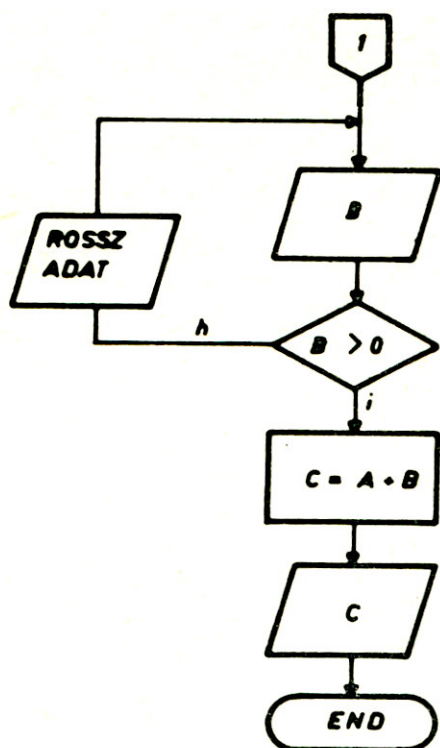
```
10 PRINT " "
20 INPUT A$
30 IF A$="KISS LAJOS" THEN GOTO 50
40 STOP
50 REM ITT INDUL A PROGRAM
```

86. Vegyük észre:

- Rossz adatnak minősül minden nem pozitív szám.
- Ha az adat rossz, újra vissza kell térni a bekérő utasításhoz.
- A műveletet csak megfelelő számokkal végezzük.

A folyamatábra:





A program:

```

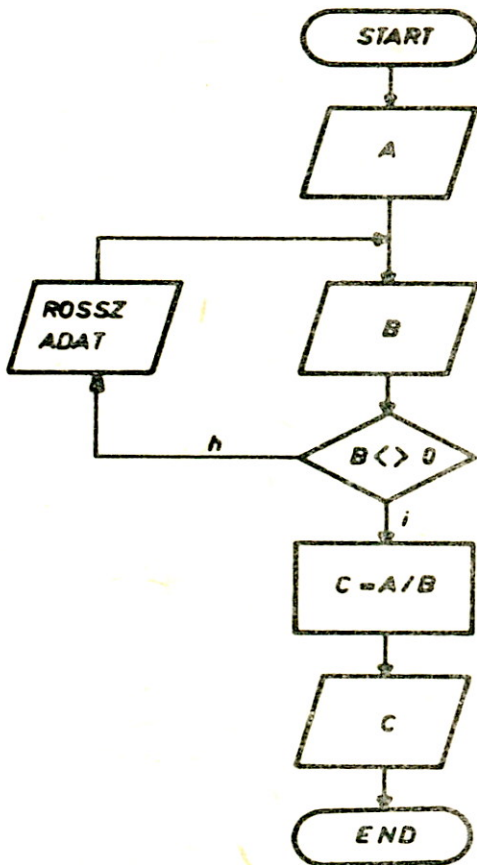
10 PRINT " "
20 PRINT TAB(11); "OSSZEADO PROGRAM"
30 PRINT
40 INPUT "ELSO POZITIV SZAM"; A
50 IF A > 0 THEN GOTO 80
60 PRINT "ROSSZ ADAT"
70 GOTO 40
80 INPUT "MASODIK POZITIV SZAM"; B
90 IF B > 0 THEN GOTO 120
100 PRINT "ROSSZ ADAT"
110 GOTO 80
120 C = A + B
130 PRINT
140 PRINT "A KET SZAM OSSZEGE: "; C
150 END
  
```

**87. Vegyük észre:**

- Osztas esetén rossz adatnak minősül, ha az osztó értéke nulla. Ilyen osztás nem értelmezett.
- Az osztandóra nincs megkötés.



A folyamatábra:



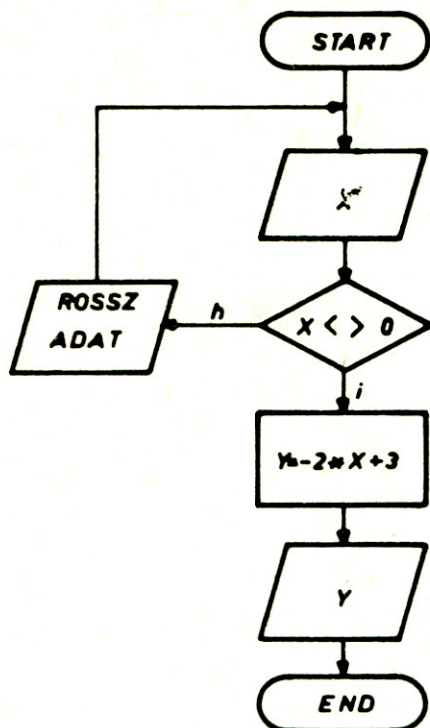
A program:

```
10 PRINT " "
20 PRINT TAB(11); "KET SZAM HANYADOSA"
30 PRINT
40 INPUT "AZ OSZTANDO: "; A
50 INPUT "AZ OSZTO: "; B
60 IF B <> 0 THEN GOTO 90
70 PRINT "ROSSZ ADAT"
80 GOTO 50
90 C = A/B
100 PRINT
110 PRINT "A KET SZAM HANYADOSA: "; C
120 END
```

**88. Vegyük észre:**

- Rossz adatnak minősül, ha az X értéke nulla.

A folyamatábra:



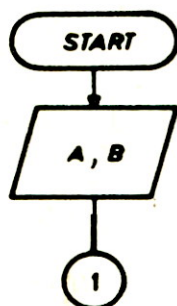
A program:

```
10 PRINT " "
20 PRINT "Y=-2*X+3 FV.ERTEKENEK SZAMITASA X<>0-RA"
30 PRINT
40 INPUT "X=" ; X
50 IF X<>0 THEN GOTO 80
60 PRINT "ROSSZ ADAT"
70 GOTO 40
80 Y=-2*X+3
90 PRINT "X=" ; X ; "-RE Y=-2*X+3 ERTEKE=" ; Y
100 END
```

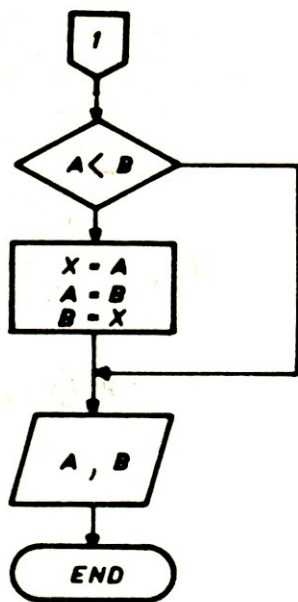
**89. Vegyük észre:**

- Ha az első szám volt a kisebb, akkor a begépelés sorrendjében kell azokat kiíratni.
- Ha az első szám nem nagyobb, akkor elvégezve a változók cseréjét a memóriában, a változókat itt is az eredeti sorrendben kell kiíratni.

A folyamatábra:







A program:

```

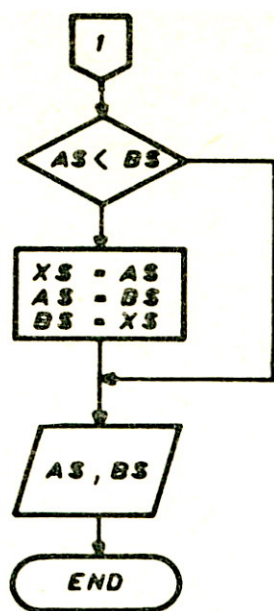
10 PRINT " "
20 PRINT TAB(2); "KET SZAM NOVEKVO SORRENDBE RENDEZESE "
30 PRINT
40 INPUT "ELSO SZAM"; A
50 INPUT "MASODIK SZAM"; B
60 IF A < B THEN GOTO 90
70 X = A
71 A = B
72 B = X
90 PRINT A; B
100 END
  
```

90. Vegyük észre:

- Sztring típusú változókat is lehet nagyság szerint összehasonlítani.
- Az abc-ben előrébb elhelyezkedő betű kisebb értékű.
- A feladat az előzőhöz hasonlóan oldható meg.
- A változók cseréjéhez felvett új változónak is sztring típusúnak kell lennie.

A folyamatábra:





A program:

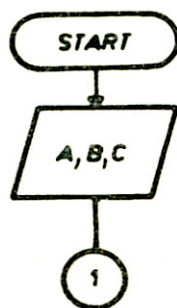
```

10 PRINT " "
20 PRINT TAB(8); "KET NEV NEVSORBA RENDEZESE"
30 PRINT
40 INPUT "ELSO NEV"; A$
50 INPUT "MASODIK NEV"; B$
60 IF A$ < B$ THEN GOTO 100
70 X$ = A$
71 A$ = B$
72 B$ = X$
100 PRINT
110 PRINT A$, B$
120 END
  
```

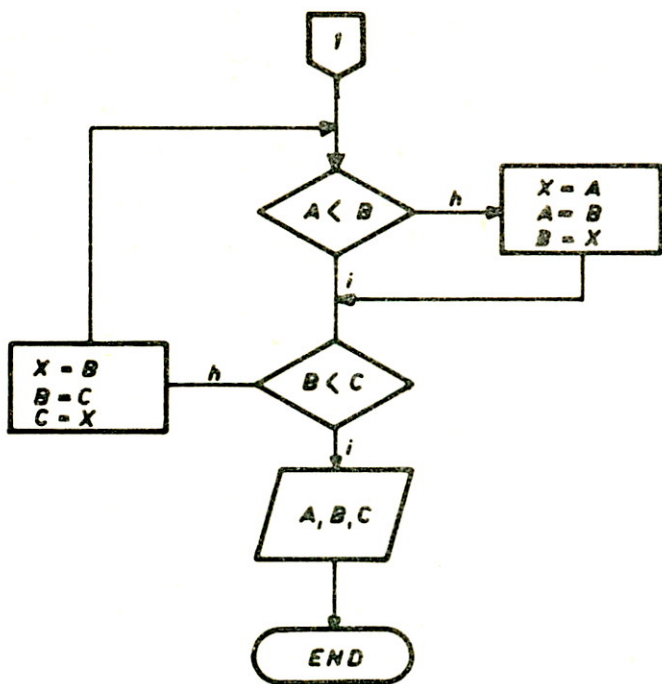
91. Vegyük észre:

- Legjobb esetben a számok már begépelésnél jó sorrendben vannak.
- Legrosszabb esetben egyik sincs a megfelelő helyen.
- Rendezés előtt ki kell íratni a sorrendet, mert a változók esetleges cseréje a változók tartalmát változtatja.

A folyamatábra:







A program:

```

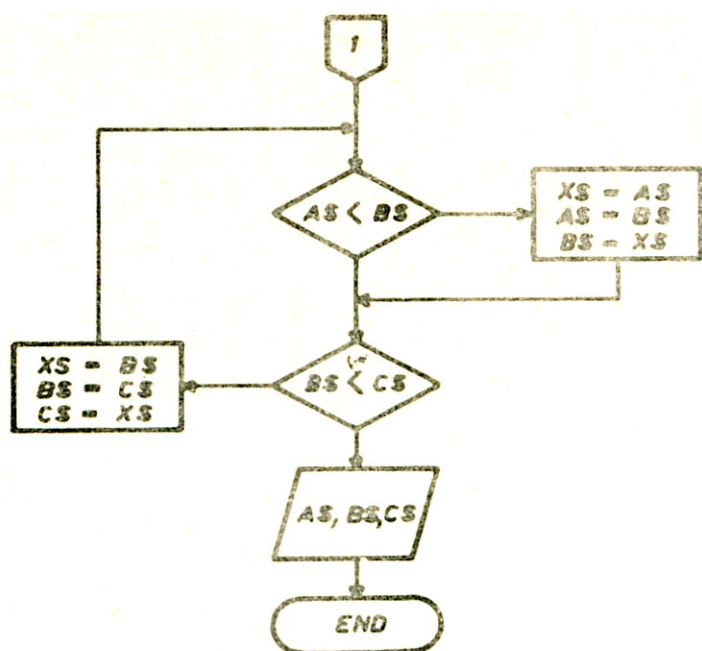
10 PRINT " "
20 PRINT TAB(1); "HAROM SZAM NOVEKVO SORRENDBE RENDEZESE "
30 PRINT
40 INPUT "ELSO SZAM"/A
50 INPUT "MASODIK SZAM"/B
60 INPUT "HARMADIK SZAM"/C
70 PRINT " "
80 PRINT " ELŐREDETI SORREND: "/A/B/C
90 IF A<B THEN GOTO 110
100 X=A:A=B:B=X
110 IF B<C THEN GOTO 140
120 X=B:B=C:C=X
130 GOTO 90
140 PRINT " ELŐRENDEZES UTANI SORREND: "/A/B/C
150 END
  
```

92. Vegyük észre:

- Szerkezetében nagyon hasonló az előző feladathoz.
- A változók ez esetben sztring típusúak.

A folyamatábra:





A program:

```

10 PRINT " "
20 PRINT TAB(10); "3 NEV NEVSORBA RENDEZESE"
30 PRINT
40 INPUT "ELSO NEV" /A#
50 INPUT "MASODIK NEV" /B#
60 INPUT "HARMADIK NEV" /C#
70 PRINT " "
80 PRINT "EREDETI SORREND:" /A#,B#,C#
90 IF A#<B# THEN GOTO 110
100 X#=A#:A#=B#:B#=X#
110 IF B#<C# THEN GOTO 140
120 X#=B#:B#=C#:C#=X#
130 GOTO 80
140 PRINT "RENDEZES UTANI:" /A#,B#,C#
150 END
  
```

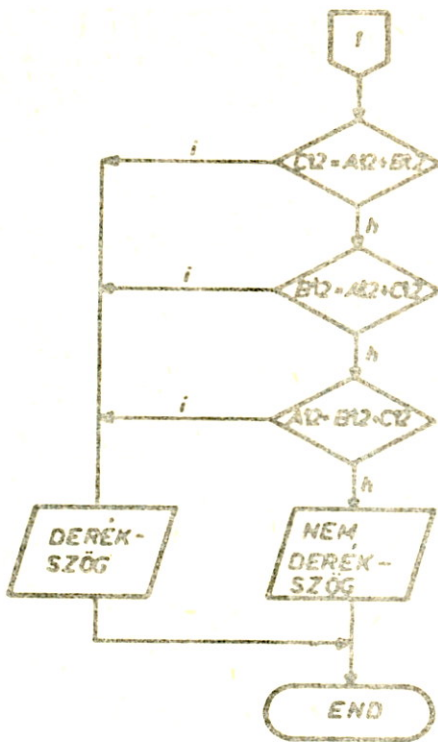
93. Vegyük észre:

- A Pitagorasz-tétel alapján a háromszög derékszögű, ha az átfogó négyzete egyenlő a befogók négyzetének összegével.
- A három oldal bekérésekor nem tudjuk, hogy melyik tekinthető befogónak. Így mindháromat meg kell vizsgálni, hogy igaz-e rá az előző állítás.

A folyamatábra:







A program:

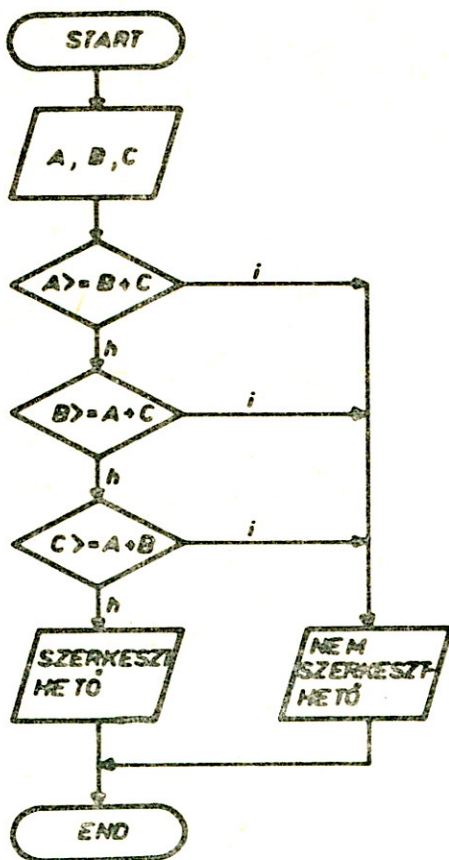
```

10 PRINT "Q"
20 PRINT TAB(5); "DEREKSZOGU-E A HAROMSZOG?"
30 PRINT
40 PRINT "A HAROM OLDAL:"
50 INPUT "A="; A
55 INPUT "B="; B
60 INPUT "C="; C
70 IF C^2=A^2+B^2 THEN GOTO 120
80 IF B^2=A^2+C^2 THEN GOTO 120
90 IF A^2=B^2+C^2 THEN GOTO 120
100 PRINT "A HAROMSZOG NEM DEREKSZOGU"
110 GOTO 130
120 PRINT "A HAROMSZOG DEREKSZOGU"
130 END
  
```

94. Vegyük észre:

- Három szakaszból szerkeszthető háromszög, ha minden oldalára igaz, hogy kisebb, mint a másik két oldal.
- Ha bármely oldalára ez nem igaz, akkor a három szakaszból nem szerkeszthető háromszög.

A folyamatábra:



A program:

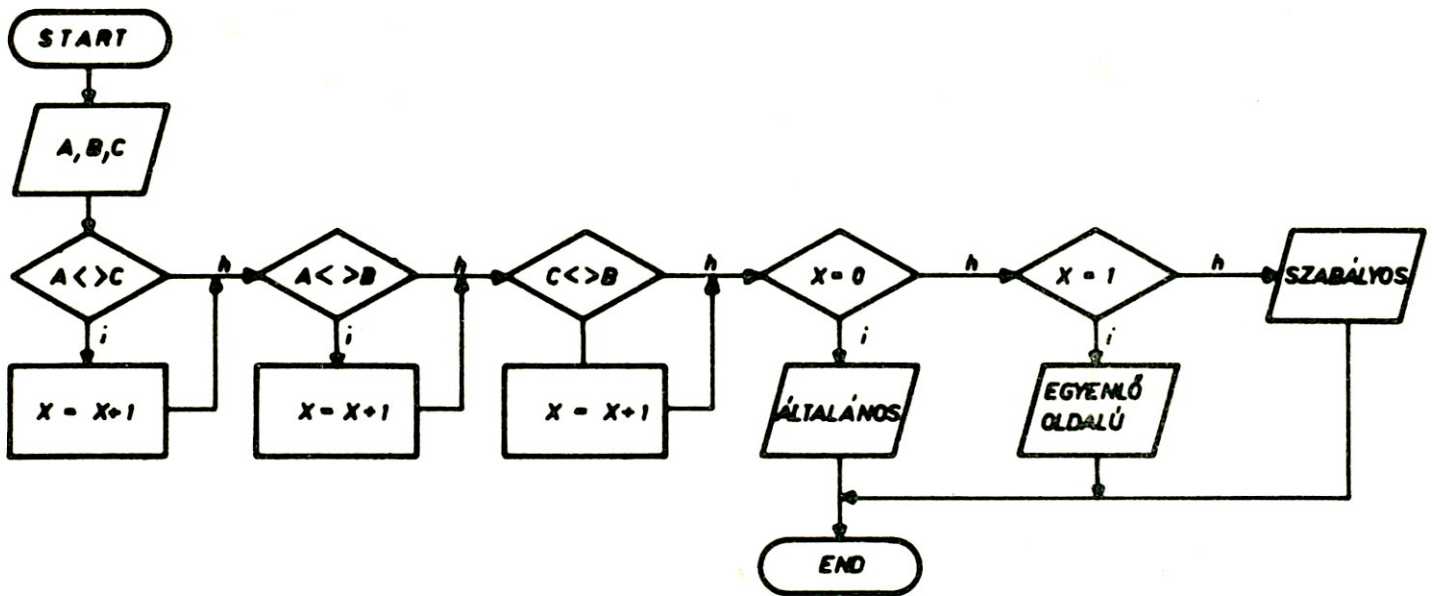
```
10 PRINT " "
20 PRINT TAB(5); "SZERKESZTHETO-E HAROMSZOG?"
30 PRINT
40 PRINT "A HAROM OLDAL: "
50 INPUT "A="; A
60 INPUT "B="; B
70 INPUT "C="; C
80 IF A >= B + C THEN GOTO 130
90 IF B >= A + C THEN GOTO 130
100 IF C >= A + B THEN GOTO 130
110 PRINT "SZERKESZTHETO HAROMSZOG"
120 GOTO 140
130 PRINT "NEM SZERKESZTHETO HAROMSZOG"
140 END
```

95. Vegyük észre:

- Egy háromszög lehet általános, egyenlő szárú vagy szabályos.
- X egy kapcsoló. Értéke a következőképpen változzon:
  - X=0, ha a háromszög oldalai különbözők (általános)
  - X=1, ha a háromszög két oldala egyenlő (egyenlő szárú)
  - X=3, ha a háromszög három oldala egyenlő (szabályos).



A folyamatábra:



A program:

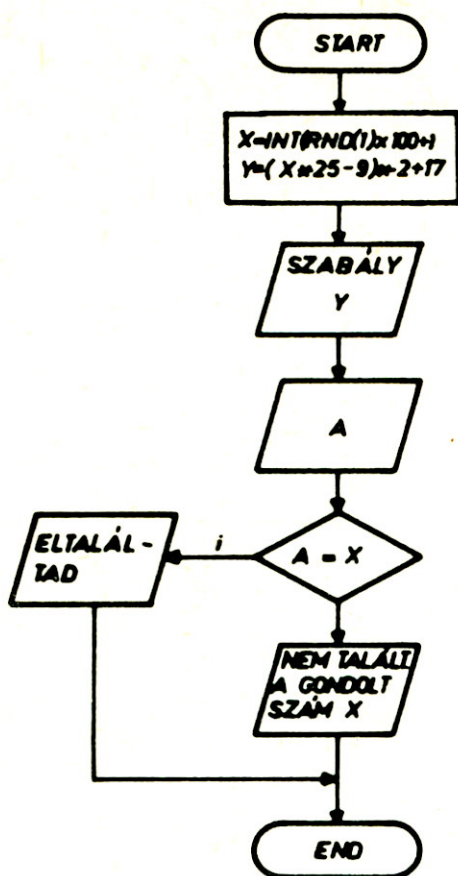
```

10 PRINT "□"
20 PRINT TAB(5); "HAROMSZOG OLDALAINAK HASONLITASA"
30 PRINT
40 X=0
50 PRINT "A HÁROM OLDAL: "
60 INPUT "A="; A
70 INPUT "B="; B
80 INPUT "C="; C
90 IF A <> C THEN GOTO 110
100 X=X+1
110 IF A <> B THEN GOTO 130
120 X=X+1
130 IF C <> B THEN GOTO 150
140 X=X+1
150 IF X=0 THEN PRINT "ÁLTALÁNOS HAROMSZOG"; GOTO 180
160 IF X=1 THEN PRINT "EGYENLŐSZARU HAROMSZOG"; GOTO 180
170 PRINT "SZABÁLYOS HAROMSZOG"
180 END
  
```

96. Vegyük észre:

- A gondolt számot véletlenszám generátorral kell létrehozni.
- Azt, hogy milyen számkörben dolgozzunk, célszerű a játékosok korához szabni. A mi megoldásunkban 1–100 közötti számokat generál a gép.
- Rossz válasz esetén célszerű a jó megoldást is megmutatni, hogy a játékos ellenőrizni tudja saját megoldásait.

A folyamatábra:



A program:

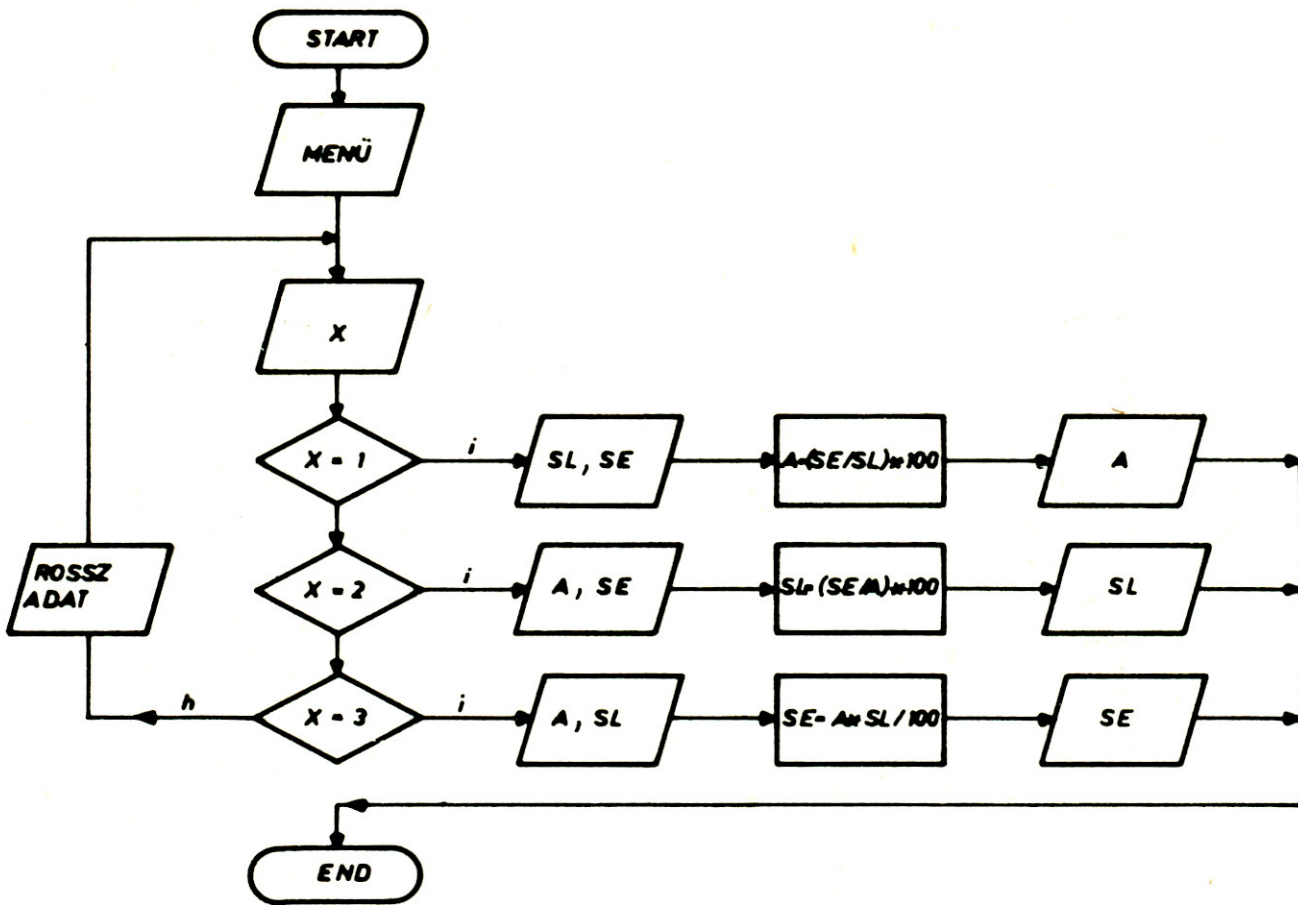
```
10 PRINT " "
20 PRINT TAB(11); "SZÁMKITALALO"
30 PRINT
40 X=INT(RND(1)*100)+1
50 Y=(X*25-9)*2+17
60 PRINT "GONDOLTAM EGY SZÁMOT. SZOROZTAM 25-TEL."
70 PRINT "ELVETTEM BELOLE 9-ET, SZOROZTAM 2-VEL"
80 PRINT "VEGUL HOZZAADTAM 17-ET. EREDMENY" Y
90 INPUT "MI A GONDOLT SZÁM" A
100 IF A=X THEN GOTO 130
110 PRINT "NEM TALÁLTAD EL. A GONDOLT SZÁM" X
120 GOTO 140
130 PRINT "ELTALÁTAD"
140 END
```

#### 97. Vegyük észre:

- Az egyes programrészletek megírása nem jelent problémát, hiszen csak az ismert képleteket kell alkalmazni.
- A programrészletek összefűzésére nagyon jól használható az úgynevezett menü, amelyet a program 40...80. sora tartalmaz.
- X értékének megfelelően a program a használó által kért programrészen folytatódik.



A folyamatábra:



A program:

```

10 PRINT "M"
20 PRINT TAB(12); "SZAZALEKSZAMITAS"
30 PRINT
40 PRINT "MIT SZAMITUNK"
50 PRINT "1. ALAPOT"
60 PRINT "2. SZAZALEKLABAT"
70 PRINT "3. SZAZALEKERTEKET"
80 INPUT X
90 IF X=1 THEN GOTO 140
100 IF X=2 THEN GOTO 190
110 IF X=3 THEN GOTO 240
120 PRINT "ROSSZ ADAT"
130 GOTO 80
140 INPUT "SZAZALEKLAB" : SL
150 INPUT "SZAZALEKERTEK" : SE
160 A = (SE/SL)*100
170 PRINT "AZ ALAP" : A
180 GOTO 280
190 INPUT "ALAP" : A

```

```

200 INPUT "SZAZALEKERTEK" : SE
210 SL = (SE/A) * 100
220 PRINT "A SZAZALEKLAB" : SL
230 GOTO 260
240 INPUT "ALAP" : A
250 INPUT "SAZALEKLAB" : SL
260 SE = A * SL / 100
270 PRINT "A SZAZALEKERTEK" : SE
280 END

```

98.

Mindkét feladat szerkezetében azonos a 97. feladattal. Ezért a feladatok folyamatábráját külön nem közöljük.

A program:

```

10 PRINT "Q"
20 PRINT TAB(15) : "A NYOMAS"
30 PRINT
40 PRINT "MIT SZAMITUNK"
50 PRINT "1. NYOMAS"
60 PRINT "2. NYOMOTT FELULET"
70 PRINT "3. NYOMOERO"
80 INPUT X
90 IF X=1 THEN GOTO 140
100 IF X=2 THEN GOTO 190
110 IF X=3 THEN GOTO 240
120 PRINT "ROSSZ ADAT"
130 GOTO 80
140 INPUT "NYOMOTT FELULET (M^2)" : A
150 INPUT "NYOMOERO (N)" : F
160 P = F / A
170 PRINT "A NYOMAS (PA)" : P
180 GOTO 280
190 INPUT "NYOMAS (PA)" : P
200 INPUT "NYOMOERO (N)" : F
210 A = F / P
220 PRINT "A NYOMOTT FELULET (M^2)" : A
230 GOTO 280
240 INPUT "NYOMAS (PA)" : P
250 INPUT "NYOMOTT FELULET (M^2)" : A
260 F = P * A
270 PRINT "A NYOMOERO (N)" : F
280 END

```



99.

A program:

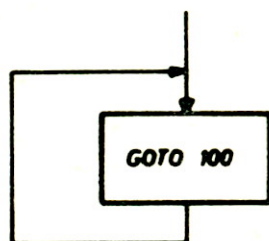
```
10 PRINT "V"
20 PRINT TAB(12); "OHM TORVENYE"
30 PRINT
40 PRINT "MIT SZAMITUNK"
50 PRINT "1. ELLENALLAS"
60 PRINT "2. ARAMEROSSEG"
70 PRINT "3. FESZULTSEG"
80 INPUT X
90 IF X=1 THEN GOTO 140
100 IF X=2 THEN GOTO 180
110 IF X=3 THEN GOTO 240
120 PRINT "ROSSZ ADAT"
130 GOTO 80
140 INPUT "ARAMEROSSEG (A)" ; I
150 INPUT "FESZULTSEG (V)" ; U
160 R=U/I
170 PRINT "ELLENALLAS (OHM)" ; R
180 GOTO 280
180 INPUT "ELLENALLAS (OHM)" ; R
200 INPUT "FESZULTSEG (V)" ; U
210 I=U/R
220 PRINT "AZ ARAMEROSSEG (A)" ; I
230 GOTO 280
240 INPUT "ELLENALLAS (OHM)" ; R
250 INPUT "ARAMEROSSEG (A)" ; I
260 U=I*R
270 PRINT "FESZULTSEG (V)" ; U
280 END
```

### 3.6 A ciklus (IF... THEN alkalmazásával)

100. Vegyük észre:

- A gép akkor írja ki a READY-t, amikor a program futása befejeződik.
- Ezt megakadályozhatjuk, ha a program utolsó sorába nem END-et, hanem egy önmagára ugrató utasítást írunk. Ekkor egy végtelen ciklus jön létre, amely csak billentyűzetről szakítható meg (RUN/STOP).

A folyamatábra:



A program:  
1000 GOTO 1000

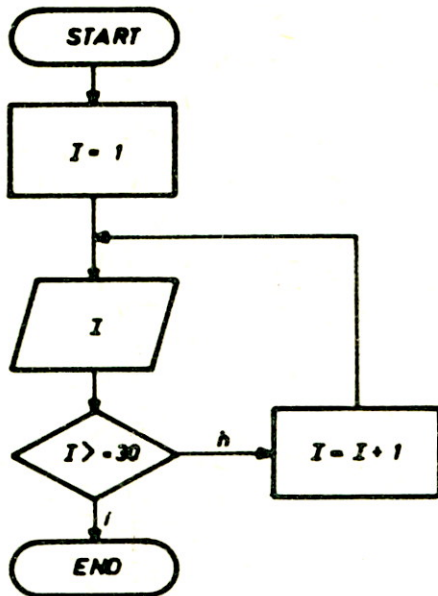
**101. Vegyük észre:**

- A feladatot kétféleképpen is megközelíthetjük:
  1. A ciklusváltozó értéke a lekérdezés után módosul.
  2. A ciklusváltozó értéke a lekérdezés előtt változik.

**a/1. Vegyük észre:**

- A ciklusváltozó kezdőértéke 1, és amíg az kisebb mint 30, az értéke egyesével nő.

A folyamatábra:



A program:

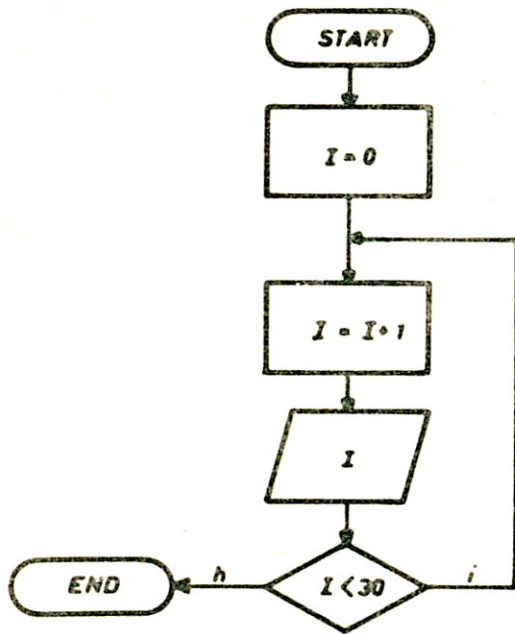
```
10 PRINT " "
20 PRINT TAB(15); "SZAMOLO"
30 I=1
40 PRINT I
50 IF I >= 30 THEN GOTO 80
60 I=I+1
70 GOTO 40
80 END
```

**a/2. Vegyük észre:**

- Ilyen szervezés mellett a ciklusváltozó kezdőértéke 0, hiszen most azonnal növelem eggyel. A ciklusváltozó értéke itt is 30-ig nő.



