

# ALAKFELISMERÉS

**4.** Az alakfelismerést biztosító programokba — ha azok valóban a lényegi közös ismérvek alapján ismernek fel ábrákat, figurákat, akkor — a meghatározásra szükséges törvényszerűséget (absztraktiót?) be kell építeni. Az itt közölt 1. program a számítógép képernyőjére rajzolt sikidomot „felismeri” és a meghatározást kiírja.

A 20-50. sorok a klinálású adatokat és a program használatához szükséges minimális információkat tartalmazzák.

A 70-100. sorok biztosítják a HT grafikus lehetőséget adta körülök között a sikidom felnajolását a képernyőre. A pont a számjegyekkel megoldható. A gyors rajzolás feltétele, hogy a számítógép az alap BASIC-et használja. A nyole billentyű rajzolására való felhasználása lehetővé teszi a ferde vonalak rajzolhatóságát. A rajz elrontása vagy újrakezdésének igénye esetében a CLEAR gombot kell lenyomni, mint-hogy az itt közölt programváltozattal nincs lehetőség rádiózásra.

A késznek tartott ábrát a NEWLINE

billentyű lenyomásával adhatjuk át a programnak további feldolgozásra, a sikidom alakjának felismerésére.

A program zárt sikidom meghatározását végez el, de „nem jön zavarba” pont vagy egyenes vonaldarab esetén sem. Nem képes viszont meghatározni az egymást keresztező vagy nem sarokpontron csatlakozó vonalakat tartalmazó sikidomokat.

A 110-310. sorokon tapogatja le a képernyőt a program, és határozza meg a sarokpontok számát.

A 330. sor segítségével a megfelelő sokszög további elemzését végez el.

A sikidomok, pl. négyzetek osztályozásakor a lényegi közös ismérő nem az oldal hossza vagy a sikidom helyzete a papíron (képernyőn), hanem a szomszédos és átellenes oldalak (szögek) egymáshoz viszonyított helyzete, illetve aránya. Ezt a vizsgálatot négyzög esetében a 400-490., illetve az 580-590. sorok végezik el. A vizsgálat kiterjed a megfelelő oldalak párhuzamosságának,

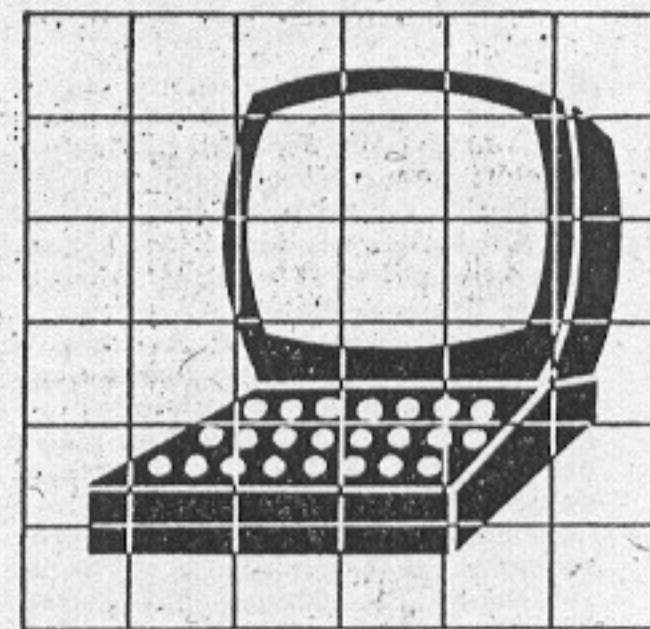
```

10 REM MESTERSEGES INTELLIGENCIA ALAPJAI * EUKLIDES *
20 CLEAR 500:CLS:PRINT"BAL FELSO SZEL KOORDINATAI"
30 INPUT "5<= X <=50 ";X:IFX<50RX>50 THEN GOTO30
40 INPUT "5<= Y <=30 ";Y:IFY<50RY>30 THEN GOTO40
50 CLS:PRINT@768,"RAJZ: 7 0 1 UJ:'CL',VEGE:'NL'":
PRINT@839,"6 2":PRINT@903,"5 4 3";
60 FORI=66TO700STEP64:PRINT@I,STRING$(29,"."):NEXT
70 SET(X,Y):E=PEEK(14400):IFE=1,GOTO110ELSEIFE=2,RUN
80 E=PEEK(14352):IF(Y>4ANDY<33)AND(X>4ANDX<58),
X-X+(E-128)-(E-2)-(E-8)+(E-64)-(E-4):
Y=Y+(E-128)+(E-2)-(E-32)-(E-8)-(E-16)+(E-1)
90 IFY<5.Y=5ELSEIFY>32,Y=32ELSEIFX<5.X=5ELSEIFX>57,X=57
100 GOTO70
110 PRINT@768,CHR$(230):PRINT@896,CHR$(230)
120 FORF=3TO33:FORV=3TO61
130 IFPOINT(V,F),X=V:Y=F:V=70:F=40: GOTO150
140 SET(V,F):FORT=1TO10:NEXT:RESET(V,F)
150 NEXT V,F:Z=0:FORI=1TO4:O(I)=1:NEXT
160 IFV<70,PRINT@768,"URES":
170 J=0:FORF=YTOY+1:FORV=X-1TOX+1:IFPOINT(V,F),J=J+1
180 NEXTV,F:IFJ=1,PRINT@768,"EZ EGY PONT": GOTO520
190 I$="":XK=X:YK=Y: GOTO210
200 IFXK=XANDYK=Y THEN GOTO300
210 J=1:IFJ>T AND POINT(X,Y-1),I$=I$+"1":
T=5:Y=Y-1:F=POINT(X,Y-1): GOTO540
220 J=2:IFJ>T AND POINT(X+1,Y-1),I$=I$+"2":
T=6:X=X+1:Y=Y-1:F=POINT(X+1,Y-1): GOTO540
230 J=3:IFT<>J AND POINT(X+1,Y),I$=I$+"3":
T=7:X=X+1:F=POINT(X+1,Y): GOTO540
240 J=4:IFJ>T AND POINT(X+1,Y+1),I$=I$+"4":
T=8:X=X+1:Y=Y+1:F=POINT(X+1,Y+1): GOTO540
250 J=5:IFJ>T AND POINT(X,Y+1),I$=I$+"5":
T=1:Y=Y+1:F=POINT(X,Y+1): GOTO540
260 J=6:IFJ>T AND POINT(X-1,Y+1),I$=I$+"6":
T=2:X=X-1:Y=Y+1:F=POINT(X-1,Y+1): GOTO540
270 J=7:IFJ>T AND POINT(X-1,Y),I$=I$+"7":
T=3:X=X-1:F=POINT(X-1,Y): GOTO540
280 J=8:IFJ>T AND POINT(X-1,Y-1),I$=I$+"8":
T=4:X=X-1:Y=Y-1:F=POINT(X-1,Y-1): GOTO540
290 PRINT@768,"A SIKIDOM NEM ZART":Z=1
300 S=1:FORI=2TOLEN(I$):
IFMID$(I$,I,1) < > MID$(I$,I-1,1) THEN S=S+1
310 NEXT:I IFZ=1ANDS>1 THEN GOTO360
320 IFZ=0 THEN PRINT@768,"ZART SIKIDOM"
330 ON S GOTO 350,360,370,380,500,510
340 PRINT@768,"A PR. CSAK 6 VONALIG HASZNALHATO":GOTO520
350 PRINT@832,"EZ EGY EGYES VONALARAB": GOTO520
360 PRINT@832,S"DARABBOL ALLO TORT VONAL": GOTO520
370 PRINT@832,"DEREKSZOGU HAROMSZOG": GOTO520
380 PRINT@832,"NEGYSZOG":E$=LEFT$(I$,1):Z=1
390 FORI=2TOLEN(I$):IFE$=MID$(I$,I,1)THEN O(Z)=0(Z)+1:
I(Z)=VAL(E$)ELSEEE$=MID$(I$,I,1):Z=Z+1
400 NEXT I :FOR I=1 TO 4
410 IF(I)/2=INT(I(I)/2),O(I)+1.41*O(I):NEXTLENEXT
420 IF(I)+2=I(2)ANDO(1)=O(3),GOTO580: 'NEGYZET,TEGLALAP
430 IFO(1)=O(3)ANDO(2)=O(4), GOTO590: 'ROMBUSZ,ROMBOID
440 IFO(1)=O(3)ORO(2)=O(4) THEN
PRINT@896,"EGYENLOSZARU TRAPEZ": GOTO520
450 F=0:FORI=1TO2:IFABS(I(I)-I(I+2))=4,F=1:NEXTLENEXT
460 IFF=1,PRINT@896,"DEREKSZOGU TRAPEZ": GOTO520
470 F=0:FORI=1TO2:IFABS(O(I)-O(I+1))<2,F=1:NEXTLENEXT
480 IF F=1,PRINT@896,"DELTOID": GOTO520
490 PRINT@896,"HURNEGYSZOG": GOTO520
500 PRINT@832,"OTSZOG": GOTO520
510 PRINT@832,"HATSZOG"
520 PRINT@960,"UJRA 'T' BETU LENYOMASARA":
530 IF INKEY$<>"T" THEN GOTO 530 ELSE RUN
540 RESET(X,Y):FORI=1TO10:NEXT:SET(X,Y)
550 IFLEN(I$)>200,PRINT@832,"EZ EGY 'FULES' ZART SIKIDOM"
PRINT"KEREM 'FUL' NELKUL": GOTO520
560 IFF=-1,F=0:ONJGOTO210,220,230,240,250,260,270,280
570 GOTO200
580 IFO(1)=O(2)ANDO(3)=O(4),PRINT@896,"NEGYZET": GOTO520
ELSEPRINT@896,"TEGLALAP": GOTO520
590 IFABS(O(1)-O(2))<1,PRINT@896,"ROMBUSZ": GOTO520
ELSEPRINT@896,"ROMBOID": GOTO520

