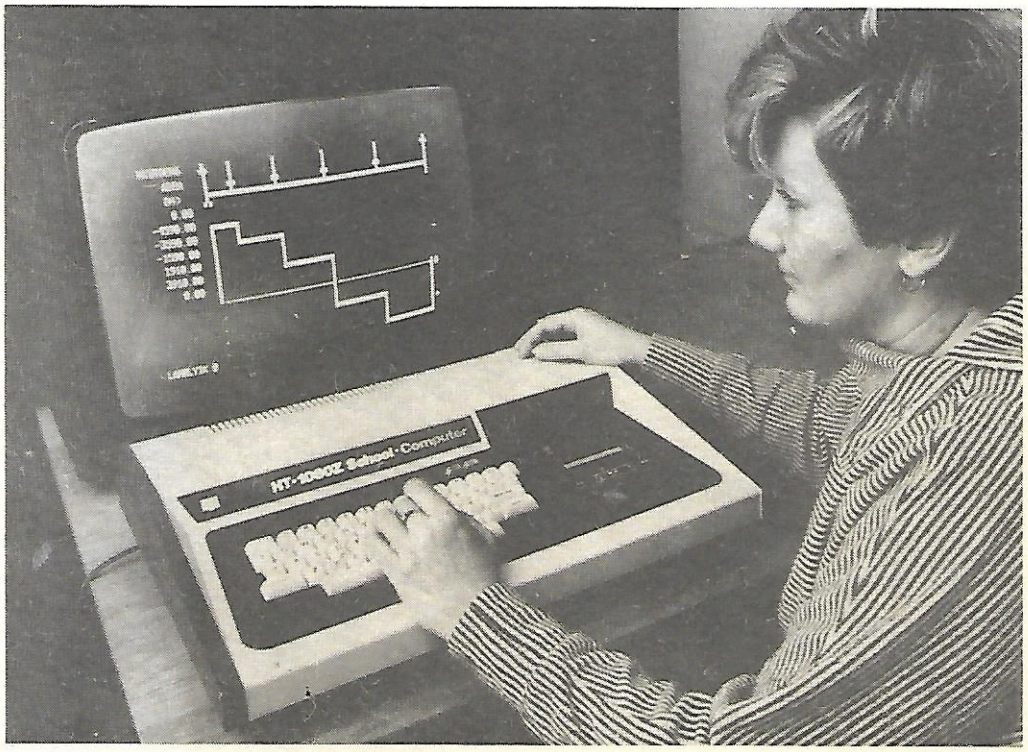
FŐVÁROSI PEDAGÓGIAI INTÉZET
TUDOMÁNYSZERVEZÉSI
ÉS INFORMATIKAI INTÉZET SZÁMÍTÁSTECHNIKAI SOROZATA
IV.

Appel György–Kőhegyi János–Zsakó László

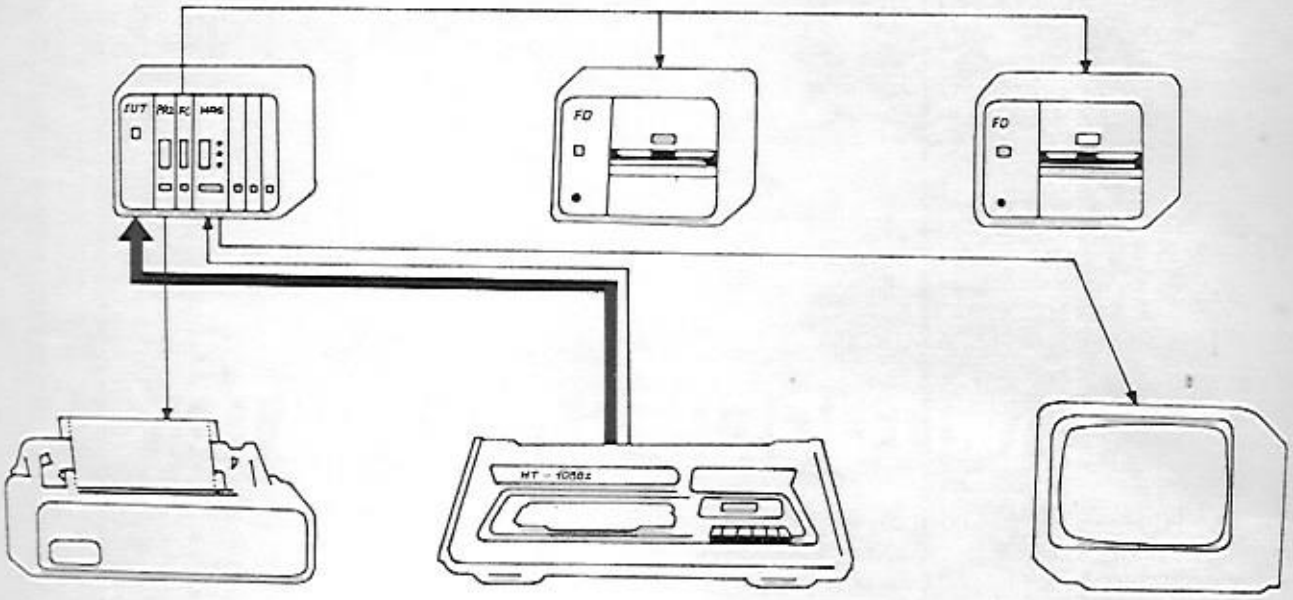
SZÁMÍTÓGÉPES FELADATOK

(példatár személyi számítógépekre)



INTERPRESS
BUDAPEST 1985

HIRADÁSTECHNIKA
SZÖVETKEZET
1519. Budapest, Pf. 268



HT – 1080Z/64 School Computer

Az alapgép már 64 kbyte memóriakapacitású ! BASIC és FORTH programozási nyelv!

Nyomtató illesztő

Minden-u. n. Centronics tipusú nyomtatóhoz alkalmazható (pl. Epson RX - 80, Seikosha GP80 GP100, TERTA TMT 12X, MOM K6311, stb.)

HT – 1080Z/PR1 kábelre szerelt változat
HT – 1080Z/PR2 bővitő egységbe tolható kártya

Bővitő egység HT –1080Z/EU7

Különféle illesztő és bővitő kártyák / max. 7db. / fogadására alkalmas doboz. Busz illesztő és tápegy - séget tartalmaz.

Floppy illesztő HT – 1080Z/F1

Illesztő egység hajlékony mágneslemezes tárolóhoz / 4 db. MOM 1800/900 tipusú drive meghajtásra alkal - mas. / Bővitő egységbe tolható kártya.

Nagybontású grafika HT – 1080Z/HRG

512 x 512 felbontású grafikus megjelenitést tesz lehetővé. Minden képpont külön címezhető. Kép - pontonként négyféle fénnyességi állitható be. Nagyitási lehetőség ! BASIC – ből programozható.

FŐVÁROSI PEDAGÓGIAI INTÉZET
TUDOMÁNYSZERVEZÉSI
ÉS INFORMATIKAI INTÉZET SZÁMÍTÁSTECHNIKAI SOROZATA
IV.

Appel György–Kőhegyi János–Zsakó László

SZÁMÍTÓGÉPES FELADATOK

(példatár személyi számítógépekre)

INTERPRESS
BUDAPEST, 1985.

Szerzők:

dr. Appel György
Kőhegyi János
Zsakó László

Lektorok:

Hubert Tibor
dr. Varga András

Szerkesztők:

dr. Appel György
Mihályfi János

L.02 40 46

© **Appel György, Kőhegyi János, Zsakó László, Budapest, 1985.**

ISBN 963 7222 00 6
ISSN –0236–9230

85.73 Szolnoki Nyomda Vállalat
Felelős vezető: GOMBKÖTŐ BÉLA

TARTALOMJEGYZÉK

Bevezető 5

1. Elemi feladatok a BASIC nyelv önálló tanulásához 7
2. Gyakorló feladatok kezdők számára 19
3. Gyakorló feladatok haladók számára 23
4. Versenyfeladatok 29

Feladatmegoldások 35

5. Feladatmegoldások kezdők számára 37
6. Feladatmegoldások haladók számára 71

Államigazgatási Számítógépes Szolgálat 129

Irodalomjegyzék 131

BEVEZETŐ

Hazánkat is utolérte a „házi-számítógépes" hullám. Sokan vásárolnak külföldön különböző típusú számítógépeket, és ma már hazai gyártású személyi számítógépek is megjelentek. A kötetnek az a célja, hogy az eddigiekben az e témakörben hiányt jelentő, számítástechnikai eszközökkel megoldható feladatokon átsegítse a számítástechnikával ismerkedőket.

Ez a kiadvány a HT–1080Z típusú iskolaszámítógépre készült, de kevés munkával átdolgozható bármilyen számítógépre.

A kiadvány első része az önállóan tanulóknak nyújt feladatmegoldási lehetőséget. Itt rögtön közöljük a megoldásokat is. A második részben olyan gyakorló feladatok vannak, amelyek részben a kezdők, részben a haladók számára nyújtanak feladatmegoldási segítséget. Itt a példatári módszernek megfelelően sorban közöljük a feladatokat, ezek megoldásait külön fejezetben adjuk meg. Ezzel az a célunk, hogy „kényszerítsük" először a T. Olvasót az önálló megoldásra. Ne keseredjen el azonban, ha a megoldás nem egyezik a mienkkel, hiszen amit közlünk, az csak egy a lehetséges megoldások közül. Végezetül néhány hazai számítástechnika-verseny feladatsorait mutatjuk be.

A feladatok egy kis részét különböző folyóiratokból, kiadványokból gyűjtöttük össze, ezek irodalomjegyzékét a példatár végén közöljük.

A könyv az *Országos Oktatástechnikai Központ* és az *Országos Pedagógiai Intézet* közös együttműködésének eredménye.

Ezúton is szeretnénk köszönetet mondani Hubert Tibornak, a Kvassay Jenő Szakközépiskola vezető tanárának és Veszprémi Annának, Kozics Sándornak, Sándor Antalnak, valamint Szlávi Péternek, az ELTE TTK Számítástechnikai Tanszék munkatársainak az anyag összegyűjtésében, ellenőrzésében nyújtott segítségükért.

Budapest, 1984 november

A szerzők

1. Elemi feladatok a BASIC nyelv önálló tanulásához

Értékadó utasítások

K.1. Írjunk programot, amely kiszámítja a henger felszínét és térfogatát!

```
 10   CLS : PRINT "A HENGER FELSZÍNE ÉS TÉRFOGATA" : PRINT
 20   PRINT "KÉREM A HENGER SUGARÁT (CM)";
 30   INPUT R : IF R<0 THEN 20
 40   PRINT "KÉREM A HENGER MAGASSÁGÁT (CM)";
 50   INPUT M : IF M<0 THEN 40
 60   V=R*R*3.14*M
 70   F=(2*R*R*3.14)+(2*R*3.14*M)
 80   PRINT "TÉRFOGAT=";V;"KÖBCM"
 90   PRINT "FELSZÍN=";F;"NÉGYZETCM"
100   STOP
```

K.2. Írjunk programot, amely az elektromos áram és feszültség ismeretében meghatározza a teljesítményt!

```
 10   CLS : PRINT "TELJESÍTMÉNY SZÁMÍTÁS" : PRINT
 20   PRINT "KÉREM A FESZÜLTSÉGET (V)";
 30   INPUT U
 40   PRINT "KÉREM AZ ÁRAMERŐSSÉGET (A)";
 50   INPUT I
 60   N=U*I
 70   PRINT "TELJESÍTMÉNY=";N;"WATT"
 80   PRINT
 90   STOP
```

K.3. Két pont koordinátáiból határozzuk meg a két pont távolságát!

```
 10   CLS : PRINT "KÉT PONT TÁVOLSÁGA" : PRINT
 20   PRINT "AZ ELSŐ PONT ELSŐ KOORDINÁTÁJA";
 30   INPUT X1
 40   PRINT "AZ ELSŐ PONT MÁSODIK KOORDINÁTÁJA";
 50   INPUT Y1
 60   PRINT "A MÁSODIK PONT ELSŐ KOORDINÁTÁJA";
 70   INPUT X2
 80   PRINT "A MÁSODIK PONT MÁSODIK KOORDINÁTÁJA";
 90   INPUT Y2
100   D=SQR ( (X2-X1)↑2+(Y2-Y1)↑2 )
110   PRINT "TÁVOLSÁG=";D
120   STOP
```

1. Elemi feladatok a BASIC nyelv önálló tanulásához

Elágazás

K.4. Határozzuk meg az x —> 1/(x+1) függvény néhány értékét!

```
 10   CLS : PRINT "Y=1/(X+1) FÜGGVÉNY ÉRTÉKEI" : PRINT
 20   PRINT "KÉREM X ÉRTÉKÉT";
 30   INPUT X
 40   IF X=-1 THEN 100
 50   Y=1/(X+1)
 60   PRINT "Y=";Y
 70   STOP
100   PRINT "AZ X=-1 ESETÉN A NEVEZŐ=0,"
110   PRINT "A FÜGGVÉNYÉRTÉK NEM SZÁMÍTHATÓ KI!"
120   STOP
```

K.5. Írjunk programot, amely a feszültség és az áramerősség ismeretében meghatározza az ellenállást!

```
 10   CLS : PRINT "ELLENÁLLÁS SZÁMÍTÁS" : PRINT
 20   PRINT "KÉREM AZ ÁRAMERŐSSÉGET (A)";
 30   INPUT I
 40   PRINT "KÉREM A FESZÜLTSÉGET (V)";
 50   INPUT U
 60   IF I=0 THEN 100
 70   R=U/I
 80   PRINT "ELLENÁLLÁS=";R;"OHM"
 90   STOP
100   PRINT "ÁRAMERŐSSÉG NEM LEHET=0"
110   STOP
```

K.6. Írjunk programot, amely egy évszámról eldönti, hogy szökőév-e!

A feladat megoldásához egy kevés ismeret kell a naptárról:

a szökőévek a néggyel osztható évek közül kerülnek ki, de nem minden százzal osztható év szökőév (pl. 1900 nem volt az); a százzal osztható évek közül csak a négyszázzal oszthatók szökőévek. (Másképp megfogalmazva: szökőév van, ha négyszázzal osztható, vagy ha néggyel osztható, de százzal nem.)

```
 10   CLS : PRINT "A PROGRAMMAL EGY ÉVSZÁMRÓL EL LEHET"
 20   PRINT "DÖNTENI, HOGY SZÖKŐÉV-E"
 30   PRINT
```

```
40    INPUT "ÉVSZÁM";E : IF E<>INT(E) THEN 40
50    IF E/400=INT (E/400) OR (E/4=INT (E/4) AND
      E/100<>INT(E/100)) THEN PRINT "SZÖKŐÉV" ELSE
      PRINT "NEM SZÖKŐÉV"
60    STOP
```

K.7. Írjunk programot, amely kiírja egy adott intervallumba eső páros számokat!

```
10    CLS : PRINT "PÁROS SZÁMOK" : PRINT
20    PRINT "KÉREM AZ INTERVALLUM ALSÓ HATÁRÁT";
30    INPUT A
40    PRINT "KÉREM AZ INTERVALLUM FELSŐ HATÁRÁT";
50    INPUT B
60    C=A/2
70    IF C<>INT (C) THEN K=A+1 ELSE K=A
80    IF K>B THEN 120
90          PRINT K;
100         K=K+2
110   GOTO 80
120   STOP
```

Szöveg típusú változók

K.8. Írjunk programot, amely egy adott áru értékét bizonyos százalékkal le-, ill. felértékeli!

```
10    CLS : PRINT "ÁRU LE- ILL. FELÉRTÉKELÉS" : PRINT
20    PRINT "AZ EREDETI ÉRTÉKE";
30    INPUT A : IF A<=0 THEN 20
40    PRINT "LEÉRTÉKELÉS VAN (I , N)";
50    INPUT V$
60    IF V$<>"I" AND V$<>"N" THEN 40
70    PRINT "MENNYI SZÁZALÉKKAL";
80    INPUT S
90    IF V$="I" THEN S=-S
100   F=A+A*S/100
110   PRINT "ÚJ ÁR=";F
120   STOP
```

K.9 Írjunk programot, amely a dollár értékét átváltja forintra, ill. a forint értékét dollárra!

```
 10   CLS : PRINT "PÉNZVÁLTÓ" ; PRINT
 20   PRINT "FORINTJA VAN (I,N)";
 30   INPUT V$
 40   IF V$<>"I" AND V$<>"N" THEN 20
 50   PRINT "HÁNY FORINT EGY DOLLÁR";
 60   INPUT V
 70   IF V<=0 THEN 50
 80   IF V$="I" THEN 200
 90   PRINT "MENNYI DOLLÁROD VAN";
100   INPUT D
110   IF D<0 THEN 90
120   F=V*D
130   PRINT "EZ"; F;"FORINT"
140   STOP
200   PRINT "MENNYI FORINTOD VAN";
210   INPUT F
220   IF F <0 THEN 200
230   D=F/V
240   PRINT "EZ";D;"DOLLÁR"
250   STOP
```

A ciklus fogalma

K.10. Írjunk programot, amely kiszámítja az első N páratlan szám összegét!

```
 10   CLS : PRINT "AZ ELSŐ N PÁRATLAN SZÁM ÖSSZEGE" : PRINT
 20   PRINT "HÁNY PÁRATLAN SZÁMOT ADJUNK ÖSSZE";
 30   INPUT P
 40   IF P<1 THEN 20
 50   S=0:D=0
 60        S=S+2*D+1
 70        D=D+1
 80   IF D<=P THEN 60
 90   PRINT "ÖSSZEG=";S
100   STOP
```

Ciklus utasítás

K.11. Írjunk programot, amely az x —> 1/(x+3) [–4,4] intervallum egész értékeihez tartozó függvény értékeit írja ki!

```
 10    CLS : PRINT "FÜGGVÉNYTÁBLÁZAT" : PRINT
 20    PRINT "AZ ÉRTÉKEK:"
 30    FOR X=–4 TO 4
 40          IF X=–3 THEN 80
 50          Y=1/(X+3)
 60          PRINT "F(";X;")=";Y
 70          GOTO 90
 80          PRINT "F(–3 ) NEM DEFINIÁLT"
 90    NEXT X
100    STOP
```

K.12 Írjunk programot, amely egy adott intervallumot adott darabszámú, egyenlő hosszúságú részekre oszt fel!

```
 10    CLS : PRINT "INTERVALLUM FELOSZTÁS" : PRINT
 20    PRINT "KÉREM AZ ALSÓ ÉS A FELSŐ HATÁRT";
 30    INPUT A,F
 40    IF A>F THEN 20
 50    PRINT "HÁNY SZAKASZRA OSZTJUK';
 60    INPUT N
 70    IF N<1 THEN 50
 80    PRINT "AZ OSZTÓPONTOK:";
 90    FOR I=A TO B STEP (B–A)/N
100          PRINT I;
110    NEXT I
120    STOP
```

K.13. Készítsünk programot a kamatos kamat számítására!

```
 10    CLS : PRINT "KAMATOS KAMAT" : PRINT
 20    PRINT "KÉREM AZ ALAPÖSSZEGET";
 30    INPUT A
 40    IF A<0 THEN 20
 50    PRINT "KÉREM AZ ÉVEK SZÁMÁT";
 60    INPUT E
 70    IF E<1 THEN 50
 80    PRINT "KÉREM A SZÁZALÉKOT";
 90    INPUT S
```

```
100     IF S<0 THEN 80
110     FOR I=1 TO E
120         A=A*(1+S/100)
130         PRINT I;" . ÉV UTÁN";A
140     NEXT  I
150     STOP
```

K.14. Írjunk programot átlagszámításra!

```
 10     CLS  : PRINT "ÁTLAGSZÁMÍTÁS" : PRINT
 20     PRINT  "MENNYI SZÁM LESZ";
 30     INPUT  N
 40     IF N<1 THEN 20
 50     S=0
 60     FOR I=1 TO N
 70         PRINT "KÖVETKEZŐ SZÁM";
 80         INPUT X
 90         S=S+X
100     NEXT  I
110     PRINT  S/N
120     STOP
```

Adattárolás programban

K.15. Ali, Bea, Ede, Ili, Leó, Oti, Ria közül öten barátok. Írjunk programot, amely kiírja az összes lehetséges párosításukat!

```
 10     CLS  : PRINT "HÉTBŐL ÖTEN BARÁTOK" : PRINT
 20     DATA  ALI,BEA,EDE,ILI,LEO,OTI,RIA
 30     FOR  I=1 TO 7 : READ A$ (I) : NEXT I
 40     FOR  I=1 TO 3
 50      FOR  J=I+1 TO 4
 60       FOR  K=J+1 TO 5
 70        FOR  L=K+1 TO 6
 80         FOR  M=L+1 TO 7
 90             PRINT A$(I);A$(J);A$(K);A$(L);A$(M)
100             S=S+1 : IF S<14 THEN 130
110             PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!"
120             IF INKEY$="" THEN 120 ELSE CLS : S=0
130     NEXT  M,L,K,J,I
140     STOP
```

Indexes változók

K.16. Készítsünk programot, amely egy hónap sorszáma alapján megadja a hónap nevét!

```
10    CLS : PRINT "HÓNAP NÉV MEGHATÁROZÁS" : PRINT
20    DATA JANUÁR,FEBRUÁR,MÁRCIUS,ÁPRILIS,MÁJUS,JÚNIUS
30    DATA JÚLIUS,AUGUSZTUS,SZEPTEMBER,OKTÓBER,NOVEMBER,DECEMBER

40    DIM H$(12)
50    FOR I=1 TO 12 : READ H$(I) : NEXT I
60    PRINT "HÓNAP SORSZÁM";
70    INPUT H : IF H<1 OR H>12 THEN 60
80    PRINT "A NEVE: "H$(H)
90    STOP
```

K.17 Készítsünk programot, amely egy hónap nevéről megadja, hogy a hónap melyik évszakban van!

```
10    CLS : PRINT "ÉVSZAK NÉV MEGHATÁROZÁS" : PRINT
20    DATA JANUÁR,FEBRUÁR,MÁRCIUS,ÁPRILIS,MÁJUS,JÚNIUS
30    DATA JÚLIUS,AUGUSZTUS,SZEPTEMBER,OKTÓBER,NOVEMBER,DECEMBER

40    DATA TÉL,TAVASZ,NYÁR,ŐSZ
50    DIM H$(12),E$(3)
60    FOR I=1 TO 12 : READ H$(I) : NEXT I
70    FOR I=0 TO 3 : READ E$(I) : NEXT I
80    PRINT "A HÓNAP NEVE";
90    INPUT H$
100   H$ (0)=H$(12) : REM DECEMBER LESZ A 0. HÓNAP IS
110   I=0
120   IF I>11 THEN PRINT "ILYEN HÓNAP NINCS : STOP
130   IF H$(I)=H$ THEN 160
140       I=I+1
150   GOTO 120
160   E=INT(I/3) : REM ÉVSZAK SORSZÁM
170   PRINT "A HÓNAPHOZ TARTOZÓ ÉVSZAK: ";E$(E)
180   STOP
```

1. Elemi feladatok a BASIC nyelv önálló tanulásához

Grafika

K.18. Írjunk programot, amely egy 10 egységnyi négyzetet és az átlóit rajzolja!

```
10    CLS
20    FOR I=10 TO 20
30          SET (I,10) : SET (I,20) : REM FELSŐ– ALSÓ SOR
40          SET (10,I) : SET (20,I) : REM BAL– JOBBOLDAL
50          SET (I,I) : SET (I,30–I) : REM ÁTLÓK
60    NEXT I
70    STOP
```

K.19. Rajzoljunk ki a képernyő bal oldali felére egy koordinátarendszert, amelynek origója a képernyő középpontjába (32,24) esik, s készítsünk hozzá egy 10-es skálabeosztást!

```
10    CLS
20    FOR I=0 TO 63 : SET(I,24) : NEXT I
30    FOR I=0 TO 47 : SET(32,I) : NEXT I
40    FOR I=32 TO 63 STEP 10 : SET(I,23) : SET(64–I,23) : NEXT I
50    FOR I=24 TO 47 STEP 10 : SET(31,I) : SET(31,48–I) : NEXT I
60    STOP
```

K.20. Írjunk programot, amely egyenletes sebességgel mozgat a képen egy pontot, a pont a kép széléről visszaverődik.

```
10    CLS
20    Y=24 : X=32 : SET(X,Y) : REM KEZDŐPONT
30    S=2*3.14*RND(0) : REM KEZDŐSZÖG
40    P=SIN(S) : Q=COS(S) : REM Y,X IRÁNYÚ ELMOZDULÁS
50          IF Y+P<0 OR Y+P>47 THEN P=–P : REM VISSZAVERŐDÉS
60          IF X+Q<0 OR X+Q>127 THEN Q=–Q : REM VISSZAVERŐDÉS
70          RESET(X,Y)
80          X=X+Q : Y=Y+P
90          SET(X,Y)
100   GOTO 50
```

K.21. Írjunk programot, amely egy szabadon eső test mozgását rajzolja ki!

```
10    CLS : G=9.81
20    X=10 : S=0 : REM OSZLOP, SOR
30    SET(X,S)
```

```
40      FOR T=0 TO 4 STEP .1
50          S=G/2*T*T
60          IF S>47 THEN 80
70          SET(X,S) : REM ÚJ PONT KIGYÚJTÁSA
80      NEXT T
90      STOP
```

K.22. Készítsünk rajzoló programot, amely az L(le), F(fel), J(jobbra), B(balra) billentyűk lenyomására rajzol a képernyőre!

```
10      CLS
20      A=0 : B=0 : REM KEZDŐPONT
30          SET(A,B)
40          V$=INKEY$
50          IF V$="J" AND A<127 THEN A=A+1
60          IF V$="B" AND A>0 THEN A=A-1
70          IF V$="L" AND B<47 THEN B=B+1
80          IF V$="F" AND B>0 THEN B=B-1
90      GOTO 30
```

Véletlenszámok

K.23. Írjunk programot, amely N kockadobás után az egyes dobásértékek gyakoriságát írja ki!

```
10      CLS : PRINT "KOCKADOBÁS" : PRINT
20      PRINT "DOBÁSOK SZÁMA";
30      INPUT N : IF N<1 THEN 20
40      DIM A(6)
50      FOR I=1 TO N
60      X=RND(6)
70      A(X)=A(X)+1
80      NEXT I
90      FOR I=1 TO 6
100         PRINT I; "ÉRTÉKŰ DOBÁS:"A(I)
110     NEXT I
120     STOP
```

1. Elemi feladatok a BASIC nyelv önálló tanulásához

K.24. Írjunk programot, amely pénzfeldobást szimulál és számolja az egyes dobások számát!

```
 10    CLS : PRINT "PÉNZFELDOBÁS" : PRINT
 20    PRINT "DOBÁSOK SZÁMA";
 30    INPUT N : IF N<1 THEN 20
 40    F=0 : I=0
 50    FOR J=1 TO N
 60          IF RND(0)<.5 THEN F=F+1 ELSE I=I+1
 70    NEXT J
 80    PRINT "FEJEK SZÁMA:";F
 90    PRINT "ÍRÁSOK SZÁMA:";I
100    STOP
```

K.25. Írjunk programot, amely egy pontot véletlenszerűen mozgat a képernyőn!

```
 10    CLS
 20    Y=24 : X=32 : REM KEZDŐPONT
 30    SET(X,Y)
 40          P=RND(3)-2 : Q=RND(3)-2 : REM ELMOZDULÁS
 50          IF Y+P<0 OR Y+P>47 OR X+Q<0 OR X+Q>127 THEN 40 :
                     REM KÉPHATÁR
 60          RESET(X,Y)
 70          X=X+Q : Y=Y+P : REM ÚJ PONT
 80          SET(X,Y)
 90    GOTO 40
```

K.26. A gép gondol egy számot 1 és 50 között, a játékosnak kell kitalálni. (A gép nagyon kis eséllyel csaljon!) Írjunk erre a feladatra egy játékprogramot!

```
 10    CLS : PRINT "SZÁMKITALÁLÓS" : PRINT
 20    PRINT "GONDOLTAM EGY SZÁMOT 1 ÉS 50 KÖZÖTT, TALÁLD KI!"
 30    X=RND(50) : REM A GONDOLT SZÁM
 40    IF RND(0)<.001 THEN X=X+RND(0) : REM CSALÁS
 50    S=1
 60          PRINT "MI A TIPPED"; : INPUT A
 70          IF A=X THEN 130 : REM TALÁLT
 80          IF A<X THEN PRINT "NAGYOBBAT"; ELSE PRINT "KISEBBET";
 90          PRINT "GONDOLTAM"
100          S=S+1
110          IF S>50 THEN PRINT "SOSE FOGOD ELTALÁLNI!" : STOP
120    GOTO 60
130    PRINT "ELTALÁLTAD";S;"LÉPÉSBŐL
140    STOP
```

Szövegkezelés

K.27. Írjunk programot, amely beolvas egy nevet, majd megmondja, hogy hány betűből áll (az angol ABC 26 nagybetűjét használjuk)!

```
 10    CLS : PRINT "BETŰK SZÁMA" : PRINT : CLEAR 1000
 20    PRINT "KÉREM A NEVET";
 30    INPUT N$ : IF LEN (N$)=0 THEN 20
 40    S=0
 50    FOR I=1 TO LEN(N$)
 60          X$=MID$(N$,I,1) : REM I. JEL
 70          IF X$>="A" AND X$<="Z" THEN S=S+1 : REM BETŰ-E?
 80    NEXT I
 90    PRINT "A BETŰK SZÁMA:";S
100    STOP
```

K.28. Készítsünk programot, amely megállapítja, hogy egy beolvasott szövegben mennyi magánhangzó van!

```
 10    CLS : PRINT "MAGÁNHANGZÓK SZÁMA" : PRINT : CLEAR 1000
 20    DATA 5,A,E,I,O,U
 30    READ M : REM AZ ISMERT MAGÁNHANGZÓK SZÁMA
 40    DIM M$(M)
 50    FOR I=1 TO M: READ M$(I) : NEXT I
 60    PRINT "KÉREM A SZÖVEGET";
 70    INPUT S$ : IF LEN(S$)=0 THEN 60
 80    S=0
 90    FOR I=1 TO LEN(S$)
100          X$=MID$(S$,I,1) : REM I. BETŰ
110          FOR J=1 TO M
120                IF X$=M$(J) THEN S=S+1 : REM HA MAGÁNHANGZÓ
130          NEXT J
140    NEXT I
150    PRINT "MAGÁNHANGZÓK SZÁMA:";S
160    STOP
```

K.29. Készítsünk programot, amely egy adott szövegben minden A-betűt kicserél E-re!

```
 10    CLS : PRINT "BETŰ-CSERE" : PRINT : CLEAR 1000
 20    PRINT "KÉREM A SZÖVEGET";
 30    INPUT S$ : IF LEN(S$)=0 THEN 20
 40    E$="" : REM EREDMÉNY SZÖVEG
```

```
 50    FOR I=1 TO LEN(S$)
 60        X$=MID$(S$,I,1) : REM I. BETŰ
 70        IF X$="A" THEN E$=E$+"E" ELSE E$=E$+X$
 80    NEXT I
 90    PRINT "AZ ÁTALAKÍTOTT SZŐVEG: ";F$
100    STOP
```

Adattárolás kazettán

K.30. Kazettán tároljuk emberek nevét és születési évét. Készítsünk programot, amely ezeket szétválogatja az 1945 előtt, ill. után születettekre, majd ezeket külön-külön kiírja kazettára! Egy adatsor max. 100 adatot tartalmazhat, a 0 születési év jelzi a végét.

```
 10    CLS : PRINT "SZÉTVÁLOGATÁS" : PRINT : CLEAR 1000
 20    PRINT "ÁLLÍTSD A KAZETTÁT AZ ADATSORHOZ!"
 30    GOSUB 500 : REM VÁRAKOZÁS
 40    INPUT ■1,V$ : REM AZ ELSŐ ADAT= "ADATSOR"
 50    IF V$<>"ADATSOR"THEN 40
 60    DIM N$(100),E(100)
 70    I=0
 80        I=I+1
 90        INPUT ■1,N$(I),E(I)
100    IF E(I)>0 THEN 80 : REM AMÍG ÉRVÉNYES ADAT JÖTT
110    PRINT "1. ADATSOR FELVÉTELE"
120    GOSUB 500 : REM VÁRAKOZÁS
130    PRINT ■1,"ADATSOR1"
140    FOR J=1 TO I-1
150        IF E(J)<1945 THEN PRINT ■1,N$(J),E(J)
160    NEXT J
170    PRINT ■1,N$(I),E(I) : REM UTOLSÓ ADAT
180    PRINT "2. ADATSOR FELVÉTELE"
190    GOSUB 500 : REM VÁRAKOZÁS
200    PRINT ■1,"ADATSOR2"
210    FOR J=1 TO I-1
220        IF E(J)>=1945 THEN PRINT ■1,N$(J),E(J)
230    NEXT J
240    PRINT ■1,N$(I),E(I) : REM UTOLSÓ ADAT
250    STOP
500    PRINT "HA FELKÉSZÜLTÉL, NYOMJ LE EGY BILLENTYŰT!";
510    IF INKEY$="" THEN 510
520    RETURN
```

2. Gyakorló feladatok kezdők számára

GY.1. Ismerjük két kör középpontjának koordinátáit és sugaraikat. Határozzuk meg kölcsönös helyzetüket (pl. koncentrikusak, metszik egymást, az egyik belülről érinti a másikat stb.)!

