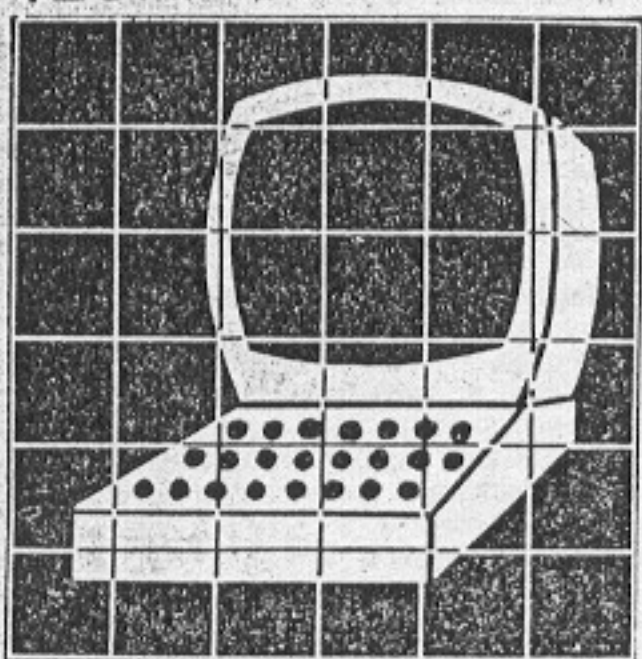




MC-PROGRAMOK

Az itt közölt 1. program lehetővé teszi azt, hogy a „lehető legcsúnyább” programlistából áttekinthető, a program lo-

gikáját kiemelő listát készítsünk akár a képernyőre, akár nyomtatón.

Ha beírjuk 1. és 2. programlistát és futtatjuk azt, akkor a floppylemezen tárolt bármely C-4 számítógépen frt programról tagolt, szép kiírást nyerünk, amint azt a 4. listánk mutatja. Minthogy az 1. és 2. részből álló program begépelése igen sok „rabszolgamunkát” igényel, a következő programkészítési menetet javasoljuk.

A 2. listán látható DATA sorokat készítsük el a számítógéppel.

Írjuk be a 3. programot. Ezt futtatva először elkészíti a huszonegy DATA sort és ezeket kiírja a képernyőre. Amikor ez a lista megjelenik, mozgassuk a kurzort az első (1000-es) sorszáma és a RETURN billentyű ismételt lenyomásával vegyük fel azokat a programba (ezt a részt lehetne úgynevezett kváziparancs üzemmód segítségével is végrehajtani). Ezután RUN 2050 segítségével futtassuk tovább a programot. A kapott lista ne tévesszen meg senkit. Minden egyes sorát újra RETURN-nal le kell zárni, mert

ettől lesz a programsor tartalma tényleg az, amit a listán látunk. Ha ezt a körülmenyesnek látszó programönfejlesztő menetet végigjártasszuk, akkor már nyugodtan beírhatjuk az 1. programlistát, hiszen a folytatásként közölt (2. program) már elkészült, igaz most még 100-tól kezdődő sorszámmal. Ha végeztünk az 1. lista beírásával, akkor kiadhatjuk a RENUMBER parancsot.

Ha az így elkészített programot el-

indítjuk, az megkérdezi, hogy nyomtatóra vagy képernyőre készítse-e el a listát.

Amennyiben megadjuk a kilistázandó program nevét, úgy a gép a lemezzől byte-onként beolvassa azt, és a szükséges tagolások, bekezdések, LET értékesítő utasítások beírásával elkészíti az áttekinthető, a program szerkezetét bemutató listát.

Dr. Sz. Lukács János

```

10 DIMB$(255)
11 FORI=128TO255:READB$(I):NEXT
12 INPUT"NYOMTATO, KEPERNYO N/K";A$
13 P=8: IFA$="K" THENP=0:GOTO30
14 IFA$="N" THENPR=1:OPEN4,4,7:GOTO30
15 GOTO12
16 PRINT#3:IFPR=0 THENRETURN
17 PRINT#4:RETURN
18 O=3
19 PRINT#O,SPC(H)S$;:IFO=4 THEN21
20 IFPR=1 THEN:O=4:GOTO19
21 IFR THENR=0:GOTO16
22 IFY3=1 THENY3=0:K=K+2
23 RETURN
24 J=K:H=0:GOSUB18:K=J:S$="":RETURN
25 GET#1,A$:X=ASC(A$+CHR$(O))
26 IF(64ANDST)=0 THENRETURN
27 GOTO65
28 GOSUB25:X1=X:GOSUB25
29 X=X1+256*X:RETURN
30 PRINT:OPEN3,3
31 INPUT"PR.NEV:";N$:GOSUB16
32 OPEN2,8,15:OPEN1,8,2,N$+"",P,R"
33 GOSUB64:GOSUB28
34 PRINT"□";
35 GOSUB28:IFX=0 THEN27
36 GOSUB28:
37 S$=RIGHT$(" "+STR$(X),5):E=1
38 H=0:GOSUB18:S$="":Y2=0
39 GOSUB25:T$=" LET "
40 IFX=32 AND Y2=0 THEN39
41 IFE=1 THENE=0:IFX<127ANDX>65 THENS$=S$+T$
42 IFX=131 THENY2=2:GOTO44
43 IFX=143 THENY2=1:GOTO44:ELSE 45
44 S$=S$+" "+B$(X)+" ":A$=""
45 IF LEN(S$)=0 AND X=58 THEN39
46 IFX=255 THENS$=S$+"π":GOTO39
47 IFY6ANDY1=0ANDX=44 THENK=K-2
48 IFX=58ANDY2=2 THENY2=0
49 IFK<0 THENK=0:S$=S$+"!!HIBA??"
50 IFX<>58 THEN54
51 IFY2<>0 THEN54
52 S$=S$+A$:A$="":R=1:H=K:GOSUB18
53 GOSUB22: S$="":Y6=0:E=1
54 IFX=58ANDY2=0ANDY4=0 THENY4=1
55 IFX=0 THENR=1:Y6=0:H=K:GOSUB18:GOTO35
56 IFX=34 THENY2=NOT Y2
57 IFX<128ORY2 THEN S$=S$+A$:GOTO39
58 IFY2 THEN39
59 IFX>127 THENS$=S$+" "+B$(X)+" "
60 IFX2=100 THENS$=S$+B$(X)
61 IF(X=129ORX=235) THENY3=1
62 IF(X=130ORX=236) THENK=K-2:Y6=1
63 GOTO39
64 INPUT#2,A$,B$:IFA$="00" THENRETURN
65 FORI=1TO8:CLOSEI:NEXT:END

```

1. programlista.

```

66 DATA END, FOR, NEXT, DATA, INPUT#
67 DATA INPUT, DIM, READ, LET, GOTO
68 DATA RUN, IF, RESTORE, GOSUB, RETURN
69 DATA REM, STOP, ON, WAIT, LOAD
70 DATA SAVE, VERIFY, DEF, POKE, PRINT#
71 DATA PRINT, CONT, LIST, CLR, CMD
72 DATA SYS, OPEN, CLOSE, GET, NEW
73 DATA TAB(, TO, FN, SPC(, THEN
74 DATA NOT, STEP, +, -, *
75 DATA /, ↑, AND, OR, >
76 DATA =, <, SGN, INT, ABS
77 DATA USR, FRE, POS, SQR, RND
78 DATA LOG, EXP, COS, SIN, TAN
79 DATA ATN, PEEK, LEN, STR$, VAL
80 DATA ASC, CHR$, LEFT$, RIGHT$, MID$
81 DATA GO, RGR, RCLR, RLUM, JOY
82 DATA RDOT, DEC, HEX$, ERR$, INSTR
83 DATA ELSE, RESUME, TRAP, TRON, TROFF
84 DATA SOUND, VOL, AUTO, PUDEF, GRAPHIC
85 DATA PAINT, CHAR, BOX, CIRCLE, GSHAPE
86 DATA SSHAPE, DRAW, LOCATE, COLOR
87 DATA SCNCLR, SCALE, HELP, DO, LOOP, EXIT
88 DATA DIRECTORY, DSAVE, DLOAD, HEADER
89 DATA SCRATCH, COLLECT, COPY, RENAME
90 DATA BACKUP, DELETE, RENUMBER, KEY
91 DATA MONITOR, USING, UNTIL, WHILE, , π

```

2. programlista.

```

1022 DATA ,π
2000 GRAPHIC CLR: FORI=1 TO 21
2010 PRINT1000+I;"DATA 1,2,3,4,5,6"
2020 NEXT
2030 LIST-1021:END: MAJD SOK RETURN !!
2050 C=4103:FORI=0TO20:FORK=0TO5
2060 POKE C,128+I*6+K:C=C+2
2070 NEXT:C=C+6:NEXT
2080 LIST-1999:REM SOK RETURN!!!
2090 DELETE 2000-

```

3. programlista: lemezre felvéve.

```

1022 data ,π
2000 graphic clr :
      for i = 1 to 21
2010   print 1000 + i;"data 1,2,3,4,5,6"
2020 next
2030 list - 1021:
      end :
      let majdsok return !!
2050 let c = 4103:
      for i = 0 to 20:
        for k = 0 to 5
2060   poke c,128 + i * 6 + k:
        let c = c + 2
2070   next :
        let c = c + 6:
      next
2080 list - 1999:
      rem sok return!!!
2090 delete 2000 -

```

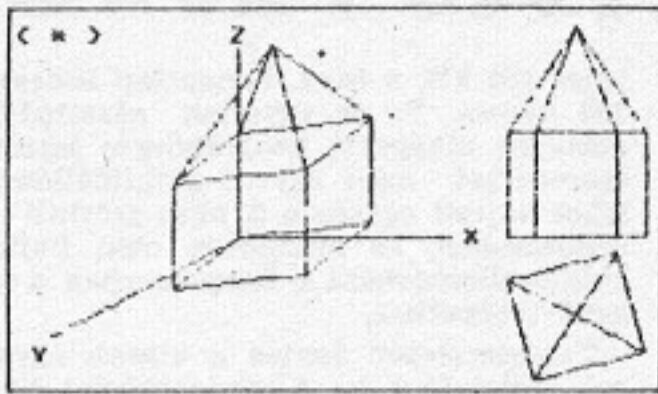
4. programlista: nyomtatási kép.

VETÜLETI ÁBRÁZOLÁS: AXONOMETRIA

A műszaki rajz oktatása során lépten-nyomon előforduló feladat, hogy vetületi képekből rekonstruáljuk a tárgy axonometrikus képét, vagy éppen fordítva: egy tárgyról készítsük el annak vetületi képeit.

Az itt közölt program 870-920 soraiba kell beírni a test csúcspontjainak és éleinek számát. A valamilyen logikai sorrend szerint megszámozott csúcsok x, y, z koordinátáit találhatjuk a 880-890 sorokban. A 910-920 sorokba pedig a csúcspontok összekötési sorrendjét kell beírni. Ezekből az adatokból a program elkészíti a test csúcspontjainak mátrixát.

A 40-200 sorok megrajzolják a test felül- és oldalnézeteit, valamint drótvázmodelljének axonometrikus képét. Ebből az alapállapotból kiindulva a RETURN lenyomásával juthatunk a menühöz.



Ez választási lehetőséget ad arra nézve, hogy az ábrázolt testet forgatni vagy csúsztatni kívánjuk-e a tengelyek körül, illetve mentén.

A forgatás eredményeként a csúcspontok új helyzetbe kerülnek, a pont térbeli

kordinátáit a 390-580 programsorok közötti rész számítja ki.

A csúcspontokba mutató vektor kiszámítását láthatjuk például a 420. sorban.

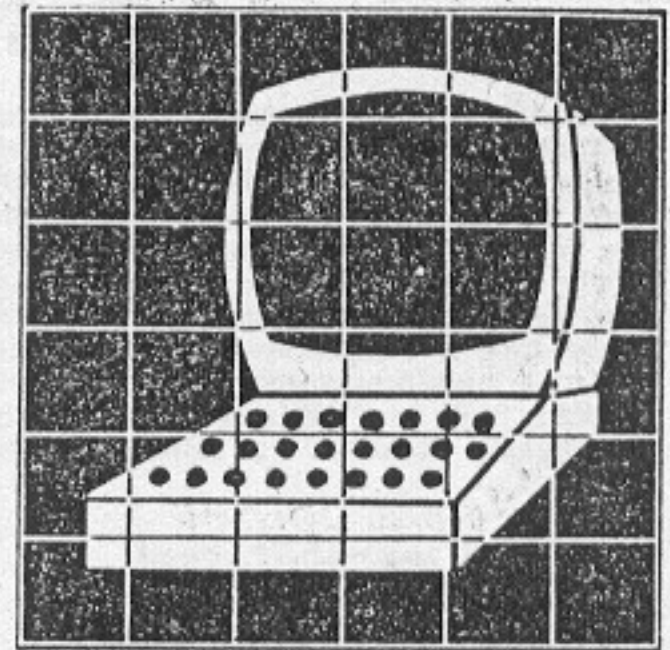
Az alapállapotot a 2. menüpont segítségével állíthatjuk vissza. Ilyenkor a program tulajdonképpen a pont rajzolásához felhasznált tömbelemeket felülírja a kiindulási adatokkal (590-610 sorok).

A test elcsúsztatását, az elmozdított test ábrázolásához szükséges koordinátákat a 630-690 programsorok számítják ki.

A térlátás elősegítéséhez nagymértékben hozzájárul az axonometrikus képhez felrajzolt tengelykereszt. A tengelyek által bezárt szög AL és BT értékét a 30. sorban határoztuk meg.

A program tovább komfortosítható a csúcspontok koordinátáinak INPUT-ként való betöltésével, az adatok DATA sorbeli tárolása helyett a lemezen vagy magnószalagon való rögzítésével. Ilyen komfortos térbeli rajzoló biztosító segédprogramok mind a COMMODORE, mind a SINCLAIR gépekhez kaphatók. Az itt leírt, C-4-en futó program vi-

TECHNISKOLA



MC-PROGRAMOK

szont a rajzolás és a számítás mechanizmusának jobb megértését segíti elő.
Dr. Sz. Lukács János

```
10 REM AXONOMETRIA+VETULET (PI=PI)
20 CLR:DIMPT(100,3),ZP(100),PA(100,3)
30 GOSUB 840:AL=120*PI/180:BT=90*PI/180
40 REM RAJZOLAS
50 GRAPHIC1,1:A=ZP(0):Q=COS(AL)
60 XA=100-SIN(AL)*PT(A,2)+SIN(BT)*PT(A,1)
70 YA=100-PT(A,3)-Q*PT(A,2)-COS(BT)*PT(A,1)
80 FORB=1TOSP:A=ZP(B):Q=COS(AL)
90 XS=100-SIN(AL)*PT(A,2)+SIN(BT)*PT(A,1)
100 IFXS<0THENXS=0:ELSEIFXS>319THENXS=319
110 YS=100-PT(A,3)-Q*PT(A,2)-COS(BT)*PT(A,1)
120 IFYS<0THENYS=0:ELSEIFYS>199THENYS=199
130 DRAW,XA,YATOXS,YS:IFB=1THENZE=100-PT(0,3)
140 IFB=1THENXE=PT(0,1)+220:YE=120+PT(0,2)
150 DRAW1,XE,YETOPT(A,1)+220,120+PT(A,2)
160 DRAW1,XE,ZETOPT(A,1)+220,100-PT(A,3)
170 XE=PT(A,1)+220:YE=120+PT(A,2)
180 ZE=100-PT(A,3):XA=XS:YA=YS:NEXT
190 D$="< * >":CHAR1,0,0,D$
200 GETA$:IFA$<>CHR$(13)THEN200
```

```
210 GRAPHIC0,1:PRINT"□"TAB(10)"MENU "
220 PRINT"███(1) = A TEST FORGATASA
230 PRINT"███(2) = ALAPALLAPOT"
240 PRINT"███(3) = TEST ELCSUSZTATASA "
250 PRINT"███(4) = KOORD.TENG.JELOLESE"
260 PRINTTAB(10)"██(0) = VEGE"
270 GETKEY$:IFA$<"0"ORA$>"4"THEN270
280 ONVAL(A$)+1GOTO290,300,590,630,700
290 END
```

```
300 PRINT"███(1) TEST FORGATASA "
310 PRINT"███(1) = FORGATAS Z-KORUL.
320 PRINT"███(2) = FORGATAS Y-KORUL.
330 PRINT"███(3) = FORGATAS X-KORUL.
340 GETA$:IFA$<"1"ORA$>"3"THEN340
350 Z=VAL(A$):CHAR,2,2*Z,">":CHAR,0,15,""
360 PRINTTAB(7)"███ ELFORGATAS SZOGE:";
370 INPUTDE:DE=DE*PI/180:ONZGOTO380,450,520
380 REM FORG Z-KORUL
390 FORX=0TOEP:Q=ATN(PT(X,1)/(PT(X,2)-1E-32))
400 EQ=Q-PI*(PT(X,2)<=0)-2*PI*(PT(X,1)<0ANDPT(X,2)>0)
410 EO=EO+DE:IFEQ>2*PI THENEO=EO-2*PI
420 R=SQR(PT(X,1)^2+PT(X,2)^2)
430 PT(X,1)=SIN(EQ)*R:PT(X,2)=COS(EQ)*R
440 NEXT:GOTO40
450 REM FORG Y-KORUL
460 FORX=0TOEP:Q=ATN(PT(X,3)/(PT(X,1)-1E-32))
470 EQ=Q-PI*(PT(X,1)<=0)-2*PI*(PT(X,3)<0ANDPT(X,1)>0)
480 EO=EO+DE:IFEQ>2*PI THENEO=EO-2*PI
490 R=SQR(PT(X,1)^2+PT(X,3)^2)
500 PT(X,1)=COS(EQ)*R:PT(X,3)=SIN(EQ)*R
510 NEXT:GOTO40
520 REMFORGX-KORUL
530 FORX=0TOEP:Q=ATN(PT(X,2)/(PT(X,3)-1E-32))
540 EQ=Q-PI*(PT(X,3)<=0)-2*PI*(PT(X,2)<0ANDPT(X,3)>0)
550 EO=EO+DE:IFEQ>2*PI THENEO=EO-2*PI
560 R=SQR(PT(X,2)^2+PT(X,3)^2)
570 PT(X,3)=COS(EQ)*R:PT(X,2)=SIN(EQ)*R
580 NEXT:GOTO40
590 REM KIINDULASI ALLAS
600 FORX=0TOEP:FORY=1T03
610 PT(X,Y)=PA(X,Y):NEXTY,X:GOTO40
620 GRAPHIC1,0:GOTO200
```

```
630 REMABRAELCSUSZTATAS
640 PRINT"███(1) TEST ELCSUSZTATASA:"
650 PRINT"AZ ELCSUSZTATAS ADATAI:"
660 INPUT"███X,Y,Z :";PX,PY,PZ
670 FORN=0TOEP:PT(N,1)=PT(N,1)+PX
680 PT(N,2)=PT(N,2)+PY
690 PT(N,3)=PT(N,3)+PZ:NEXT:GOTO40
700 REM
710 GRAPHIC1,0:REMTENGELYEK
720 IFKO=1THENTY=0:KO=0:GOTO740
730 KO=1:TY=1
740 XS=100-SIN(AL)*95:X1=100-SIN(AL)*105
750 YS=100-COS(AL)*95:Y1=100-COS(AL)*105
760 DRAW1,100,100TOXS,YS:IFYS>180THENYS=180
770 CHAR1,XS/8-1,YS/8+1,"Y"
780 XS=100+SIN(BT)*95:X1=100+SIN(BT)*102
790 YS=100-COS(BT)*95:Y1=100-COS(BT)*105
800 DRAW1,100,100TOXS,YS:IFYS>180THENYS=180
810 DRAW,100,100TOXS,YS:IFYS>180THENYS=180
820 DRAW1,100,100TO100,10:CHAR1,12,0,"Z"
830 GOTO190
```

```
840 READ EP,SP:FORX=0TOEP-1:FORY=1T03
850 READ PT(X,Y):PA(X,Y)=PT(X,Y):NEXTY,X
860 FORX=0TOSP:READZP(X):NEXT:RETURN
870 DATA 9,23:REM CSUCS,EL
880 DATA 0,0,0,50,0,0,50,50,0,0,50,0
890 DATA 0,0,50,50,0,50,50,50,0,50,50
900 DATA 25,25,100
910 DATA 0,1,2,3,0,4,5,1,5,6,2,6,7,3,7
920 DATA 4,0,4,8,5,6,8,7,4
```