```

1. program

## TECHNIKOLA



### MC-PROGRAMOK

Illetve merőlegességeinek elemzésére, de a hosszúságokat is figyelembe veszi. A 370., illetve 490. sor meghatározása nem geometriai elemzés, hanem a számítógép, illetve a rajzolási módszer korlátaiiból adódó szükségszerű következtetés.

Ez a program látványos, szépen szemlélteti a gépi alakfelismerést, de BASIC-ben elfogadható gyorsasággal működő és ennél összetettebb ilyen jellegű feladatot megoldó program készítése nem célszerű.

A 2. program a hasonló — formailag hasonló — szöveg felismerését végez el. A számítógépek nem ismerik a hozzávetőleg (körülbelül) azonos fogalmát.

Szövegek elemzése — adatbázisból, címjegyzékből való keresés, feleltető program válaszának értelmezése — esetén egy-egy betű elírása miatt a számítógép nem képes a helyes választ megadni.

Egy névsorból különböző okok miatt (rosszul) értett, ékezes vagy ékezet nélküli alak, szóvégi i vagy y, esetleg nem találunk meg egy betűről-betűre azonos nevet, pedig egy betű eltérése esetén a névsort vizsgáló személy intelligenciája (és jóindulata) alapján rákérdezhet a helyes névre.

A program 30-40. során írjuk be az adatbázisba az (itt most csak öt) etalon adatot. A ciklusváltó végértékének növelésével bővíthetjük az adatbázist. (Ne feledkezzünk meg a tömb kijelölé-

séről!). Az 50-es soron írható be az azonosítandó szöveg. A 60-80. soron egy-egy összehasonlítható alapján vagy megtagadja a betű szerint megegyező etalon szöveget, vagy jelzi, hogy azonos szöveget nem talált. A program további része hasonló szöveget keres. A 220. soron megállapított 75%-os betűegyezséssel a két stringet hasonlónak ítéli és kiírja a képernyőre. Természetesen ez az arányszám pl. a szöveg hosszának figyelembevételével módosítható.

Sz. Lukács János

### 10 REM MESTERSEGES INTELLI ALAPJAI KVATI

```

20 CLEAR 200 :CLS
30 FORI=1TO5:PRINTI,:INPUT",RDAT=";J$(I)
40 NEXT I :CLS
50 PRINT:INPUT"KERESEM ":";V$ :FORI=1TO5
60 IF J$(I)=V$ THENPRINT"AZONOS VAN";I=10
70 NEXT :IFI<10,PRINT"AZONOS NIINCSE":PRINT
80 PRINT"HASONLÓ KERESÉS":I=0:FORF=1TO5
90 H1=LEN(J$(F)):H2=LEN(V$):R=0
100 IF H1>H2 THEN E$=J$(F) ELSE E$=V$
110 IF E$=J$(F) THEN M$=V$ ELSE M$=J$(F)
120 Q=0:E=1
130 FORC=ETOLEN(E$):FORR=1TOLEN(M$)
140 IF MID$(E$,C,1)=MID$(M$,R,1),GOSUB180
150 NEXT :E=F+1:NEXT :GOSUB220:NEXT
160 IF J=0 THEN PRINT"HASONLÓ NIINCSE"
170 GOTO50
180 H1=LEN(E$)-C:H2=LEN(M$)-R
190 E$=LEFT$(E$,C-1)+"-"+RIGHT$(E$,H1)
200 M$=LEFT$(M$,R-1)+"-"+RIGHT$(M$,H2)
210 Q=Q+1 : R=0/LEN(E$) : RETURN : TSZLJ
220 IFR>.75ANDV$<>J$(F),PRINTJ$(F):J=J+1
230 RETURN