GY.2. Az emberek vércsoportját egy génpár határozza meg, amelynek mindegyik tagja lehet "a", vagy "b" vagy "0". Az "a" és a "b" is domináns a "0" felett, ők maguk pedig "egyenlő erősek". Írjunk programot, amely egy adott génpár ismeretében meghatározza az illető vércsoportját!

GY.3. Az emberek vércsoportját egy génpár határozza meg, amelynek mindegyik tagja lehet "a", vagy "b" vagy "0". "A" vércsoportúak az "aa" és az "a0" genotípusba tartozók, "B" vércsoportúak a "bb" és "b0" genotípusúak, "AB" vércsoportúak az "ab" genotípusúak és "0" vércsoportúak a "00" genotípusúak. Ismerjük két szülő vércsoportját, határozzuk meg, hogy gyermekeik milyen vércsoportúak lehetnek (egy gyermek mindegyik szülőjének egyik génjét örökli)!

GY.4. Az általános gáztörvény alakja: P*V=R*T, ahol P a gáz nyomása, V a térfogata, T a hőmérséklete, R pedig a gázállandó(8.31432). Írjunk programot, amely P,V és T közül bármelyiket ki tudja számolni a másik kettő ismeretében! (1 mól mennyiségű gázról van szó, a mértékegységek rendre: pascal, köbméter, kelvin.)

GY.5. Egy szerkezetben két, egymáshoz kapcsolódó fogaskerék egyikének egyetlen fogára egy olajcseppet teszünk. Számoljuk ki, hogy ez a másik keréken (és saját magán) hány helyen fog olajfoltot hagyni! (Adott a két fogaskerék fogszáma.)

GY.6. Adott egy hónapra minden nap átlaghőmérséklete. Keressünk egy olyan egyhetes időszakot, amikor az átlaghőmérséklet a legnagyobb volt!

GY.7. Írjunk programot, amely úgy hoz létre egy számsorozatot, hogy első M tagja 1, s ezután minden M-nél nagyobb sorszámú eleme a közvetlenül előtte levő M db szám összege! (M–Fibonacci számok)

GY.8. Készítsünk függvénytáblázatot a színusz, a koszinusz és a tangens függvény értékeiről 0 és 2π között 0.1 lépésközzel!

GY.9. Adott N ember jövedelme. Mindegyik nyugdíját úgy kell kiszámolni, hogy vesszük a jövedelmének X százalékát 100 Ft-ra kerekítve. A nyugdíj minimális értéke Y forint, ha az előbbi számítással kevesebb jönne ki, akkor automatikusan erre az összegre kell emelni.

2. Gyakorló feladatok kezdők számára

GY.10. Ismerjük N ember magasságát, különböző magasságúak. Adjuk meg, hogy melyik közülük a legmagasabb!

GY.11. Ismerjük N ember magasságát, lehetnek köztük egyenlő magasságúak. Adjuk meg, hogy melyek közülük a legmagasabbak!

GY.12. Egy egyenes szakaszon megmértük a felszín tengerszint feletti magasságát adott lépésközzel. Adjuk meg a mért szakaszon a legalacsonyabb hegycsúcs helyét és magasságát!

GY.13. Ismerjük a síkon N db pont koordinátáit, határozzuk meg, melyik van legközelebb az origóhoz!

GY.14. Egy térképre egy jól megválasztott négyzetrácsot helyeztünk és minden rácspontnak megadtuk a tengerszint feletti magasságát. Határozzuk meg a térképen a hegycsúcsok helyeit!

GY.15. Készítsünk programot, amely egy függvény gyökhelyét tudja meghatározni az intervallumfelezés módszerével!

GY.16. Egy lövészversenyen minden versenyző 20 lövést ad le, amelyből a legjobb tizet értékelik. Írjunk olyan programot, amely egy versenyző 20 lövésének eredményéből kiszámolja a legjobb 10 összegét!
(Egy-egy lövés eredménye 0-tól 10-ig terjedhet.)

GY.17. András és Béla úgy játszanak, hogy megállnak egymástól pontosan 100 lépésre, majd felváltva egymás felé mennek – egy alkalommal legalább 2, de legfeljebb 9 lépést tesznek meg. Az nyer, aki a másik helyére tud lépni, vagy átlép azon. A vesztes kólát fizet. Andrásnak nincs pénze. Segítsünk rajta: írjunk programot, amely Béla minden lépése után megmondja, hogy András hányat lépjen előre, hogy ne kelljen fizetnie! Azt tudjuk, hogy először András lép.

GY.18. Állítsunk fel egy kör mentén N bábut! Döntsük fel a K-adik álló bábut, utána a 2K-adik álló bábut, a 3K-adikat stb. A döntögetést addig folytassuk, amíg egyetlen álló bábu sem marad! (Pl. ha N=10 és K=7, akkor a bábuk a következő sorrendben dölnek fel: 7,4,2,1,3,6,10,5,8,9.) Készítsünk programot, amely kinyomtatja a bábuk sorszámát a feldöntésük sorrendjében!

GY.19. Írj programot, amely kört rajzol a képernyőre!

GY.20. Készítsünk programot, amely végpontjaival megadott szakaszt rajzol a képernyőre!

GY.21. Készítsünk programot, amely egyenes szakaszokból álló síkidomot tud a képernyőre rajzolni!

GY.22. Készítsünk programot, amely egy adott egyváltozós függvényt tud kirajzolni a képernyőre!

GY.23. Ismerünk egy dátumot, "hónap, nap" formában, valamint tudjuk, hogy az adott év január elseje a hét milyen napjára esett. Írjunk programot, amely a fenti napról meghatározza, hogy a hét melyik napja!

GY.24. Erősen összetett számoknak nevezzük azokat a természetes számokat, amelyek osztói száma több, mint bármely náluk kisebb természetes szám osztói száma. Készítsünk programot, amely kiírja N-ig az erősen összetett számokat!

GY.25. Tökéletesnek nevezzük azokat a természetes számokat, amelyek osztói összege (az 1-et beleértve, az adott számot nem) kiadja magát a számot. Írjunk programot, amely kiírja N-ig a tökéletes számokat!

GY.26. Két természetes számot barátságosnak nevezünk, ha az egyik osztói összege megegyezik a másikkal, a másik osztói összege pedig az egyikkel (az 1 beleszámít az összegbe, maguk a számok pedig nem). Olyan program készítendő, amely megadja N-ig a barátságos számpárokat!

GY.27. Egy osztályba N tanuló jár és M tantárgyat tanulnak, de nem minden tárgyat mindenki (pl. nyelvek, testnevelésből felmentett stb.). Az I-edik tanuló eredménye a J-edik tárgyból legyen X(I,J), ami 0, ha az illető nem tanulja a megfelelő tárgyat. Készítsünk programot, amely kiszámítja minden tanuló átlageredményét, illetve az osztály tantárgyankénti átlagait!

GY.28. Egy, az Egyenlítő fölött keringő mesterséges holdról 5 kilométerenként megmértük a felszín tengerszint feletti magasságát. Elképzelésünk szerint ott van a tenger, ahol a mérési eredményünk 0. A mérést szárazföld fölött kezdtük el és fejeztük be. Adjuk meg, hogy a mért szakaszon mennyi sziget van!

2. Gyakorló feladatok kezdők számára

GY.29. Egy, az Egyenlítő fölött keringő mesterséges holdról 5 kilométerenként megmértük a felszín tengerszint feletti magasságát. Elképzelésünk szerint ott van a tenger, ahol a mérési eredményünk 0. A mérést szárazföld fölött kezdtük el és fejeztük be. Adjuk meg, hogy a mért szakaszon mennyi a tenger szélessége!

GY.30. Egy, az Egyenlítő fölött keringő mesterséges holdról 5 kilométerenként megmértük a felszín tengerszint feletti magasságát. Elképzelésünk szerint ott van a tenger, ahol a mérési eredményünk 0. A mérést szárazföld fölött kezdtük el és fejeztük be. Határozzuk meg a mért szakaszon a legszélesebb sziget szélességét!

GY.31. Az egyik legegyszerűbb titkosírást az ABC betűinek ciklikus eltolásából kaphatjuk meg. Olyan programot kell írni, amely egy szöveg és egy eltolási konstans ismeretében egy szöveget tud kódolni úgy, hogy minden betű helyére a nála ABC-ben adott hellyel hátrább levő betűt tesszük!

GY.32. Egy titkosírást a következő módon készítünk el: helyezzünk el betűket egy 10x10-es mátrixban, s erre tegyünk rá egy kockás papírlapot, amelyen bizonyos kockák ki vannak vágva. A látható betűk alkotják a szöveg első részét. Ezután fordítsuk el ezt a lapot 90 fokkal jobbra és ismételjük meg a látható betűk kigyűjtését. Ezt a műveletet még kétszer elvégezve megkapjuk a teljes szöveget. Írjunk programot, amely egy így titkosított szöveg megfejtését készíti el!

GY.33. Írjunk programot, amely egy (néhány soros) szöveget beolvas, tárol, kiír, majd egy (inputként) megadott betűcsoport minden egyes előfordulását kicseréli egy (ugyancsak inputként megadott) betűcsoportra!

GY.34. Írjunk programot, mely nyomtatványt tud készíteni, az alábbi követelmények szerint:
A szöveget meghatározott méretű sorokra és lapokra tagoljuk, a sorokat szóköznél, ill. írásjel után lehet tördelni.

GY.35. Készítsünk olyan programot, amellyel egy szöveget lehet kódolni morze-jelekkel, illetve egy morze-üzenetet vissza lehet fejteni!

GY.36. Adott egy szöveg, határozzuk meg benne a magas, a mély és a vegyes hangrendű szavak számát!

GY.37. Adott egy maximum 20 soros vers. Határozzuk meg a rímképletét (az utolsó két szótagot vizsgáljuk és nem csak a tiszta rímeket, hanem az asszonáncokat is nézzük)!

GY.38. Ismerjük egy labdarúgó-bajnokság összes mérkőzésének eredményét, készítsük el a bajnoki táblázatot!

GY.39. Rendelkezésünkre állnak egy futóverseny adatai: a benevezett versenyzők rajtszáma, neve és nemzetisége. Ezek rajtszám szerint rendezve vannak, de a rajtszámokat nem folytonosan adták. Ugyancsak adva vannak az elért eredmények (rajtszám és az idő másodpercekben) a befutás sorrendjében. Készítsük el az eredménylistát (helyezés, név, nemzetiség, idő)!

3. Gyakorló feladatok haladók számára

GY.40. Mesterséges holdak működésekor fontos feladat a helyi függőleges (a mesterséges holdat a bolygó középpontjával összekötő szakasz) irányának meghatározása. Ezt általában úgy végzik el, hogy a bolygó várható helyéről egy infravörös fényben készült képet készítenek (a bolygó fényessége itt lényegesen felülmúlja környezetének fényességét). Ha a bolygó képét (jó közelítéssel) körnek vesszük, akkor a helyi függőleges a kör középpontjába mutat (azaz a középpont koordinátáit kell meghatározni). A fényképből egy digitalizált fénykép úgy készül, hogy egy négyzetrácsot helyezünk rá és minden pontban feljegyezzük az intenzitás értékét. Készítsünk olyan programot, amely egy digitalizált kép alapján megadja a helyi függőleges irányát (a középpont koordinátáit).

GY.41. Ismerjük egy megye városainak koordinátáit, valamint kör alakú vonzáskörzetük sugarait. Rajzoljuk ki a képernyőre azon pontokat, amelyek nem tartoznak egyetlen város vonzáskörzetébe sem!

GY.42. Adott egy céltábla, amelyre többször, de legfeljebb 20-szor lőttek. Egy-egy lövés eredményét a becsapódási hely koordinátáival adjuk meg. Értékeljünk ki egy céltáblát, ha az értékeléskor a 10 legjobb lövést kell figyelembe venni!

GY.43. Valaki 500 darabos magánkönyvtárát a következő adatok szerint tartja nyilván: szerző, cím, kiadó, évszám. Írjunk programot, amely a fenti 4 adat bármelyike szerint rendezve kiírja a katalógust!

GY.44. Egy raktárkészlet-nyilvántartásban minden áruról 4 féle adatot tárolunk (az áru kódja, a raktáron levő mennyiség, egységár, minimális készlet). Határozzuk meg a raktáron levő áru értékét és készítsünk listát azokról az árukról, amelyek raktáron levő mennyisége nem éri el a minimálisan szükséges mennyiséget!

GY.45. Van N db termékünk T1,T2,...,Tn. Az A vállalatnak ezekből rendre A1,A2,...,An kell, a B-nek B1,B2,...,Bn. Az egyes termékek tonnánkénti ára forintban: F1,F2,...,Fn. Program készítendő, mely kinyomtatja a következőket: a két vállalatnak együttesen melyik termékből kell a legtöbb, a termékek tonnánkénti árának átlagát, a termékenkénti költségeket (a két vállalatra együttesen), valamint az egyes vállalatok összköltségét, valamint a közös összköltséget.

GY.46. Egy vállalat számláit kazettán tároljuk. Egy rekord egy számlát tartalmaz, a következő adatokkal:

- számlaszám
- előző egyenleg
- bevétel
- kiadás

Az adatok a számlaszám szerint növekvő sorrendbe rendezettek, s ha több azonos számlaszám következik egymás után, akkor az "előző egyenleg" rovat csak az első rekordban van kitöltve. Határozzuk meg minden számlaszámhoz az új egyenlegeket, s külön jelöljük meg az 1000 forintnál kisebbeket!

GY.47. Földrengésekkel kapcsolatos adatok közül az alábbiakat tároljuk:
- az észlelés helye
- a rengés erőssége
- a rengés időtartama (másodperc)
- a rengés melyik földrészen volt
- a rengést előre jelezték-e

Készítsünk listát azokról a földrengésekről, amelyek időtartama meghaladta a 0.3 percet! Nyomtassuk az előre nem jelzett földrengések átlagos erősségét földrészenkénti bontásban! Adjuk meg, hogy melyik földrészen volt a legtöbb földrengést!

GY.48. Egy szállítási vállalat a következő adatokat jegyzi fel:
- szállítási távolság (kilométer)
- szállított mennyiség (tonna)

Egy kilométerre 1 tonna szállításáért a fuvardíj F forint, de a vállalat 10–19 km esetén 5%, 100 km felett 10% engedményt ad, ill. 2 tonnánál nagyobb mennyiség esetén további 3% engedményt ad.
Határozzuk meg az egyes szállításokért és az összesen járó fuvardíjat.

GY.49. Az ország 19 megyéjének településeire és Budapestre vonatkozóan adott az influenzás megbetegedések száma. Adatainak felépítése:
- a megye neve
- a helység kódja
- a megbetegedések száma

Írjunk programot, ami meghatározza a megbetegedések megyénkénti megoszlását százalékosan (azaz, hogy az ország összes megbetegedéseiből hány esett a megyére)! Határozzuk meg továbbá, hogy melyik megyében volt a legtöbb megbetegedés, és mennyi; illetve melyik városban volt a legtöbb és az mennyi volt!

GY.50. Házi telefonkönyvünket szeretnénk gépesíteni: a telefonkönyvben neveket, címeket és telefonszámokat tárolunk. Olyan programot kell írni, amelytől neveket kérdezhetünk, s az megmondja az illető telefonszámát!

3. Gyakorló feladatok haladók számára

GY.51. Adott háttértárolón (pl. kazettán) egy növényhatározó, amelyben virágok leírását tároljuk. Egy virágról a következőket tudjuk:

- neve
- színe
- nyílási évszaka
- minimális magassága
- maximális magassága

Készítsünk egy olyan programot, amely beolvasott virágleírásokhoz megadja az illető virág nevét, ha tudja, ill. ha több ilyen van, akkor az összeset.
Egy virágot a következők jellemeznek:

- színe
- szedési hónapja
- magassága

A program tudjon tetszőleges számú ilyen virágot meghatározni!

GY.52. Adott egy keresztrejtvény ábrája (fehér és fekete kockák). Írjuk bele a megfelelő fehér kockákba a keresztrejtvényeknél szokásos számokat, amelyekhez majd a megfejtendő kérdéseket kell adni. Ezenkívül készítsünk egy listát, hogy melyik számokhoz kell vízszintesen, ill. függőlegesen kérdéseket adni!

GY.53. Megadtuk egy űrfelvétel minden képpontjának az intenzitásértékét. Határozzuk meg a képen az éleket (a különböző területek határvonalait – az intenzitás nagyot változik)! Ehhez adjuk meg, hogy egy pont milyen kritérium alapján tartozik egy élhez!

GY.54. Megádtuk egy gyűrűs szénhidrogén szerkezetét úgy, hogy minden szénatomhoz meghatároztuk a kapcsolódó szénatomokat. Hagyjuk el belőle az olyan szénatomokat, amelyek nem tartoznak egyetlen gyűrűhöz sem, ill. nem kötnek össze két gyűrűt.

GY.55. Ismerjük két rádióadás-sorozat adásidőit, mindkettőt az adás kezdete szerint rendezve. Döntsük el, hogy a két adássorozatot adhatta-e ugyanaz az adó. (Biztosan nem adhatta: ha volt olyan időpont, amikor mindkét adás egyszerre szólt.)

GY.56. Egy hőelemes műszerrel 5 percenként mérjük egy kazán belsejének hőmérsékletét. A műszert időnként ki kell kapcsolni, ekkor 0 értéket olvashatunk le róla. Készítsük el a hőmérsékletváltozást leíró adatsorozatot, feltéve, hogy az ilyen szakaszokon lineárisan változott a hőmérséklet. (Az első és az utolsó mérés eredményét ismerjük.)

GY.57. Rajzoltassuk ki az összes lehetséges 3 lyukú buszjegylyukasztást! Azokat az eseteket, ahol a buszjegy fordítva került a lyukasztóba, vegyük azonosnak, és csak egyszer tüntessük fel!

GY.58. Galaktikus rádiózajból úgy tudunk használható jelsorozatot kiszűrni, hogy a vett jelsorozatban valamilyen periódikusságot keresünk. (Ez ugyanis csak valamilyen szabályossággal működő forrásból jöhet.) Adott egy ilyen jelsorozat, keressünk benne periódikusságot valamilyen pontossággal! (Ilyen módszert használtak pl. más értelmes lények után kutató programokban is.)

GY.59. Egy vállalat N állást hirdet meg. Jelentkezik rá ugyanannyi pályázó, és mindegyik közli, hogy melyik állásokat tudná betölteni. Döntsük el, hogy a pályázókkal betölthető-e minden állás, s ha igen, akkor adjuk meg, hogy melyiket melyik állásra vesszük fel!

GY.60. Adott egy csillagtérkép, ahol a csillagokat egy térbeli derékszögű koordinátarendszerben a koordinátáikkal adjuk meg (3 dimenziós térkép). Csillagsűrűsödésnek nevezzük azokat a lehető legnagyobb csoportokat, amelyekben legalább 'N' darab csillag van és bármely tagjáról bármely más tagjára el lehet jutni a csoport tagjain keresztül legfeljebb 'K' fényéves lépésekkel. Határozzuk meg térképünkön a csillagsűrűsödések számát!

V.1. Készítsünk programot a következő feladatra:

Adottak a Magyarországon élő emberek személyi számai, amely a következőket tartalmazza:

A: neme (1 jegyű szám) férfi: 1
nő: 2

B: születés dátuma (6 jegyű szám)
év a század nélkül
hónap
nap

C: 3 jegyű sorszám
D: 1 jegyű ellenőrző szám
Az ellenőrző számot az alábbi algoritmus alapján készítik:
D $(x^1+2*x^2+3*x^3+\ldots+10*x^{10}) \bmod 11$

ahol x^i az i-edik helyen levő szám.

A 10-es maradékot adó személyi szám nem kerül kiosztásra.

A készítendő program adja meg, hogy a mai napon született emberek között mely számokat nem fogják kiosztani.

V.2. Összesen 33 iskolába küldtek értesítést, ebből 17 a vidéki. Készítsünk algoritmust, amelynek végeredménye az a szám, ahány különböző permutáció rendelkezik azzal a tulajdonsággal, hogy mindegyik értesítés elküldése után a vidékre küldöttek vannak többségben!

V.3. Készítsünk algoritmust az alábbi – K és N (K<=N) természetes számoktól függő – S(N,K) mennyiség meghatározására:

N különböző tárgyat S(N,K)-féleképpen lehet K nem üres részre osztani. (Pl. K=3, N=5 esetén az 1 ! 2,5 ! 3,4 és a 2,5 ! 1 ! 3,4 nem különböző elosztások, de az 1 ! 3,5 ! 2,4 már különbözik tőlük.)

4. Versenyfeladatok

Az 1982. évi PROGRESSZ'82 verseny feladatai

V.4. Adott egy tömb, amely egy programozási nyelv kulcsszavait tartalmazza: mégpedig soronként egyet, nem feltétlenül balra igazítva – vagyis lehet, hogy egy vagy több szóköz előzi meg.

Írj olyan programot, amely az angol ABC összes betűjére megadja, hogy hány kulcsszó kezdődik azzal a betűvel!

V.5. Adott egy eszperantó nyelvű szöveg (max. 80 karakter), amelyben a nyelv "kalapos" betűit a normál betű utáni X jelzi.

Feladat: olyan program elkészítése, amely az adott szöveget két sorban írja vissza. Az alsó sorban a szöveg helyezkedik el, a felső sorban pedig a kalapos betűk fölött egy "–" helyezkedik el.

Az 1983. évi I. Országos Középiskolai Számítástechnikai Verseny feladatai

V.6. Készítsünk algoritmust a következő feladatra: minimalizálandó x+y+z az alábbi feltételek mellett

1. x,y,z nem negatív számok
2. ax+by+cz=d, ahol a,b,c,d egész paraméterek

V.7. Adva van véges sok (de elegendően nagy számú) azonos L hosszúságú cső. Szükségünk van N1 darab L1 hosszúságú és N2 darab L2 hosszúságú csőre. E célból feldarabolunk bizonyos számú L hosszúságú csövet. Készítsünk olyan algoritmust, ami minimalizálja a feldarabolandó csövek számát!

V.8. Egy gépre különböző munkák érkeznek előre ismert időpontokban. A munkákat meghatározott határidőre be kell fejezni. (Minden munkának saját beérkezési és határideje van.) A munkák elvégzéséhez szükséges idő ismert. Egy munka elvégzése nem szakítható meg.

Készítsünk algoritmust a következő feladatra: adja meg a gép a munkák elvégzésének olyan sorrendjét, hogy a legtöbbet késő munka (határideje után készül el) késése a lehető legkisebb legyen. (A határidő megsérthető, a beérkezési idő nem.)

Az 1984. évi II. Országos Középiskolai Számítástechnikai Verseny feladatai

V.9. 3 jegyű Armstrong számoknak nevezik azokat a 3 számjegyből álló számokat, amelyek számjegyei köbének összege kiadja magát a számot.
Írj programot az összes 3 jegyű Armstrong szám kiszámítására!

V.10. Adott egy körgyűrű és egy pont. Írj programot, amely meghatározza viszonyaikat! A körgyűrű és a pont megadását is neked kell definiálnod!

V.11. Írj programot, amely beolvassa két halmaz elemeit, majd kiírja a két halmaz metszetét és unióját!

V.12. Írj programot, amely táblázatot készít egy H magasságból eső test helyzeti és mozgási energiájáról, adott lépésközzel!

V.12. Írj programot, amely kiszámítja N szám legnagyobb közös osztóját!

V.13. Írj programot, amely megszámolja egy adott magyar nyelvű szöveg szótagszámát!

V.14. Adott N db síkbeli szakasz, amelyek egymáshoz csatlakoznak. Minden szakasznak adott a hossza és az előző szakasszal bezárt szöge (az első szakasznál az X-tengellyel bezárt szöget adjuk meg). Az első szakasz egyik végét helyezzük az origóba, majd csatlakoztassuk ehhez sorba a szakaszokat! Irjunk BASIC programot, amely az így megadott vonalsorozat másik végpontját határozza meg!

V.15. BASIC program készítendő, amely két INPUT-tal bevitt, legfeljebb ötjegyű hexadecimális pozitív egész szám összegét képezi és kijelzi. A program tartalmazzon védelmet a meg nem felelő számok és karakterek bevitele ellen.

V.16. Készítsünk programot, amely megadja a $(k, \sqrt{k+1} \cdot (\sin^2 k))$, $(k=0,1,\ldots,999)$ és az (1000,0) koordinátájú pontok konvex burkát.
Egy ponthalmaz konvex burkán azt a legszűkebb konvex ponthalmazt értjük, amely tartalmazza az eredeti halmaz tetszőleges pontpárját összekötő szakaszt, a végpontjait is beleértve. Nyilvánvaló, hogy a konvex burok egy konvex poligon lesz, adjuk meg a poligon csúcspontjainak koordinátáit.

4. Versenyfeladatok

Az 1984. évi novoszibirszki válogató verseny feladatai

V.17. Írjunk BASIC nyelvű programot! Adott egy sok szóból álló szöveg, melyben nincsenek írásjelek; a szavakat szóközök választják el. Írassuk ki a képernyőre a következő szabályok betartásával:

1. Minden sorba a lehető legtöbb szót kell kiírni.
2. Szavakat elválasztani nem lehet.
3. A sorok belsejébe helyezendő szóközökkel a szöveget a jobb oldali margóhoz is igazítani kell.

V.18. Készítsünk folyamatábrát, mely az N=1,2,3,...20 számhalmazból kiválasztja az összes olyan számhármast, melynek összege nem nagyobb 30-nál és prímszám.

V.19. A számítógép véletlenszám-generátorával pénzfeldobási sorozatot szimulálunk. Készítsünk grafikont, mely 0-tól 1000 dobásig 10 dobásonként szemlélteti, hogy az addig elvégzett dobásokban mennyi a fej és az írás dobások száma közötti különbség!

V.20. Egy zárt teremben 30 dohányos ember van. Minden időegység végén 20% annak a valószínűsége, hogy eggyel nő, 20%, hogy eggyel csökken az égő cigaretták száma, 60%, hogy nincs változás. Egy cigaretta időegység alatt egy egységnyi füstmennyiséget termel. Ha a teremben a füstmennyiség eléri a 200 egységet, akkor a számítógép elindítja az elszívóberendezést, amely egy időegység alatt 30 egységgel csökkenti a teremben a füstmennyiséget. Ha a teremben a füstmennyiség 40-nél kevesebb, akkor a gép kikapcsolja az elszívót. Készítsünk programot, amely:

a) vezérli az E változó 0 vagy 1 értékével az elszívó ki-, ill. bekapcsolását, a füstmennyiségtől függően,

b) kiírja időegységenként a füstmennyiség értékét, a dohányzók számát, és az elszívó kapcsolójának állását!

Az 1984. évi Szolnok megyei középiskolai tanulók I. Számítástechnikai Versenyének feladatai

V.21. Készítsen BASIC programot a következő feladat megoldására! "N" napig mértünk, naponként K alkalommal hőmérsékletet. A program adja meg a napi hőmérséklet-maximumok közül a legkisebb értékét!

V.22. Ismerjük "N" ember magasságát, programmal határozzuk meg azok sorszámait, akik a legmagasabbak (több egyenlő magasságú is lehet)!

V.23. A Balaton fölé egy négyzetrácsot helyeztünk és minden pontjában megmértük a víz hőmérsékletét. (Ahol nem víz volt, ott a mérés értéke 0, a víz esetén pedig >0.) Készítsen BASIC-programot, amely megadja a víz átlaghőmérsékletét, valamint a part menti víz átlaghőmérsékletét (az a mérés volt part mellett, amelynek 8 szomszédja közül legalább egy parti mérés)!

V.24. A számítógép véletlenszám-generátorát felhasználva elvégeztünk "N" db pénzfeldobást. Írjon programot, amely megadja a leghosszabb fej-sorozat hosszát!

V.25. 5 kilométerenként megmértük a felszín tengerszint feletti magasságát. Ott van sziget, ahol a mérés 0-nál nagyobb volt, ahol pedig 0-t mértünk, ott tenger. A méréseket szárazföld fölött kezdtük és végeztük. Írjon BASIC programot, amely megadja, hogy a mért szakaszon hány sziget van!

FELADATMEGOLDÁSOK

GY.1. "körök"

A két kör egymással 7 különböző viszonyban lehet. Jelölje R1,R2 a körök sugarait, D pedig középpontjaik távolságát!

1. D=0 és R1=R2	→	azonosak
2. D=0 és R1<>R2	→	koncentrikusak
3. D<ABS(R1–R2)	→	egyik tartalmazza a másikat
4. D=ABS(R1–R2)	→	belülről érintik egymást
5. D>ABS(R1–R2) és D<R1+R2	→	két pontban metszik egymást
6. D=R1+R2	→	kívülről érintik egymást
7. D>R1+R2	→	elkerülik egymást

Ennek megfelelően a BASIC program a középpontok koordinátáinak beolvasásával így alakul:

```
 10   CLS: PRINT "KÖRÖK VISZONYA" : PRINT
 20   INPUT "1. KÖR KÖZÉPPONT KOORDINÁTÁI";X1,Y1
 30   INPUT "1. KÖR SUGARA";R1 : IF R1<=0 THEN 30
 40   INPUT "2. KÖR KÖZÉPPONT KOORDINÁTÁI";X2,Y2
 50   INPUT "2. KÖR SUGARA";R2 : IF R2<=0 THEN 50
 60   D=SQR ( (X1–X2)↑2+(Y1–Y2)↑2)
 70   IF D=0 THEN IF R1=R2 PRINT "AZONOSAK" : STOP ELSE PRINT
      "KONCENTRIKUSAK" : STOP
 80   A=ABS(R1–R2)
 90   IF D<A THEN PRINT "AZ EGYIK TARTALMAZZA A MÁSIKAT" : STOP
100   IF D=A THEN PRINT "BELÜLRŐL ÉRINTIK EGYMÁST" : STOP
110   B=R1+R2
120   IF D<B THEN PRINT "KÉT PONTBAN METSZIK EGYMÁST" : STOP
130   IF D=B THEN PRINT "KÍVÜLRŐL ÉRINTIK EGYMÁST" : STOP
140   PRINT "ELKERÜLIK EGYMÁST" : STOP
```

• • •

GY.2. "vércsoport"

A vércsoportot egy génpár határozza meg. Mindegyik tagja lehet "a", vagy "b" vagy "0". Egy ember
"A" vércsoportú, ha "aa" vagy"a0" génpárja van;
"B" vércsoportú, ha "bb" vagy "b0" génpárja van;
"AB" vércsoportú, ha "ab" génpárja van;
"0" vércsoportú, ha "00" génpárja van.

Így a megoldó algoritmusnak be kell olvasnia a génpár két elemét (A$,B$) majd ebből meg kell határoznia a vércsoportot. Arra kell még ügyelni, hogy az "a0" és a "0a" pár ugyanazt jelenti: itt csak felírásbeli sorrend van.