Megjelent a Technika Magazin!



Szépsegédnálóh. beszámolóknak. Szórakoztató elektronika digitális. Ház. számítógépek. Vá. háló. változások a hazai autópiacon. Biotechnika. Amig. élet. a. nap.

Kapható az újságárusoknál. Ára: 46,- Ft.

TÜKRÖZÉSI FELADATOK

A grafikus adottságok egyre bővülő lehetőségei egy sor olyan példa megoldását, szemléltetését teszik lehetővé, amit korábban a fantáziára kellett bízni, vagy hosszú ideig tartó szerkesztéssel lehetett csak elvégezni.

Igaz, ha a rajzok, tükrözések, forgatások, axonometriák elkészítését a számítógépre bizzuk, akkor a programba be kell építeni a megoldáshoz szükséges matematikai összefüggéseket.

Valaha azt tanultuk, azt tanítjuk, hogy

ami megszerkeszthető, az ki is számítható.

Az első program egy síkidom tengelyre vett tükörképét szerkeszti meg és rajzolja ki a képernyőre.

A feladat megoldásához ismerni kell a síkidomot, annak csúcspontjai helyzetét. A csúcspontok számát, valamint azok x és y koordinátáit az 1. program 11–16. programsorok olvassák be, majd tárolják az R tömbben. Ezek a régi koordináták. Ezekből a régi koordinátákból fogja a program az (U) új koordinátákat meg-

határozni. Ehhez ismerni kell azt a tengelyt, amire a tükrözést végezni kell. Ennek ismeretéhez a program kéri a tengely iránytangensét és y tengelymenti metszetét.

A 20–24. programsorok számítják ki a síkidom tükrözés utáni új koordinátáit.

A program 32–51. programsorai egy sor trükköt tartalmaznak. Nevezetesen, hogy más számítógépeken is futtatható legyen, a program lehetőséget biztosít arra, hogy a tükrözött képet csak a képernyőre, vagy csak annak egy kerülettel határolt részére rajzolja ki.

A Commodore +4 nem jelez hibát, ha a képernyőn kívülre rajzolunk, más gépek viszont hibajelzéssel leállnak, ha ilyet kívánnánk tőlük.

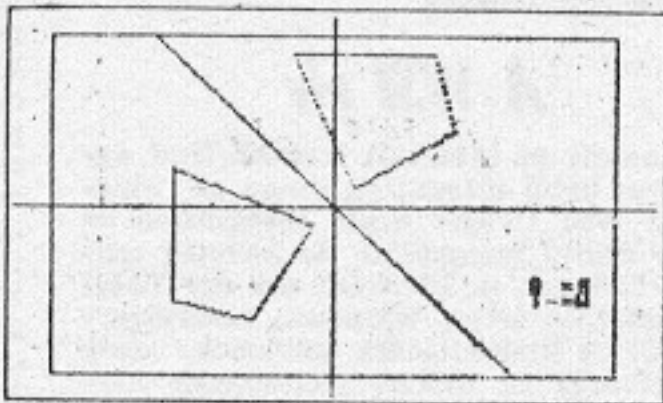
Az oldalak rajzolása közben tehát vizsgálja a program, hogy nem kerültünk-e kereten kívüli koordinátára.

Az oldalak rajzolása tehát pontonként történik. Az oldalak megrajzolása ezzel a módszerrel két buktatót is rejt. A ciklusban rajzolandó pontokat hol $x_1 < x_2$, hol $x_1 > x_2$ feltétel mellett kell rajzolni. Gondoskodni kell tehát a ciklus lépéskörének meghatározásáról. Erre két példát is láthatunk a 39. és 43. soron.

Nehézséget okozott volna a függőleges egyenesek megrajzolása, mert ilyenkor az oldal iránytangensének kiszámítása nullával való osztáshoz vezetne. A függőleges vonalak rajzolását a 42–45. programsorok végzik.

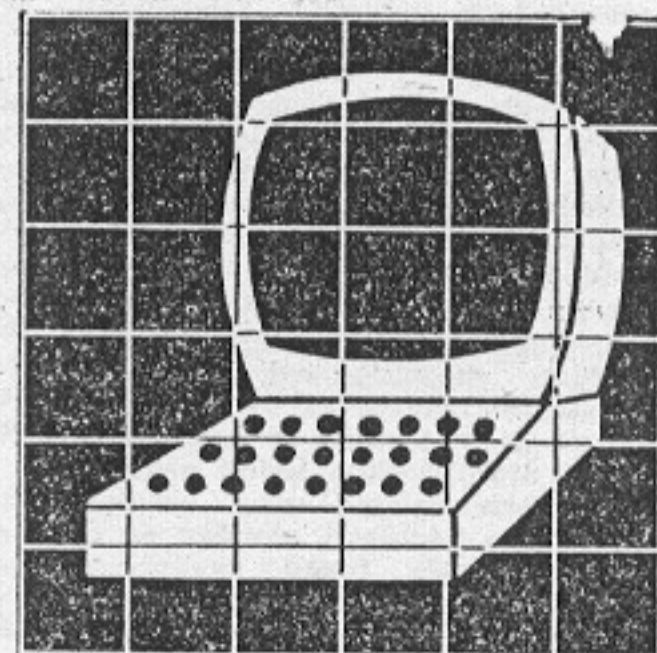
A program futásának eredményét az 1. ábra mutatja.

Egy másfajta tükrözésre nézzünk példát a Commodore +4 adta RDOT függ-



1. ábra

TECHNISKOLA



MC-PROGRAMOK

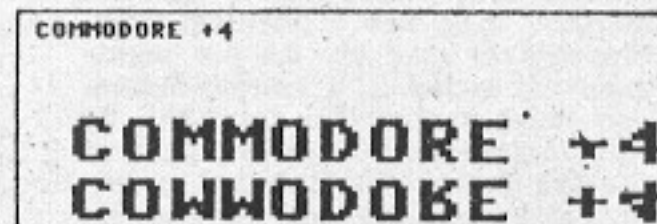
vény használatával. A 2. program futásának eredményét a 2. ábra mutatja.

A program egyrészt lehetővé teszi, hogy a maximum 14 karakter hosszú szöveget az eredeti karakter 3×3 méretű nagyításával a képernyő közepére kiirassuk. A grafikus képernyőre írt karakter pontjait vizsgálja az RDOT (2) függvény. Ha az adott képernyőponton van tintaszínű pont, akkor azt a 170. soron kezdődő ciklusokban a kívánt méretre nagyítja. A ciklusok végértékeinek változtatásával megtehető, hogy x és y irányban különböző arányban nagyítsunk.

A 190. sor a szöveg fejreállítását végzi el.

Egészen fantasztikus képet kaphatunk az x és y irányok felcserélésével. A kísérletező kedvűek ennek a módszernek az alkalmazásával akár ferdén is írhatnak a képernyőre.

Dr. Sz. Lukács János



2. ábra

```
10 REM TUKROZES TENGELERE COMMODORE +4
11 INPUT "CSUCSPONTOK SZAMA";N
12 DIM R(N+1,2),U(N+1,2)
13 FORI=1TON
14 :PRINT "X(";I;")=";:INPUT R(I,1)
15 :PRINT "Y(";I;")=";:INPUT R(I,2)
16 NEXT
17 R(N+1,1)=R(1,1):R(N+1,2)=R(1,2)
18 INPUT "IRANY TAN";M:INPUT "Y METSZ";B
19 BX=-140:BY=-90:HX=280:HY=180:S=160
20 FORI=1TON+1
21 :X=(R(I,1)+M*(-B+R(I,2)))/(M+2+1)
22 :Y=M*X+B
23 :U(I,1)=2*X-R(I,1):U(I,2)=2*Y-R(I,2)
24 NEXT
25 GRAPHIC1,1
26 DRAW,S,OTOS,200:DRAW,0,100TO320,100
27 BOX1,S+BX,100-BY,S+BX+HX,100-BY-HY
28 FORX=BXT0BX+HX
29 Y=M*X+B
30 IFY<BY ORY>BY+HY THENGOTO32
31 DRAW1,160+X,100-Y
32 NEXT
33 FOR V=1TO2
34 :FORI=1TON
35 : A=-R(I,1)*(V=1)-U(I,1)*(V=2)
36 : B=-R(I+1,1)*(V=1)-U(I+1,1)*(V=2)
37 : C=-R(I,2)*(V=1)-U(I,2)*(V=2)
38 : D=-R(I+1,2)*(V=1)-U(I+1,2)*(V=2)
39 : L=1:IF A>B THEN L=-1
40 : FORX=ATO B STEP L
41 : IF X<BX ORX>BX+HXTHEN GOTO 50
42 : IF B-A<>0 THEN GOTO47
43 : FORF=C TOD STEP-ABS(C-D)/(C-D)
44 : IFF<BY ORF>BY+HY THEN 46
45 : DRAW1,A+160,100-F
46 : NEXT :GOTO50
47 : Y=(D-C)/(B-A)*(X-A)+C
48 : IF Y<BY ORY>BY+HYTHEN GOTO 50
49 : DRAW1,160+X,100-Y
50 : NEXT
51 :NEXT
52 NEXT
53 GETKEYA$:GRAPHICO
```

1. program

```
100 REM SZOVEG NAGYITAS/TOKROZES +4
110 GRAPHIC2,1
120 GOSUB240
130 FORV=0TO8*LEN(A$)
140 : FORF=0TO7
150 : LOCATE V,F :J=RDOT(2)
160 : IF J=0 THEN GOTO 210
170 : FOR A=0TO2:FORB=0TO2
180 : DRAW J,K+3*V+A,50+3*F+B
190 : DRAW J,K+3*V+A,100-3*F-B
200 : NEXT:NEXT
210 : NEXT
220 NEXT
230 END
240 INPUTA$:CHAR1,0,0,A$
250 IF LEN(A$)>14 THEN GOTO240
260 K=INT((320-LEN(A$)*24)/2)
270 RETURN
```

2. program

Egyszerű szívizomvizsgálat

Az ultrahang az új kutatások szerint jobb lehetőséget szolgáltat a szív problémáinak műtét nélküli felderítésére, mint az eddig ismeretek és használat.

A szívverés képi megjelenítésére már esztendőnk óta használnak ultrahangot. Az új számítógépes elemzéssel azonban a szívizom szövetszerkezetére vonatkozó adatokat is tudnak szerezni David Jan Skorton, az USA Iowa város orvosgyetemének kutatója szerint.

A normál és az attól eltérő szövetek különbözőképpen verik vissza az ultrahangot, aminek számítógépes elemzésével az orvosoknak hamarosan módjukban áll majd osztályozni a szívizom állapotát. A jelenlegi állapotok

szerint megkülönböztethető az egészséges izom, a túlzottan kevés oxigénnel ellátott és így károsodott izom, a közelmúltban lezajlott szívroham következtében gyógyíthatatlanná vált szívizom és régebbi roham hegeitől sérült izom.

Az ultrahang nemcsak az orvostudomány szempontjából nagy jelentőségű. Szakértők szerint az ultrahanggal végzett besugárzás a 21. század számos technológiájának fogja alapját képezni, beleértve a magas hőmérsékletű szupravezetőket is.

Közönséges folyadékba bocsátva az ultrahang mikroszkopikus méretű izzó pontokat tud létesíteni, molekulákat szétválasztani vagy egyesíteni.

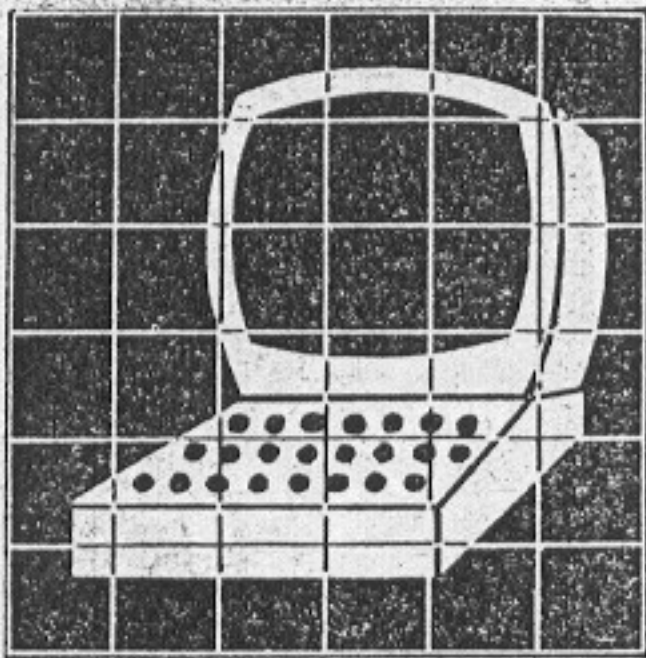
Liaz Sali hűtőkocsi



A csehszlovák Motokov Külkereskedelmi Vállalat új hűtőkocsi-típust mutatott be Liaz Sali 323 típusjellel.

A jármű alváza és vezetőfülkéje a Liaz-gyár 110 020 típusa, amelynek hajlasképlete 4×2. A hathengeres, soros, vízhűtéses, 11,94 literes dízelmotor teljesítménye 2000/min fordulatszámon 156 kW, maximális nyomatéka 1400/min fordulatszámon 788 Nm.

A jól szigetelt hűtőfelépítmény alapterülete 13,9 négyzetméter, térfogata 30 köbméter. A 8700 kg saját tömegű, 4300 kg teherbírású járműnek 10+2 fokozatú sebességváltója, hidropneumatikus tengelykapcsolója, kétkörös pneumatikus fékberendezése, hidraulikus szervókormány, 24 V-os villamos berendezése van, maximális sebessége 92 km/h.



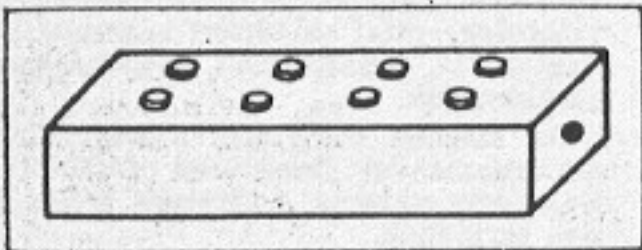
MC-PROGRAMOK

TÉGLALOGO

Az informatika és a programozás tanításában a LOGO nyelvnek és a Lego építőkészletnek figyelemreméltó érdemei vannak. A kettő ötvöztetéséről, a Lego-Logo játékról, egy új képességfejlesztő módszerről egyre többet hallhatunk. A „LOGO-környezet” — ami a LOGO programnyelv nemcsak számítógépes alkalmazását jelenti a gondolkodás módszereinek fejlesztésére — alapvető része a teknőgrafika. A „teknőc” az előre-hátra, jobbra-balra szavak és a haladás, fordulás mértékét megadó paraméterek segítségével a sík bármely pontjára vezethető. Ha „író tollat nem emelte fel”, haladásának nyoma látható (papíron, képernyőn). A teknőgrafika szintaktikája percek alatt elsajátítható, majd ennek segítségével egy új geometriai szemléletmód fejleszthető ki. A LOGO nyelvről már magyarul is értékes irodalom található. A LOGO környezet algoritmizálást, térszemléletet, informatikát hatékonyan fejlesztő része a Lego-Logo.