```

2. program

## C-16 TUNING - 1

A képtartalom a memoriában normál kiírás esetén a 3072 dec.címen kezdődik. Átfordíthatjuk bármely képponton lévő karaktert inverzbe az alábbi program-sorral:

POKE K, PEEK (K)+128+256\*  
(PEEK (K)\*127)

Ahol a K a képernyő memóriaterületre eső cím.

Ha az egész képernyőt inverzbe ki-vánjuk váltani, akkor a K értékét ciklusban változtatni kell.

Egy ilyen Basic-program nagyon lassan végzi el a feladatot. Az itt közölt kis program DATA soraiban tárolt gépi kódú rutin a képernyő átfordítását a másodperc tört része alatt elvégzi.

### 1. REM KEPERNYO ATFORDITAS

```

2 DATA160,4,162,0,189,0,12,73,128,157,0
3 DATA12,202,208,245,238,6,4,238,11,4,136
4 DATA208,234,169,12,141,6,4,141,11,4,96
5 FORI=1024TO1056:READA:POKEI,A:NEXT
10 REM PROBAFUTAS
30 SCNCLR
40 FORY=0TO24:FORX=0TO39
50 POKE 3072+Y*40+X,Y+1
60 NEXT:NEXT
100 FORI=1TO11:SYS1024
110 FORT=1TO600:NEXTT
120 NEXTI
130 GETKEY$:SCNCLR:LIST

```

# ÖNTANULÓ, TANÍTHATÓ PROGRAMOK

A mesterséges intelligenciával foglalkozó munkák külön fejezetet szentelnek a tanuló, önmagukat továbbfejlesztő programoknak. Nézzük mi is néhány ilyen példát öntanuló programra.

Az 1. program a HT adta grafikus lehetőségeket kihasználva a 20-70 sorokon véletlenszerű kialakítással felrajzol egy labirintust. A labirintus bejárata a képernyő jobb alsó sarkán van, és ha átjárható a labrintus, akkor a kijárat a bal felső sarokban van.

A bejáratnál kirajzolt „?” végighalad a labirintuson. A kijárat megkereséséhez azt a közismert módszert alkalmazza, amely szerint a labirintusban minden jobbra (balra) tartva kell a falak mentén végighaladni.

Akádálynak (felnak) ütközve a kérdőjelünk az óramutató járásával megegyező irányba negyed fordulatokat tesz, és

az új haladási irányba megkíséri a továbbhaladást. Ez a falérzékelés és forgás a 80-250 sorokon történik, a képernyő-memória letapogatásával.

A labirintuson ezzel a módszerrel a pont bonyosan keresztülvégödik, de közben egy sor pontot feleslegesen érint.

A program „tanulása” abban áll, hogy minden pontot (azoknak a pontoknak a koordinátáit), amelyeken már „járt”, törli a memoriából, a H tömbből. Ezt a felejtést a 350-370 programsorok biztosítják. Minthogy ezt a programrész a kijárat keresése közben minden újabb pontra való érkezés esetében felkeresi a program, a kijárathoz érve már csak azokra a pontokra „emlékezik”, azon pontok koordinátái vannak a H tömbben, amelyeket a kijárat megtaglásához feltétlenül igénybe kell venni. Igy a kijárathoz vezető egyik lehet-

```

10 CLS:DIMH(500):REM MESTERSEGES INTELL.ALAPJAI ARIADNE
20 FORI=1TO13:PRINT@I*64,CHR$(191):TAB(31)CHR$(191)::NEXT
30 PRINT@666.STRING$(30,191)::PRINT@832.STRING$(30,191):
40 FORI=3TO11STEP2:FORK=1TO31:PRINT@I*64+K,CHR$(191)::NEXTK,I
50 FORI=3TO11STEP2:PRINT@RND(191)+10+64*I,"";X=30:Y=13
60 PRINT@RND(10)+64*I,"";NEXT:PRINT@926,"[";PRINT@1,"["
70 FORI=2TO12STEP2:PRINT@RND(28-I)+I*64,CHR$(191)::NEXT
80 IFY=1THEN260 : REM FELFELE
90 IFPEEK(15360+X+1+64*Y)<>191THEN150
100 IFPEEK(15361+X+64*Y)=191ANDPEEK(15360+X+64*(Y-1))=191,230
110 Y=Y-1:PRINT@64*Y+X,CHR$(128)::GOSUB340:GOTO80
120 PRINT@64*(Y+1)+X,CHR$(128)::GOSUB340:GOTO130
130 IFPEEK(15360+X+64*(Y+1))<>191THEN200: REM JOBBRA
140 IFPEEK(15360+X+1+64*Y)=191THEN80
150 X=X+1:PRINT@X+64*Y?"":W=W+1
160 PRINT@X-1+64*Y,CHR$(128)::GOSUB340:GOTO130
170 IF Y=14 THEN GOTO430: REM LEFELE
180 IFPEEK(15360+X+1+64*Y)<>191THEN240
190 IFPEEK(15360+X+64*(Y+1))=191THEN130
200 Y=Y+1:PRINT@X+64*Y?"": 
210 PRINT@64*(Y-1)+X,CHR$(128)::GOSUB340:GOTO170
220 IFPEEK(15360+X+64*(Y-1))<>191THEN110: REM BALRA
230 IFPEEK(15360+X-1+64*Y)=191THEN170
240 X=X-1:PRINT@X+64*Y?"": 
250 PRINT@X+1+64*Y,CHR$(128)::GOSUB340:GOTO220
260 Q=0:X=30:Y=14:D$(4)="":D$(5)=" "
270 D$(0)="#"":D$(1)="#"":D$(2)="#"":D$(3)="#"": 
280 PRINT@30+13*64,D$(Q)::GOSUB380
290 IFQ=10RQ=30RQ=5THEN320
300 PRINT@960,"UJ LABIRINT / MEGEGYSZER EZ ?";
310 C$=INKEY$:IFC$=""THEN310
320 IFC$="M" THENQ=Q+1:IFQ=6 THEN Q=0
330 IFC$="M" THEN270ELSEIFC$="U" THENRUNE1SE310
340 IFINKEY$="U" THENRUN
350 U=U+1:H(U)=100*X+Y:IFU>3THENRETURN
360 S=H(U):FORI=1TO3:IFS=H(U-I)THENU=U-I-1:RETURN
370 NEXT:RETURN
380 FORI=1TOU:X=INT(H(I)/100):Y=H(I)-100*X
390 IFYE=YANDAES(X-XE)=2THENPRINT@64*Y+(X+XE)/2,D$(Q);
400 IFXE=XANDAES(YE-Y)<>1THENPRINT@32*(YE+Y)+X,D$(Q);
410 IFABS(XE-X)=1ANDABS(YE-Y)=1THENP1=64*Y+XE:P2=64*YE+X;
IFPEEK(P1+15360)=191,PRINT@P2,D$(Q);ELSE
IFPEEK(P2+15360)=191THENPRINT@P1,D$(Q);
420 PRINT@64*Y+X,D$(Q)::XE=X:YE=Y:NEXT:RETURN
430 PRINT@960,"SAJNOS NEM LEHET ATJUTNI !";
440 IFINKEY$=""THEN440 ELSE RUN