Program:

```
Be: A$,B$        [0,a,b lehet az értékük]
Ha  A$="0" és B$="0" akkor Ki:"0–ÁS VÉRCSOPORT"
Ha  A$="a" és B$="b" vagy A$="b" és B$="a"
            akkor Ki:"AB-S VÉRCSOPORT"
Ha  A$="a" és B$<>"b" vagy A$<>"b" és B$="a"
            akkor Ki: "A-S VÉRCSOPORT"
Ha  A$="b" és B$<>"a" vagy A$<>"a" és B$="b"
            akkor Ki:"B-S VÉRCSOPORT"
Program vége.
```

Egy ügyes trükkel azonban elkerülhetjük, hogy a programban mindkét változattal kelljen foglalkoznunk. Rendezzük sorba a két gént (itt 0 < a < b)!

Program:

```
Be: A$,B$        [0,a,b lehet az értékük]
Ha  B$="0" akkor Csere(A$,B$)
Ha  B$="0" akkor Ki:"0-ÁS VÉRCSOPORT"
      különben Ha A$="a" és B$="b" akkor Ki:"AB-S VÉRCSOPORT"
          különben Ha B$="a" akkor Ki:"A-S VÉRCSOPORT"
              különben Ki:"B-S VÉRCSOPORT"
Program vége.
```

A második változat BASIC programja (a bonyolult elágazást sorokra tagolva):

```
10    CLS  : PRINT "VÉRCSOPORT MEGHATÁROZÁS" : PRINT
20    INPUT  "A KÉT GÉN";A$,B$
30    IF A$<>"a" AND A$<>"b" AND A$<>"0" THEN 20
40    IF B$<>"a" AND B$<>"b" AND B$<>"0" THEN 20
50    IF A$>B$ THEN X$=A$ : A$=B$ : B$=X$
60    IF B$="0" THEN PRINT "0-ÁS VÉRCSOPORT" : STOP
70    IF A$="a" AND B$="b" THEN PRINT "AB-S VÉRCSOPORT" : STOP
80    IF B$="a" THEN PRINT "A-S VÉRCSOPORT" ELSE
            PRINT "B-S VÉRCSOPORT"
90    STOP
```

• • •

GY.3. "utód vércsoportja"

Nézzük meg, hogy a szülők lehetséges vércsoport-párosításából milyen utód születhet:

Szülők		Utód	Vércsoport	Genotípus
A,A	:	A,0	A	aa,a0
A,B	:	A,B,AB,0	B	bb,b0
A,AB	:	A,B,AB	AB	ab
A,0	:	A,0	0	00
B,B	:	B,0		
B,AB	:	A,B,AB		
B,0	:	B,0		
AB,AB	:	A,B,AB		
AB,0	:	A,B		
0,0	:	0		

A feladat ezek szerint megoldható lenne egy olyan elágazással, amely 10 lehetőséget tartalmaz, eljárhatunk azonban egyszerűbben is. Négyféle eredményt adhat programunk az utód vércsoportjára: A,B,AB,0. Fogalmazzuk meg azt, hogy egy adott vércsoportú utód milyen szülőktől származhat!

Program:

```
Be: A$,B$
Ki: "Lehetséges vércsoportok:"
Ha A$<>"AB" és B$<>"AB" akkor Ki: "0"
Ha A$="A" vagy A$="AB" vagy B$="A" vagy B$="AB" akkor Ki: "A"
Ha A$="B" vagy A$="AB" vagy B$="B" vagy B$="AB" akkor Ki: "B"
Ha A$<>"0" és B$<>"0" és A$<>B$ vagy A$="AB" és B$="AB"
        akkor Ki: "AB"
```

Program vége.

A BASIC program az egyes vércsoportokat egy sorba írja ki, egymástól szóközökkel elválasztva:

```
10    CLS : PRINT "UTÓD LEHETSÉGES VÉRCSOPORTJAI" : PRINT
20    INPUT "A SZÜLŐK VÉRCSOPORTJAI";A$,B$
30    PRINT "LEHETSÉGES VÉRCSOPORTOK: ";
40    IF A$<>"AB" AND B$<>"AB" THEN PRINT "0 ";
50    IF A$="A" OR A$="AB" OR B$="A" OR B$="AB" THEN PRINT "A ";
60    IF A$="B" OR A$="AB" OR B$="B" OR B$="AB" THEN PRINT "B ";
70    IF A$<>"0" AND B$<>"0" AND A$<>B$ OR A$="AB" AND B$="AB"
            THEN PRINT "AB"
80    STOP
```

• • •

5. Feladatmegoldások kezdők számára

GY.4. "gáztörvény"

Olyan programot kell készítenünk, amely egy képletben szereplő három mennyiség bármelyikét ki tudja számolni a további kettőből. A program működését egy menü (választék) kiírásával kezdi, majd a felhasználó választásától függően más-más funkciót hajthat végre. Az egyes funkciók végrehajtása után megkérdezi, hogy kell-e még számolni, s ha nem, akkor befejezi működését.

Program:

```
        Ciklus
                Menükiírás
                I:=választás sorszám
                Ha
                    1.  I=1 akkor Nyomás számítás
                    2.  I=2 akkor Térfogat számítás
                    3.  I=3 akkor Hőmérséklet számítás
                Elágazás vége
        amíg kell előlről
        Ciklus vége
Program vége.
```

A BASIC programban a 3 irányú elágazást ON GOSUB utasítással valósítjuk meg. Ha az utolsó kérdésre beírt válasz első betűje I, akkor a választ IGEN-nek vesszük, egyébként NEM-nek.

```
 10   R=8.31432 : REM GÁZÁLLANDÓ
 20   CLS  : PRINT "ÁLTALÁNOS GÁZTÖRVÉNY" : PRINT
 30   PRINT "ALAKJA: P*V=R*T"
 40   PRINT "P – NYOMÁS, V – TÉRFOGAT, T – HŐMÉRSÉKLET, R=8.31432";
            " – GÁZÁLLANDÓ"
 50   PRINT "AZ ADATOKAT SI MÉRTÉKEGYSÉGBEN KELL MEGADNI!"
 60   PRINT : PRINT "LEHETŐSÉGEK"
 70   PRINT TAB(10);"1. NYOMÁS SZÁMÍTÁS"
 80   PRINT TAB (10);"2. TÉRFOGAT SZÁMÍTÁS"
 90   PRINT TAB(10);"3. HŐMÉRSÉKLET SZÁMÍTÁS"
100   INPUT "VÁLASZTÁS";I : IF I<1 OR I>3 OR I<>INT (I) THEN 20
110   ON I GOSUB 200,300,400
120   INPUT "KELL MÉG SZÁMOLNI (I,N)";V$
130   IF LEN(V$)=0 THEN 120 ELSE V$=LEFT$(V$,1)
140   IF V$="I" OR V$="i" THEN 20 ELSE STOP
200   REM NYOMÁS SZÁMÍTÁS
```

```
210     CLS : INPUT "TÉRFOGAT, HŐMÉRSÉKLET";V,T
220     IF V<=0 OR T<=0 THEN 210
230     P=R*T/V
240     PRINT "NYOMÁS=";P;"PASCAL"
250     RETURN
300     REM TÉRFOGAT SZÁMÍTÁS
310     CLS : INPUT "NYOMÁS, HŐMÉRSÉKLET";P,T
320     IF P<=0 OR T<=0 THEN 310
330     V=R*T/P
340     PRINT "TÉRFOGAT=";V;"KÖBMÉTER"
350     RETURN
400     REM HŐMÉRSÉKLET SZÁMÍTÁS
410     CLS : INPUT "NYOMÁS, TÉRFOGAT";P,V
420     IF P<=0 OR V<=0 THEN 410
430     T=P*V/R
440     PRINT "HŐMÉRSÉKLET=";T;"KELVIN"
450     RETURN
```

• • •

GY.5. "fogaskerék"

Meg kell határoznunk azokat a fogakat, amelyek az első keréken levő egyetlen olajfolttól olajosak lesznek. Ha egy fog olajos lett, akkor természetesen tovább olajozhat másokat (ti. amelyekkel érintkezhet). A megoldáshoz (az olajos fogak száma a két fogaskeréken) fel kell használnunk a két fogszám legnagyobb közös osztóját.

Program:

```
        Be: A,B         [A,B>2 és egészek]
        X:=A : Y:=B
        Ciklus amíg X<>Y
                Ha X>Y akkor X:=X-Y különben Y:=Y-X
        Ciklus vége
        Ki: A/X,B/X
Program vége.
```

A BASIC program:

```
10      CLS : PRINT "FOGASKEREKEK OLAJOS FOGAI" : PRINT
20      INPUT "AZ 1. FOGAI SZÁMA";A : IF A<2 OR A<>INT(A) THEN 20
30      INPUT "A 2. FOGAI SZÁMA";B : IF B<2 OR B<>INT(B) THEN 30
```

```
 40     X=A : Y=B
 50     IF X=Y THEN 80
 60             IF X>Y THEN X=X-Y ELSE Y=Y-X
 70     GOTO 50
 80     REM X=LNKO(A,B)
 90     PRINT "AZ OLAJOS FOGAK SZÁMA:" : PRINT
100     PRINT TAB(10);"1. KERÉK:";A/X
110     PRINT TAB(10);"2. KERÉK:";B/X
120     STOP
```

• • •

GY.6. "átlaghőmérséklet"

A feladat megoldásához szükségünk lesz a hónap napjai számára(N), s az egyes napok átlaghőmérsékletére (A(N)). Ki kell számolnunk minden egyhetes időszakra az átlaghőmérsékletek összegét, s ezek közül a maximálisat kiválasztani. A megoldásunk tetszőleges időtartamú megfigyelés adatainak feldolgozására lesz képes.

Program:

```
        Be: N[N>7 és egész ]
        Ciklus  I=1-től N-ig
                Be: A(I)
        Ciklus vége
        S:=0
        Ciklus I=1-től 7-ig
                S:=S+A(I)
        Ciklus  vége            ; az első hét hőmérsékletösszege
        M:=1 : X:=S             ; az eddigi maximum kezdete, értéke
        Ciklus  I=2-től N-6-ig ; az I-edik nappal kezdődő hét hőmérsékletösszegének
                                 számítása
                S:=S-A(I-1)+A(I+6) ; új összeg
        Ha S>X akkor M:=I : X:=S
        Ciklus  vége
        Ki: M,X/7
Program vége.
```

Írjuk meg a megoldás BASIC programját!

```
 10     CLS : PRINT "EGY HÓNAP LEGMAGASABB ÁTLAGHŐMÉRSÉKLETŰ HETE"
 20     PRINT
 30     INPUT "A HÓNAP NAPJAI SZÁMA";N : IF N<8 OR N<>INT(N) THEN 30
 40     DIM A(N)
 50     FOR I=1 TO N
 60          PRINT I; " . NAP ÁTLAGHŐMÉRSÉKLETE";
 70          INPUT A(I)
 80     NEXT I
 90     REM ELSŐ HÉT HŐMÉRSÉKLETÖSSZEGE
100     S=0
110     FOR I=1 TO 7
120          S=S+A(I)
130     NEXT I
140     M=1 : X=S : REM EDDIGI MAXIMUM
150     FOR I=2 TO N-6 : REM AZ I. NAPON INDULÓ HÉT ADATAI
160          S=S-A(I-1)+A(I+6)
170          IF S>X THEN M=I : X=S : REM ÚJ MAXIMUM
180     NEXT I
190     PRINT : PRINT "A LEGMELEGEBB IDŐSZAK AZ";M;" . NAPON KEZDŐDÖTT"
200     PRINT "AZ ÁTLAGHŐMÉRSÉKLET";X/7;"FOK VOLT."
210     STOP
```

• • •

GY.7. "m-Fibonacci számok"

Az első M szám mindegyike legyen az 1! A következőt ezek összegéből állítjuk elő, s újra csak az utolsó M darabot őrizzük meg (a legrégibbet kell kicserélni az új számra). Az első N+1 ilyen számot fogjuk megadni.

Program:

```
        Be: M,N          [M>0 , N>M és egészek]
        Ciklus  I=0-tól M-1-ig
                F(I):=1:Ki: F(I)
        Ciklus vége
        F:=M                          ; az első M összege
        Ciklus  I=M-től N-ig
                K:=I-INT(I/M)*M       ; a legrégebbi sorszáma
                X:=F : F:=F-F(K)+F    ; új összeg
                F(K):=X : Ki: F(K)
        Ciklus vége
Program vége.
```

Az ebből kapott BASIC program a számokat egymás mögé írja a képernyőn:

```
 10    CLS : PRINT "M-FIBONACCI SZÁMOK" : PRINT
 20    INPUT "M ÉRTÉKE";M : IF M<1 OR M<>INT(M) THEN 20
 30    INPUT "MEDDIG KELL KISZÁMOLNI";N : IF N<M OR N<>INT(N) THEN 30
 40    DIM F(M-1)
 50    FOR I=0 TO M-1
 60        F(I)=1 : PRINT F(I);
 70    NEXT I
 80    F=M : REM AZ ELSŐ M ÖSSZEGE
 90    FOR I=M TO N
100        K=I-INT(I/M)*M : REM A LEGRÉGEBBI SORSZÁMA
110        X=F : F=F-F(K)+F
120        F(K)=X : PRINT F(K);
130    NEXT I
140    STOP
```

• • •

GY.8. "függvénytáblázat"

A megoldásban két dologra kell tekintettel lennünk: egyrészt a képernyőn 4 oszlopba tesszük az eredményeket (x,sinx,cosx,tanx), másrészt pedig a teljes táblázat nem fér el egyetlen képernyőlapon, s ezért gondoskodni kell a lapozásról. Legyen a kezdőérték A, a végérték B, a lépésköz pedig C! A képre kiírható értékes sorok száma 12.

Program:

```
        Be: A,B,C        [A<B, C>0]
        S:=0             ; aktuális sor sorszáma
        N:=12            ; kiírható sorok száma
        Ciklus X=A-tól B-ig C-esével
                Ha S=N akkor S:=0 : Lapozás
                Ha S=0 akkor Fejlécírás
                Ki: X,SIN(X),COS(X),TAN(X)
                S:=S+1
        Ciklus vége
Program vége.
```

Az ebből kapott BASIC programban lapozásra az INKEY$ függvényt használjuk, a képernyő törlését pedig a fejléc kiírásába értjük bele. A cím a 0., a lapozást kérő szöveg a 15. sorba kerül.

```
 10    CLS : PRINT "FÜGGVÉNYTÁBLÁZAT" : PRINT
 20    INPUT "METTŐL, MEDDIG, HÁNYASÁVAL";A,B,C
```

```
 30     IF A>=B OR C<=0 THEN 20
 40     S=0
 50     N=12
 60     FOR X=A TO B STEP C
 70          IF S=N THEN 80 ELSE 100
 80          S=0 : PRINT@15*64,"BILLENTYŰ LENYOMÁSRA LAPOZOK.";
 90          IF INKEY$="" THEN 90
100          IF S=0 THEN CLS : PRINT "X","SIN(X)","COS(X)","TAN(X)" :
                  PRINT
110          PRINT X,SIN(X),COS(X),TAN(X)
120          S=S+1
130     NEXT X
140     STOP
```

• • •

GY.9. "nyugdíj"

Olyan programot kell készíteni, amely egy számsorozaton – az emberek jövedelmei (A(N)) – elvégez egy transzformációt, s ha a kapott eredmény egy érték alá csökkenne, akkor felemeli erre az értékre. Az így kapott eredmény lesz a nyugdíj (B). Felhasználjuk a nyugdíj minimális értékét (Y) és a jövedelem után járó százalékot (X).

Program:

```
        Be:N,X,Y        [N>0 és egész, 0<=X<=100, Y>0]
        X:=X/100        ; hányszorosa lesz a nyugdíj a jövedelemnek
        Ciklus  I=1-től N-ig
                Be: A(I)        [A(I)>0]
        Ciklus vége
        Ciklus  I=1-től N-ig
                B:=INT (X*A(I)/100+0.5)*100         ; 100-asra kerekítés
                Ha B<Y akkor B:=Y
                Ki: B
        Ciklus vége
Program vége.
```

A HT–1080Z BASIC program:

```
 10     CLS  : PRINT "NYUGDÍJ SZÁMÍTÁS" : PRINT
 20     INPUT  "EMBEREK SZÁMA";N : IF N<1 OR N<>INT(N) THEN 20
 30     INPUT  "NYUGDÍJ SZÁZALÉK ÉS ALSÓ HATÁR";X,Y
 40     IF X<0 OR X>100 OR Y<=0 THEN 30 ELSE X=X/100
 50     DIM A(N)
```

```
 60     FOR I=1 TO N
 70          PRINT I;". EMBER JÖVEDELME";
 80          INPUT A(I) : IF A(I)<=0 THEN 70
 90     NEXT I
100     CLS
110     REM NYUGDÍJ SZÁMÍTÁS
120     FOR I=1 TO N
130          B=INT(X*A(I)/100+0.5)*100
140          IF B<Y THEN B=Y
150          PRINT "A(Z)";I;". EMBER NYUGDÍJA=";B
160     NEXT I
170     STOP
```

• • •

GY.10. "legmagasabb ember"

A feladat megoldásához szükségünk van az emberek számára (N), valamint magasságaikra (A(N)) és a szép megoldás kedvéért a neveikre (A$(N)). Tudjuk, hogy azonos magasságúak nincsenek közöttük. Az algoritmusban az M jelöli a legmagasabb ember sorszámát, S pedig a magasságát. A megoldás elve: ha ismerjük I–1 ember közül a legmagasabbat, akkor az I. embert hozzávéve a csoporthoz, a legmagasabb vagy az eddigi legmagasabb lesz, vagy pedig az új ember.

Program:

```
        Be: N  [N>0 és egész]
        Ciklus  I=1-től N-ig
                Be: A$(I),A(I) [A(I)>0]
        Ciklus vége
        M:=0 : S:=0      ; minden magasság nagyobb 0-nál
        Ciklus  I=1-től N-ig
                Ha S<A(I) akkor S:=A(I) : M:=I
        Ciklus vége
        Ki: A$(M),S
Program vége
```

Az ebből kapott BASIC program:

```
 10     CLS  : PRINT "LEGMAGASABB EMBER MEGHATÁROZÁSA"
 20     PRINT
 30     INPUT  "EMBEREK SZÁMA";N : IF N<1 OR N<>INT(N) THEN 30
 40     DIM A$(N),A(N)
 50     FOR I=1 TO N
 60          PRINT I;". EMBER NEVE ÉS MAGASSÁGA";
```

```
 70          INPUT A$(I),A(I) : IF A(I)<=0 THEN 60
 80     NEXT I
 90     REM MAXIMUM KIVÁLASZTÁS
100     M=0 : S=0
110     FOR I=1 TO N
120          IF S<A(I) THEN S=A(I) : M=I
130     NEXT I
140     PRINT "A LEGMAGASABB EMBER: ";A$(M)
150     PRINT "MAGASSÁGA=";S
160     STOP
```

• • •

GY.11. "legmagasabb emberek"

A feladat megoldásához szükségünk van az emberek számára (N), valamint magasságaikra (A(N)) és a szép megoldás kedvéért a neveikre (A$(N)). Vannak közöttük azonos magasságúak, s ha több legmagasabb van, akkor az összeset ki kell írnia a programnak.
Az algoritmusban az S jelöli a legmagasabb ember magasságát. A megoldás elve: ha ismerjük I–1 ember közül a legmagasabb magasságát, akkor az I. embert hozzávéve a csoporthoz, ez a magasság vagy az eddigi lesz, vagy pedig az új emberé. Ezután újra végignézve az összes ember magasságát, kiválaszthatók a legmagasabbak.

Program:

```
        Be: N  [N>0 és egész]
        Ciklus I=1-től N-ig
                Be: A$(I),A(I)   [A(I)>0]
        Ciklus vége
        S:=0          ; minden magasság nagyobb 0-nál
        Ciklus I=1-től N-ig
                Ha S<A(I) akkor S:=A(I)
        Ciklus vége
        Ciklus I=1-től N-ig
                Ha A(I)=S akkor Ki: A$(I),S
        Ciklus vége
Program vége.
```

A BASIC program írásánál az eredményeket egy 2 oszlopos táblázatban fogjuk megadni, amelyben szerepel a legmagasabbak neve és magassága.

```
 10     CLS : PRINT "LEGMAGASABB EMBEREK MEGHATÁROZÁSA"
 20     PRINT
 30     INPUT "EMBEREK SZÁMA";N : IF N<1 OR N<>INT(N) THEN 30
 40     DIM A$(N),A(N)
```

```
 50     FOR I=1 TO N
 60          PRINT I;". EMBER NEVE ÉS MAGASSÁGA";
 70          INPUT A$(I),A(I) : IF A(I)<=0 THEN 60
 80     NEXT I
 90     REM MAXIMUM KIVÁLASZTÁS
100     S=0
110     FOR I=1 TO N
120          IF S<A(I) THEN S=A(I)
130     NEXT I
140     PRINT ,"A LEGMAGASABB EMBEREK "
150     PRINT "NEVE",,"MAGASSÁGA" : PRINT
160     FOR I=1 TO N
170          IF A(I)=S THEN PRINT A$(I);TAB(32);S
190     NEXT I
200     STOP
```

• • •

GY.12. "legalacsonyabb hegycsúcs"

Hegycsúcsnak nevezzük azokat a mérési pontokat, amelyekben a mért magasságérték nagyobb, mint amit a két szomszédjukban mértünk. Legyenek a mért értékek az A(N) vektorban! A megoldás algoritmusa:

```
Program:
     Be: N                    [N>2 és egész]
     Ciklus  I=1-től N-ig
             Be: A(I)
     Ciklus vége
     J:=0 : M:=100 000    ; segédérték a minimumhoz
     Ciklus  I=2-től N-1-ig
             Ha A(I-1)<A(I) és A(I)>A(I+1)
                   akkor Ha A(I)<M akkor M:=A(I) : J:=I
     Ciklus vége
     Ha J>0 akkor Ki: J,M különben Ki: "Nincs hegycsúcs"
Program vége.
```

A BASIC program:

```
 10     CLS  : PRINT "LEGALACSONYABB CSÚCS KIVÁLASZTÁSA" : PRINT
 20     INPUT  "MÉRÉSEK SZÁMA(>2)";N : IF N<3 OR N<>INT(N) THEN 20
 30     DIM A(N)
 40     FOR I=1 TO N
 50          PRINT I;".MÉRÉS"; : INPUT A(I)
```

```
 60     NEXT I
 70     J=0 : M=100000
 80     FOR I=2 TO N-1
 90     IF A(I-1)<A(I) AND A(I)>A(I+1) THEN
                IF A(I)<M THEN M=A(I) : J=I
100     NEXT I
110     IF J>0 THEN PRINT "A LEGALACSONYABB CSÚCS A(Z)";J;
                ". MÉRÉSNÉL VOLT, MAGASSÁGA:";M
                ELSE PRINT "NINCS HEGYCSÚCS"
120     STOP
```

• • •

GY.13. "legközelebbi pont"

A megoldáshoz be kell olvasnunk a pontok számát (N), majd pedig a pontok koordinátáit, amelyből rögtön origótól való távolságot számolunk (A(N)). Ezek után ki kell választanunk közülük az origóhoz legközelebbit. A legegyszerűbb megoldáson annyival túlléphetünk, hogy ha egy pont pontosan az origóba esik, akkor biztosan az van hozzá a legközelebb, így további hasonlításokat már nem kell végeznünk.

Program:

```
Be: N  [N>0 és egész]
J:=0     ; értéke egy sorszám lesz, ha az ilyen sorszámú pont távolsága 0
Ciklus I=1-től N-ig
        Be: X,Y
        A(I):=SQR(X↑2+Y↑2)
        Ha A(I)=0 akkor J:=I
Ciklus vége

. . .
```

Amennyiben volt 0 távolságú, akkor ki kell írni a sorszámát és a távolságát, ha pedig nem volt, akkor meg kell keresni a legközelebbit. Tegyük fel, hogy van olyan pont, amely egy maximálisnak tekintett értéknél (1E+20) közelebb van az origóhoz: ezt az értéket a minimumkiválasztásban segédértéknek fogjuk használni.

. . .

```
        Ha J>0 akkor X:=0
                 különben X:=1E+20        ; X=maximális távolság, J=0
                          Ciklus I=1-től N-ig
                                   Ha X>A(I) akkor X:=A(I) : J:=I
                          Ciklus vége
        Elágazás vége
        Ki: J,X
Program vége.
```

A teljes program:

```
 10     CLS  : PRINT "AZ ORIGÓHOZ LEGKÖZELEBBI PONT MEGHATÁROZÁSA"
 20     PRINT
 30     INPUT  "HÁNY PONTOT VIZSGÁLUNK";N
 40     IF N<1 OR N<>INT(N) THEN 30
 50     DIM A(N)
 60     FOR I=1 TO N
 70          PRINT "A(Z)";I;". PONT KOORDINÁTÁI";
 80          INPUT X,Y
 90          A(I)=SQR(X↑2+Y↑2)
100          IF A(I)=0 THEN J=I
110     NEXT I
120     REM EREDMÉNY SZÁMÍTÁS
130     IF J>0 THEN X=0 : GOTO 190
140     REM MINIMUM KERESÉS
150     X=1E+20 : REM MAXIMÁLIS ÉRTÉK
160     FOR I=1 TO N
170          IF X>A(I) THEN X=A(I) : J=I
180     NEXT I
190     PRINT "AZ ORIGÓHOZ LEGKÖZELEBBI PONT SORSZÁMA:";J
200     PRINT "TÁVOLSÁGA:";X
210     STOP
```

• • •

GY.14. "hegycsúcsok"

A hegycsúcs a térképen [A(N,M)] olyan pont, amely minden szomszédjánál (8 szomszédja van) magasabb. A térkép szélein (első, utolsó sor, illetve oszlop) levő pontokról nem tudjuk megállapítani, hogy hegycsúcsok-e. Így programunk nagyon egyszerűen alakul:

Program:

```
        Be: N,M        [N,M>2 és egész]
        Ciklus I=1-től N-ig
            Ciklus J=1-től M-ig
                Be: A(I,J)
            Ciklus vége
        Ciklus vége
        Ciklus I=2-től N-1-ig
            Ciklus J=2-től M-1-ig
                Ha A(I,J) magasabb szomszédainál akkor Ki: I,J
            Ciklus vége
        Ciklus vége
Program vége.
```

A BASIC program:

```
 10     CLS : PRINT "HEGYCSÚCS KERESÉS" : PRINT
 20     INPUT "KÉPMÉRET";N,M
 30     IF N<3 OR M<3 OR N<>INT(N) OR M<>INT(M) THEN 20
 40     FOR I=1 TO N : PRINT I;". SOR ADATAI:"
 50         FOR J=1 TO M
 60             PRINT J;". ADAT"; : INPUT A(I,J)
 70         NEXT J
 80     NEXT I
 90     CLS
100     FOR I=2 TO N-1 : FOR J=2 TO M-1
110             S=0
120             FOR K=I-1 TO I+1 : FOR L=J-1 TO J+1
130                     IF I=K AND J=L THEN 150
140                     IF A(I,J)<=A(K,L) THEN 170
150             NEXT L : NEXT K
160             PRINT I;J;"HEGYCSÚCS"
170     NEXT J : NEXT I
180     STOP
```

• • •

GY.15. "gyökkeresés"

A gyökkeresés módszere a következő lesz: a megoldáshoz a felhasználónak meg kell adnia egy intervallumot, ahol a függvény gyökhelyét keressük. Az intervallum két végpontjában a függvénynek különböző előjelűnek kell lennie. Vesszük az intervallum felezőpontját, s az itt kiszámított függvényérték előjelétől függően döntünk, hogy a gyököt melyik intervallumfélben keressük tovább. Az eljárást befejezzük, ha a vizsgált intervallum rövidebb egy, a felhasználó által megadott korlátnál, vagy esetleg pontosan megtaláltunk egy gyökhelyet.

Program:

```
        Be: A,B,E        [F(A)*F(B)<=0, E>0]
        Ciklus
                K:=(A+B)/2
                Ha F(K)*F(A)<0 akkor B:=K különben A:=K
        amíg F(K)<>0 és B-A>E
        Ciklus vége
        Ki: K,F(K)
Program vége.
```

A BASIC programban a függvényt egy szubrutin számítja ki. A szubrutin paramétere X, eredménye pedig az Y változóban lesz. Gondoskodnunk kell a függvényértékek tárolásáról is (nem akarunk feleslegesen többször kiszámolni egy függvényértéket).

```
  10    CLS : PRINT "GYÖKKERESÉS" : PRINT
  20    INPUT "INTERVALLUM VÉGPONTJAI";A,B
  30    X=A : GOSUB 1000 : Y1=Y : REM Y1=F(A)
  40    X=B : GOSUB 1000 : Y2=Y : REM Y2=F(B)
  50    IF A>B OR Y1*Y2>0 THEN 20
  60    INPUT "PONTOSSÁG";E : IF E<=0 THEN 60
  70    REM GYÖKKERESÉS
  80          K=(A+B)/2
  90          X=K : GOSUB 1000 : REM Y=F(K)
 100          IF Y*Y1<0 THEN B=K : Y2=Y ELSE A=K : Y1=Y
 110    IF Y<>0 AND B-A>E THEN 70
 120    PRINT "A ZÉRUSHELY: ";K,"FÜGGVÉNYÉRTÉK:";Y
 130    STOP
1000    REM Y=F(X)
1010          Y=SIN(2*X+2)-EXP(-X)
1020    RETURN
```

• • •

GY.16. "lövészverseny"

A megoldáshoz fel kell használnunk a lövések eredményeit (A(20)), ezeket sorba kell rendeznünk, majd kiszámítani a legjobb 10 összegét, azaz a versenyző eredményét (E).

Program:

```
N:=20 : L:=10   ; 20-ból a legjobb 10
Ciklus I=1-től N-ig
        Be: A(I)        [A(I)=0,1,..,10 lehet]
Ciklus vége
Ciklus I=1-től N–1-ig   ; rendezés maximum kiválasztással
        Ciklus J=I+1-től N-ig
                Ha A(J)>A(I) akkor Csere (A(I),A(J))
        Ciklus vége
Ciklus vége
E:=0
Ciklus I=1-től L-ig
        E:=E+A(I)
Ciklus vége
Ki: L,E
Program vége.
```

A BASIC programban az A vektort egész típusúként használjuk, kizárva ezzel a tört értékű lövéseket.

```
 10   CLS
 20   N=20 : L=10
 30   PRINT N;"LÖVÉSBŐL A LEGJOBB";L;"EREDMÉNYÉNEK KISZÁMÍTÁSA"
 40   PRINT
 50   DIM A%(N)
 60   FOR I=1 TO N
 70        PRINT "A(Z)";I;". LÖVÉS EREDMÉNYE";
 80        INPUT A%(I) : IF A%(I)<0 OR A%(I)>10 THEN 70
 90   NEXT I
100   REM RENDEZÉS
110   FOR I=1 TO N–1
120        FOR J=I+1 TO N
130             IF A%(J)>A%(I) THEN X%=A%(I) : A%(I)=A%(J) : A%(J)=X%
140   NEXT J : NEXT I
150   REM EREDMÉNY SZÁMÍTÁS
160   E=0
170   FOR I=1 TO L
180        E=E+A%(I)
190   NEXT I
200   PRINT "A LEGJOBB";L;"LÖVÉS EREDMÉNYE=";E
210   STOP
```

• • •

5. Feladatmegoldások kezdők számára

GY.17. "lépegetős játék"

Andrásnak úgy kell lépnie a játékban, hogy az utolsó lépést ő tehesse meg. Jelölje A a lépések minimális számát, B pedig a maximálisat! Ekkor Andrásnak olyan helyekre előnyös lépnie, amelyek Bélától mért távolsága A+B többszöröse. A megoldásban V jelöli az indulási, X pedig az aktuális távolságot, L értéke 1, ha András következik és 2, ha Béla.