Crawford Craig publikálta, hogyan alkalmazták a LOGO programnyelvet egy könnyen algoritmizálható művelet: a Lego elemekből való építkezés leírásához. A Lego-Logo mesterséges nyelv, a 3-D (háromdimenziós) teknőgeometriára emlékeztet. Teknőcszerű mozgást végző Lego téglákkal bármely Lego konstrukció elkészíthető. Az utasításkészlet megválasztásához Craig a következőket javasolta:

A nyolcszagos Lego elemet nevezzük négyszögletesnek, jelöljük így: 4,2. Képzeljük el a teknőcöt először ilyen alakúnak. (Elnézést az állatvédőktől!) A téglát egyik végére rajzoljunk egy pontot, ez lesz a teknőc orra. A téglát helyzetével egy geometriai pontot, teknőccé minősítésével egy vektort határoz meg.



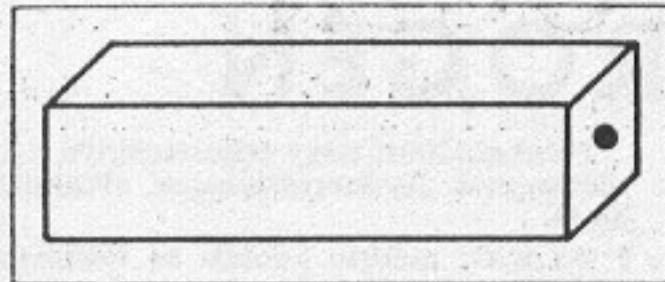
1. ábra

A téglateknőc mozgatható. A mozgást polárkoordináta-rendszerben írjuk le. Craig az utasításkészletre is ad egy javaslatot, hangsúlyozva, hogy ez a készlet csak egyike a lehetőségeknek, és nem biztos, hogy a legjobb. Az angol téglateknőc a következő szavakat érti:

FORWARD n
BACK n
UP n
DOWN n
RIGHT n
LEFT n
BUILD

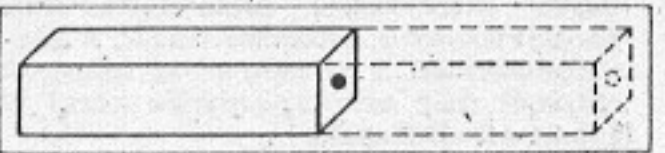
A felhasználók tehát maguk is alkothatnak, a céljaiknak jobban megfelelő teknőnyelvet. Ezzel mi is megpróbálkozunk. Természetesen magyar szavakkal fogjuk irányítani tégláinkat, és igyekezzünk átvenni a LOGO nyelv szintaktikájából és szemantikájából minél többet.

A három dimenziótengely körüli forgásból (egyelőre) csak a vízszintes síkúakat használjuk. Elég sok szerkezet megépíthető a vízszintes tengely körüli csavarás és a vízszintes síkból való kibillentés nélkül is. Relatív koordináta-rendszert használunk, tehát a mozgásokat mindenkor a teknőc helyzetéhez és haladási irányához viszonyítjuk. Lego elemek helyett inkább téglákkal játszunk. (A modellhezhez megfelel dominó, gyufásdoboz stb.). Ha külön nem jelezzük, a téglateknőc oldalainak aránya 4:2:1. (A gyufásdoboz fiókját kissé kihúzáva beállítható ez az arány). A téglateknőc alaphelyzete legyen a következő:



2. ábra

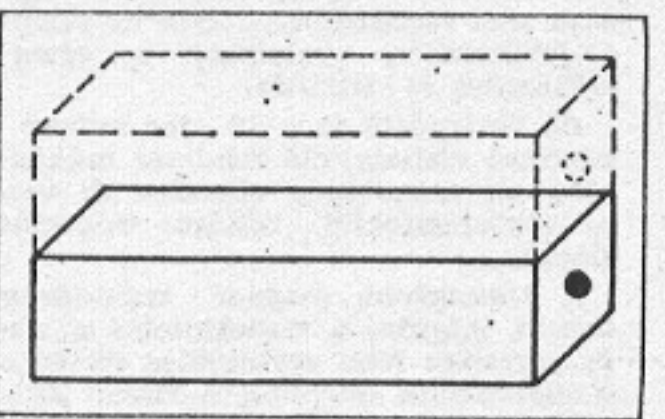
Mekkora legyen a lépésegység? Ha fél és negyed téglákra, illetve ilyen hosszúságú elmozdulásokra számítunk (ami mind a falazás, mind a Lego-szerkezetek esetében gyakori), a téglát hosszúságú előrehaladást célszerű négy egységnek választani. Az előre haladás parancs-száva legyen „tovább 4”, jelentése tehát:



3. ábra

A hátrafelé mozgáshoz új szó nem is kellene, hiszen „tovább: A” jelenti az „A” egységnyi visszafelé haladást, de mi téglateknőceinkkel elsősorban a kisiskolásokat akarjuk szórakoztatni, ezért használjuk a „hátrál” utasítást!

Legyen A „fő 1” jelentése a téglát vastagságának megfelelő, függőleges irányban egy lépés:

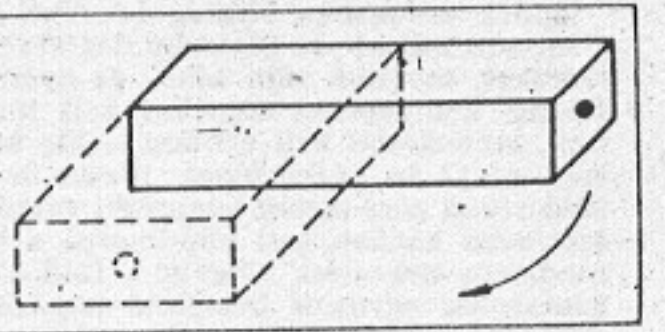


4. ábra

Alapértelmezésben az egyes elhagyható. A süllyedés parancsszáva „alá” legyen!

Az elforduláshoz kézenfekvő a jobbra és a balra szavak használata. A fordulás mértékét fokokban adjuk meg. Mérőszám nélkül az utasítások jobbra, illetve balra átot jelentsenek!

A jobbrafordulás értelmezését a következő rajz érzékelteti:

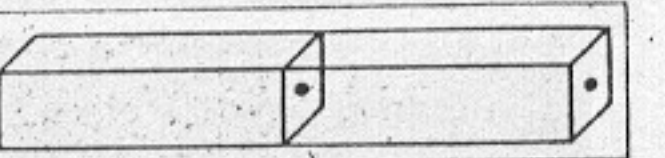


5. ábra

A LOGO nyelv mintájára használjuk az ismétlés és az eljárás szervezését. Ehhez szavaink az „ismétel” és a „tudd”.

Három irány, irányonként két irányítás, ez hat utasítás, ehhez járul a két logoprimitív, a kilencedik parancsral teljes lesz utasításkészletünk. Egy téglát helyhez kötésére, rögzítésére válasszuk a „rak” szót. A „rak” tehát az origó áthelyezését jelenti. A téglateknőc mindig elődje helyéről (vagy alaphelyzetből) indul, és mozgása végén beépül. Kezdhessük a játékot.

rak tovább 4 rak
eredménye:



6. ábra

Tisztelt Olvasónk most már bizonyára tud programot írni három téglát egymás után helyezésére. Tegye meg!

N elemű futó téglalorakásának eljárása:

Tudd sor :N
ismétel :N (rak tovább 4)
H darab téglából álló oszlop:
Tudd oszlop :H
ismétel :H (rak föl)

(Felcserélhetők a zárójelben levő utasítások?)
A kötés nélküli téglarakás építésének programja a sor- és az oszlopeljárások segítségével is írható:

Tudd rakás :N :H
ismétel :H (sor :N föl hátrál :N*4)
vagy

Tudd rakás2 :N :H
ismétel :N (oszlop :H tovább 4 alá :H)

Miután megszoktuk az utasításszavakat, rövidíthetjük is azokat:

t tovább
h hátrál
j jobbra
b balra
f föl
a alá

A rak utasítás az algoritmus leírásakor vagy olvasásakor „fix pontokat” jelöl, ne rövidítsük, mert így jobban kiemelkedik, s megkönnyíti az értelmezést! A „Tudd” és az „ismétel” utasítások szerepe a műveleti parancsoktól eltérő, megkülönböztetésüket — nagybetűvel írás — célszerűnek találtuk.

Használjuk a bevezetett rövidítéseket! Eljárás futó sorban, kötésben rakott fal építésére:

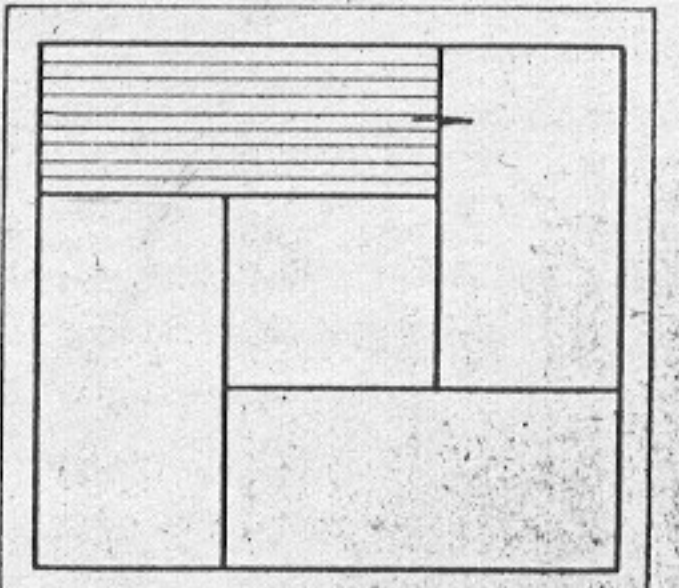
Tudd fal :N :H
ismétel :H/2 (sor :N f h 4* :N—2 sor :N f h 4* :N+2)

Ha falazni tudunk, már épülhet a ház. Építkezés közben, ha téglalorakát változtatunk, az új darab úgy helyezendő el, hogy orrával ugyanazt a síkot érintse, mint amelyiket elődje is tenné. A „rak” után ilyenkor kettő vagy három paraméter mutatja be az új elemet. A fél-tégla például 2,2, a dupla vastagságú falazóelem 4,2,2.

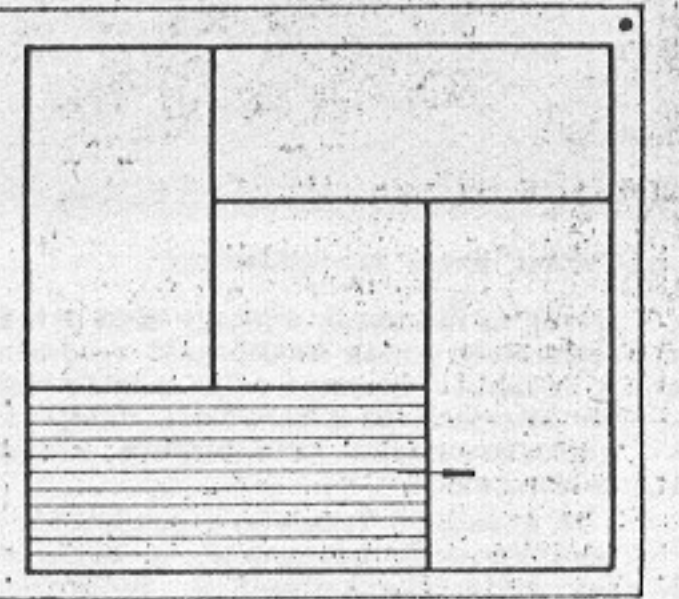
A téglalogóban is érvényesülnek a teknőgeometria szabályai:

Építsünk kéményt!

Az első szint téglakiosztása:



7. ábra



8. ábra

Az elsőnek lerakott téglát vonalkázással jelöltük. A program:

ismétel 4 (rak t 4 jobbra)

A négy ismétlés végén a kiinduló helyzetbe jutunk.

A második szint eltérő, azért, hogy azt kötéssel rakjuk. (8. ábra)

ismétel 4 (rak t 4 balra)

Vegyük észre, hogy a két kiosztás egymásnak tükörképe, ennek megfelelően a programban — a teknőgrafika szabályai szerint — a jobbra és balra utasítások cserélődtek.

A kémény építése a két szint váltakozó rakásából áll. Az ismétlődő művelet sor eljárása:

Tudd kéményelem
szint1 emelés1 szint2 emelés2

A kéményelem eljárás részei:

Tudd szint1
ismétel 4 (rak t 4 j)
Tudd szint2
ismétel 4 (rak t 4 b)
Tudd emelés1
föl jobbra
Tudd emelés2
föl balra

Az emelési eljárásnak a szint1 eljárás utolsó (és egyben első) helyzetéből a szint2 eljárás kezdő helyzetébe kell juttatnia a téglateknőcöt, de mivel a szint2 eljárás ciklikusan szimmetrikus, így indulhatunk más helyről is.

Végül a :H magasságú kémény építésére az eljárás:

Tudd kémény :H
ismétel :H/2 (szint1 emelés1 szint2 emelés2)

Ennek a kéménynek a belső mérete fél téglahossz, fél téglahossz. Írjon másfél téglát belsőátmérőű kéményt építő eljárást, majd általános (modultöbbszörös) eljárást!

További gyakorló feladatokat:

1. Mit építhetünk a következő programmal?

Tudd szint10
rak b t 2 j rak
Tudd szint20
b rak j t 2 b rak
Tudd emel1
b h 2 j f
Tudd emel2
j h 2 f

Tudd épületelem :H
ismétel :H/2 (szint10 emel1 szint20 emel2)

2. Írjon programot pillér építésére! Az alaprajzi kiosztások:



9. ábra

3. Építsen (lehetőleg Lego elemekből) teknőcöt!

rak 2,2 t 4 rak
b t 2 rak t 2 rak
b t 4 rak b
f t 3 b t 3 rak 4,2
b t 2 j rak
j t b t 2 rak 2,2
f h 3 rak

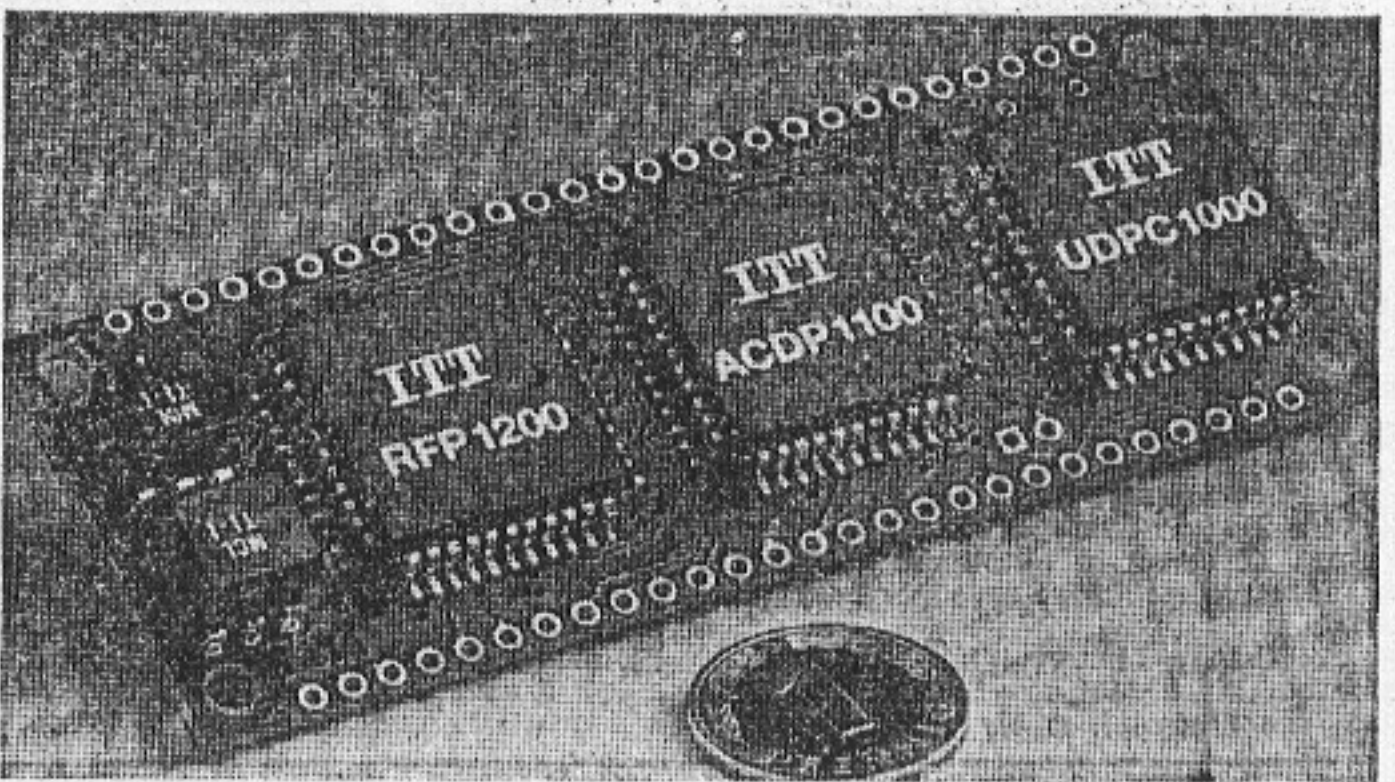
Farkas Károly

Háromchipes autórádió

Az autórádiókkal egybeépített kazettás magnók, illetve CD-lemezjátszók szükségessé teszik az elektronika minél nagyobb mérvű integrálását a készülék méreteinek korlátok közötti tartásához. Ennek figyelembevételével az ITT szakemberei olyan chipet fejlesztettek ki, amelyekkel egy komplett rádióvevő alakítható ki. Valamennyi RF és HF jelfeldolgozási funkciót három chip lát el.