```

1. program

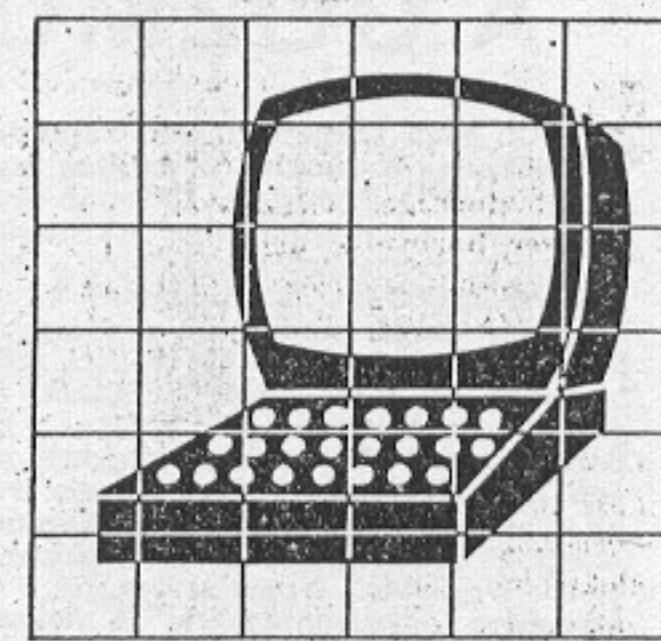
```

10 REM MESTERSEGES INTELL. ÖNTANULÓ BARKOCHÁ
11 CLEAR2000:DIMA(100,2),A$(100)
12 A$(1)="HAZI SZARNYAS":A$(2)="CSIRKE"
13 A$(3)="RIGO":A(1,0)=3:A(1,1)=2:K=4
14 PRINT"GONDOLJ EGY MADARRA, A SEP KITALALJA"
15 PRINT"KEZIDESHEZ NYOMJ LE EGY BILL.-T":N=1
16 IF INKEY$="" THEN GOTO 16 ELSE CLR
17 PRINTA$(N)::INPUT" I/N ";V$
18 IFV$="I",S=1ELSEIFV$="N",S=0ELSE 17
19 L=A(N,S):IFA(L,0)=0,20 ELSE N=L:GOT017
20 PRINTA$(L)::INPUT" I/N ";V$
21 IF V$<>"I" AND V$<>"N": THEN GOTO 21
22 IFV$="I",PRINT"EZ SIKERULT":PRINT:GOT014
23 IF K>97 THEN GOTO 32
24 PRINT"SZABAD A GAZDA":PRINT:U$="":J$=""
25 INPUT" MELYEN ALLATRA GONDOLTAL ";U$
26 IF U$="" THEN GOTO 25
27 INPUT" ROVID JELLEMZESE ";J$:IFJ$="",26
28 A$(K)=J$:A(N,S)=K:A(K,0)=L:A(K,1)=K+1
29 A$(K+1)=U$: K=K+2
30 PRINT" MOST MAR EZT IS KI FOGOM TALALNI "
31 PRINT: GOTO 14
32 PRINT" TOVABB NEM TANITHATO A PROGRAM"
33 GOTO 14

```

2. program

## TECHNIKOLA



### MC-PROGRAMOK

séges — felesleges hurkokat nem tartalmazó — útvonalat az „M” betű leírására a pont újra meg újra képes lesz megrajzolni, hiszen megtanulta”. A program esetleges használata — bemutatása — során gondoljuk végig azt, hogy mi a különbség egy labirintusból haladva megtalálni a kijáratot, és a között a feladat között, amelyben egy papírlapra rajzolt labirintust felülről nézve kell megkeresni annak végpontját. (A természetes intelligencia fejlesztéséhez pedig nézzük meg a görög mitológia iidekapcsolódó részeit.)

A 2. program öntanuló barkochába. Ez a program (ez a játékos) a kijárat kezdetén igen korlátolt. Mindössze két madarak ismer, és ha nem arra gondolt a gép kezelője, akkor már más feladatot ismer, és lehet, hogy a gép kezelője, a tanítóember fárad ki.

A 25-31. sorok ugyanis lehetővé teszik a program által eddig nem ismert újabb és újabb (itt most) madarak nevének és rövid jellemzésének beírását. Ezek a jellemzések és meghatározások az A\$ tömbbe kerülnek. A 28. sor gondoskodik arról, hogy a kérdésre adott igen/nem válasz esetén a program melegen segítségével a tömbváltozókat adatként hagyja a képernyőre. A 29. sor meghatározza, hogy a gép kezelője, a tanítóember fárad ki.

Sz. Lukács János

## C-16 TUNING-2

A nagyfelbontású (grafikus) képernyő képi információját 8000 byte-on tárolja a C-16. A képtárolás a 8192 dec memoriáciámen lévő képelemek inverzbe fordítatottak a POKE K, 255-PEEK (K) programsor segítségével. A csaknem 8 kbyte-nyi terület átfordítása hosszú időt

vesz igénybe. A grafikus képernyő látványos gyors átfordítását végzi el a bemutatott — rövid gépi kódú rutint tartalmazó — program.