A folyamatábra:



A program:

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=0  
40 I=I+1  
50 PRINT I  
60 IF I<30 THEN GOTO 40  
70 END
```

Megjegyzés:

- A feladat többi része szerkezetében megegyezik az itt közölt programok egyikével. Ezért folyamatábrát külön nem közlünk.

b/1

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=20  
40 PRINT I  
50 IF I>=40 THEN GOTO 80  
60 I=I+2  
70 GOTO 40  
80 END
```

b/2

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=18  
40 I=I+2  
50 PRINT I  
60 IF I<40 THEN GOTO 40  
70 END
```

c/1

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=3  
40 PRINT I  
50 IF I >= 9 THEN GOTO 80  
60 I=I+0.5  
70 GOTO 40  
80 END
```

c/2

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=2.5  
40 I=I+0.5  
50 PRINT I  
60 IF I < 9 THEN GOTO 40  
70 END
```

d/1

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=10  
40 PRINT I  
50 IF I >= 200 THEN GOTO 80  
60 I=I+10  
70 GOTO 40  
80 END
```

d/2

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=0  
40 I=I+10  
50 PRINT I  
60 IF I < 200 THEN GOTO 40  
70 END
```

e/1

```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=50  
40 PRINT I  
50 IF I <= 20 THEN GOTO 80  
60 I=I-1  
70 GOTO 40  
80 END
```

e/2

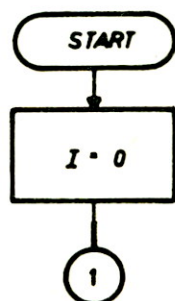
```
10 PRINT "□"  
20 PRINT TAB(15); "SZAMOLO"  
30 I=51  
40 I=I-1  
50 PRINT I  
60 IF I > 20 THEN GOTO 40  
70 END
```

102. a)

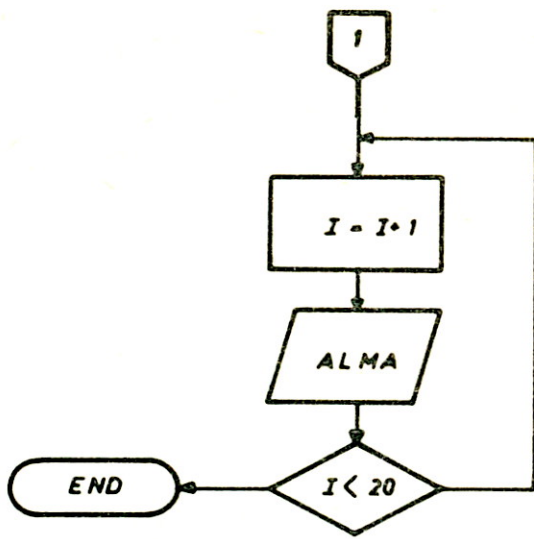
Vegyük észre:

- Az ALMA szó hússzor jelenik meg a képernyőn, ezért a ciklusmagnak hússzor kell végrehajtódnia.

A folyamatábra:







A program:

```

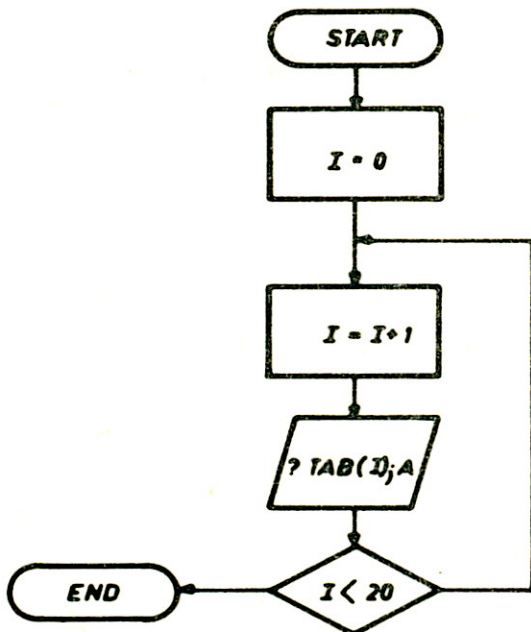
10 PRINT "1"
20 I = 0
30 I = I + 1
40 PRINT "ALMA"
50 IF I < 20 THEN GOTO 30
60 END
  
```

b)

Vegyük észre:

- Az ALMA szót a sor anyyadik oszlopába kell kiíratni, ahányadszor a ciklusmag végrehajtódik. Így a PRINT TAB(I) használata célszerű.

A folyamatábra:



A program:

```
10 PRINT "□"  
20 I=0  
30 I=I+1  
40 PRINT TAB(1); "ALMA"  
50 IF I<20 THEN GOTO 30  
60 END
```

c)

Vegyük észre:

- A páratlan számú sorokban az ALMA szó mindig az első, a páros számú sorokban mindig a harmadik oszlopban kezdődik. A PRINT TAB(X) használata célszerű.
- A feladat szerkezetében hasonló az előző két feladathoz.

A program:

```
10 PRINT "□"  
20 I=0  
30 I=I+1  
40 PRINT "ALMA"  
50 PRINT TAB(3); "ALMA"  
60 IF I<10 THEN GOTO 30  
70 END
```

d)

Vegyük észre:

- A program csak minden páratlan számú sorba ír, tehát minden páros sor üres.

A program:

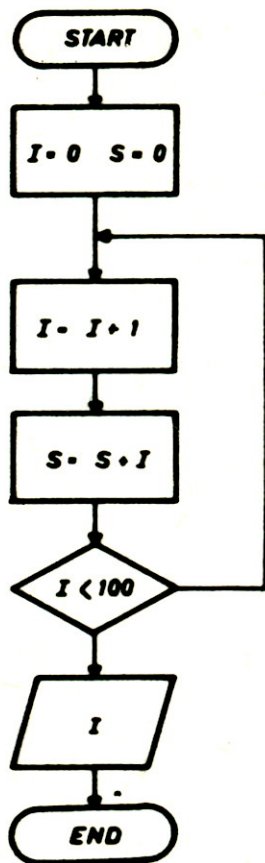
```
10 PRINT "□"  
20 I=0  
30 I=I+1  
40 PRINT "ALMA"  
50 PRINT  
60 IF I<20 THEN GOTO 30  
70 END
```

103. Vegyük észre:

- Mivel 100 szám összegéről van szó, a ciklusmag, vagyis az összeadást 100-szor hajtja végre a gép.
- Ha a ciklusváltozó értékét 1-től 100-ig egyesével változtatjuk, akkor annak aktuális értéke megegyezik azzal a számmal, amit a részletösszeghez kell adnunk.



A folyamatábra:



A program:

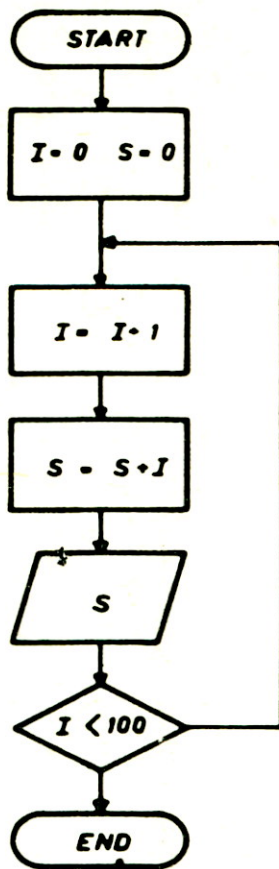
```
10 PRINT " "
20 PRINT TAB(8); "ELSO SZAZ SZAM OSSZEGE "
30 PRINT
40 I=0
45 S=0
50 I=I+1
60 S=S+1
70 IF I<100 THEN GOTO 50
80 PRINT "AZ ELSO SZAZ SZAM OSSZEGE " ; S
90 END
```

a)

Vegyük észre:

- Változtatnunk csak annyit kell, hogy a kiíró utasítást a ciklusmagba helyezzük, mert most a kiíratásnak minden egyes részletösszeg számítás után meg kell jelennie.

A folyamatábra:



A program:

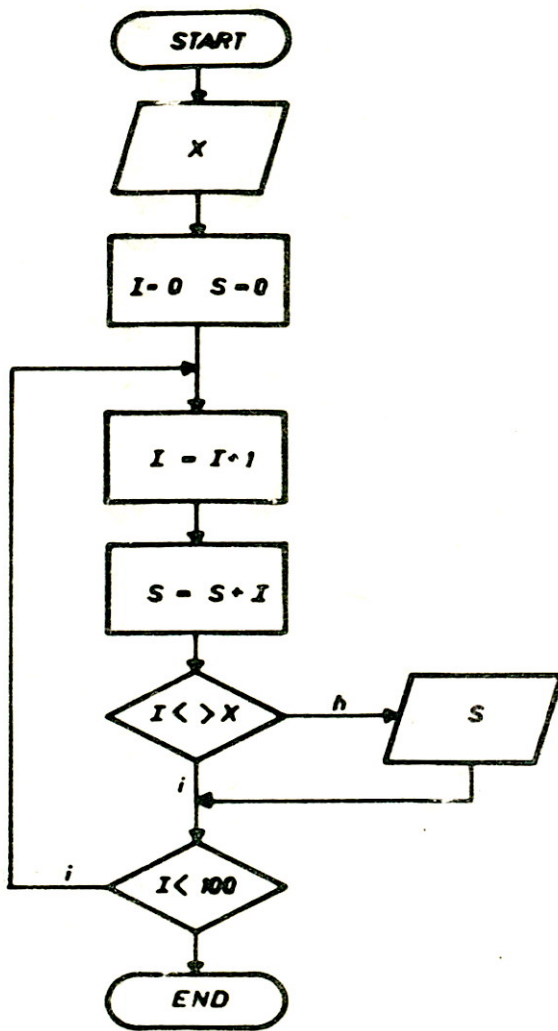
```
10 PRINT " "
20 PRINT TAB(8); "ELSO SZAZ SZAM OSSZEGE "
30 PRINT
40 I=0
50 S=0
60 I=I+1
70 S=S+I
80 PRINT S
90 IF I<100 THEN GOTO 60
100 END
```

**b)**

Vegyük észre:

– A kiíratást feltételhez kell kötnünk.

A folyamatábra:



A program:

```
10 PRINT " "
20 PRINT TAB(8); "ELSO SZAZ SZAM OSSZEGE "
30 PRINT
40 INPUT " HANYADIK RESZLETOSZEGET KERI "; X
50 I = 0
60 S = 0
70 I = I + 1
80 S = S + I
90 IF I <> X THEN GOTO 110
100 PRINT X; ". RESZLETOSZEG "; S
110 IF I < 100 THEN GOTO 70
120 END
```



**104. Vegyük észre:**

- Célszerű úgy szervezni a ciklust, hogy a ciklusváltozó adja meg azt a számot, amelynek éppen a négyzetgyökét akarom számolni.
- A szép táblázat érdekében a PRINT TAB(X) használata célszerű.

A program:

```
10 PRINT "□"  
20 PRINT TAB(4); "ELSO HUSZ SZAM NEGYZETGYOKE"  
30 PRINT  
40 PRINT "A SZAM":  
50 PRINT TAB(20); "A NEGYZETGYOKE"  
60 I=0  
70 I=I+1  
80 N=SQR(I)  
90 PRINT I;  
100 PRINT TAB(20);N  
110 IF I<20 THEN GOTO 70  
120 END
```

**105. Vegyük észre:**

- Szervezhetjük úgy a programot, hogy a ciklusváltozó aktuális értéke egyben a hatványkitevőnek is megfeleljen.
- Táblázatot készítünk, ezért célszerű a PRINT TAB(X) használata.
- A kiíratás a ciklusmagba kerül.

A program:

```
10 PRINT "□"  
20 PRINT TAB(10); "HATVANYOK"  
30 PRINT  
40 INPUT "A SZAM":X  
50 PRINT "KITEVO"  
60 PRINT TAB(20); "HATVANYAI"  
70 I=0  
80 I=I+1  
90 H=X↑I  
100 PRINT I;  
110 PRINT TAB(20);H  
120 IF I<10 THEN GOTO 80  
130 END
```

**106. Vegyük észre:**

- A kamat a bentlevő összeg  $3\%$ -a.
- A következő évben a bentlevő pénzhez a kamat is hozzájárul. Így a kamat és a végösszeg számítása a ciklusmagba kerül.

A program:

```
10 PRINT "Q"
20 PRINT TAB(11); "TAKAREKBETET"
30 PRINT
40 INPUT "AZ OSSZEG"; P
50 PRINT "A PENZ EVENTE 3%-OT KAMATOZIK"
60 INPUT "HANY EVIG VOLT A TAKAREKBAN"; E
70 I=0
80 I=I+1
90 K=P*0.03
100 P=P+K
110 IF I<E THEN GOTO 80
120 PRINT E; "EV UTAN A TAKAREKBOL KIVEHETO OSSZEG"
130 PRINT P; "FT"
140 END
```

107. Vegyük észre:

- A pontszámot öttel osztva és annak egészrészét véve az érdemjegnél eggyel kisebb számot kapok.
- Érdemes a jegyzet számoló ciklusváltozóját úgy meghatározni, hogy az mindig a felső ponthatárokat adja.

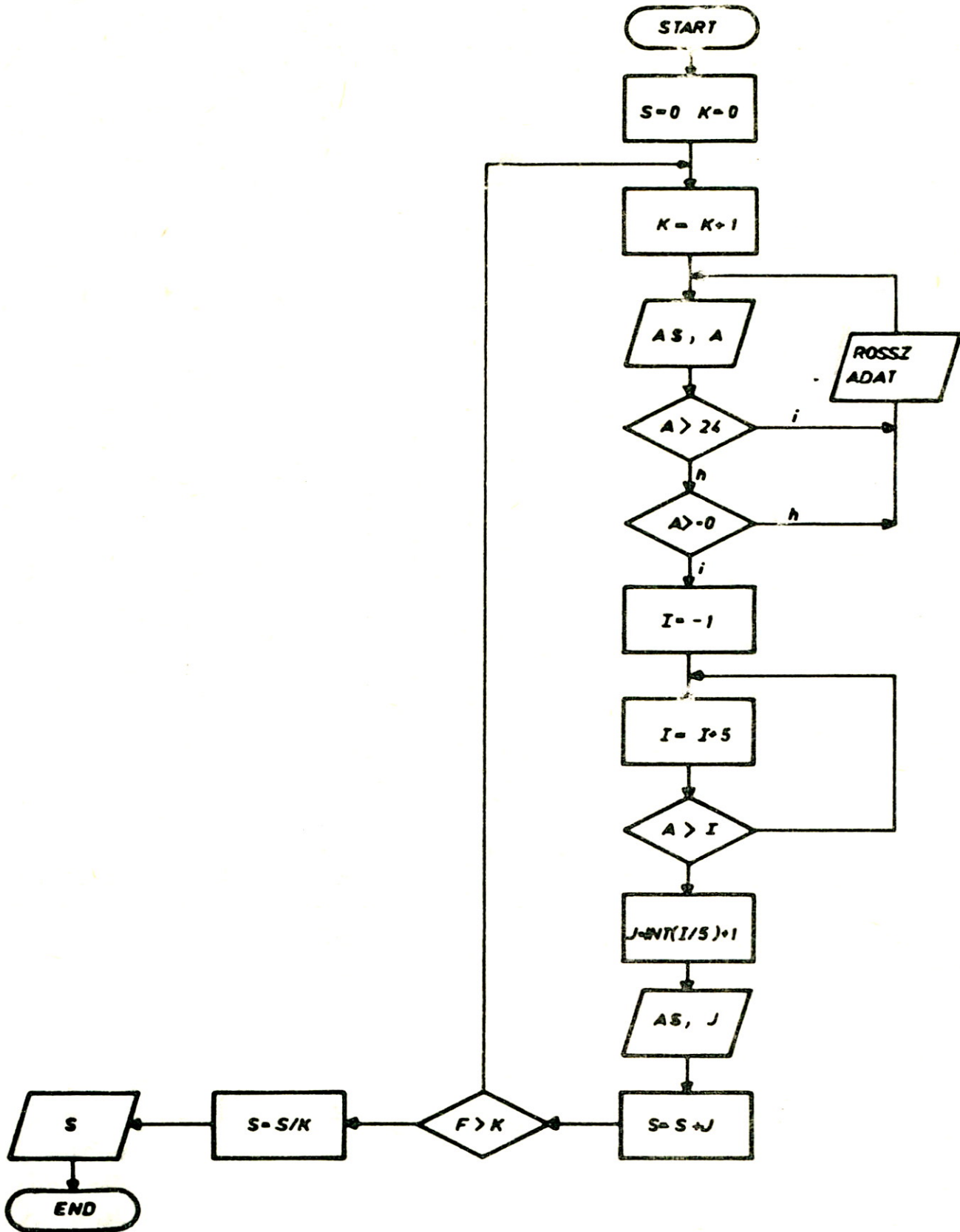
A program:

```
10 PRINT "Q"
20 PRINT TAB(8); "DOLGOZATOK ERTEKELESE"
30 PRINT
40 INPUT "A TANULO NEVE"; A#
50 INPUT "A TANULO PONTJAI"; A
60 IF A>24 THEN GOTO 80
70 IF A<=0 THEN GOTO 100
80 PRINT "ROSSZ ADAT"
90 GOTO 50
100 I=-1
110 I=I+1
120 IF A>I THEN GOTO 110
130 J=INT(I/5)+1
140 PRINT A#; "ERDEMJEGE"; J
150 END
```

108. Vegyük észre:

- Az eredeti program köré egy ciklust kell szervezni, ami számolja a futásokat. Mivel az eredeti is tartalmazott ciklust, így egymásba skatulyázott ciklusokat kapunk.
- Az osztályátlag számítását célszerű mindkét cikluson kívülre tenni.

A folyamatábra:





A program:

```
10 PRINT "□"
20 PRINT TAB(8), "DOLGOZATOK ERTEKELESE" ; F
30 PRINT
40 INPUT "HANYSZOR FUSSON A PROGRAM" ; F
50 S=0
60 K=0
70 K=K+1
80 PRINT
90 INPUT "A TANULO NEVE" ; A$
100 INPUT "A TANULO PONTJAI" ; A
110 IF A>24 THEN 130
120 IF A>=0 THEN 150
130 PRINT "ROSZ ADAT"
140 GOTO 100
150 I=-1
160 I=I+5
170 IF A>I THEN 160
180 J=INT(I/5)+1
190 PRINT
200 PRINT A$ ; " ERDEMJEGBE" ; J
210 S=S+J
220 IF F>K THEN 70
230 S=S/K
240 PRINT
250 PRINT "A TANULOCSOPORT ATLAGA " ; S
260 END
```

109. Vegyük észre:

- A ciklusváltozót célszerű úgy felvenni, hogy annak aktuális értéke egyben az oldalak hosszát is megadja. A ciklus addig fusson, amíg a számított területérték el nem éri, ill. meg nem haladja az előre kikötöttet.

A program:

```
10 PRINT "□"
20 PRINT TAB(15) ; "NEGYZETEK"
30 PRINT
40 INPUT "MEKKORA TERULETERTEKIG DOLGOZZUNK (CM2)" ; A
50 PRINT "OLDAL (CM)" ;
60 PRINT TAB(12) ; "KERULET (CM)" ;
70 PRINT TAB(25) ; "TERULET (CM2)" ;
80 I=0
90 I=I+1
100 T=INT(I2)
110 K=4*I
```

```

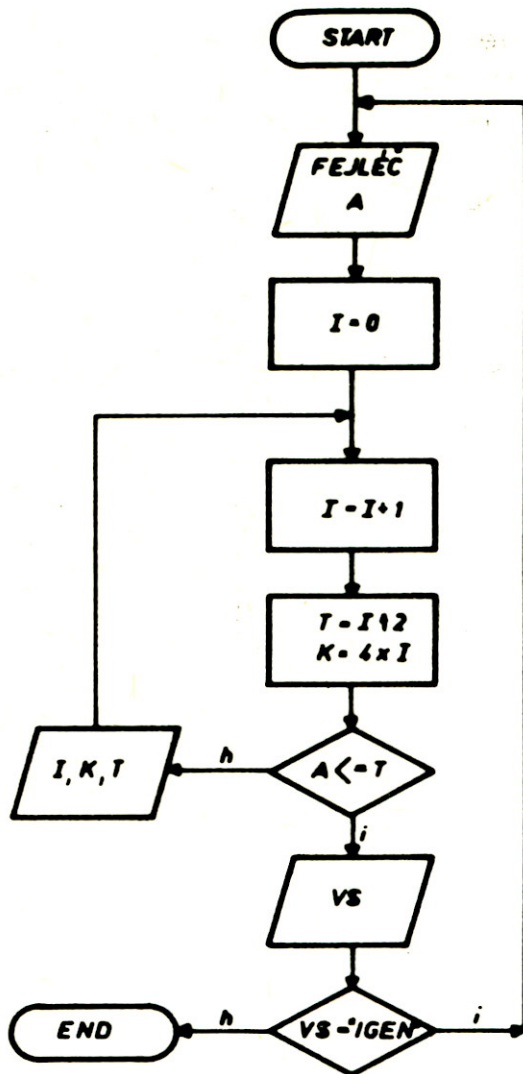
120 IF A<=T THEN 150
130 PRINT I;TAB(12);K;TAB(25);T
140 GOTO 90
150 END

```

**110. Vegyük észre:**

- Az előző programot úgy is bővíthetjük, hogy a végére egy kérdést teszünk. A kérdésre adott választól függően a program újra indul vagy megáll.

A folyamatábra:



A program:

```

10 PRINT "□"
20 PRINT TAB(15); "NEGYZETEK"
30 PRINT
40 INPUT "MEKKORA TERULETERTEKIG DOLGOZZUNK (CM^2) "; A
50 INPUT "DOLGAOZZUNK MEG", V
60 PRINT TAB(12); "KERULET (CM) ";
70 PRINT TAB(25); "TERULET (CM^2) "
80 I=0

```

```

90 I=I+1
100 T=INT(I↑2)
110 K=4*I
120 IF A<=T THEN 150
130 PRINT I;TAB(12);K;TAB(25);T
140 GOTO 90
150 END
160 IF V$="IGEN" THEN 10
170 END

```

### 111. Vegyük észre:

– A feladat két, elvében különböző módon is megoldható.

1. Minden percben pont az előző mennyiség kétszerese található a lombikban. Így egy cikluson belül az előre meghatározott értékig, a kért érték meghatározható.

2. Figyeljük meg az idő és a sejtszám kapcsolatát!

Idő:           1 2 3 4 5 6 7

Sejtszám:     1 2 4 8 16 32 64

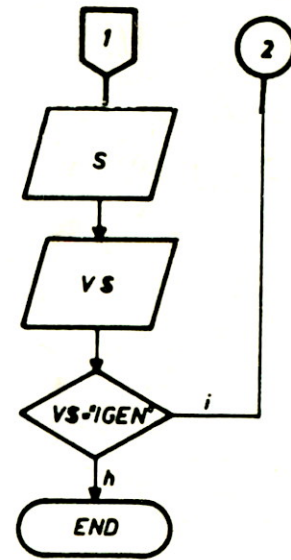
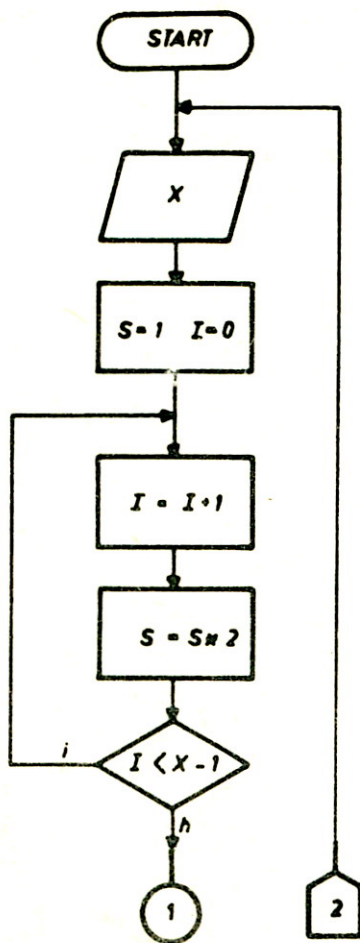
Ez pont:     2↑0 2↑1 2↑2 2↑3 2↑4 2↑5 2↑6-nak

felel meg. Ezt az összefüggést felfedezve a feladat ciklus nélkül is megoldható.



A folyamatábra:

1.



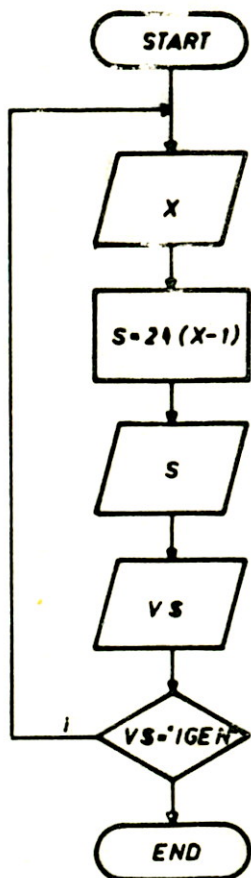
A program:

1.

```

10 PRINT " "
20 PRINT TAB(8); "SZAPORODO EGYSEJTUEK"
30 PRINT
40 PRINT "AZ EGYSEJTUEK SZAMA AZ ELSO PILLANAT-    BAN EGY."
50 PRINT "SZAMUK PERCENKENT MEGKETSZEREZODIK."
60 INPUT "HANYADIK PERCBEN KIVANCSI A SZAMUKRA"; X
70 S=1
80 I=0
90 I=I+1
100 S=S*2
110 IF I<X-1 THEN 90
120 PRINT X; ". PERCBEN AZ EGYSEJTUEK SZAMA "; S
130 INPUT "DOLGOZZUNK MEG"; V$
140 IF V$="IGEN" THEN 10
150 END
  
```

A folyamatábra:  
2.



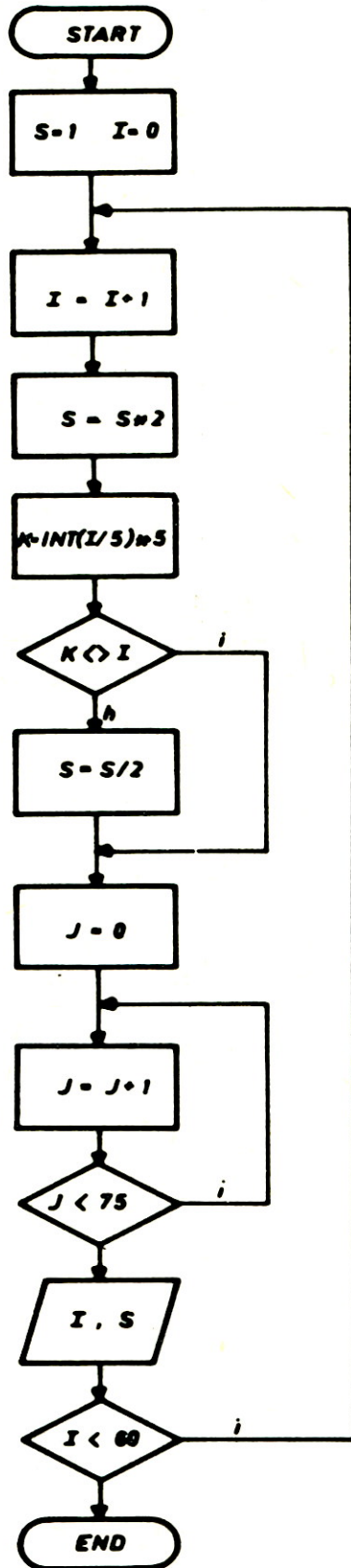
A program:  
2.

```
10 PRINT "Q"
20 PRINT TAB(8) / "SZAPORODO EGYSEJTUEK"
30 PRINT
40 PRINT "AZ EGYSEJTUEK SZAMA AZ ELSO PILLANAT-      BAN EGY."
50 PRINT "SZAMUK PERCENKENT MEGKETSZEREZODIK."
60 INPUT "HANYADIK PERCBEN KIVANCSI A SZAMUKRA" / X
70 S = 2^X (X-1)
80 PRINT X / ". PERCBEN AZ EGYSEJTUEK SZAMA " / S
90 INPUT "DOLGOZZUNK MEG" / V#
100 IF V# = "IGEN" THEN 10
110 END
```

112. Vegyük észre:

- A 111. feladat 1-es megoldását bővíthetjük tovább.
- Egy szám osztható egy másik számmal, ha a hányadosuk egész szám.
- Figyelni kell, hogy a ciklusváltozó, amely az időt figyeli, mikor osztható öttel (90–100. sor).
- A program a kiíratást olyan gyorsan végzi, hogy az nem olvasható. A lassítás egy ciklusmag nélküli ciklus beiktatásával érhető el (120–140. sor).

A folyamatábra:





## A program:

```
10 PRINT "▼"  
20 PRINT TAB (8); "SZAPOROD EGYSEJTUEK "  
30 PRINT  
40 PRINT " EGYSEJTUEK SZAMA AZ ELSO EGY ORABAN"  
50 S=1  
60 I=0  
70 I=I+1  
80 S=S*2  
90 K=INT(I/5)*5  
100 IF K<>I THEN GOTO 120  
110 S=S/2  
120 J=0  
130 J=J+1  
140 IF J<75 THEN GOTO 130  
150 PRINT I; ".PERC"; TAB(15); S  
160 IF I<60 THEN GOTO 70  
170 END
```

### 3.7 A tömb

– DIM A (100)      DIM A (100,25)      DIM A (10,2,4)

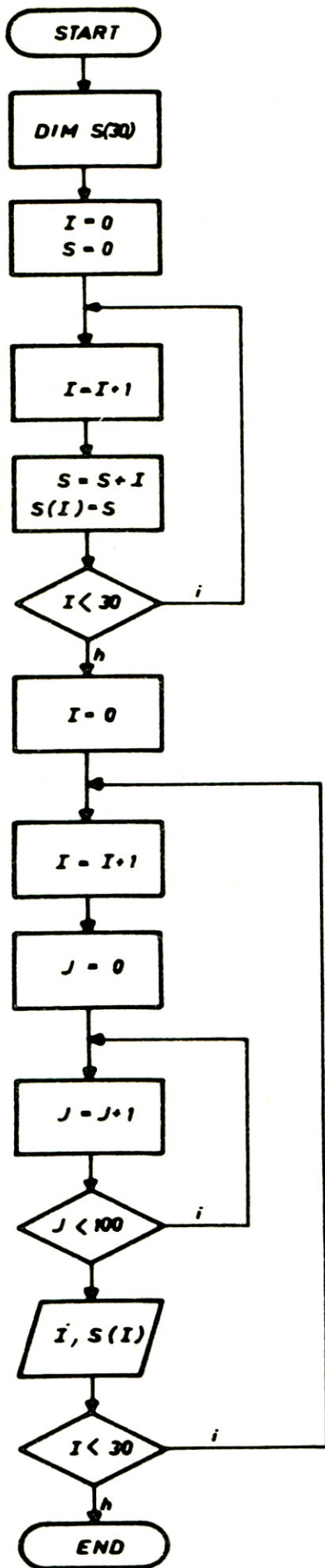
A tömb nem más, mint olyan elemek halmaza, amelyekről tudom, hogy hányadik helyen melyik elem áll. Ha 10-nél több elemmel dolgozunk, a tömb számára helyet kell foglalnunk a memóriában.

- A tömb lehet egydimenziós (ekkor egy egyenes mentén helyezem el a számokat), kétdimenziós (ekkor egy síkbeli koordináta-rendszerben helyezem el a számokat), ill. lehet többdimenziós is.
- A dimenzionáláson a program csak egyszer haladhat át.

#### 113. Vegyük észre:

- Mivel a részletösszegeket utólag kell kiíratni, ezért azokat tömbben gyűjtjük.
- Az olvashatóság miatt célszerű a kiíratást lassítani (130...150. sor).

A folyamatábra:



A program:

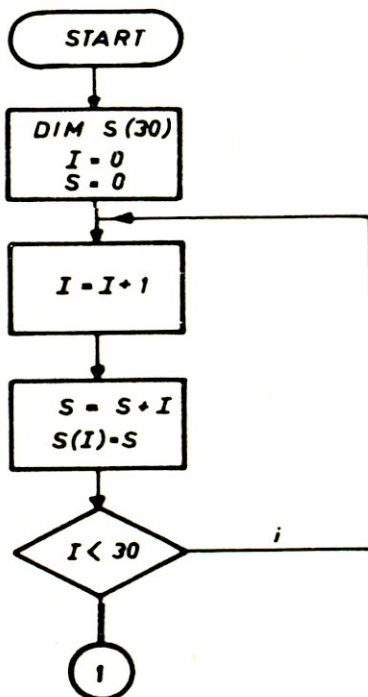
```
10 PRINT"¶"  
20 DIM S(30)  
30 PRINT TAB(7);"ELSO HARMINC POZITIV SZAM OSSZEGE"  
40 PRINT  
50 I=0  
60 S=0  
70 I=I+1  
80 S=S+I  
90 S(I)=S  
100 IF I<30 THEN GOTO 70  
110 I=0  
120 I=I+1  
130 J=0  
140 J=J+1  
150 IF J<100 THEN GOTO 140  
160 PRINT I;" . RESZLEGOSSZEG";S(I)  
170 IF I<30 THEN GOTO 120  
180 END
```

a)

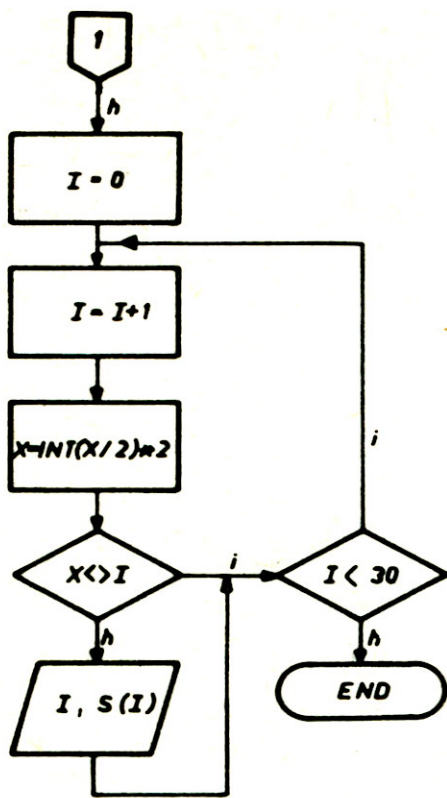
Vegyük észre:

- Minden második természetes szám páros.
- Minden páros szám osztható kettővel.

A folyamatábra:







A program:

```

10 PRINT " "
20 DIM S(30)
30 PRINT TAB (7); "ELSO HARMINC POZITIV SZAM OSSZEGE"
40 PRINT
50 I=0
60 S=0
70 I=I+1
80 S=S+1
90 S(I)=S
100 IF I<30 THEN GOTO 70
110 I=0
120 I=I+1
130 X=INT(I/2)*2
140 IF X<>I THEN GOTO 160
150 PRINT I; ".RESZLEG OSSZEG"; S(I)
160 IF I<30 THEN GOTO 120
170 END
  
```

b)

Vegyük észre:

- 1, 3, 5, ... páratlan számok.
- A páratlan számok nem oszthatók kettővel.
- 30 db szám közül a fele páros, a fele páratlan. Ennyi szám elfér egymás alatt a képernyőn, így mint az előző esetben, nem szükséges a lassítás.
- A program szerkezetében hasonló az a) feladathoz, csak a kérdésfeltevése más.

A program:

```
10 PRINT "A"  
20 DIM S(30)  
30 PRINT TAB(7); "ELSO 30 POZITIV SZAM OSSZEGE"  
40 PRINT  
50 I=0  
60 S=0  
70 I=I+1  
80 S=S+I  
90 S(I)=S  
100 IF I<30 THEN GOTO 70  
110 I=0  
120 I=I+1  
130 X=INT(I/2)*2  
140 IF X=1 THEN GOTO 160  
150 PRINT I; ".RESZOSSZEG"; S(I)  
160 IF I<30 THEN GOTO 120  
170 END
```

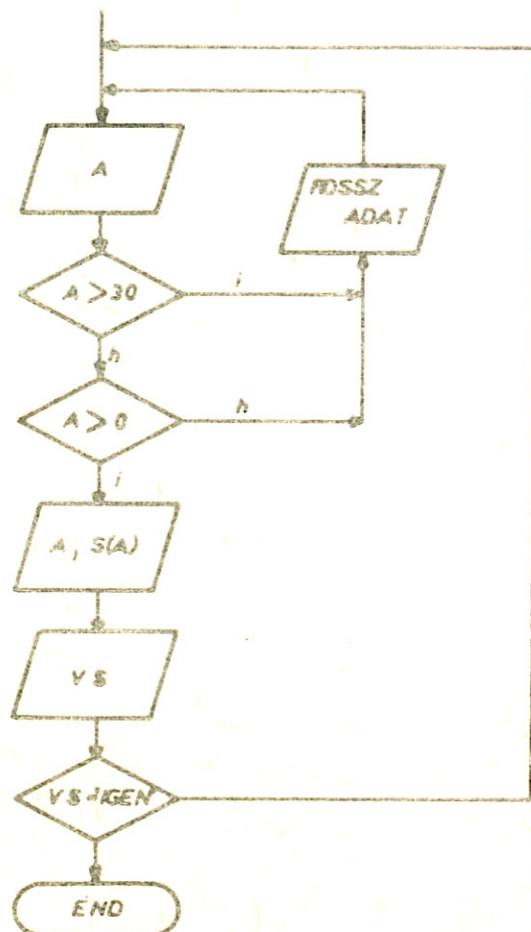
c)

Vegyük észre:

- Az index segítségével a tömb bármely eleme kiolvasható.
- Hibás adat, ha olyan indexű elemet kerestünk, amely nem szerepel a tömbben.

A folyamatábra:

- A beolvasó rész megegyezik az előbbi programéval.



A program:

```
10 PRINT "□"
20 DIM S(30)
30 PRINT TAB(7); "ELSO 30 POZITIV SZAM OSSZEGE"
40 PRINT
50 I=0
60 S=0
70 I=I+1
80 S=S+I
90 S(I)=S
100 IF I<30 THEN GOTO 70
110 INPUT "HANYADIK ELEMET KERI" ; A
120 IF A>30 THEN GOTO 140
130 IF A>0 THEN GOTO 160
140 PRINT "ROSSZ ADAT"
150 GOTO 110
160 PRINT A ; ". ELEM" ; S(A)
170 INPUT "DOLGOZUNK MEG" ; V#
180 IF V#="IGEN" THEN GOTO 110
190 END
```

**114. Vegyük észre:**

– A feladat szerkezetében hasonló a 113. c) feladathoz. Ezért folyamatábrát nem közlünk.

A program:

```
10 PRINT "□"
15 DIM N(100)
20 PRINT TAB(4); "ELSO 100 POZITIV SZAM NEGYZETE"
30 PRINT
40 I=0
50 N=0
60 I=I+1
70 N=I*I
80 N(I)=N
90 IF I<100 THEN GOTO 60
100 INPUT "MELYIK SZAM NEGYZETET KERI" ; X
110 IF X>100 THEN GOTO 130
120 IF X>0 THEN GOTO 150
130 PRINT "ROSSZ ADAT"
140 GOTO 100
150 PRINT X ; " NEGYZETE: " ; N(X)
160 INPUT "DOLGOZUNK MEG" ; V#
170 IF V#="IGEN" THEN GOTO 100
180 END
```



**115. Vegyük észre:**

- Valós és sztring tömböt kell használni.
- Minden tömbnek van nulladik eleme is. Ha a program sajátosságai ezt nem akadályozzák, célszerű ezt kihasználni.

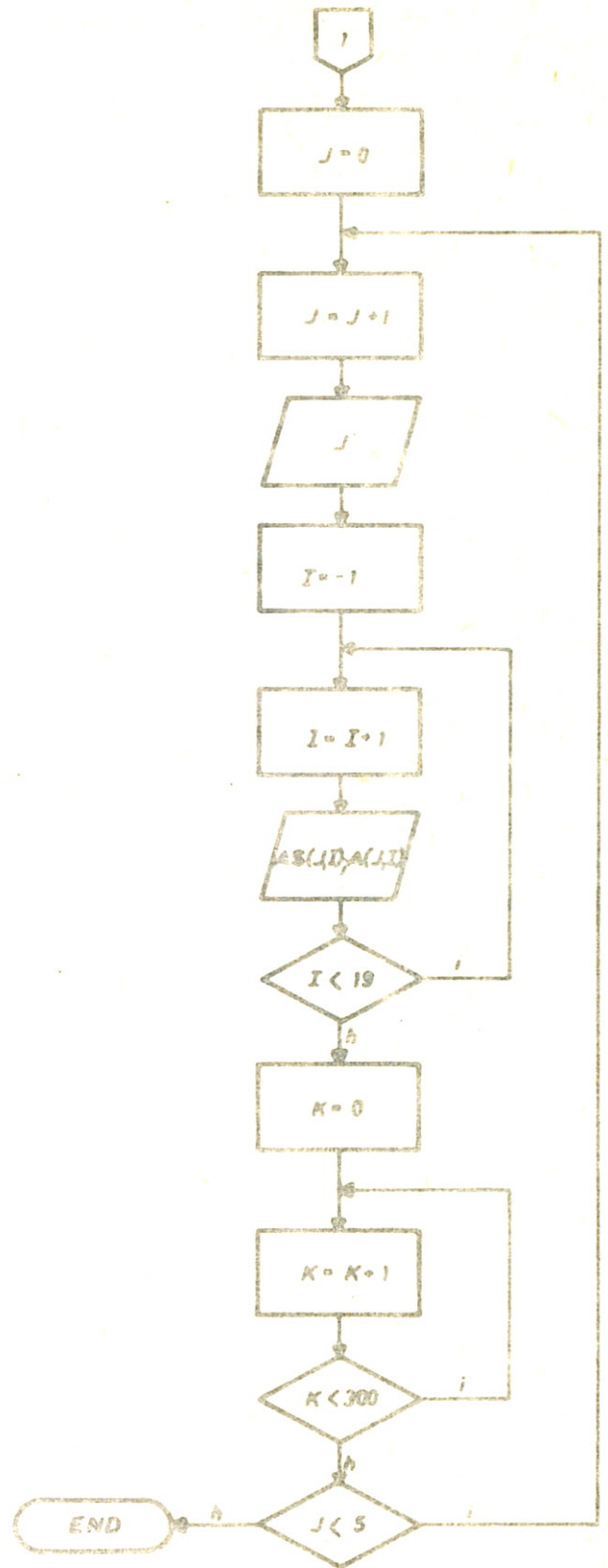
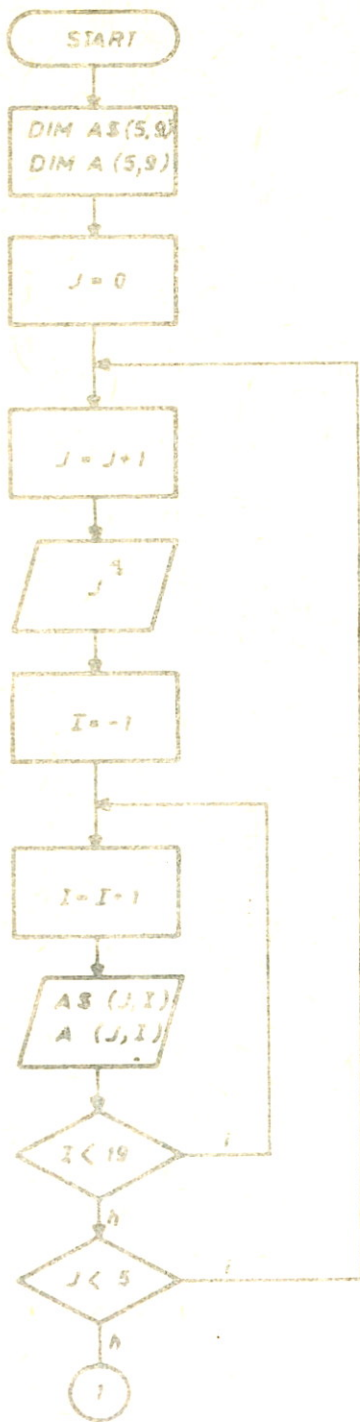
A program:

```
10 PRINT "☐"
20 DIM A$(19)
30 DIM A(19)
40 PRINT TAB(10); "ZOLDSEGLELTAR☐"
50 I=-1
60 I=I+1
70 INPUT "ZOLDSEG NEVE"; A$(I)
80 INPUT "MENNYISEGE"; A(I)
90 IF I<19 THEN GOTO 60
100 PRINT "☐"
110 PRINT TAB(13); "LELTARIV"
120 PRINT "☐ZOLDSEG"; TAB(20); "MENNYISEG (KG)☐"
130 I=-1
140 I=I+1
150 PRINT A$(I); TAB(20); A(I)
160 IF I<19 THEN GOTO 140
170 END
```

**116. Vegyük észre:**

- Nagyon jól használható a feladat megoldására a kétdimenziós tömb.
- Az első tömbindex a napot, a második a zöldséget jelöli.
- A leltárívek olyan gyorsan követhetik csak egymást, hogy azok olvashatók legyenek.

A folyamatábra:



A program:

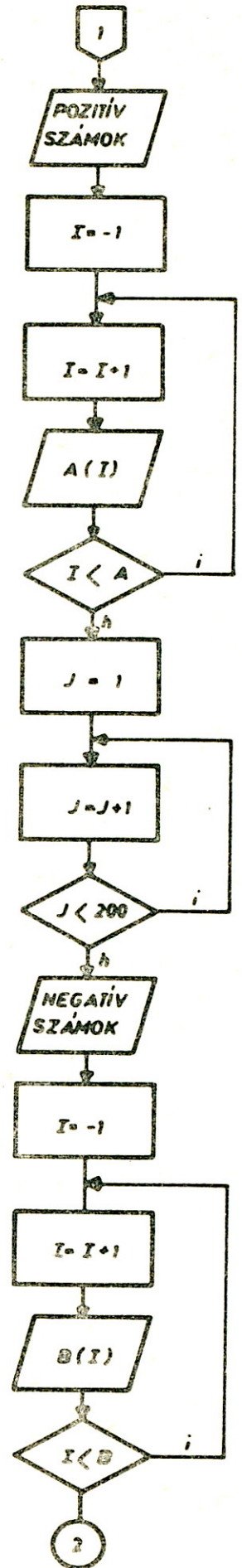
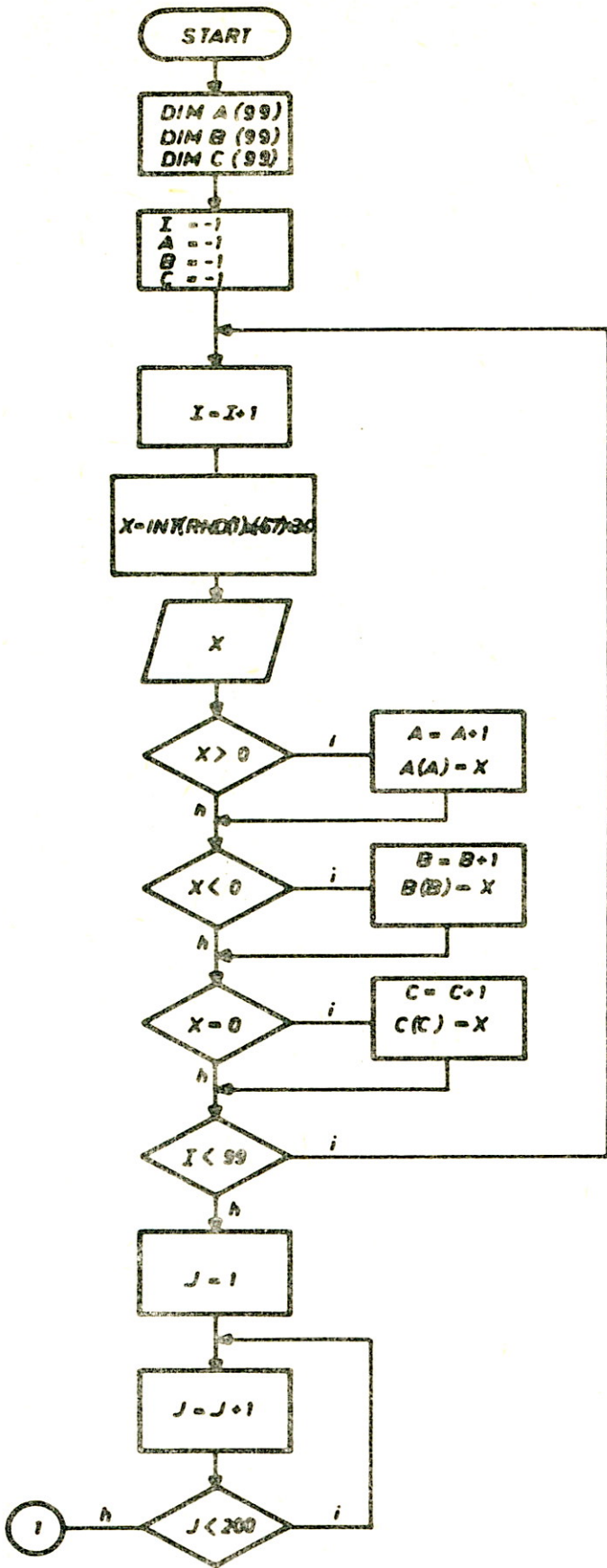
```
10 PRINT " "
20 DIM A$(5,19)
30 DIM A(5,19)
40 J=0
50 J=J+1
60 PRINT " "
70 PRINT TAB(10); "ZOLDOSEGLELTAR"/J/". NAP "
80 I=-1
90 I=I+1
100 INPUT "ZOLDOSEG NEVE"/A$(J,I)
110 INPUT "MENNYISEGE"/A(J,I)
120 IF I<19 THEN GOTO 90
130 IF J<5 THEN GOTO 50
140 J=0
150 J=J+1
160 PRINT " "
170 PRINT TAB(13); "LELTARIV"/J/". NAP "
180 PRINT "ZOLDOSEG"/TAB(20); "MENNYISEG (KG) "
190 I=-1
200 I=I+1
210 PRINT A$(J,I);TAB(25);A(J,I)
220 IF I<19 THEN GOTO 200
230 K=0
240 K=K+1
250 IF K<300 THEN GOTO 240
260 IF J<5 THEN GOTO 150
```

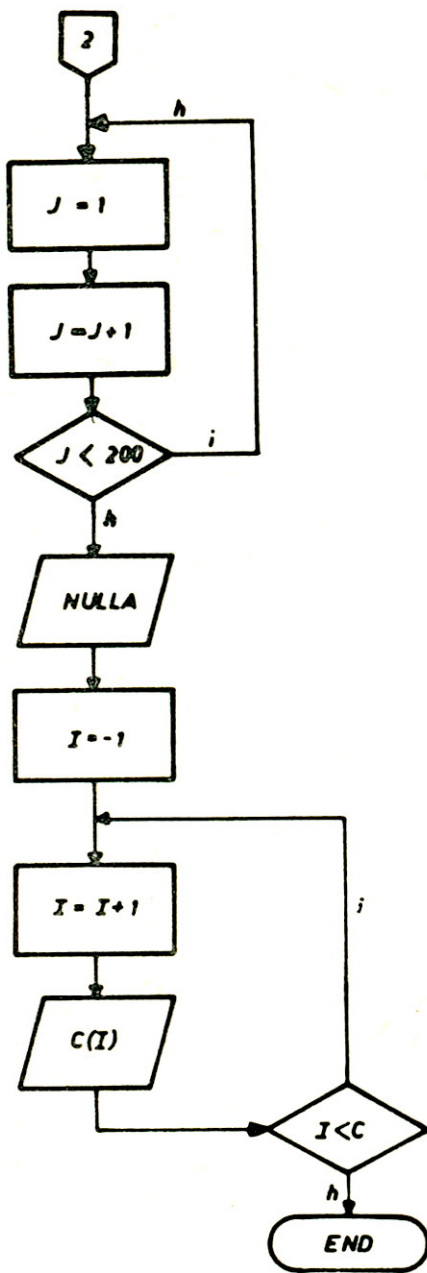
#### 117. Vegyük észre:

- Egy szám előjelét tekintve lehet pozitív, negatív vagy nulla. Ezért három tömböt kell felvenni.
- Az index nem egyezik a mindenkorai ciklusváltozóval, mint ahogy az előzőekben láttuk (130... 150. sor).
- Jobban követhető a kiíratás, ha azt lassítjuk. A lassítást a J ciklusváltozójú ciklusok végzik.



A folyamatábra:





A program:

```

10 PRINT " "
20 PRINT TAB(4); "VELETLENSZAMOK CSOPORTOSITASA"
30 DIM A(99)
40 DIM B(99)
50 DIM C(99)
60 I = -1
70 A = -1
80 B = -1
90 C = -1
100 I = I + 1
110 X = INT(RND(1) * (-67)) + 34
120 PRINT X;
130 IF X > 0 THEN A = A + 1 : A(A) = X
140 IF X < 0 THEN B = B + 1 : B(B) = X
  
```

```

150 IF X=0 THEN C=C+1:C(C)=X
160 IF I<99 THEN GOTO 100
170 J=1
180 J=J+1
190 IF J<200 THEN GOTO 180
200 PRINT " "
210 PRINT "POZITIV SZAMOK:"
220 I=-1
230 I=I+1
240 PRINT A(I);
250 IF I<A THEN GOTO 230
260 J=1
270 J=J+1
280 IF J<200 THEN GOTO 270
290 PRINT
300 PRINT "NEGATIV SZAMOK:"
310 I=-1
320 I=I+1
330 PRINT B(I);
340 IF I<B THEN GOTO 320
350 J=1
360 J=J+1
370 IF J<200 THEN GOTO 360
380 PRINT
390 PRINT "0 NULLA:"
400 I=-1
410 I=I+1
420 PRINT C(I);
430 IF I<C THEN GOTO 410
440 END

```

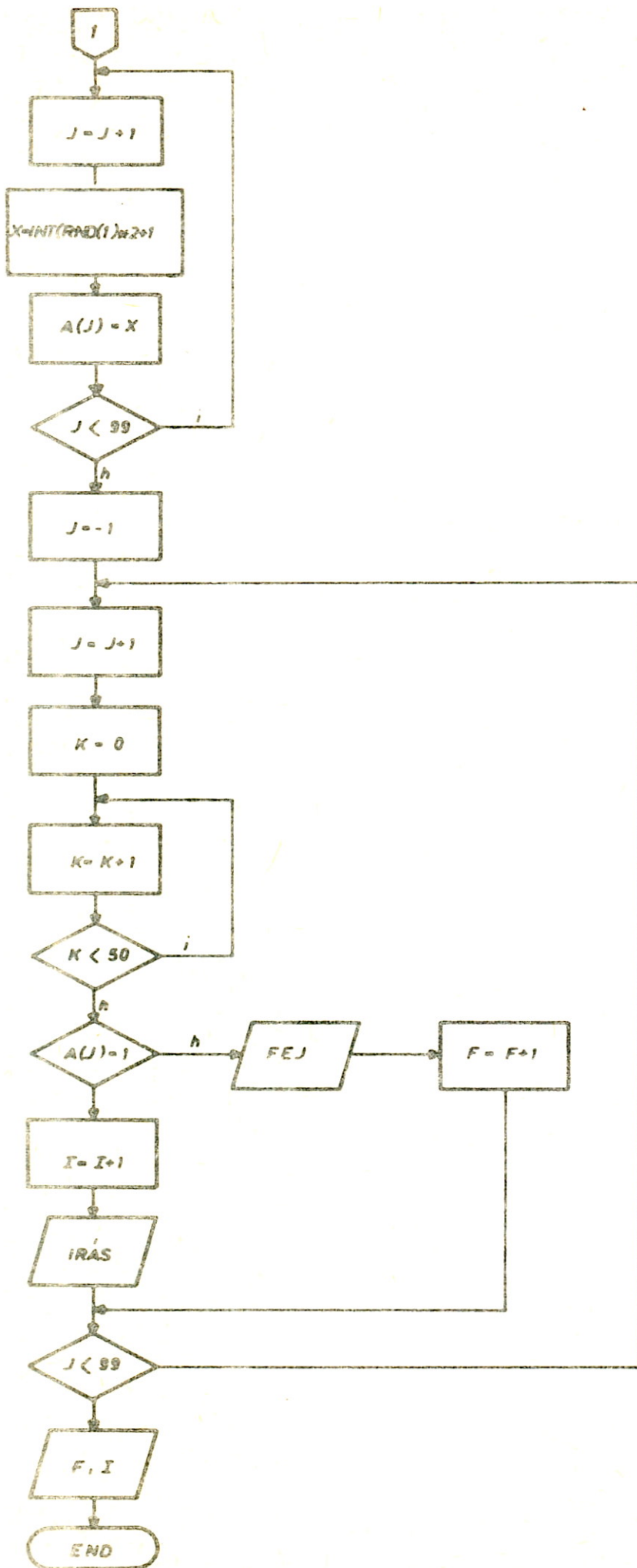
118. Vegyük észre:

– Ha véletlenszámokat generálok 1-től 2-ig, akkor az egyik megfeleltethető a FEJ, a másik az ÍRÁS dobásnak.

A folyamatábra:







A program:

```
10 PRINT "□"
20 PRINT TAB(13); "PENZFELDOBAS□"
30 DIM A(99)
40 J=-1
50 J=J+1
60 X=INT(RND(1)*2)+1
70 A(J)=X
80 IF J<99 THEN GOTO 50
90 J=-1
100 J=J+1
110 K=0
120 K=K+1
130 IF K<50 THEN GOTO 120
140 IF A(J)=1 THEN GOTO 180
150 F=F+1
160 PRINT "FEJ"
170 GOTO 200
180 I=I+1
190 PRINT "IRAS"
200 IF J<99 THEN GOTO 100
210 PRINT
220 PRINT "100 DOBAS: "; F; "DB FEJ ES "; I; "DB IRAS VOLT"
230 END
```

119. Vegyük észre:

- Az előző programot mint alprogramot építhetjük be.
- Mivel ez játék, izgalmasabb, ha a dobások közé hatásszünetet teszünk (140...160...260...280. sor).
- Az eredményt is jelezni kell (370...450. sor).
- Adjunk lehetőséget a folyamatos újrajátszásra is (460...470. sor).

A program:

```
10 PRINT "□"
20 PRINT TAB(8); "PENZFELDOBOS JATEK□"
30 PRINT "□□□AZ ELSO HAROM DOBAS A TIED,"
40 PRINT "A MASIK HAROM AZ ENYEM"
50 PRINT "AZ IRAS 1 A FEJ 2 PONTOT ER."
60 K=0
70 K=K+1
80 IF K<100 THEN GOTO 70
90 PRINT "□□DOBASAID: □"
100 I=0
110 I=I+1
120 X=INT(RND(1)*2)+1
130 T=T+X
```

```

140 K=0
150 K=K+1
160 IF K<100 THEN GOTO 150
170 IF X=1 THEN GOTO 200
180 PRINT "FEJ",
190 GOTO 210
200 PRINT "IRAS",
210 IF I<3 THEN GOTO 110
220 PRINT
230 PRINT "AZ EN DOBASAIM: "
240 I=0
250 I=I+1
260 K=0
270 K=K+1
280 IF K<100 THEN GOTO 270
290 X=INT(RND(1)*2)+1
300 G=G+X
310 IF X=1 THEN GOTO 340
320 PRINT "FEJ",
330 GOTO 350
340 PRINT "IRAS",
350 IF I<3 THEN GOTO 250
360 PRINT
370 PRINT "T PONTJAID: " ; T,
380 PRINT "AZ EN PONTJAIM: " ; G
390 IF T>G THEN GOTO 430
400 IF G>T THEN GOTO 450
410 PRINT "DONTETLEN"
420 GOTO 460
430 PRINT "NYERTEL"
440 GOTO 460
450 PRINT "EN NYERTEM"
460 INPUT "JATSZUNK MEG" ; V#
470 IF V#="IGEN" THEN GOTO 10
480 END

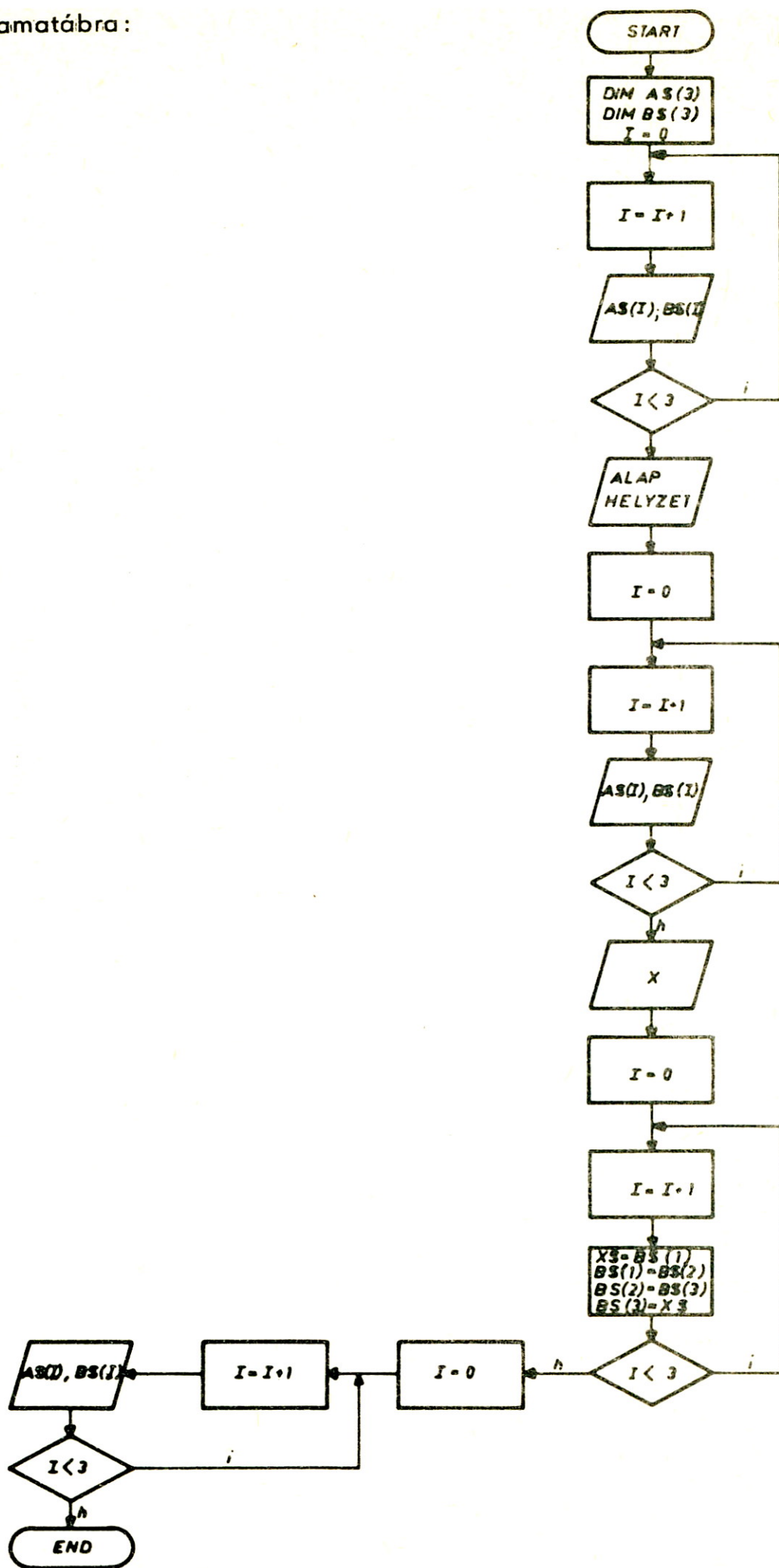
```

**120. Vegyük észre:**

- A neveket és gyümölcsöket tömbben is tárolhatjuk.
- A cserék mindig ugyanabban az irányban történnek, minder. gyümölcs csak egy hellyel kerül odébb.



A folyamatábra:



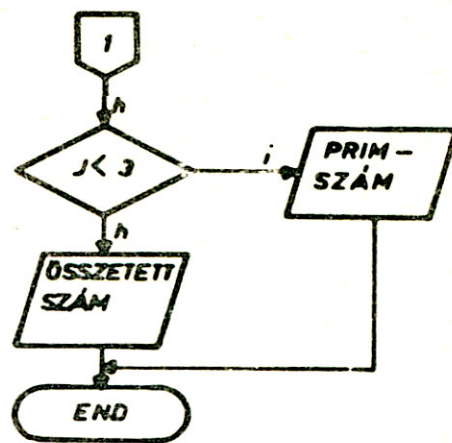
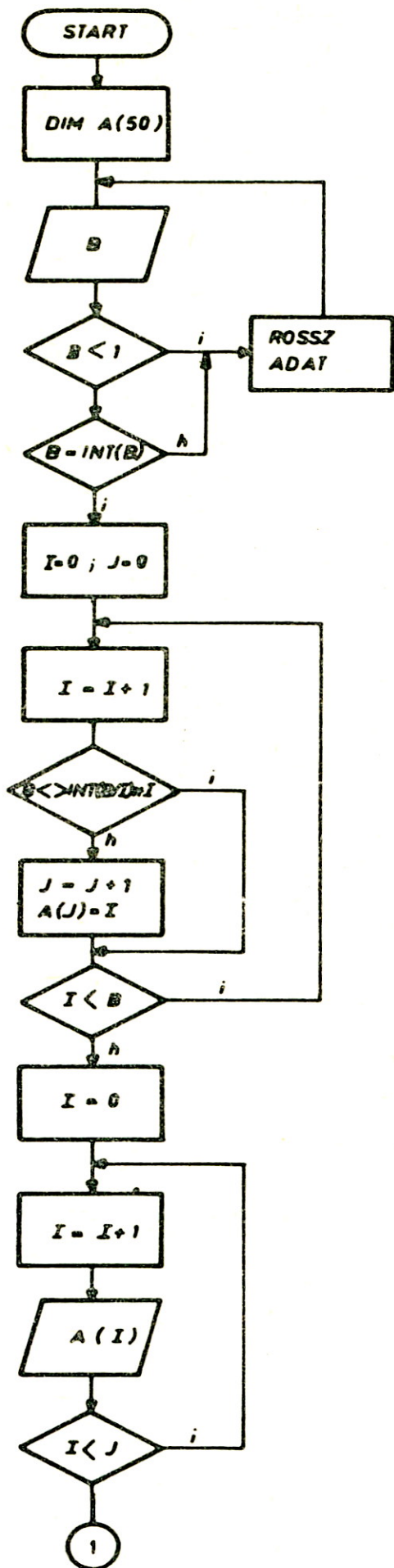
A program:

```
10 PRINT "1"
20 DIM A$(3)
30 DIM B$(3)
40 PRINT TAB(10); "GYUMOLCSADOGATO"
50 I=0
60 I=I+1
70 INPUT "A NEV";A$(I)
80 INPUT "A GYUMOLCS";B$(I)
90 IF I<3 THEN GOTO 60
100 PRINT "2"
110 PRINT "10. ALAPHELYZET: "
120 I=0
130 I=I+1
140 PRINT A$(I),B$(I)
150 IF I<3 THEN GOTO 130
160 INPUT "HANYADIK CSERE UTANI HELYZETRE KIVANCSI";X
170 I=0
180 I=I+1
190 X#=B$(I)
200 B$(1)=B$(2)
210 B$(2)=B$(3)
220 B$(3)=X#
230 IF I<X THEN GOTO 180
240 I=0
250 I=I+1
260 PRINT A$(I),B$(I)
270 IF I<3 THEN GOTO 250
280 END
```

121. Vegyük észre:

- Az oszthatóság csak pozitív egész számokra értelmezett.
- Ezért ki kell védeni a hibás adatokat.
- Az osztókat célszerű tömbben tárolni.
- Az osztók száma alapján eldönthető, hogy egy szám összetett vagy prím szám-e! Ezért célszerű az osztók számát figyelni.
- A program addig fut, amíg meg nem találjuk a legnagyobb osztót, ami maga a szám.

A folyamatábra:





## A program:

```
10 PRINT " "
20 PRINT "összettszoleges egész szám összes osztója"
30 DIM A(50)
40 INPUT "A szám":B
50 IF B<1 THEN GOTO 70
60 IF B=INT(B) THEN GOTO 80
70 PRINT "rossz adat"
80 GOTO 40
90 I=0
100 J=0
110 I=I+1
120 IF B<>INT(B/I)*I THEN GOTO 150
130 J=J+1
140 A(J)=I
150 IF I<B THEN GOTO 110
160 I=0
170 I=I+1
180 PRINT A(I),
190 IF I<J THEN GOTO 170
200 PRINT
210 IF J<3 THEN GOTO 240
220 PRINT "összetett szám"
230 GOTO 250
240 PRINT "primszám"
250 END
```

### 3.8 FOR...TO...STEP és NEXT ciklusutasítások

- Ciklusképző utasítás.
- FOR I=A B STEP C  
NEXT I  
A két programsor közötti rész annyiszor hajtódik végre, amíg I nem lesz nagyobb B-nél, C-vel növekedve.
- Mivel a ciklus futása akkor fejeződik be, amikor  $I=B$ , ezért a NEXT utáni sorban vizsgálva  $I=B+1$ .
- Ha  $C=1$ , a STEP C-t nem kell kiírni.
- Vigyázzunk arra, hogy nem minden IF...THEN-nel szervezett ciklus írható át FOR...NEXT ciklussá!

#### 122. Vegyük észre:

- A ciklusmagot 100-szor kell végrehajtani.
- STEP 1-et nem kell kiírni.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(13); "SZAMOK 1-100☐"  
30 FOR I=1 TO 100  
40 PRINT I;  
50 NEXT I  
60 END
```

a)

Vegyük észre:

– Az első kiírandó szám a 2, az utolsó a 100. Minden páros számot kell kiírni.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(13); "SZAMOK 1-100☐"  
30 FOR I=2 TO 100 STEP 2  
40 PRINT I;  
50 NEXT I  
60 END
```

b)

Vegyük észre:

– Az első kiírandó szám az 1., az utolsó a 99. Köztük minden páratlan számot ki kell írni.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(13); "SZAMOK 1-100☐"  
30 FOR I=1 TO 100 STEP 2  
40 PRINT I;  
50 NEXT I  
60 END
```

c)

Vegyük észre:

– Az első kiírandó szám a 100, az utolsó 1. Tehát visszafelé kell számolni.

A program:

```
10 PRINT "☐"  
20 PRINT TAB(13); "SZAMOK 1-100☐"  
30 FOR I=100 TO 1 STEP -1  
40 PRINT I;  
50 NEXT I  
60 END
```

**123. Vegyük észre:**

- A szorzatot számláló változó kezdőértéke 1.
- A kiíratás a cikluson kívülre kerül.

A program:

```
10 PRINT "□"  
20 PRINT TAB(3); "ELSO 30 POZITIV SZAM SZORZATA□"  
30 F=1  
40 FOR I=1 TO 30  
50 F=F*I  
60 NEXT I  
70 PRINT F
```

**a)**

**Vegyük észre:**

- A kiíratás a ciklusmagba kerül.