Az RFP 1200 típusú RF-processzor lényegében egy AM/FM vevőegység, amely a null-KF-elv alapján működik. Az

ACDP 1100 feladata a hangátalakítás, demodulálás, a digitális FM-demoduláció és a sztereó dekódolás a magában foglalt A/D és D/A átalakítók igénybevételével. A programozható digitális UDPC 1000 jelprocesszor a különböző HF-funkciókat végzi el, amilyen a magas és mély hangszínszabályozás, a balanszbeállítás, a hangerő-szabályozás és a közlekedési adók kódjeleinek dekódolása. Ehhez a chiphez interfész használatával más hangforrások is csatlakoztathatók, pl. CD-lemezjátszó vagy DAT-magnó (ábra).



REJTVÉNYEK GENERÁLÁSA

TECHNISKOLA

A nyári szünetben talán játékosabb dolgokkal akar foglalkozni a számítógép tulajdonosa is. A programírás természetesen komoly dolog, de a futás eredménye a játék lehetőségét, a játékos tanulás lehetőségét biztosíthatja.

Az ábra az 1. program futtatásának eredményét szemlélteti. A program 110–240 sorai előállítanak négy véletlen számot, és ezeknek „keresztbe hosszába” képezi az összegét. Ezt az összeget kilenc Q(n) számot, annak számjegyeit a 250–390 sorok kódolják, és számjegy helyett egy véletlenül kiválasztott betűt (lehetne grafikus jel is) helyettesít a Cs(n)-be.

JMH	+	JMJ	=	AXB
VJE	+	MAX	=	AXH
MEA	+	GXW	=	WAM
238	+	232	=	470
126	+	347	=	473
364	+	579	=	943

Az 1. program futtatásának eredménye

Ezeket a kódolt számokat a tulajdonképpeni rejtvényt a 400–460 sorok írják ki a képernyőre. A feladvány megoldását ezután bárki papíron ceruzával megkísérélheti megoldani, a türelmét vesztett rejtvénytfejtő a T betű lenyomásával megkaphatja a számok és betűk összerendezésének táblázatát (470–510 sorok). Ennek segítségével már könnyen kitölthető a rejtvény, de ha már úgyis „szabad a gazda”, akkor újabb T lenyomásra az 530–630 sorok a feladat teljes megoldását kiírják a képernyőre. Ez a program Commodore 64-re készült.

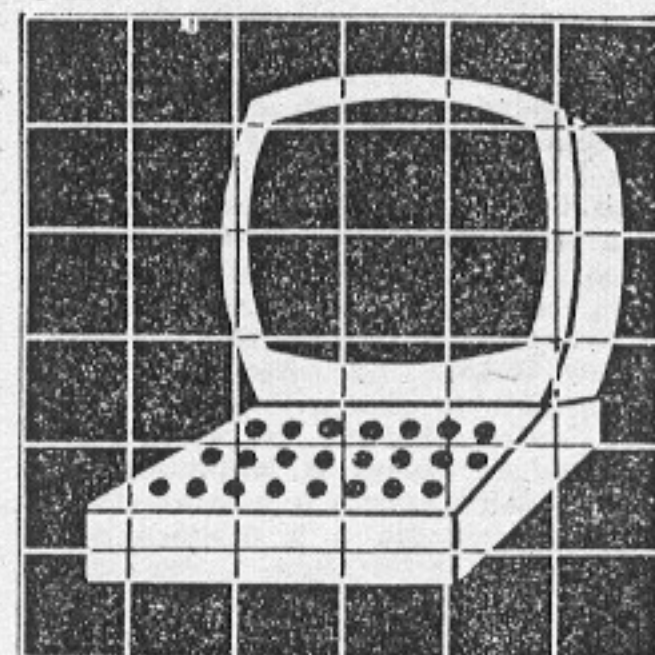
A 2. program C+4 számítógépre készült. A sor a DATA sorokban tárolt szavak közül véletlenszerűen kiválaszt egyet és annyi pontot ír a képernyőre, ahány betűt tartalmaz a kitalálendő szó. A program számontartja a gép előtt ülő játékos próbálkozásainak számát, de segít is a szó kitalálásában. Amennyiben a próba során olyan betűt közlünk a programmal, amilyen van a kitalálendő szóban, akkor a program a megfelelő pont(ok) helyére beírja a már megtalált betűt. A játék egészen a gép által „gondolt” szó kitalálásáig folytatható.

Egy sor trükköt tartalmaz a 3. program, melyben a DATA sorokba felvett

szavak közül a gép előtt ülő játékosnak kell gondolkodnia egyet. A program ki fogja találni azt úgy, hogy rendre betűket ír ki a képernyőre és a játékosnak az I vagy az N betű lenyomásával kell segítenie a gépet aszerint, hogy az illető betű szerepel-e a gondolt szóban. A program addig küzd, amíg ki nem szűri az általa ismert szókészletről azt a szót, amelyben csak a már kitalált betűk szerepelnek. Ha becsapjuk a programot vagy elnézünk egy betűt, akkor a gép nem fogja tudni kitalálni a szót és jelzi, hogy „baj van”.

A DATA sorok tovább fejleszthetők, és kis átalakítással tetszőleges hosszúságú szavakkal való játékra is alkalmassá tehető. Egy vérbeli programozónak talán ez okoz nagyobb örömet, mint maga a játék. Nosza, lássunk hozzá!

Dr. Sz. Lukács János



MC-PROGRAMOK

```

100 REM SZO KITALALO
110 READ N:DIMS$(N),P$(15)
120 FORI=1TON:READS$(I):NEXT
130 SCNCLR:P=0:I=INT(1+N*RND(0))
140 F$=S$(I):H=LEN(F$)
150 FORI=1TOH:CHAR,15+I,12,"."
160 P$(I)="0":NEXT
170 CHAR,0,20,"":INPUT"TIPIP=";T$
180 FORI=1TOH:A$=MID$(F$,I,1)
190 IFA$=T$THENCHAR,15+I,12,T$:P$(I)="1"
200 NEXT
210 J=0:FORI=1TOH:IFP$(I)="0"THENJ=1
220 NEXT:IFJ=1THEN170
230 CHAR,0,20,""
240 FORI=2544TO2544+H:POKEI,128:NEXT
250 GETKEYA$:GOTO130
260 DATA 9,ALMA,KORTE,KUTYA,MACSKA,MADAR
270 DATA PIROS,TILOS,JIHASZ,BOJTAR
    
```

2. program

```

10 REM SZOKITALALO GONDOLJ !
12 REM CSAK NEGYBETUS SZAVAKRA
14 READ N:K=N:DIM S$(N),E$(N),B(26)
16 W$="TEHAT A SZO: ":P$="*"
18 FORI=1TON:READS$(I):E$(I)=S$(I):NEXT
20 FORI=1TON:PRINT S$(I),E$(I):NEXT
22 IF K=0 THEN PRINT "BAJ VAN":END
24 IF K=1 THEN GOTO 74
26 B=INT(26*RND(0)):IFB(B)=1THEN 26
28 B(B)=1:B$=CHR$(65+B)
30 PRINT B$:INPUT "I/N ";V$
32 IF V$="N" THEN GOTO 56
34 IF V$<>"I" THEN GOTO 30
36 REM VAN BENNE ILYEN BETU
38 FORV=1TON:J=0
40 V$=E$(V):IFV$="" THEN GOTO 54
42 FORX=1TO4
44 IF MID$(V$,X,1)<>B$ THEN GOTO 50
46 V$=LEFT$(V$,X-1)+P$+RIGHT$(V$,4-X)
48 J=1
50 NEXT:E$(V)=V$:IFJ=0THENE$(V)="" :K=K-1
52 IF E$(V)="****" THEN GOTO 72
54 NEXT:GOTO 20
56 REM NINCS BENNE ILYEN BETU, TOROLNI
58 FORV=1TON
60 :IF E$(V)="" THEN GOTO 70
62 :FORX=1TO4
64 :IFB$<>MID$(E$(V),X,1) THEN GOTO 68
66 :E$(V)="" :K=K-1:X=5
68 :NEXT
70 NEXT:GOTO 20
72 PRINT W$:S$(V):END
74 FORI=1TON
76 :IF E$(I)<>"*" THEN PRINTW$:S$(I)
78 NEXT:END
80 DATA 6:REM AHANY ADAT KOVETKEZIK
82 DATA SAMU,ALMA,AUTO,TYUK,PAPA,ROKA
    
```

3. program

```

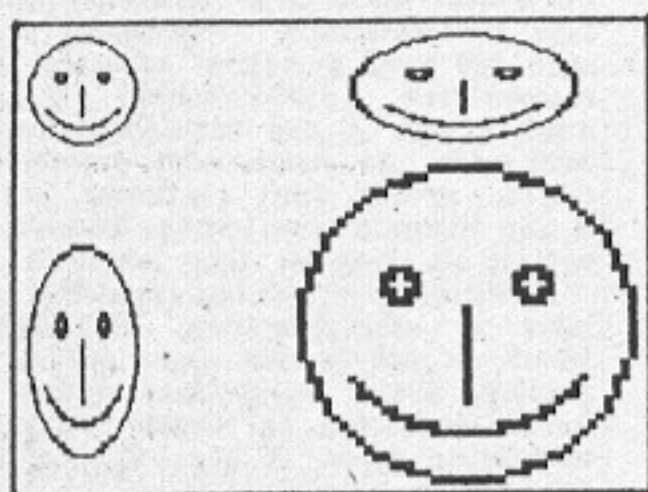
100 REM BETUSZAMTAN REJTVENY
110 FORI=1TO4
120 Z(3,I)=INT(10*RND(0))
130 Z(2,I)=INT(10*RND(0))
140 Z(1,I)=INT(7*RND(0))
150 Q(I)=100*Z(1,I)+10*Z(2,I)+Z(3,I)
160 NEXT
170 Q(5)=Q(1)+Q(2):Q(6)=Q(3)+Q(4)
180 Q(7)=Q(1)+Q(3):Q(8)=Q(2)+Q(4)
190 Q(9)=Q(7)+Q(8):IFQ(9)>999 THEN 110
200 FORI=5TO9
210 Z(1,I)=INT(Q(I)/100)
220 Z(2,I)=INT((Q(I)-Z(1,I)*100)/10)
230 Z(3,I)=Q(I)-Z(1,I)*100-Z(2,I)*10
240 NEXT
250 A$(0)=CHR$(RND(0)*26+65)
260 FORI=1TO9
270 A$(I)=CHR$(RND(0)*26+65)
280 FORJ=0TOI-1
290 IF A$(J)=A$(I) THEN J=I:NEXTJ:GOTO 270
300 NEXTJ
310 NEXTI
320 FORI=1TO9
330 S1$=A$(Z(1,I))
340 IFZ(1,I)=0 THEN S1$=""
350 S2$=A$(Z(2,I))
360 IFZ(2,I)=0 AND Z(1,I)=0 THEN S2$=""
370 S3$=A$(Z(3,I))
380 C$(I)=S1$+S2$+S3$
390 NEXT
400 PRINT CHR$(147):PRINT:PRINT
410 P$=""
420 PRINTP$:C$(1);M$:C$(2);" = ";C$(5)
430 PRINT:PRINT" + + +"
440 PRINTP$:C$(3);M$:C$(4);" = ";C$(6)
450 PRINT:PRINT" = = ="
460 PRINTP$:C$(7);M$:C$(8);" = ";C$(9)
470 GET A$:IF A$<>"T" THEN GOTO 470
480 PRINT:PRINT
490 FORI=0TO9:PRINTI;" ";:NEXT:PRINT
500 FORI=0TO9:PRINT" ";A$(I);" ";:NEXT
510 PRINT
520 GET A$:IF A$<>"T" THEN GOTO 520
530 PRINT:PRINT
540 FORI=1TO9
550 FORJ=1TO3
560 F$(I)=F$(I)+RIGHT$(STR$(Z(J,I)),1)
570 NEXT
580 NEXT
590 PRINTP$:F$(1);M$:F$(2);" = ";F$(5)
600 PRINT:PRINT" + + +"
610 PRINTP$:F$(3);M$:F$(4);" = ";F$(6)
620 PRINT:PRINT" = = ="
630 PRINTP$:F$(7);M$:F$(8);" = ";F$(9)
    
```

1. program

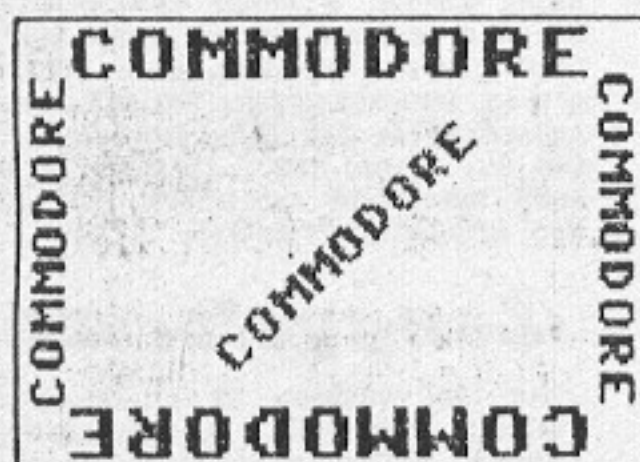
KÉPERNYŐKEZELÉS C+4 SZÁMÍTÓGÉPEN

A játékprogramokhoz a grafikus képernyő adta lehetőségeket már a kezdő programíró is fel akarja használni. Sokszor jó lenne, ha valamilyen alakzatot tudnánk mozgatni a képernyőn. Erre remek lehetőséget kínál a C 64 számítógép, nekünk viszont a C+4 számítógépen meg kell elégedni a SPRITE helyett a kicsit nehezebben használható SHAPE-pel. Az együtt kezelhető alakzat 255 pont (pixel) területet fedhet le a képernyőn. Ez a korlátozás már csak azért sem vészes, mert mozgó alakzat esetében a nagyobb SHAPE-ek elég lassan helyezhetők át. Minél kisebb méretű területet helyezünk át, annál folyamatosabbnak tűnő mozgás érhető el.

A SHAPE szerkesztéséhez ad segítséget az 1. program, amelynek használatával jól áttekinthető méretben szerkeszthetjük meg az alakzatot. A program kéri az alakzat, a SHAPE méretét, s ennek megfelelően kirajzolja azt a rasztert,



1. ábra



2. ábra

amelyen a kurzormozgató billentyű segítségével a kívánt helyre vezérelhetjük a pontot. Ott azután a 0 billentyű segítségével törölhetünk vagy az 1. billentyű segítségével újabb pontot helyezhetünk az ábrába. Az elkészült alakzatot az S betű lenyomásával magnószalagon vagy mágneslemezen tárolhatjuk.

A 2. program ahhoz ad segítséget, hogy a tárolt SHAPE-et visszatöltsük a számítógépbe. A program a SHAPE pontjait a grafikus képernyőre visszatölti és az N \$ stringváltozóban tárolja azt.

A SHAPE-ek speciális alkalmazása lehet az, amikor egy-egy ékezetes betűt vagy speciális karaktert megszerkesztünk és azután azt a kiírt szövegben az adott helyen SHAPE-ként jelenítjük meg.

A SHAPE megjelenítése, áthelyezése során kijelölhető az is, hogy az alakzat letakarja-e vagy rámásolódik-e a korábbi képernyőtartalomra. Ez a SHAPE megjelenítése során előírt AND, OR, XOR, (0,1,2) logikai feltétel előírásával érhető el.

A SHAPE-ek segítségével az alakzat csak azonos méretben és állásban helyezhető át. Amennyiben az alakzatot, betűt (szöveget) nagyítani akarjuk, akkor a 3. programot használhatjuk fel. Ennek segítségével az INPUT-ként közölt szöveget a képernyő közepére nagyítva tudjuk kiírni. Az X és az Y értékének megváltoztatásával a nagyítás mértéke vízszintesen, illetve függőlegesen (akár különböző mértékben is) módosítható.

Az itt bemutatott lehetőséget használja fel a 4. program, amely megrajzolja a kiindulási alakzatot, majd pedig elkészíti a fej (alakzat) különböző arányú nagyított (torzított) képét (1. ábra).

A képernyőtartalom letapogatása — RDOT (2) — és az áthelyezett ábrák természetesen nemcsak azonos állású kép előállítását teszik lehetővé.

A 2. ábrát az 5. program segítségével készítettük. A ferde irányú (elforgatott) ábra készítése során vegyük figyelembe, hogy bizonyos mértékű torzítás léphet fel.

További kísérletezéssel, az itt felsorolt módszerek kombinálásával ez a játék a végkimerülésig folytatható. Sok sikert!