Program:

```
V:=100 : A:=2 : B:=9
X:=V : L:=1          ; András következik
Ciklus amíg X>0
    Ki: X            ; aktuális távolság
    Ha L=1 akkor
        X:=X-A       ; minimális lépés
        Y:=X-(A+B)*INT(X/(A+B))
        Ha Y<=B-A akkor X:=X-Y   ; előnyös helyre lépés
        L:=2         ; Béla következik
    különben
        Be: Z [Z>=A és <=B és egész]
        X:=X-Z
        L:=1         ; András következik
    Elágazás vége
Ciklus vége
Ha L=1 akkor Ki: "Béla nyert" különben Ki: "András nyert"
Program vége.
```

A BASIC program az aktuális távolságot a képernyő 3. sorába írja, Béla lépését pedig az 5.-be:

```
 10   CLS  : PRINT "LÉPEGETŐS JÁTÉK" : PRINT
 20   V=100 : A=2 : B=9
 30   X=V : L=1 : REM ANDRÁS KÖVETKEZIK
 40   IF X<=0 THEN 170
 50        PRINT @3*64,"AKT. TÁVOLSÁG=";X
 60        IF L=1 THEN 70 ELSE 120
 70             X=X-A : REM MINIMÁLIS LÉPÉS
 80             Y=X-(A+B)*INT(X/(A+B))
 90             IF Y<=B-A THEN X=X-Y : REM ELŐNYÖS HELYRE LÉPÉS
100             L=2 : REM BÉLA KÖVETKEZIK
110        GOTO 160
120             PRINT @5*64,"BÉLA LÉPÉSE(2-9)"; : INPUT Z
130             IF Z<A OR Z>B OR Z<>INT(Z) THEN 120
140             X=X-Z
150             L=1 : REM ANDRÁS KÖVETKEZIK
160   GOTO 40
```

```
170     IF L=1 THEN PRINT "BÉLA NYERT" ELSE PRINT "ANDRÁS NYERT"
180     STOP
```

• • •

GY.18. "bábudöntögetés"

N bábut állítunk fel egy kör mentén. A bábukat egy vektorban azonosítjuk (B(N)), a megfelelő érték 1, ha a bábu még áll és 0, ha már nem. Minden K-adik álló bábut kell feldöntenünk, egészen addig, amíg van még álló bábu. Ezt úgy tehetjük meg, hogy a döntést N-szer ismételjük.

Program:

```
        Be: N,K         [N,K>0 és egészek]
        Ciklus I=1-től N-ig
                B(I):=1
        Ciklus vége
        A:=1                        ; aktuális bábu sorszáma
        B(A):=0 : Ki: A
        Ciklus M=2-től N-ig.
                K-adik keresése     ; A lesz a sorszáma
                B(A):=0 : Ki: A
        Ciklus vége
Program vége.
```

A K-adik keresését úgy végezzük, hogy K-szor megkeressük a következőt. A következő keresésében arra kell ügyelnünk, ha túl mennénk a vektor végén, akkor kezdjük elölről a keresést.

K-adik keresése(A):

```
        Ciklus I=1-től K-ig
                Ciklus
                    A:=A+1
                    Ha A>N akkor A:=1
                amíg B(A)=0
                Ciklus vége
        Ciklus vége
Eljárás vége.
```

A BASIC program:

```
10      CLS : PRINT "BÁBUDÖNTÖGETÉS" : PRINT
20      INPUT "BÁBUK SZÁMA";N : IF N<1 OR N<>INT(N) THEN 20
30      INPUT "DÖNTÉS TÁVOLSÁG";K : IF K<1 OR K<>INT(K) THEN 30
40      CLS : PRINT "FELDÖNTÉSI SORREND" : PRINT
50      DIM B(N)
60      FOR I=1 TO N : B(I)=1 : NEXT I
```

```
 70    A=1 : REM AKTUÁLIS BÁBU
 80    B(A)=0 : PRINT A;
 90    FOR M=2 TO N
100          GOSUB 500 : REM K-ADIK KERESÉSE
110          B(A)=0 : PRINT A;
120    NEXT M
130    STOP
500    REM K-ADIK KERESÉSE
510    FOR I=1 TO K
520          REM KÖVETKEZŐ KERESÉSE
530               A=A+1
540               IF A>N THEN A=1
550          IF B(A)=0 THEN 520
560    NEXT I
570    RETURN
```

• • •

GY.19. "kör rajzolás"

Akármilyen módszerrel is fogunk hozzá, elég egy origó középpontú kör rajzát elkészíteni, majd ezt eltolni a képernyő megfelelő pontjára. Másik hasznosítható felismerés az, hogy elég egy negyed körívet rajzolni, s ebből egyszerű transzformációkkal előállítható a teljes kör. Ugyanis, ha az (x,y) pont a körvonal egy pontja, akkor a (–x,y), az (x,–y) és a (–x,–y) is az. Így az 1. síknegyedbe eső pontokat kell csak megrajzolnunk. Amennyiben a kör középpontját a képernyő A. oszlopának B. sorába helyezzük, akkor az (x,y) pont rajzolása az (x+A,y+B) képpont kigyújtását jelenti. A megoldásunk elve: ha egy pontot kirajzoltunk, akkor a következőt próbáljuk tőle egy egységgel felfelé elhelyezni. Ha ezt nem lehet, akkor balra felfelé, s ha ezt sem, akkor balra. Egy pont akkor helyezhető el, ha az "ideális" körvonalon még belül lenne és akkor nem, ha kívülre kerülne. A rajzolás az (R,0) pontban kezdődik és a negyedkörív végéig tart.

Program körrajzolásra:

```
Be: R          [R>0, a kör sugara]
Y:=0 : X:=R          ; indulási pont
B:=20 : A:=40        ; a középpont koordinátái
Ciklus
      Rajzolás(X,Y)
      Rajzolás(–X,Y)
      Rajzolás(X,–Y)
      Rajzolás(–X,–Y)
      Ha (Y+1)²+X² < R² akkor Y:=Y+1
            különben Ha (Y+1)²+(X–1)² < R² akkor Y:=Y+1 : X:=X–1
                  különben X:=X–1
amíg X >=0
Ciklus vége
Program vége.
```

A BASIC programot így kapjuk

```
 10    CLS : PRINT "KÖR RAJZOLÁS" : PRINT
 20    INPUT "A KÖR SUGARA";R : IF R<=0 THEN 20
 30    CLS
 40    Y=0 : X=R
 50    B=20 : A=40 : REM A KÖZÉPPONT KOORDINÁTÁI
 60    REM RAJZOLÁS
 70          SET(X+A,Y+B)
 80          SET(-X+A,Y+B)
 90          SET(X+A,-Y+B)
100          SET(-X+A,-Y+B)
110          IF (Y+1)↑2+X↑2<R↑2 THEN Y=Y+1 : GOTO 130
120          IF (Y+1)↑2+(X-1)↑2<R↑2 THEN Y=Y+1 : X=X-1 ELSE X=X-1
130    IF X>=0 THEN 60
140    STOP
```

• • •

GY.20. "szakasz rajzolás"

Rajzolásnál általában arra kell figyelemmel lennünk, hogy a szokásos koordináta-rendszer nem egyezik meg a képernyő beosztásával. Így gondoskodnunk kell a kép képernyőre transzformálásáról. Legyen a képernyőn az origó az (A,B) pontban! Jelölje X1,Y1 a szakasz egyik végpontját, X2,Y2 pedig a másikat. A rajzoló algoritmusnak arra kell ügyelnie, hogy a szakaszt lehetőleg folytonos vonallal rajzolja, ill. ha a képen kívülre kellene rajzolnia, akkor azt ne engedje!

Program

```
        Be: X1,Y1,X2,Y2
        H1:=X2-X1 : H2:=Y2-Y1                  ; koordinátánkénti különbség
        I1:=ABS(H1) : I2:=ABS(H2)              ; koordinátánkénti távolság
        Ha I1>I2 akkor H:=I1 különben H:=I2    ; távolságok maximuma
        X:=X1 : Y:=Y1                          ; kezdőpont
        Ha H=0 akkor Rajzolás(X,Y)
             különben S1:=H1/H : S2:= H2/H     ; rajzolási lépésközök
                      Ciklus K=1-től H+1-ig    ; rajzolandó pontok száma
                           Rajzolás(X,Y)
                           X:=X+S1 : Y:=Y+S2   ; következő pontok
                      Ciklus vége
        Elágazás vége
Program vége
```

A BASIC program az origót a (60,20) pontba állítja. A Rajzolás(X,Y) eljárásnak megfelelő utasítás pedig (ellenőrzések után) a SET(X+A,B–Y).

```
 10    CLS : PRINT „EGYENES RAJZOLÁS" : PRINT
 20    A=60 : B=20
 30    INPUT "1. VÉGPONT KOORDINÁTÁI";X1,Y1
 40    INPUT "2. VÉGPONT KOORDINÁTÁI";X2,Y2 : CLS
 50    H1=X2–X1 : H2=Y2–Y1 : REM KOORDINÁTÁNKÉNTI KÜLÖNBSÉG
 60    I1=ABS(H1) : I2=ABS(H2) : REM KOORDINÁTÁNKÉNTI TÁVOLSÁG
 70    IF I1>I2 THEN H=I1 ELSE H=I2 : REM TÁVOLSÁGOK MAXIMUMA
 80    X=X1 : Y=Y1 : REM KEZDŐPONT
 90    IF H=0 THEN GOSUB 200 : GOTO 150
100    S1=H1/H : S2=H2/H : REM RAJZOLÁSI LÉPÉSKÖZÖK
110    FOR K=1 TO H+1
120         GOSUB 200 : REM RAJZOLÁS (X,Y)
130         X=X+S1 : Y=Y+S2 : REM KÖVETKEZŐ PONT
140    NEXT K
150    STOP
200    REM RAJZOLÁS(X,Y)
210    IF A+X<0 OR A+X>127 OR B–Y<0 OR B–Y>47 THEN RETURN
220    SET(A+X,B–Y)
230    RETURN
```

GY.21. "síkidom rajzolás"

A síkidomot azokkal a szakaszokkal adjuk meg, amelyek határolják. Egy másik feladatban szakaszt kellett rajzolnunk, most pedig szakaszok sorozatát. Így felhasználva a szakaszrajzolást, egyszerű megoldást kapunk. Legyen N a szakaszok száma, X(N*2,2) tartalmazza a szakaszok kezdő- és végpontjainak koordinátáit (a páratlan első indexűek a kezdőpontok, a párosak pedig a végpontok).

Program:

```
        Be: N                          ; szakaszok száma
        Ciklus I=1-től N-ig
             Be: X(I*2–1,1),X(I*2–1,2)  ; kezdőpont
             Be: X(I*2,1),X(I*2,2)      ; végpont
        Ciklus vége
          Ciklus I=1-től N-ig
             Az I. szakasz rajzolása
          Ciklus vége
Program vége.
```

A szakaszrajzolás szinte teljesen megegyezik a korábban közölt megoldással.

Az I. szakasz rajzolása:

```
        H1:=X(I*2,1)-X(I*2-1,1) : H2:=X(I*2,2)-X(I*2-1,2)
        I1:=ABS(H1) : I2:=ABS(H2)                ; koordinátánkénti távolság
        Ha I1>I2 akkor H:=I1 különben H:=I2      ; távolságok maximuma
        X:=X(I*2-1,1) : Y:=X(I*2-1,2)            ; kezdőpont
        Ha H=0 akkor Rajzolás(X,Y)
            különben S1:=H1/H : S2:=H2/H         ; rajzolási lépésközök
                     Ciklus K=1-től H+1-ig       ; rajzolandó pontok száma
                        Rajzolás(X,Y)
                        X:=X+S1 : Y:=Y+S2        ; következő pont
                     Ciklus vége
        Elágazás vége
Eljárás vége.
```

A BASIC program az origót a (60,20) pontba állítja. A Rajzolás(X,Y) eljárásnak megfelelő utasítás pedig (ellenőrzések után) a SET(X+A, B–Y).

```
 10   CLS : PRINT "SÍKIDOM RAJZOLÁS" : PRINT
 20   A=60 : B=20
 30   INPUT "A SÍKIDOMOT HATÁROLÓ SZAKASZOK SZÁMA";N
 40   IF N<1 OR N<>INT(N) THEN 30
 50   DIM X(N*2,2)
 60   FOR I=1 TO N
 70        PRINT I;". KEZDŐPONT KOORDINÁTÁI";
 80        INPUT X(I*2-1,1),X(I*2-1,2)
 90        PRINT I;". VÉGPONT KOORDINÁTÁI";
100        INPUT X(I*2,1),X(I*2,2)
110   NEXT I
120   CLS
130   FOR I=1 TO N
140        GOSUB 200 : REM AZ I. SZAKASZ RAJZOLÁSA
150   NEXT I
160   STOP
200   H1=X(I*2,1)-X(I*2-1,1) : H2=X(I*2,2)-X(I*2-1,2)
210   I1=ABS(H1) : I2=ABS(H2) : REM KOORDINÁTÁNKÉNTI TÁVOLSÁG
220   IF I1>I2 THEN H=I1 ELSE H=I2 : REM TÁVOLSÁGOK MAXIMUMA
230   X=X(I*2-1,1) : Y=X(I*2-1,2) : REM KEZDŐPONT
240   IF H=0 THEN GOSUB 400 : GOTO 300
250   S1=H1/H : S2= H2/H : REM RAJZOLÁSI LÉPÉSKÖZÖK
260   FOR K=1 TO H+1
270        GOSUB 400 : REM RAJZOLÁS (X,Y)
280        X=X+S1 : Y=Y+S2 : REM KÖVETKEZŐ PONT
```

```
290     NEXT K
300     RETURN
400     REM RAJZOLÁS(X,Y)
410     IF A+X<0 OR A+X>127 OR B-Y<0 OR B-Y>47 THEN RETURN
420     SET(A+X,B-Y)
430     RETURN
```

• • •

GY.22. "függvényrajzolás"

A megoldást úgy fogjuk elkészíteni, hogy a függvény ábráját a képernyő méretének megfelelő méretre transzformáljuk. Ehhez szükségünk van a függvény értelmezési tartományára ([A,B] intervallum), valamint a függvény minimális és maximális értékére (vagy helyettük alsó és felső korlátjára) – MAX,MIN. Ismerni kell a képernyő szélső oszlopai, ill. sorai közötti távolságot (N,M). Ezekből az adatokból meg kell határoznunk a szorzótényezőket, a rajzolási lépésközt, illetve a képernyő más koordináta-rendszere miatti átalakítás szabályait. A transzformáció miatt nem engedjük meg az A=B, MAX=MIN eseteket.
Így a rajzoló eljárás:

Függvényrajzolás:

```
        KX:=N/(B-A) : KY=M/(MAX-MIN)        ; szorzótényezők
        Ciklus X=A-tól B-ig (B-A)/N-esével
            Y:=F(X)
            S:=M-(Y-MIN)*KY                 ; sorkoordináta
            O:=(X-A)*KX                     ; oszlopkoordináta
            Rajzolás(O,S)
        Ciklus vége
Eljárás vége.
```

A BASIC programban a függvényt az 1000-es sorszámú soron definiáljuk egy szubrutinban, rajzolásnál kerekíteni kell a kiszámított értékeket.

```
 10     CLS : PRINT "FÜGGVÉNY RAJZOLÁS" : PRINT
 20     INPUT "ÉRT. TARTOMÁNY";A,B : IF A>=B THEN 20
 30     INPUT "ALSÓ, FELSŐ KORLÁT";MIN,MAX : IF MIN>=MAX THEN 30
 40     CLS
 50     N=127 : M=47 : REM HOSSZÚSÁG ADATOK
 60     KX=N/(B-A) : KY=M/(MAX-MIN)
 70     FOR X=A TO B STEP (B-A)/N
 80         GOSUB 1000 : REM Y=F(X)
 90         S=M-(Y-MIN)*KY : S=INT(S+.5)
100         O=(X-A)*KX : O=INT(O+.5)
```

```
 120        SET(O,S)
 130   NEXT X
 140   STOP
1000   REM Y=F(X)
1010        Y=SIN(X*X)+X
1020   RETURN
```

• • •

GY.23. "milyen napra esik?"

Egy hónap nevét és egy nap sorszámát (N) fogja kapni a program, így ismernünk kell a hónapok neveit [N$(12)] és napszámait [S(12)]. Ki kell számolnunk, hogy a kérezett nap az év hányadik napja (S), ezt el kell osztanunk héttel, s az osztási maradék fogja megadni, hogy a hét melyik napjára [H$(7)] esik ez a dátum. A program kiszámolja a kért hónap sorszámát (H), ekkor a megoldás lényegi része:

```
•••
S:=0                    ;  nap-sorszám
Ciklus I=1-től H-1-ig
      S:=S+S(I)
Ciklus vége             ;  előző hónapok napjai száma
S:=S+N-1
S:=S-INT (S/7)*7        ;  osztási maradék
Ki:H$(S+1)              ;  a 0 indexűt nem használjuk
•••
```

A BASIC programban a szükséges információkat DATA utasításokban tároljuk, s az egyszerűség kedvéért feltesszük, hogy január elseje hétfő volt. A nap sorszámának kiszámításába pedig belevesszük az ellenőrzést is.

```
 10   CLS : PRINT "A HÉT MELYIK NAPJÁRA ESIK" : PRINT
 20   DATA HÉTFŐ,KEDD,SZERDA,CSÜTÖRTÖK,PÉNTEK,SZOMBAT,VASÁRNAP
 30   DATA JANUÁR,31,FEBRUÁR,28,MÁRCIUS,31,ÁPRILIS,30
 40   DATA MÁJUS,31,JÚNIUS,30,JÚLIUS,31,AUGUSZTUS,31
 50   DATA SZEPTEMBER,30,OKTÓBER,31,NOVEMBER,30,DECEMBER,31
 60   DIM H$(7),N$(12),S(12)
 70   FOR I=1 TO 7 : READ H$(I) : NEXT I
 80   FOR I=1 TO 12 : READ N$(I),S(I) : NEXT I
 90   S=0
100   INPUT "HÓNAP NÉV, NAP SORSZÁM";N$,N
120   FOR I=1 TO 12
130        IF N$=N$(I) THEN 160 ELSE S=S+S(I)
140   NEXT I
150   PRINT "NINCS ILYEN HÓNAP" : GOTO 100
```

```
160     IF N>S(I) OR N<1 THEN PRINT "NINCS ENNYI NAPJA A HÓNAPNAK" :
                GOTO 100
170     S=S+N-1
180     S=S-INT(S/7)*7 : REM MARADÉK
190     PRINT "EZ A NAP ";H$(S+1)
200     STOP
```

• • •

GY.24. "erősen összetett számok"

A feladat megoldásához meg kell határoznunk N-ig az összes természetes szám osztói számát, s ha az aktuális szám osztói száma több, mint az összes korábbié, akkor ki kell írni ezt a számot. Az első két ilyen szám az 1 és a 2, így N értéke legalább 3 lesz. A megoldásban kihasználjuk, ha egy osztót megtaláltunk, akkor a vele való osztás hányadosa is osztó, így egyszerre kettőt is számolhatunk (kivétel a szám négyzetgyöke).

Program:

```
        Be: N           [N>3 és egész]
        Ki:1,2
        M:=2                    ;  a 2 osztói száma
        Ciklus I=3-tól N-ig
            S:=2                ;  I-nek legalább 2 osztója van
            J:=2                ;  első osztója a 2 lehet
            Ciklus amíg J<SQR(I)

                Ha J osztója I-nek akkor S:=S+2
                J:=J+1
            Ciklus vége
            Ha J*J=I akkor S:=S+1       ;  a négyzetgyök osztó?
            Ha S>M akkor M:=S : Ki: I   ;  új ilyen szám
        Ciklus vége
Program vége.
```

A BASIC programban az erősen összetett számokat egymás mögé fogjuk kiírni.

```
10      CLS  : PRINT "ERŐSEN ÖSSZETETT SZÁMOK" : PRINT
20      INPUT  "MEDDIG ADJAM MEG ŐKET";N : IF N<3 OR N<>INT(N) THEN 20
30      PRINT : PRINT "ERŐSEN ÖSSZETETT:";1;2;
40      M=2 : REM A 2 OSZTÓI SZÁMA
50      FOR I=3 TO N
60          S=2 : REM MINDEN SZÁMNAK LEGALÁBB 2 OSZTÓJA VAN
```

```
 70          J=2 : REM ELSŐ OSZTÓJA A 2 LEHET
 80          IF J<SQR(I) THEN 90 ELSE 120
 90               IF INT (I/J)=I/J THEN S=S+2
100               J=J+1
110          GOTO 80
120          IF J*J=I THEN S=S+1
130          IF S>M THEN M=S : PRINT I;
140     NEXT I
150     STOP
```

• • •

GY.25. "tökéletes számok"

Meg kell határoznunk minden egyes szám osztói összegét. Elég megkeresni az osztókat a szám négyzetgyökéig, mert ha egy szám osztó, akkor a vele való osztás hányadosa is osztó, s így rögtön mindkettőt hozzáadhatjuk az összeghez (egyetlen kivétel a szám négyzetgyöke, amennyiben az egész). Az egyet beszámítjuk az osztók közé, magát a számot pedig nem. Mindezt meg kell tennünk az első N számra, pontosabban az 1 számra nincs értelme a feladatnak.

Program:

```
        Be: N        [N>1 és egész]
        Ciklus I=2-től N-ig
                S:=1        ;  az 1 biztosan osztó
                J:=2        ;  először a 2-vel osztást nézzük
                Ciklus amíg J<SQR(I)
                        Ha J osztója I-nek akkor S:=S+J+I/J
                        J:=J+1
                Ciklus vége
                Ha J*J=I akkor S:=S+J
                Ha S=I akkor Ki: I
        Ciklus vége
Program vége.
```

A Basic programban a tökéletes számokat egymás mögé fogjuk kiírni.

```
 10      CLS  : PRINT "TÖKÉLETES SZÁMOK" : PRINT
 20     INPUT  "MEDDIG ADJAM ŐKET";N : IF N<2 OR N<>INT(N) THEN 20
 30     PRINT : PRINT "TÖKÉLETES:";
 40     FOR I=2 TO N
```

```
 50          S=1 : REM AZ 1 MINDEN SZÁMNAK OSZTÓJA
 60          J=2 : REM ELSŐ OSZTÓJA 2 LEHET
 70          IF J<SQR(I) THEN 80 ELSE 110
 80               IF INT (I/J)=I/J THEN S=S+J+I/J
 90               J=J+1
100          GOTO 70
110          IF J*J=I THEN S=S+J
120          IF S=I THEN PRINT I;
130     NEXT I
140     STOP
```

• • •

GY.26. "barátságos számok"

Meg kell határoznunk a barátságos számpárokat N-ig. Olyan megoldást készítünk, amely a feladatot időben optimális módon oldja meg, s a megoldáshoz nagy tárterületet használ (A(N) vektor). Kiszámoljuk az összes szám osztói összegét, s ha ez az összeg az aktuális számnál nagyobb, akkor a tárolóterületen tároljuk, ha pedig nem, akkor megnézzük, hogy az ilyen sorszámú tárolt szám osztói összege (amit már ismerünk) azonos-e az aktuális számmal (ha azonos, akkor találtunk egy barátságos számpárt).

Program:

```
        Be: N        [N>1 és egész]
        Ciklus I=2-től N-ig
                S:=1        ;  az 1 minden számnak osztója
                J:=2
                Ciklus amíg J<SQR(I)
                        Ha INT(I/J)=I/J akkor S:=S+J+I/J
                        J:=J+1
                Ciklus vége
                Ha J*J=I akkor S:=S+J
                A(I):=S
                Ha S<=I és I=A(S) akkor Ki: S,I
        Ciklus vége
Program vége.
```

A BASIC program:

```
 10     CLS  : PRINT "BARÁTSÁGOS SZÁMOK" : PRINT
 20     INPUT "MEDDIG SZÁMOLJAM ŐKET";N
 30     IF N<2 OR N<>INT(N) THEN 20
 40     DIM A(N)
 50     FOR I=2 TO N
 60          S=1 : REM AZ 1 MINDEN SZÁMNAK OSZTÓJA
```

```
 70          J=2 : REM AZ ELSŐ VIZSGÁLANDÓ OSZTÓ
 80          IF J>=SQR(I) THEN 120
 90               IF INT (I/J)>=I/J THEN S=S+J+I/J
100               J=J+1
110          GOTO 80
120          IF J*J=I THEN S=S+J
130          A(I)=S
140          IF S<=I AND I=A(S) THEN PRINT "BARÁTSÁGOSAK:";S;I
150     NEXT I
160     STOP
```

• • •

GY.27. "átlagok"

Jelölje N a tanulók, M a tantárgyak számát! X(N,M) mátrixban lesznek a tanulmányi eredmények. T1(N) az egyes tanulók osztályzatai száma, T(N) pedig az osztályzatai összege; S1(M) az egyes tantárgyakból adott osztályzatok száma, S(M) pedig az összege.

Program:

```
        Osztályzatok beolvasása
        Táblázatok törlése
        Ciklus I=1-től N-ig
              Ciklus  J=1-től M-ig
                    Ha X(I,J)>0 akkor T1(I):=T1(I)+1 : T(I):=T(I)+X(I,J)
                                      S1(J):=S1(J)+1 : S(J):=S(J)+X(I,J)
              Ciklus vége
        Ciklus vége
        Eredmények kiírása
Program vége.
```

Az eredményeket csak abban az esetben írjuk ki, ha a tanuló kapott osztályzatot, ill. ha az adott tárgyat tanulta legalább egy tanuló.

Eredmények kiírása:

```
        Ciklus I=1-től N-ig
              Ha T1(I)>0 akkor Ki:I,T(I)/T1(I)
        Ciklus vége
        Ciklus I=1-től M-ig
              Ha S1(I)>0 akkor Ki:I, S(I)/S1(I)
        Ciklus vége
Eljárás vége.
```

5. Feladatmegoldások kezdők számára

A BASIC programban az eredmények kiírását és a ki nem fejtett programrészeket szubrutinként írjuk meg. Minden képernyőlapra maximum 12 sort írunk ki, de kevesebb is kerülhet, ha egy-egy tanuló/tantárgy hiányzik.

```
10    CLS : PRINT "OSZTÁLYZAT – STATISZTIKA" : PRINT
20    GOSUB 1000 : REM OSZTÁLYZATOK BEOLVASÁSA
30    GOSUB 2000 : REM TÁBLÁZATOK TÖRLÉSE
40    FOR I=1 TO N
50          FOR J=1 TO M
60                IF X(I,J)=0 THEN 90
70                T1(I)=T1(I)+1 : T(I)=T(I)+X(I,J)
80                S1(J)=S1(J)+1 : S(J)=S(J)+X(I,J)
90    NEXT J : NEXT I
100   GOSUB 3000 : REM EREDMÉNYEK KIÍRÁSA
110   STOP
1000  REM OSZTÁLYZATOK BEOLVASÁSA
1010  INPUT "TANULÓK, TANTÁRGYAK SZÁMA";N,M
1020  IF N<1 OR M<1 OR N<>INT(N) OR M<>INT(M) THEN 1010
1030  DIM X(N,M),T(N),T1(N),S(M),S1(M)
1040  FOR I=1 TO N : PRINT I;". TANULÓ OSZTÁLYZATAI:" : PRINT
1050  FOR J=1 TO M
1060        PRINT J;". OSZTÁLYZAT"; : INPUT X(I,J)
1070        IF X(I,J)<1 OR X(I,J)>5 OR X(I,J)<>INT(X(I,J)) THEN 1060
1080  NEXT J : NEXT I
1090  RETURN
2000  REM TÁBLÁZATOK TÖRLÉSE
2010  FOR I=1 TO N : T(I)=0 : T1(I)=0 : NEXT I
2020  FOR I=1 TO M : S(I)=0 : S1(I)=0 : NEXT I
2030  RETURN
3000  REM EREDMÉNYEK KIÍRÁSA
3010  CLS : PRINT "TANULÓK ÁTLAGAI"
3020  FOR I=1 TO N
3030        IF INT(I/12)<I/12 THEN 3060
3040        PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
3050        IF INKEY$="" THEN 3050 ELSE CLS
3060        IF T1(I)>0 THEN PRINT I,T(I)/T1(I)
3070  NEXT I
3080  PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
3090  IF INKEY$="" THEN 3090
4010  CLS : PRINT "TANTÁRGYAK ÁTLAGAI"
```

```
4020    FOR I=1 TO M
4030        IF S1(I)>0 THEN PRINT I,S(I)/S1(I)
4040        IF INT(I/12)<I/12 THEN 4070
4050        PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
4060        IF INKEY$="" THEN 4060 ELSE CLS
4070    NEXT I
4080    RETURN
```

• • •

GY.28. "szigetek száma"

A mért szakaszon ott van sziget, ahol a tenger két szélső partja között pozitív értékeket találunk. Nem lenne jó megoldás azonban, ha azt számolnánk, hogy hány helyen találtunk pozitív számot, a szigetek ugyanis különböző méretűek lehetnek (egy vagy több mérési pont tartozhat hozzájuk). A megoldást úgy találjuk meg, hogy a szigetek száma helyett mást számolunk, amelyek száma azonban megegyezik a szigetek számával. Ez a más a szigetek bal oldali partja lesz. A mérési adatok számát N jelöli, a mérési adatokat pedig az A(N) vektorban tároljuk. Az eredményt az S változóban kapjuk. Sziget bal oldala ott van, ahol egy mérési pontban pozitív értéket találunk úgy, hogy az előző még 0 volt. Csupán egy kivételt találunk: a tenger jobb oldali partja is ilyen, úgyhogy a kapott számból egyet levonva kapjuk meg a szigetek számát.

Program:

```
        Be: N          [N>2 és egész, legalább 2 mérési pontnak lenni kell]
        Ciklus I=1-től N-ig
            Be: A(I)                    [A(I)>=0,A(1)>0,A(N)>0]
        Ciklus vége
        S:=0
        Ciklus I=2-től N-ig             ; tudjuk, hogy az első mérési pont még
                                        ; a szárazföldhöz tartozik
            Ha A(I)>0 és A(I-1)=0 akkor S:=S+1
        Ciklus vége
        S:=S-1                          ; a szigetek száma
        Ha S<0 akkor S:=0               ; nem volt tenger sem
        Ki: S
Program vége.
```

A BASIC programban külön fogjuk ellenőrizni A(1) és A(N) értékét.

```
 10    CLS : PRINT "SZIGETEK SZÁMA" : PRINT
 20    INPUT "MÉRÉSI PONTOK SZÁMA(>2)";N
 30    IF N<3 OR N<>INT(N) THEN 20
 40    DIM A(N)
 50    FOR I=1 TO N
 60          PRINT I;". MÉRÉSI ADAT";
 70          INPUT A(I)
 80          IF A(I)<0 THEN 60
 90          IF (I=1 OR I=N) AND A(I)=0 THEN 60
100    NEXT I
110    REM SZIGETEK SZÁMÁNAK MEGHATÁROZÁSA
120    S=0
130    FOR I=2 TO N
140          IF A(I)>0 AND A(I-1)=0 THEN S=S+1
150    NEXT I
160    S=S-1
170    IF S<0 THEN S=0
180    PRINT "A SZIGETEK SZÁMA=";S
190    STOP
```

• • •

GY.29. "tengerszélesség"

A mért szakaszon a tenger az első 0 értéknél kezdődik és az utolsónál fejeződik be. Ha tudjuk, hogy van 0 érték az adatok között, akkor ezek megkeresése igen egyszerű feladat, a tenger szélessége pedig a (két hely indexének különbsége +1)*5 km lesz.
A mérési adatok számát N jelöli, a mérési adatokat pedig az A(N) vektorban tároljuk. Az eredményt az S változóban kapjuk.