A 30-60 sorok közötti kis rajz önmagában is látványosan mutatja be a C-16 grafikus adottságait.

```

1 REM GRAFIKUS KEPERNYÖ ÁTFORDÍTÁS
2 DATA160,32,162,0,189,0,32,73,255,157,0
3 DATA32,202,208,245,238,6,4,238,11,4,136
4 DATA208,234,169,32,141,6,4,141,11,4,96
5 FORI=1024TO1056:READA:POKEI,A:NEXT
10 REM PROBAFUTAS
30 GRAPHIC1,1
40 FORI=180TO1STEP-2
50 CIRCLE1,160,100,100,,IYI
60 NEXT
70 CHAR1,0,0,"SHMU":CHAR1,36,24,"SHMU",1
100 FORI=1TO11:SYS1024
110 FORT=1TO 600:NEXTT
120 NEXTI
130 GETKEYA$:GRAPHIC0,1:LIST

```

## C-16 TUNING-3

A különböző számítógépek képernyőkezelésében a pozícionált szöveg kiíratásában nagy eltérések vannak. Igy a HT PRINT a, a PRIMQ PRINT \$, a SINCLAIR PRINT AT utasítás segítségével biztosítja a képernyő megfelelő helyére való kiíratást.

A Commodore gépeken ez nem meggyil az egyszerűen. A C 64-en:

POKE 211, sor: POKE 214, oszlop: SYS 58732,

a C 16-on:

POKE 205, sor: POKE 202, oszlop: SYS 55369; SYS 55353 utasítássor segítségével lehet az aktuális képernyőpozíciót kijelölni.

A C 16-on alkalmazható a CHAR utasítás, amelynek segítségével egyszerűen kijelölhetjük a kiíratás helyét a képernyón. Az itt közölt kilenc byte hosszú segédprogram szintén segíti a képernyő megfelelő helyére történő kiírást.

```

10 REM POZICIONALT KIIRAS C16-
20 DATA166,216,164,217,24,32,240,255,96
30 FORI=218TO226:READA:POKEI,A:NEXT
40 REM PROBAFUTASOK
50 INPUT"SOR=";S : IFS<0DRS>24THEN50
60 INPUT"OSZLOP=";Ô : IFO<0OR0>39THEN60
70 INFUT"SZOVEG=";X$:PRINTCHR$(147)
80 POKE216,S:POKE217,Ô:SYS218:PRINTX$:
90 GETKEYA$:GOTO40

```

# ÖNTANULÓ JÁTÉKPROGRAMOK

A játékprogramok nagy része, a sakkprogramok csaknem mindegyike a gép lépéseihez megtételéhez megfelelő — nyerésre törekvő — algoritmus ismertetőben dönt. A játékok bonyolultsága egyre bonyolultabb módszerek alkalmazását igénylik. Ezek a programok (programozóik sokoldalú intelligenciája és munkájá alapján) jó, sőt kiváló játékokról képviselhetnek.

Az egyszerűbb játékprogramok jóval egyszerűbb stratégiával beépített igénylis. De talán az egyszerű játéknál nem is érdemes a nyerő stratégiát beépíteni a programba. Jó ellenfél től mi magunk is hamar ellessük az általunk korábban nem ismert játék fortélyait.

Az itt közölt program igen egyszerű, háromszor hármas sakktáblán 3-3 gyalogot tételez fel.

A gyalog egyenesen lép, fordán út Győz, aki előbb átér az ellenfél alapvonala, leüti összes bábját vagy lehetetlenne teszi annak lépését.

A program jelen formájában több funkciót lát el: felerajzolja a képernyőt az állást, ellenőri a játékos lépését és adminisztrálja a nyert és vesztett játékok számát.

Minthogy a játéknak nagyon hamar második, harmadik lépésben vége lehet, a gép — a program — nagyon hamar érzékel azt, hogy egy lépést érdemes volt meglépni vagy sem. S mint Murphy törvényeiből ismerhetjük: mindenki képes a helyes döntés meghozatalára, ha már az összes helytelent kipróbalta, a programmal a helytelen, vesztéshez vezető lépést el kell fejezni! A tanulás folyamata ezzel a programmal: a tapasztalat alapján a rossz lépésvariáció elfejtése.

Nyolc-tiz játék alatt, ha a gép előtt ülő játékos elég jól játszik, a program „elfelejt” az összes rossz lépéslehetőséget. Nyert játékból ez a program nem tanul.

A tanulási folyamat után már a gép és a játékos közel egyenlő esélyel mérkőzik egymással.