Dr. Sz. Lukács János

```
10 REM SHAPE ELOALLITO COMMODORE +4
12 SCNCLR:PRINT "SHAPE SZERKESZTES"
14 CHAR 1,5,6,"KURZOR MOZGATAS"
16 CHAR1,8,7,"NYILAKKAL"
18 CHAR1,5,9,"1 PONT HELYEZES"
20 CHAR1,5,11,"0 PONT TORLES"
22 CHAR1,5,13,"S SHAPE LEMEZRE":PRINT
24 INPUT "A SHAPE NEVE:";N$
26 INPUT "SZELESSEGE :";O:IF O>40 THEN26
28 INPUT "MAGASSAGA :";S:IFS>25 THEN28
30 IF O*S>255 THEN PRINT "BAJ LESZ":END
32 DIM R(S,O):SCNCLR:A=3072:Q=46
34 FOR Y=0 TO S-1:FOR X=0 TO O-1
36 R(Y,X)=46:POKE A+Y*40+X,46
38 NEXT:NEXT
40 X=0:Y=0:GOTO 60
42 GETKEY Q$:Q=46
44 P=A+X+40*Y:POKEP,R(Y,X):POKEP-1024,0
46 IF Q$="↑" THEN Y=Y+1:IF Y>S-1 THEN Y=0
48 IF Q$="↓" THEN Y=Y-1:IF Y<0 THEN Y=S
50 IF Q$="←" THEN X=X-1:IF X<0 THEN X=O
52 IF Q$="→" THEN X=X+1:IF X>O-1 THEN X=O
54 IF Q$="1" THEN R(Y,X)=81:Q=81
56 IF Q$="0" THEN R(Y,X)=46
58 IF Q$="S" THEN GOTO 64
60 P=A+X+40*Y:POKEP,Q:POKEP-1024,128
62 GOTO 42
64 OPEN2,8,2,"S0:" + N$ + "S,W"
66 PRINT#2,CHR$(O-1);CHR$(S-1);
68 FOR Y=0 TO S-1:FOR X=0 TO O-1
70 PRINT#2,CHR$(PEEK(A+Y*40+X));
72 NEXT:NEXT:CLOSE 2
```

1. program

```
10 REM SHAPE VISSZATOLTO
15 INPUT "NEVE";N$
20 OPEN2,8,2,N$+"S,R"
25 IF DS<>0 THEN PRINTDS$:CLOSE2:GOTO 15
30 GRAPHIC1,1
35 GET#2,Q$:X=ASC(Q$+CHR$(0))
40 GET#2,Q$:Y=ASC(Q$+CHR$(0))
45 FOR F=0 TO Y:FOR V=0 TO X
50 GET#2,Q$:P=ASC(Q$+CHR$(0))
55 IF P=81 THEN DRAW1,V,F
60 NEXT:NEXT:CLOSE2
65 SSHAPE N$,0,0,X,Y
70 REM MOST MAR HASZNALHATO A SHAPE
```

2. program

```
100 REM SZOVEG NAGYITAS +4
110 GRAPHIC2,1:X=2:Y=3
120 INPUTA$:CHAR1,0,0,A$
130 H=LEN(A$):IF H*X*8>320 THEN 120
140 K=(300-H*X*8)/2
150 FORF=0TO7
160:FORV=0TO8*H
170:LOCATE V,F
180:IFRDOT(2)=0 THEN 220
190:FOR A=0TOX-1:FORB=0TOY-1
200:DRAW 1,K+X*V+A,80+Y*F+B
210:NEXT:NEXT
220:NEXT
230 NEXT
```

3. program

```
100 REM ABRAATHELYEZES +4
110 GRAPHIC1,1
120 CIRCLE1,20,20,20:DRAW1,20,18TO20,30
130 CIRCLE1,12,15,2:CIRCLE1,28,15,2
140 CIRCLE1,20,20,16,14,120,240
150 FOR V=0 TO 40:FOR F=0 TO 40
160:LOCATEV,F:A=RDOT(2):IF A=0 THEN260
170:FOR I=2*VTO2*V+1:REM VIZSZINTES
180:DRAW,120+I,F
190:NEXT
200:FOR I=2*FTO2*F+1:REM HOSSZUKAS
210:DRAW,V,80+I
220:NEXT
230:FOR I=3*VTO3*V+2:FOR J=3*FTO3*F+2
240:DRAW 1,100+I,50+J
250:NEXT:NEXT
260:NEXT:NEXT
```

4. program

```
10 REM FERDE KIIRAS C+4
12 GRAPHIC1,1:CHAR1,0,0,"COMMODORE"
14 FOR V=0 TO 71
16:FOR F=0 TO7
18:LOCATE V,F:A=RDOT(2):IF A=0 THEN 26
20:FOR I=3*VTO3*V+2:FOR J=3*FTO3*F+2
22:DRAW 1,40+I,8+J:DRAW1,254-I,190-J
24:NEXT:NEXT
26:NEXT
28 NEXT
30 FOR V=0 TO 71
32:FOR F=0 TO7
34:LOCATE V,F:A=RDOT(2):IF A=0 THEN 46
36:FOR I=2*VTO2*V+1:FOR J=2*FTO2*F+1
38:DRAW,26+J,170-I:DRAW,270-J,26+I
40:Q=.7*I:W=.7*J
42:DRAW,INT(100+W+Q),INT(146+W-Q)
44:NEXT:NEXT
46:NEXT
48 NEXT
50 CHAR1,0,0,"
```

5. program

FELADATOK MEGOLDÁSA, ADATOK ELLENŐRZÉSE

A különböző feladatok megoldásához a képletben megfogalmazott összefüggést az ismert adatoktól függően más és más alakban kell alkalmazni. Azt, hogy mikor, milyen alakra rendezett képlettel kell dolgozni, azt a még oly egyszerű Ohm-törvény esetében is csak egy jól szerkesztett menüvel vagy igen sok IF-utasítás felhasználásával jelölhetjük ki.

Az 1. program olyan megoldást mutat, amelyben az egyenáramú körbe helyezett fogyasztó vagy az áramkör bár-

melyik két jellemző fizikai mennyiségének ismeretében a hiányzó, a kérdezhető másik két adat meghatározható. A program futása közben a keresett, ismeretlen adatok helyett nullát, illetve a Commodore gépek esetében csak RETURN-t kell inputként közölni a géppel. Az A változó számolja az érkezett adatokat és a 160-170. sorok segítségével elbírálja, hogy a feladat egyáltalán megoldható-e a közölt adatok száma alapján. Nem adtunk-e túl sok vagy netán kevés

adatot a számítás elvégzéséhez? A D változó értéke attól függ, hogy melyik két mennyiség ismert. A 190. sor jelöli ki, hogy a képlet melyik alakját kell felhasználni, és azt, hogy a számítás milyen sorrendben kell elvégezni.

A közölt programmal kapcsolatban még annyit, hogy az $S=(U=0)$ alakú értékadások logikai igazság alapján adnak értéket az egyenlet bal oldalán levő változóknak. Amennyiben a zárójelben lévő kifejezés (reláció) igaz, akkor S értéke -1, amennyiben hamis, akkor viszont az érték 0 lesz. (Sinclair-tulajdonosok vigyázzat! Náluk a logikai igaz értéke +1.)

A 2. program az előző gondolatmenet felhasználásával általános háromszög bármely, egymástól független három adatából a háromszög további kiszámítható adatainak meghatározását végzi el.

A program által kiszámított adatok sorát az eredmények kiírató 395-495. programsorok magyarázó szövegeinek átolvasásával ismerjük meg.

Az adatbekérés során a 110-165. sorokban itt is nullát (csak RETURN) kell az ismeretlen adatok helyett a programmal közölni.

A 200. sor az adatok jellegétől függően (három oldal, egy oldal és a rajta fekvő

két szög vagy két oldal és egy szög) más-más ágon biztosítja a program további futását.

A feladat megoldásához a szinusz- és a koszinusz-tétel kerül felhasználásra, hogy ne kelljen minden oldal- és szögkombinációhoz felírni a lehetséges képletvariációkat. Ezért a számítás végző szubrutinokat olyan segédváltozókkal (N, O, P) hívjuk meg, amelyek az ismert adatok alapján kapnak értéket. A szubrutinokból való visszatéréskor a program az eredményt az M változóval hozza.

A listában szereplő KO változó a fokradián átszámításhoz szükséges. A 675. soron az A\$ értékadás a Commodore inverz kiíratáshoz szükséges RVS ON kódja. Az eredménykiírató rutinban ugyanis az indulási, ismert adatok inverz alakban jelennek meg.

Azt, hogy melyik adatok ismertek, azt a P(n) tömbváltozó értéke jelzi az a, b, c, alfa, beta, gamma sorrendben úgy, hogy az INPUT-ként nem nulla értékkel érkező adott sorszámú adat tömb-elemének 1 értéket ad a 665. program-sor.

A 685-695. programsorok csak a lista megfelelő tördelése érdekében képeznek önálló sorokat, azokat a 235-245. sorok folytatásaként is nyugodtan beírhatjuk a programba.

Dr. Sz. Lukács János

```
100 REM OHM
110 D=-2
120 INPUT "U=";U : S=(U<>0):D=D-1*S:A=A-S
130 INPUT "R=";R : S=(R<>0):D=D-2*S:A=A-S
140 INPUT "I=";I : S=(I<>0):D=D-3*S:A=A-S
150 INPUT "P=";P : S=(P<>0):D=D-5*S:A=A-S
160 IF A<2 THEN PRINT"KEVES" :END
170 IF A>2 THEN PRINT"SOK" :END
180 PRINT
190 ON D GOTO 200,210,220,240,230,250
200 I=U/R:GOTO 260
210 R=U/I:GOTO 260
220 U=I*R:GOTO 260
230 U=SQR(P*R)
240 I=P/U:GOTO 210
250 U=P/I:GOTO 210
260 P=U*I
270 PRINT "U=";U;"V":PRINT "I=";I;"A"
280 PRINT "R=";R;"OHM":PRINT "P=";P;"W"
```

1. program

```
100 REM HÁROMSZÖG
105 PRINT CHR$(147) :PRINT :KO=.01745
110 INPUT "A OLDAL=";A:IF A<0 THEN 110
115 M=A :GOSUB 660
120 INPUT "B OLDAL=";B:IF B<0 THEN 120
125 M=B :GOSUB 660
130 INPUT "C OLDAL=";C:IF C<0 THEN 130
135 M=C :GOSUB 660
140 PRINT:PRINT"SZÖGEK FOKOKBAN":PRINT
145 INPUT "ALFA =" ;X:IF X<0 OR X>=180 THEN 145
150 M=X :GOSUB 660
155 INPUT "BETA =" ;Y:IF Y<0 OR Y>=180 THEN 155
160 M=Y :GOSUB 660
165 INPUT "GAMMA =" ;Z:IF Z<0 OR Z>=180 THEN 165
170 M=Z :GOSUB 660
175 IF X+Y>=180 OR X+Z>=180 OR Z+Y>=180 THEN H=1
180 IF H=1 THEN PRINT"HIBA A SZÖGEKNEL" :END
185 IF D<3 OR D=4 OR D=5 OR D=8 THEN H=3
190 IF H=3 THEN PRINT"KEVES ADAT" :END
195 IF D=7 OR D>10 THEN PRINT"SOK ADAT":END
200 ON D/3 GOTO 205,230,295
205 REM HÁROM OLDAL
210 IF X=0 THEN N=B:O=C:P=A:GOSUB 595:X=M
215 IF Y=0 THEN N=A:O=C:P=B:GOSUB 595:Y=M
220 IF Z=0 THEN N=A:O=B:P=C:GOSUB 595:Z=M
225 GOTO 345
230 REM KÉT OLDAL ES KÖZBEZART SZÖG
235 IF A<>0 AND B<>0 AND Z<>0 THEN 685
240 IF B<>0 AND C<>0 AND X<>0 THEN 690
245 IF A<>0 AND C<>0 AND Y<>0 THEN 695
250 REM KÉT OLDAL ES EGY SZÖG
255 IF X<>0 THEN N=A:P=X:GOTO 265
260 GOTO 275
265 IF B<>0 THEN O=B:GOSUB 620:Y=M:GOTO 295
270 O=C:GOSUB 620 :Z=M:GOTO 295
275 IF Y<>0 THEN N=B:P=Y:GOSUB 650:GOTO 285
280 GOTO 290
285 O=C :GOSUB 620:Z=M:GOTO 295
290 N=C:P=Z:GOSUB 650:O=B:GOSUB 620:Y=M
295 REM EGY OLDAL KÉT SZÖG
300 S=180-X-Y-Z:IF X=0 THEN X=S
305 IF Y=0 THEN Y=S
310 IF Z=0 THEN Z=S
315 IF C<>0 THEN N=C:P=Z :GOTO 330
320 IF B<>0 THEN N=B:P=Y :GOTO 330
325 IF A<>0 THEN O=Y:GOSUB 585:B=M:GOTO 340
330 O=X :GOSUB 590:A=M
335 IF B=0 THEN O=Y:GOSUB 585:B=M:GOTO 345
340 O=Z:GOSUB 585:C=M
345 MA=B*SIN(KO*Z):MB=C*SIN(KO*X):MC=A*SIN(KO*Y)
350 N=A/2:O=C:P=Y:GOSUB 615:FA=M
355 N=B/2:O=A:P=Z:GOSUB 615:FB=M
360 N=C/2:O=B:P=X:GOSUB 615:FC=M
365 N=C :O=Y:P=X:GOSUB 640:SA=M
370 N=A :O=Z:P=Y:GOSUB 640:SB=M
375 N=B :O=X:P=Z:GOSUB 640:SC=M
380 KER=A+B+C:TER=A*B*SIN(KO*Z)/2
395 PRINT CHR$(147):PRINT "MEGOLDAS":PRINT
```

```
390 CIM=T1+1
395 GOSUB 675:PRINT"A OLDAL" :=";A$;A
400 GOSUB 675:PRINT"B OLDAL" :=";A$;B
405 GOSUB 675:PRINT"C OLDAL" :=";A$;C
410 GOSUB 675:PRINT"ALFA" :=";A$;X
415 GOSUB 675:PRINT"BETA" :=";A$;Y
420 GOSUB 675:PRINT"GAMMA" :=";A$;Z
425 PRINT
430 PRINT"KERULET" :=";KER
435 PRINT"TERULET" :=";TER
440 PRINT
445 PRINT"MAGASSÁGVONAL A =" ;MA
450 PRINT"MAGASSÁGVONAL B =" ;MB
455 PRINT"MAGASSÁGVONAL C =" ;MC
460 PRINT
465 PRINT"OLDALFELEZO A =" ;FA
470 PRINT"OLDALFELEZO B =" ;FB
475 PRINT"OLDALFELEZO C =" ;FC
480 PRINT
485 PRINT"SZÖGFELEZO A =" ;SA
490 PRINT"SZÖGFELEZO B =" ;SB
495 PRINT"SZÖGFELEZO C =" ;SC
500 IF T0<>2 THEN 575
505 FOR I=1 TO 10 :GET A$:NEXT
510 PRINT"MASODIK MEGOLDAS BILL. NYOMASRA"
515 GET A$:IF A$="" THEN 515
520 T0=0:IF P(E-2)=1 THEN 540
525 IF P(E-1)=1 THEN 550
530 IF P(E-3)=1 THEN B=0:Y=0:GOTO 570
535 A=0:X=0:GOTO 565
540 IF P(E-4)=1 THEN C=0:Z=0:GOTO 565
545 B=0:Y=0:GOTO 560
550 IF P(E-5)=1 THEN C=0:Z=0:GOTO 570
555 A=0:X=0
560 Z=180-Z:GOTO 295
565 Y=180-Y:GOTO 295
570 X=180-X:GOTO 295
575 GET A$:IF A$="" THEN 575
580 END
585 N=A:P=X
590 M=N*SIN(KO*O)/SIN(KO*P):RETURN
595 M=(N^2+O^2-P^2)/(2*N*O)
600 IF M>1 OR M<=-1 THEN H=4
605 IF H=4 THEN PRINT"NINCS MEGOLDAS":END
610 M=(-ATN(M/SQR(1-M^2))+1.57)/KO:RETURN
615 M=SQR(N^2+O^2-2*N*O*COS(KO*P)):RETURN
620 IF N<0 THEN T0=2
625 M=O*SIN(KO/P)/N:IF M=1 THEN M=90:RETURN
630 IF M>1 THEN H=4:GOTO 605
635 M=ATN(M/SQR(1-M^2))/KO :RETURN
640 M=N*SIN(KO*O)/SIN(KO*(180-P/2-O))
645 RETURN
650 IF A<>0 THEN O=A:GOSUB 620:X=M:GOTO 295
655 RETURN
660 E=E+1:IF M=0 THEN RETURN
665 P(E)=1:D=D+1:IF E>=4 THEN D=D+3
670 RETURN
675 A$="":IF P(CIM)=1 THEN A$="^R"
680 CIM=CIM+1 :RETURN
685 N=A:O=B:P=Z:GOSUB 615:C=M:GOTO 210
690 N=B:O=C:P=X:GOSUB 615:A=M:GOTO 210
695 N=A:O=C:P=Y:GOSUB 615:B=M:GOTO 210
```

2. program

SZÁNTÁS-VETÉS A KÉPERNYŐN

A mezőgazdasági szakképzést végző iskolák mindegyikében tananyag az alapvető talajművelés, a szántás. Az eke munkája, a barázdaszelet átfordításának határesetét bemutatható, szemléltethető az itt közölt 1. programmal, amely C-4 számítógép, 3,5 Basic-nyelven íródott, de C 64-en valamelyik grafikus lehetőséget támogató segédprogram (Simon's Basic) használatával is könnyen futtatható.

A program futásának eredményét az 1. ábra mutatja. A program 110-120. során a talajművelő eszköz munkaszélességét és munkamélységét input adatként kerli. Ha az adatok a talajszelvény átfordulását nem biztosítják, akkor a program el sem kezdi a számolást és rajzolást. A 180-190. sorok kiszámítják és kiírják azt a szántási munkamélységadatokat, amely az átfordítás határesetéhez tartozik, és az

őszi mélyszántás normál — legnagyobb felületet biztosító — mélységértékét.

A 250. soron kezdődő szubrutin az eke rajzolja fel, míg a 310-370. sorok közötti programrész a barázdaszelet rajzban történő átfordítását végzi el a különböző fázisok felrajzolásával.

A termesztési technológia magágykészítés utáni fontos művelete a vetés. A vetőgép beállítása igen sok fázisból áll. Az egyértelműen kiszámítható adatok: a felszerelhető vetőelemek száma, a munkaszélesség és a csatlakozó sor beállításához szükséges nyomjelző hossz.

A 2. ábra szerinti futási eredményt a 110-150. soron bekért adatok alapján biztosítja a 2. program.