A program:

```
10 PRINT "□"  
20 PRINT TAB(3); "ELSO 30 POZITIV SZAM SZORZATA□"  
30 F=1  
40 FOR I=1 TO 30  
50 F=F*I  
60 PRINT F  
70 NEXT I
```

**b)**

**Vegyük észre:**

- Ha a STEP változtatásával dolgoznánk, akkor a gép csak minden második számot szorozná össze.
- Páros a szám, ha osztható kettővel.

A program:

```
10 PRINT "□"  
20 PRINT TAB(3); "ELSO 30 POZITIV SZAM SZORZATA□"  
30 F=1  
40 FOR I=1 TO 30  
50 F=F*I  
60 IF I(>)INT(I/2)*2 THEN GOTO 60  
70 PRINT I,F  
80 NEXT I
```

**c)**

**Vegyük észre:**

- Egy szám páratlan, ha nem osztható kettővel.



A program:

```
10 PRINT "□"  
20 PRINT TAB(3); "ELSO 30 POZITIV SZAM SZORZATA□"  
30 F=1  
40 FOR I=1 TO 30  
50 F=F*I  
60 IF I=INT(I/2)*2 THEN GOTO 80  
70 PRINT I,F  
80 NEXT I
```

d)

Vegyük észre:

– A ciklust csak addig az értékig kell futtatni, amit a használó kér.

A program:

```
10 PRINT "□"  
20 PRINT TAB(3); "ELSO 30 POZITIV SZAM SZORZATA□"  
30 INPUT "HANYADIK ELEMET KERI" ; A  
40 F=1  
50 FOR I=1 TO 30  
60 F=F*I  
70 NEXT I  
80 PRINT A,F  
90 END
```

124. Vegyük észre:

- Egy sor 40 karaktert tartalmaz.
- 23 sort tudok kiíratni a képernyőre.
- A feladat megoldható:
  - a) A ciklusmagba 40 csillagot kiírató utasítással.
  - b) A feladat végére END helyett végtelen ciklust írva, így a képernyőn nem jelenik meg a READY.
  - c) Két egymásba skatulyázott ciklussal. Ekkor, míg a külső ciklus egyet lép, a belső 40-szer lefut.

A program:

a)

```
10 PRINT "□"  
20 FOR I=1 TO 23  
30 PRINT "*****"  
40 NEXT I  
50 END
```

**b)**

```
10 PRINT "□"  
20 FOR I=1 TO 23  
30 PRINT "*****"  
40 NEXT I  
50 GOTO 50
```

**c)**

```
10 PRINT "□"  
20 FOR I=1 TO 23  
30 FOR J=1 TO 40  
40 PRINT "*";  
50 NEXT J  
60 NEXT I  
70 GOTO 70
```

**125. Vegyük észre:**

– A feladat megoldható

a) A ciklusmagba kiíró utasításba egy sornyi karaktert helyezek.

b) Két egymásba skatulyázott ciklust írva, amíg a külső ciklus egyszer fordul, addig a belső 20-szor.

A program:

**a)**

```
10 PRINT "□"  
20 FOR I=1 TO 23  
30 PRINT "* * * * *"  
40 NEXT I  
50 GOTO 50
```

**b)**

```
10 PRINT "□"  
20 FOR I=1 TO 23  
30 FOR J=1 TO 20  
40 PRINT "* ";  
50 NEXT J  
60 NEXT I  
70 GOTO 70
```

**126. Vegyük észre:**

- Ha az I értékét használom az átlag számítására, ügyelni kell arra, hogy  $I=101$ .
- A programot többször futtatva azt tapasztaljuk, hogy a véletlenszámok átlaga 50 körüli szám.

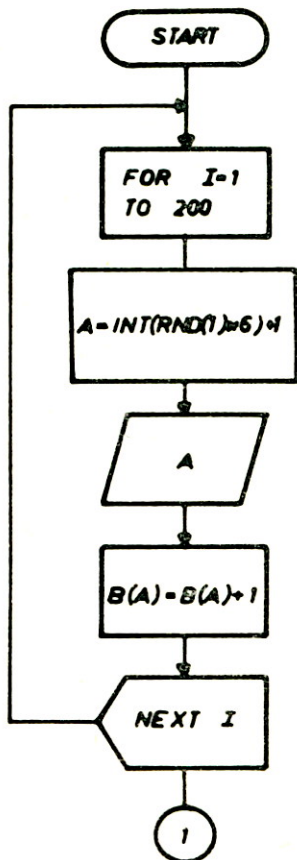
A program:

```
10 PRINT "I"
20 PRINT TAB(8); "VELETLENSZÁMOK ÁTLAGA (1-100)"
30 S=0
40 FOR I=1 TO 100
50 A=INT(RND(1)*100)+1
60 PRINT A;
70 S=S+A
80 NEXT I
90 S=S/(I-1)
100 PRINT
110 PRINT "A VELETLENSZÁMOK ÁTLAGA: "; S
120 END
```

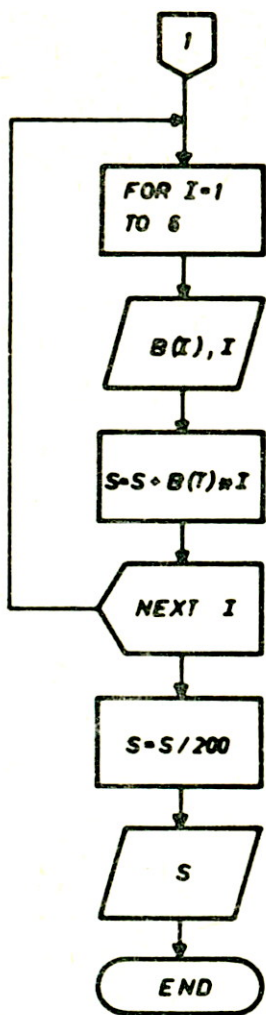
**127. Vegyük észre:**

- A gép 1-6 között generál egész számokat.
- Az első ciklus (40...80. sor) kiírja a számokat. Itt nem a generált számokat érdemes tömbben gyűjteni, hanem azt, hogy egy szám hányszor szerepelt.
- A második ciklus (90...130. sor) kiírja, hogy melyik pont hányszor szerepelt, és számolja a dobások összegét.
- Többször futtatva azt tapasztaljuk, hogy a dobások átlaga három körüli érték.

A folyamatábra:







A program:

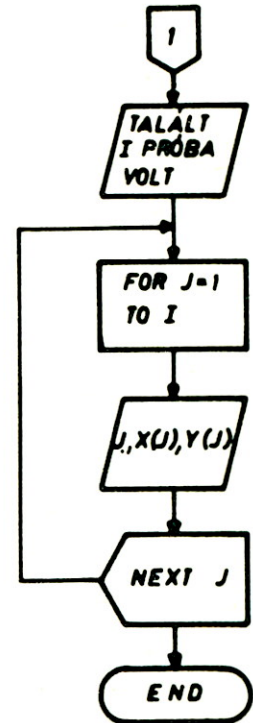
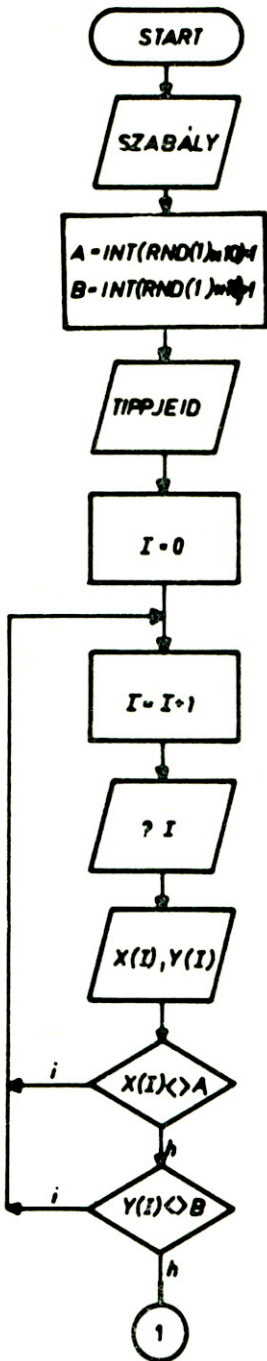
```

10 PRINT "1"
20 PRINT TAB(13); "KOCKADOBALO"
30 DIM B(6)
40 FOR I=1 TO 200
50 A=INT(RND(1)*6)+1
60 PRINT A;
70 B(A)=B(A)+1
80 NEXT I
90 FOR I=1 TO 6
100 PRINT
110 PRINT B(I); "DB"; I; "-S DOBAS VOLT"
120 S=S+B(I)*I
130 NEXT I
140 S=S/200
150 PRINT "BA DOBASOK ATLAGA"; S
160 END
  
```

128. Vegyük észre:

- A begépelte koordinátákat gyűjteni kell, hogy a játék végén kiírva a játékos ellenőrizhesse a lövéseit.
- A lövések számát jegyezni kell. Ezt az adatot a kiírató ciklus változójánál fel lehet használni (200-220).

A folyamatábra:



A program:

```

10 PRINT "□"
20 DIM X(100)
30 DIM Y(100)
40 PRINT TAB(14); "TORPEDO"
50 PRINT "A GEP EGY 10*10-ES NEGYZETEN ELREJTETT EGY HAJÓT"
60 PRINT "A HAJÓT KILÓNI A KOORDINÁTAK MEGADÁSA-VAL LEHET"
  
```

```

70 PRINT "A GEP SZAMOLJA , HANYADIK TIPPRE TALA-LOD EL."
80 A=INT(RND(1)*10)+1
90 B=INT(RND(1)*10)+1
100 PRINT "TIPPJEID:"
110 I=0
120 I=I+1
130 PRINT I;"."
140 INPUT "X KOORDINATA:";X(I)
150 INPUT "Y KOORDINATA:";Y(I)
160 IF X(I)<>A THEN GOTO 120
170 IF Y(I)<>B THEN GOTO 120
180 PRINT "TALALT. ELSULLYEDT"
190 PRINT "I;"I;"PROBALKOZASOD VOLT"
200 FOR J=1 TO I
210 PRINT J;" (";X(J);";";Y(J);");"
220 NEXT J
230 END

```

#### 129. Vegyük észre:

- Első ciklus (50–80. sor) a tippek begépelésére szolgál.
- Második ciklus a generált számokat állítja elő (100... 130).
- Minden generált számot össze kell hasonlítani minden begépelte számmal (140... 180. sor). Ez két egymásba skatulyázott ciklus.

A hasonlítás menete:

1. begépelte 1. véletlen
1. begépelte 2. véletlen

. .  
. .  
. .

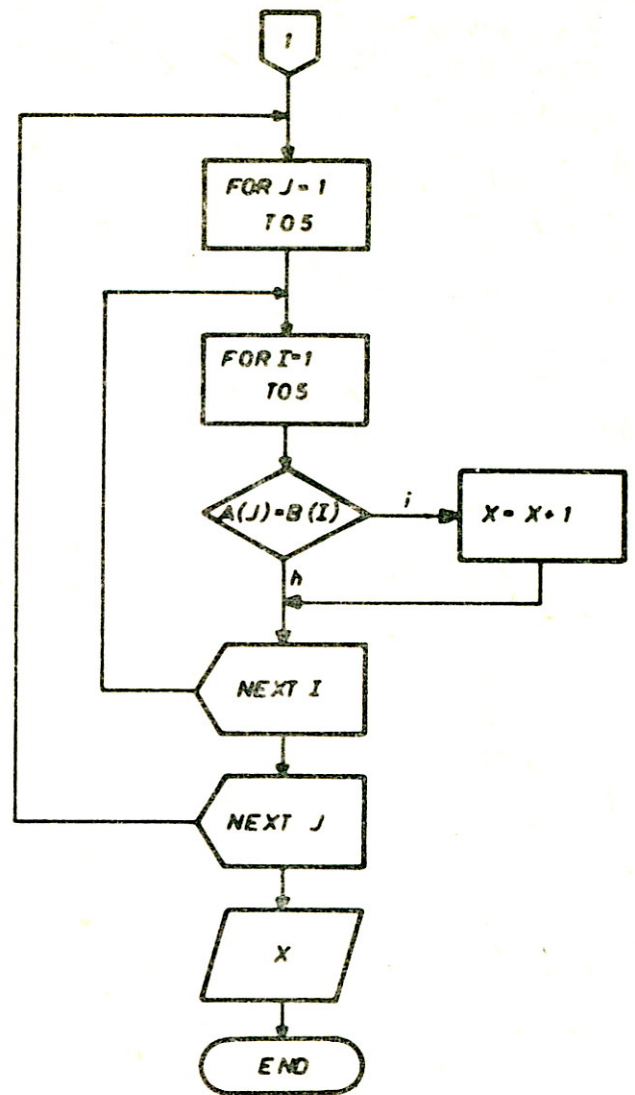
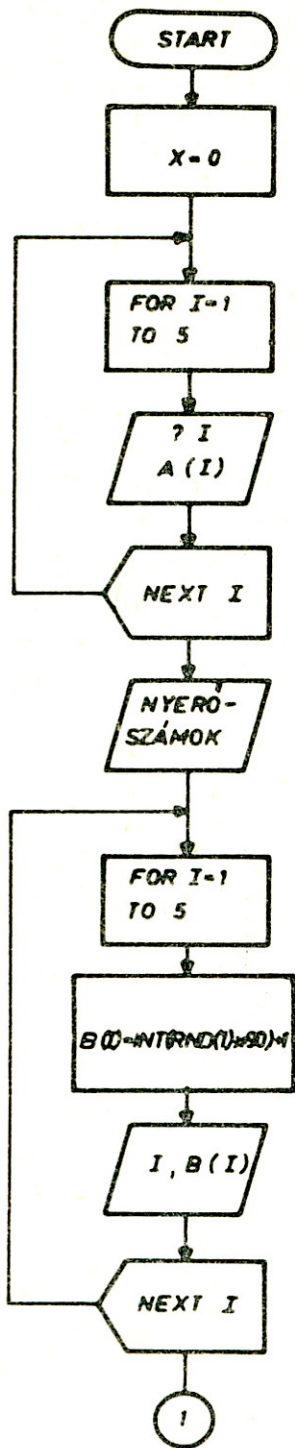
1. begépelte 5. véletlen
2. begépelte 1. véletlen

. .  
. .  
. .

5. begépelte 5. véletlen



A folyamatábra:



A program:

```

10 PRINT " "
20 PRINT TAB(14); "LOTTO"
30 PRINT TAB(9); "KEREM AZ OT TIPPET"
40 X=0
50 FOR I=1 TO 5
60 PRINT I; ". TIPP:"
70 INPUT A(I)
80 NEXT I
90 PRINT TAB(9); "A NYERO SZAMOK:"
  
```

```

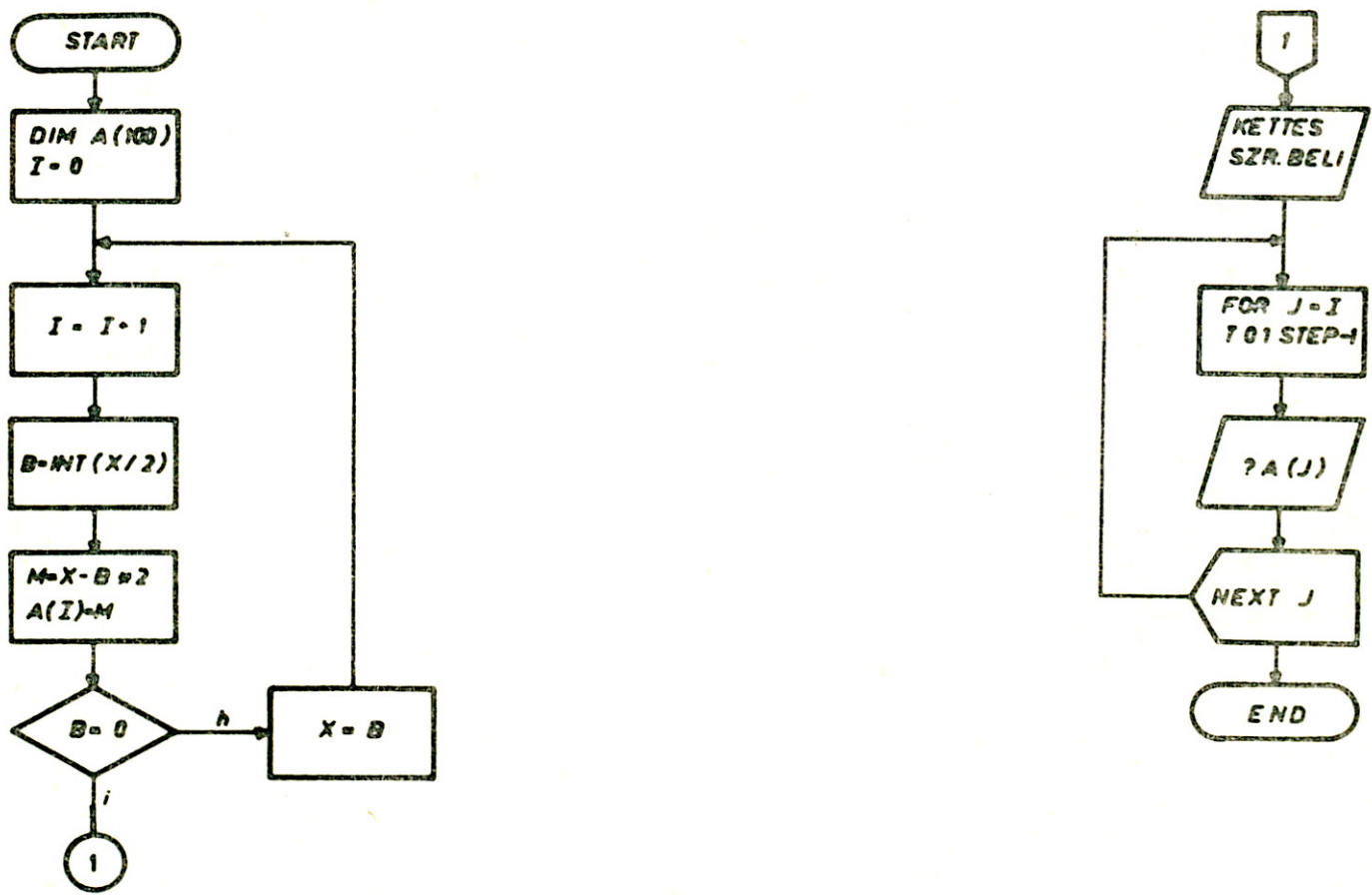
100 FOR I=1 TO 5
110 B(I)=INT(RND(1)*90)+1
120 PRINT I;" NYEROSZAM";B(I)
130 NEXT I
140 FOR J=1 TO 5
150 FOR I=1 TO 5
160 IF A(J)=B(I) THEN X=X+1
170 NEXT I
180 NEXT J
190 PRINT TAB(5);"A LOTTON";X;"TALALATA VAN"

```

**130. Vegyük észre:**

- Tíz-es számrendszerből más számrendszerbe való átírás euklideszi maradékos osztással történik. Menete: A tízes számrendszerbeli számot (X) osztom az új számrendszer alapszámával. A maradékot megjegyzem ( $A(I)=M$ ). Ha a hányados (B) nem nulla, akkor azzal mint tízes számrendszerbeli számmal az előző műveletet hajtom végre ( $X=B$ ). Ha a hányados nulla, a maradékot fordított sorrendben kiíratom (140...160. sor).

A folyamatábra:



A program:

```
10 PRINT "Q"
20 PRINTTAB(10), "SZAMRENDSZEREKQ"
30 INPUT "MIA TIZES SZAMRENDSZERBELI SZAM" X
40 DIM A(100)
50 I=0
60 I=I+1
70 B=INT(X/2)
80 M=X-B*2
90 A(I)=M
100 IF B=0 THEN GOTO 130
110 X=B
120 GOTO 60
130 PRINT "A SZAM KETTES SZAMRENDSZERBELI ALAKJA:Q"
140 FOR J=I TO 1 STEP -1
150 PRINT A(J)
160 NEXT J
170 END
190 PRINT TAB(5); "MA LOTTON" X; "TALALATA VAN"
```

131.

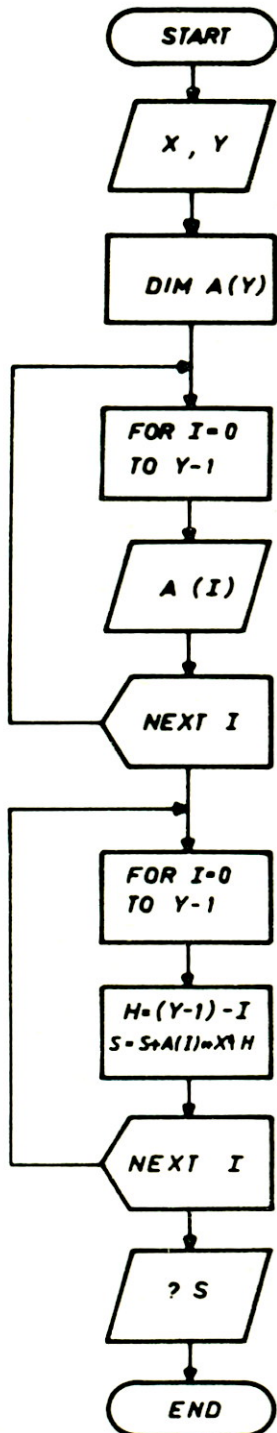
```
10 PRINT "Q"
20 PRINTTAB(10), "SZAMRENDSZEREKQ"
30 PRINT "MIA TIZES SZAMRENDSZERBELI SZAM" X
40 INPUT "MIA TIZES SZAMRENDSZERBELI SZAM" X
50 INPUT "MA SZAMRENDSZER ALAPSZAMA" Y
60 DIM A(100)
70 I=0
80 I=I+1
90 B=INT(X/Y)
100 M=X-B*Y
110 A(I)=M
120 IF B=0 THEN GOTO 150
130 X=B
140 GOTO 80
150 PRINT "MA SZAM " Y; "SZAMRENDSZERBELI ALAKJA:Q"
160 FOR J=I TO 1 STEP -1
170 PRINT A(J)
180 NEXT J
190 END
```



132. Vegyük észre:

- A tetszőleges számrendszerbeli számot célszerű számjegyként bekérni (80...100. sor).
- Ahányadik helyen gépeltük be a számjegyet, a hozzá tartozó helyiérték az alapszám ennél eggyel kisebb hatványa (120. sor).
- A tízes számrendszerbeli számot megkapjuk, ha az alaki és a helyiértékeket összeszorozzuk, majd a szorzatokat összeadjuk (130. sor).

A folyamatábra:



A program:

```
10 PRINT " "
20 PRINT "■■■■■■■■■■SZAMRENDSZER"
30 PRINT "■■■■■■TETSZOLEGESBOL TIZESBE"
40 INPUT "AZ ALAPSZAM" : X
50 INPUT "HANY SZAMJEGYBOL ALL A SZAM" : Y
60 DIM A(Y)
70 PRINT "KEREM A SZAMOT SZAMJEGYENKENT"
80 FOR I=0 TO Y-1
90 INPUT A(I)
100 NEXT I
110 FOR I=0 TO Y-1
120 H=(Y-1)-I
130 S=S+A(I)*X^H
140 NEXT I
150 PRINT "■A SZAM TIZES SZAMRENDSZERBELI ALAKJA: ■"
160 PRINT S
170 END
```

**133. Vegyük észre:**

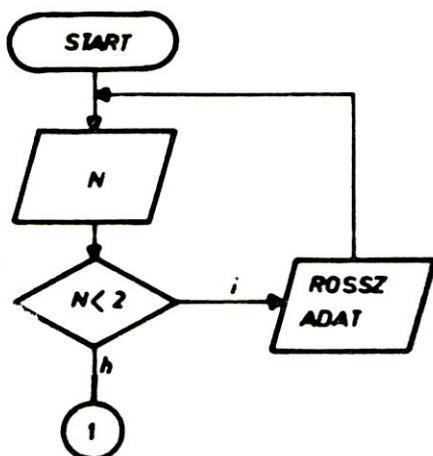
- Két, három elemmel már végeztünk rendezéseket. A tetszőleges elemszám azonban új problémát jelent.
- A feladat megoldására két ismert módszert mutatunk be:

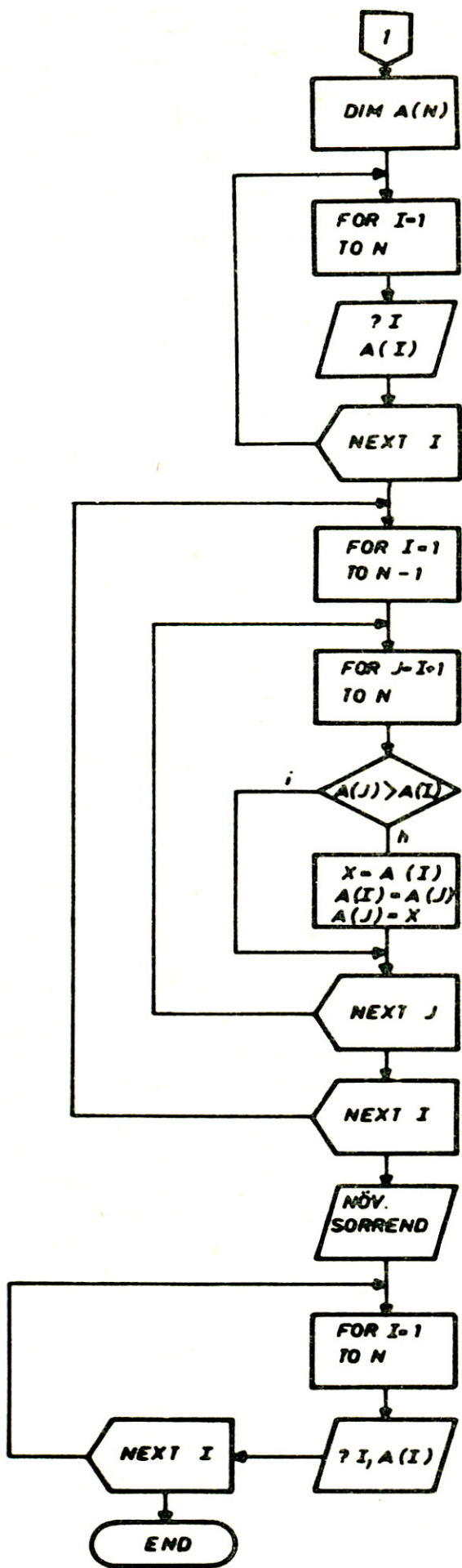
1. Buborék módszer:

- A feladat legalább két rendezendő elemet tételez fel.
- A rendezendő elemeket tömbben célszerű tárolni (70...100).
- A rendezés menete:

Vesszük az első elemet. Ehhez hasonlítom a másodikat. Ha a második nagyobb vagy egyenlő az elsővel, vesszük a harmadik elemet. Ha a második elem kisebb az elsőnél, a két változó értékét felcseréljük, majd vesszük a harmadik elemet. Ezt folytatjuk az N. elemig. Ezután a második elemet hasonlítjuk a harmadik, negyedik stb. elemhez. Ezt a algoritmust végigjátszva, az elemek a megfelelő helyre kerülnek. A kisebb számok a kisebb, a nagyobbak a nagyobb indexű helyekre kerülnek. Ez a mozgás hasonlít a különböző sűrűségű részecskéknek folyadékban való mozgására. Innen a módszer neve.

A folyamatábra:







A program:

```
10 PRINT "U"
```

```
20 PRINT "SZAMOK NOVEKEDO SORRENDBE RENDEZESE (N>1)U"
```

```
30 PRINT "BUBOREK MODSZERU"
```

```
40 INPUT "HANY ELEMET RENDEZUNK" ; N
```

```
50 IF N<2 THEN PRINT "ROSSZ ADAT" : GOTO 40
```

```
55 DIM A(N)
```

```
60 PRINT "KEREM A RENDEZENDO ELEMeket"
```

```
70 FOR I=1 TO N
```

```
80 PRINT I ; ". " ;
```

```
90 INPUT A(I)
```

```
100 NEXT I
```

```
110 FOR I=1 TO N-1
```

```
120 FOR J=I+1 TO N
```

```
130 IF A(J)>=A(I) THEN GOTO 170
```

```
140 X=A(I)
```

```
150 A(I)=A(J)
```

```
160 A(J)=X
```

```
170 NEXT J
```

```
180 NEXT I
```

```
190 PRINT "A BEGEPELT SZAMOK NOVEKVO SORRENDbenU"
```

```
200 FOR I=1 TO N
```

```
210 PRINT I ; ". SZAM" ; A(I)
```

```
220 NEXT I
```

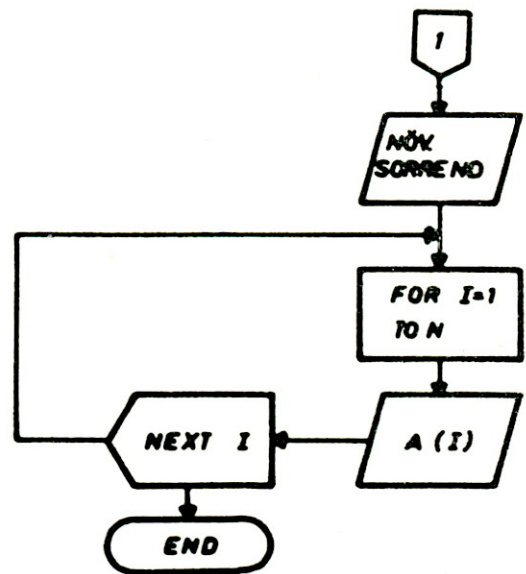
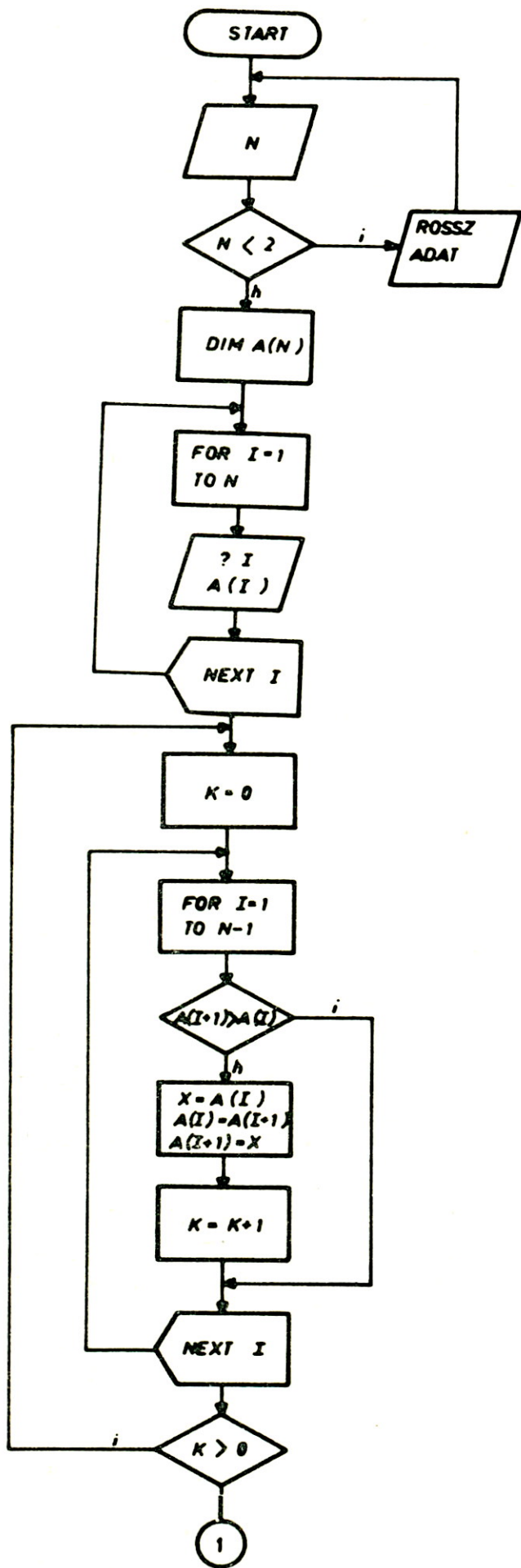
```
230 END
```

## 2. Szomszédos elemek cseréjének módszere:

- A feladat legalább két rendezendő elemet tételez fel.
- Az elemeket tömbben gyűjtjük (80...110. sor).
- A rendezés menete:

Egy ciklus segítségével elvégezzük az első és a második, a második és a harmadik, a harmadik és a negyedik, ... N-1 és N. elem összehasonlítását, szükség esetén cseréjét. (130...190. sor). A program fontos eleme a kapcsoló (K), amelynek kezdőértéke 0. Ha a cikluson belül végrehajtottunk egy cserét, K értékét is megváltoztatjuk. A ciklus végén, ha K értéke nem nulla, a K-t lenullázzuk, majd újra indítjuk az összehasonlításokat. Ha K értéke nulla, ez azt jelenti, hogy minden elem a megfelelő helyen van, csere már nem történt.

A folyamatábra:



A program:

```
10 PRINT "□"
20 PRINT "SZAMOK NOVEKEDO SORRENDBE RENDEZESE (N>1) □"
30 PRINT "SZOMSZEDOS ELEMEK CSEREJENEK MODSZERE □"
40 INPUT "HANY ELEMET RENDEZUNK" ; N
50 IF N<2 THEN PRINT "ROSSZ ADAT" : GOTO 40
60 DIM A(N)
70 PRINT "□□ KEREM A RENDEZENDO ELEMEKET"
80 FOR I=1 TO N
90 PRINT I ; ". " ;
100 INPUT A(I)
110 NEXT I
120 K=0
130 FOR I=1 TO N-1
140 IF A(I+1)>A(I) THEN GOTO 130
150 X=A(I)
160 A(I)=A(I+1)
170 A(I+1)=X
180 K=K+1
190 NEXT I
200 IF K>0 THEN GOTO 120
210 PRINT "□ A BEGEPELT ELEMEK NOVEKVO SORRENBEN □"
220 FOR I=1 TO N
230 PRINT I ; ". SZAM" ; A(I)
240 NEXT I
250 END
```

**134. Vegyük észre:**

– A probléma megegyezik az előzővel, csak itt sztring típusú változókkal kell dolgozni.