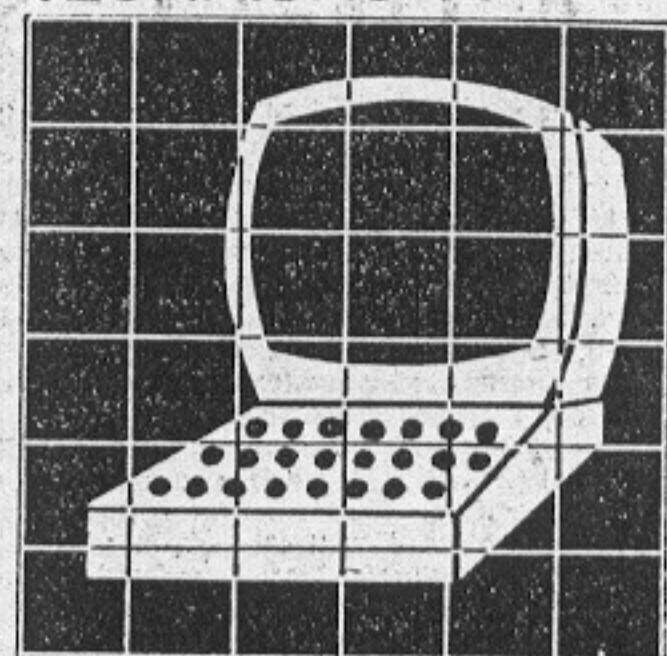
Mint látható, egy ilyen egyszerű játék helyes lépéseinek megtanulása is aránylag hosszú program frását igényli, pedig ebben a játékból összesen 51 lépésvariáció létezik, s a tábla szimmetrikus.

```

10 REM MESTERSEGES INTELLIGENCIA COMPUTER LUDENS
20 CLEAR2000:DIML(50).J(50.5),A$(50);CLS:JA=-1
30 FORI=1TO44STEP2:READA$(I):PRINT@320+I,"*"
40 FORJ=0TO6STEP3:FÖRL=3TO1STEP-1
50 A$(I+1)=A$(I+1)+MID$(A$(I),L+J,1):NEXT L,J
60 READL(I):L(I+1)=L(I):FORL=1TOL(I):READJ(I,L)
70 S=J(I,L):J(I+1,L)=(S-50)*(S<40)+(S-116)*(S>40)
80:NEXTL:NEXTI
90 A$="GGG...JJJ":N=0:JA=JA+1:KI=-174:CLS
100 PRINT" 123 JATEKOS: EP =":PRINT"A GGG"
110 PRINT"B...":PRINT"CJJJ":PRINT@24.(JA-NY)":"NY;
120 REM VAN E A JATEKOSEN LEHETOSEGE LEPESERE?
130 N$="GEP":E$="G":F$="J":E=7:F=9
140 GOSUB 480: IF J=1 THEN GOTO 340
150 GOSUB500:IFJ=0,GOTO340
160 Z=-3:L=0:FORI=1TOJ:V=H(I)
170 IFV=7ORV=4THEN E=2:GOSUB520;IFL>0,I=5
180 IFV=9ORV=6THEN E=4:GOSUB520;IFL>0,I=5
190 IFV=8ORV=5THEN E=2:GOSUB520:E=4:GOSUB520
200 NEXT:IFL=0THEN340
210 KI=KI+250:JATEKOS LEPHET?-LEPJEN
220 PRINT@830,"-",INPUT"HONNAN PL C1 ":";L$
230 GOSUB540:IFJ=1THENGOTO220 ELSEH1=H
240 IFMID$(A$.H,1)<>"J"THENGOTO220
250 PRINT@830,"-",INPUT"HOVA PL B1:";L$:GOSUB540
260 IFJ=1THENGOTO250
270 H2=H : E = HI-H2:V$=MID$(A$,H,1)
280 IFE=3ANDV$=". THEN GOTO330
290 IF(H1=4ORH1=7)ANDE=2ANDV$="G"THEN GOTO330
300 IF(H1=6ORH1=9)ANDE=4ANDV$="G"THEN GOTO330
310 IF(H1=5ORH1=8)AND(E=2ORE=4)ANDV$="G",GOTO330
320 GOTO220
330 GOSUB580:GOTO370
340 PRINT@830,"-",PRINT"NYERT A ":";N$;" TOVABB "T"
350 IFN$="GEP":NY=NY+1
360 IFINKEY$>"T"THEN360 ELSE90
370 E$="J":F$="G":N$="JATEKOS":KI=KI+6:N=N+1:'GEP
380 KI=5:V1=24:IFN>3,KI=25:V1=44
390 IF N<2 THEN KI=1:V1=4
400 FORI=KITOV1:IFA$=A$(I),Q=I:I=100
410 NEXT:IFI<100THEN440 ELSE H=RND(L(Q))
420 HI=INT(J(Q,H)/10):H2=J(Q,H)-10^H1:U=Q:W=H
430 GOSUB580:GOTO120
440 REM TANULJ AZAZ FELEJTSD EL AZ UTOLSO ROSSZAT
450 IFL(Q)=1THEN340
460 FORI=WTO3:J(Q,I)=J(Q,I+1):NEXT:L(Q)=L(Q)-1
470 GOTO 340
480 J=0:FORI=ETOIF:IFMID$(A$,I,1)=E$,J=1
490 NEXT:RETURN
500 J=0:FORI=1TO9:IFMID$(A$,I,1)=F$,J=J+1:H(J)=I
510 NEXT:RETURN
520 IFMID$(A$,V+Z,1)=". ,L=L+1:RETURN
530 IFMID$(A$,V-E,1)=F$ ,L=L+1:RETURN ELSERETURN
540 J=0:K1=ASC(LEFT$(L$,1))-65
550 IF K1<0 OR K1>2 THEN J=1:RETURN
560 K2=VAL(RIGHT$(L$,1)):IFK2<1ORK2>3,J=1:RETURN
570 H-K1*3+K2:RETURN
580 A1=H1:A2=H2:IFH1>H2,A1=H2:A2=H1
590 A$=LEFT$(A$,H1-1)+" "+MID$(A$,H1+1)
600 A$=LEFT$(A$,H2-1)+F$ +MID$(A$,H2+1)
610 PRINT@KI,LEFT$(A$,3):PRINT@KI+64,MID$(A$,4,3);
620 PRINT@KI+128,RIGHT$(A$,3):RETURN
630 DATA GGGJ...JJ,3,24,25,36,G.GJJ..J.,3,15,35,36
640 DATA G.GJ...J,1,36 ..G.GJJ..J,4,15,35,36,47
650 DATA G.GG.J.J.,2,47,48,GGG.J.J.J,4,14,15,35,36
660 DATA GG.JJG..J,2,15,24 ..GG.J.J..J,3,24,25,26
670 DATA .GGJG..J,3,24,58,59 ..GG.J..J,2,35,36
680 DATA .GG.J.J.,2,35,36 ..GG.GJJ..J,3,26,57,58
690 DATA ..GGJ...,3,35,36,47 ..GGGJ...,2,47,58
700 DATA G..GJ...,2,15,47 ..G..GGJ...,2,47,58
710 DATA ..GGJ...,3,35,36,47 ..G..JJ...,1,26
720 DATA G..JJJ...,1,15 ..G.JJ...,1,24
730 DATA .G.JG...,2,24,58 ..G.JG...,2,35,69:

```

## TECHNIKOLA



### MC-PROGRAMOK

Az itt közölt program elemzése után akár fordított logikával is felépíthetjük azt a programot, amelynek során a játékos jó lépéseiit tanulja el a gép.

Dr. Sz. Lukács János

## C-16 Tuning-5

Programozás közben sokszor jó lenne „szabálytalan” megoldáshoz folyamodni. Ilyen lehet egy ciklus vagy egy szubrutin elhagyása.

A szubrutinból RETURN utasítással tér vissza a program a meghívás által meghatározott helyre. Ha a szubrutinból

a RETURN igénybevétele nélkül lépünk ki, akkor ez, mivel a visszatérési cémet a veremben hagyuk, nehezen felfedezhető hibához vezet. Amennyiben mégis ki akarunk lépni a szubrutinból, akkor a vermet „rendbe kell tenni”.