A bekért adatokkal kapcsolatban célszerű bizonyos ellenőrzéseket elvégezni. A sortáv minimumát az egy sorba egy-

más mellé felhelyezett vetőelem szerkezeti szélessége határozza meg. A gerendely hosszadatát mi itt csak egy egészen értelmetlen esetre ellenőriztük. A traktor mellsőkerék nyomtávértékét a konkrét típus adatainak ismeretében kell ellenőrizni.

A 170-180. sorok tartalmazzák a számítás alapvető összefüggéseit. A 230-410. sorok többé-kevésbé léptékhelyesen felrajzolják a „gépcsoport” alapállapotát.

A 420-470. sorok a visszafelé jövő gépcsoport rajzát készítik el. Ebben a párosos részben a képernyőterület letapogatását végzi el az RDOT (2) függvény. (Más gépeknél erre a célra a POINI szolgál.)

A mezőgazdasági szakirányú képzés céljaira egy sor program készen is kap-

ható, ezek egyike felé az alapvető gépészeti erőgépek ismeretére vonatkozó oktatási, illetve számonkérő, feleltető program. A másik csoportban azok a programok találhatók, amelyek a növénytermesztés, illetve állattenyésztés adatbázisainak használatára készítik fel a program használatát.

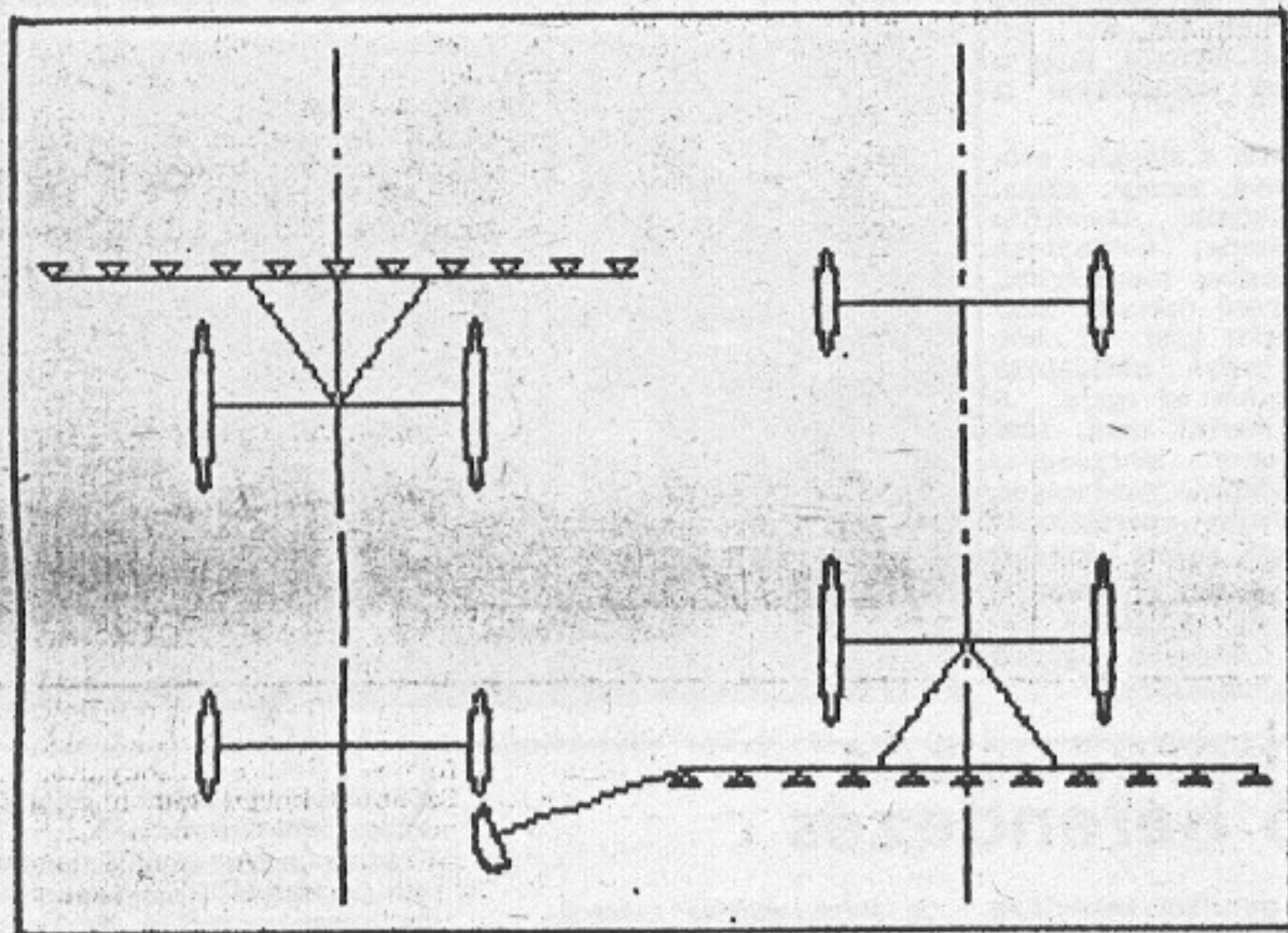
Az adatbázis-kezelő programok — bármely egyszerű célra is készülnek —, csak magnóval komolyabb háttértár alkalmazásával adhatnak kielégítő megbízhatóságú és sebességű megoldást.

Ilyen programokat mi magunk is összeállítottunk, és ha a kezelő feltöltő kezet program elég rugalmas, akkor csak — (csak?) — az adatállomány feltöltésének hangsúlyos részét munkáját kell a különböző feladatokhoz elvégezni.

Dr. Sz. Lukács János



1. ábra



2. ábra

```
100 GRAPHIC2,1 :REM EKE MUNKAJA
110 INPUT "SZÉLESSEG";B
120 INPUT "MELYSEG ";A
130 IF B/A>1.26 THEN GOTO 150
140 PRINT "NEM FORDUL AT":GOTO 110
150 S=SQR(B*B-A*A):L=ATN(S/A)*57.3
160 DRAW 1,0,100 TO 319,100
170 DRAW 1,0,100-A TO 250,100-A
180 CHAR,0,0,"HATAR :"+STR$(INT(B/1.27))
190 CHAR,0,2,"NORMAL :"+STR$(INT(B/1.41))
200 FOR I=250 TO 10 STEP -B
210 Q=0:X=I+B:GOSUB250
220 Q=1:X=I :GOSUB250:GOSUB310:GOSUB380
230 NEXT
240 END
250 DRAW Q,X,99TOX,94-B*.8
260 DRAW Q,X,99TOX+B,99TOX+B-6,94TOX,94
270 DRAW Q,X,94-B*.8 TOX+B,80-B*.8
280 CIRCLEQ,X+B,90-B*.8,10,,350,140,,20
290 R=RDOT(0):T=RDOT(1)
300 DRAW Q,R,T TO X+B-6,94 :RETURN
310 BOX 1,I,100-A,I+B,99 :GOSUB 380
320 BOX 0,I,100-A,I+B,99 :GOSUB 250
330 BOX 1,I+B,100-B,I+B+A,99 :GOSUB 380
340 BOX 0,I+B,100-B,I+B+A,99
350 DRAW1,I+B+A,100TOB:LTOA:L+270
360 DRAW TO B:L+180 TO A:L+90:GOSUB 380
370 RETURN
380 FOR T= 1 TO 200:NEXT :RETURN
```

1. program

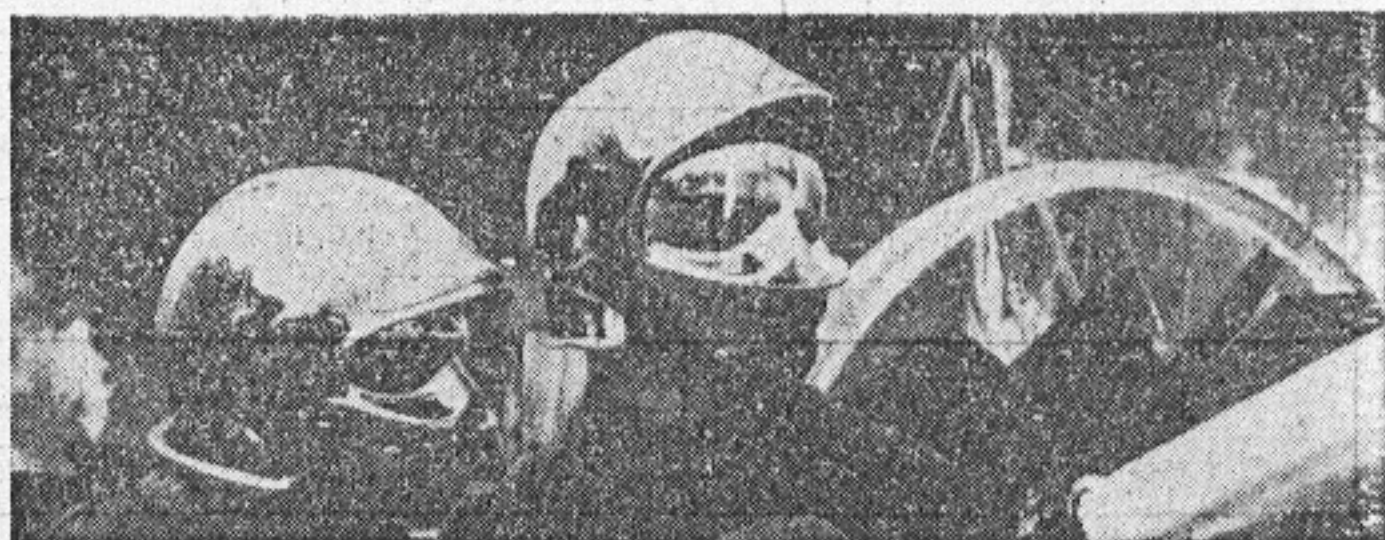
```
100 REM VETOGEPI BEALLITASA
110 INPUT"SORTAV ";T
120 IF T<10 OR T>100 THEN GOTO 110
130 INPUT"GERENDELY ";L
140 IF L<T THEN PRINT"HIBAS":GOTO 130
150 INPUT"TR. NYOMTAV":A
160 IF A<100 OR A>200 THEN GOTO 150
170 N=INT(L/T)+1:B=N*T:P=(N-1)*T/2
180 NJ=B-A/2-P
190 PRINT"CSOROSZLYASZAM="N"DB"
200 PRINT"MUNKASZÉLESSEG="B
210 PRINT"NYOMJELZO ="NJ
220 GETKEY A$:GRAPHIC1,1
230 DRAW1,0,0TO0,20:DRAW1,0,24TO0,26
240 SSHAPE A$,0,0,1,30:DRAW0,0,0TO0,30
250 FOR Y=0 TO 180 STEP 30
260 GSHAPE A$,100,Y :GSHAPE A$,100+B/2,Y
270 NEXT
280 DRAW1,100-L/4,30TO100+L/4,30
290 FOR V=100-P/2 TO 100+P/2STEP T/2
300 DRAW1,V,29TOV+3,26TOV-3,26TOV,29
310 NEXT
320 DRAW1,80,30TO100,60TO120,30
330 CIRCLE,100-A/4,60,2,20
340 CIRCLE,100+A/4,60,2,20
350 CIRCLE,100-A/4,140,2,12
360 CIRCLE,100+A/4,140,2,12
370 DRAW1,102-A/4,60TO98+A/4,60
380 DRAW1,102-A/4,140TO98+A/4,140
390 DRAW1,100+L/4,30TO NJ/2;70
400 X=RDOT(0):Y=RDOT(1)
410 CIRCLE1,X,Y,4,8,330,150:DRAWTO15;340
420 FOR Y=0 TO 132
430 FOR X=-P/2-3 TO P/2+3
440 LOCATE 100+X,26+Y:W=RDOT(2)
450 IF W=0 THEN 470
460 DRAW1,100+B/2-X,150-Y
470 NEXT: NEXT
```

2. program

Fémbevonatú műanyag

A párizsi tűzoltóság az ábrán látható kivitelű, fémestett műanyagból készült tűzoltósíkok használatát rendszeresítette. A síkokat a francia Société CGF Gallet Rt. fröccsöntéssel gyártja a Bayer vállalat ABC polikarbon elemből. Az elegyenek nagy a hőellenálló-képessége, ütési szívóssága, könnyen feldolgozható és fémmel bevonható.

A sisakeszézből és a felsapható sisakrostélyból álló sisakot kalakítása után galvanikusan rezezik, majd nikkelezik és végül fényesre polírozzák. A fémréteg elsődlegesen a hő visszaverésére, és ezáltal a tűzoltók fokozottabb védelmének biztosítására szolgál. A fémesen csillogó sisak tetszetős, és lényegesen kényelmesebb, mert könnyebb.



MÉRŐESZKÖZÖK LEOLVASÁSA

A számítógéppel veszélyhelyzetek elhárítása, gépek, berendezések kezelése sajátítható el anélkül, hogy a tanuló téves döntése kárt okozna. A középfokú oktatás folyamán egyszerűbb eszközök, műszerek kezelésére, leolvasására kell megtanítani, felkészíteni a tanulókat.

Az 1. program tizedes pontosságú tolmérce leolvasásának gyakorlásához nyújt segítséget. A program futása közben a 110–140. sorok és a meghívott szubrutinok segítségével az 1. ábra szerinti képernyőképet biztosítja.

A véletlenszerűen felrajzolt alakú és

méretű alkatrész méretét kell leolvasni a rajzról és inputként be kell írni a programba. A 150–170. sorok minősítik a beírt leolvasási eredményt. A program nem enged tovább a felhasználót, amíg helyes eredményt nem kap.

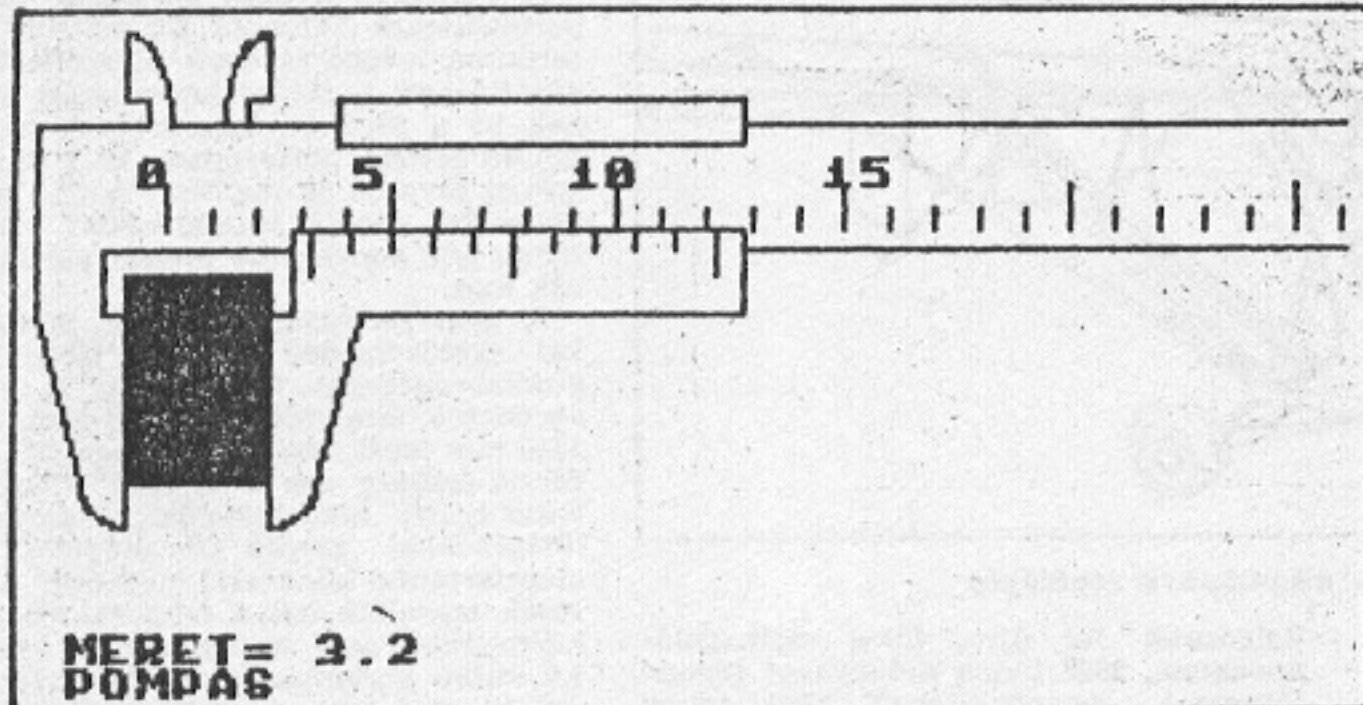
Az elfogadható méretű és leolvasást biztosító fő- és segédszálak felrajzolását az adott számítógép képpontjainak száma határozza meg. A nívószál hossza 0,1 mm pontosságú leolvasáshoz 9×10 raszterpont hosszúságú kell legyen. Az ötszázados tolmérő nívószálának felrajzolása 19×20 képpont hosszúságú lenne. Ezt a Commodore gépek felbontóképessége nem teszi lehetővé.

A körmutatós műszer skáláján mutatott érték leolvasását a 2. ábrán látható kép segítségével gyakoroltathatjuk.

A 2. program 110–190. sorai rajzolják fel a műszer képét. A 200. sor állítja elő a véletlenszerűen kijelzendő értéket. A mutató kilendülésének rajzi megvalósítását a 210–240. sorok biztosítják. A mutató által jelzett leolvasott érték a 250. soron közölhető a programmal. A 260–290. sorok minősítik a beírt értéket, majd rajzilag visszakerüli a mutató az alapállásba.

A helyes mérési sorrend, az univerzális műszer üzemmód és méréshatár választásának, tennivalóinak gyakorlására összetettebb program szükséges. Ilyen jellegű gyakorló programok C-4 és C-64 gépekre a TUDORG-nál és az OPI-nál beszerezhetők, de a vállalkozó kedvűek maguk is összeállíthatnak ilyen programokat.