1. Buborék módszer:

A program:

```
10 PRINT "□"
20 PRINT "NEVEK NEVSORBA RENDEZESE (N>1) □"
30 PRINT "□□□□□□□□ BUBOREK MODSZER □"
40 INPUT "HANY NEVET RENDEZUNK" ; N
50 IF N<2 THEN PRINT "ROSSZ ADAT" : GOTO 40
55 DIM A(N)
60 PRINT "□□ KEREM A RENDEZENDO NEVEKET"
70 FOR I=1 TO N
80 PRINT I ; ". " ;
90 INPUT A$(I)
100 NEXT I
110 FOR I=1 TO N-1
120 FOR J=I+1 TO N
```



```

130 IF A$(J)>=A$(I) THEN GOTO 170
140 X$=A$(I)
150 A$(I)=A$(J)
160 A$(J)=X$
170 NEXT J
180 NEXT I
190 PRINT"▲▲ BEGEPELT NEVEK NEVSORBAN▲▲"
200 FOR I=1 TO N
210 PRINT I;" NEV "A$(I)
220 NEXT I
230 END

```

2. A szomszédos elemek cseréjének módszere:

A program:

```

10 PRINT"▲▲"
20 PRINT"NEVSORBA RENDEZES (N>1)▲▲"
30 PRINT"SZOMSZEDOS ELEMEK CSEREJENEK MODSZERE▲▲"
40 INPUT"HANY NEVVEL DOLGOZUNK" ;N
50 IF N<2 THEN PRINT"ROSSZ ADAT" : GOTO 40
60 DIM A(N)
70 PRINT"▲▲KEREM A RENDEZENDO NEVEKET"
80 FOR I=1 TO N
90 PRINT I;" NEV" ;
100 INPUT A$(I)
110 NEXT I
120 K=0
130 FOR I=1 TO N-1
140 IF A$(I+1)>A$(I) THEN GOTO 190
150 X$=A$(I)
160 A$(I)=A$(I+1)
170 A$(I+1)=X$
180 K=K+1
190 NEXT I
200 IF K>0 THEN GOTO 120
210 PRINT"▲▲ BEGEPELT NEVEK NEVSORBAN▲▲"
220 FOR I=1 TO N
230 PRINT I;" NEV "A$(I)
240 NEXT I
250 END

```

135. Vegyük észre:

- A menü eljárással már foglalkoztunk.
- A feladat két részre bontható:
  1. Az író neve szerinti vizsgálat:

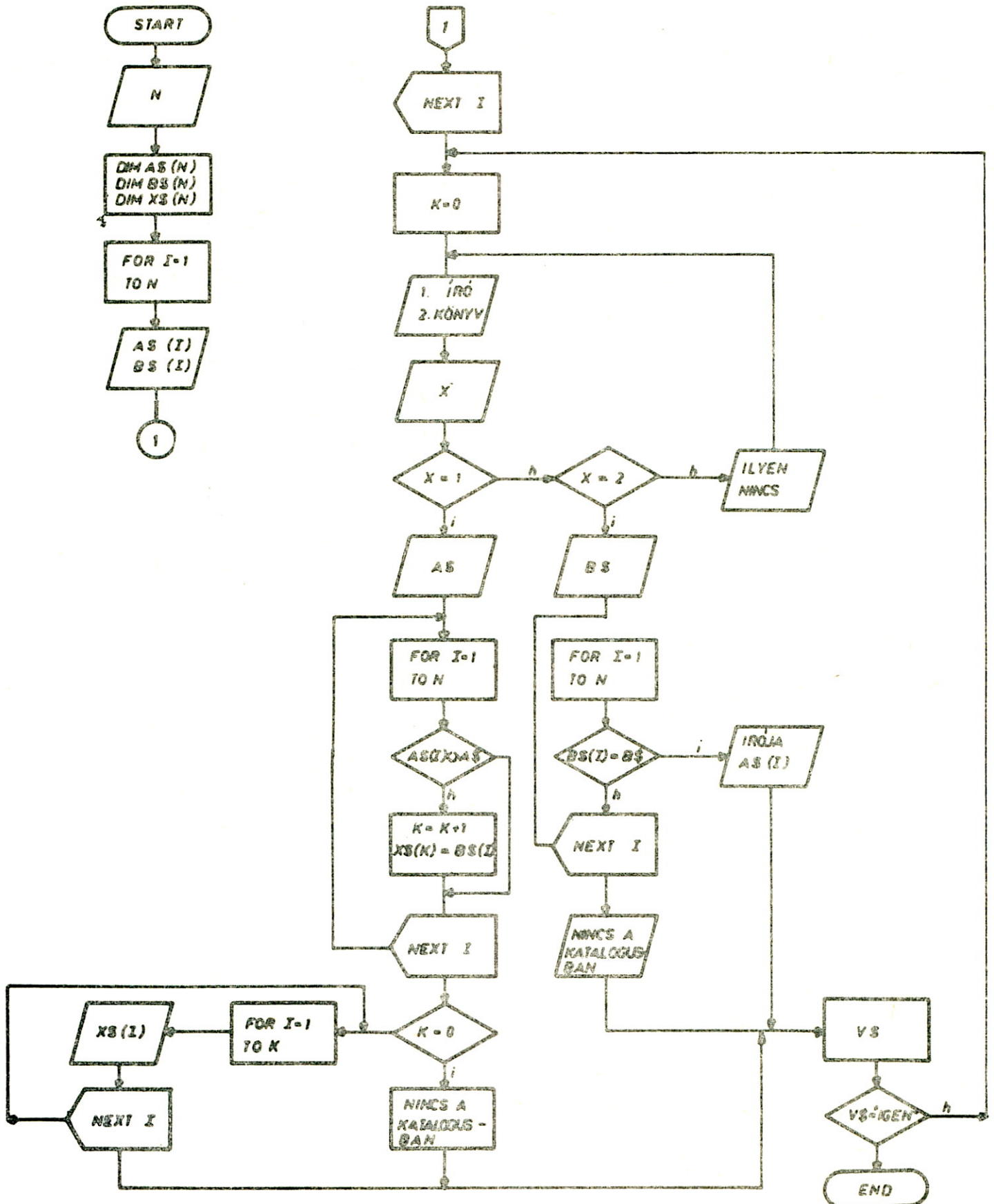
Egy íróhoz több könyv is tartozhat. Ezért célszerű a megtalált könyveket tömbben gyűjteni addig, amíg a keresés be nem fejeződik.

$\langle X\$ (I) \rangle$ . K értéke megmutatja, hogy hány könyvet találtunk. A megtalált könyveket ki kell íratni (230 ... 330. sor).

2. A könyv címe alapján:

Mivel egy könyvhöz csak egy szerzőt rögzítettünk, ezért biztos, hogy ha megtaláltuk egy könyv szerzőjét, tovább már nem kell keresni. A megtalált szerzőt ki kell íratni (350 ... 410. sor).

A folyamatábra:



A program:

```
10 PRINT " "
20 PRINT "KONYVJEGYZEK "
30 INPUT "HANY KONYVVEL DOLGOZUNK " ; N
40 DIM A$(N)
50 DIM B$(N)
60 DIM X$(N)
70 PRINT "KEREM AZ ADATOKAT "
80 FOR I=1 TO N
90 INPUT "SZERZO NEVE " ; A$(I)
100 INPUT "KONYV CIME " ; B$(I)
110 PRINT
120 NEXT I
130 PRINT " "
140 K=0
150 PRINT "MELYIK ADAT ALAPJAN DOLGOZUNK ? "
160 PRINT "1. AZ IRO NEVE "
170 PRINT "2. A KONYV CIME "
180 INPUT X
190 IF X=1 THEN GOTO 230
200 IF X=2 THEN GOTO 350
210 PRINT "ILYEN LEHETOSEG NINCS "
220 GOTO 150
230 INPUT "AZ IRO NEVE " ; A#
240 FOR I=1 TO N
250 IF A# < A$(I) THEN GOTO 260
260 K=K+1
270 X$(K)=B$(I)
280 NEXT I
290 IF K=0 THEN PRINT "NINCS A KATALOGUSBAN" : GOTO 420
300 PRINT "A KONYVEI : "
310 FOR I=1 TO K
320 PRINT X$(I)
330 NEXT I
340 GOTO 420
350 INPUT "A KONYV CIME " ; B#
360 FOR I=1 TO N
370 IF B# = B$(I) THEN GOTO 410
380 NEXT I
390 PRINT "NINCS A KATALOGUSBAN"
400 GOTO 420
410 PRINT "AZ IROJA " ; A$(I)
420 INPUT "DOLGOZUNK MEG " ; V#
430 IF V# = "IGEN" THEN GOTO 130
440 END
```



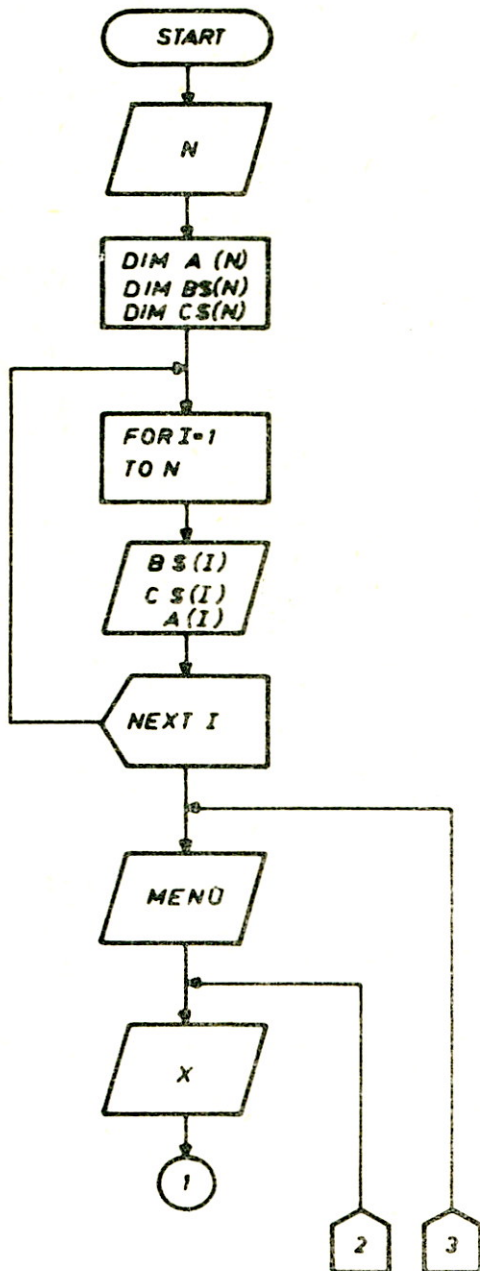
136. Vegyük észre:

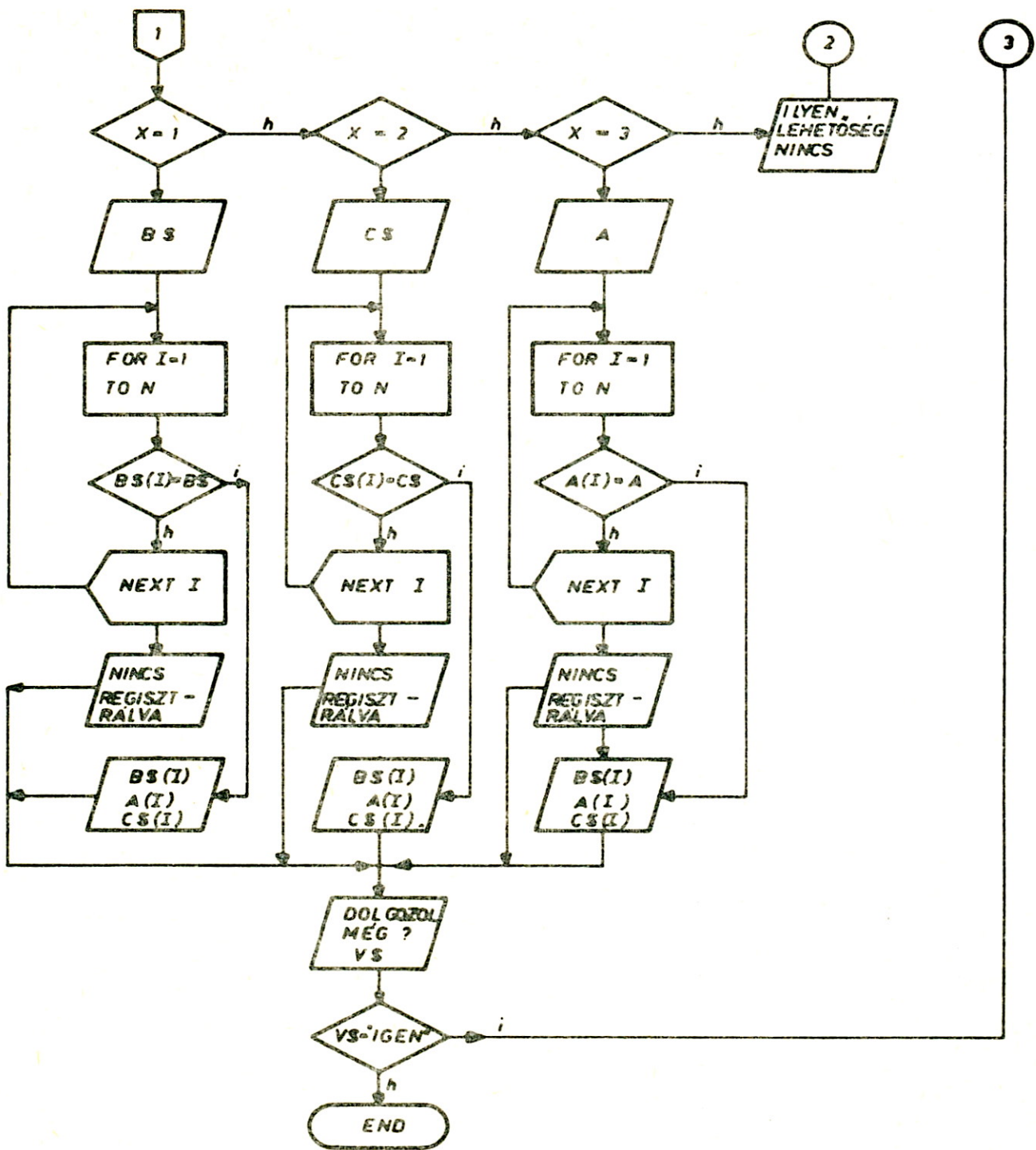
- A begépelte adatokat tömbökben tárolhatjuk (80...130. sor).
- A menütechnika már ismert (150...240. sor).  
Név alapján történő lekérdezés (250...300. sor)  
Lakcím alapján történő lekérdezés (330...460. sor)  
Telefonszám alapján történő lekérdezés (410...460. sor).

Mindhárom esetben egy névhez egy telefonszámot és egy címet tételezünk fel.

- A program tetszés szerinti futást engedélyez.

A folyamatábra:





A program:

```
10 PRINT "□"
20 PRINT "██████████TELEFONREGISZTER███"
30 INPUT "HANY EMBER ADATAIT ROGZITJUK" ; N
40 DIM A$(N)
50 DIM B$(N)
60 DIM C$(N)
70 PRINT "KEREM AZ ADATOKAT"
80 FOR I=1 TO N
90 INPUT "NEVE " ; B$(I)
100 INPUT "LAKCIME " ; C$(I)
110 INPUT "TELEFONSZAMA " ; A$(I)
120 PRINT
130 NEXT I
140 PRINT "□"
150 PRINT "██████MELYIK ADAT ALAPJAN DOLGOZUNK?███"
160 PRINT "1. NEV"
170 PRINT "2. LAKCIM"
180 PRINT "3. TELEFONSZAM"
190 INPUT X
200 IF X=1 THEN GOTO 250
210 IF X=2 THEN GOTO 330
220 IF X=3 THEN GOTO 410
230 PRINT "ILYEN LEHETOSEG NINCS"
240 GOTO 140
250 INPUT "A NEV " ; B#
260 FOR I=1 TO N
270 IF B$(I)=B# THEN GOTO 310
280 NEXT I
290 PRINT "NINCS REGISZTRALVA"
300 GOTO 480
310 PRINT B# ; " TELEFONSZAMA " ; A$(I) ; " CIME " ; C$(I)
320 GOTO 480
330 INPUT "A LAKCIM " ; C#
340 FOR I=1 TO N
350 IF C$(I)=C# THEN GOTO 390
360 NEXT I
370 PRINT "NINCS REGISZTRALVA"
380 GOTO 480
390 PRINT B$(I) ; " TELEFONSZAMA " ; A$(I) ; " CIME " ; C$(I)
400 GOTO 480
410 INPUT "A TELEFONSZAM" ; A#
420 FOR I=1 TO N
430 IF A$(I)=A# THEN GOTO 470
```



```

440 NEXT I
450 PRINT "NINCS REGISZTRÁLVA"
460 GOTO 480
470 PRINTB$(I); " TELEFONSZÁMA " ; A$(I); " CÍME " ; C$(I)
480 INPUT "DOLGOZUNK MEG" ; V$
490 IF V$ = "IGEN" THEN GOTO 140
500 END

```

### 3.9 READ, DATA, RESTORE

- Összetartozó utasítások.
- A gép a READ utasításra a DATA után következő adatokat változóiban helyezi el.
- Ahányszor a program READ utasításhoz érkezik, mindig a következő adatot olvassa be.
- Ha több a READ, mint az adat, a gép hibát jelez.
- A DATA a programban bárhol elhelyezhető. Általában a program elejére vagy végére teszik.
- Ha ugyanazokat az adatokat többször be akarjuk olvasni, a RESTORE utasítással az adatok elejéről kezdhethetjük az olvasást.
- A DATA után bármilyen adatot elhelyezhetünk (sztring, valós...). Ügyelni kell arra, hogy a READ utasításra a megfelelő változóba gyűjtsük az adatokat!

#### 137. a)

Vegyük észre:

- Több adatot helyeztünk el a DATA-kban, mint amire szükségünk van.
- A változók feltöltése 30...50. sorban történik.
- A tömböt nem szükséges dimenzionálni, mert csak 10 eleme van. Próbáld ki úgy is a programot!

A program:

```

10 PRINT "A"
20 PRINT "#####DATA GYAKORLO#"
25 DIM A(10)
30 FOR I=1 TO 10
40 READ A(I)
50 NEXT I
60 FOR I=1 TO 10
70 PRINT A(I)
80 NEXT I
90 DATA 1,2,3,4,5,6,7,8,9,10
100 DATA 11,12,13,14,15,16,17,18,19,20
110 END

```

#### b)

Vegyük észre:

- 10 adattal kevesebbet helyeztünk el a DATA-kban, mint amennyire szükségünk van.

- Használjuk a RESTORE utasítást a hús meglevő elem beolvasása után!
- A feladat többféleképpen is megoldható. Pl:
  1. A ciklus 30-szor fut le. Minden futás után vizsgáljuk, hogy elértük-e a 20-at. Ha igen, RESTORE.
  2. Két ciklust alkalmazunk. Egyik fut 20-szor, a másik 21...30-ig. A kető között RESTORE.

A programok:

1.

```

10 PRINT "1."
20 PRINT "#####DATA GYAKORLO1"
30 DIM A(30)
40 FOR I=1 TO 30
50 READ A(I)
60 IF I<>20 THEN GOTO 80
70 RESTORE
80 NEXT I
90 FOR I=1 TO 30
100 PRINT A(I)
110 NEXT I
120 DATA 1,2,3,4,5,6,7,8,9,10
130 DATA 11,12,13,14,15,16,17,18,19,20
140 END

```

2.

```

10 PRINT "2."
20 PRINT "#####DATA GYAKORLO2"
30 DIM A(30)
40 FOR I=1 TO 20
50 READ A(I)
60 NEXT I
70 RESTORE
80 FOR I=21 TO 30
90 READ A(I)
100 NEXT I
110 FOR I=1 TO 30
120 PRINT A(I)
130 NEXT I
140 DATA 1,2,3,4,5,6,7,8,9,10
150 DATA 11,12,13,14,15,16,17,18,19,20
160 END

```

c) Vegyük észre:

- Az első öt számot kell beolvasnunk, így minden ötödik beolvasása után RESTORE-t kell alkalmazni.
- Akkor olvastuk be az ötödik számot, ha a ciklusváltozó osztható öttel.





**139. Vegyük észre:**

- Az INPUT utasítások előtt célszerű kiírni, hogy milyen tantárgy eredményét várja a gép.
- Vigyázni kell arra, hogy a magatartás és a szorgalom nem számít a tanulmányi átlagba!
- Ha valaki bármilyen tantárgyból bukik, az átlaga 1.

A program:

```

10 PRINT " "
20 PRINT "TANULMÁNYI ÁTLAGSZÁMÍTÓ"
30 DIM A$(13)
40 FOR I=1 TO 13
50 READ A$(I)
60 NEXT I
70 INPUT "A TANULO NEVE" : N$
80 PRINT "KEREM A TANULO FELEVI JEGYEIT !"
90 FOR I=1 TO 13
100 PRINT A$(I) ;
110 INPUT J
120 IF J>5 THEN GOTO 150
130 IF J<1 THEN GOTO 150
140 IF J=INT(J) THEN GOTO 160
150 PRINT "ROSSZ ADAT" : GOTO 100
160 IF J=1 THEN A=1 : GOTO 210
170 S=S+J
180 NEXT I
190 A=S/13
200 PRINT
210 PRINT N$ ; " TANULMANYI ÁTLAGA" ; A
220 END
230 DATA MAGYAR NYELV ,MAGYAR IRODALOM ,OROSZ
240 DATA TORTENELEM ,FOLDRAJZ ,MATEMATIKA ,FIZIKA ,KEMIA
250 DATA BIOLOGIA ,TECHNIKA ,RAJZ ,ENEK ,TESTNEVELES

```

**140. Vegyük észre:**

- Az egyes címletek generálása helyett célszerűbb azokat DATA-ban tárolni, és onnan kiolvasni.
- A címletek értékével számolni kell, ezért azokat a kiolvasáskor valós változóként célszerű kezelni.
- Célszerű csak azokat a címleteket kiírni, amelyek darabszáma nem nulla.

A program:

```

10 PRINT " "
20 PRINT TAB(15) ; "CIMLETEZŐ"
30 DIM A(12) : DIM B(12)
40 FOR I=1 TO 12

```

```

50 READ A(I)
60 NEXT I
70 INPUT "MEKKORA AZ OSSZEG" : X
80 FOR I=1 TO 12
90 IF X < A(I) THEN GOTO 140
100 K=K+1
110 B(I)=INT(X/A(I))
120 M=X-B(I)*A(I)
130 X=M
140 NEXT I
150 PRINT " "
160 PRINT TAB(15); "CIMLETEZO"
170 IF K=0 THEN PRINT "TUL KICSI AZ OSSZEG" : GOTO 220
180 FOR I=1 TO 12
190 IF B(I)=0 THEN GOTO 210
200 PRINT A(I); "FT"; TAB(20); B(I); "DB"
210 NEXT I
220 END
230 DATA 1000,500,100,50,20,10,5,2,1,0.5,0.2,0.1

```

#### 141. Vegyük észre:

- A feladat egy emberről három adatot tárol (név, lakcím, telefonszám).
- Lehetővé kell tennünk, hogy egy adat ismeretében a használó a másik két adatot is megkapja.
- Ha a DATA-ban levő adatokat bővítjük, a tömbök dimenzionálásán is változtatni kell!
- Feltételeztük, hogy minden névhez egy cím, egy telefonszám tartozik.

A program:

```

10 PRINT " "
20 N=6
30 PRINT "TELEFONREGISZTER"
40 PRINT "SZEMELY ADATAIT TARTJUK NYILVAN"
50 DIM A(N)
60 DIM B$(N)
70 DIM C$(N)
80 FOR I=1 TO N
90 READ B$(I)
100 READ C$(I)
110 READ A(I)
120 NEXT I
130 PRINT "MELYIK ADAT ALAPJAN DOLGOZUNK?"
140 PRINT "1. NEV"
150 PRINT "2. LAKCIM"
160 PRINT "3. TELEFONSZAM"
170 INPUT X

```







```

70 DIM X$(N)
80 FOR I=1 TO N
90 READ A$(I)
100 READ B$(I)
110 NEXT I
120 K=0
130 PRINT "MELYIK ADAT ALAPJAN DOLGOZUNK?"
140 PRINT "1. AZ IRO NEVE"
150 PRINT "2. A KONYV CIME"
160 INPUT X
170 IF X=1 THEN GOTO 210
180 IF X=2 THEN GOTO 330
190 PRINT "ILYEN LEHETOSEG NINCS"
200 GOTO 130
210 INPUT "AZ IRO NEVE" ; A$
220 FOR I=1 TO N
230 IF A$ < A$(I) THEN GOTO 260
240 K=K+1
250 X$(K)=B$(I)
260 NEXT I
270 IF K=0 THEN PRINT "NINCS A KATALOGUSBAN" : GOTO 400
280 PRINT "KONYVEI:"
290 FOR I=1 TO K
300 PRINT X$(I)
310 NEXT I
320 GOTO 400
330 INPUT "A KONYV CIME" ; B$
340 FOR I=1 TO N
350 IF B$ = B$(I) THEN GOTO 390
360 NEXT I
370 PRINT "NINCS A KATALOGUSBAN"
380 GOTO 400
390 PRINT "AZ IROJA " ; A$(I)
400 INPUT "DOLGOZUNK MEG" ; V$
410 IF V$ = "IGEN" THEN GOTO 120
420 END
430 DATA JOKAI MOR,KOSZIVU EMBER FIAI
440 DATA FEKETE ISTVAN,VUK
450 DATA FEKETE ISTVAN,TUSKEVAR
460 DATA MILNE,MICIMACKO,TRAVERS,CSUDALATOS MARY

```

### 143. Vegyük észre:

- Egy kazettán több program lehet. Az is fontos, hogy az adott program milyen fordulatonál található.

A program:

```
10 PRINT " "
20 N=9
30 PRINT "PROGRAM KATALOGUS"
40 PRINT N; "PROGRAMOT ROGZITETTUNK"
50 DIM A$(N)
60 DIM B$(N)
70 DIM C(N)
80 DIM X$(N)
90 DIM Y(N)
100 FOR I=1 TO N
110 READ A$(I)
120 READ B$(I)
130 READ C(I)
140 NEXT I
150 K=0
160 PRINT "MELYIK ADAT ALAPJAN DOLGOZUNK?"
170 PRINT "1. A KAZETTA MEGNEVEZESE "
180 PRINT "2. A PROGRAM NEVE"
190 INPUT X
200 IF X=1 THEN GOTO 240
210 IF X=2 THEN GOTO 380
220 PRINT "ILYEN LEHETOSEG NINCS"
230 GOTO 160
240 INPUT "A KAZETTA MEGNEVEZESE ";A$
250 FOR I=1 TO N
260 IF A$<>A$(I) THEN GOTO 300
270 K=K+1
280 X$(K)=B$(I)
290 Y(K)=C(I)
300 NEXT I
310 IF K=0 THEN PRINT "NINCS A KATALOGUSBAN":GOTO 460
320 PRINT "A KAZETTAN LEVO PROGRAMOK:"
330 PRINT "PROGRAMNEV";TAB(23);"FORDULATSZAM"
340 FOR I=1 TO K
350 PRINT X$(I);TAB(23);Y(I)
360 NEXT I
370 GOTO 460
380 INPUT "A PROGRAM NEVE ";B$
390 FOR I=1 TO N
400 IF B$=B$(I) THEN GOTO 440
410 NEXT I
420 PRINT "NINCS A KATALOGUSBAN"
430 GOTO 460
440 PRINT "A KAZETTA MEGNEVEZESE ";A$(I)
```



```

450 PRINT "A FORDULATSZAM "; C(I)
460 INPUT "MOLGOZUNK MEG "; V$
470 IF V$="IGEN" THEN GOTO 150
480 END
490 DATA TDK1,SZAMRENDSZEREK(10-2),012,AGFA,NEGYZETEK,976
500 DATA TDK1,TERULETSZAMOLO,009,SANYO,NEGYZETEK,234
510 DATA AGFA,SZOVETARAZO,326,TDK2,EGYSEJTUEK,886
520 DATA TDK1,OHM TORVENYE,828,TDK2,ELLENALLAS,953

```

### 3.10 GET A\$

- A\$ helyén bármilyen sztringváltozó szerepelhet.
- Az utasítás hasonló szerepet tölt be, mint az INPUT utasítás.
- Mikor a program futás közben a GET A\$ utasításhoz ér, az A\$ változó felveszi az abban a pillanatban megnyomott billentyű karakterértékét. Ha abban a pillanatban éppen nem nyomtunk meg billentyűt, A\$ értéke " " lesz. A GET A\$ tehát nem várakozik, csak az adott állapotot figyeli.
- Az előbbieket miatt A\$ értéke mindig csak egy karakternyi lesz.
- Játékprogramok, kapcsolókkal ellátott programok nélkülözhetetlen eleme.

#### 144. Vegyük észre:

- Ha a program a GET A\$ utasításon csak egyszer halad át, igen nehéz eltalálni a megfelelő pillanatot a billentyű lenyomására.
  - Célszerű olyan végtelen ciklust alkalmazni, amely egészen addig visszatér a GET A\$ utasításra, amíg a feltétel nem teljesül.
- A program: (csak a C= billentyű lenyomására fut tovább!)

```

10 PRINT "□"
20 GET A$
30 IF A$(">")="*" THEN GOTO 20

```

#### 145. Vegyük észre:

- A feladat hasonló az előzőhöz, csak most a feltétel, hogy A\$ értéke " " - tól különbözzön.

A program:

```

10 PRINT "□"
20 GET A$
30 IF A$=" " THEN GOTO 20

```

#### 146. Vegyük észre:

- Egy képernyő akkor „lapozható”, ha a használó dönti el, hogy az adott képernyőtartalmat meddig hagyja meg és mikor kéri az újabbat.
- Ennek megoldására jól használható a GET utasítás.
- A feladat megoldására két lehetőséget mutatunk:
  1. Egy ciklust alkalmazva, és a cikluson belül figyelve, hogy a ciklusváltozó értéke mikor lesz 20, vagy annak többszöröse.





**147. Vegyük észre:**

- Ha TI-t csak egyszer íratjuk ki, a program nem adja folyamatosan az aktuális gépidőt.
- A kiírás csak akkor tűnik óraszerűnek, ha az aktuális gépidő mindig a képernyő ugyanazon helyén jelenik meg.

A program:

```
10 PRINT "□"  
20 PRINT "████████████████████A PILLANATNYI GEPIDO "TI$"  
30 PRINT "□"  
40 GOTO 20
```

**148. Vegyük észre:**

- A stopperórát indítani és megállítani is lehet.
- A megjelenített képnek óra hatást kell keltenie, ezt úgy érjük el, hogy az időt mindig a képernyő ugyanazon helyén íratjuk ki.
- TI\$-t az induláskor le kell nullázni.
- TI az időt hatvanad másodpercben méri, így a TI változó értékét módosítanunk kell, hogy a céljainknak megfeleljen.

A program:

```
10 PRINT "□"  
20 PRINT "████████████████████STOPPERORA□"  
30 PRINT "INDITAS ES MEGALLITAS A * GOMBBAL "  
40 GET A$  
50 IF A$(">")="*" THEN GOTO 40  
60 PRINT "□"  
70 TI$="000000 "  
80 I=INT(TI/6)  
90 I=I/10  
100 PRINT "██████████████████████[0]███" ; I  
110 GET A$  
120 IF A$="*" THEN GOTO 150  
130 PRINT "□"  
140 GOTO 80  
150 END
```

**149. Vegyük észre:**

- A TI\$-t nem csak nullázni, hanem más értékekre állítani is lehet.
- Vigyázni kell, hogy az óra, perc, másodperc értékek együtt hat karaktert tegyenek ki. Ha valamelyik érték egyjegyű szám, akkor elé nullát kell beírni.
- Jól használható az összefűzés (konkatenáció) művelete a TI\$ értékének megadására (120. sor).

A program:

```
10 PRINT "□"  
20 PRINT "████████████████████ORORA□"  
20 PRINT "ALLITSA BE AZ ORAT! GEPELJE BE AZ ORAT,□"
```







```

180 TI#="000000"
190 FOR K=1 TO 800
200 GET X#
210 IF X#="" THEN GOTO 240
220 IF X#=A$(A) THEN S=S+1
230 GOTO 250
240 NEXT K
250 X=X+TI
260 NEXT J
270 PRINT "U"
280 PRINT "#####20 PROBABOL" ; S ; "TALALAT"
290 PRINT "#####MEGOLDASI ISDO" ; X/60 ; "MASODPERC"
300 INPUT "#####JATSZUNK MEG" ; V#
310 IF V#="IGEN" THEN GOTO 140
320 END
330 DATA 1,2,3,4,5,6,7,8,9,0,W,E,R,T,Y,U,I,O,P,@,+,-
340 DATA A,S,D,F,G,H,J,K,L,":",",",*,Z,X,C,V,B,N,M,.,/,&,"=",
350 DATA !,#,%,&,',(,),†,<,>?,+,†,[,]

```

### 3.12 Képernyőkezelés, grafika

#### Színek

COLOR utasítás

Háttér színezése:

COLOR 4, színkód, árnyalat

Papír színezése:

COLOR 0, színkód, árnyalat

Kurzor színezése:

COLOR 1, színkód, árnyalat

Az árnyalatokat 0–7 közötti számmal adhatjuk meg.

#### Színkódok:

1. fekete	6. zöld	11. sárgászöld
2. fehér	7. kék	12. rózsaszín
3. piros	8. sárga	13. kékeszöld
4. enciánkék	9. narancs	14. világoskék
5. bíbor	10. barna	15. sötétkék
		16. világoszöld

#### Grafikus üzemmód

A grafikus üzemmód adta lehetőségeket csak akkor használhatjuk ki, ha a Graphic utasítással áttérünk a grafikus üzemmódra.

#### Lehetséges üzemmódok:

Szöveges üzemmód:	GRAPHIC 0, 1
Finomfelbontás:	GRAPHIC 1, 1
Finomfelbontás+szöveg:	GRAPHIC 2, 1
Sokszínű üzemmód:	GRAPHIC 3, 1
Sokszínű+szöveg:	GRAPHIC 4, 1

A finomfelbontásban a gép 320 oszlopot és 200 sort hoz létre. A finomfelbontás

+szöveg üzemmódban csak 160 sorral dolgozik és a képernyő alját megtartja a normál képernyő üzemmódban.

### A grafikus üzemmód speciális utasításai

#### DRAW

Az utasítás alkalmas pont, egyenes, megtört vonal, egyenesek által határolt sokszög rajzoltatására.

Pont: DRAW színkód, oszlop, sor

Egyenes: DRAW színkód, oszlop, sor TO oszlop, sor

Egyenes a legutóbbi pontból: DRAW szín TO oszlop, sor

#### CIRCLE

Az utasítás alkalmas kör, ellipszis, ív és ezekbe rajzolható síkidomok megrajzolására.

Kör: CIRCLE 1, középpont oszlop, középpont sor, sugár

Ellipszis: CIRCLE 1, középpont oszlop, középpont sor, szélesség, magasság

Ív: CIRCLE 1, középpont oszlop, középpont sor, szélesség, magasság, kezdet, vég

Elforgatott ellipszis: CIRCLE 1, középpont oszlop, középpont sor, szélesség, magasság, szög

Sokszög: CIRCLE 1, középpont oszlop, középpont sor, szélesség, magasság, 360/szögek száma (itt a szög a középpontból a két szomszédos csúcshoz húzott sugár által bezárt szög).

#### BOX

Az utasítás téglalap alakú területek rajzolására alkalmas.

Körvonal: BOX szín, oszl. 1, sor 1, oszl. 2, sor 2 (A két átellenes csúcs koordinátái.)

Elfordítva: BOX szín, oszl. 1, sor 1, oszl. 2, sor 2, szög

Tömör felület: BOX szín, oszl. 1, sor 1, oszl. 2, sor 2, kitöltve

Elfordítva: BOX szín, oszl. 1, sor 1, oszl. 2, sor 2, szög, kitöltve

#### PAINT

Az utasítás összefüggő vonallal körülhatárolt területek színezésére alkalmas. A színezés a COLOR utasításban megadott kurzorszínnel történik.

Színezés: PAINT színkód, belső pont oszlop, belső pont sor.

#### CHAR

A grafikus üzemmódban a képernyőre PRINT utasítással nem lehet írni. A CHAR utasítás a képernyőt 40 oszlop és 25 sor felbontásúnak tekinti, és normál karaktereket helyez a megjelölt helyre.

Kiírás: CHAR szín, oszlop, sor „SZÖVEG”.

#### SCNCLR

A grafikus üzemmódban a képernyőtörlésre szolgáló utasítás.