Program:

```
Be:N          [N>2 és egész --> legalább 2 mérési pontnak lenni kell]
Ciklus I=1-től N-ig
      Be: A(I)          [A(I)>=0,A(1)>0,A(N)>0, van 0 értékű]
Ciklus vége
E:=1                    ; bal oldali part keresése
Ciklus amíg A(E)>0
      E:=E+1
Ciklus vége
U:=N                    ; jobb oldali part keresés
```

```
        Ciklus amíg A(U)>0
                U:=U-1
        Ciklus vége
        S:=U-E+1                    ; a tenger szélessége
        Ki:S*5
Program vége.
```

A BASIC programban külön fogjuk ellenőrizni A(1) és A(N) értékét. A J változóban jelezzük, hogy volt-e a mért szakaszon tenger (értéke ekkor 1 lesz).

```
 10     CLS : PRINT "A TENGER SZÉLESSÉGE" : PRINT
 20     INPUT "MÉRÉSI PONTOK SZÁMA(>2)";N
 30     IF N<3 OR N<>INT(N) THEN 20
 40     DIM A(N) : J=0 : REM ÉRTÉKE >0 LESZ, HA VAN TENGER
 50     FOR I=1 TO N
 60          PRINT I;. MÉRÉSI ADAT"; : INPUT A(I)
 70          IF A(I)<0 THEN 60
 80          IF (I=1 OR I=N) AND A(I)=0 THEN 60
 90          IF A(I)=0 THEN J=1
100     NEXT I
110     IF J=0 THEN PRINT "NINCS TENGER" : STOP
120     E=1 : REM BAL OLDALI TENGERPART
130     IF A(E)=0 THEN 160
140          E=E+1
150     GOTO 130
160     U=N : REM JOBB OLDALI TENGERPART
170     IF A(U)=0 THEN 200
180          U=U-1
190     GOTO 170
200     S=U-E+1
210     PRINT "A TENGER SZÉLESSÉGE=";S*5;"KM"
220     STOP
```

GY.30. "legszélesebb sziget"

A mért szakaszon ott van sziget, ahol a tenger két szélső partja között pozitív értékeket találunk. A megoldást úgy találjuk meg, hogy figyeljük a szigetek bal oldali partját és az ezután következő jobb oldali partnál feljegyezzük a távolságot. Ezek közül kell kiválasztani a legnagyobbat. A mérési adatok számát N jelöli, a mérési adatokat pedig az A(N) vektorban tároljuk. Az eredményt az S változóban kapjuk. Sziget bal oldala ott van, ahol egy mérési pontban pozitív értéket találunk úgy, hogy az előző még 0 volt, jobb oldal pedig ott, ahol az aktuális mérés értéke már 0 (benne vagyunk a tengerben) és az előző pozitív.

```
Program:
        Be:N        [N>2 és egész --> legalább 2 mérési pontnak lenni kell]
        Ciklus I=1-től N-ig
                Be: A(I)            ; [A(I)>=0,A(1)>0,A(N)>0]
        Ciklus vége
        S:=0                        ; a legszélesebb sziget szélessége
        Ciklus  I=2-től N-ig        ; tudjuk, hogy az első mérési pont még
                                    ; a szárazföldhöz tartozik
                Ha(A(I)>0 és A(I-1)=0 akkor E:=I
                Ha A(I)=0 és A(I-1)>0 akkor
                        Ha I-E>S akkor S:=I-E
        Ciklus vége
        Ki:S
Program vége.
```

A BASIC programban külön fogjuk ellenőrizni A(1) és A(N) értékét.

```
 10     CLS  : PRINT "LEGSZÉLESEBB SZIGET" : PRINT
 20     INPUT  "MÉRÉSI PONTOK SZÁMA";N
 30     IF N<3 OR N<>INT(N) THEN 20
 40     DIM A(N)
 50     FOR I=1 TO N
 60          PRINT I;". MÉRÉSI ADAT";
 70          INPUT A(I)
 80          IF A(I)<0 THEN 60
 90          IF (I=1 OR I=N) AND A(I)=0 THEN 60
100     NEXT I
110     REM A LEGSZÉLESEBB SZIGET SZÉLESSÉGÉNEK MEGHATÁROZÁSA
120     S=0
130     FOR I=2 TO N
140          IF A(I)>0 AND A(I-1)=0 THEN E=I
150          IF A(I)=0 AND A(I-1)>0 THEN IF I-E>S THEN S=I-E
160     NEXT I
170     PRINT  "A LEGNAGYOBB SZÉLESSÉGE=";S
180     STOP
```

Gy.31. "ciklikus titkosírás"

A szöveget (S$) betűnként fogjuk kódolni, ill. dekódolni. A kódolás esetén a betű belső ábrázolási kódjához hozzáadva az eltolási konstanst (L) kapjuk meg az eredményt, ill. ha ez több lenne a legnagyobb kódú betű kódjánál, akkor levonjuk még belőle a kódtáblázat elemszámát (így kapjuk a ciklikus eltolást). Dekódoláskor a hozzáadás helyett kivonást alkalmazunk. Készítsük el az egy betű (A$) kódolását végző eljárást (eredmény=B$), ill. a dekódolást! A megoldásban az angol ABC 26 betűjét használjuk.

Betűkódolás(A$,B$):

```
        B:=Kód(A$)
        B:=B+L : Ha B>Kód("Z") akkor B:=B-26
        B$:=Karakter(B)        ; a B kódú karakter előállítása
Eljárás vége.
```

Betűkódolás(A$,B$):

```
        B:=Kód(A$)
        B:=B-L : Ha B<Kód("A") akkor B:=B+26
        B$:=Karakter(B)        ; a B kódú karakter előállítása
Eljárás vége.
```

A teljes BASIC programban választhatunk, hogy kódolást, vagy dekódolást akarunk elvégezni, majd a tevékenység elvégzése után dönthetünk, hogy akarunk-e még valamit csinálni.

```
 10    CLS : PRINT "TITKOSÍRÁS" : PRINT
 20    PRINT "LEHETŐSÉGEK:" : PRINT
 30    PRINT TAB(5);"1. SZÖVEG KÓDOLÁSA"
 40    PRINT TAB(5);"2. SZÖVEG DEKÓDOLÁSA"
 50    PRINT : INPUT "VÁLASZTÁS";V
 60    IF V<1 OR V>2 OR V<>INT(V) THEN 10
 70    INPUT "AZ ÁTALAKÍTANDÓ SZÖVEG";S$
 80    INPUT "AZ ELTOLÁSI KONSTANS(1-26)";L
 90    IF L<1 OR L>26 OR L<>INT(L) THEN 80
100    E$=""
110    FOR K=1 TO LEN(S$)
120        A$=MID$(S$,K,1) : REM A K. BETŰ
130        ON V GOSUB 300,400 : REM KÓDOLÁS VAGY DEKÓDOLÁS
140        E$=E$+B$ : REM AZ EREDMÉNY HOZZÁVÉTELE
150    NEXT K
180    PRINT "AZ EREDMÉNY=";E$
190    INPUT "FOLYTASSUK(I,N)";V$ : IF LEN(V$)=0 THEN 190
```

```
200     V$=LEFT$(V$,1)
210     IF V$="I" OR V$="i" THEN 10
220     STOP
300     REM KÓDOLÁS
310         B=ASC(A$)
320         B=B+L : IF B>ASC("Z") THEN B=B-26
330         B$=CHR$(B)
340     RETURN
400     REM DEKÓDOLÁS
410         B=ASC(A$)
420         B=B-L : IF B<ASC("A") THEN B=B+26
430         B$=CHR$(B)
440     RETURN
```

• • •

GY.32. "titkosírás"

A papírlap modelljének egy egész típusú mátrixot használunk [A%(10,10)], amelyben logikai értékeket tárolunk. Az "IGAZ" érték jelentse azt, hogy ott a lap ki van vágva, a "HAMIS" pedig azt, hogy nincs. A$(10,10) tartalmazza a betűket, S$ lesz a megfejtés. A szöveg első részének meghatározása (N=10):

Eljárás:

```
        Ciklus I=1-től N-ig
            Ciklus J=1-től M-ig
                Ha A%(I,J) akkor S$=S$+A$(I,J)
            Ciklus vége
        Ciklus vége
Eljárás vége.
```

Nézzük, mit jelent az elforgatás 90 fokkal jobbra. Először azt az elemet kell nézni, amelyik az 1. oszlop legalján volt, utána a fölötte levőt, és így tovább. Ezután ugyanezt végre kell hajtani a 2., 3. stb. oszlopokkal is. Azaz, ha eddig az (I,J) elemet vizsgáltuk, ezután az (N–J+1,I) elemet kell vizsgálni miközben I=1,...,N; J=1,...,N. A következő elfordítás után az (N–I+1,N–J+1), majd pedig a (J,N–I+1) elemet. Írhatnánk még három eljárást a további három rész meghatározására, de módosíthatjuk az első eljárásunkat is. Így egy lépésben a szöveg 4 részét fogjuk megfejteni (S1$,S2$,S3$,S4$).

Eljárás:

```
        Ciklus I=1-től N-ig
                Ciklus J=1-től M-ig
                        Ha A%(I,J)              akkor S1$=S1$+A$(I,J)
                        Ha A%(N-J+1,I)          akkor S2$=S2$+A$(I,J)
                        Ha A%(N-I+1,N-J+1)      akkor S3$=S3$+A$(I,J)
                        Ha A%(J,N-I+1)          akkor S4$=S4$+A$(I,J)
                Ciklus vége
        Ciklus vége
Eljárás vége.
```

Így a BASIC programban ezen 4 rész összefűzéséről kell gondoskodnunk (valamint a változóink kezdőértékeiről), s megkapjuk a megfejtést.

```
 10     CLS  : PRINT "TITKOSÍRÁS MEGFEJTÉSE" : PRINT
 20     N=10 : REM 10x10-ES MÁTRIXOT HASZNÁLUNK
 30     DIM A$(N,N),A%(N,N)
 40     FOR I=1 TO N : FOR J=1 TO N
 50          PRINT "(";I;",";J;"). HELYEN LEVŐ BETŰ";
 60          INPUT A$(I,J) : IF LEN (A$(I,J))<>1 THEN 50
 70     NEXT J,I
 80     DATA ....... (ITT KELL SZEREPELNI A MASZK ÉRTÉKEINEK SOR-
                     FOLYTONOSAN: -1 JELZI A KIVÁGÁST, 0 , HA NINCS)
 90     FOR I=1 TO N : FOR J=1 TO N
100          READ A%(I,J)
110     NEXT J : NEXT I
120     REM MEGFEJTÉS ELKÉSZÍTSÉSE
130     S1$="" : S2$="" : S3$="" : S4$=""
140     FOR I=1 TO N : FOR J=1 TO N
150          IF A%(I,J)                 THEN S1$=S1$+A$(I,J)
160          IF A%(N-J+1,I)             THEN S2$=S2$+A$(I,J)
170          IF A%(N-I+1,N-J+1)         THEN S3$=S3$+A$(I,J)
180          IF A%(J,N-I+1)             THEN S4$=S4$+A$(I,J)
190     NEXT J,I
200     S$=S1$+S2$+S3$+S4$
210     PRINT "A MEGFEJTÉS:" : PRINT S$
220     STOP
```

• • •

6. Feladatmegoldások haladók számára

GY.33. "szövegben rész-szövegek cseréje"

Adott három szöveg, jelöljük őket A,B,C-vel! Cseréljük ki A szövegben B szöveg minden előfordulását C szövegre! Két segédváltozót használunk, P1 mutatja, hogy meddig dolgoztuk fel az A szöveget; P2 azt, hogy hol találtuk meg a B szöveget.
Az algoritmus: megkeressük a B első előfordulását, az eredményhez hozzáfűzzük az A szöveg változatlan részét P1-től P2-ig (P2–P1 db betűt) és az új C szöveget, majd P1 változót megnövelve a B hosszával a keresést folytatjuk. Az eljárás befejeződik, ha már nem találtunk B szöveget az A-ban (ilyenkor a maradék részt még át kell másolni), vagy ha P1 változó már nagyobb, mint az A hossza – ez az az eset, ha az A szöveg pont a B szöveggel végződik. Az eredmény szöveget D$ változóban állítjuk elő.

```
Program:
        Be: A$,B$,C$
        P1:=1 : P2:=1 : D$:=""
        A:=hossz(A$)-hossz(B$)+1          ; ezután már nem lehet B$
        Ciklus amíg P2<=A
                Ciklus amíg P2<=A és nem B$ következik A$-ban
                        P2:=P2+1
                Ciklus vége
                Ha P2<=A akkor Szövegrészcsere: P2:=P2+hossz(B$) : P1:=P2
        Ciklus vége
        Maradékmásolás (P1-től, D$-ba)
        Ki:D$
Program vége.
```

A BASIC program:

```
 10     CLS  : PRINT "SZÖVEGEK ÁTGYÚRÁSA" : PRINT : CLEAR 1000
 20     INPUT "KÉREM AZ ÁTALAKÍTANDÓ SZÖVEGET";A$
 30     INPUT  "MIT CSERÉLJEK";B$
 40     INPUT  "MIRE";C$
 50     REM SZÖVEGEK ÁTALAKÍTÁSA
 60     P1=1 : P2=1 : D$=""
 70     A=LEN(A$)-LEN(B$)+1
 80     IF P2>A THEN 210
 90          IF P2>A OR B$=MID$(A$,P2,LEN(B$)) THEN 120
100               P2=P2+1
110          GOTO 90
120          IF P2>A THEN 200
```

```
130          REM SZÖVEGRÉSZLETEK CSERÉJE
140          D$=D$+MID$(A$,P1,P2-P1)+C$
150          P2=P2+LEN(B$) : P1=P2
200     GOTO 80
210     REM MARADÉK ÁTMÁSOLÁSA
220     D$=D$+MID$(A$,P1)
230     PRINT "A KICSERÉLT SZÖVEG:"
240     PRINT D$
250     STOP
```

• • •

GY.34. "tördelés"

A program beolvas egy N sorból álló szöveget. L lesz a képre írható lapok száma, H pedig az egy sorba írható betűk száma. Feltesszük, hogy beíráskor a szavakat nem választották el. Sor végén és elején szóközt nem tettek, a szövegben semmilyen formázásra (pl. bekezdés) nincs szükség.
A megoldásban a szöveget szavanként írjuk ki a képernyőre, s ha egy sor betelt, akkor új sort kezdünk, ha egy lap betelt, akkor új lapot kezdünk.

Program:

```
        Be: N,L,H        [N,L,H>0 és egészek]
        Ciklus I=1-től N-ig
              Be: A$(I)
              A$(I):=A$(I)+" "        ; a végére szóközt teszünk
        Ciklus vége
        S:=0 : T:=0        ; aktuális sor és oszlop
        Ciklus K=1-től N-ig
             I:=1
             Ciklus amíg I<=hossz(A$)
              J:= a következő szó eleje
              Ha T+J-I>H akkor Új sor : S:=S+1 : T:=0
                                  Ha S>L akkor Új lap : S:=0
              Elágazás vége
              Ki: I-től J-1-ig terjedő rész : T:=T+J-I
              I:=J
           Ciklus vége
        Ciklus vége
Program vége.
```

6. Feladatmegoldások haladók számára

A BASIC program nem tördeli a szót, ha olyan szót kellene kiírnia, amely nem fér el a kép egy sorában. A szöveg csak nagybetűket tartalmaz.

```
 10   CLS : PRINT "SZÖVEG TÖRDELÉS" : PRINT : CLEAR 1000
 20   INPUT "HÁNY SOR LESZ";N : IF N<1 OR N<>INT(N) THEN 20
 30   INPUT "SOROK, OSZLOPOK SZÁMA A KÉPEN";L,H
 40   IF L<1 OR H<1 OR L<>INT(L) OR H<>INT(H) THEN 30
 50   DIM A$(N)
 60   FOR I=1 TO N
 70        PRINT I;". SOR";
 80        INPUT A$(I)
 90        A$(I)=A$(I)+" "
100   NEXT I
110   S=0 : T=0
120   FOR K=1 TO N
130        I=1
140        IF I>LEN(A$(K)) THEN 330
150             J=I
160             IF J>LEN(A$(K)) THEN 250
170             IF MID$(A$(K),J,1)<"A" OR
                        MID$(A$(K),J,1)>"Z" THEN 200
180                  J=J+1
190             GOTO 160
200             REM KÖVETKEZŐ SZÓ ELEJÉNEK KERESÉSE
210             IF J>LEN(A$(K)) THEN 250
220             IF MID$(A$(K),J,1)>="A" AND
                        MID$(A$(K),J,1)<="Z" THEN 250
230                  J=J+1
240             GOTO 210
250             IF T+J-I<=H THEN 300
260             PRINT : S=S+1 : T=0
270             IF S<L THEN 300
280             PRINT@15*64,"NYOMJ LE EGY BILLENTYŰT!";
290             IF INKEY$="" THEN 290 ELSE CLS : S=0
300             PRINT MID$(A$(K),I,J-I); : T=T+J-I
310             I=J
320        GOTO 140
330   NEXT K
340   STOP
```

• • •

GY.35. „morze"

Az angol ABC nagybetűi morze jeleit tároljuk az A$(26) szöveg típusú vektorban, "—" jelöli a hosszú, "." pedig a rövid jeleket, " " a szüneteket. A szavak közötti szóközt egy újabb " " jel jelöli. Minden betű morze kódja tartalmazza az őt követő szünetjelet is. Legyen a kódolandó, ill. a dekódolandó szöveg a K$ változóban, eredménye pedig az E$-ban! Ekkor a kódolás menete a következő:

Kódolás:

```
        Ciklus I=1-től K$ hosszáig
                B$:=K$ I. karaktere : B:=B$ kódja-64
                Ha B$=" " akkor E$:=E$+B$ különben E$:=E$+A$(B)
        Ciklus vége
Eljárás vége.
```

A dekódolás pedig így végezhető (ha az értelmetlen jelsorozatok helyett a ■-jelet használjuk):

Dekódolás:

```
        I:=1
        Ciklus amíg I<K$ hossza
                Ha K$ I. karaktere=" " akkor E$:=E$+" " : I:=I+1
                        különben J:=1
                                Ciklus amíg J<=26 és K$ I-edik helyén nem A$(J) kezdődik
                                        J:=J+1
                                Ciklus vége
                                Ha J>26 akkor E$:=E$+"■" : I:=I+1
                                        különben E$:=E$+a J-edik betű
                                                I:=I+a J-edik betű hossza
                                Elágazás vége
                Elágazás vége
        Ciklus vége
Eljárás vége.
```

A BASIC programban az egyes betűk morze-kódjait DATA utasításban adjuk meg (egy sorban öt betű lesz).

```
10      DATA "._ ","_... ","_._. ","_.. ",". "
20      DATA "..._. ","__. ",".... ",".. ",".___ "
30      DATA "_._ ",".._.. ","__ ","_. ","___ "
```

```
 40   DATA ".—–. ","—–.— ",".—. ","... ","— "
 50   DATA "..— ","...— ",".—— ","—..— ","—.—— "
 60   DATA "——.. "
 70   DIM A$(26)
 80   FOR I=1 TO 26
 90        READ A$(I)
100   NEXT I
110   CLS  : PRINT "MORZE KÓDOLÁS, DEKÓDOLÁS" : PRINT
120   PRINT "LEHETŐSÉGEK:" : PRINT
130   PRINT TAB(5);"1. KÓDOLÁS"
140   PRINT TAB(5);"2. DEKÓDOLÁS"
150   PRINT : INPUT "VÁLASZTÁS";V
160   IF V<>1 AND V<>2 THEN 110
170   INPUT "AZ ÁTALAKÍTANDÓ SZÖVEG";K$
180   ON V GOSUB 300,400
190   PRINT "AZ EREDMÉNY:" : PRINT E$
200   INPUT "VAN MÉG TEVÉKENYSÉG(I,N)";V$
210   IF V$="I" OR V$="i" THEN 110
220   STOP
300   REM KÓDOLÁS
310   E$=""
320   FOR I=1 TO LEN(K$)
330        B$=MID$(K$,I,1) : B=ASC(B$)–64
340        IF B$=" " THEN E$=E$+B$ ELSE
                  IF B<27 AND B>0 THEN E$=E$+A$(B)
350   NEXT I
360   RETURN
400   REM KÓDOLÁS
410   I=1
420   IF I>=LEN(K$) THEN 550
430        IF MID$(K$,I,1)=" " THEN E$=E$+" " : I=I+1 : GOTO 520
440        J=1
450        IF J>26 THEN 500
460        IF MID$(K$,I,LEN(A$(J)))=A$(J) THEN 500
470             J=J+1
480        GOTO 450
500        IF J>26 THEN E$=E$+"■" : I=I+1 : GOTO 520
510        E$=E$+CHR$(64+J) : I=I+LEN(A$(J))
520   REM KÖZÖS ÁG
530   GOTO 420
550   RETURN
```

• • •

GY.36. "hangrend"

Először el kell döntenünk, hogy mit értünk a szó fogalmán: a szó olyan betűvel kezdődő, betűkből és jelből (kötőjel) álló jelsorozat, amelyet két végén más jel határol (vagy a szöveg eleje és vége).
A definícióból látható, hogy a szavak felismerésében speciális dolgokra is szükségünk lenne (a szöveg elejének és végének kezelése), ezért a szöveg elejére és végére tegyünk például egy "*" jelet. Ezzel a definíció utolsó megjegyzése elhanyagolható!
Jelölje H az aktuális szó hangrendjét: értéke 1, ha magas; 2, ha mély és 3, ha vegyes hangrendű a szó, illetve 0, ha nem tartalmaz magánhangzót. Az S(3) vektor megfelelő indexű elemei fogják tartalmazni az egyes hangrendű szavak számát.
A hangrend-meghatározó eljárásnak úgy kell működnie, hogy magas hangrendű magánhangzóhoz az 1, mély hangrendűhöz a 2, egyéb jelhez pedig a 0 értéket rendelje. Ekkor vegyük egy szó betűihez rendelt értékek logikai összegét! Ebben a 0-k nem számítanak, azon szavakhoz, amelyekben csak magas magánhangzó volt (1-esek logikai összege) 1-et rendelünk hozzá; hasonlóan a csak mély magánhangzókhoz 2-t; amelyekben pedig magas és mély magánhangzó is volt, azokhoz az 1 és a 2 logikai összegét, a 3-at.

Program:

```
        Be:S$
        S$="*"+S$+"*"
        Ciklus I=2-től S$ hosszáig
                B$:=S$ I. betűje
                Ha szó kezdődik akkor H:=0
                X:=B$ hangrendje
                H:=H vagy X
                Ha szó végződik akkor S(H):=S(H)+1
        Ciklus vége
        Ki: S(1),S(2),S(3)
Program vége.
```

A BASIC programban használni fogjuk a szövegkezelő függvényeket. Bevezetünk néhány egyszerűsítést:

- csak nagybetűket használunk,
- az ékezetes HT–1080Z lehetőségeit kihasználjuk,
- betűnek tekintjük azt a karaktert, amelynek kódja 64 és 94 között van.

```
 10   CLS : PRINT "HANGREND VIZSGÁLAT" : PRINT : CLEAR 1000
 20   INPUT "A VIZSGÁLANDÓ SZÖVEG";S$
 30   S$="*"+S$+"*" : X1$=CHR$(64) : X2$=CHR$(94)
 40   FOR I=2 TO LEN(S$)
 50        B$=MID$(S$,I,1) : C$=MID$(S$,I-1,1)
 60        IF (C$<X1$ OR C$>X2$) AND (B$>=X1$ AND B$<=X2$)
                THEN H%=0
 70        GOSUB 1000 : REM X%=B$ HANGRENDJE
 80        H%=H%ORX%
 90        IF(C$>=X1$ AND C$<=X2$) AND (B$<X1$ OR B$>X2$)
                THEN S(H%)=S(H%)+1
100   NEXT I
110   PRINT "MAGAS HANGRENDŰ SZAVAK SZÁMA:";S(1)
120   PRINT "MÉLY HANGRENDŰ SZAVAK SZÁMA:";S(2)
130   PRINT "VEGYES HANGRENDŰ SZAVAK SZÁMA:";S(3)
140   STOP
1000  REM X%=B$ HANGRENDJE
1010  IF B$="A" OR B$="O" OR B$="U" OR B$="Á" THEN X%=2 : RETURN
1020  IF B$="É" OR B$="E" OR B$="I" OR B$="Ö" OR B$="Ü" THEN
                X%=1 : RETURN
1030  X%=0 : RETURN
```

• • •

GY.37. "rímképlet"

A vers maximum N soros, az egyes sorokat a B$(N) vektorban tároljuk. Felhasználunk egy 3 oszlopos táblázatot, amelyben a sorok utolsó két magánhangzóját, illetve a rímképletben hozzá tartozó betűt tároljuk (A$(N,3)).

Program:

```
        Be: N        [N>1 és egész]
        Ciklus I=1-től N-ig
                Be: B$(I)
        Ciklus vége
        Ciklus I=1-től N-ig
                A$(I,1):=B$(I) utolsó magánhangzója
                A$(I,2):=B$(I) utolsó előtti magánhangzója
        Ciklus vége
        Rímképlet-meghatározás
        Ciklus I=1-től N-ig
                Ki: A$(I,3)
        Ciklus vége
Program vége.
```

Rímképlet meghatározás:

```
        A$:="A" : A$(1,3):=A$
        Ciklus I=2-től N-ig
                J:=1
                Ciklus amíg J<I és ( A$(J,1)<>A$(I,1) vagy
                                A$(J,2)<>A$(I,2) )
                        J:=J+1
                Ciklus vége
                Ha J<I akkor A$(I,3):=A$(J,3)
                        különben A$:=következő betű: A$(I,3):=A$
        Ciklus vége
Eljárás vége.
```

A BASIC programban feltesszük, hogy minden sorban van legalább 2 magánhangzó (amennyiben nincs, úgy a feladatot nem tudjuk megoldani). A rímképlet betűit A-tól folyamatosan használjuk, s a képen egymás után írjuk ki őket.

```
 10   CLS  : PRINT "RÍMKÉPLET VIZSGÁLAT" : PRINT
 20   INPUT "SOROK SZÁMA";N : IF N<2 OR N<>INT(N) THEN 20
 30   DIM B$(N),A$(N,3)
 40   FOR I=1 TO N
 50        PRINT I;". SOR"; : INPUT B$(I)
 60   NEXT I
 70   REM UTOLSÓ 2 MAGÁNHANGZÓ SORONKÉNT
 80   FOR I=1 TO N
 90        J=LEN(B$(I))
100        GOSUB 1000 : REM K= 1, HA A J. BETÚ (Y$) MAGÁNHANGZÓ
110        IF K= 1 THEN 140
120             J=J-1
130        GOTO 100
140        A$(I,1)=Y$
150        GOSUB 1000 : REM K=1, HA A J. BETŰ (Y$) MAGÁNHANGZÓ
160        IF K=1 THEN 190
170             J=J-1
180        GOTO 150
190        A$(I,2)=Y$
200   NEXT I
210   REM RÍMKÉPLET MEGHATÁROZÁS
220   A$="A" : A$(I,3)=A$
```

```
230    FOR I=2 TO N
240        J=1
250        IF J<I AND ( A$(J,1)<>A$(I,1) OR A$(J,2)<>A$(I,2) )
                THEN 260 ELSE 280
260            J=J+1
270        GOTO 250
280        IF J<I THEN A$(I,3)=A$(J,3) ELSE A$=CHR$(ASC(A$)+1) :
                A$(I,3)=A$
290    NEXT I
300    REM EREDMÉNY KIÍRÁS
310    CLS : PRINT "RÍMKÉPLET" : PRINT
320    FOR I=1 TO N
330        PRINT A$(I,3);
340    NEXT I
350    STOP
1000   REM K=1, HA A J. BETŰ (Y$) MAGÁNHANGZÓ
1010   DATA 9,A,Á,E,É,I,O,Ö,U,Ü
1020   RESTORE
1030   READ M : Y$=MID$(B$(I),J,1)
1040   FOR M1=1 TO M
1050       READ X$
1060       IF X$=Y$ THEN K=1 : RETURN
1070   NEXT M1
1080   K=0 : RETURN
```

• • •

GY.38. "labdarúgó-bajnokság"

A bajnokságban N csapat szerepelt, egymással N*(N–1) mérkőzést játszottak (mindenki mindenkivel kétszer játszik). A mérkőzések eredményeit az A(N*(N–1),4) mátrixban kapja a program, ahol az 1. oszlopban az első csapat sorszáma, a 2. oszlopban a második sorszáma, a 3-ban az első rúgott góljai, a 4-ben pedig a második rúgott góljai szerepelnek. Meg kell határoznunk az egyes csapatok győzelmeinek, döntetlenjeinek, illetve vereségeinek számát, a rúgott és a kapott gólok számát, valamint az elért pontszámot. Ezután sorba kell rendezni a kapott eredményeket (első szempontunk a pontszám, második a gólkülönbség), majd ki kell írni a képernyőre.

Program:

Adatok beolvasása
Táblázat számítása
Sorbarendezés
Eredmények kiírása
Program vége.

Adatok beolvasása:

```
        Be: N          [N>1 és egész]
        Ciklus I=1-től N*(N-1)-ig
                Be: A(I,1),A(I,2)          [1<=A(I,1),A(I,2)<=N]
                Be: A(I,3),A(I,4)
        Ciklus vége
Eljárás vége.
```

Az eredménytáblázat kiszámításához sorba vesszük az eredményeket. A táblázat adatait (a fenti sorrendben) az E(N,6) táblázatban fogjuk tárolni.