Dr. Sz. Lukács János

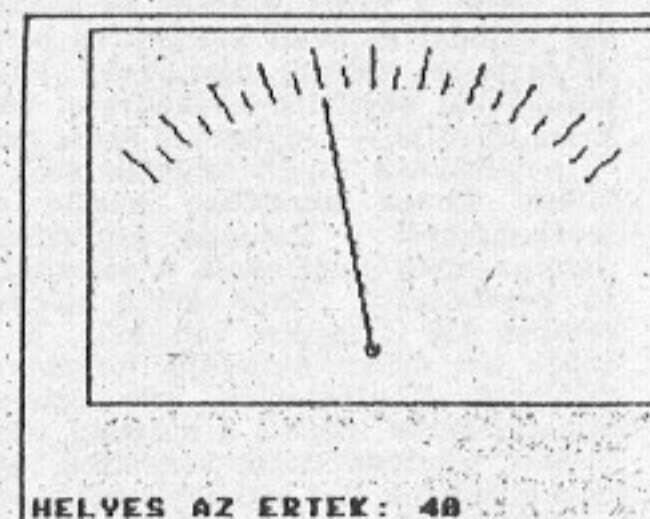


1. ábra

```

100 GRAPHIC 2,1 :REM TOLOMERCE GYAKORLO
110 GOSUB 200
120 M=(INT(200*RND(0)))/10 :X=10*M+30
130 ON INT(1+3*RND(0)) GOSUB 520,530,570
140 GOSUB 360
150 INPUT "MERET=";N
160 IFM<>NTHENPRINT"HIBAS LEOLV":GOTO150
170 PRINT"POMPAS"
180 PRINT"BILL.NYOMASRA UJ FELADAT"
190 GETKEY A$:RUN
200 REM ALLORESZ
210 DRAW 1,300,55TO40,55
220 CIRCLE 1,30,55,10,20,0,90
230 DRAW 1,30,35TO30,50TO35,50TO35,55
240 DRAW TO10,55TO10,100TO20,140
250 CIRCLE 1,30,140,10,180,270
260 DRAW 1,30,150TO30,100TO25,100TO25,85
270 DRAW TO 300,85
280 FOR V=39 TO 300 STEP 10
290 DRAW 1,V,75TOV,80
300 NEXT
310 CHAR 1,4,8,"0 5 10 15"
320 FOR V=39TO300STEP50
330 DRAW 1,V,70TOV,75
340 NEXT :RETURN
350 REM MOZGO
360 DRAW 1,X+105,100TOX+105,80TOX+5,80
370 DRAW TO X+5,100TOX,100TOX,150
380 CIRCLE 1,X,140,10,90,180
390 DRAW 1,X+10,140TOX+20,100TOX+105,100
400 DRAW 0,X+6,85TOX+104,85
410 BOX 1,X+15,50,X+105,60
420 DRAW 0,X+16,55TOX+104,55
430 DRAW 1,X-5,55TOX-5,50TOX,50TOX,35
440 CIRCLE 1,X,55,10,20,270,360
450 FOR V=X+9 TO X+105 STEP 9
460 DRAW1,V,80TOV,85
470 NEXT
480 DRAW 1,X+9,85TOX+9,90
490 DRAW 1,X+54,85TOX+54,90
500 DRAW 1,X+99,85TOX+99,90
510 RETURN
520 BOX1,30,90,X,140,,1 :RETURN
530 IF M<3 THEN RETURN
540 BOX 1,20,20,30,55,,1
550 BOX 1,X,20,10+X,55,,1
560 DRAW1,20,20 TO X,20 :RETURN
570 IF M>15 THEN RETURN
580 CIRCLE1,30+5*M,130,5*M,45
590 PAINT 1,31,130 :RETURN
    
```

1. program



2. ábra

```

100 GRAPHIC2,1 :REM KORMUTATOS LEOLV
110 BOX1,60,0,260,140
120 FOR B=-50 TO 50 STEP 5
130 A=B*0.01745:H=7
140 IFB/10=INT(B/10)THEN H=15
150 X=160+100*SIN(A):Y=120-100*COS(A)
160 DRAW1,X,Y TO H;B+360
170 NEXT
180 CIRCLE1,160,120,2,,,,,20
190 DRAW1,160,120 TO 96;310
200 Z=INT(100*RND(0))-50
210 FOR I=-50 TO Z
220 DRAW0,160,120 TO 96;I+359
230 DRAW1,160,120 TO 96;I+360
240 NEXT
250 INPUT"ERTEK";W
260 IF ABS(W-Z-50)<2 THEN GOTO 280
270 PRINT"PONTATLAN LEOLVASAS":GOTO 250
280 IF W=Z+50 THEN PRINT"EZAZ":GOTO 300
290 PRINT"MAJDNEM, PONTOSAN:";Z+50
300 GETKEY A$
310 FOR I=Z TO-50 STEP -1
320 DRAW0,160,120 TO 96;I+361
330 DRAW1,160,120 TO 96;I+360
340 NEXT
350 GETKEY A$
360 PRINT:PRINT:PRINT:PRINT:PRINT
370 GOTO 200
    
```

2. program

Olcsó űrrakéta

Kisméretű műholdak alacsony föld körüli pályára történő felbocsátására tervezték a LittleLeo névre keresztelt olcsó rakétát. Az elképzelések szerint első útjára 1991-ben indul majd a norvégiai Andøyából.

Az új rakéta a jelenleg használatosokhoz viszonyítva 5...10-szeres költségcsökkentést tesz lehetővé. Ez a költségcsökkentés pedig megnyitja az utat számos, nem túl gazdag ország számára az űrkutatásban való részvételre. Az elkövetkezendő húsz évre azt várják, hogy az indított műholdak 50%-a a Földhöz közeli pályán helyezkedik majd el.

Kipróbált szilárd tüzelőanyagú hajtással és bevált, viszonylag olcsó alkatrészekkel ez a rakéta nyereséges üzleti vállalkozás alapjának is tekinthető. Az űrkutatás egyre növekszik, és a tudósok számára az általuk megszabott időben kell hordozórakétát biztosítani. Az angol General Technology Systems cég munkatársai, akik kifejlesztették a LittleLeo-rakétát, azt tervezik, hogy teljes körű szolgáltatást is biztosítsanak. Ez azt jelenti, hogy adott hasznos teherre építik meg a rakétákat, valamint a Földre vissza is hoznak tárgyakat, például feldolgozott anyagokat. Az űrkutatás terén ugyanis azt várják, hogy az a világűrben való tevékenységet is átfogja majd.



VAS-SZÉN ÁLLAPOTÁBRA

A műszaki iskolákban majdnem minden diák retteg a vas-szén állapotábrától, hiszen ha megtanulja, akkor sem biztos az elfajátás mértékében. Az itt közölt program elősegíti a Fe-C ábrával való megbarátkozást, lehetővé teszi a lehűlés közben kialakuló szövetszerkezetek megtanulását, és mint feleltetőprogram ellenőrző is a tudást. Így a tanuló már nyugodtabban nézhet a felelés, a tanári szeméremelés elé.

Nézzük akkor a programot: azok a programsorok, amelyek a listában két-tősponttal kezdődnek, a feleltetőág esetében szükségesek, a tanulóág működéséhez be sem kell gépelni.

A 110-150. sorok a DATA sorokban lévő adatokat a megfelelő tömbökbe töltik át.

A 160-180. sorok az igen szerény memóriaigényű választást a tanulási vagy feleltetőág kiválasztását biztosítják. Bármely ágon X betű válaszként való adásával ide térhetünk vissza (200. és 580. sorok).

A 360. soron kezdődő szubrutin készíti el a képernyőn megjelenő ábrát.

A 200. soron beírt százalékos összetételnek megfelelően arányos helyen megjelenik az ötvöztetjőző vonal kezdőpontja. Az ötvöztetjőző vonal a hőmérséklet csökkenését feltételezve fut végig az ábrán.

Valahányszor változás (át-kristályosodás) következik be, ami az állapotábra valamely vonalának metszésében ismerhető fel, a program leáll.

A 340. soron kezdődő szubrutin kiírja a szövetszerkezettel kapcsolatos minimális információkat, és billentyűnyomásra vár.

A lehűlés befejeztével újabb ötvöztetjőző vonal a T betű lenyomása után kérhető a programtól.

A feleltetőágon a 750. soron megadott kérdésszám alapján véletlenszerűen ad kérdést a program és várja a helyes választ.

A program számontartja a jó és hibás

válaszok számát. Hibája, hogy ugyanazt a kérdést többször feladhatja. Ez kivédhető, ha az 550-640. sorok közti programterületet átírjuk.

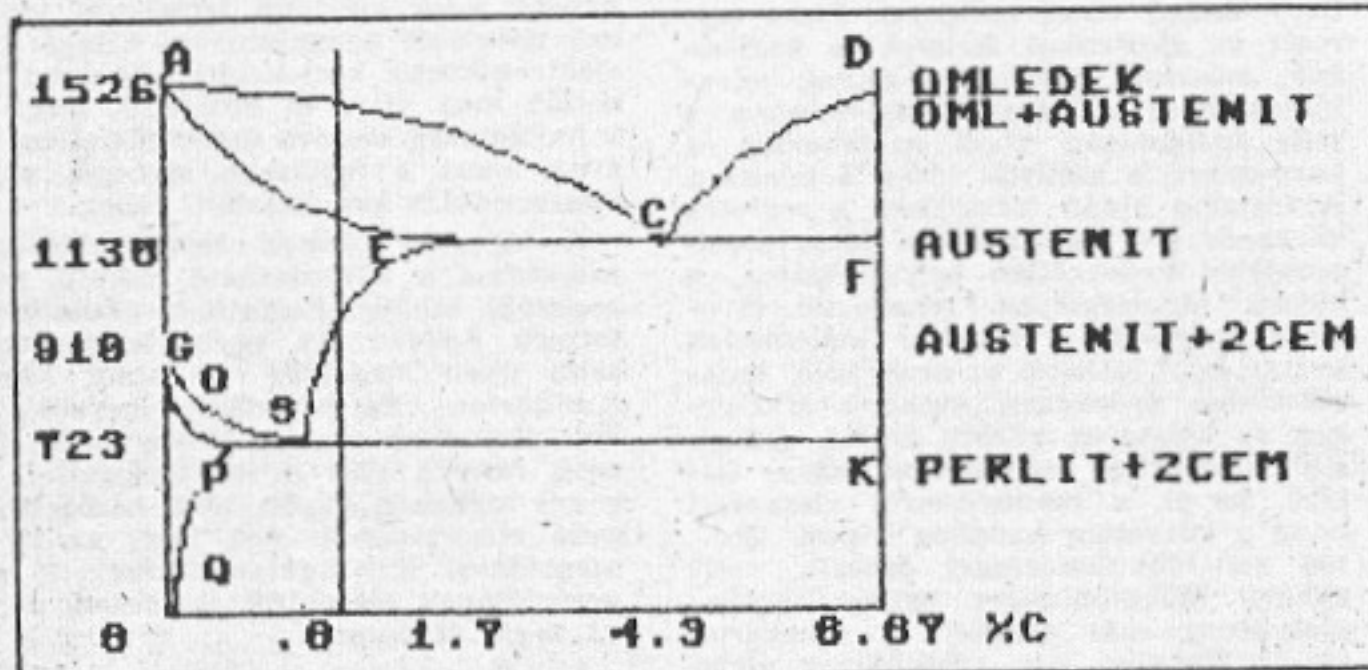
A ciklusban adott kérdések számát például be kell határolni úgy, hogy a DATA sorokba felvett kérdések számának kétharmadánál többet egy sorozatban ne kérdezzon a program. Ugyanis ha minden kérdést fel akarjuk adni, akkor egyre többet kell várni, míg a még fel nem tett kérdést megtalálja a program.

A VOLT (X) tömbbe a már kért kérdések sorszámaival tároljuk. (Hasonló megoldást kell alkalmazni a legegyszerűbb lottósorsoló program esetében is.)

A kérdésekkel kapcsolatos DATA sorok tovább bővíthetők, illetve finomíthatók.

Ne sajnáljuk a fáradságot, hátha tényleg könnyebben eredményesen meg lehet tanulni a számítógép segítségével ezt a tananyagot.

Dr. Sz. Lukács János



```
100 REM VAS-KARBON ALLAPOTABRA
110 FOR L=1 TO 5 :FOR J=1 TO 6
120 READS$(L,J)
130 NEXT :NEXT
140 :READ K :DIM K$(K),F$(K)
150 :FOR I=1 TO K:READ K$(I),F$(I):NEXT
160 :INPUT "TANUL T, FELEL F";V$
170 :IF V$="T" THEN GOTO 190
180 :IF V$="F" THEN GOTO 550:ELSE GOTO 160
190 GOSUB 360
200 INPUT "C%=";S$:IF S$="X" THEN GOTO 160
210 S=VAL(S$):IFS<=0 OR S>=6.67 THEN 200
220 IFS>4.2 THEN X=180:L=5
230 IFS<4.3 THEN X=120:L=4
240 IFS<1.7 THEN X=73:L=3
250 IFS<0.83 THEN X=48:L=2
260 IFS<0.4 THEN X=39:L=1
270 J=0 :DRAW1,X,2
280 FOR Y=2 TO 141
290 LOCATE X,Y:IF RDOT(2) THEN GOSUB 340
300 DRAW1,X,Y
310 NEXT :PRINT:PRINT "T UJ FELADAT"
320 GETKEY A$:IF A$="T" THEN 190
330 GOTO 320
340 J=J+1:CHAR1,26,1+Y/8,S$(L,J)
350 GETKEY A$:RETURN
360 GRAPHIC2,1:REM FEC ALLAPOTABRA
370 DRAW1,32,8 TO 32,142 TO 200,142 TO 200,8
380 DRAW,100,50 TO 200,50:DRAW,45,100 TO 200,100
390 CIRCLE1,32,210,200,,0,36:REM AC
400 CIRCLE1,224,100,90,,303,344:REM CD
410 CIRCLE1,100,-42,77,,180,240:REM AE
420 CIRCLE1,100,100,35,50,270,360:REM SE
430 CIRCLE1,47,80,15,20,180,270:REM GK
440 CIRCLE1,65,71,33,28,180,270:REM GS
450 CIRCLE1,50,142,16,42,270,350:REM QK
460 DRAW0,39,112 TO 39,124
470 CHAR1,0,1,"1526":CHAR1,0,12,"723"
480 CHAR1,0,6,"1130":CHAR1,0,9,"910"
490 CHAR,2,18,"0.8 1.7 4.3 6.67 %C"
500 :IF V$="F" THEN RETURN
510 CHAR,4,0,"A":CHAR,18,5,"C":CHAR,24,0,"D"
520 CHAR,5,10,"O":CHAR,7,11,"S":CHAR,4,9,"G"
530 CHAR,10,6,"E":CHAR,24,7,"F":CHAR,5,16,"Q"
540 CHAR,5,13,"P":CHAR,24,13,"K":RETURN
```

```
550 :GOSUB 360 :J=0:R=0:REM FELELTET
560 :X=INT(1+K*RND(0)):PRINT:PRINT
570 :PRINTK$(X):INPUT F$
580 :IF F$="X" THEN GOTO 160
590 :IF F$=F$(X) THEN J=J+1:GOTO 620
600 :IF F$<>F$(X) THEN R=R+1
610 :PRINT "A HELYES VALASZ: "F$(X)
620 :CHAR,26,2,"EDDIG JO="+STR$(J)
630 :CHAR,29,4,"HIBAS="+STR$(R)
640 :GOTO 560
```

```
650 DATA OMLEDEK,OML+AUSTENIT,AUSTENIT
660 DATA AUSTENIT+FERRIT,FERRIT,FERRIT+3CEM
670 DATA OMLEDEK,OML+AUSTENIT,AUSTENIT
680 DATA AUSTENIT+FERRIT,FERRIT+PERLIT,
690 DATA OMLEDEK,OML+AUSTENIT,AUSTENIT
700 DATA AUSTENIT+2CEM,PERLIT+2CEM,
710 DATA OMLEDEK,OML+AUSTENIT
720 DATA EUT.+AUST.+2C,EUT.+PERLIT+2C,,
730 DATA OMLEDEK,OML+CEMENTIT
740 DATA LEDEBURIT+CEM,LEDEBURIT+CEM..
```

```
750 :DATA 20:REM KERDESEK SZAMA
760 :DATA A LEGKEMENYEBB SZOVETEM NEVE,FERRIT
770 :DATA DERMEDES KEZDETET JELZO VONAL,ACD
780 :DATA TISZTA AUSTENIT TARTOMANYA,GOSEA
790 :DATA A LEGKEMENYEBB SZOVETEM,CEMENTIT
800 :DATA EUTEKTIKUM SZOVETSZERKEZETE,LEDEBURIT
810 :DATA AZ EUTEKTIKUM OLV.PONTJA,1130
820 :DATA AUSTENIT-FERRIT ATALAKULAS TARTOMANYA,GOSPG
830 :DATA AZ EUTEKTIDOIDOS SZOVETSZERK.,PERLIT
840 :DATA MELY TARTOMANY FELETT VAN CSAK OMLEDEK,ACD
850 :DATA AUSTENIT PERLITTE ALAKULAS HOMERSEKLETE,723
860 :DATA AUSTENIT PERLITTE ALAKULAS AA.VONALA,PK
870 :DATA A1 HOMERSEKLET,1526,A3 HOMERSEKLET,910
880 :DATA MI VALIK KI AZ SE VONAL MENTEN,CEMENTIT
890 :DATA A DERMEDES BEFEJEZESUT MUTATO VONAL,AECD
900 :DATA A CEMENTIT SZENTARTALMA,2.67
910 :DATA AZ EUTEKTIKUM SZENTARTALMA,4.3
920 :DATA A LEGKEMENYEBB SZOVETEM,CEMENTIT
930 :DATA AUSTENITBEN A VAS TERRACSA,LAPKOZEPESES
940 :DATA PERLITBEN A VAS TERRACSA,TERKOZEPESES
950 :DATA MI DERMED KI 4.3%FELETT,CEMENTIT
```

Stabilizált hab tűzoltáshoz

Bár szigorodnak a rendszabályok, mégis gyakran előfordul, hogy veszélyes anyagokat szállító járműveket ér baleset, és akkor ezek az anyagok a környezetbe jutnak. Mások szivárgás vagy emberi mulasztás okozta bajt. Ilyen esetekben a legrovidebb időn belül intézkedni kell a veszély elhárításáról. Jelenleg erre az egyik legjobb megoldás a hab és haboltószert használata. Ez befedi a veszélyes anyagot és megakadályozza terjedését, égését. Sajnos, az ilyen habok nagy hátránya, hogy viszonylag hamar megszűnik stabilitásuk, és akkor egyszerűen „összeroppannak”, elvesztik hatásukat.