### 151. Vegyük észre:

– Az indításra és megállításra jól használható a GET A\$ utasítás.



- Az oszlop koordinátái 0–320, a sor koordinátái 0–200 közötti számok lehetnek.

A program:

```

10 GRAPHIC 1,1
20 SCNCLR
30 GET A#
40 IF A#<>"I" THEN GOTO 30
50 X=INT(RND(1)*320)
60 Y=INT(RND(1)*200)
70 DRAW 1,X,Y
80 GET A#
90 IF A#<>"V" THEN GOTO 50
100 END

```

### 152. Vegyük észre:

- Nincs más dolgunk, mint a koordinátáknak megfelelő pontokat DRAW utasítással megjeleníteni.
- A megfelelő pontokat összekötve egy vitorláshajót kapunk. A pontok összekötése az a) programmal történik.

A programok:

a)

```

10 GRAPHIC 1,1
20 DRAW 1,40,110
30 DRAW 1,80,110
40 DRAW 1,140,110
50 DRAW 1,260,110
60 DRAW 1,300,110
70 DRAW 1,80,130
80 DRAW 1,260,130
90 DRAW 1,140,20
100 DRAW 1,140,20
110 DRAW 1,140,60
120 DRAW 1,140,110
130 DRAW 1,140,130

```

b)

```

10 GRAPHIC 1,1
20 DRAW 1,40,110 TO 300,110
30 DRAW 1,80,130 TO 260,130
40 DRAW 1,80,130 TO 40,110
50 DRAW 1,140,20 TO 260,110
60 DRAW 1,140,20 TO 140,110
70 DRAW 1,140,60 TO 80,110
80 DRAW 1,260,130 TO 300,110

```

### 154.

a) Vegyük észre:

- A programot futtatva azt tapasztaljuk, hogy a gép a képernyő minden 10. sorába húz egy vízszintes vonalat.
- Az ok: A külső ciklus tartalmazza a sorok változását, a belső az oszlopokét. Így ugyanarra a J értékre az I értékének 0...320-ig történő változása, vagyis egy vízszintes vonal húzása jut.

b) Vegyük észre:

- A programot futtatva azt tapasztaljuk, hogy a gép a képernyő minden 10. sorában egyszerre húzza a vízszintes vonalat.



- Az ok: A külső ciklus tartalmazza az oszlopok, a belső a sorok változását. Így ugyanarra az I értékre a J értékének 0-tól 200-ig való változása jut. Így a gép nem az egymás melletti, hanem az egymás alatti pontokat jelöli ki egy körülfordulás alkalmával. Mivel a rajzolás gyorsan történik, ezért úgy tűnik, mintha az összes vízszintes vonalat egyszerre húzná a gép.

**155. Vegyük észre:**

- Vonalat a géppel úgy rajzolhatunk, ha megadjuk a kezdő- és a végpont koordinátáit.
- GRAPHIC 1, 1 üzemmódban X koordináta 0...320, Y koordináta 0...200 lehet.

A program:

```

10 GRAPHIC 1,1
20 SCNCLR
30 X1=INT(RND(1)*321)
40 X2=INT(RND(1)*321)
50 Y1=INT(RND(1)*201)
60 Y2=INT(RND(1)*201)
70 DRAW 1,X1,Y1 TO X2,Y2
80 GOTO 30

```

**156. Vegyük észre:**

- Mivel a rajzoltatás a bal felső saroktól (0;0 koordinátájú pont) indul, ezt a pontot külön meg kell rajzolni.
- Ha összefüggő vonalról van szó, az azt jelenti, hogy a következő szakasz az előző végpontjából indul ki. Ezt kihasználhatjuk a programozás során.

A program:

```

10 GRAPHIC 1,1
20 SCNCLR
30 DRAW 1,0,0
40 X=INT(RND(1)*321)
50 Y=INT(RND(1)*201)
60 DRAW 1 TO X,Y
70 GOTO 40

```

**157. Vegyük észre:**

- A feladat többféleképpen megoldható:
  1. DRAW utasítással: A négy csúcspont koordinátáit adjuk meg, amelyeket összekötünk.
  2. BOX utasítással: A két átellenes csúcspont koordinátáit adjuk meg.
  3. A CIRCLE utasítással: A középpont koordinátáit, a szélességet, a magasságot és az átlók által bezárt szöget adjuk meg.

A programok:

1.

```
10 GRAPHIC 1,1
20 SONCLR
30 DRAW 1,100,80 TO 180,80
40 DRAW 1,180,80 TO 180,120
50 DRAW 1,180,120 TO 140,120
60 DRAW 1,140,120 TO 140,80
70 END
```

2.

```
10 GRAPHIC 1,1
20 SONCLR
30 BOX 1,140,80,180,120
40 END
```

3.

```
10 GRAPHIC 1,1
20 SONCLR
30 CIRCLE 1,140,80,40,40,,,,,90
40 END
```

158. Vegyük észre:

- A színezés mindig az aktuális kurzorszínnel történik. Ha ez nem felel meg, a kurzor színén állítani kell.
- A színezéshez meg kell adni egy belső pont koordinátáját.

A program:

```
10 GRAPHIC 1,1
20 SONCLR
30 COLOR 1,1,7
40 BOX 1,140,80,180,120
50 PAINT 1,160,100
60 END
```

159. Vegyük észre:

- A megfelelő színek beállítása után két egymásba skatulyázott ciklussal a feladat könnyedén megoldható.

A program:

```
10 PRINT "□"
20 COLOR 0,1,7
30 COLOR 4,2,7
40 COLOR 1,3,7
50 FOR I=1 TO 23
60 FOR J=1 TO 40
70 PRINT "*";
80 NEXT J
90 NEXT I
100 GOTO 100
```

**160. Vegyük észre:**

– Színezés előtt a kurzor színét be kell állítani.

A programok:

a)

```
10 GRAPHIC 1,1
20 SONCLR
30 DRAW 1,150,50 TO 200,150
40 DRAW 1,200,150 TO 100,150
50 DRAW 1,100,150 TO 150,50
60 END
```

b)

```
10 GRAPHIC 1,1
20 SONCLR
30 DRAW 1,150,50 TO 200,150
40 DRAW 1,200,150 TO 100,150
50 DRAW 1,100,150 TO 150,50
60 COLOR 1,1,7
70 PAINT 1,150,60
80 END
```

**161. Vegyük észre:**

– Az átlók kezdő- és végpontjának koordinátái a kör adataiból egyszerűen számíthatók.

A program:

```
10 GRAPHIC 1,1
20 SONCLR
30 CIRCLE 1,140,100,40
40 DRAW 1,100,100 TO 180,100
50 DRAW 1,140,60 TO 140,140
60 END
```

**162. Vegyük észre:**

– A koncentrikus körök középpontja megegyezik, csak sugaruk különböző.

A program:

```
10 GRAPHIC 1,1
20 DRAW 1,160,100
30 CIRCLE 1,160,100,40
40 CIRCLE 1,160,100,50
50 CIRCLE 1,160,100,60
60 CIRCLE 1,160,100,70
70 CIRCLE 1,160,100,80
```



**163. Vegyük észre:**

- A szabályos hatszög minden oldala egyenlő, minden szöge egyenlő, a szomszédos átlók által bezárt szög 60 fokos.

A programok:

**a)**

```
10 GRAPHIC 1,1
20 SGNCLR
30 CIRCLE 1,160,100,40,40,,,,60
40 END
```

**b)**

```
10 GRAPHIC 1,1
20 SGNCLR
30 CIRCLE 1,160,100,40,40,,,,60
40 CIRCLE 1,160,100,40
50 END
```

**164. Vegyük észre:**

- A téglalap rajzolása legegyszerűbb a BOX utasítással.
- Az átlók DRAW-val húzhatók.

A program:

```
10 GRAPHIC 1,1
20 BOX 1,70,50,200,150
30 DRAW 1,70,50 TO 200,150
40 DRAW 1,70,150 TO 200,50
50 END
```

**165. Vegyük észre:**

- A feladat megoldása némi előzetes rajzolást igényel.
- Kockás papíron vonalzóval rajzolj egy kockát, amelynek csúcsai egy-egy rácspontra esnek.
- Tervezd meg, hogy egy-egy rácspont a képernyőn hány pontnak felel meg, majd állapítsd meg a csúcsok koordinátáit!

A program:

```
10 GRAPHIC 1,1
20 BOX 1,100,130,180,50
30 DRAW 1,100,50 TO 120,30
40 DRAW 1,120,30 TO 200,30
45 DRAW 1,200,30 TO 180,50
50 DRAW 1,200,30 TO 180,50
60 DRAW 1,200,30 TO 200,110
70 DRAW 1,200,110 TO 180,130
80 END
```

### 166. Vegyük észre:

- Megint célszerű elővenni a kockás lapot, és azon megtervezni az ábrát.
- A feliratokat csak CHAR utasítással helyezhetjük a képernyőre. A koordináták ekkor 0...40 és 0...25 lehetnek.

A program:

```
10 GRAPHIC 1,1
11 FOR I=1 TO 39
12 CHAR 1,11,2,"LAKASUNK"
13 NEXT
20 BOX 1,30,50,290,190
30 DRAW 1,110,50 TO 110,190
40 DRAW 1,190,50 TO 190,190
50 DRAW 1,150,90 TO 150,190
60 DRAW 1,110,90 TO 190,90
70 DRAW 1,30,150 TO 110,150
80 CHAR 1,15,9,"ELOSZOBA"
90 CHAR 1,6,14,"SZOBA"
100 CHAR 1,5,21,"ERKELY"
110 CHAR 1,26,14,"SZOBA"
120 CHAR 1,19,14,"FUR-"
130 CHAR 1,19,15,"DO"
140 CHAR 1,14,14,"K.HA"
```

### 3.13 Zene

Kódtáblázat:

A zenei hang	A hang kódja
A	7
H	118
C	169
D	262
E	345
F	383
G	453
A	516
H	571
C	596
D	643
E	685
F	704
G	739
A	770
H	798
C	810
D	834
E	854
F	864

## Kódtáblázat:

A zenei hang	A hang kódja
G	881
A	897
H	911
C	917
D	929
E	939
F	944
G	953

## VOL X

A hangerőt állítja be. X=0 kikapcsolás, X=7 maximális hangerő.

## SOUND A, B, C

A hanggenerátor beállítása:

A: 1–3 lehet, a hanggenerátor megjelölése.

B: a zenei hangnak megfelelő kód a táblázatból.

C: a hang megszólaltatásának hossza. Értéke 0–1015 lehet. Egy egység a másodperc 1/50. részét jelenti.

## 167. Vegyük észre:

– A táblázat adatait nem érdemes megtanulni. Ha zenélni szeretnél, vedd elő, és ebből könnyen dolgozhatsz!

A program:

```
10 PRINT "Q"  
20 VOL 7  
30 SOUND 1,770,100  
50 END
```

## 168. Vegyük észre:

– A program megoldható pl.:

1. A hang megszólaltatása után GET A\$. Ha A\$ nem "V", szóljon a következő hang.
2. A hangmagasságot változóként kezelem. Ha a változó páratlan szám, mély, ha páros szám, akkor a magas hang kódjával lesz egyenlő. Így csak egy GET A\$ utasításra van szükség.

A programok:

### 1.

```
10 PRINT "Q"  
20 VOL 7  
30 SOUND 1,810,50  
40 GET A$  
50 IF A$="V" THEN GOTO 80  
60 SOUND 1,817,50  
70 GET A$  
80 IF A$<>"V" THEN GOTO 30  
90 END
```



## 2.

```
10 PRINT"□"  
20 VOL 7  
30 I=0  
40 I=I+1  
50 IF I=INT(I/2)*2 THEN X=817 : GOTO 70  
60 X=810  
70 SOUND 1,X,50  
80 GET A$  
90 IF A$<>"V" THEN GOTO 40  
100 END
```

### 169. Vegyük észre:

- Mivel az egyes hangok kódszámai nem szabályosan követik egymást, ezért célszerű azokat DATA-ban tárolni.

A program:

```
10 PRINT"□"  
20 DIM A(28)  
30 VOL 7  
40 FOR I=1 TO 28  
50 READ A(I)  
60 NEXT I  
70 FOR I=1 TO 28  
80 SOUND 1,A(I),25  
90 NEXT I  
100 DATA 7,118,169,262,345,383,453,516,571,596,643,685  
110 DATA 770,798,810,834,854,864,881,897,911,917,929,9
```

### 170. Vegyük észre:

- Két hanggenerátort egyszerre kell működtetni.

A program:

```
10 PRINT"□"  
20 VOL 7  
30 SOUND 1,897,50  
40 SOUND 2,770,25  
50 GOTO 30
```

### 171. Vegyük észre:

- Nemcsak a koordináták véletlenszerűek, hanem az is, hogy vízszintes vagy függőleges vonalakat akarok-e húzni.
- Vízszintes vonalat húz a gép, ha csak az X koordináták változnak.
- Függőleges vonalat húz a gép, ha csak az Y koordináták változnak.



### 173. Vegyük észre:

- Nem csak a hangmagasságokat, hanem a ritmust is tárolnunk kell.
- Ügyelni kell arra, hogy a szüneteket is megtartsuk a megfelelő helyen.

A program:

```
10 DIM A(40):DIM B(40)
20 PRINT"□"
30 PRINT"██████████PAL KATA PETER JO REGGELT██████"
40 FOR I=1 TO 40
50 READ A(I)
60 READ B(I)
70 NEXT I
80 VOL 7
90 FOR I=1 TO 40
100 SOUND 1,A(I),B(I)
110 NEXT I
120 END
130 DATA 810,25,810,12,834,12,854,25,810,25,834,25,798,25,810,25,0,25
140 DATA 854,25,854,12,864,12,881,25,884,25,864,25,834,25,854,25,0,25
150 DATA 881,12,881,12,881,12,881,12,881,25,897,12,881,12
160 DATA 864,25,881,12,864,12,854,25,0,25
170 DATA 854,12,854,12,854,12,854,12,854,25,864,12,854,12,834,25,881,
25,810,25
```

### 3.14 Karakterfüzér függvények

Karakterkódok táblázata

CHR\$ kód	Hatás	CHR\$ kód	Hatás
0		19	HOME
1		20	DEL
2		21	
3		22	
4		23	
5	fehér	24	
6		25	
7		26	
8	SHIFT+C=kikapcs.	27	ESC
9	SHIFT+C=bekapcs.	28	piros
10		29	kurzor jobbra
11		30	zöld
12		31	kék
13	RETURN	32	szóköz
14	karakterkészlet csere kis/nagy betű	33	!
15		34	"
16		35	#
17	kurzor le	36	\$
18	RVS bekapcs.	37	"/
		38	&



CHR\$ kód	Hatás	CHR\$ kód	Hatás
39	9	88	X
40	(	89	Y
41	)	90	Z
42	*	91	[
43	+	92	£
44	»	93	]
45	-	94	↑
46	"	95	←
47	/	96	-
48	Ø	97	↑
49	1	98	
50	2	99	-
51	3	100	-
52	4	101	-
53	5	102	-
54	6	103	
55	7	104	
56	8	105	~
57	9	106	~
58	:	107	~
59	;	108	L
60	<	109	\
61	=	110	/
62	>	111	┐
63	?	112	└
64	@	113	⊙
65	A	114	-
66	B	115	⊙
67	C	116	
68	D	117	~
69	E	118	X
70	F	119	□
71	G	120	⊙
72	H	121	
73	I	122	◆
74	J	123	+
75	K	124	⋮
76	L	125	
77	M	126	π
78	N	127	▼
79	O	128	
80	P	129	narancs
81	Q	130	villogó be
82	R	131	villogó ki
83	S	132	
84	T	133	f1
85	U	134	f3
86	V	135	f5
87	W	136	f7

CHR\$ kód	Hatás	CHR\$ kód	Hatás
137	f2	164	—
138	f4	165	—
139	f6	166	■
140	HELP	167	—
141	RETURN	168	■
142	karakterkészlet csere kis/nagy betű	169	■
143		170	—
144	fekete	171	—
145	kurzor fel	172	■
146	RVS kikapcs.	173	—
147	CLEAR	174	—
148	INST	175	—
149	barna	176	—
150	sárgászöld	177	—
151	rózsaszín	178	—
152	kékeszöld	179	—
153	világoskék	180	—
154	sötétkék	181	■
155	világoszöld	182	■
156	bíbor	183	—
157	kurzor balra	184	■
158	sárga	185	■
159	enciánkék	186	—
160	szóköz	187	■
161	■	188	■
162	■	189	—
163	—	190	■
		191	■

### CHR\$(X)

Ha X helyére a táblázat értékeit helyettesítjük, a táblázatban megjelölt hatás jön létre.

### LEFT\$(X\$,A)

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

A\$ értéke az X\$-ban levő sztring balról számított A db karaktere lesz.

### LEN(X\$)

A=LEN(X\$)

Az A értéke annyi lesz, ahány karakterből áll az X\$-ban levő sztring.

### RIGHT\$(X\$,A)

A\$=RIGHT\$(X\$,A)

A\$ értéke az X\$-ban levő sztring jobbról számított A db karaktere lesz.

### STR\$(X)

A\$=STR\$(X)

Az X változóban levő számot az A\$ sztringváltozóba helyezi.

174. Vegyük észre:

- Véletlenszerűen eddig csak számokat tudtunk kiírni.
- A karakterkódok 65-től 90-ig tartalmazzák a betűket.
- Ha a véletlenszámokat ilyen tartományon belül generáltatjuk, majd a számot CHR\$(A) függvénybe helyettesítjük, egy betűt kapunk.

A program:

```
10 PRINT "□"  
20 PRINT "#####VELETLENSZERU BETUKI IRATAS#####"  
30 FOR I=1 TO 100  
40 A=INT(RND(1)*26)+65  
50 A#=CHR$(A)  
60 PRINT A#;  
70 NEXT I  
80 END
```

175. Vegyük észre:

- Idézőjel csak úgy tudunk kiírni, hogy ha a CHR\$(34)-et használjuk, mert ez az idézőjel kódja.

A program:

```
10 PRINT "□"  
15 PRINT "#####"  
20 A#=CHR$(34)  
30 PRINT A#;"A TUDAS GYOKERE KESERU, GYUMOLOSE EDES";A#  
40 PRINT "#####(KAPACZAI)#####"
```

176. Vegyük észre:

- Ha megpróbáljuk lefuttatni az 1. programot, furcsa módon nem az összes karaktert kapjuk meg, hanem a képernyőn változások történnek. Ez a 129... 159 közötti kódok jelentésének következménye.
- Az előzőekből okulva, ezeket a kódszámokat célszerű kihagyni. Pl. a 2. program futtatásával.

A programok:

1.

```
10 PRINT "□"  
20 PRINT "####A KARAKTERKESZLET KIRATASA####"  
30 FOR I=65 TO 101  
40 A#=CHR$(I)  
50 PRINT A#;  
60 NEXT I  
70  
80 END
```



## 2.

```
10 PRINT " "
20 PRINT "##### KARAKTERKESZLET KIRATASA ####"
30 FOR I=33 TO 128
40 A#=CHR$(I)
50 PRINT A#;
60 NEXT I
70 FOR I=161 TO 191
80 A#=CHR$(I)
90 PRINT A#;
100 NEXT I
110 END
```

### 177. Vegyük észre:

- 4 – Ha a betű titkosírásbeli kódja megegyezik a karakterkódjával, akkor azt nem kell tárolni, csak a CHR(X) függvénnyel megfejteni.
- Ha hosszabb, de 255 karakternél rövidebb szöveget akarunk dekódolni, jól használható az összefűzés művelete.
- A program leállítására olyan karaktert célszerű választani, amely szövegben nem fordul elő. Pl.:

A program:

```
10 PRINT " "
20 PRINT "#####KODOLO PROGRAM####"
30 PRINT "#####BETUROL SZOVEGRE ####"
40 B#=""
50 INPUT A
60 IF A=64 THEN GOTO 100
70 A#=CHR$(A)
80 B#=B#+A#
90 GOTO 50
100 PRINT B#
110 INPUT "MOLGOZUNK MEG" V#
120 IF V#="I" THEN GOTO 40
130 END
```

### 178. Vegyük észre:

- Fontos tudnunk, hogy az illetőnek hány tagú a neve.
- A nevet tagonként bekérve, azok első betűjét kell kiíratni.
- Probléma, hogy ha a név kettős betűvel kezdődik, akkor is csak az első betűt írja ki a gép.

A program:

```
10 PRINT " "
20 PRINT "#####MONOGRAM KESZITO PROGRAM####"
30 INPUT "HANY TAGU A NEV" X
40 FOR I=1 TO X
```

```

50 INPUT A$(I)
60 B$(I)=LEFT$(A$(I),1)
70 NEXT I
80 PRINT "A MONOGRAMMJA"
90 FOR I=1 TO X
100 PRINTB$(I);". ";
110 NEXT I
120 END

```

**179. Vegyük észre:**

- A szót INPUT utasítással bekérve a hossza egyszerűen megállapítható.
- A program:

```

10 PRINT " "
20 PRINT "SZOHOSSZ MEGALLAPITO PROGRAM"
30 INPUT A$
40 X=LEN(A$)
50 PRINT
60 PRINT "A BEGEPELT SZO HOSSZA" X; "BETU"
70 END

```

**180. Vegyük észre:**

- Ha a szöveget a GET utasítással kérjük be, akkor a szöveg végét olyan karakterrel érdemes jelezni, amely egy szövegrészben nem fordul elő. Pl.:
- A program:

```

10 PRINT " "
20 PRINT "SZOHOSSZ MEGALLAPITO PROGRAM"
30 GET A$
40 IF A$="" THEN GOTO 30
50 IF A$="*" THEN GOTO 30
60 PRINT A$;
70 B$=B$+A$
80 GOTO 30
90 X=LEN(B$)
100 PRINT
110 PRINT "A BEGEPELT SZO HOSSZA" X; "BETU"
120 END

```

**181. Vegyük észre:**

- Ikes igéről beszélünk, ha az egyes szám harmadik személyben az ige ik-re végződik. Ezt kell vizsgálni.
- A program:

```

10 PRINT " "
20 PRINT "IK-ES IGET GEPELUNK -E"
30 INPUT "AZ IGE 1. SZAM 3. SZEMELYBEN " A$

```

```

40 B$=RIGHT$(A$,2)
50 IF B$="IK" THEN PRINT"IK-ES IGET GEPELT." : GOTO70
60 PRINT"NEM IK-ES IGET GEPELT."
70 END

```

**182. Vegyük észre:**

- Az adott szót az elsőtől az utolsó betűig betűnként meg kell vizsgálni.
- Ehhez szükséges tudnunk, hogy a szó hány karaktert tartalmaz.

A program:

```

10 PRINT" "
20 PRINT"#####HANY E BETUT TARTALMAZ A SZO####"
30 INPUT"A SZO" : A$
40 X=LEN(A$)
50 FOR I=1 TO X
60 B$=MID$(A$,I,1)
70 IF B$="E" THEN K=K+1
80 NEXT I
90 PRINT"#####A SZO" : K : "DARAB E BETUT TARTALMAZ."
100 END

```

**183. Vegyük észre:**

- Az előző feladattól annyiban különbözik, hogy most a szót kétbetűnként kell megvizsgálni.

A program:

```

10 PRINT" "
20 PRINT"#####HANY LY-T TARTALMAZ A SZO####"
30 INPUT"A SZO" : A$
40 X=LEN(A$)
50 FOR I=1 TO X
60 B$=MID$(A$,I,2)
70 IF B$="LY" THEN K=K+1
80 NEXT I
90 PRINT"#####A SZO" : K : "DARAB LY-T TARTALMAZ."
100 END

```

**184. Vegyük észre:**

- Ha a mondatot betűnként GET utasítással kérjük be, akkor célszerű az összefűzést alkalmazni.
- A mondat végét pont, kérdőjel vagy felkiáltójel jelzi. Ezt kihasználhatjuk a bekérés megállítására.
- A mondat betűnként való kiíratása érdekesebb, ha az egyes betűk között érzékelhető szüneteket tartunk.





**186. Vegyük észre:**

- Ha egy név kettős betűvel kezdődik, akkor a monogramnak is ezt a kettős betűt kell tartalmaznia.

A program:

```

10 PRINT "□"
20 PRINT "■■■■■■■■■■JAVITOTT MONOGRAM KESZITO■■■■■■■■■■"
30 PRINT "HA A NEV KETTOS BETUVEL KEZDODIK, KEREM"
40 PRINT "AZT A * BILLENTYU LENYOMASAVAL A NEV"
50 PRINT "ELOTT JELEZZE!■■■■■■■■■■"
60 INPUT "HANY TAGU A NEV" ; X
70 PRINT "A NEV TAGONKENT■■■■■■■■■■"
80 FOR I=1 TO X
90 INPUT A$
100 B$(I)=LEFT$(A$,1)
110 IF B$(I)<>"*" THEN GOTO 130
120 B$(I)=MID$(A$,2,2)
130 NEXT I
140 PRINT
150 FOR I=1 TO X
160 PRINT B$(I) ; " " ;
170 NEXT I
180 END
190 END

```

**187. Vegyük észre:**

- Számokat sztringváltozóba csak úgy tehetünk, ha azokat idézőjel közé írtuk.
- Ezért egy ciklus ciklusváltozójának aktuális értékét nem is tehetjük egyszerűen egy sztringváltozóba.
- A feladat megoldását az STR\$(X) sztringfüggvény adja.
- Vigyázni kell arra, hogy ezekkel a számokkal már nem végezhetők a matematikai műveletek, mert ezek már mint sztringek szerepelnek, ha azokat a változókból íratjuk ki.

A program:

```

10 DIM A$(20)
20 PRINT "□"
30 PRINT "STRINGTOMBEN SZAMOK TAROLASA CIKLUSSAL"
40 FOR I=1 TO 20
50 A$(I)=STR$(I)
60 PRINT "A$(" ; I ; ")=" ; A$(I)
70 NEXT I
80 END

```

**188. Vegyük észre:**

- A grafikus üzemmódban a képernyő tetszőleges helyére a CHAR utasítással tudunk szöveget íratni.

- A szöveget nem csak idézőjelek közé írva, hanem sztringváltozóként is megadhatom.

A program:

```
10 GRAPHIC 2,1
20 SONDCLR
30 CHAR 1,13,1,"LEVELCIMZES"
40 BOX 1,40,20,280,130
50 BOX 1,240,30,270,50
60 DRAW 1,160,73 TO 272,73
70 DRAW 1,184,89 TO 272,89
80 DRAW 1,184,90 TO 272,90
90 DRAW 1,184,105 TO 272,105
100 DRAW 1,184,121 TO 272,121
110 PRINT"A CIMZES"
120 INPUT"NEV "N$
130 X=LEN(N$)
140 IF X>14 THEN PRINT"TUL HOSSZU" : GOTO 120
150 INPUT"VAROS "V$
160 X=LEN(V$)
170 IF X>11 THEN PRINT"TUL HOSSZU" : GOTO 150
180 INPUT"UTCA "U$
190 X=LEN(U$)
200 IF X>11 THEN PRINT"TUL HOSSZU" : GOTO 180
210 INPUT"IRANYITOSZAM "I$
220 X=LEN(I$)
230 IF X<>4 THEN PRINT"ROSSZ ADAT" : GOTO 210
240 CHAR 1,20,8,N$
250 CHAR 1,23,10,V$
260 CHAR 1,23,12,U$
270 CHAR 1,23,14,I$
280 END
```

### 3.15 Logikai műveletek (OR, AND, NOT) és ON...GOTO

#### OR (VAGY)

IF 1. feltétel OR 2. feltétel THEN ...

Ha a két feltétel közül legalább az egyik igaz, az állítás logikai értéke igaz.

#### AND (ÉS)

IF 1. feltétel AND 2. feltétel THEN ...

Ha a két feltétel közül mindkettő igaz, az állítás logikai értéke igaz.

#### NOT (NEM)

IF NOT (feltétel) THEN ...

Ha a zárójelben levő feltétel hamis, az állítás logikai értéke igaz.

#### ON...GOTO

ON X GOTO ...



Az ON...GOTO jól használható például menük esetében. A GOTO után írt címek közül a program futása annyiadíkon folytatódik, amennyi az X értéke volt.

**189. a) Vegyük észre:**

- Egy szám páros, ha osztható kettővel.
- Előfordulhat, hogy minden generált szám páros, így a páros számok tömbjét 50-re kell dimenzionálni.

A program:

```
10 DIM A(50)
20 K=0
30 PRINT "□"
40 PRINT "██████████20-200 KÖZÖTTI VELETLENSZÁMOK ████████"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A=INT(A/2)*2 THEN K=K+1 : A(K)=A
90 NEXT I
100 PRINT
110 PRINT "██████████PÁROSÁK VOLTAK: □"
120 FOR I=1 TO K
130 PRINT A(I);
140 NEXT I
150 END
```

**b) Vegyük észre:**

- Lehet, hogy minden szám nagyobb lesz 100-nál, így a dimenzionálásra az előbbiek érvényesek.

A program:

```
10 DIM A(50)
20 K=0
30 PRINT "□"
40 PRINT "██████████20-200 KÖZÖTTI VELETLENSZÁMOK ████████"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A>110 THEN K=K+1 : A(K)=A
90 NEXT I
100 PRINT
110 PRINT "██████████110-NEL NAGYOBBÁK VOLTAK: □"
120 FOR I=1 TO K
130 PRINT A(I);
140 NEXT I
150 END
```

**c) Vegyük észre:**

- Páratlan a szám, ha nem osztható kettővel.
- Páratlan és ötten osztható egy szám, ha mindkét feltétel teljesül rá.
- A feladat megoldható két IF-fel (1.) és megoldható az AND használatával is (2.).

A programok:

**1.**

```
10 DIM A(50)
20 K=0
30 PRINT "Q"
40 PRINT "#####20-200 KOZOTTI VELETLENSZAMOK#####"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A=INT(A/2)*2 THEN GOTO 100
90 IF A=INT(A/5)*5 THEN K=K+1 : A(K)=A
100 NEXT I
110 PRINT
120 PRINT "#####PARATLANOK ES OTTEL OSZTHATOK VOLTAK:#####"
130 FOR I=1 TO K
140 PRINT A(I);
150 NEXT I
160 END
```

**2.**

```
10 DIM A(50)
20 K=0
30 PRINT "Q"
40 PRINT "#####20-200 KOZOTTI VELETLENSZAMOK#####"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A=INT(A/2)*2 AND A=INT(A/5)*5 THEN K=K+1 : A(K)=A
90 NEXT I
100 PRINT
110 PRINT "#####PARATLANOK ES OTTEL OSZTHATOK VOLTAK:#####"
120 FOR I=1 TO K
130 PRINT A(I);
140 NEXT I
150 END
```

**d) Vegyük észre:**

- Egy számra akkor igaz, hogy 80-nál nagyobb vagy 4-gyel osztható, ha a két állítás közül legalább az egyik igaz.

- A feladat megoldható két IF-fel (1.) vagy az OR utasítás felhasználásával (2.).

A programok:

1.

```
10 DIM A(50)
20 K=0
30 PRINT "1"
40 PRINT "20-200 KÖZÖTTI VELETLENSZÁMOK"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A>80 THEN K=K+1 : A(K)=A : GOTO 100
90 IF A=INT(A/4)*4 THEN K=K+1 : A(K)=A
100 NEXT I
110 PRINT
120 PRINT "80-NAL NAGYOBB VAGY NEGGYEL OSZTHATO:"
130 FOR I=1 TO K
140 PRINT A(I);
150 NEXT I
160 END
```

2.

```
10 DIM A(50)
20 K=0
30 PRINT "1"
40 PRINT "20-200 KÖZÖTTI VELETLENSZÁMOK"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A>80 OR A=INT(A/4)*4 THEN K=K+1 : A(K)=A
90 NEXT I
100 PRINT
110 PRINT "80-NAL NAGYOBB, VAGY NEGGYEL OSZTHATO:"
120 FOR I=1 TO K
130 PRINT A(I);
140 NEXT I
150 END
160 NEXT I
```

e) Vegyük észre:

- Az állítás akkor igaz, ha mind a két feltétel teljesül.
- A feladat megoldható két IF-fel (1.) vagy AND és NOT segítségével is (2.).



## A programok:

### 1.

```
10 DIM A(50)
20 K=0
30 PRINT "Q"
40 PRINT "#####20-200 KOZOTTI VELETLENSZAMOK#####"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A<=100 THEN GOTO 100
90 IF A<=150 THEN K=K+1 : A(K)=A
100 NEXT I
110 PRINT
120 PRINT "###100-NAL NAGYOBB, ES 150-NEL NEM NAGYOBB:###"
130 FOR I=1 TO K
140 PRINT A(I);
150 NEXT I
160 END
```

### 2.

```
10 DIM A(50)
20 K=0
30 PRINT "Q"
40 PRINT "#####20-200 KOZOTTI VELETLENSZAMOK#####"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A>100 AND NOT A>150 THEN K=K+1 : A(K)=A
90 NEXT I
100 PRINT
110 PRINT "###100-NAL NAGYOBB, ES 150-NEL NEM NAGYOBB:###"
120 FOR I=1 TO K
130 PRINT A(I);
140 NEXT I
150 END
```

#### f) Vegyük észre:

- A feladatot megoldhatjuk két IF-fel (1.) vagy a NOT és az AND használatával is (2.).

A programok:

1.

```
10 DIM A(50)
20 K=0
30 PRINT " "
40 PRINT "20-200 KOZOTTI VELETLENSZAMOK"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF A>90 THEN GOTO 100
90 IF A<>INT(A/2)*2 THEN K=K+1 : A(K)=A
100 NEXT I
110 PRINT
120 PRINT "NEM IGAZ, HOGY NAGYOBB 90-NEL ES PAROS:"
130 FOR I=1 TO K
140 PRINT A(I);
150 NEXT I
160 END
```

2.

```
10 DIM A(50)
20 K=0
30 PRINT " "
40 PRINT "20-200 KOZOTTI VELETLENSZAMOK"
50 FOR I=1 TO 50
60 A=INT(RND(1)*181)+20
70 PRINT A;
80 IF (A>90 AND A<>INT(A/2)*2) THEN K=K+1 : A(K)=A
90 NEXT I
100 PRINT
110 PRINT "NEM IGAZ, HOGY 90-NEL NAGYOBB ES PARATLAN:"
120 FOR I=1 TO K
130 PRINT A(I);
140 NEXT I
150 END
160 PRINT A(I);
170 NEXT I
```

190. Vegyük észre:

- A program az előzőekből gyakorlatilag összeállítható.
- Mivel a program így hat különböző feladat megoldására alkalmas, célszerű a menü használata.
- Mivel mindegyik feladat tartalmazza a véletlenszámok generálását, ezt

- elég egy helyre beiktatni, mielőtt a speciális feladatokra térne át a program.
- Sok IF utasítást takaríthatunk meg, ha használjuk a logikai műveleteket (AND, OR, NOT).
- Hat IF utasítást takaríthatunk meg, ha a speciális feladatokra való elágazásnál az ON X GOTO utasítást használjuk.
- Mivel a feltételeknek megfelelő számokat tömbben gyűjtjük, és azokat ki is kell íratnunk, a kiíró ciklust érdemes olyan helyre tenni, ahová a program a véletlenszámok válogatása után visszatérhet.
- A számok kiírása előtt célszerű azt is kiírni, hogy milyen feltételek alapján választottuk ki ezeket az értékeket.