Táblázat számítása:

```
        Ciklus I=1-től N*(N-1)-ig
                X:=A(I,1) : Y:=A(I,2)
                Ha A(I,3)>A(I,4) akkor  E(X,6):=E(X,6)+2        ; 2 pont
                                        E(X,1):=E(X,1)+1        ; győzelem
                                        E(Y,3):=E(Y,3)+1        ; vereség
                Ha A(I,3)<A(I,4) akkor  E(Y,6):=E(Y,6)+2        ; 2 pont
                                        E(Y,1):=E(Y,1)+1        ; győzelem
                                        E(X,3):=E(X,3)+1        ; vereség
                Ha A(I,3)=A(I,4) akkor  E(X,6):=E(X,6)+1        ; 1 pont
                                        E(Y,6):=E(Y,6)+1        ; 1 pont
                                        E(X,2):=E(X,2)+1        ; döntetlen
                                        E(Y,2):=E(Y,2)+1        ; döntetlen
                E(X,4):=E(X,4)+A(I,3)                   ; rúgott gólok
                E(X,5):=E(X,5)+A(I,4)                   ; kapott gólok
                E(Y,4):=E(Y,4)+A(I,4)                   ; rúgott gólok
                E(Y,5):=E(Y,5)+A(I,3)                   ; kapott gólok
        Ciklus vége
Eljárás vége.
```

Helyezzük el a táblázat 0. oszlopába minden csapat helyezési számát (0-tól számozva), majd ezen számok szerinti sorrendben írjuk ki az eredményt! A helyezési szám így egyszerűen azon csapatok száma, amelyek az adott csapatnál jobb eredményt értek el.

```
Sorbarendezés:
        Ciklus I=1-től N–1-ig
                Ciklus J=I+1-től N-ig
                        Ha
                            1. E(I,6)>E(J,6) akkor E(J,0):=E(J,0)+1
                            2. E(I,6)<E(J,6) akkor E(I,0):=E(I,0)+1
                            3. E(I,6)=E(J,6) akkor Gólkülönbség-függő növelés
                        Elágazás vége
                Ciklus vége
        Ciklus vége
Eljárás vége.

Gólkülönbség-függő növelés:

        X:=E(I,4)–E(I,5)
        Y:=E(J,4)–E(J,5)
        Ha X>Y akkor E(J,0):=E(J,0)+1
                különben Ha X<Y akkor E(I,0):=E(I,0)+1
Eljárás vége.

Eredmények kiírása:

        I:=0
        Ciklus amíg I<N
                Ciklus amíg Van I. helyen végzett csapat
                        Ki: I,E(sorszám)                ;minden adatot kiír
                Ciklus vége
                I:=I+1
        Ciklus vége
Eljárás vége.
```

A BASIC program:

```
  10    CLS  : PRINT "BAJNOKSÁG TÁBLÁZATA" : PRINT
  20    GOSUB 1000 : REM ADATOK BEOLVASÁSA
  30    GOSUB 2000 : REM TÁBLÁZAT SZÁMÍTÁSA
  40    GOSUB 3000 : REM SORBARENDEZÉS
  50    GOSUB 4000 : REM EREDMÉNYEK KIÍRÁSA
  60    STOP
1000    REM ADATOK BEOLVASÁSA
1010    INPUT "CSAPATOK SZÁMA";N : IF N<2 OR N<>INT (N) THEN 1010
1020    DIM A(N*(N–1),6),E(N,6)
```

```
1030    FOR I=1 TO N*(N-1)
1040        INPUT "CSAPATSORSZÁMOK";A1,A2
1050        IF A1<1 OR A1>N OR A1<>INT(A1) OR A2<1 OR A2>N OR A1=A2
                OR A2<>INT(A2) THEN 1040 ELSE A(I,1)=A1 : A(I,2)=A2
1060        INPUT "EREDMÉNY";A(I,3),A(I,4)
1070    NEXT I
1080    RETURN
2000    REM TÁBLÁZAT SZÁMÍTÁSA
2010    FOR I=1 TO N*(N-1)
2020        X=A(I,1) : Y=A(I,2)
2030        IF A(I,3)>A(I,4) THEN E(X,6)=E(X,6)+2 : E(X,1)=E(X,1)+1 :
                E(Y,3)=E(Y,3)+1 : GOTO 2060
2040        IF A(I,3)<A(I,4) THEN E(Y,6)=E(Y,6)+2 : E(Y,1)=E(Y,1)+1 :
                E(X,3)=E(X,3)+1 : GOTO 2060
2050        E(X,6)=E(X,6)+1 : E(Y,6)=E(Y,6)+1 : E(X,2)=E(X,2)+1 :
                E(Y,2)=E(Y,2)+1
2060        E(X,4)=E(X,4)+A(I,3) : E(X,5)=E(X,5)+A(I,4)
2070        E(Y,4)=E(Y,4)+A(I,4) : E(Y,5)=E(Y,5)+A(I,3)
2080    NEXT I
2090    RETURN
3000    REM SORBARENDEZÉS
3010    FOR I=1 TO N-1
3020        FOR J=I+1 TO N
3030            IF E(I,6)>E(J,6) THEN E(J,0)=E(J,0)+1 : GOTO 3070
3040            IF E(I,6)<E(J,6) THEN E(I,0)=E(I,0)+1 : GOTO 3070
3050            X=E(I,4)-E(I,5) : Y=E(J,4)-E(J,5)
3060            IF X>Y THEN E(J,0)=E(J,0)+1 ELSE IF X<Y
                    THEN E(I,0)=E(I,0)+1
3070        NEXT J
3080    NEXT I
3090    RETURN
4000    REM EREDMÉNYEK KIÍRÁSA
4010    I=0
4020    S1=0 : S2=12
4030    IF I=N THEN 4140
4040        J=1
4050        IF J>N THEN 4120
4060            IF E(J,0)=I THEN 4070 ELSE 4110
4070            IF S1=S2 THEN S1=0 : GOSUB 5000 : REM LAPOZÁS
4080            IF S1=0 THEN CLS : PRINT "HELY. CSAPAT   GY.   D.   V.";
                    "      GÓLOK   PONT"
4090            S1=S1+1 : PRINT I+1;J;
```

```
4100            FOR K=1 TO 6 : PRINT TAB(15+K*5);E(J,K); : NEXT K
4110            PRINT : J=J+1 : GOTO 4050
4120        I=I+1
4130    GOTO 4030
4140    RETURN
```

• • •

GY.39. "futóverseny"

N versenyző vett részt a versenyen, jelölje R(N) a rajtszámokat, N$(N) a neveket, O$(N) pedig az országneveket! A célba legfeljebb N versenyző érkezett (N-nél kevesebb lehet), számuk M lesz, itt tudjuk a beérkezési sorrendben az időket (I(N)) és a rajtszámokat (R1(N)). Az eredmények kiírásánál ügyelni kell a holtversenyre is (észrevehetjük, ha az aktuális versenyző az előzővel azonos időt ért el).

Program:

```
        Nevezések beolvasása
        Versenyeredmények beolvasása
        Ciklus I=1-től M-ig
            J:=az R1(I) rajtszámú versenyző sorszáma az első 3 táblázatban
            Ha I(I)<>I(I-1) akkor Ki: I
            Ki: R(J);N$(J),O$(J);I(I)
        Ciklus vége
Program vége
```

Adott elemet rendezett sorozatban leggyorsabban logaritmikus kereséssel találhatunk meg.

Versenyző keresése:

```
        A:=1 : F:=N
        Ciklus
            J:=INT((A+F)/2)
            Ha R1(I)<R(J) akkor F:=J-1 különben
                Ha R1(I)>R(J) akkor A:=J+1
        amíg A<=F és R1(I)<>R(J)
        Ciklus vége
        Ha A>F akkor Hiba
Eljárás vége.
```

A BASIC programban az adatok beolvasását végző részt kazettás adatbeolvasással írjuk meg. Az első file a "NEVEZÉS", a második pedig az "EREDMÉNY" azonosítóval kezdődik.

```
 10   CLS : PRINT "FUTÓVERSENY ÉRTÉKELÉSE" : PRINT
 20   GOSUB 1000 : REM NEVEZÉSEK BEOLVASÁSA
 30   GOSUB 2000 : REM EREDMÉNYEK BEOLVASÁSA
 40   S1=0 : S2=12 : REM KÉPKEZELÉS SEGÉDVÁLTOZÓI
 50   FOR I=1 TO M
 60        GOSUB 500 : REM J=A VERSENYZŐ SORSZÁMA
 70        IF S1<S2 THEN 100
 80        S1=0 : PRINT @15*64,"BILLENYTŰ LENYOMÁSRA LAPOZOK";
 90        IF INKEY$="" THEN 90
100        IF S1>0 THEN 130
110        CLS : PRINT "HELYEZÉS","RSZ. NÉV","ORSZÁG","IDŐ"
120        PRINT
130        IF I(I)=I(I-1) THEN PRINT , ELSE PRINT I,
140        PRINT R(J);N$(J),O$(J),I(I)
150        S1=S1+1
160   NEXT I
170   STOP
500   REM KERESÉS
510   A=1 : F=N
520        J=INT((A+F)/2)
530        IF R1(I)<R(J) THEN F=J-1 : GOTO 550
540        IF R1(I)>R(J) THEN A=J+1
550   IF A<=F AND R1(I)<>R(J) THEN 520
560   IF A>F THEN PRINT "NEM NEVEZETT VERSENYZŐ";R1(I) : STOP
570   RETURN
1000  N$="NEVEZÉS" : GOSUB 3000 : REM FELKÉSZÜLÉS
1020  INPUT■-1,N : REM NEVEZÉSEK SZÁMA
1030  DIM R(N),N$(N),O$(N)
1040  FOR I=1 TO N
1050       INPUT■-1,R(I),N$(I),O$(I)
1060  NEXT I
1070  RETURN
2000  N$="EREDMÉNY" : GOSUB 3000 : REM FELKÉSZÜLÉS
2010  DIM I(N),R1(N)
2020  M=1
2030  INPUT■-1,I(M),R1(M)
2040  IF I(M)=0 THEN 2070
2050       M=M+1
```

```
2060    GOTO 2030
2070    M=M-1 : REM A BEÉRKEZETTEK SZÁMA
2080    RETURN
3000    REM FELKÉSZÜLÉS
3010    PRINT "ÁLLÍTSD A KAZETTÁT A(Z) ";N$;" ADATAIHOZ!"
3020    PRINT "NYOMJ LE EGY BILLENTYŰT!";
3030    IF INKEY$="" THEN 3030
3040    INPUT ■1,V$
3050    IF V$<>N$ THEN 3040
3060    RETURN
```

• • •

GY.40. "helyi függőleges"

N*M mérési adatunk van, az egyszerűség kedvéért 0 és 1 értékűek lehetnek. 1 jelenti a "fényes" pontokat, 0 pedig a "sötéteket". A középpontnak abban az oszlopban kell lennie, ahol a körbe tartozó legmagasabban (ill. legalacsonyabban) elhelyezkedő pont található. A középpont sor sorszáma hasonlóan lenne megtalálható, de lesz rá egyszerűbb megoldásunk is. A megoldásban feltesszük, hogy a teljes kör a térképünkön van. A határoló 0-kat pedig a program veszi hozzá az adatokhoz.

Program:

```
        Be: N,M          [N,M>0 és egész]
        Mérési adatok beolvasása
        Legfelső pont keresése          ; O=az oszlop sorszáma
        Legbaloldalibb pont keresése    ; S=a sor sorszáma
        Ki: S,O
Program vége.
```

A legfelső pont meghatározása abból áll, hogy megvizsgáljuk, melyik sorban jelenik meg először egy 1-es, s ha itt több van, akkor vesszük közülük a középsőt (ha páros számúan vannak, akkor a kerekítés miatt egy kicsit pontatlan lesz az eredményünk).

Legfelső pont keresése:

```
        I:=1
        Ciklus amíg az I. sorban nincs 1-es
                I:=I+1
        Ciklus vége
        E:=1          ; az első 1-es oszlopszáma lesz
```

```
        Ciklus amíg M(I,E)=0
                E:=E+1
        Ciklus vége
        U:=E          ; az utolsó 1-es sorszáma lesz
        Ciklus amíg M(I,U+1)<>0
                U:=U+1
        Ciklus vége
        O:=(U+E)/2
Eljárás vége.
```

A középpont sorának megkeresésére ezután egyszerűbb eljárást írhatunk. A kapott oszlopban keressük meg a legalsó 1-est, s a fent megtalált, ill. az alsó közötti középső sorban lesz a középpont.

Sormeghatározás:

```
        J:=I          ; itt volt a legfelső 1-es
        Ciklus amíg M(J+1,O)<>0
                J:=J+1
        Ciklus vége
        S:=(J+I)/2            ; az alsó és a felső között középen
Eljárás vége.
```

A BASIC programban elfogadunk 0-tól és 1-től különböző pozitív értékeket, de feltesszük, hogy ami nem 0, az 1-est jelent.
Abban a sorban van nem 0 érték, amelynek sorösszege nagyobb, mint 0.

```
 10     CLS : PRINT "HELYI FÜGGŐLEGES MEGHATÁROZÁSA" : PRINT
 20     INPUT "KÉP MÉRETEK";N,M
 30     IF N<1 OR M<1 OR N<>INT(N) OR M<>INT(M) THEN 20
 40     DIM M(N+1,M+1)
 50     FOR I=1 TO N
 60             PRINT I;". SOR ADATAI:"
 70             FOR J=1 TO M
 80                     PRINT J;". ADAT:";
 90                     INPUT M(I,J) : IF M(I,J)<0 THEN 80
100             NEXT J
110     NEXT I
120     GOSUB 200 : REM OSZLOP KOORDINÁTA
130     GOSUB 400 : REM SOR KOORDINÁTA
140     PRINT "A KERESETT KOORDINÁTÁK:";S;O
150     STOP
200     REM LEGFELSŐ PONT – OSZLOP KOORDINÁTA
210     I=1
```

```
220     A=0 : FOR J=1 TO M : A=A+M(I,J) : NEXT J : IF A>0 THEN 250
230         I=I+1
240     GOTO 220
250     REM ELSŐ NEM 0 MEGKERESÉSE
260     E=1
270     IF M(I,E)<>0 THEN 300
280         E=E+1
290     GOTO 270
300     REM UTOLSÓ 1-ES MEGKERESÉSE
310     U=E
320     IF M(I,U+1)=0 THEN 350
330         U=U+1
340     GOTO 320
350     O=(U+E)/2 : RETURN
400     REM SOR KOORDINÁTA MEGHATÁROZÁSA
410     J=I
420     IF M(J+1,O)=0 THEN 450
430         J=J+1
440     GOTO 420
450     S=(J+I)/2
460     RETURN
```

• • •

GY.41. "vonzáskörzet"

Jelölje N a megye városainak számát, X(N,2) a városok koordinátáit, R(N) pedig a vonzáskörzetük sugarát! A megye teljes térképe elfér a képernyő egy M*M-es négyzetén, a koordinátákat úgy adjuk meg, hogy a városok is erre a területre esnek.
A megoldásban fordított eljárást fogunk követni: először kirajzolunk minden pontot, majd a vonzáskörzetekbe esőket kitöröljük.

Program:

```
        Városok adatainak beolvasása
        Teljes ábra kitöltése
        Ciklus I=1-től N-ig
            Az I. város vonzáskörzetének törlése
        Ciklus vége
Program vége.
```

Egy város vonzáskörzete mindig egy kör, így egy körlap pontjait kell törölnünk.

Az I. város vonzáskörzetének törlése:

```
        Ciklus X=–R(I)-től R(I)-ig
                Z:=SQR(R(I) ↑ 2–X ↑ 2)
                Ciklus Y=–Z-től Z-ig
                        Törlés (X+X(I,1),Y+X(I,2))
                Ciklus vége
        Ciklus vége
Eljárás vége.
```

A BASIC program:

```
 10     CLS  : PRINT "VONZÁSKÖRZETEK MEGÁLLAPÍTÁSA" : PRINT
 20     M=40 : REM HASZNÁLHATÓ KÉPMÉRET
 30     INPUT "VÁROSOK SZÁMA";N : IF N<0 OR N<>INT(N) THEN 30
 40     DIM X(N,2),R(N)
 50     FOR I=1 TO N
 60          PRINT I;". VÁROS KOORDINÁTÁI"; : INPUT X(I,1),X(I,2)
 70          INPUT "VONZÁSKÖRZETÉNEK SUGARA";R(I) : IF R(I)<0 THEN 70
 80     NEXT I
 90     GOSUB 200 : REM TELJES ÁBRA KITÖLTÉSE
100     FOR I=1 TO N
110          GOSUB 300 : REM VÁROS VONZÁSKÖRZET TÖRLÉSE
120     NEXT I
130     STOP
200     REM TELJES ÁBRA
210     CLS
220     FOR I=1 TO M : FOR J=1 TO M
230                SET(I,J)
240     NEXT J : NEXT I
250     RETURN
300     REM I. VÁROS VONZÁSKÖRZET TÖRLÉSE
310     FOR X=–R(I) TO R(I)
320          Z=INT(SQR(R(I) ↑ 2–X ↑ 2))
330          FOR Y=–Z TO Z
340                RESET(X+X(I,1),Y+X(I,2))
350          NEXT Y
360     NEXT X
370     RETURN
```

• • •

6. Feladatmegoldások haladók számára

GY.42. "lövésértékelés"

Fogalmazzuk meg pontosan, mit értünk a feladaton, milyen feltételek mellett oldjuk meg! Koordinátarendszerünk origója a céltábla középpontjában van, a céltáblán a körök 1 cm távolságra helyezkednek el egymástól, az egyes koordinátákat is ebben a mértékegységben adjuk meg. 10 pontot ér az a lövés, amely a legbelső körvonalon belül, vagy magán a vonalon van, 9-et az első körgyűrűbe vagy külső körvonalára ért lövés,... A táblára 1 és 20 közötti tetszőleges darabszámú lövés érkezett. A versenyző eredménye a legjobb 10 lövés (ha van annyi) pontértékének összege lesz. A megoldás első lépésében be kell olvasni a lövések koordinátáit, majd kiszámoljuk a pontértékeket (vigyázni kell a határoknál, és a táblán kívülieket 0 értékűnek kell venni).

Jelölések:

N – lövések száma
X,Y – az aktuális lövés koordinátái
P(N) – a lövések pontértékei
L – az értékelendő lövések száma (10)
E – a versenyző eredménye

Program:

```
Be: N          [1 <= N <= 20 és N egész]
L:=10
Ciklus I=1-től N-ig
        Be: X,Y
        T:=SQR(X*X+Y*Y)
        Ha T=0 akkor P(I):=10
                 különben P(I):=INT(11-T) : Ha P(I)<0 akkor P(I):=0
Ciklus vége
Versenyző értékelése          [E = eredmény]
Pontszámok kiírása
Ki: E
Program vége.
```

Ezután meg kell határozni a 10 legjobb lövést és ki kell számolni a pontösszeget! Számoljuk meg, hogy mennyi 10-es, 9-es, ... pontértékű lövés volt, s ezt használjuk fel a megoldáshoz (L(I) jelöli az I értékű lövések számát).

Versenyző értékelése:

```
Ciklus J=1-től N-ig
        L(P(J)):=L(P(J))+1
Ciklus vége
```

```
        K:=10 : E:=0 : S:=0
        Ciklus amíg K>0 és S<L
                Ha S+L(K)>L akkor X:=L–S különben X:=L(K)
                        [ X= ennyi K pontértékű lövést kell figyelembe venni ]
                E:=E+X*K : S:=S+X
                K:=K–1
        Ciklus vége
Eljárás vége.
```

Készítsük el a HT–1080Z BASIC programot!

```
 10     CLS
 20     PRINT TAB(20);"LŐTÁBLA ÉRTÉKELÉS" : PRINT
 30     PRINT "MAX. 20 LÖVÉSBŐL A LEGJOBB 10 PONTÉRTÉKE"
 40     PRINT
 50     INPUT "A LÖVÉSEK SZÁMA(1–20)";N
 60     IF N<1 OR N>20 OR N<>INT(N) THEN 50
 70     L=10
 80     DIM P(N),L(L)
 90     FOR I=1 TO N
100             PRINT "AZ";I;". LÖVÉS KOORDINÁTÁI";
110             INPUT X,Y
120             T=SQR(X*X+Y*Y)
130             IF T=0 THEN P(I)=10
                        ELSE P(I)=INT(11–T) : IF P(I)<0 THEN P(I)=0
140     NEXT I
150     REM LÖVÉS SZÁMOLÁS
160     FOR J=1 TO N
170             L (P(J))=L (P(J))+1
180     NEXT J
190     REM PONTSZÁM SZÁMOLÁS
200     K=10 : E=0 : S=0
210     IF K>0 AND S<L THEN 220 ELSE 260
220             IF S+L(K)>L THEN X=L–S ELSE X=L(K)
230             E=E+X*K : S=S+X
240             K=K–1
250     GOTO 210
260     CLS  : PRINT "PONTSZÁMOK:";
270     FOR I=1 TO N : PRINT P(I); : NEXT I
280     PRINT : PRINT "ÖSSZPONTSZÁM:";E
290     STOP
```

• • •

GY.43. "könyvtárnyilvántartás"

Olyan megoldást kell találnunk a feladatra, amely a tárolt adatokat bármilyen szempont szerint sorba tudja rendezni. A rendező adatok nagy száma miatt a megoldásnak lehetőleg el kell kerülnie az egyes elemek konkrét cserélgetését, mivel ez nagyon lassítaná az eredmények elkészítését. A rendező adatokat kazettáról az A$(500,4) mátrixba fogjuk beolvasni, s N lesz a könyvek aktuális száma. Felveszünk egy X(N) vektort, amelyben az adott szempont szerinti rendezettség sorrendjét fogjuk előállítani.

Rendezés:

```
        X(0):=0
        Ciklus I=1-től N-ig
                Az I. beillesztése          ; az első I db már rendezett
        Ciklus vége
Eljárás vége.
```

A beillesztést a rendezett részsorozatba úgy végezzük, hogy az adatok növekvő sorrendbe legyenek rendezve. A V. oszlop szerint rendezünk.

Az I. beillesztése:

```
        P:=0 : J:=X(P)
        Ciklus amíg J>0 és A$(J,V)<A$(I,V)
                P:=J : J:=X(P)
        Ciklus vége
        X(I):=J : X(P):=I
Eljárás vége.
```

A BASIC program az adatokat kazettán a "KÖNYVTÁR" nevű sor után keresi. Itt kell megadni a könyvek aktuális számát, majd soronként egy-egy könyv leírását.

```
10      CLS  : PRINT "KÖNYVTÁR KATALÓGUS KIÍRÁS" : PRINT
20      GOSUB 1000 : REM ADATOK BEOLVASÁSA
30      CLS  : PRINT "SORBARENDEZÉS"
40      PRINT TAB(5);"1. SZERZŐ SZERINT"
50      PRINT TAB(5);"2. CÍM SZERINT"
60      PRINT TAB(5);"3. KIADÓ SZERINT"
70      PRINT TAB(5);"4. MEGJELENÉSI ÉV SZERINT"
80      PRINT : INPUT "VÁLASZTÁS";V
90      IF V<1 OR V>4 OR V<>INT(V) THEN 30
```

```
100   GOSUB 500 : REM SORBARENDEZÉS
110   S1=0 : S2=12 : REM SEGÉDVÁLTOZÓK LAPOZÁSHOZ
120   P=X(0)
130   IF P=0 THEN 200
140         IF S1=S2 THEN S1=0 : GOSUB 2000 : REM VÁRAKOZÁS
150         IF S1=0 THEN CLS : PRINT "SZERZŐ","CÍM","KIADÓ","ÉV" : PRINT
160         PRINT A$(P,1),A$(P,2),A$(P,3),A$(P,4) : S1=S1+1
170         P=X(P)
180   GOTO 130
200   PRINT @15*64,"KELL MÁS RENDEZÉS(I,N)"; : INPUT V$
210   IF V$="I" OR V$="i" THEN 30
220   STOP
500   REM SORBARENDEZÉS
510   X(0)=0 : REM ÜRES A LISTA
520   FOR I=1 TO N
530         P=0 : J=X(P)
540         IF J=0 OR A$(J,V)>=A$(I,V) THEN 570
550               P=J : J=X(P)
560         GOTO 540
570         X(I)=J : X(P)=I
580   NEXT I
590   RETURN
1000  REM ADATOK BEOLVASÁSA
1010  PRINT "ÁLLÍTSD A KAZETTÁT A KÖNYVTÁR ADATAIHOZ!"
1020  GOSUB 2000 : REM VÁRAKOZÁS
1030  INPUT■-1,V$
1040  IF V$<>"KÖNYVTÁR" THEN 1030
1050  INPUT■-1,N
1060  DIM A$(N,4),X(N)
1070  FOR I=1 TO N
1080        INPUT■-1,A$(I,1),A$(I,2),A$(I,3),A$(I,4)
1090  NEXT I
1100  RETURN
2000  REM VÁRAKOZÁS
2010  PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
2020  IF INKEY$=""THEN 2020
2030  RETURN
```

• • •

GY.44. "raktárkészlet"

Az adatokat kazettáról fogjuk beolvasni, K(N) jelöli az áruk kódjait, R(N) a raktáron levő mennyiséget, E(N) az egységárat és M(N) a minimális készletet.

Program:

```
Adatok olvasása kazettáról
A:=0          ; összérték
Ciklus I=1-től N-ig
        A:=A+R(I)*E(I)
        Ha R(I)<M(I) akkor Ki: K(I),R(I),E(I),M(I)
Ciklus vége
Ki: A
```

Program vége.

Az adatkazetta szerkezete:

1. rekord: "KÉSZLET"
2. rekord: adatok száma
3.–..rekord: 1–1 adatsor

A BASIC program:

```
 10   CLS  : PRINT "RAKTÁRKÉSZLET NYILVÁNTARTÁS": PRINT
 20   PRINT "KÉSZLET ALATTI MENNYISÉGEK MEGHATÁROZÁSA"
 30   GOSUB 500 : REM ADATOK BEOLVASÁSA
 40   A=0
 50   S1=0 : S2=12 : REM SEGÉDVÁLTOZÓK KÉPKEZELÉSHEZ
 60   FOR I=1 TO N
 70        A=A+R(I)*E(I)
 80        IF R(I)<M(I) THEN GOSUB 1000 : REM KIÍRÁS
 90   NEXT I
100   PRINT "TELJES KÉSZLET ÉRTÉKE:";A
110   STOP
500   REM ADATOK BEOLVASÁSA
510   PRINT "ÁLLÍTSD A KAZETTÁT A KÉSZLETEK ADATAIHOZ!"
520   GOSUB 2000 : REM VÁRAKOZÁS
530   INPUT■-1,V$
540   IF V$<>"KÉSZLET" THEN 530
550   INPUT■-1,N
560   DIM K(N),R(N),E(N),M(N)
570   FOR I=1 TO N
580        INPUT■-1,K(I),R(I),E(I),M(I)
590   NEXT I
600   RETURN
1000  REM KIÍRÁS
1010  IF S1=S2 THEN S1=0 : GOSUB 2000 : REM VÁRAKOZÁS
1020  IF S1=0 THEN CLS : PRINT "KÓD","KÉSZLET","EGYSÉGÁR",
                 "MIN. MENNY."
1030  PRINT K(I),R(I),E(I),M(I)
1040  S1=S1+1
1050  RETURN
```

```
2000    REM VÁRAKOZÁS
2010    PRINT@15*64,"NYOMJ LE EGY BILLENTYŰT!";
2020    IF INKEY$="" THEN 2020
2030    RETURN
```

• • •

GY.45. "vállalatok költségei"

A megoldáshoz – mint az adatfeldolgozási feladatok többségében – sok adatra van szükségünk. Ezek:

N	–	termékek száma
A(N)	–	A vállalat rendelései
B(N)	–	B vállalat rendelései
F(N)	–	termékek tonnánkénti árai

A program kiszámolja:

A	–	A vállalat által rendelt összmennyiség
B	–	B vállalat által rendelt összmennyiség
KA	–	A vállalat költségei
KB	–	B vállalat költségei
MM	–	legnagyobb rendelt mennyiség
K(N)	–	termékenkénti közös költség

A program 3 fő részből fog állni:

Program:

```
        Adatok beolvasása
        A,B,KA,KB,MM:=0
        Eredmények számítása
        Ha MM=0 akkor Ki:"nincs rendelés"
                 különben Eredmények kiírása
Program vége.
```

Adatok beolvasása:

```
        Be:N        [N>1 és egész]
        Ciklus I=1-től N-ig
                Be: A(I)        [A vállalat rendelése]
                Be: B(I)        [B vállalat rendelése]
                Be: F(I)        [Tonnánkénti ár forintban]
        Ciklus vége
Eljárás vége.
```

Eredmények számítása:

```
        Ciklus I=1-től N-ig
                A:=A+A(I)
                B:=B+B(I)
                KA:=KA+A(I)*F(I)
                KB:=KB+B(I)*F(I)
                Q:=A(I)+B(I)
                K(I):=Q*F(I)
                Ha MM<Q akkor MM:=Q
        Ciklus vége
Eljárás vége.
```

Eredmények kiírása:

```
        Ha A<>0 akkor Ki: KA/A  [ tonnánkénti átlagár
        Ha B<>0 akkor Ki: KB/B  [ tonnánkénti átlagár
        Ki:  KA                 [ összköltség
        Ki:  KB                 [ összköltség
        Ki:  KA+KB              [ közös összköltség
        Ciklus I=1-től N-ig     ; legnagyobb rendelések
                Ki: I,K(I)
                Ha MM=A(I)+B(I) akkor Ki:"max. mennyiség"
        Ciklus vége
Eljárás vége.
```

A BASIC program az algoritmus részeknek megfelelően épül fel. Az adatok beolvasását és az eredmények kiírását – a nagy adatmennyiség miatt – lapokra tagoljuk.

```
  10    CLS  : PRINT "KÖLTSÉG SZÁMÍTÁS" : PRINT
  20    GOSUB 1000 : REM ADATOK BEOLVASÁSA
  30    A=0 : B=0 : KA=0 : KB=0 : MM=0
  40    GOSUB 2000 : REM EREDMÉNYEK SZÁMÍTÁSA
  50    IF MM=0 THEN PRINT "NINCS RENDELÉS" ELSE GOSUB 3000 :
                  REM EREDMÉNYEK KIÍRÁSA
  60    STOP
1000    REM ADATOK BEOLVASÁSA
1010    INPUT "TERMÉKEK SZÁMA(>1)";N : IF N<2 OR N<>INT(N) THEN 1010
1020    DIM A(N),B(N),F(N),K(N)
1030    FOR I=1 TO N
1040          CLS : PRINT I;". TERMÉK ADATAI:" : PRINT
1050          INPUT "'A' VÁLLALAT RENDELÉSE";A(I) : IF A(I)<0 THEN 1050
1060          INPUT "'B' VÁLLALAT RENDELÉSE";B(I) : IF B(I)<0 THEN 1060
1070          INPUT "TONNÁNKÉNTI ÁR";F(I) : IF F(I)<0 THEN 1070
1080    NEXT I
1090    RETURN
2000    REM EREDMÉNYEK SZÁMÍTÁSA
```

```
2010    FOR I=1 TO N
2020        A=A+A(I)
2030        B=B+B(I)
2040        KA=KA+A(I)*F(I)
2050        KB=KB+B(I)*F(I)
2060        Q=A(I)+B(I)
2070        K(I)=Q*F(I)
2080        IF MM<Q THEN MM=Q
2090    NEXT I
2100    RETURN
3000    REM EREDMÉNYEK KIÍRÁSA
3010    CLS
3020    IF A<>0 THEN PRINT "A VÁLLALAT TONNÁNKÉNTI ÁTLAGÁRA:";KA/A
3030    IF B<>0 THEN PRINT "B VÁLLALAT TONNÁNKÉNTI ÁTLAGÁRA:";KB/B
3040    PRINT : PRINT "A VÁLLALAT ÖSSZKÖLTSÉGE:";KA
3050    PRINT "B VÁLLALAT ÖSSZKÖLTSÉGE:";KB
3060    PRINT "KÖZÖS ÖSSZKÖLTSÉG        :";KA+KB
3070    S1=12 : S2=12 : REM LAPOZÁSI SEGÉDVÁLTOZÓK
3080    FOR I=1 TO N
3090        IF S1=S2 THEN 3100 ELSE 3130
3100        S1=0 : PRINT@15*64,"BILLENTYŰ LENYOMÁSRA LAPOZOK";
3110        IF INKEY$="" THEN 3110
3120        CLS : PRINT "SORSZÁM","KÖZÖS KÖLTSÉG","MEGJEGYZÉS"
3130        PRINT I,K(I),
3140        IF MM=A(I)+B(I) THEN PRINT "MAXIMÁLIS MENNYISÉG" ELSE PRINT
3150        S1=S1+1
3160    NEXT I
3170    RETURN
```

• • •

GY.46. "számlák"

Minden számláról (N db) 4 adatot tárolunk: számlaszám (S(N)), előző egyenleg (E(N)), bevétel (B(N)) és kiadás (K(N)). Ki kell számolnunk az összes számlához tartozó új egyenlegeket. Nehezíti a feladat megoldását, hogy egy számlaszám (egymás után) többször is előfordulhat. A számlákat kazettán tároljuk, a kazettás file első rekordja tartalmazza a "SZÁMLA" szöveget, a második pedig a rekordok számát. Ezután következnek a számlák adatai.

```
Program
        Számlák adatainak beolvasása kazettáról
        I:=1
        Ciklus amíg I<=N
            J:=I
```

```
            Ciklus amíg J<=N és S(J)=S(I)
                J:=J+1
            Ciklus vége
            X:=E(I)                    ; előző egyenleg
            Ciklus K=I-től J-1-ig
                X:=X+B(K)-K(K)         ; új egyenleg
            Ciklus vége
            Ki: S(I),X
            Ha X<1000 akkor Ki: "****"
            I:=J                       ; új számla
        Ciklus vége
Program vége.
```

A BASIC program kazettáról addig olvas, amíg meg nem találja a "SZÁMLA" szöveget, majd beolvassa a számlák adatait. Az S vektort a szükségesnél eggyel nagyobb elemszámúra választjuk, így oldjuk meg a túlindexelés elkerülését. S1, S2 a képernyőlapozás szervezésére szolgáló változók.

```
 10   CLS : PRINT "SZÁMLÁK SZÁMÍTÁSA" : PRINT
 20   GOSUB 500 : REM ADATBEOLVASÁS
 30   I=1 : S1=0 : S2=12
 40   IF I>N THEN 180
 50       J=I
 60       IF J>N OR S(J)<>S(I) THEN 90
 70           J=J+1
 80       GOTO 60
 90       X=E(I)
100       FOR K=I TO J-1
110           X=X+B(K)-K(K)
120       NEXT K
130       GOSUB 300 : REM LAPOZÁS
140       PRINT S(I);"SZÁMLA: ";X;"FT.", : S1=S1+1
150       IF X<1000 THEN PRINT "****"ELSE PRINT
160       I=J
170   GOTO 40
180   STOP
300   REM LAPOZÁS
310   IF S1=S2 THEN S1=0 : GOSUB 1000 : REM VÁRAKOZÁS
320   IF S1=0 THEN CLS : PRINT "ÚJ EGYENLEGEK" : PRINT
330   RETURN
500   REM ADATBEOLVASÁS
510   PRINT "ÁLLÍTSD A KAZETTÁT A SZÁMLÁK ADATAIHOZ!"
520   GOSUB 1000 : REM VÁRAKOZÁS
```

```
 530    INPUT■-1,V$
 540    IF V$<>"SZÁMLA" THEN 530
 550    INPUT■-1,N
 560    DIM S(N+1),E(N),B(N),K(N)
 570    FOR I=1 TO N
 580        INPUT■-1,S(I),E(I),B(I),K(I)
 590    NEXT I
 600    RETURN
1000    REM VÁRAKOZÁS
1010    PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
1020    IF INKEY$="" THEN 1020
1030    RETURN
```

• • •

GY.47. "földrengések adatai"

Ebben a feladatban sok adatra (N) van szükségünk, amelyeket kazettáról olvasunk be. A rengés helyét a H$(N), erősségét az E(N), időtartamát az I(N), földrésze sorszámát az F(N) vektorban tároljuk, azt pedig, hogy előre jelezték-e a J(N) vektorban (ha értéke 0, akkor nem, különben igen). Az eredmények számításához felhasználunk három 5 elemű vektort (5 földrész van), amelyekbe a földrészenkénti rengések számát (F0(5)), az előre nem jelzett rengések számát (F1(5)), valamint ezek erősségeinek összegét (F2(5)) számoljuk ki. Ezen kívül megadjuk annak a földrésznek a sorszámát, ahol a legtöbb rengés volt. A kazettán levő file neve "FÖLDRENGÉS" lesz. Ki kell még írni a 0.3 percnél (18 másodperc) tovább tartó földrengések adatait.