A 3M Deutschland cégnek sikerült stabilizált habot előállítani. FS-70000 típusú habstabilizátorából elegendő hatóanyagot keverni, ahhoz hogy hosszú ideig megmaradjon habtakarót kapjanak. A habstabilizátor ugyanis nagyon rövid idő alatt zselésíti a kijuttatott habot, és ezzel a benne lévő vizet sokáig megköti. Ezért ez a habfajta kitűnően alkalmas éghető és/vagy toxikus folyadékok, szilárd anyagok és porok betakarásához. Így módon megakadályozza ezeknek az anyagoknak a meggyulladását és gázoknak az atmoszférába jutását.

Maga a stabilizált hab rugalmas és gáztömör. A habtakaró kevés gőzt és

gázt vesz fel az alatta lévő anyagokból, ezért a bomlás utáni eltakarítása lényegesen egyszerűbb. Régebbi veszélyes hulladékok, anyagok szanálásakor szintén előnyösen használható a stabilizált hab. Szennyezett talajok kiemelése után például be szokták fedni a kiemelt földet és a gödröt. Mindkét célra nagyszerűen megfelel a stabilizált hab, amely max. hét napon keresztül marad meg hatásosan. E habfajta az előzőektől kissé eltérő, FX-9161 jelű habstabilizátora van.

Lakótelepek közelében lévő hulladéklerakó-helyekre számos országban előírás, hogy a napi lerakás befejezése után ideiglenes takaróréteget hordjanak fel a hulladék szétszóródásának és a szag terjedésének megakadályozására. E célra többnyire építési törmelék, homokot vagy földet használnak. A takaróréteg napi felvitele igen drága és ráadásul jelentős helyet vesz el a hulladék elől. Ezért helyette a 3M speciális stabilizált habot javasol. Ennek stabilizátora EC-9400 típusú. Mivel túlnyomó részt csak levegőt és vizet tartalmaz, az új hulladékréteg terítése után a légbuborékok megszűnnek a terhelés miatt, viszont a gyakorlatilag tömörített cellafalak megmaradnak. Ez a megoldás tehát számos előnnyel jár.

POLAROID FOTÓK AZ IGAZOLVÁNYBAN

Magyarországon a polaroidtechnikával készült képeket még néhány évvel ezelőtt is csodálattal vegyes óvatossággal kezelték. Széles körű elterjedésüket, annak ellenére, hogy a szakemberek használhatóságát egyértelműen elismerték, elsősorban viszonylag magas ára gátolta. A képek minőségének megítélése döntően szubjektív vélemények, semmint korrekt mérések alapján történt. Mendedák keringtek foltosodó, eltűnő képekről. Mivel kevés szakember számára adatott meg az állítások felülvizsgálatának lehetősége, az azonnali papírképek előhívási sebességének elismerése mellett a gondolatokban az instabilitás kötődött.

Gyors térhódítás

Eközben a világ többi táján, elsősorban az USA-ban, Kanadában és Nyugat-Európában a polaroid technika hihetetlen gyorsasággal hódított tért az ipar, a tudomány és a kultúra legkülönbözőbb területein.

Egyik jellegzetes, a lakosság által leginkább megkedvelt felhasználási terület az igazolványfotók. 1988-ig a polaroid igazolványfotókat a világ 79 állama — köztük az USA, Ausztrália, Nagy-Britannia, Japán stb. — nyilvánította technikailag alkalmasnak különböző biztonsági okmányokhoz.

1989-ben a Belügyminisztérium Igazgatásrendészeti Csoportfőnöksége — fotótechnikában érintett vállalatok, szövetkezetek, kisiparosok kérésére — minden részletre kiterjedő alapos vizsgálatot követően engedélyezte a polaroid fényképek felhasználását személyi okmányok készítéséhez. A döntés eredményeként ezentűl az ország valamennyi rendőrkapitánysága elfogadja az útlevél, személyi igazolvány és jogosítvány készítéséhez leadott, méretre vágott, fekete-fehér polaroid fotókat. A látványos külsőségek nélkül hozott határozat elsősorban a falvakban, kisvárosokban élő lakosság részére jelent komoly segítséget, hiszen egy-egy fényképért sok esetben a másik faluba vagy közeli városba kellett utazni, esetleg többször is.

A polaroid anyagok alkalmazása esetén nincs szükség várakozásra, fölösleges utazgatásra annak érdekében, hogy bárki az ügyintézéshez fényképet készíthessen. Ezek előhívásához ugyanis mindössze 1 perc van szükség. Igaz, a fotók ára meghaladja a hagyományos technikával készült képeket, viszont az idő, a minőség és az ügyintézés egyszerűsége sok mindenért kárpótolt. A gyors előhívás kizárja az esetleges hibás exponálás vagy hívás okozta bonyodalmakat.

Kétféle film

Az igazolványképekkel szemben támasztott igényeket az MSZ 16 970-85 számú szabvány rögzíti. A Polaroid cég igazolványfotókhoz elsősorban a 667 és 669 típusú filmjeit ajánlja. A 667 típusú fekete-fehér film érzékenysége 36 DIN (3000 ASA). Közepes kontrasztú, felbontóképessége 12...14 vonalpár/mm. Hívási ideje a hőmérséklet növekedésével fordítottan arányos.

A Polaroid 669 típusú színes film érzékenysége 20 DIN (80 ASA). Az optimális képstabilitás mindkét film esetén 24°C-on 1 perc hívási idő alatt érhető el. A hasznos képméret 7,3×9,5 cm. Magyarországon jelenleg csak a 667 típusú filmre készült igazolványképek használhatók ilyen célra.

A Polaroid által forgalmazott Mini-portrait 402, 403 és a 404 típusú kamerákkal ezekre az anyagokra egyidőben négy fotó készíthető. Felvetődik a kérdés: miért érdemes annyit foglalkozni egy egyszerű fotóval, mi különbözteti meg a többi fekete-fehér fotótól?

A legalapvetőbb eltérés éppen rendeltetéséből adódik. Más az elvárás egy művészeti alkotásért, esztétikai élményt nyújtó portréval és már az igazolványképpel szemben. Az útlevélképek rendeltetészerű használat során csekély fény-, hő-, vegyi- és mechanikai hatásnak vannak kitéve. Lényeges azonban, hogy 10...15 év használat esetén is lehető kell tennie a megbízható azonosítást. A személyi igazolványt az emberek többsége nap mint nap magával hordja, emiatt a bennük elhelyezett fényképek élettartamát döntően a kezeléstől függő kopthatóság, a hőmérséklet és az izzadság befolyásolja. Ilyen körülmények között a

fotóktól elvárt minimális élettartam 7...10 év.

Különböző fotók esetében általában, de az igazolványfotók esetében különösen fontos, hogy a képek stabilitása garantált legyen, függetlenül attól, hogy polaroid vagy hagyományos eljárással készültek. Ezeknek a követelményeknek eleget tenni nem is olyan egyszerű feladat. Kutatásaival szinte valamennyi fotóanyaggyártó cég a képstabilitás problémáinak mind tökéletesebb megoldására törekszik. A fejlesztések a fekete-fehér anyagok mellett kiterjednek a színes filmek, papírok és vegyszerek körére is, mivel a színes igazolványfotókat a világ minden táján elterjedten használják.

Az ideális papírkép korlátlan ideig megőrzi tónusárnyalatát és részletgazdagságát. Ez tehát a nagy cél. Megoldandó probléma még ma is a fotókémiai folyamatok során keletkező vegyszermaradványok kivonása vagy közömbösítése, a fotók előhívása után a tárolás körülményeitől, a külső vegyi hatásoktól való függetlenség biztosítása.

Az élettartam vizsgálata

A rögtönkész fényképezés feltalálása óta a Polaroid cég folyamatosan javítja képeinek minőségét, és élen jár a felhasználási cél által támasztott funkcionális követelmények vizsgálatában. Mivel az ipari szabványok nem minden esetben fogalmaznak meg pontosan, hogy különböző fotók esetében milyen körülmények között, mennyi idő alatt, milyen mértékű minőségromlás fogadható el, egyértelmű kidolgozása és az elvek elfogadtatása nélkülözhetetlen. A képállandóság minimális szintjének meghatározásában és a mérési módszerek kidolgozásában a Polaroid cég figyelemre méltó vizsgálatokat végzett. Ezeket főként saját filmjeinek objektív, mérőszámokkal igazolható minőségellenőrzésére fejlesztette ki.

A tesztelés során vizsgálták a magas hőmérséklet, a nagy és kicsi páratartalom, a külső kémiai, illetve az erős fény, valamint ezen paraméterek kombinációjának hatását a polaroidfotók élettartamára.

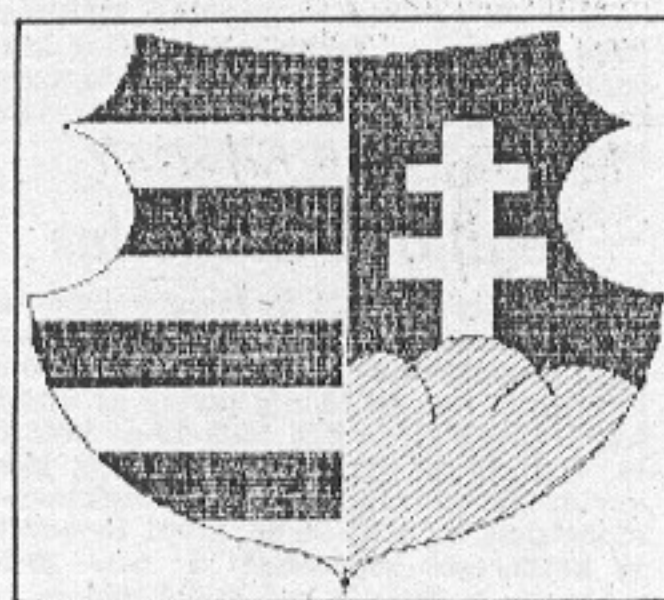
A vizsgálatok azt igazolták, hogy a Polaroid 667 típusú film természetes fényen történő, 30 napig tartó megvilágításra a hagyományos anyagokkal azonos módon reagál, míg a hasonló technológiájú instant filmek fénynek kitett részén erős sárgulás jelentkezett. A minták mesterséges fényvel (xenon ívfény) 5 napon át tartó megvilágítása a más gyártók által előállított filmen 0,08 denzitásnövekedést okozott. A Polaroid 667 denzitása eközben változatlan maradt. Nem befolyásolta minőségét a gyorsított (10 napig tartó) öregedési vizsgálat sem. 37,8°C (100 Farenheit) és 96% relatív páratartalom hatására ugyanakkor a konkurens instant filmek keretei, illetve a képek semleges fehér felületei elszárgultak. Igazolódott továbbá, hogy a más, azonnal kész filmekre készült felvételek elkészültükkor nagyobb denzitást mutattak és túlhívás esetén sem növekszik olyan mértékben kontrasztjuk, mint a Polaroid 667-es anyagé.

A színes filmeket annak ellenére, hogy a biztonsági okmányfotók döntően könnyszerűen összehajtott állapotban, tehát sötétben vannak tárolva, természetes fényben (napon) 30 napig tartó vizsgálatnak vetették alá. Fénynek kitett felületek a hagyományos eljárással készült fotókhoz képest alig érzékelhetően vesztek színsűrűségükből. A más gyártótól származó azonnali képek minőségromlása elsősorban a kék és bíbor színezékben jelentős volt. Főként ennek kapcsán vetődött fel a képstabilitás szabványosításának szükségessége. A fotók információ tartalmának gyors elvesztése — a felhasználó bizalmának megrendülésén kívül — éppen arra tenné alkalmatlanná, amire pl. igazgatásrendészeti, gyógyászati területen alkalmazták; személyazonosításra, dokumentálásra.

A mesterséges fényforrással 320 órán át tartó megvilágítás hatására a konkurens filmek jelentős színtorzulást szenvedtek. A Polaroid 669 típusú film csupán színtelítettségéből vettett kiismertékben, hasonlóan a hagyományos eljárással készült képekhez.

A vizsgálatok ráirányították a figyelmet arra is, hogy a tesztelt használt konkurens anyagok denzitásértékei elkészültükkor meghaladták a Polaroid 669-

RAJZOLTASSUNK KOSSUTH-CÍMERT



A Magyar Közlöny idén október 23-án megjelent 74. száma az Alkotmány módosításáról szóló törvényt ismerteti. Az utolsó oldalon nemzetiszínű keretbe foglalva a Magyar Népköztársaság majdan népszavazással kiválasztásra kerülő címerének változatai vannak.

Az itt látható Kossuth-címer C+4 számú mítőzépére írt programmal rajzoltatható a képernyőre.

```
100 GRAPHIC1,1:COLOR4,2
110 CIRCLE,160,-212,220,,120,200
120 CIRCLE,250,0,30,33,200,270
130 CIRCLE,70,0,30,33,90,160
140 CIRCLE,250,60,30,,180,340
150 CIRCLE,70,60,30,,20,180
160 CIRCLE,160,90,90,,90,172
170 CIRCLE1,160,90,90,,188,270
180 CIRCLE,175,195,15,,270,350
190 CIRCLE,145,195,15,,10,90
200 DRAW,160,8TO160,190
210 FORI=30TO190STEP23
220 FORV=159TO70STEP-1:LOCATEV,I
230 IFRDOT(2)THEN V=0:GOTO250
240 DRAW1,V,I
250 NEXT:NEXT
260 FORI=29TO190STEP46:PAINT,159,I:NEXT
270 CIRCLE,196,140,25,35,310,65
280 CIRCLE,170,135,15,25,314,90
290 CIRCLE,225,140,25,25,290,46
300 DRAW1,187,108TO187,85TO172,85
310 DRAWTO172,70TO187,70TO187,55
320 DRAWTO177,55TO177,45TO187,45TO187,30
330 DRAWTO202,30TO202,45TO212,45TO212,55
340 DRAWTO202,55TO202,70TO217,70
350 DRAWTO217,85TO202,85TO202,106
360 PAINT1,161,31:V=10
370 FORI=119TO187STEP4:V=V+4
380 FORK=0TOV:DRAW1,160,ITO160+K,I-K
390 NEXT:NEXT
400 FORX=162TO192STEP4:Y=185:A=X
410 DO:LOCATEA,Y:A=A+1:Y=Y-1
420 LOOPUNTILRDOT(2)=1
430 DO:DRAW,A,Y:A=A+1:Y=Y-1
440 LOCATEA,Y:LOOPWHILERDOT(2)=0
450 NEXT:DRAW0,160,8TO160,184
```

es film értékeit. A fluoreszcens fényállóság próbak során azonban ezek értékükből — színként eltérő mértékben — rohamosan vesztek. A zöld denzitása pl. 350 óras fényhatásra 1,9-ről 0,7-re. Ez idő alatt a Polaroid 669 hasonló értéke 1,5, ami a vizsgálat végéig gyakorlatilag változatlan maradt. A többi szín esetében is hasonló eredmények adódtak, csak a fényhatás időtartama növekedett.

A kutatások alapján megállapítható, hogy a fixálóbevonatot nem igénylő polaroid anyagok szennitometriai tulajdonságai, stabilitása, folyamatos korszerűsítésük eredményeként az utóbbi években jelentősen javult.

A rögtönkész fényképezés területén a Polaroid cég által alkalmazott módszerrel — a színezékeknek fényvegyületekkel történő kombinálásával — sikerült összekötni a kiváló képminőséget a hosszú élettartammal. Ennek eredménye használatának terjedése azokon a területeken is, amelyekben korábban a hagyományos fotón kívül más alkalmazására gondolni sem lehetett.

Pintér László

Rendszerváltó videomagnó

Jóval korábban, mint ahogy a hollywoodi videofilmek forgalomba kerülnek az európai videokazetták, a Panasonic cég alacsony árfekvésű új videomagnójával azok lejátszhatók az amerikai kazettákról az európai rendszerű tv-vevők képernyőjén.

Ez korábban csak igen drága berendezésekkel volt lehetséges, szabvány átalakítással, tekintettel az amerikai és az európai színes tv-rendszerek jelentős különbségére. Az amerikai 525 soros bontással és a másodpercenként megjelent 30 képpel szemben az európai tv-képbontás 625 soros és másodpercenként 25 képet jelenít meg. Ezen túlmenően jelentős különbség van még a színjelek továbbítási módjában az amerikai NTSC és a PAL, illetve a SECAM-rendszereket tekintve.

Az új Panasonic videomagnó a szükséges rendszerváltást jóval egyszerűbben, mikrochippekkel végzi el, az amerikai NTSC-rendszerű műsoros kazetták lejátszásakor a tv-vevőkészüléket PAL-rendszerű jelekkel táplálja.