A program:

```

10 DIM A(50)
20 K=0
30 PRINT "1"
40 PRINT "20-200 KÖZÖTTI VELETLENSZÁMOK"
50 PRINT "1. PÁROSOK"
60 PRINT "2. 110-NEL NAGYOBBOK"
70 PRINT "3. PARATLANOK ES OTTEL OSZTHATOK"
80 PRINT "4. 80-NAL NAGYOBB, VAGY 4-GYEL OSZTHATOK"
90 PRINT "5. 100-NAL NAGYOBB ES 150-NEL NEM NAGYOBB"
100 PRINT "6. NEM IGAZ, HOGY NAGYOBB 90-NEL ES PÁROS"
110 INPUT "MELYIK MÓDON DOLGOZUNK" : X
120 FOR I=1 TO K
130 PRINT "20-200 KÖZÖTTI VELETLENSZÁMOK"
140 FOR I=1 TO 50
150 A=INT(RND(1)*181)+20
160 PRINT A:
170 ON X GOTO 260,290,320,350,380,410
180 K=K+1 : A(K)=A
190 NEXT I
200 PRINT
210 PRINT "A#:" : A#:""
220 FOR I=1 TO K
230 PRINT A(I):
240 NEXT I
250 END
260 IF A=INT(A/2)*2 THEN GOTO 180
270 A#="PÁROSOK:"
280 GOTO 180
290 IF A>110 THEN GOTO 180
300 A#="110-NEL NAGYOBBOK:"
310 GOTO 180
320 IF A<>INT(A/2)*2 AND A=INT(A/5)*5 THEN GOTO 180
330 A#="PARATLAN ES OTTEL OSZTHATOK"

```



```

340 GOTO 180
350 IF A>80 OR A=INT(A/4)*4 THEN GOTO 180
360 A$="80-NAL NAGYOBB VAGY 4-GYEL OSZTHATO"
370 GOTO 180
380 IF A>100 AND NOT A>150 THEN GOTO 180
390 A$="100-NAL NAGYOBB ES 150-NEL NEM NAGYOBB"
400 GOTO 180
410 IF NOT(A>90 AND A=INT(A/2)*2) THEN GOTO 180
420 A$="NEM IGAZ, HOGY NAGYOBB 90-NEL ES PAROS"
430 GOTO 180

```

### 191. Vegyük észre:

- Az adatokat érdemes DATA-ban tárolni, így a kikapcsoláskor azok nem vesznek el.
- Egy tanuló háromnál több szakkörnek nem lehet tagja.
- Ha a tanulók szakköri elfoglaltságához könnyen szeretnénk hozzáférni, azokat célszerű kétdimenziós tömbben tárolni. Az egyik index a tanulóra, a másik a szakkör sorszáma utal.
- A fel nem használt szakköri lehetőségeket jelölni kell, pl. Ø-val.

#### a) Vegyük észre:

- A sportversenyen induló csapatba elsősorban a sportolók jelölhetők.

A program:

```

10 PRINT " "
20 PRINT "HÁROM SZAKKÖR AZ ORS TAGJAINAK JELELESE VETELKEDŐKRE"
30 FOR I=1 TO 10
40 READ A$(I)
50 PRINT A$(I)
60 FOR J=1 TO 3
70 READ B$(I,J)
80 PRINT TAB(10);B$(I,J);", "
90 NEXT J
100 PRINT
110 NEXT I
120 PRINT "A SPORTVERSENYRE JAVASOLTAK"
130 FOR I=1 TO 10
140 FOR J=1 TO 3
150 IF B$(I,J)="SPORT" THEN K=K+1 : A$(K)=A$(I)
160 NEXT J
170 NEXT I
180 PRINT
190 FOR I=1 TO K
200 PRINT A$(I),
210 NEXT I
220 END
230 DATA ZSUZSA,ENEKKAR,MÁTEK,SPORT,EVA,RAJZ,Ø,Ø

```

```

240 DATA LACI,OROSZ,MATEK,0,BELA,KRESZ,FIZIKA,0
250 DATA ROBI,SPORT,KRESZ,0,KATI,ENEKKAR,RAJZ,KEMIA
260 DATA ANETT,MATEK,RAJZ,0,JOCI,KRESZ,SPORT
270 DATA FERI,SPORT,RAJZ,0

```

**b) Vegyük észre:**

- KRESZ-vetélkedőre elsősorban azok menjenek, akik KRESZ-szakkörre jártak.

**A program:**

```

10 PRINT " "
20 PRINT "XXXXXXXXXX AZ ORS TAGJAINAK JELOLESE VETELKEDOKRE "
30 FOR I=1 TO 10
40 READ A$(I)
50 PRINT A$(I)
60 FOR J=1 TO 3
70 READ B$(I,J)
80 PRINT TAB(10);B$(I,J);";"
90 NEXT J
100 PRINT
110 NEXT I
120 PRINT "XXXXXXXXX KRESZVETELKEDOKRE JAVASOLTAK "
130 FOR I=1 TO 10
140 FOR J=1 TO 3
150 IF B$(I,J)="KRESZ" THEN K=K+1:A$(K)=A$(I)
160 NEXT J
170 NEXT I
180 PRINT
190 FOR I=1 TO K
200 PRINT A$(I),
210 NEXT I
220 END
230 DATA ZSUZSA,ENEKKAR,MATEK,SPORT,EVA,RAJZ,0,0
240 DATA LACI,OROSZ,MATEK,0,BELA,KRESZ,FIZIKA,0
250 DATA ROBI,SPORT,KRESZ,0,KATI,ENEKKAR,RAJZ,KEMIA
260 DATA ANETT,MATEK,RAJZ,0,JOCI,KRESZ,SPORT,0
270 DATA FERI,SPORT,RAJZ,0

```

**c) Vegyük észre:**

- Nép daléneklési versenyre azt célszerű jelölni, aki énekkarra járt.

**A program:**

```

10 PRINT " "
20 PRINT "XXXXXXXXXX AZ ORS TAGJAINAK JELOLESE VETELKEDOKRE "
30 FOR I=1 TO 10
40 READ A$(I)
50 PRINT A$(I)

```



```

60 FOR J=1 TO 3
70 READ B$(I,J)
80 PRINT TAB(10);B$(I,J);";"
90 NEXT J
100 PRINT
110 NEXT I
120 PRINT "NEPDALENEKLESES VERSENYRE JAVASOLTAK"
130 FOR I=1 TO 10
140 FOR J=1 TO 3
150 IF B$(I,J)="ENEKKAR" THEN K=K+1:A$(K)=A$(I)
160 NEXT J
170 NEXT I
180 PRINT
190 FOR I=1 TO K
200 PRINT A$(I),
210 NEXT I
220 END
230 DATA ZSUZSA,ENEKKAR,MATEK,SPORT,EVA,RAJZ,0,0
240 DATA LACI,OROSZ,MATEK,0,BELA,KRESZ,FIZIKA,0
250 DATA ROBI,SPORT,KRESZ,0,KATI,ENEKKAR,RAJZ,KEMIA
260 DATA ANETT,MATEK,RAJZ,0,JOCI,KRESZ,SPORT,0
270 DATA FERI,SPORT,RAJZ,0

```

**d) Vegyük észre:**

- Mivel egyéni művészeti vetélkedőről van szó, ezért olyan pájtásokat célszerű jelölni, akik rajzsakkörre és énekkarra is járnak.
- Több IF-et megtakaríthatunk, ha használjuk a logikai műveleteket.

A program:

```

10 PRINT " "
20 PRINT "2 ORS TAGJAINAK JELOLESE VETELKEDOKRE"
30 FOR I=1 TO 10
40 READ A$(I)
50 PRINT A$(I);
60 FOR J=1 TO 3
70 READ B$(I,J)
80 PRINT TAB(10);B$(I,J);";"
90 NEXT J
100 PRINT
110 NEXT I
120 PRINT "EGYENI MUVESZETI VETELKEDOKRE JAVASOLTAK"
130 FOR I=1 TO 10
140 L=0
150 FOR J=1 TO 3
160 IF B$(I,J)="RAJZ" OR B$(I,J)="ENEKKAR" THEN L=L+1
170 IF L=2 THEN K=K+1:A$(K)=A$(I):GOTO 130
180 NEXT J

```





```

260 DATA ROBI,SPORT,KRESZ,0,KATI,ENEKKAR,RAJZ,KEMIA
270 DATA ANETT,MATEK,RAJZ,0,JOCI,KRESZ,SPORT,0
280 DATA FERI,SPORT,RAJZ,0

```

f) A program:

```

10 PRINT "A"
20 PRINT "AZ ORS TAGJAINAK JELOLESE VETELKEDOKRE"
30 FOR I=1 TO 10
40 READ A$(I)
50 PRINT A$(I);
60 FOR J=1 TO 3
70 READ B$(I,J)
80 PRINT TAB(10);B$(I,J);", ";
90 NEXT J
100 PRINT
110 NEXT I
120 PRINT "MATEMATIKAVERSENYRE JAVASOLTAK"
130 FOR I=1 TO 10
140 FOR J=1 TO 3
150 IF B$(I,J)="MATEK" THEN K=K+1:A$(K)=A$(I)
160 NEXT J
170 NEXT I
180 PRINT
190 FOR I=1 TO K
200 PRINT A$(I);
210 NEXT I
220 END
230 DATA ZSUZSA,ENEKKAR,MATEK,SPORT,EVA,RAJZ,0,0
240 DATA LACI,OROSZ,MATEK,0,BELA,KRESZ,FIZIKA,0
250 DATA ROBI,SPORT,KRESZ,0,KATI,ENEKKAR,RAJZ,KEMIA
260 DATA ANETT,MATEK,RAJZ,0,JOCI,KRESZ,SPORT,0
270 DATA FERI,SPORT,RAJZ,0

```

192. A program:

```

10 DIM A$(10)
20 K=0
30 PRINT "A"
40 PRINT "AZ ORS TAGJAINAK JELOLESE VETELKEDOKRE"
50 PRINT "1. SPORTVETELKEDO"
60 PRINT "2. CSAPAT KRESZ VERSENY"
70 PRINT "3. NEPDALENEKLESI VERSENY"
80 PRINT "4. EGYENI MUVESZETI VETELKEDO"
90 PRINT "5. TERMESZETTUDOMANYI (CSAPAT) VETELKEDO"
100 PRINT "6. MATEKVERSENY"

```



```

110 INPUT "MELYIK MODON DOLGOZUNK"?X
120 PRINT "Q"
130 PRINT "MELYIK AZ ORS TAGJAINAK JELOLESE VETELKEDOKRE"
140 FOR I=1 TO 10
150 READ A$(I)
160 PRINT A$(I);
170 FOR J=1 TO 3
180 READ B$(I,J)
190 PRINT TAB(10);B$(I,J);", ";
200 NEXT J
210 PRINT
220 NEXT I
230 FOR I=1 TO 10
235 L=0
240 FOR J=1 TO 3
250 ON X GOTO 350,380,410,440,480,510
260 K=K+1:A$(K)=A$(I)
270 NEXT J
280 NEXT I
290 PRINT
300 PRINT "Q";A$;"Q"
310 FOR I=1 TO K
320 PRINT A$(I),
330 NEXT I
340 END
350 A$="SPORTVERSENYRE JAVASOLTAKQ"
360 IF B$(I,J)="SPORT" THEN GOTO 260
370 GOTO 270
380 A$="KRESZVETELKEDORE JAVASOLTAKQ"
390 IF B$(I,J)="KRESZ" THEN GOTO 260
400 GOTO 270
410 A$="NEPDALENEKLESI VERSENYRE JAVASOLTAKQ"
420 IF B$(I,J)="ENEKKAR" THEN GOTO 260
430 GOTO 270
440 A$="MUVESZETI VETELKEDOKRE JAVASOLTAKQ"
450 IF B$(I,J)="ENEKKAR" OR B$(I,J)="RAJZ" THEN L=L+1
460 IF L=2 THEN K=K+1:A$(K)=A$(I):GOTO 280
470 GOTO 270
480 A$="TERMESZETTUDOMANYI VERSENYRE JAVASOLTAKQ"
490 IF B$(I,J)="BIOLOGIA" OR B$(I,J)="FIZIKA"
    OR B$(I,J)="KEMIA" THEN GOTO 260
500 GOTO 270
510 A$="MATEMATIKA VERSENYRE JAVASOLTAKQ"
520 IF B$(I,J)="MATEK" THEN GOTO 260

```



```

530 GOTO 270
540 DATA ZSUZSA,ENEKKAR,MATEK,SPORT,EVA,RAJZ,0,0
550 DATA LACI,OROSZ,MATEK,0,BELA,KRESZ,FIZIKA,0
560 DATA ROBI,SPORT,KRESZ,0,KATI,ENEKKAR,RAJZ,KEMIA
570 DATA ANETT,MATEK,RAJZ,0,JOCI,KRESZ,SPORT,0
580 DATA FERI,SPORT,RAJZ,0

```

### 3.16 A sornymató használata

#### OPEN le, fe

Nyomatás előtt az OPEN utasítással nyitjuk meg a file-t. A le jelentése: logikai egység száma 1-től 255-ig. A fe jelentése: fizikai egység száma. Ez nyomtató esetén 4 vagy 5.

#### CMD fe

A CMD utasítással átadjuk a vezérlést a számítógépről a nyomtatónak. Ha a CMD utasítás után LIST következik, a gépben levő program listája megjelenik a papíron. Ha a CMD után PRINT utasítás van, a PRINT után írt változó értéke vagy szöveg megjelenik a papíron.

#### CLOSE le

A nyomtatás után a file-t le kell zárni.

#### PRINT le

Ha a CMD utasítást használtuk, a CLOSE utasítás előtt a PRINT le utasítással kell zárunk a csatornát.

#### Vezérlő kódok

A speciális funkciók használata a PRINT le után írt CHR(X) kódokkal történik.

Kód	Funkció
CHR\$(10)	soremelés nyomtatás után
CHR\$(13)	soremelés
CHR\$(8)	áttérés grafikus üzemmódba
CHR\$(14)	áttérés dupla szélességű karakter módba
CHR\$(15)	áttérés normál karakter módba

A leggyakrabban használt vezérlő kódokon kívül a nyomtatók más ilyen kódokat is ismerhetnek. Ezt mindig a gépkönyv alapján állapíthatjuk meg. Ezek a kódok az MPS-801 típusú grafikus nyomtatóra vonatkoznak.

Ha a programnak sornymató outputja van, a következőképpen kell eljárni:

- Az első nyomtatási utasítás előtt a sornymatót meg kell nyitni: OPEN le, fe
- A kiíratás mindig PRINT le, X utasítással történik. A PRINT le utasítás után a PRINT utasításnak megfelelő szintaktikával írhatunk változókat, szövegváltozókat, konstansokat, szöveget, szerkesztő karaktereket.

```

Pl.: PRINT 4,X          PRINT 4,X          PRINT 4,"szöveg"
      PRINT 4,TAB(X);   "szöveg"        PRINT 4, CHR$(13)

```

- Ha a sornymatóra a programban tovább nincs szükség, vagy a program végére értünk, a sornymatót le kell zárni:

CLOSE le

#### PRINT USING

A gép az USING utasítás mögött szereplő maszkoknak megfelelően szerkeszti az utána írt nem szöveges változót. Pl. PRINT USING "####.##";X

A gép a kiírt változót a tizedespontra illeszti. Ha az utasítást sornyomtatón kívánjuk használni, a PRINT helyett PRINT# le, USING . . . használandó.

Ha a változó a tizedesponntól balra több helyiértéken tartalmaz számjegyeket, mint a maszk, a gép a nyomtatási képben a számjegyek helyett a maszknak megfelelő darabszámú csillagot jelenti.

### 193. Vegyük észre:

- Hogy a programot ki tudjuk nyomtatni, ahhoz először az OPEN le,fe utasítással meg kell nyitni a file-t.
- A CMD fe utasítással át kell adni a vezérlést a nyomtatónak.
- A LIST utasítással el kell indítani a program listázását.
- A listázás végén le kell zárni a csatornát, PRINT# le utasítással.
- A CLOSE le utasítással le kell zárnunk a file-t.
- Ezeket az utasításokat nem kell beépíteni a programba, hanem sorszám nélkül, parancsként is alkalmazhatjuk őket.

A program:

```
10 OPEN 4,4
20 CMD 4
```

LIST

```
10 PRINT "□"           10 PRINT#4
20 FOR I=1 TO 10       20 CLOSE 4
30 PRINT "*"
40 NEXT I
50 END
```

### 194. Vegyük észre:

- A Commodore gépekhez használatos nyomtatók általában 80 oszlopos papírra dolgoznak. Így egy sorban 80 db A betű fér el.
- A gép az A betűket akkor írja egymás mellé, ha az A betű után pontosvesszőt használunk.
- A nyomtatás előtt meg kell nyitni a file-t.
- A nyomtatás után a file-t le kell zárni.
- Ha a programban CMD le nélkül használjuk a PRINT# 4 utasítást, annak jelentése egy üres sor lesz.

A program:

```
10 OPEN 4,4
20 FOR I=1 TO 5
30 FOR J=1 TO 80
40 PRINT#4,"A"J
50 NEXT J
60 NEXT I
70 PRINT#4
80 CLOSE 4
90 END
```



## Az eredmény:

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

### 195. Vegyük észre:

- Az egyes karakterekből a gép soronként 80-80 darabot tud rajzolni.
- Üres sort húzhatunk a vezérlőkódok segítségével is. A soremelés kódja CHR\$(13)
- A file-t a nyomtatás előtt meg kell nyitni, a nyomtatás végeztével le kell zárni.

A program:

```
10 OPEN 4,4
20 FOR I=1 TO 4
30 FOR J=1 TO 80
40 PRINT#4,"+";
50 NEXT J
60 PRINT#4,CHR$(13)
70 FOR K=1 TO 80
80 PRINT#4,"*";
90 NEXT K
100 PRINT#4,CHR$(13)
110 NEXT I
120 PRINT#4
130 CLOSE 4
140 END
```

## Az eredmény:

```
+++++
*****
+++++
*****
+++++
*****
+++++
*****
```





A program:

```
10 OPEN 4,4
20 FOR I=1 TO 6
30 IF I=INT(I/2)*2 THEN X=14:GOTO 50
40 X=15
50 PRINT#4,CHR$(X)
60 PRINT#4,"COMMODORE 16"
70 NEXT I
80 PRINT#4
90 CLOSE 4
```

Az eredmény:

COMMODORE 16

COMMODORE 16

COMMODORE 16

COMMODORE 16

COMMODORE 16

COMMODORE 16

198. Vegyük észre:

- A kinyomtatható karakterek a 33–129 és a 161–191 CHR kódok között helyezkednek el.
- A többi karakter kinyomtatása nem célszerű, mert egyes karaktereket a gép vezérlőkódként értelmez, így nem a várt karakterek jelennek meg, hanem áttér valamilyen speciális üzemmódba.

A program:

```
10 FOR I=1 TO 4
20 READ X(I)
30 NEXT I
40 OPEN 4,4
50 FOR J=1 TO 4 STEP 2
60 FOR J=X(J) TO X(J+1)
70 A#=CHR$(I)
80 PRINT#4,TAB(10);I;TAB(30);A#
90 NEXT I
100 NEXT J
110 CLOSE 4
120 DATA 33,129,161,191
130 END
```

Az eredmény egy részlete:

33	!
34	"
35	#
36	\$
37	%
38	&
39	'
40	(
41	)
42	*
43	+
44	,

199. Vegyük észre:

- A fejléc kiírása dupla szélességű üzemmódban történik.
- A szorzótábla elemeinek kiírása előtt vissza kell térni normál karaktermódba.
- Elfogadható tablóformát kapunk a TAB(X) felhasználásával.

A program:

```
10 PRINT "X"
20 PRINT "XXXXXXXXXXXXXXXXX SZORZOTABLA SORNYOMTATORA XXXXX"
30 INPUT " HANYAS SZORZOTABLÁT IRJUK "; X
40 OPEN 4,4
50 PRINT#4,CHR$(14)
60 PRINT#4,TAB(10);X;" SZORZOTABLÁJA"
70 PRINT#4,CHR$(13)
80 PRINT#4,TAB(2);"SZORZANDÓ";TAB(18);"SZORZAT"
90 PRINT#4,CHR$(13);PRINT#4,CHR$(15)
100 FOR I=1 TO 20
110 S=X*I
120 PRINT#4,TAB(12);I;TAB(48);S
130 NEXT I
140 PRINT#4
150 CLOSE 4
160 END
```

200. Vegyük észre:

- A fejlécet dupla szélességű karakterekkel íratjuk.
- Az adatok kinyomtatása normál üzemmódban történik.
- Mivel a telefonszámokon kívül a többi adat nem egyforma hosszú, a TAB(X) használatával nem kapunk szép táblázatokat.
- Ha meg tudjuk oldani, hogy a gép az egyes adatokat mindig ugyanolyan hosszán jelenítse meg, a TAB(X) segítségével újra szép táblázat nyerhető.

A megoldás menete:

- Mivel egy név általában nem hosszabb 30 karakternél, vegyünk fel egy 30 üres karakterből álló füzér változót (50. sor).



- A begépelte nevet, ill. címet fűzzük össze ezzel a 30 üres karakterrel (200–220. sor).
- Az így kapott karakterváltozók első 30 karakterét véve (230–240. sor) mindig ugyanolyan hosszú sztringeket nyomtatunk ki.
- Ha az eredeti sztringváltozó hossza kisebb volt harmincnél, akkor ez a művelet sor kiegészíti azt üres karakterekkel.
- A megoldás korlátai: 30 karakternél hosszabb név, ill. cím nem nyomtatható vele.

A program:

```

10 PRINT "□"
20 INPUT "AZ ORS NEVE " ; H$
30 INPUT "HANY TAGU AZ ORS" ; X
40 DIM N$(X) ; DIM C$(X) ; DIM T$(X)
50 B$ = "
60 PRINT "□"
70 PRINT TAB(4) ; A$ ; " ORS TAGJAINAK ADATAI"
80 FOR I=1 TO X
90 INPUT "NEV " ; N$(I)
100 INPUT "CIM" ; C$(I)
110 INPUT "TELEFONSZAM " ; T(I)
120 NEXT I
130 OPEN 4,4
140 PRINT#4,CHR$(14)
150 PRINT#4,TAB(4) ; A$ ; " ORS TAGJAINAK ADATAI"
160 PRINT#4,CHR$(13)
170 PRINT#4,"NEV" ; TAB(13) ; "LAKCIM" ; TAB(6) ; "TEL.SZAM"
180 PRINT#4,CHR$(15)
190 FOR I=1 TO X
200 N$=N$(I) ; C$=C$(I)
210 N$=N$+B$
220 C$=C$+B$
230 N$=LEFT$(N$,30)
240 C$=LEFT$(C$,30)
250 PRINT#4,N$ ; C$ ; T(I)
260 NEXT I
270 PRINT#4
280 CLOSE 4
290 END

```

### 201. Vegyük észre:

- Hogy a pénztárgép működése követhető legyen, érdemes az adatokat képernyőn is megjeleníteni.
- Mivel egy pénztár esetében nem mindig tudjuk előre megmondani, hogy hány árucikkal dolgozunk, ezért egy kapcsolót kell alkalmazni, ami eldönti, hogy további rögzítés történik-e vagy a végösszeget akarjuk-e kiírni.
- Erre szolgál a 100–120. sor.





Adatok kiírása mágnesszalagra:

- A magnót a kiírás megkezdése előtt meg kell nyitni.
- Íratás esetén a logikai egység szám (le): 1–255 lehet, a fizikai egység száma (fe) mindig 1, a mód (m) mindig 1.
- Ha a kiírás folyik, a képernyőn nem látunk semmit, így a magnó és a képernyő együttes használata nem lehetséges. Ha mégis szükség van mindkét egységre, azt a két egység szakaszos használatával oldhatjuk meg.
- A file névre azért van szükség, mert a visszaolvasáskor ennek alapján találja meg a gép a keresett adathalmazt.

PRINT le

- Az adatok kiírása a PRINT le utasítással történik.
- Ha egy PRINT le utasítással több adatot is akarunk íratni, azokat egymástól pontosvesszővel és a CHR\$(13) vezérlőkóddal célszerű elválasztani. Pl. 100 PRINT 1, X; CHR\$(13); A\$, CHR\$(13); B\$(1).

Adatok visszaolvasása magnókazettáról.

- A mágnesszalagot visszaolvasás előtt is meg kell nyitni.
- Olvasás esetén a logikai egység szám 1–255 lehet, a fizikai egység szám mindig 1, a mód mindig 0.
- Az adatok visszaolvasásakor a file névnek meg kell egyeznie a kiíratáskor létrehozott file névvel.
- Az adatok visszaolvasásakor sem használható a képernyő.

INPUT# le

- Az adatok visszaolvasása az INPUT# le utasítással történik.
- Ha egy INPUT# le utasítással egyszerre több adatot akarunk beolvasni, az adatokat vesszővel kell elválasztani egymástól. Pl.: 100 INPUT# 1,X,A\$,B\$(1)
- Az adatok beolvasásakor nem szükséges az egész állományt visszaolvasni. Ha a file-t lezártuk, az adatok visszaolvasása csak a file elejéről történhet, új OPEN utasítás után.

ST

- A lefoglalt változó mindenkor INPUT–OUTPUT művelet lefolyásáról ad tájékoztatást. Ezért visszaolvasáskor az ST értékét folyamatosan figyelni kell. Ha ST=64, ez azt jelenti, hogy beolvastuk az utolsó adatot a file-ból, így az olvasást be kell fejezni.

CLOSE le

- Az INPUT–OUTPUT műveletek végén a file-t le kell zárni.

A mágnesszalagon az adatokat csak sorosan tárolhatjuk. Mivel a magnó kis fordulatszámúval működik, célszerű a file helyét megjegyezni. Ha a magnóval akarjuk az adott file-t megkeresni, az esetenként igen hosszú időbe telik.

**202. Vegyük észre:**

- Az írás előtt a file-t meg kell nyitni, kiíratás után pedig le kell zárni.



A program:

```
10 PRINT "100 DB VELETLENSZAM KAZETTARA"  
20 FOR I=1 TO 500:NEXT I  
30 OPEN 1,1,1,"RANDOM"  
40 FOR I=1 TO 100  
50 N=INT(RND(1)*90)+1  
60 PRINT#1,N  
70 NEXT I  
80 CLOSE 1  
90 END
```

**203. Vegyük észre:**

- Visszaolvasás előtt a file-t meg kell nyitni, a végén pedig le kell zárni.
- Minden beolvasott érték után az ST értékét ellenőrizni kell. Ha ST=64, az olvasást be kell fejezni, a file-t le kell zárni.

A program:

```
10 PRINT "Q"  
20 INPUT "HANY SZAMOT OLVASUNK VISSZA" ;N  
30 N=INT(N):DIM R(100)  
40 OPEN 1,1,0,"RANDOM"  
50 FOR I=1 TO N  
60 INPUT#1,R(I)  
70 IF ST=64 THEN GOTO 90  
80 NEXT I  
90 CLOSE 1  
100 FOR I=1 TO N  
110 PRINT I;R(I)  
120 NEXT I  
130 END
```

**204. Vegyük észre:**

- Mivel egy órs általában 3 fősnél nagyobb, és 15 fősnél kisebb, érdemes ezekkel az értékekkel dolgozni.
- Hogy a 90. sor olvasható legyen, a gépet várakoztatni kell egy ciklus beiktatásával.

A program:

```
10 PRINT "Q"  
20 INPUT "HANY FOS AZ ORS" ;N  
30 IF N<3 OR N>15 THEN GOTO 10  
40 N=INT(N)  
50 DIM N$(N)  
60 FOR I=1 TO N  
70 INPUT "NEW" ;N$(I)  
80 NEXT I
```

```

90 PRINT "A KIMENTES ELINDUL "
100 FOR I=1 TO 500:NEXT I
110 OPEN 1,1,1,"NEVSOR"
120 FOR I=1 TO N
130 PRINT#1,N$(I)
140 NEXT I
150 CLOSE 1
160 PRINT " A KIMENTESNEK VEGE "
170 END

```

**205. Vegyük észre:**

- Mivel a magnó és a képernyő egyszerre nem használható, a memóriába való beolvasást és a képernyőre való kiíratást külön ciklus tartalmazza.

A program:

```

10 PRINT "Q"
20 DIM N$(15)
30 OPEN 1,1,0,"NEVSOR"
40 I=0
50 I=I+1
60 INPUT#1,N$(I)
70 IF ST<>64 THEN GOTO 50
80 CLOSE 1
90 FOR J=1 TO I
100 PRINT J$N$(J)
110 NEXT J
120 END

```

**206. Vegyük észre:**

- Egy PRINT# le-t használva az adatokat egymástól el kell választani (100. sor).

A program:

```

10 PRINT "Q"
20 INPUT "HANY FOS AZ OSZTALY" $N
25 IF N<10 OR N>42 THEN GOTO 10
30 N=INT(N):DIM N$(N):DIM A(N)
40 FOR I=1 TO N
50 INPUT "NEV" $N$(I)
60 INPUT "TANULMANYI ATLAGA" $A(I)
70 NEXT I
80 OPEN 1,1,1,"ATLAGOK"
90 FOR I= 1 TO N
100 PRINT#1,N$(I);CHR$(13);A(I)
110 NEXT I
120 CLOSE 1
130 END

```

**207. Vegyük észre:**

- A névsorba rendezést megoldhattuk volna a szomszédos elemek cseréjének módszerével is.

A program:

```
10 PRINT "Q"
20 PRINT "OSZTALYNEVSOR ES TANULMANYI ATLAGOK":
   DIM N$(42):A(42)
30 OPEN 1,1,0,"ATLAGOK"
40 FOR I=1 TO 42
50 INPUT#1,N$(I),A(I)
60 IF ST=64 THEN GOTO 80
70 NEXT I
80 CLOSE 1
90 FOR J=1 TO 10
100 PRINT N$(J),A(J)
110 NEXT J
120 PRINT "A BEOLVASAS VEGE"
130 PRINT #RENDEZES ELINDUL"
140 FOR J=1 TO I-1
150 FOR K=J+1 TO I
160 IF N$(J)<N$(K) THEN GOTO 200
170 V=N$(J):V=A(J)
180 N$(J)=N$(K):A(J)=A(K)
190 N$(K)=V:A(K)=V
200 NEXT K
210 NEXT J
220 FOR J=1 TO I
230 PRINT N$(J),A(J)
240 NEXT J
250 END
```

**208. Vegyük észre:**

- A file adatainak módosítása csak úgy oldható meg, hogy az adatokat beolvassuk a memóriába, itt elvégezzük a változtatásokat, majd a javított adatállományt újra kiíratjuk a kazettára.
- Egy tanuló távozása esetén az ő adatait ki kell törölni a file-ból (170... 230. sor).
- Ha új tanuló érkezik, meg kell vizsgálni, hogy az osztály létszáma nem érte-e el a 42-t, mert akkor osztálybontást kell alkalmazni. Ha ez nem áll fenn, a tömbbe, ill. a file-ba kell írni az adatokat (240... 290. sor).
- Mivel félévente a tanulmányi átlagok tömeges változtatásával kell számolni, ezért itt célszerű a változások rögzítésére két munkatömböt használni (V, v). Az adatok módosítása a 300... 400. sorban történik.
- Ha az összes szükséges javítást végrehajtottuk, az adatokat kazettára rögzítjük.
- Mivel a 206. feladat mint rögzítő program, a 207. feladat mint adatállomány munkát végző program és a 208. feladat mint az adatállomány





```

410 OPEN 1,1,1,"ATLAGOK"
420 FOR J=1 TO I
430 PRINT#1,N*(J)*CHR*(13)*A(J)
440 NEXT J
450 CLOSE 1
460 END

```

### 3.18 Mit csinál a gép? Elemezd a programokat!

A 209. és a 210. feladat a változók cseréjével foglalkozik.

- 209.**
- Ha a begépelte A érték nem kisebb a B begépelte értékénél, a gép felcseréli A és B értékét, majd kiírja azt.
  - Ha az állítás nem igaz, A és B értékét a gép változatlanul írja ki.

- 210.**
- Az A, B, C értékét begépeljük.
  - Ha  $A=B$  vagy B nagyobb, mint C, a begépelte értékeket változatlanul írja ki a gép.
  - Ellenkező esetben C felveszi B, A felveszi C értékét, B változatlanul marad. Az így előállt értékeket a gép kiírja.

**211.** Elemezd végig a programokat!

- |                        |                      |                       |                      |
|------------------------|----------------------|-----------------------|----------------------|
| a) A=6<br>B=5<br>C=11  | b) A=2<br>B=2<br>C=4 | c) A=8<br>B=0<br>C=10 | d) A=6<br>B=7<br>C=6 |
| e) A=-3<br>B=6<br>C=7  | f) D=4.5             | g) C=84               | h) A=21.2<br>B=16.2  |
| i) A=5<br>B=-1<br>C=20 |                      |                       |                      |

A 212...218. számú feladatok a ciklusokkal foglalkoznak.

- 212.**
- A tiszta képernyő egy sorának annyiadik oszlopába helyez a gép egy + jelet, ahányadik sort húzza. Negyven sor után újra az első oszloptól kezdi az írást. Ezért úgy érzékeljük, mintha a képernyőn 40 karakterből álló 45 fokos szögben húzott vonalak futnának felfelé.

- 213.**
- A gép a rajzolást a tiszta képernyő első sorában kezdi.
  - Minden DATA-ban megadott jélből egy sornyt húz a gép, majd a 100. sor miatt az egészet kezdi előlről.

- 214.**
- Az INPUT utasításnál kapott sztringet a gép a 40. utasítás miatt végtelen sokszor kiírja egymás mellé.

- 215.**
- A gép 100 darab csillagot ír ki egymás mellé a képernyőre. Ezáltal egy teljes képernyőnyi csillag jelent meg.  
A három egymásba skatulyázott ciklus helyettesíthető egy ciklussal is.



A program:

```
20 FOR I=1 TO 100
30 PRINT "*"
40 NEXT I
```

- 216.** – A gép 100 darab 2 és 200 közötti számot jelenít meg egymás mellett a képernyőn.
- 217.** – A gép 100 darab 1 -201 közötti számot jelenít meg egymás mellett a képernyőn.
- 218.** – Kiírja a gép, hogy az első húsz szám tetszőleges értékkel osztva milyen maradékot ad.
- A 219...223. feladatok a logikai kapcsolatokkal foglalkoznak.
- 219.** – A gép két csillagot ír ki, ha a begépelte X érték osztható 8-cal, egy csillagot ír ki, ha nem.
- 220.** – Ha a begépelte A érték pozitív és B értéke negatív, a gép nem csinál semmit, a program megáll. Ellenkező esetben a gép letörli a képernyőt, majd a közepére két csillagot ír.
- 221.** – Ha a begépelte karakter csillag vagy + jel, a gép adott karakterrel teleírja a képernyőt. A gép addig várakozik, amíg a két kijelölt karakter valamelyikét be nem gépeljük.
- 222.** – A gép egy véletlenszerűen előállított 1 és 10 közé eső számot vizsgál. Ha ez a szám nagyobb, mint 5 és nem osztható 3-mal, a gép kiírja: ALMA, ellenkező esetben: KÖRTE.
- 223.** a-2; b-3; c-6; d-5; e-1; f-4
- A 224...227. feladatok a sztringfüggvényekkel foglalkoznak.
- 224.** – A gép folyamatosan kiírja az első I darab szám összegét, egészen 9-ig.
- 225.** – A gép folyamatosan összefűzi az első I darab sztringet, egészen 9-ig. Ezeket ki is írja.
- 226.** – A gép X értékének megfelelő hosszt vág le az A\$ sztringváltozó bal oldaláról, és ezt az üres karaktert tartalmazó B\$ sztringváltozóval együtt kiírja a ciklusnak megfelelően addig, míg X értéke nem lesz nagyobb 10-nél.
- 227.** – A gép ugyanazt az eredményt produkálja, mint a feladat.
- A 228. és a 229. feladat az idő mérésével foglalkozik.
- 228.** – A gép kiírja, hogy egy ciklusmag nélküli FOR ciklus 100-szori futásához hány másodperc szükséges.