Ilyen megoldásra ad példát a mellékelt program.

```

10 REM C16 SUBRUTINBOL GOTO-VHL KILEP
20 DATA169,141,133,2,32,113,136,240,8
30 DATA1162,12,44,162,17,76,131,134,32
40 DATA105,167,160,5,76,114,167
50 FORI=1015TO1039:READA:POKEI,A:NEXT
60 REM PROBEFUTASOK
70 FOR I=1TO100
80 PRINT I:GOSUB130
90 PRINT"SZABALYOSAN VISSZA ERT"
100 NEXT I
110 GOTO70
120 PRINT"VEREM RENDEZVE":GOTO100
130 PRINT:PRINT"SZUBRUTINBOL"
140 PRINT"RETURN=BARMI GOTO=R"
150 GETKEYR$ 
160 IFR$="R"THENSYS1015 : GOTO120
170 RETURN

```

## C-16 Tuning-6

Függvényvizsgálatokhoz, ábrázolásokhoz a függvénykapcsolatot, a program javításával mindenig újra meg újra be kell írni a programba.

Az itt közölt program INPUT-ként közt y=f(x) függvény értelmezésére, értékészletének meghatározására, ábrázolására alkalmas.

A program első két sorának megváltoztatása esetén a C értékét is módosí-

tani kell! Futás közben a program átirja önmagát, és a 20 sorszámú soron Y= után tokenizálva letárolja a függvényt. Az újabb függvények beírásával módosul a sor tartalma. A mindenkor függvény értékét e sort szubrutinként meghívja számíthatjuk ki. Erre ad példát a program a 120—160 sorok közötti része.

Ismételt futtatáshoz a 170. sorra GOTO 120 irandó.

```

10 GOTO80
20 Y:::::IDE:80:DB:KETTOSPONT::::::::::
30 INPUT"Y=";F$ 
40 GRAPHICCLR : SYS 1015 : F=0
50 DO:C=4111+F:R=PEEK(F+512)
60 POKEC,R:F=F+1:LOOPUNTILR=0:POKEC,58
70 POKEC+1,142:POKEC+2,58:RETURN
80 DATA165,59,72,165,60,72,162,0
90 DATA134,59,160,2,132,60,76,92,137
100 FORI=1015TO1031:READA:POKEI,A:NEXT
110 REM PROBEFUTAS
120 PRINT : DOSUB 30
130 INPUT"XMIN=";A
140 INPUT"XMAX=";B
150 PRINT"Y=";F$ : PRINT"X",Y
160 FORX=ATOB:GOSUB 20:PRINTX,Y:NEXT

```

# CNC-ESZTERGA SZIMULÁCIÓJA

TECHNIKOLA

**1** Egyre több — szakirányú képzést nyújtó — középiskola tananyagában található meg az NC- és CNC-szerszámgépekkel kapcsolatos ismeretanyag.

A konkrét szerszámgép üzemeltetése, a megmunkálást lehetővé tevő program megírása és kipróbálása csak a szerszámgép mellett lehetséges. Az oktatást segítő, iskolákban található CNC megmunkáló gépek száma sajnos kicsi. Az EMCO COMPACT 5 CNC oktatási készterga mellett vannak hazai kialakításuk is, mint például a miskolci Zalka Máté Szakközépiskoláé.

Az itt bemutatott C64 típusú számító-

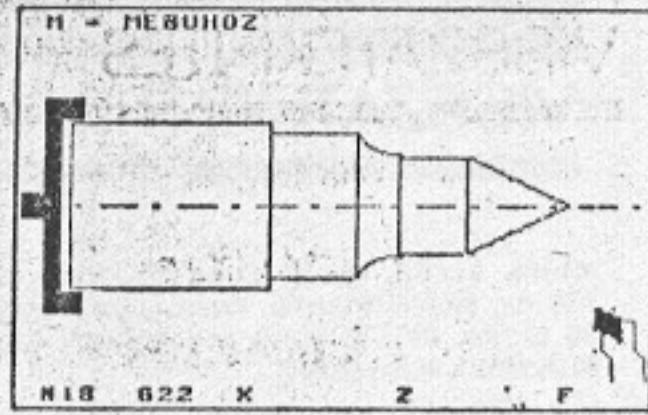
gépre írt program (100—114 sor) feladata, hogy az EMCO 5 CNC szerszámgép programozását és üzemét szimulálja, és ha az olvasók fantáziáját meghozza, akkor a program továbbfejlesztéséhez adjon ötletet, indittatást. A program menüje (284—312 sor) lehetővé teszi a megfelelő ágon való futtatást.

A DEMO ág (592—636 sor) ismerteti a program lehetőségeit és korlátait. Ezek a korlátok a rajzi megjelenítés szempontjai miatt kisebb kiindulási átmérőt ésemegmunkálható munkadarabhoz engednek meg, mint a valóság.

Az itt közölt változat értelmezni és végrehajtja a 600—620 sorokon bemutatott haladási kódokat.

A programlépések (mondatok) számának korlátozása csak képernyőkezelési megfontolásokból adódik, és igen egyszerűen programozható, akár a fejlesztett változatú készterga lépésszáma. A data sorokban (638—660) tárolt rövid megmunkálóprogram a szokásos mondat-szerkezet felépítésének megfelelő sorrendben tartalmazza az ábrán bemutatott munkadarab megmunkálásának programját.

A tanprogramok frászhoz igen jó segítséget nyújthat az OPI Szakképzési Igazgatósága gondozásában megjelent



A megmunkálás vázlata

```

100 REM      EMCO COMPACT 5 CNC MÓDELLEZESE
102
104 REM      C=64 + SIMON'S BASIC
106
108 GOTO284
110 REM MEGALLITAS CSAK UJ MONDATON??
112 GOSUB310:GOSUB380:GOSUB392:GOTO284
114 GOSUB314:GOSUB270

```

Program

```

284 PRINT"J":REM MENU
285 CENTRE"TOVABBII OHAJAD":PRINT
286 PRINT" 1 UJ FELADAT":PRINT
288 PRINT" 2 FORGOROSOLAS":PRINT
290 PRINT" 3 PROGRAM LISTA":PRINT
292 PRINT" D DEMO":PRINT
294 PRINT" 0 PR. VEGET":PRINT
296 INPUT" VALASZTARSA: " ;N$:IFN$="":
300 IFN$="0"THEN END
302 IFN$="1"THEN RUN 112
304 IFN$="2"THEN GOTO114
306 IFN$="3"THEN GOTO510
308 IFN$="D"THEN GOTO592 :ELSE GOTO 284
310 DIMG$(20),X$(20),Z$(20),F$(20)
312 DIMM$(20): RETURN