Program:

```
        Adatok beolvasása kazettáról
        Ciklus I=1-től N-ig
            F0(F(I)):=F0(F(I))+1
            Ha J(I)=0 akkor  F1(F(I)):=F1(F(I))+1
                             F2(F(I)):=F2(F(I))+E(I)
            Ha I(I)>18 akkor Ki: H$(I),E(I),I(I),F(I),J(I)
        Ciklus vége
        X:=0 : Y:=0              ; maximum kiválasztás
        Ciklus I=1-től 5-ig
            Ha F1(I)>0 akkor Ki: I,F2(I)/F1(I)
            Ha X<F0(I) akkor X:=F0(I) : Y:=I
        Ciklus vége
        Ki: Y,X
Program vége.
```

A BASIC programban az S1,S2 változók szolgálnak a lapozás segítésére, S1 értékét a minden kiírás előtt kötelezően meghívott 1000-es szubrutin növeli, amely számára F$-ban kell megadni a fejlécet.

```
10    CLS : PRINT "FÖLDRENGÉSEK ADATAINAK FELDOLGOZÁSA" : PRINT
20    GOSUB 500 : REM ADATOK BEOLVASÁSA
30    F$="HELY        ERŐSSÉG  IDŐTARTAM  FÖLDRÉSZ  JELEZTÉK-E?"
40    S1=0 : S2=12 : REM SEGÉDVÁLTOZÓK LAPOZÁSHOZ
50    FOR I=1 TO N
60          F0(F(I))=F0(F(I))+1
70          IF J(I)=0 THEN F1(F(I))=F1(F(I))+1 :
                  F2(F(I))=F2(F(I))+E(I)
80          IF I(I)>18 THEN GOSUB 1000 : PRINT H$(I);TAB(15);E(I);
                  TAB(28);I(I);TAB(40);F(I);TAB(50);J(I)
90    NEXT I
100   REM MAXIMUM KIVÁLASZTÁS
110   F$="FÖLDRÉSZ        ÁTLAGOS RENGÉS-ERŐSSÉG"
120   S1=S2 : REM ÚJ LAP KEZDŐDJÖN
130   X=0 : Y=0
140   FOR I=1 TO 5
150         IF F1(I)>0 THEN GOSUB 1000 : PRINT I,F2(I)/F1(I)
160         IF X<F0(I) THEN X=F0(I) : Y=I
170   NEXT I
180   S1=S2 : REM ÚJ LAP KEZDŐDJÖN
190   F$="LEGTÖBB FÖLDRENGÉS"
200   GOSUB 1000 : PRINT "FÖLDRÉSZ:";Y,"SZÁM=";X
210   STOP
500   REM ADATBEOLVASÁS
510   PRINT "ÁLLÍTSD A KAZETTÁT A FÖLDRENGÉSEK ADATAIHOZ!"
520   GOSUB 2000 : REM VÁRAKOZÁS
530   INPUT■-1,V$
540   IF V$<>"FÖLDRENGÉS" THEN 530
550   INPUT■-1,N
560   DIM H$(N),E(N),I(N),F(N),J(N)
570   FOR I=1 TO N
580         INPUT■-1,H$(I),E(I),I(I),F(I),J(I)
590   NEXT I
600   RETURN
1000  REM ALAPOZÁS A KÉPERNYŐN
1010  IF S1=S2 THEN S1=0 : GOSUB 2000 : REM VÁRAKOZÁS
1020  IF S1=0 THEN CLS : PRINT F$ : PRINT
1030  S1=S1+1
1040  RETURN
2000  REM VÁRAKOZÁS
2010  PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
2020  IF INKEY$="" THEN 2020
2030  RETURN
```

• • •

GY.48. "szállítási ár"

Szükségünk van a szállítások darabszámára (N), a szállítási távolságokra (T(N)) és mennyiségekre (M(N)). A megoldásban kiszámoljuk, hogy az adott szállításért milyen kedvezmény jár (E), majd megadjuk az árat. Az adatokat ebben a feladatban a billentyűzetről olvassuk be. Kétféle szállítástávolság esetén adhatunk engedményt, s ehhez jöhet még a mennyiség miatti árcsökkentés. 1 tonnát 1 kilométerre a vállalat X forintért szállít.

Program:

```
        Be: X,N                 [X>0, N>0 és egész]
        Ciklus I=1-től N-ig
                Be: T(I),M(I)   [T(I),M(I)>0]
        Ciklus vége
        A:=0
        Ciklus I=1-től N-ig
                E:=0
                Y:=T(I)*M(I)            ; teljes ár
                Ha T(I)>100 akkor E:=.1
                Ha 10<=T(I) és T(I)<=19 akkor E:=.05
                Ha M(I)>2 akkor E:=E+.03
                Ki: I,Y*(1-E)
                A=A+Y*(1-E)
        Ciklus vége
        Ki: A*X
Program vége.
```

A BASIC program:

```
 10     CLS  : PRINT "SZÁLLÍTÁSI ÁR SZÁMÍTÁS" : PRINT
 20     INPUT "SZÁLLÍTÁSI ÁR";X : IF X<=0 THEN 20
 30     INPUT "SZÁLLÍTÁSOK SZÁMA";N : IF N<1 OR N<>INT(N) THEN 30
 40     DIM T(N),M(N)
 50     FOR I=1 TO N
 60             PRINT I;". TÁVOLSÁG ÉS MENNYISÉG"; : INPUT T(I),M(I)
 70             IF T(I)<=0 OR M(I)<=0 THEN 60
 80     NEXT I
 90     S1=0 : S2=10 : REM SEGÉDVÁLTOZÓK LAPOZÁSHOZ
100     FOR I=1 TO N
110             E=0
120             Y=T(I)*M(I)
130             IF T(I)>100 THEN E=.1
140             IF 10<=T(I) AND T(I)<=19 THEN E=.05
150             IF M(I)>2 THEN E=E+.03
160             IF S1<S2 THEN 190 : REM NEM KELL MÉG LAPOZNI
170             PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
```

```
180         IF INKEY$="" THEN 180 ELSE S1=0
190         PRINT I;". SZÁLLÍTÁS ÁRA=";Y*(1-E) : S1=S1+1
200         A=A+Y*(1-E)
210     NEXT I
220     PRINT "ÖSSZBEVÉTEL=";A*X
230     STOP
```

• • •

GY.49. "influenza-nyilvántartás"

A nyilvántartott adatokat kazettán tároljuk, a kazetta első rekordja az "INFLUENZA" szöveg, a második az adatok száma, utána pedig következnek a felvett adatok. Az M(N) vektor tartalmazza a megyék sorszámát, a H(N) a helység kódját, a B(N) pedig a megbetegedések számát. A megoldásban D(20) jelöli az egyes megyékben megbetegedettek számát (Budapest kapja a 20. sorszámot).

Program:

```
        Adatok beolvasása kazettáról
        X:=0 : P:=0 : Q:=0
        Ciklus I=1-től N-ig
                D(M(I)):=D(M(I))+B(I)
                X:=X+B(I)
                Ha B(I)>P akkor P:=B(I) : Q:=I
        Ciklus vége
        M:=0 : B:=0
        Ciklus I=1-től 20-ig
                Ha D(I)>M akkor B:=D(I) : M:=I
                Ki: I,D(I)/X*100
        Ciklus vége
        Ki: M,B
        Ki: H(Q),M(Q),B(Q)
Program vége.
```

A BASIC program egy sorba 4 megye adatait írja ki.

```
10      CLS  : PRINT "INFLUENZA NYILVÁNTARTÁS" : PRINT
20      GOSUB 500 : REM ADATBEOLVASÁS
30      X=0 : P=0 : Q=0
40      FOR I=1 TO N
50          D(M(I))=D(M(I))+B(I)
60          X=X+B(I)
70          IF B(I)>P THEN P=B(I) : Q=I
```

```
 80    NEXT I
 90    M=0 : B=0
100    CLS  : PRINT : "MEGYÉNKÉNTI BETEGEK SZÁMA:"
110    FOR I=1 TO 20
120          IF D(I)>M THEN B=D(I) : M=I
130          PRINT I;":";D(I)/X*100,
140    NEXT I
150    PRINT : PRINT "LEGTÖBB MEGBETEGEDÉS"
160    PRINT M;". MEGYE:";B;"DB."
170    PRINT H(Q);"VÁROS(";M(Q);". MEGYE):";B(Q);"DB."
180    STOP
500    REM ADATBEOLVASÁS
510    PRINT "ÁLLÍTSD A KAZETTÁT A MEGBETEGEDÉSEK ADATAIHOZ!"
520    PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
530    IF INKEY$="" THEN 530
540    INPUT■-1,V$
550    IF V$<>"INFLUENZA" THEN 540
560    INPUT■-1,N
570    DIM M(N),H(N),B(N),D(20)
580    FOR I=1 TO N
590          INPUT■-1,M(I),H(I),B(I)
600    NEXT I
610    RETURN
```

• • •

GY.50. "telefonkönyv"

Egy olyan keresési módszert szeretnénk bemutatni, amelyhez nincs szükség rendezettségre, ezért először beszéljünk az adatok tárolásáról. Az ötletet a gyakorlatból vesszük, egy ilyen "telefonkönyv" úgy nézhet ki, hogy minden betűnek van egy lap, amelyen az azzal a betűvel kezdődő nevek találhatók tetszőleges sorrendben.
Csináljunk egy ilyen rendszert, vegyünk fel egy 26 elemű vektort (ennyi betűnk van), s feleltessük meg az elemeket a betűknek. Ha egy elem értéke 0, az jelentse azt, hogy ilyen kezdőbetűs név nincsen (pl. Q,X,Y); ha nem 0, akkor adja meg az első ilyen kezdőbetűs név sorszámát egy táblázatban.

A táblázat álljon négy oszlopból:

név	–	N$(N)
cím	–	C$(N)
telefonszám	–	T$(N)
A következő ilyen kezdőbetűs név sorszáma a táblázatban (ha 0, akkor nincs több)	–	S(N)

Az előbbi vektor neve legyen E!

Program:

```
        Adatok beolvasása kazettáról
        Ciklus
                Be: N$
                N$ keresése (K=0, ha nincs, egyébként a sorszáma)
                Ha K>0 akkor Ki: N$(K),C$(K),T$(K)
        amíg van keresendő adat
        Ciklus vége
Program vége.
```

A keresést azzal kezdjük, hogy megnézzük: van-e ilyen kezdőbetűs név, s ha van, akkor végignézzük ezeket.

N$ keresése:

```
        K:=N$ kezdőbetűjének sorszáma
        K:=E(K)                 ; 0 vagy az első sorszáma
        Ciklus amíg K>0 és N$(K)<>N$
                K:=S(K)         ; 0 vagy a következő sorszáma
        Ciklus vége
Eljárás vége.
```

A BASIC program kazettán egy "TELEFONKÖNYV" tartalmú sorról ismeri fel a telefonkönyv adatait.

```
 10   CLS  : PRINT "TELEFONKÖNYV KEZELÉS" : PRINT : CLEAR 1000
 20   GOSUB 1000 : REM ADATOK BEOLVASÁSA KAZETTÁRÓL
 30   REM KÉRDEZÉS
 40   CLS  : INPUT "A KERESETT NÉV";N$ : IF N$="" THEN 40
 50   GOSUB 500 : REM KERESÉS
 60   IF K=0 THEN PRINT "NINCS ILYEN EMBER" : GOTO 90
 70   PRINT "CÍME              :";C$(K)
 80   PRINT "TELEFONSZÁMA :";T$(K)
 90   PRINT
100   INPUT "VAN MÉG KERESENDŐ NÉV(I,N)";V$
110   IF V$="I" OR V$="i" THEN 30
120   STOP
500   REM KERESÉS
510   K=ASC(LEFT$(N$,1))-64
520   K=E(K)
530   IF K=0 OR N$(K)=N$ THEN 560
540         K=S(K)
550   GOTO 530
```

```
  560  RETURN
 1000  REM ADATOK BEOLVASÁSA
 1010  PRINT "ÁLLÍTSD A KAZETTÁT A TELEFONKÖNYVHÖZ!"
 1020  PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
 1030  IF INKEY$="" THEN 1030
 1040  INPUT■-1,V$
 1050  IF V$<>"TELEFONKÖNYV" THEN 1040
 1060  INPUT■-1,N
 1070  DIM N$(N),C$(N),T$(N),S(N),E(26)
 1080  FOR I=1 TO N
 1090        INPUT■-1,N$(I),C$(I),T$(I),S(I)
 1100  NEXT I
 1110  FOR I=1 TO 26
 1120        INPUT■-1,E(I)
 1130  NEXT I
11440  RETURN
```

• • •

GY.51. "növényhatározó"

Jelölések a növényhatározóban és az ismeretlen virág jellemzői:

N	–	virágleírások száma
N$(N)	–	a virág neve
S$(N)	–	a virág színe
E$(N)	–	nyílásának évszaka
M1(N),M2(N)	–	legkisebb és legnagyobb magassága
S$	–	az ismeretlen színe
H$	–	az ismeretlen szedési hónapja
M	–	az ismeretlen konkrét magassága
E$	–	az ismeretlen szedési évszaka

A BASIC programban el kell végeznünk a hónapok és az évszakok egymáshoz rendelését: vegyünk fel egy 4 soros táblázatot, minden sor első eleme legyen az évszak neve és mögötte sorban a hozzá tartozó hónapok. A növényhatározót kazettáról olvassuk be. Ezután az ismeretlen virágok tulajdonságait olvassuk be (mindaddig, amíg a felhasználó ad újabb leírásokat), s megkeressük, hogy tárolt tudásunk alapján ez milyen virág lehet (előfordulhat, hogy a leírás több virágra is illik).

Program:

```
        Adatok beolvasása kazettáról
        Hónap–évszak megfeleltetés
        Ciklus
                Be: S$,H$,M          [H$ hónap név,M>0]
                E$=H$ hónapot tartalmazó évszak
                Ciklus I=1-től N-ig
                        Ha S$=S$(I) és M1(I)<=M és M<=M2(I) és E$=E$(I)
                                akkor Ki: N$(I)
                Ciklus vége
        amíg Van meghatározandó virág
        Ciklus vége
Program vége.
```

A BASIC programban feltételezzük, hogy az összes megfelelő virág neve elfér egyetlen képernyő-képen. A kazettán a növényhatározó első sorában a "VIRÁGOK" szöveg található, a másodikban a virágok száma, s utána következnek az egyes virágok leírásai.

```
  10    CLS  : PRINT "NÖVÉNYHATÁROZÓ" : PRINT
  20    GOSUB 2000 : REM ADATOK BEOLVASÁSA
  30    GOSUB 1000 : REM HÓNAP–ÉVSZAK MEGFELELTETÉS
  40    REM KEZDŐDIK A VIRÁGHATÁROZÁS
  50          CLS
  60          INPUT "A VIRÁG SZÍNE";S$
  70          INPUT "SZEDÉSI HÓNAPJA";H$
  80          I=0
  90          IF I<=3 AND M$(1,I)<>H$ AND M$(2,I)<>H$ AND M$(3,I)<>H$
                      THEN 100 ELSE 120
 100                I=I+1
 110          GOTO 90
 120          IF I<=3 THEN E$=M$(0,I) ELSE 70
 130          INPUT "MAGASSÁGA";M : IF M<=0 THEN 130
 140          FOR I=1 TO N
 150                IF S$=S$(I) AND M1(I)<=M AND M<=M2(I) AND E$=E$(I)
                      THEN PRINT N$(I),
 160          NEXT I
 170    PRINT : INPUT "VAN MÉG MEGHATÁROZANDÓ VIRÁG(I,N)";V$
 180    IF V$="I" OR V$="i" THEN 40
 190    STOP
1000    REM ÉVSZAK–HÓNAP MEGFELELTETÉS
```

```
1010    DATA "TÉL", "DECEMBER", "JANUÁR", "FEBRUÁR"
1020    DATA "TAVASZ", "MÁRCIUS", "ÁPRILIS", "MÁJUS"
1030    DATA "NYÁR", "JÚNIUS", "JÚLIUS", "AUGUSZTUS"
1040    DATA "ŐSZ", "SZEPTEMBER", "OKTÓBER", "NOVEMBER"
1050    RESTORE
1060    DIM M$(3,3)
1070    FOR I=0 TO 3
1080        FOR J=0 TO 3
1090            READ M$(J,I)
1100        NEXT J
1110    NEXT I
1120    RETURN
2000    REM ADATBEOLVASÁS
2010    PRINT "ÁLLÍTSD A KAZETTÁT A VIRÁGOK LEÍRÁSÁHOZ!"
2020    PRINT @15*64,"NYOMJ LE EGY BILLENTYŰT!";
2030    IF INKEY$="" THEN 2030
2040    INPUT■-1,V$
2050    IF V$<>"VIRÁGOK" THEN 2040
2060    INPUT■-1,N
2070    DIM N$(N),S$(N),E$(N),M1(N),M2(N)
2080    FOR I=1 TO N
2090        INPUT■-1,N$(I),S$(I),E$(I),M1(I),M2(I)
2100    NEXT I
2110    RETURN
```

• • •

GY.52. "keresztrejtvény"

Az egyszerű keresztrejtvényekben olyan helyeken találhatók számok, ahol pl. a sor fölött fekete kocka van, alatta pedig fehér, illetve az oszlop előtt fekete kocka van, mögötte pedig fehér. Ha a 0. sorba és oszlopba fekete kockákat képzelünk, továbbá az N+1. sorba és az M+1. oszlopba szintén, akkor ez a definíciónk a széleken is helyes lesz (A(N,M) a keresztrejtvény). Az X és az Y vektorokban fogjuk tárolni a kiírandó vízszintes és függőleges sorszámokat.

Program:

```
A keresztrejtvény beolvasása
Ciklus I=1-től N-ig
    Ciklus J=1-től M-ig
        K:=vízszintesen kell-e írni számot?
        L:=függőlegesen kell-e írni számot?
```

```
                    Ha K vagy L akkor S:=S+1 : A(I,J):=S
                    Ha K akkor L1:=L1+1 : X(L1):=S
                    Ha L akkor L2:=L2+1 : Y(L2):=S
               Ciklus vége
          Ciklus vége
          Eredmények kiírása
Program vége.
```

A BASIC programban a vízszintes és függőleges sorokhoz tartozó sorszámokat elkülönítve írjuk ki. A táblázatban –1 jelzi, ha ott fekete kocka van, 0 ha üres és a sorszám, ha oda valamilyen számot írunk. A keresztrejtvénybe maximum 100 számot írhatunk. A program a keresztrejtvény kirajzolását nem tartalmazza.

```
 10    CLS  : PRINT "KERESZTREJTVÉNY-KÉSZÍTÉS" : PRINT
 20    GOSUB 500 : REM ADATOK BEOLVASÁSA
 30    FOR I=1 TO N
 40         FOR J=1 TO M
 50              IF J=1 THEN K=A(I,J+1)=0 : GOTO 70
 60              IF J<N THEN K=A(I,J–1)=–1 AND A(I,J+1)=0 ELSE K=0
 70              IF I=1 THEN L=A(I+1,J)=0 : GOTO 90
 80              IF I<N THEN L=A(I–1,J)=–1 AND A(I+1,J)=0 ELSE L=0
 90              IF K OR L THEN S=S+1 : A(I,J)=S
100              IF K THEN L1=L1+1 : IF L1<SM THEN X(L1)=S
110              IF L THEN L2=L2+1 : IF L2<SM THEN Y(L2)=S
120         NEXT J
130    NEXT I
140    GOSUB 800 : REM EREDMÉNYEK KIÍRÁSA
150    STOP
500    REM ADATOK BEOLVASÁSA
510    INPUT "A KERESZTREJTVÉNY MÉRETEI";N,M
520    IF N<3 OR M<3 OR N<>INT(N) OR M<>INT(M) THEN 510
530    SM=100 : DIM A(N,M),X(SM),Y(SM)
540    FOR I=1 TO N : FOR J=1 TO M
550         PRINT I;J;" HELY FEKETE(I/BÁRMI)";
560         INPUT V$ : A(I,J)=V$="I"
570    NEXT J,I
580    S=0 : L1=0 : L2=0 : REM KEZDŐÉRTÉKEK
590    RETURN
800    REM EREDMÉNYEK KIÍRÁSA
```

```
810     CLS
820     IF L1=0 THEN 840 ELSE PRINT "VÍZSZINTES SOROK:"
830     FOR I=1 TO L1 : PRINT X(I); : NEXT I
840     IF L2=0 THEN 860 ELSE PRINT : PRINT "FÜGGŐLEGES SOROK:"
850     FOR I=1 TO L2 : PRINT Y(I); : NEXT I
860     RETURN
```

• • •

GY.53. "élkiemelés"

Feladatunk olyan pontok meghatározása (kirajzolása), amelyekben a fényesség gyorsan változik. Így a különböző területek kontúrjait fogjuk megkapni. Egy pont élhez tartozik, ha vízszintes vagy függőleges irányban levő szomszédainak intenzitáskülönbsége egy rögzített értéknél (K) jobban változik. Ebből a szempontból a kép szélein levő pontokat külön kell vizsgálnunk. A program felhasználja a kép méretét (N,M), valamint a mért fényességeket (X(N,M)). A kép méretének legalább 3*3-asnak kell lenni.

Program

```
        Be: N,M,K        [N,M>2 és egész, K>0]
        Ciklus I=1-től N-ig
                Ciklus J=1-től M-ig
                        Be: X(I,J)        [X(I,J)>=0]
                Ciklus vége
        Ciklus vége
        Ciklus I=2-től N–1-ig
                Ha ABS(X(I–1,1)–X(I+1,1))>K akkor Rajzol(I,1)
                HA ABS(X(I–1,M)–X(I+1,M))>K akkor Rajzol(I,M)
                Ciklus J=2-től M–1-ig
                        Ha ABS(X(I–1,J)–X(I+1,J))>K vagy ABS(X(I,J–1)–
                                X(I,J+1))>K akkor Rajzol (I,J)
                Ciklus vége
        Ciklus vége
        Ciklus J=2-től M–1-ig
                Ha ABS(X(1,J–1)–X(1,J+1))>K akkor Rajzol(1,J)
                Ha ABS(X(N,J–1)–X(N,J+1))>K akkor Rajzol(N,J)
        Ciklus vége
Program vége.
```

A BASIC program:

```
 10     CLS  : PRINT "ÉLKIEMELÉS" : PRINT
 20     INPUT "KÉP MÉRET(LEGALÁBB 3x3), INT. KÜLÖNBSÉG";N,M,K
 30     IF N<3 OR M<3 OR N<>INT(N) OR M<>INT(M) OR K<=0 THEN 20
 40     DIM X(N,M)
 50     FOR I=1 TO N : FOR J=1 TO M
 60          PRINT I;J;"FÉNYESSÉGE (0–1)";
 70          INPUT X(I,J) : IF X(I,J)<0 OR X(I,J)>1 THEN 60
 80     NEXT J,I
 90     CLS
100     FOR I=2 TO N–1
110          IF ABS(X(I–1,1)–X(I+1,1))>K THEN SET(I,1)
120          IF ABS(X(I–1,M)–X(I+1,M))>K THEN SET(I,M)
130          FOR J=2 TO M–1
140               IF ABS(X(I–1,J)–X(I+1,J))>K OR ABS(X(I,J–1)–
                       X(I,J+1))>K THEN SET(I,J)
150          NEXT J
160     NEXT I
170     FOR J=1 TO M–1
180          IF ABS(X(1,J–1)–X(1,J+1))>K THEN SET(1,J)
190          IF ABS(X(N,J–1)–X(N,J+1))>K THEN SET(N,J)
200     NEXT J
210     STOP
```

• • •

GY.54. "szénhidrogén"

A szénhidrogén szerkezetét egy mátrixban (A(N,N)) adjuk meg, ennek (I,J) helyén 1-es van, ha az I-edik és a J-edik szénatom között van kötés, 0 pedig akkor, ha nincs. A definícióból látható, hogy a mátrix szimmetrikus, így például elég csak a főátló feletti részt beolvasni.

Program:

```
        Be: N          [N>1 és egész]
        Ciklus I=1-től N–1-ig
                Ciklus J=I+1-től N-ig
                        Be: A(I,J)          [A(I,J) 0 vagy 1 lehet]
                        A(J,I):=A(I,J)
                Ciklus vége
        Ciklus vége
        Szénatomok lehagyása
        Eredményírás
Program vége.
```

El kell hagynunk minden olyan szénatomot, amely nem tartozik gyűrűbe, illetve nem köt össze gyűrűt. Első lépésben keressünk egy olyan atomot, amelynek csak egyetlen másikkal van kötése és ezt hagyjuk el a mátrixból. Ezt a lépést ismételjük addig, amíg találunk ilyen atomot. Ha már nincs ilyen atom, akkor csak azok az atomok maradhattak meg, amelyek gyűrűben vannak, vagy pedig gyűrűket kötnek össze.

Szénatomok lehagyása:

```
        Ciklus amíg van 1 kötésű szénatom
                Az 1 kötésű szénatom lehagyása
        Ciklus vége
Eljárás vége.
```

A BASIC programban a megmaradt kötés-mátrixot írjuk ki eredményül, feltesszük, hogy a teljes mátrix elfér a képernyőn.

```
 10     CLS  : PRINT "SZÉNHIDROGÉN ÁTALAKÍTÁS" : PRINT
 20     INPUT "SZÉNATOMOK SZÁMA";N : IF N<2 OR N<>INT(N) THEN 20
 30     DIM A(N,N)
 40     FOR I=1 TO N-1
 50          FOR J=I+1 TO N
 60               PRINT I;J;"ATOMOK KÖTÉSE(0 VAGY 1)";
 70               INPUT A(I,J) : IF A(I,J)<>0 AND A(I,J)<>1 THEN 60
 80               A(J,I)=A(I,J)
 90          NEXT J
100     NEXT I
110     REM AMÍG VAN 1 KÖTÉSŰ SZÉNATOM
120          I=1
130          S=0 : FOR J=1 TO N : S=S+A(I,J) : NEXT J
140          IF S=1 AND I<N THEN 150 ELSE 170
150               I=I+1
160          GOTO 130
170          IF S<>1 THEN 210
180          REM SZÉNATOM ELHAGYÁSA
190          FOR J=1 TO N : A(I,J)=0 : A(J,I)=0 : NEXT J
200     GOTO 120
210     FOR I=1 TO N : FOR J=1 TO M
220          PRINT A(I,J);
230     NEXT J,I
240     STOP
```

• • •

GY.55. "rádióadók"

Határozzuk meg a feladat matematikai modelljét: adott két intervallumsorozat, döntsük el, hogy van-e közös részük. Akkor van, ha van olyan intervallumpár, ahol az első intervallumban van a második kezdete, vagy a másodikban az első kezdete. N1,N2 jelöli a két adó adássorozatainak számát, A(N1,2),B(N2,2) pedig ezek kezdő és végidőit. Az időket az éjfél óta eltelt percekkel adjuk meg, csak egy nap adatait vizsgáljuk.

Program:

```
Be: N1,N2                    [N1,N2>0 és egész]
Ciklus I=1-től N1-ig
      Be: A(I,1),A(I,2)      [0<A(I,1)<A(I,2)]
Ciklus vége
Ciklus I=1-től N2-ig
      Be: B(I,1),B(I,2)      [0<B(I,1)<B(I,2)]
Ciklus vége
I:=1 : J:=1
Ciklus amíg I<=N1 és J<=N2 és A(I,),B(J,) nem átfedőek
      Ha A(I,2)<B(J,1) akkor I:=I+1
            különben Ha A(I,1)>B(J,2) akkor J:=J+1
Ciklus vége
Ha I=N1+1 vagy J=N2+1 akkor Ki: "Lehetnek azonosak"
      különben Ki: "Nem lehetnek azonosak"
Program vége.
```

A megoldás BASIC programja:

```
 10   CLS  : PRINT "RÁDIÓADÓK VIZSGÁLATA" : PRINT
 20   INPUT "1. ADÓ ADÁSAI SZÁMA";N1
 30   IF N1<0 OR N1<>INT(N1) THEN 20 ELSE DIM A(N1,2)
 40   INPUT "2. ADÓ ADÁSAI SZÁMA";N2
 50   IF N2<0 OR N2<>INT(N2) THEN 40 ELSE DIM B(N2,2)
 60   PRINT "1. ADÓ:"
 70   FOR I=1 TO N1
 80        PRINT I;". ADÁS KEZDETE ÉS VÉGE";
 90        INPUT A(I,1),A(I,2)
100        IF A(I,1)<0 OR A(I,1)>=A(I,2) THEN 80
```

```
110     NEXT I
120     PRINT "2. ADÓ:"
130     FOR I=1 TO N2
140         PRINT I;". ADÁS KEZDETE ÉS VÉGE";
150         INPUT B(I,1),B(I,2)
160         IF B(I,1)<0 OR B(I,1)>=B(I,2) THEN 140
170     NEXT I
180     I=1 : J=1
190     IF I<=N1 AND J<=N2 THEN 200 ELSE 240
200     IF A(I,2)<B(J,1) OR B(J,2)<A(I,1) THEN 210 ELSE 240
210     IF A(I,2)<B(J,1) THEN I=I+1: GOTO 230
220     IF A(I,1)<B(J,2) THEN J=J+1
230     GOTO 190
240     IF I=N1+1 OR J=N2+1 THEN PRINT "A KÉT ADÓ LEHET AZONOS" ELSE
        PRINT "A KÉT ADÓ NEM LEHET AZONOS"
230     STOP
```

• • •

GY.56. "méréskiegészítés"

A mérési eredményeket az A(N) vektorban tároljuk, az első és az utolsó mérés eredménye nagyobb, mint 0 (azaz valódi mérés). A vektorban meg kell keresni a nullákból álló szakaszokat. Ezt a következőképpen csináljuk: megyünk előre, amíg nem találunk 0-t (vagy a vektor végére nem érünk), azt a helyet megjegyezzük, majd elmegyünk a következő 0-tól különböző elemig. Ezután meg kell határozni a két érték különbségét és a növekményt el kell osztani egyenlően a köztük levő szakaszokra.