- 229.**
- A gép egy véletlenszámot generál 33–94 között. Ezt a számot írja a képernyőre. A gép azt az időt méri, amíg a játékos megtalálja a véletlenszámhoz, mint CHR kódhoz tartozó karaktert.
- A 230 . . . 237. feladatok a grafikával foglalkoznak.
- 230.**
- A képernyőn kijelölt papírt folyamatosan beszínezi az összes lehetséges színnel, és ezek összes árnyalatával.
- 231.**
- A gép a képernyő felső sorából egyszerre húz függőleges vonalakat minden tizedik oszlopban.
- 232.**
- A gép az első oszloptól kezdve, minden tizedik oszlopba függőleges vonalat húz a képernyőre.
- 233.**
- A gép a hátteret fehérre, a papírt feketére színezi, majd teleírja a képernyőt vörös + jelekkel.
- 234.**
- A gép vonalat húz a képernyő 0,0 pontjából a 200,200-as pontjába.
- 235.**
- A gép hét darab 40 egység sugarú kört rajzol, melyek középpontjai a 100. sorban találhatóak.
- 236.**
- A gép egy négyzetet rajzol a képernyőre. A felrajzolt ellipszis közül az egyik a vízszintes, a másik a függőleges oldalai mentén érinti a négyzetet.
- 237.**
- A gép a képernyő közepére egy kört rajzol. Ha a kör területén kijelölünk a vízszintes és a függőleges átmérőknek a körvonallal alkotott metszéspontjait, ezek négy pontot adnának. A gép ezek mint középpontok köré az eredetivel azonos sugárral köröket rajzol.

### 3.19 Néhány vegyes feladat

#### 238. Vegyük észre

- A lineáris függvény általános képletéből a konkrét függvényt úgy kaphatjuk meg, hogy megadjuk az A és B értékét.
- Egy lineáris függvény növekvő, ha a nagyobb X értékekhez tartozó Y értékek is nagyobbak.
- Ha a nagyobb X értékekhez tartozó Y értékek kisebbek, akkor a függvény csökkenő.
- Ha bármilyen X értékekre az Y tényezők állandók, a függvény menete állandó.

A program:

```

10 PRINT " "
20 PRINT " LINEARIS FV MENETEK MEGHATÁROZÁSA"
30 PRINT " A FV ÁLTALÁNOS KEPLETE Y=A*K+B"
40 INPUT "MENNYI A ÉRTEKE" :A
50 INPUT "MENNYI B ÉRTEKE" :B

```





```

90 FOR I=1 TO 10
100 A=INT(RND(1)*20)+1
110 PRINT X;"*";A;"=";
120 INPUT B
130 IF B=A(A) THEN PRINT "JO VALASZ":K=K+1:GOTO 150
140 PRINT "NEM JO. A JO VALASZ";A(A)
150 NEXT I
160 PRINT "□"
170 PRINT "██████████████████10 FELADATBOL";K;"JO MEGOLDAS"
180 PRINT "██MEGOLDASI IDO";TI/60;"MASODPERC"
190 END

```

#### 241. Vegyük észre:

- Egy nyolcadikos tanulót alapul véve egy tanuló 13 tantárgyból kaphat érdemjegyet, ezek begépelése akkor egyszerű, ha a gép kiírja, milyen tantárgy érdemjegyét várja. Célszerű a tantárgyak nevét DATA-ban tárolni.
- A DATA-ból az adatokat tömbbe olvasva, a tömbindex egyértelműen meghatározza, hogy melyik tárgyról van szó.
- Ezért érdemes definiálni ugyanilyen indexeléssel egy másik tömböt, amelyben a tárgyhoz tartozó jegyek összegezését végezzük.
- Ha egy tanuló érdemjegye elégtelen, ez most a tantárgyi átlag szempontjából nem jelent semmi különös elbánást.

A program:

```

10 PRINT "□"
20 PRINT "██████████AZ OSZTALY ATLAGA TARGYANKENT███"
30 INPUT "HANY FOS AZ OSZTALY";X
40 PRINT "□"
50 DIM A(13):DIM A$(13)
60 FOR I=1 TO 13
70 READ A$(I)
80 NEXT I
90 FOR J=1 TO X
100 PRINT J;" . TANULO JEGYE I███"
110 FOR I=1 TO 13
120 PRINT A$(I);
130 INPUT A
140 IF A>5 OR A<1 OR A<>INT(A) THEN PRINT "ROSSZ":GOTO 130
150 A(I)=A(I)+A
160 NEXT I
170 PRINT "□"
180 NEXT J
190 PRINT "□"
200 PRINT "AZ OSZTALY TANULMANYI ATLAGA TARGYANKENT███"
210 FOR I=1 TO 13
220 PRINT A$(I);TAB(20);A(I)/X

```



```

230 NEXT I
240 END
250 DATA MAGYAR NYELV,MAGYAR IRODALOM,OROSZ,TORTENELEM
260 DATA FIZIKA,KEMIA,BIOLOGIA,TECHNIKA,RAJZ,ENEK,
TESTNEVELES

```

#### 242. Vegyük észre:

- Négy különböző művelet végzését kell ellátnia a programnak, ezért érdemes menüvel és a választás után ON X GOTO-val dolgozni.
- Életkortól függően a tanulók csak adott számkörben tudnak műveleteket végezni. Ezért fontos megadni azt a lehetőséget, hogy a használó maga határozza meg, hogy mekkora számokkal kíván dolgozni.
- Y darab számjegy esetén a számokat a  $10^{(Y-1)}$ -ig érdemes generálni.
- Problémánk az osztásnál lehet, mert ha nem egész szám a hányados, a megoldás nem mindig egyezik meg a gép által előállított értékkel. Ezért itt célszerű inkább a hányados egészrészét és a maradékot kérdezni.
- A megoldást értékelni kell, rossz megoldás esetén megadni a megfelelő választ, hogy a tanuló ellenőrizhesse a saját megoldását.
- A megoldási időt is mérje a gép!

A program:

```

10 PRINT " "
20 PRINT "          A NEGY ALAPMUEVELET GYAKORLASA"
30 PRINT "          1. OSSZEADAS"
40 PRINT "          2. KIVONAS"
50 PRINT "          3. SZORZAS"
60 PRINT "          4. OSZTAS"
70 INPUT "          MELYIK MUEVELETET GYAKOROLJUK ";X
80 INPUT "          HANY JEGYU SZAMOKKAL DOLGOZUNK ";Y
90 K=0:TI$="000000"
100 FOR I=1 TO 10
110 A=INT(RND(1)*10^(Y))
120 B=INT(RND(1)*10^(Y))
130 ON X GOTO 220,250,280,310
140 INPUT D
150 IF D=C THEN K=K+1:PRINT "JO":GOTO 170
160 PRINT "NEM JO. A JO MEGOLDAS";C
170 NEXT I
180 PRINT " "
190 PRINT "          10 FELADATBOL ";K;" JO MEGOLDAS"
200 PRINT "MEGOLDASI IDO";TI/60;" MASODPERC "
210 END
220 PRINT A;"+";B;"=";
230 C=A+B
240 GOTO 140
250 PRINT A;"-";B;"=";
260 C=A-B

```

```

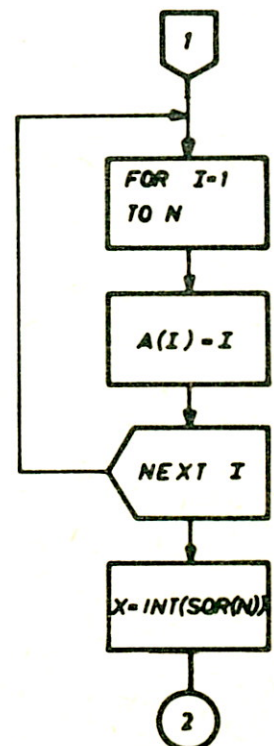
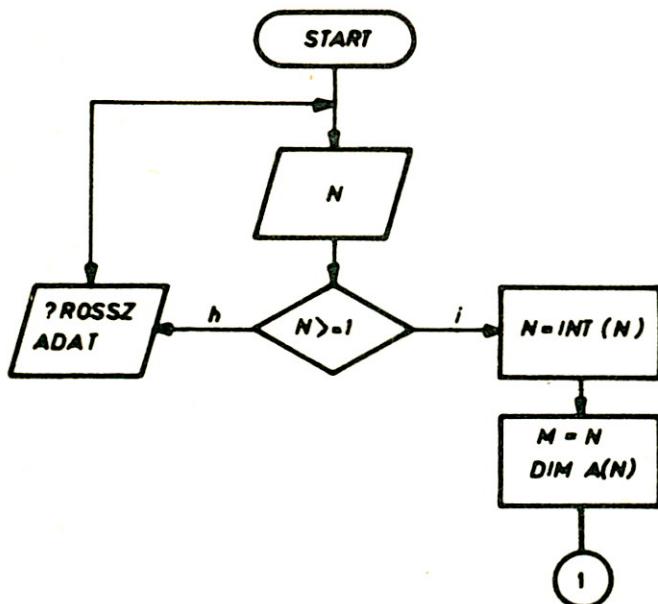
270 GOTO 140
280 PRINT A;"*";B;"=";
290 C=A*B
300 GOTO 140
310 PRINT A;" / ";B;"=";
320 C=INT(A/B)
330 M=A-C*B
340 INPUT D
350 INPUT "A MARADEK";M
360 IF C<>D THEN GOTO 380
370 IF M=N THEN GOTO 390
380 PRINT "NEM JO. A HANYADOS";C;"A MARADEK";M;GOTO 170
390 PRINT "JO";K=K+1
400 GOTO 170

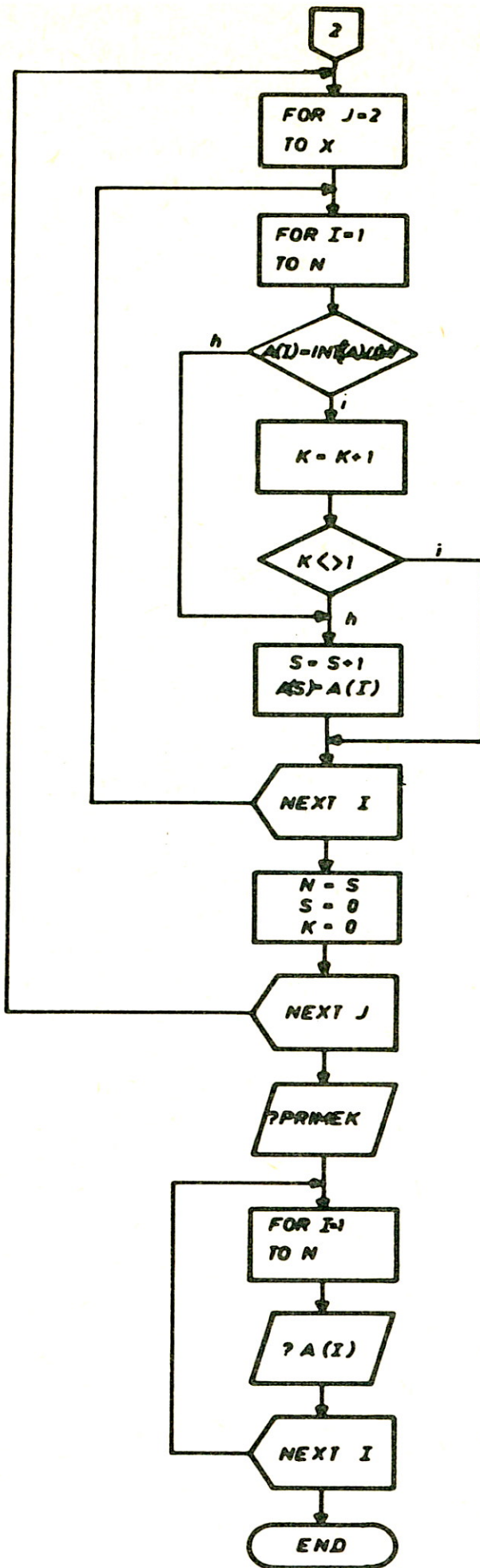
```

**243. Vegyük észre:**

- Az erathosztenészi szita a prímek kiválogatására szolgál tetszőleges N számig.
- Prímszám az a természetes szám, amelynek nincs valódi osztója.
- Belátható, hogy ha a prímeket az első N természetes szám között keressük, vizsgálódásunkat elég N-ig folytatni.
- Az oszthatóságot sorban az 1, 2, 3, ... N számokra mint osztókra vizsgáljuk.
- A következő számmal való oszthatóságot már csak azokra a számokra érdemes vizsgálni, amelyek az előző vizsgálódás során prímeknek vagy legalább az addigi osztókkal relatív prímeknek bizonyultak.
- Így az N-nel való vizsgálódás után már csak a prímek maradnak meg. Ezeket kell kiíratni.

A folyamatábra:













```

50 PRINT "A GEP EGY 10*10-ES MEZON ELDUG EGY HAJOT"
70 PRINT "BARMELY GOMB LENYOMASAVAL INDUL A JATEK"
80 GET A$
90 IF A$="" THEN GOTO 80
100 GRAPHIC2,1
110 COLOR 1,1,7
120 GOSUB 430
130 X=INT(RND(1)*10)+1
140 Y=INT(RND(1)*10)+1
160 I=0
170 I=I+1
180 INPUT "X KOORDINATA":X(I)
190 IF X(I)>10 OR X(I)<1 THEN GOTO 220
200 INPUT "Y KOORDIANATA":Y(I)
210 IF Y(I)<11 AND Y(I)>0 THEN GOTO 230
220 PRINT "ROSSZ ADAT":GOTO 180
230 IF X>X(I) THEN GOTO 250
240 IF Y=Y(I) THEN GOTO 280
250 PRINT "NEM TALALT"
260 COLOR 1,1,7
270 PAINT 1,X(I)*8+93,Y(I)*8+13
280 GOTO 170
290 PRINT "TALALT, ELSULLYEDT"
300 COLOR 1,1,7
310 PAINT 1,X(I)*8+93,Y(I)*8+13
320 PRINT I;"LOVESED VOLT."
330 FOR K=1 TO 2000:NEXT K
340 SCNCLR
350 GOSUB 430
360 COLOR 1,1,7
370 FOR J=1 TO I
380 FOR L=1 TO 200:NEXT L
390 PAINT 1,X(J)*8+93,Y(J)*8+13
400 PRINT J;" LOVES:";X(J);Y(J)
410 NEXT J
420 END
430 COLOR 1,1,7
440 CHAR 1,14,1,"TORPEDO"
450 BOX 1,100,20,180,100
460 A=8:B=1
470 FOR K=1 TO 2
480 FOR L=100 TO 180 STEP A
490 FOR J=20 TO 100 STEP B
500 DRAW 1,L,J

```



```

510 NEXT J
520 NEXT L
530 A=1:B=8
540 NEXT K
550 RETURN

```

**246. Vegyük észre:**

- Kockadobáló programot már készítettünk.
- Új probléma a kapott adatok ábrázolása. Szükséges a grafikon meg-  
rajzolása, az egységek bejelölése, az egyes tengelyek funkciójának je-  
lölése.

160–170. sor: A tengelyek

190–240. sor: A függőleges tengely egységeinek jelölése

250–300. sor: A vízszintes tengely egységeinek jelölése

320–390. sor: Az oszlopdiagram megrajzolása.

A program:

```

10 DIMA$(26)
20 PRINT "□"
30 PRINT "████████████████████KOCKADOBALO"
40 FOR I=1 TO 26
50 READA$(I)
60 NEXT I
70 FOR I=1 TO 100
80 A=INT(RND(1)*6)+1
90 PRINTA;
100 B(A)=B(A)+1
110 NEXT I
120 FORJ=1 TO 2000
130 PRINT "□"
140 GRAPHIC1,1
150 CHAR1,10,1,"KOCKADOBALO"
160 DRAW1,36,16 TO 36,184
170 DRAW1,24,176TO268,176
180 K=0
190 CHAR1,1,1,"DB"
200 FORJ=16TO168 STEP8
210 DRAW1,34,JTO38,J
220 K=K+1
230 CHAR1,1,J/8,A$(K)
240 NEXTJ
250 CHAR1,35,23,"PONT"
260 FOR I=72TO252 STEP36
270 DRAW1,I,174TOI,178
280 K=K+1
290 CHAR1,I/8,23,A$(K)
300 NEXT I

```





```

260 PRINT "*" ;
270 NEXT J
280 GET X$
290 IF X$="" THEN GOTO 280
300 IF X$(">")="£" THEN GOTO 80
310 END
320 DATA Q,1,W,2,E,3,R,4,T,5,Y,6,U,7,I,8,O,9,P,10,A,11,S,12
330 DATA F,14,G,15,H,16,76,J,17,K,18,L,19,Z,20,X,21,C,22,V,23
340 DATA N,25,M,26,1,27,2,28,3,29,4,30,5,31,6,32,7,33,8,34,9
350 DATA I,37,(,38,),39,?,40,;,41,.,42

```

#### 246. Vegyük észre:

- Célszerű az ételek neve helyett azok kódjával dolgozni, mert ez gyorsabb munkát tesz lehetővé.
- A DATA-ban tárolva az ételek nevét és árát, nincs szükség a kódok tárolására, ha azokat az ételek tömbindexével azonosnak tekintjük.
- A program nagy vonalakban a számlázó programunknak megfelelően működik.
- Ügyelni kell arra, hogy miután egy asztal számláját elkészítettük, az asztalhoz tartozó összes változó értéket törölni kell.

A program:

```

10 DIM S(10):DIM A(14):DIM I(14):DIM K(10,14)
20 DIM A$(14)
30 PRINT"☐"
40 FOR I=1 TO 14
50 READ A$(I)
60 READ A(I)
70 NEXT I
80 PRINT"☐"
90 PRINT"████████DUKRASZDAI PENZTARGEPE██████"
100 PRINT"██████MIT CSINALJUNK?██████"
110 PRINT"██████1. A FOGYASZTAS ROGZITESE"
120 PRINT"██████2. A SZAMLA"
130 INPUT M
140 ON M GOTO 160,250
150 GOTO 130
160 PRINT"☐"
170 INPUT"████████████████ASZTALSZAM" ; X
180 INPUT"██████AZ ETEL KODJA" ; K
190 I(X)=I(X)+1: I=I(X): K(X,I)=K
200 S(X)=S(X)+A(K)
210 GET A$
220 IF A$="+" THEN GOTO 180
230 IF A$="V" THEN GOTO 80
240 GOTO 210
250 INPUT"████████████████ASZTALSZAM" ; X

```



```

260 OPEN 4,4
270 B$="
280 PRINT#4,CHR$(14)
290 PRINT#4,TAB(10);X;". ASZTAL SZAMLAJA"
300 PRINT#4,CHR$(15)
310 FOR J=1 TO I(X)
320 K=K(X,J):A$=A$(K)+B$:A$=LEFT$(A$,30)
330 PRINT#4,TAB(20);A$;
340 PRINT#4,USING"#####.##";A(K)
350 NEXT J
360 PRINT#4,CHR$(13)
370 PRINT#4,TAB(20);"OSSZESEN"
380 PRINT#4,USING"#####.##";S(X)
390 PRINT#4
400 CLOSE 4
410 FOR J=1 TO I(X)
420 K(X,J)=0
430 S(X)=0
440 NEXT J
450 I(X)=0
460 GOTO 80
470 DATA COLA,5,MALNA,5.3,MANGO,5.5,NARANCS,5.5,JUICE,15
480 DATA PUNCSZTORTA,4.5,DOBOSZTORTA,5.2,CITROMSZTORTA,4.5
490 DATA TONIK,5.5,FOGACSA,2.8,MARKA,4.5,FAGYI 1 GOMBOS,3
500 DATA KAVE,5.5,VANILIASZTORTA,4.

```

**249. Vegyük észre:**

- A billentyűvel való rajzolás esetén lehetőséget kell adni arra, hogy a vonalakat balra, jobbra, fel, ill. lefelé húzzuk. Törlési lehetőséget is biztosítani kell.
- Figyelni kell arra, hogy a vonalat ne húzhassa a használó a papíron túl, mert az problémát okoz.

A program:

```

10 PRINT"☐"
20 PRINT"██████████RAJZOLAS BILLENTYUVEL██████████"
30 PRINT"A GEPEL LEHET RAJZOLTATNI <R>, TOROLNI <T>"
40 PRINT"A VONALAT BALRA <B>, JOBBRA <J>, LE <L> ES"
50 PRINT"FELFELE <F> HUZNI☐"
60 PRINT"BARMELY GOMB LENYOMASARA INDUL A PROGRAM"
70 PRINT"MEGALLNI A <*> BILLENTYUVEL LEHET"
80 GET A$
90 IF A$="" THEN GOTO 80
100 GRAPHIC 1,1
110 X1=0:X2=0:Y1=0:Y2=0
120 SCNCLR
130 DRAW 1,0,0
140 GET A$

```





```

210 PRINT "A";K;"BETUTU TARTALMAZO SZOVEGBEN ELO-"
220 PRINT "FORDULO BETUK SZAZALEKOS ARANYA"
230 FOR I=1 TO 26
240 PRINT B$(I);B(I);"DB";B(I)/K*100;"%",
250 NEXT I
260 END
270 DATA Q,W,E,R,T,Y,U,I,O,P,A,S,D,F,G,H,J,K,L,Z,X,C,
      V,B,N,M

```

## 251. Vegyük észre:

- A megadott ötbetűs szavak közül a gép véletlenszerűen választ. A játék annál érdekesebb, minél több szóval játszhatunk.
- Az eltalált betűt a kipontozott rész megfelelő helyére kell írni.
- Számolni kell a próbálkozásokat!
- A lehetséges betűk helyén szereplő öt pontot, ill. a már eltalált betűket is a képernyőnek azonos helyén érdemes kiírni, mert ez kelti azt a látzatot, mintha a játékos által eltalált betű a helyére is kerülne.
- A feladat megoldható úgy, hogy a szavakat DATA-ban tároljuk (1.), és megoldható úgy is, hogy a szavakat file-ban tároljuk (2/a, 2/b).

A programok:

### 1.

```

10 DIM A$(20)
20 H$="#####"
30 S$="#####|#####|#####|#####|#####"
40 FOR I=1 TO 20
50 READ A$(I)
60 NEXT I
70 A=INT(RND(1)*20)+1
80 PRINT " "
90 PRINT "#####SZOKITALALAS|"
100 PRINT "A GEP EGY OTBETUS SZOT REJTETT EL"
110 PRINT "EGY-EGY BETU BEGEPELESEVEL TIPPELHETSZ."
120 PRINT "A PROGRAM BARMELY BILLENTYUVEL INDUL"
130 GET A$
140 IF A$="" THEN GOTO 130
150 PRINT " "
160 PRINT S$;"....."
170 INPUT "A TIPPED";X$
180 FOR I=1 TO 5
190 B$=MID$(A$(A),I,1)
200 C$=LEFT$(H$,I-1)
210 IF X$=B$ THEN K=K+1 : PRINT " " ; S$ ; C$ ; X$
220 NEXT I
230 PRINT " " ; PRINT S$
240 S=S+1

```



```

250 IF K<5 THEN GOTO 170
260 PRINT"██████ ELTALALTAD AZ ELREJTETT SZOT!██"
270 PRINT"██PROBALKOZASAI SZAMA" / S
280 END
290 DATA KALAP,TAKAR,KERES,KAVAR,CSIBE,CSALOK,SAVAS,BARNA
300 DATA KINCS,NINCS,PALLO,PILLE,BIRKA,BARKA,MEGGY,MAGAS
310 DATA POROS,SULYT,FURAT,FAKAD

```

## 2/a

```

10 DIM A$(100)
20 H$="██████████████████████████████████████████████████████████████"
30 S$="██████████████████████████████████████████████████████████"
40 OPEN 1,1,0,"SZOKI"
50 I=1
60 INPUT#1,A$(I)
70 IF ST=64 THEN GOTO 90
80 I=I+1 : GOTO 60
90 CLOSE 1
100 A=INT(RND(1)*20)+1
110 PRINT"██"
120 PRINT"██████████████████████████████████████████████████████████ SZOKITALALAO ████████"
130 PRINT"A GEP EGY OTBETUS SZOT REJTETT EL"
140 PRINT"EGY-EGY BETU BEGEPELESEVEL TIPPELHETSZ."
150 PRINT"██A PROGRAM BARMELY BILLENTYUVEL INDUL██"
160 GET A$
170 IF A$="" THEN GOTO 160
180 PRINT"██"
190 PRINT S$ / "....."
200 INPUT"A TIPPED" / X$
210 FOR I=1 TO 5
220 B$=MID$(A$(A),I,1)
230 C$=LEFT$(H$,I-1)
240 IF X$=B$ THEN K=K+1 : PRINT"██" / S$ / C$ / X$
250 NEXT I
260 PRINT"██" : PRINT S$
270 S=S+1
280 IF K<5 THEN GOTO 170
290 PRINT"██████ ELTALALTAD AZ ELREJTETT SZOT!██"
300 PRINT"██PROBALKOZASAI SZAMA" / S
310 END

```

## 2/b

```

10 PRINT"██"
20 PRINT"██████████████████████████████████████████████████████████ SZOKITALALO SZAVAINAK ROGZITESE██"
25 PRINT"██████████████████████████████████████████████████████████ MAXIMUM 100 SZO ████████"

```









```

190 B(1)=A(4):B(2)=A(9):B(3)=A(2)
200 B(4)=A(3):B(5)=A(5):B(6)=A(7)
210 B(7)=A(8):B(8)=A(1):B(9)=A(6)
220 GRAPHIC 2,1
230 BOX 1,104,40,224,112
240 FOR I=1 TO 4
250 READ X1,Y1,X2,Y2
260 DRAW 1,X1,Y1 TO X2,Y2
270 NEXT I
280 FOR I=1 TO 9 STEP 4
290 CHAR 1,X(I),Y(I),STR$(B(I))
300 NEXT I
310 K=0
320 IF L=6 THEN GOTO 400
330 K=K+1
340 INPUT "SOR SZAM" / S
350 INPUT "ERTEK" / X
360 IF B(S)=X THEN CHAR 1,X(S),Y(S),
    STR$(B(S)) : L=L+1 : GOTO 320
370 PRINT "NEM JO"
380 GOTO 330
390 PRINT "□"
400 PRINT K / "PROBALKOZASOD VOLT"
410 END
420 DATA 14,6,19,6,24,6,14,9,19,9,24,9,14,12,19,12,24,12
430 DATA 144,40,144,112,184,40,184,112
440 DATA 104,64,224,64,104,88,224,88

```

#### 254. Vegyük észre:

- Az alpprogram csekély változtatással bármely tantárgy gyakorlására alkalmas lehet.
- Csak olyan feladatokat érdemes ilyen típusú gyakorlóprogramokban felhasználni, amelyekre egy betűvel, egy adott szóval, esetleg egy pontosan előre meghatározott mondattal (pl. szabály) lehet válaszolni.
- A programot átírható úgy is, hogy az érdemjegyeket adjon.

A program:

```

10 DIM K$(100) : DIM V$(100)
20 PRINT "□"
30 PRINT "■■■■■■■■■■ FIZIKAI FELVIVO ■■■■■■"
40 INPUT "HANY FELADAT LESZ" / N
50 FOR I=1 TO N
60 INPUT "KERDES" / K$(I)
70 INPUT "VALASZ" / V$(I)
80 NEXT I
90 OPEN 1,1,1,"FIZIKAI"

```

```

100 FOR I=1 TO N
110 PRINT#1,K$(I),CHR$(13),V$(I)
120 NEXT I
130 CLOSE 1
140 END

```

A program (rögzítő):

```

10 I=0
20 OPEN 1,1,0,"FIZIKA"
30 I=I+1
40 INPUT#1,K$(I)
50 INPUT#1,Y$(I)
60 IF ST<>64 THEN GOTO 30
70 CLOSE 1
80 PRINT"☐"
90 PRINT"████████████████████FIZIKA GYAKORLO███"
100 PRINT"A GEP 20 FELADATOT AD. A KEPERNYON MEG-"
110 PRINT"JELENO FOGALOM MELLE A JELET KELL BEGE-"
120 PRINT"PELNEO. A GEP ELLENORZI ES ERTEKELI A"
130 PRINT"VALASZAJDAT.███"
140 PRINT"HA INDULHAT, NYOMJ MEG EGY GOMBOT !"
150 GET A$
160 IF A$="" THEN GOTO 150
170 PRINT"☐"
180 FOR J=1 TO 10
190 A=INT(RND(1)*I)+1
200 PRINT K$(A)
210 INPUT V$
220 IF V$=V$(A) THEN S=S+1:PRINT"JO VALASZ":GOTO 240
230 PRINT"ROSSZ VALASZ. A JO MEGOLDAS "V$(A)
240 NEXT J
250 PRINT"☐"
260 PRINT"████████████████████ERTEKELES███"
270 PRINT"10 KERDESBOL";S;"JO VALASZOD VOLT"
280 IF S<5 THEN PRINT"EZ GYENGE EREDMENY":GOTO 320
290 IF S<7 THEN PRINT"EZ KOZEPEO EREDMENY":GOTO 320
300 IF S<9 THEN PRINT"EZ JO EREDMENY":GOTO 320
310 PRINT"EZ KIVALO EREDMENY"
320 PRINT"████████████████████████████████████████DOLGOZOL MEG?███"
330 INPUT B$
340 IF B$="I" THEN GOTO 90
350 END

```

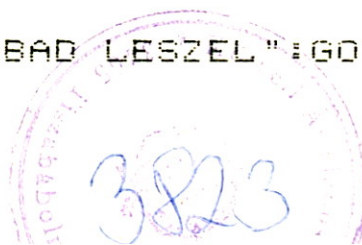


255. Vegyük észre:

- A zárok eredetileg zárva voltak, különben a rabok már régen megszöktek volna.
- Minden páratlan számú fordítás tehát nyitást, minden páros számú pedig zárást jelent.
- Mivel 1. fordítást minden ajtón, 2. fordítást minden másodikk, 3. fordítást minden harmadik, ... n. fordítást minden n. ajtón hajtanak végre, ebből látszik, hogy egy záron annyiszor fordítják el a kulcsot, ahány osztója van.
- Nyitva lesz tehát az a cella, amely sorszámának páratlan, zárva, amely sorszámának páros számú osztója van.
- Ha az első látásra kézenfekvő két egymásba skatulyázott ciklust alkalmaznánk, amelyek az első 100 szám oszthatóságát vizsgálnák az első 100 természetes számra nézve, a program nagyon sokáig futna.  
Elemezd a 160–260. sort. Ez egy gyorsabb megoldása a feladatnak.

A program:

```
10 DIM Z(100): DIM S(100)
20 PRINT " "
30 PRINT "TÖRÖK SZULTAN ES A KETALLASU ZAR"
40 PRINT "A SZULTAN BORTONEBEN 100 FOGOLY VAN."
50 PRINT "MINDEN AJTON KETALLASU ZAR VAN."
60 PRINT "EGY FEGYORNEK AZT AZ UTASITAST ADJA, HOGY"
70 PRINT "MINDEN EGYES AJTO ZARJAN FORDITSON EGYET"
80 PRINT "A MASIKNAK, HOGY MINDEN MASODIK ZARON,"
90 PRINT "A SZAZADIKNAK, HOGY MINDEN SZAZADIK ZARON"
100 PRINT "FORDITSON EGYET. AKINEK A CELLAJA NYITVA"
110 PRINT "MARAD, AZT SZABADON BOCSATJA"
120 PRINT "HA RAB LENNEL, MELYIK CELLAT VALASZTANAD?";
130 INPUT X
140 IF X < 1 OR X > 100 THEN PRINT "FIGYELJ JOBBAN!": GOTO 130
150 PRINT "AZ OROK VEGZIK A FELADATUKAT"
160 I = 1
170 J = 1
180 N = I * J
190 IF N > 100 OR J > 100 THEN GOTO 210
200 Z(N) = Z(N) + 1 : J = J + 1 : GOTO 180
210 IF I <= 100 THEN I = I + 1 : GOTO 170
220 J = 0
230 FOR I = 1 TO 100
240 IF Z(I) = INT(Z(I)/2)*2 THEN GOTO 260
250 J = J + 1 : S(J) = I
260 NEXT I
270 FOR I = 1 TO J
280 IF X = S(I) THEN PRINT "SZABAD LESZEL": GOTO 310
290 NEXT I
300 PRINT "RAB MARADSZ"
310 END
```







# TARTALOMJEGYZÉK

## I. FELADATOK

1. A billentyűzet használata	5
1.1 Példák	5
1.2 Játékok	6
2. Számok és műveletek	6
2.1 Példák	6
2.2 A változók	7
3. A programozás	8
3.1 A PRINT utasítás	8
3.2 A változók cseréje	9
3.3 Az INPUT utasítás	10
3.4 Néhány gyakori függvény	11
3.5 Az IF...THEN és a GOTO kapcsolata	12
3.6 A ciklus (IF...THEN alkalmazásával)	13
3.7 A tömb	15
3.8 FOR...TO...STEP és NEXT ciklusutasítások	16
3.9 READ, DATA, RESTORE	17
3.10 GET A\$	18
3.11 Időmérés számítógéppel (TI és TI\$)	18
3.12 Képernyőkezelés, grafika	18
3.13 Zene	19
3.14 Karakterfüzér függvények	20
3.15 Logikai műveletek (OR, AND, NOT) és ON...GOTO	20
3.16 A sornyomtató használata	21
3.17 A kazettás egység	22
3.18 Mit csinál a gép? Elemezd a programokat!	23
3.19 Néhány vegyes feladat	31

## II. MEGOLDÁSOK

1. A billentyűzet használata	34
1.1 Példák	34
2. Számok és műveletek	34
2.1 Példák	34
2.2 A változók	35
3. A programozás	36
3.1 A PRINT utasítás	37
3.2 A változók cseréje	41
3.3 Az INPUT utasítás	43
3.4 Néhány gyakori függvény	50
3.5 Az IF THEN és a GOTO kapcsolata	54
3.6 A ciklus (IF...THEN alkalmazásával)	75
3.7 A tömb	93
3.8 FOR...TO...STEP és NEXT ciklusutasítások	111
3.9 READ, DATA, RESTORE	135
3.10 GET A\$	143

3.11	Időmérés számítógéppel (TI és TI\$)	-	-	-	-	-	-	-	-	-	-	144
3.12	Képernyőkezelés, grafika	-	-	-	-	-	-	-	-	-	-	147
3.13	Zene	-	-	-	-	-	-	-	-	-	-	154
3.14	Karakterfüzér függvények	-	-	-	-	-	-	-	-	-	-	158
3.15	Logikai műveletek (OR, AND, NOT) és ON...GOTO	-	-	-	-	-	-	-	-	-	-	167
3.16	A sornyomtató használata	-	-	-	-	-	-	-	-	-	-	180
3.17	A kazettás egység	-	-	-	-	-	-	-	-	-	-	187
3.18	Mit csinál a gép? Elemezd a programokat!	-	-	-	-	-	-	-	-	-	-	193
3.19	Néhány vegyes feladat	-	-	-	-	-	-	-	-	-	-	195



Ára: 163,—Ft