Program:

```
Be: N           [N>2 és egész]
Ciklus I=1-től N-ig
    Be: A(I)    [A(I)>=0 , A(1),A(N)>0]
Ciklus vége
I:=1
Ciklus amíg I<=N
    K:=I
    Ciklus amíg K<=N és A(K)>0
        K:=K+1
    Ciklus vége        ; 0-szakasz első eleme
```

```
                K:=K–1
                Ha K<N akkor
                        V:=K+1
                        Ciklus amíg A(V)=0
                                V:=V+1
                        Ciklus vége        ; 0-szakasz utolsó utáni elem
                        A 0 értékek helyettesítése
                Elágazás vége
                I:=V+1
        Ciklus vége
Program vége.
```

A helyettesítést úgy végezzük, hogy a 0-szakasz két végpontja közötti különbséget egyenlő növekménnyel osztjuk el a köztük levő 0 értékű helyek között.

A 0 értékek helyettesítése:

```
        A:=(A(V)–A(K))/(V–K)
        Ciklus J=K+1-től V–1-ig
                A(J):=A(J–1)+A
        Ciklus vége
Eljárás vége.
```

A BASIC programban a vektort N+1 eleműnek definiáljuk, így elkerüljük a 120-as sor esetleges túlindexelését. Az eredményül kapott adatokat szorosan egymás mögé írjuk ki, s feltételezzük, hogy az összes elfér egyetlen képernyő-lapon.

```
 10     CLS  : PRINT "MÉRÉSI ADATOK KIEGÉSZÍTÉSE" : PRINT
 20     INPUT "MÉRÉSEK SZÁMA";N : IF N<3 OR N<>INT(N) THEN 20
 30     DIM A(N+1)
 40     FOR I=1 TO N
 50          PRINT I;". MÉRÉS"; : INPUT A(I)
 60          IF A(I)<0 OR (I=1 OR I=N) AND A(I)=0 THEN 50
 70     NEXT I
 80     REM KIKAPCSOLT SZAKASZ KERESÉS
 90     I=1
100     IF I>N THEN 250
110          K=I
120          IF K>N OR A(K)=0 THEN 150
130              K=K+1
140          GOTO 120
150          K=K–1 : REM MEGTALÁLTUK A KEZDETET
```

```
160         IF K=N THEN 230
170             V=K+1
180             IF A(V)>0 THEN 210
190                 V=V+1
200             GOTO 180
210             REM MEGTALÁLTUK A VÉGÉT
220             GOSUB 500 : REM HELYETTESÍTÉS
230         I=V+1
240     GOTO 100
250     CLS : PRINT "A MÉRÉSI EREDMÉNYEK:" : PRINT
260     FOR I=1 TO N
270         PRINT A(I);
280     NEXT I
290     STOP
500     REM HELYETTESÍTÉS
510     A=(A(V)-A(K))/(V-K)
520     FOR J=K+1 TO V-1
530         A(J)=A(J-1)+A
540     NEXT J
550     RETURN
```

• • •

GY.57. "buszjegy"

A buszjegy lyukasztásra érdemes részét jelöljük egy 3x3-as négyzettel és a könnyebb kezelhetőség miatt – a gyakorlatnak megfelelően – számozzuk meg a kis négyzeteket 1-től 9-ig a következő módon:

```
123
456
789
```

A lyukasztást pedig jelöljük úgy, hogy a kilyukasztott szám helyébe 0-t írunk, pl:

```
103
056
780
```

A megoldás során a feladat egyetlen komoly nehézsége a fordított (középső oszlopra szimmetrikus) lyukasztás figyelembe vétele és kiszűrése; pl. a

```
020        020
056  és a  450  kombinációk közül csak az egyik szerepeljen a listán
789        789
```

Jelölje A,B,C az aktuális jegykombináció lyukasztási helyeit, K$(9) a kirajzolandó kombinációt!
Csak olyan kombinációkat vizsgálunk, amelyben A<B<C. Egyszerűen megállapítható, hogy melyek azok a lyukkombinációk, amelyek tükörképei ebben a sorrendben már előfordultak. Ezek a következők:

"A" buszjegy jobb oldalán van (3–x–y,6–x–y)
"A" középen, "B" jobb oldalon van (2–3–x,2–6–x,5–6–x)
"A" és "B" szimmetrikusan helyezkedik el, "C" jobb oldalon van
(1–3–6,1–3–9,2–5–6,2–5–9,2–8–9,4–6–9,5–8–9)

A feladat lényegében egyetlen ciklusban megoldható. Először az A,B,C változók kapják a lehetséges lyukasztásra kerülő számhármas értékét. Ezután megvizsgáljuk, hogy az adott számhármas azonos-e egy korábban kinyomtatott lyukasztás tükörképével. Ha nem azonos, akkor kinyomtatjuk az A,B,C-nek megfelelő esetet.

Program:

```
        Kezdőértékek megadása
        Ciklus A=1-től 7-ig
          Ha A helyes akkor
            Ciklus B=A+1-től 8-ig
              Ha A és B helyes akkor
                Ciklus C=B+1-től 9-ig
                  Ha A és B és C helyes akkor Nyomtatás
                Ciklus vége
              Elágazás vége
            Ciklus vége
          Elágazás vége
        Ciklus vége
Program vége.
```

Mindezek alapján lássuk a BASIC programot! CHR$(48) jelenti a 0-t (fontos, hogy kiíráskor minden szám egy helyet foglaljon el a képen), CHR$(48+...) pedig a számjegyeket.

```
10    CLS : PRINT "BUSZJEGY-LYUKASZTÁSOK!"
20    X0=1 : Y0=0 : X=X0 : Y=Y0-4 : REM KOORDINÁTÁK A RAJZOLÁSHOZ
30    DIM K$(9)
40    FOR I=1 TO 9 : K$(I)=CHR$(48+I) : NEXT I
```

```
 50   FOR A=1 TO 7
 60     IF A=3 OR A=6 THEN 140
 70       FOR B=A+1 TO 8
 80         IF A=2 AND (B=3 OR B=6) OR A=5 AND B=6 THEN 130
 90           FOR C=B+1 TO 9
100             IF (A=1 AND B=3 OR A=2 AND B=5) AND (C=6 OR C=9)
                      OR ( (A=2 OR A=5) AND B=8 OR A=4 AND B=6)
                      AND C=9 THEN 120
110             GOSUB 200 : REM KIÍRÁS
120           NEXT C
130       NEXT B
140   NEXT A
150   STOP
200   REM JEGY KIRAJZOLÁS
210   K$(A)=CHR$(48) : K$(B)=CHR$(48) : K$(C)=CHR$(48)
220   IF S=16 THEN X=X+4 : Y=Y0 : S=0 ELSE Y=Y+4
230   S=S+1
240   PRINT @64*X+Y,K$(1);K$(2);K$(3);
250   PRINT @64*(X+1)+Y,K$(4);K$(5);K$(6);
260   PRINT @64*(X+2)+Y,K$(7);K$(8);K$(9);
270   K$(A)=CHR$(48+A) : K$(B)=CHR$(48+B) : K$(C)=CHR$(48+C)
280   RETURN
```

• • •

GY.58. "periódikusság"

A mért jelsorozatunk N jelből áll. A periódushosszt M1 és M2 között keressük. A kívánt pontosság legyen P. A vett jelsorozatot (X(N)) akkor mondjuk periódikusnak, ha van olyan intervallum-hosszúság, amely alatti mérések eltérése az első intervallum méréseitől egyenként kisebb, mint P.

Program:

```
Be: N,M1,M2,P        [N>1 és egész, M1<M2, P>0]
Ciklus I=1-től N-ig
      Be: X(I)
Ciklus vége
J:=M1
Ciklus
      Periódushossz-keresés
```

```
                [J a feltételezett periódushossz]
            Periódushossz-ellenőrzés
                [X=IGAZ, ha nem jó ez a periódus]
            Ha X akkor J:=J+1
        amíg J<=M2 és X
        Ciklus vége
        Ha X akkor Ki: "Nincs periódus" különben Ki: J
Program vége.
```

Periódushossz keresésekor úgy kell eljárnunk, hogy az aktuális hosszt növeljük a lépésközzel mindaddig, amíg az 1. és a fenti távolságra következő mérés különbsége nem lesz kisebb a várt pontosságnál.

Periódushossz-keresés:

```
        Ciklus amíg J<=M2 és |A(1)-A(J)|>P
            J:=J+1
        Ciklus vége
Eljárás vége.
```

Ellenőrzéskor meg kell állapítanunk, hogy ha az 1. és a J. közötti eltérés még megfelelő, akkor bármely két ilyen távolságú elem között is az-e!

Periódushossz-ellenőrzés:

```
        I:=J+1
        Ciklus amíg I<=N és A(I) nem tér el a periódustól
            I:=I+1
        Ciklus vége
        X:= I<=N        ; IGAZ, ha nem jó a periódus
Eljárás vége.
```

Egy mérés akkor nem tér el a periódustól, ha a neki megfelelő elemtől az első J db között az eltérése elég kicsi.

A BASIC program:

```
10      CLS  : PRINT "PERIÓDIKUSSÁG VIZSGÁLAT" : PRINT
20      INPUT "MÉRÉSEK SZÁMA(>1)";N : IF N<2 OR N<>INT(N) THEN 20
30      INPUT "MIN, MAX PERIÓDUS(<MÉR. SZÁM/2)";M1,M2
40      IF M1<0 OR M2<0 OR M1>M2 OR M2>N/2 THEN 30
50      INPUT "PONTOSSÁG";P : IF P<=0 THEN 50
60      DIM X(N)
```

```
 70    FOR I=1 TO N
 80        PRINT I;". MÉRÉS";
 90        INPUT X(I)
100    NEXT I
110    J=M1
120        GOSUB 300 : REM PERIÓDUSHOSSZ MEGHATÁROZÁS(J)
130        GOSUB 400 : REM PERIÓDUSHOSSZ ELLENŐRZÉS(X)
140        IF X THEN J=J+1
150    IF J<=M2 AND X THEN 120
160    IF X THEN PRINT "NINCS PERIÓDUS" ELSE
                 PRINT "A PERIÓDUSHOSSZ=";J
170    STOP
300    REM PERIÓDUSHOSSZ MEGHATÁROZÁS
310    IF J<M2 AND ABS(A(1)-A(J))>P THEN 320 ELSE 340
320        J=J+1
330    GOTO 310
340    RETURN
400    REM PERIÓDUSHOSSZ ELLENŐRZÉS
410    I=J+1
420    IF I>N THEN 470
430    K=I-J*(I/J) : IF K=0 THEN K=J : REM AZ ÖSSZEHASONLÍTANDÓ
440    IF ABS(A(I)-A(K))>P THEN 470
450        I=I+1
460    GOTO 420
470    X=(I<=N) : REM IGAZ, HA NEM JÓ A PERIÓDUS
480    RETURN
```

• • •

GY.59. "állások betöltése"

Jelöljük N-nel a jelentkezők (és állások) számát! Az A(N,N) mátrixban tároljuk, hogy melyik jelentkező melyik állásokra jelentkezett [A(I,J)=1, ha az I jelentkező a J-edik munkát vállalná, illetve 0, ha nem]. Olyan programot kell írni, amely eldönti, hogy lehet-e minden sorból kiválasztani úgy egy 1-es értékű elemet, hogy ezek az összes oszlopot is lefedjék (azaz minden oszlopból is csak egyet választhatunk).
A feladat megoldására a backtrack algoritmust alkalmazzuk. V(I) fogja jelölni, hogy az I-edik oszlopban hányadik sorban levő egyest választottuk ki.

```
Program:
        Adatok beolvasása
        I:=1 : V(I):=0
        Ciklus amíg I>=1 és I<=N
                Ha Van szabad állás(I) akkor I:=I+1 : V(I):=0
                        különben I:=I-1
        Ciklus vége
        Ha I>N akkor Eredményírás
                        különben Ki: „A feladat nem oldható meg"
Program vége.
```

A szabad állás keresését úgy végezzük, hogy az I. emberhez megpróbáljuk a következő állást hozzárendelni, s mindaddig vesszük a következőt, amíg az nem jó vagy pedig már nincs következő állás.

Van szabad állás(I):

```
        Ciklus
                V(I):=V(I)+1
        amíg V(I)<=N és (A(I,V(I))=0 vagy Az állás foglalt)
        Ciklus vége
        Van szabad állás:= V(I)<=N
Eljárás vége.
```

Egy állás akkor nem foglalt, ha az aktuális ember előtti sorszámúakhoz nem ezt az állást rendeltük.

Az állás foglalt:

```
        J:=1
        Ciklus amíg J<I és V(J)<>V(I)
                J:=J+1
        Ciklus vége
        Az állás foglalt:= J<I
Eljárás vége.
```

A megoldás HT–1080Z programjában az eljárásokat behelyettesítettük a megfelelő helyre. Ez feleslegessé tette a logikai értékadások használatát és így közvetlenül beírhattuk a folytatást (230,250 sorok).

```
 10    CLS : PRINT "ÁLLÁSOK BETŐLTÉSE" : PRINT
 20    INPUT "ÁLLÁSOK SZÁMA";N : IF N<1 THEN 20
 30    DIM A(N,N),V(N)
 40    FOR I=1 TO N
 50        FOR J=1 TO N
 60            PRINT I;". EMBER JELENTKEZETT A(Z)";J;
                   ".ÁLLÁSRA(I,BÁRMI)";
 70            INPUT V$
 80            IF V$="I" THEN A(I,J)=1 ELSE A(I,J)=0
 90        NEXT J
100    NEXT I
110    REM BACKTRACK
120    I=1 : V(I)=0
130    IF I<1 OR I>N THEN 300
140        REM VAN SZABAD ÁLLÁS?
150        V(I):=V(I)+1
160        IF V(I)>N THEN 240 : REM ELFOGYTAK
170        IF A(I,V(I))=0 THEN 150 : REM NEM VÁLLALJA
180        REM FOGLALT AZ ÁLLÁS?
190        J=1
200        IF J>=I OR V(J)=V(I) THEN 230
210            J=J+1
220        GOTO 200
230        IF J<I THEN 150 : REM FOGLALT
240        REM V(I)<=N, HA VAN SZABAD ÁLLÁS
250        IF V(I)<=N THEN I=I+1 : V(I)=0 ELSE I=I–1
260    GOTO 130
300    REM I<1, HA AZ ÁLLÁSOK NEM TÖLTHETŐK BE
310    IF I<1 THEN PRINT "A FELADAT NEM OLDHATÓ MEG" : STOP
320    PRINT "A MEGOLDÁS:"
330    FOR I=1 TO N
340    PRINT I;"–";V(I),
350    NEXT I
360    STOP
```

• • •

GY.60. "csillagsűrűsödések"

Pontosítsuk, mit tudunk a feladatról: adottak M db pont (csillag) koordinátái egy tetszőleges koordináta-rendszerben, a sűrűsödést meghatározó csillagszám (N) és a sűrűsödésbe tartozás távolság kritériuma (K). Helyezzük el a koordinátákat az X,Y,Z vektorokban! Az eredmény a sűrűsödések száma (S) lesz.

Program:

```
        Az adatok beolvasása
        S:= 0          ; S = a sűrűsödések száma
        A sűrűsödések számának meghatározása
        Ki: S
Program vége.
```

A sűrűsödések számának meghatározásában sorra kell vennünk azokat a csillagokat, amelyeket még nem soroltunk be egyetlen csillagcsoportba sem, s megállapítani, hogy az őket tartalmazó csoport megfelel-e a sűrűsödés definíciójának. Használunk egy V vektort, V(i) legyen IGAZ, ha az i-edik csillagot még sehova sem soroltuk be! Kezdetben a V vektor összes eleme IGAZ értékű lesz, majd a vizsgálatok során egyeseket HAMIS értékűre állítunk.

A sűrűsödések számának meghatározása:

```
        Ciklus I=1-től M-ig
                V(I):=IGAZ          ; még egyet sem vizsgáltunk
        Ciklus vége
        Ciklus I=1-től M-ig
                        [ S= a sűrűsödések száma az I. csillag vizsgálata előtt ]
                Ha V(I) akkor Az I. csillagot tartalmazó sűrűsödés vizsgálata
                        [ S= a sűrűsödések száma az I. csillag vizsgálata után ]
        Ciklus vége
Eljárás vége.
```

A következő feladatunk azon csoport meghatározása, amelybe egy adott csillag tartozik, s ha ennek elemszáma legalább N, akkor a sűrűsödések számát növelhetjük eggyel. Az aktuális csoport elemszámát jelöljük A-val!

Az I. csillagot tartalmazó sűrűsödés vizsgálata:

```
        A:=0
        Az I. csillagot tartalmazó csoport vizsgálata
        Ha A >=N akkor S:=S+1
Eljárás vége.
```

Elérkeztünk egy olyan ponthoz, ahol komolyan figyelembe kell vennünk a sűrűsödés definícióját. Milyen csillagok tartoznak egy csoportba? Az elsőnek választott csillag, a hozzá "közel" levő csillagok, az azokhoz "közel" levő csilagok stb. Ezért a kezdő csillagból kiindulva addig kell közeli csillagokat keresni, amíg az összes addig megtalált csillagot meg nem vizsgáltuk. Ehhez szükségünk van az aktuális csoportba tartozó csillagokra, tároljuk ezek sorszámait a C(M) vektorban! Ez a vektor egy különleges adatszerkezetet testesít meg, számok sorozatát fogjuk benne tárolni, a sorozatból elvenni mindig csak az elejéről lehet, hozzátenni új számokat pedig csak a végére (ezt az adatszerkezetet nevezik SOR-nak). Azt, hogy hol van a sor utolsó eleme, az A változó tartalmazza.

Vezessünk be egy új változót (E), amely a sor első elemére (amit éppen vizsgálunk) mutat, a vizsgált csillag csoportbeli sorszámát adja meg!

Az I. csillagot tartalmazó csoport vizsgálata:

```
        Az I. csillag felvétele a csoportba
                [ A=1, a csoportnak az I. csillag az első eleme ]
        E:=1
        Ciklus amíg E <=A
                A C(E). csillaghoz közeli csillagok vizsgálata (A növelése)
                E:=E+1
        Ciklus vége
Eljárás vége.
```

A csillag felvételénél jelezni kell, hogy ezt a csillagot a továbbiakban ne vizsgáljuk, az aktuális csoportba tartozó csillagok számát eggyel növelni kell, majd a C vektorba be kell jegyezni ezt a csillagot (egy új elem felvétele a sorba).

Az I. csillag felvétele a csoportba:

```
        V(I):=HAMIS
        A:=A+1
        C(A):=I
Eljárás vége.
```

Az adott csillaghoz közeli csillagokat kell még megkeresni, először határozzuk meg, hogy melyik csillagok között keressük ezeket! Biztosan nincs szükségünk az aktuálisan vizsgáltat (amiből a csoport készítését elkezdtük) megelőzőkre, a későbbiek közül pedig csak azok érdekesek, amelyeket eddig még nem vizsgáltunk.

A C(E)-edik csillaghoz közeli csillagok vizsgálata:

```
        Ciklus J=I+1-től M-ig
                Ha V(J) akkor T:= A J-edik és a C(E)-edik csillag távolsága
                        Ha T < K akkor A J-edik csillag felvétele a csoportba
                Elágazás vége
        Ciklus vége
Eljárás vége.
```

A BASIC programban logikai értékek tárolásáról kell gondoskodnunk, a HT–1080Z-n az igaz értéket a –1, a hamisat pedig a 0 jelenti.

```
 10     CLS  : PRINT "CSILLAGSŰRŰSÖDÉSEK SZÁMÁNAK MEGHATÁROZÁSA"
 20     PRINT
 30     INPUT "CSILLAGOK SZÁMA";M : IF M<1 OR M<>INT(M) THEN 30
 40     DIM X(M),Y(M),Z(M),V(M),C(M)
 50     PRINT @6*64,"CSILLAGSZÁM ÉS TÁVOLSÁG(FÉNYÉVBEN)";
 60     INPUT N,K : IF N<1 OR N>M OR K<0 OR N<>INT(N) THEN 50
 70     CLS
 80     REM A KOORDINÁTÁK BEOLVASÁSA
 90     FOR I=1 TO M
100             PRINT I;". CSILLAG KOORDINÁTÁI";
110             INPUT X(I),Y(I),Z(I)
120     NEXT I
130     S=0
140     GOSUB 400 : REM A SŰRŰSÖDÉSEK SZÁMÁNAK MEGHATÁROZÁSA
150     CLS
160     PRINT "A CSILLAGSŰRŰSÖDÉSEK SZÁMA:";S
170     STOP
```

```
400   REM A SŰRŰSÖDÉSEK SZÁMÁNAK (S) MEGHATÁROZÁSA
410   FOR I=1 TO M
420       V(I)=-1 : REM EZT A CSILLAGOT MÉG NEM VIZSGÁLTUK
430   NEXT I
440   FOR I=-1 TO M
450       IF V(I) THEN GOSUB 500 : REM AZ I. CSILLAGOT TARTALMAZÓ
              SŰRŰSÖDÉS VIZSGÁLATA
460   NEXT I
470   RETURN : REM S= A SŰRŰSÖDÉSEK SZÁMA
500   REM AZ I. CSILLAGOT TARTALMAZÓ SŰRŰSÖDÉS VIZSGÁLATA
520   A=0
530   L=I : GOSUB 600 : REM AZ I. CSILLAG FELVÉTELE A CSOPORTBA
535   E=1 : REM ELŐSZÖR A KEZDŐ CSILLAGHOZ KÖZELIEK
540   IF E<=A THEN 550 ELSE 570
550       GOSUB 700 : REM A C(E). CSILLAGHOZ KÖZELIEK
560       E=E+1 : GOTO 540
570   REM A= A CSOPORT CSILLAGJAINAK SZÁMA
580   IF A>=N THEN S=S+1
590   RETURN
600   REM AZ L. CSILLAG FELVÉTELE A CSOPORTBA (L)
610   V(L)=0
620   A=A+1 : C(A)=L
630   RETURN
700   REM A C(E). CSILLAGHOZ KÖZELI CSILLAGOK VIZSGÁLATA
710   IF I=M THEN RETURN : REM UTOLSÓ CSILLAGNÁL TARTOTTUNK
720   FOR J=I+1 TO M
730       IF V(J) THEN 740 ELSE 760
740       T=SQR((X(J)-X(C(E)))↑2+(Y(J)-Y(C(E)))↑2+(Z(J)-Z(C(E)))↑2)
750       IF T<=K THEN L=J : GOSUB 600 : REM J. CSILLAG FELVÉTELE
760   NEXT J
770   RETURN: REM A C(E). CSILLAGGAL VÉGEZTÜNK
```

Államigazgatási Számítógépes Szolgálat

Intézményünk szellemi és gépi erőforrásaival az országos hatáskörű szervek, vállalatok és egyéb intézmények számítástechnikai igényei kielégítésére vállalkozik.

Eszközbázisunk jellemzői: összesen 3 Mbyte operatív tár, 23 db 100 Mbyte-os mágneslemez-, 13 mágnesszalag-meghajtó, 2 hálózatvezérlő front-end, hálózati hardvereszközök, 50 szinkron, 27 aszinkron, továbbá 20 emulátoros terminál csatlakozik a rendszerhez. Az adatátvitel kapcsolt vagy bérelt telefonvonalakon, postai adatátviteli hálózaton keresztül történik. A hatékony működtetést a GCOS operációs rendszer és a TSS alrendszerek, segédprogramok, alacsony és magas szintű nyelvek, tesztprogramok, funkcionális és matematikai-alkalmazási programcsomagok, speciális programozási nyelvek biztosítják.

Foglalkozunk nagyméretű (országos) adattárak (adatbázisok, adatbankok) kialakításával az ÁSZSZ vagy más számítóközpontok gépi eszközeire. Vállalkozunk komplex alkalmazói (adatfeldolgozási) rendszerek tervezésére, kivitelezésére, üzemszerű futtatására, továbbfejlesztésére. Felhasználóink részére biztosítjuk, hogy saját készítésű programrendszereiket az ÁSZSZ gépére adaptálják – ehhez munkatársaink minden segítséget megadnak. Távadatfeldolgozási lehetőséget biztosítunk felhasználóink számára.

Az ÁSZSZ-ben kialakított rendszerek széles körben hasznosíthatóak és igény esetén többek által hozzáférhetőek. Az államigazgatási munka sokrétűségéből és egymáshoz kapcsolódó jellegéből következik, hogy az adatokat, rendszereket több irányúan is fel lehet használni. Az adatbázis szemléletű alapnyilvántartások általában a hálózatokon keresztül párbeszédes módon is elérhetőek a helyi kötegelt feldolgozási lehetőség mellett.

A rendszerek információtartalmának érzékeltetésére – és az esetleges csatlakozási, felhasználási igények felkeltésére – hasonló jellegű adattárak kidolgozását ajánlva felsorolunk néhány rendszert, amelyek az ÁSZSZ gépi eszközbázisára települnek:

- *Országos Ingatlannyilvántartási Rendszer*
- *Állami Népességnyilvántartó Rendszer*
- *Agrokémiai Információs és Irányítási Rendszer*
- *Egységes Országos Számítógépes Jogi Információs Rendszer*
- *Földmérési és Térképészeti Adatbázis*
- *Családi pótlék Nyilvántartási és Ütköztetési Rendszer*
- *Komplex Országos Morbiditási Vizsgálat*
- *Társadalombiztosítási Statisztikai Információs Rendszer*
- *Erdészeti és Faipari Információs és Irányítási Rendszer*
- *Környezetvédelmi Információs Rendszer*
- *Műszaki Információs Rendszer*
- *Fővárosi Közmű Adattár*
- *Országos Izotóp Nyilvántartási és Forgalmazási Rendszer*
- *Országos Műszerügyi Nyilvántartás*
- *Kristályfizikai Adatbázis*

GYULA

- *Nukleáris anyag Nyilvántartás*
- *Számítógépes Nyövényi Génbank*
- *SZOT Számítógépes Üdültetési Rendszer*
- *Nyugdíjmegállapító Rendszer*
- *Számítógépes Klinikai Információs Rendszer*
- *Magyar Nyelv Gyakorisági Szótárának Számítógépes feldolgozásai*
- *Társadalmi Beilleszkedési Zavarok Elemzése*
- *Célzott Egészségügyi Lakosságszűrés Számítógépes támogatása stb.*

A személyi számítógépek alkalmazása módot ad az információs rendszerek felhasználási körének bővítésére, különösen az adatelőállítás területén. Ezért kidolgoztuk a Honeywell Bull nagyszámítógép és a Commodore–64 kisgép csatlakoztatásához szükséges hardvert és szoftvert. Az adatelőállítási folyamatok egyrészének gépesítése és a lekérdezés többirányú lehetősége elérhető közelségbe hozza az országos információs hálózatok mindennapi használatát.

Tízéves fennállásunk során szerzett gyakorlatunk, számítógépes tapasztalataink, magas szintű képzettséggel rendelkező szakembereink kvalifikált munkájára támaszkodva az alábbi szolgáltatásokat ajánljuk jelenlegi és új megrendelőink számára:

- *országos projektek eszközhátterének biztosítása*
- *adatátviteli hálózati karbantartás, hibavizsgálási és -elhárítási munka*
- *mágneses adathordozók tisztítása*
- *mágnesszalagos adatrögzítés*
- *új számítóközpontok termelésének megszervezése és beindítása*
- *oktatási szolgáltatás mikrogéphasználattal (MØ8X, Commodore stb.)*
- *interaktív programfejlesztési lehetőség*
- *adattárak létrehozása és karbantartása, eredmények kinyomtatása és expediálása*
- *kész programrendszerek futtatása*
- *általános célú és speciális rendszerek, osztott adatbázisok kidolgozása vállalatok és intézmények számára Commodore–64 mikrogépen és Honeywell – C–64 hálózaton.*

Az alkalmazások széles skáláját segítő szoftverkészlet áll rendelkezésünkre. Az információ-rendszerek kidolgozásához különböző numerikus és szöveges adatbázis-kezelő rendszerekkel értékelő, tervező, elemző és optimalizáló programcsomagokkal rendelkezünk.

Reméljük, hogy Önt is hamarosan üdvözölhetjük a tízéves ÁSZSZ felhasználói, megrendelői körében!

1. **Dusza Árpád:** A BASIC programozási nyelv
Országos Pedagógiai Intézet, 1981. (sokszorosítás)

2. **Némethy Katalin–Veszprémi Anna–Zsakó László:** ABC80 BASIC elemei
Fővárosi Pedagógiai Intézet, 1981.

3. **Zsakó László:** Elemi programozási feladatok
ELTE TTK Számítástechnikai Tanszék ABC80-füzetek, 1982.

4. **Zsakó László:** Keresési eljárások számítógéppel
ELTE TTK Számítástechnikai Tanszék ABC80-füzetek, 1982.

5. **Donald Alcock:** Ismerd meg a BASIC nyelvet!
Műszaki Könyvkiadó, 1983.

6. ELTE TTK Számítástechnikai Tanszék alkotó munkaközössége: Kézikönyv a személyi számítógépes tanártovábbképzési tanfolyamhoz
A Fővárosi Pedagógiai Intézet számítástechnikai sorozata II., 1983.

7. **Gárdos Éva–Lakatos Piroska:** BASIC példatár
KLTE TTK jegyzet, 1983.

8. **dr. Poronyi Gábor:** 33 BASIC program
Baranya megyei Pedagógiai Intézet, 1983.

9. **Sándor Antal:** Speciális feladatok az ABC80 mikroszámítógépre
ELTE TTK Számítástechnikai Tanszék ABC80-füzetek, 1983.

10. **Zsakó László:** Néhány érdekes gyakorlati probléma számítógépes megoldása
ELTE TTK Számítástechnikai Tanszék ABC-füzetek, 1983.

11. **Dusza Árpád:** Iskolaszámítógép (Kézirat 1984.)

12. ELTE TTK Számítástechnikai Tanszék alkotó munkaközössége: A HT–1080Z iskolaszámítógép programozása
A Fővárosi Pedagógiai Intézet, a Tudományszervezési és Informatikai Intézet számítástechnikai sorozata III., 1985.

13. **Harmathy Zoltán–Helfenbein Henrik–Kőhegyi János–Zsakó László:** Ismerd meg a BASIC nyelvjárásait! (HT–1080Z, ABC80, ZX81)
Műszaki Könyvkiadó, 1984.

14. **Horváth László:** Numerikus matematikai feladatok
ELTE TTK Számítástechnikai Tanszék ABC-füzetek, 1984.

15. Mikroszámítógép-programok I–II.
Országos Oktatástechnikai Központ (Szerk.: Szűcs Pál)

16. **Szlávi Péter–Zsakó László:** A programozás ABC-je = az ABC programozása
ELTE TTK Számítástechnikai Tanszék ABC80-füzetek, 1984.

17. **Ury László:** Commodore 64, I–II.
SZAMALK, 1984.

18. **Veszprémi Anna:** Számítógépes rendezési módszerek
ELTE TTK Számítástechnikai Tanszék ABC80-füzetek, 1984.

19. Az Ipari Informatikai Intézet kiadványai (ZX Spectrum, Commodore)

FOLYÓIRATOK (CIKKSOROZATOK)

1. Középiskolai Matematikai Lapok: Számítástechnika Pontverseny
2. Számítástechnika: KÖMAL mintamegoldás
   Programozási forgácsok
3. Ötlet: BITLET számítástechnikai melléklet, Sorvezető

A kiadvány a Fővárosi Pedagógiai Intézet, a Tudományszervezési és Informatikai Intézet, valamint az INTERPRESS Kiadó és Nyomda Vállalat, Budapest gondozásában készült.

A kiadásért felel dr. Pollák Miklós, az INTERPRESS Kiadó és Nyomda Vállalat, Budapest, igazgatója.

Megjelent 16,5 (A/5) ív terjedelemben, Helvetica betűtípusból.

Eng. száma:

Készült a Szolnoki Nyomda Vállalat nyomdaüzemében
Felelős vezető: Gombkötő Béla igazgató

Szeret ?

játszva töprengeni, szórakozni ?

Szereti ?

a rejtvényeket ?

Szeretné ?

próbára tenni számítástechnikai ismereteit ?

KERESSE a

Azonos játékok,

Azonos feladatok...

... különböző

személyi számítógépekre!

c. könyvet!

amely versenyfeladatokat tartalmaz

az M08X, a HT1080Z, a Commodore 64

és a Sinclair Spectrum számítógépekre!

ÁRA: 66 Ft