```

Menü

```

592 REM      DEMO
594 PRINT"J":CENTRE"0 E M 0":PRINT
596 CENTRE"EMCO COMPACT 5 CNC":PRINT
598 CLR:GOSUB310:PRINT:D=40:L=100
600 PRINT"ALKALMAZHATO HALADASI KODOK:":N"
602 PRINT" G 00 GYORSJARAT"
604 PRINT" G 01 LINEARIS INTERPOLACIO"
606 PRINT" G 02 (03) KOR INTERPOLACIO"
608 PRINT" T01 KES MIATT X POZITIV"
610 PRINT" G 20 PROGRAMOZOTT MEGALLAS"
612 PRINT" G 21 URES MONDATHELY"
614 PRINT" G 22 PROGRAMVEG"
616 PRINT" G 84 HOSSZESTERGALO CIKLUS"
618 PRINT
620 PRINT" G 91/G 95 ALAPERTELMEZES"
622 PRINT"MAX.      ATMERO : 50 MM "
624 PRINT"MAX.      HOSSZ :130 MM "
626 PRINT"MAX. MONDATSZAM : 20 "
628 RESTORE:READZ:  FORN=0TOZ
630 READG$(N),X$(N),Z$(N),F$(N),M$(N)
632 NEXT
634 PRINT"MBILL NYOMASRA TOVABB"
636 GETB$:IFB$=""THEN 636:ELSE GOTO 510
638 DATA18,,00,-500,0,,T 01
640 DATA00,0,-400,,84,-200,-6000,100,
642 DATA84,-400,-4000,100,
644 DATA84,-600,-3800,100,
646 DATA84,-800,-3400,100,,00,-800,,,
648 DATA01,0,-3400,100,,02,1000,,,
650 DATA00,0,4400,,84,-1200,-1200,100,
652 DATA84,-1400,-800,100,
654 DATA84,-1600,-400,100,
656 DATA00,0,-100,,00,-2190,0,,,
658 DATA01,1400,-2400,50,,00,1100,,,
660 DATA00,0,2900,,22,,,

```

DEMO-ág

NC-technika molulsorozat 2/a és 2/b. füze.

A különböző ágakon futtató program minden esetben a menühöz tér vissza.

Ha a programunk további lehetőségeit kívánjuk megismerni, úgy választhatjuk a „Lista” ágat, vagy az „Új feladat” beírását. Tulajdonképpen a program ezen ága teszi lehetővé azt, hogy mire a felhasználó a szerszámgéphez kerül, már tudja (legalábbis sejtse) mit kell ott tenni.

A számítógép mellett is el kell végezni a szerszámép mellett írt megmunkálóprogram készítése során szükséges tennivalókat: a munkadarab megfelelő léptékű megrajzolását, a technológiai adatok kiszámítását.

Ezek ismeretében itt most a számítógépprogram bekéri a kiindulási méreteket és a megmunkálóprogramot (380—508 sorok). A megmunkálóprogram szavainak mondatainak beírása során a számítógép elemzi és értelmezi a kódokat. A szükséges további adatokat a gép (ellenőrzött input rutinnal) beolvassa.

A megmunkálóprogram beírása az F1 lenyomásáig vagy a G 22 kód beírásáig folytatódik.

Az öröklődő (vagy a következő mondaton ismétlődő) kódok, adatok beírására is mód van az adatbekérés során csak a RETURN lenyomásával. A bekért meg-

## JC-PROGRAMOK

munkálási adatokból a szimulációs program a továbbiakban az előtolásértékeket nem használja fel.

A megmunkálóprogram beírása után a további választási lehetőséget a menü biztosítja.

Lehetőség van egyrészt a program javítására, kiegészítésére másrészt a forgácsolás rajzok formájában — folyamatában — való „elvégzésére”, szemléltetésre. Ezt következő számban ismertetjük.

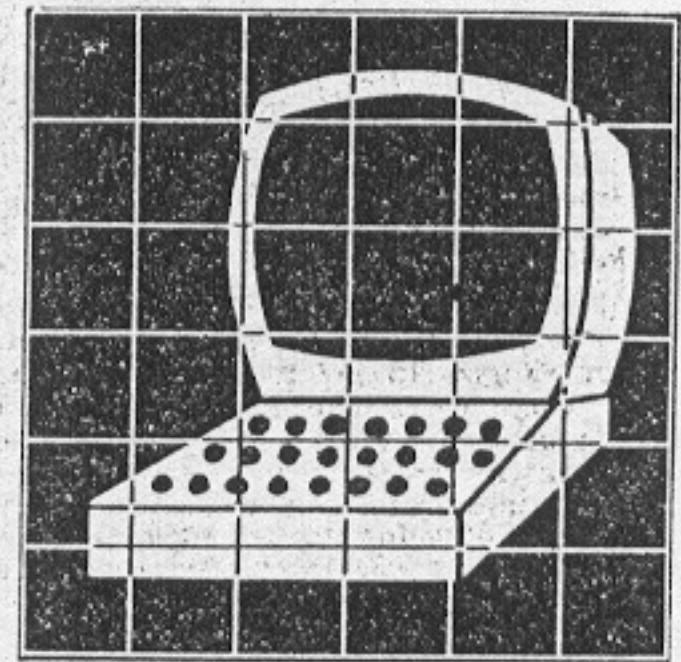
Dr. Sz. Lukács János

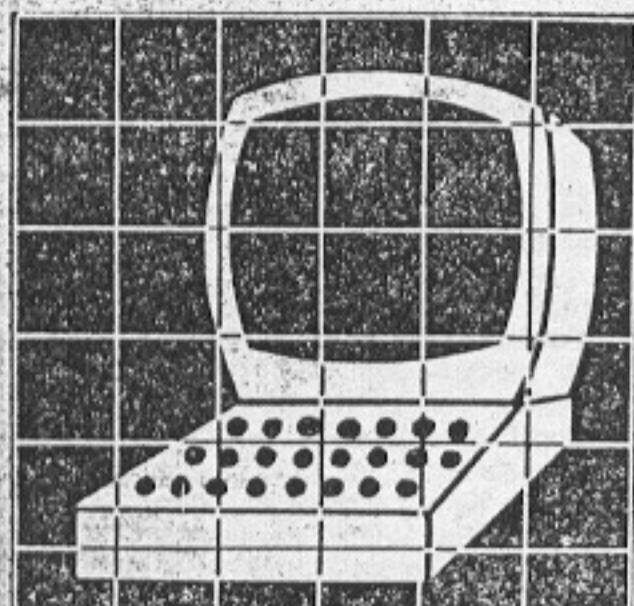
```

380 REM NYERS MERET BEIRAS
382 PRINT"JUNKI INDULASI MERETEK:":N$"
384 INPUT"KIIND. ATMERO=";D
386 IF D<50RD050:THEN GOTO384:ELSE:PRINT
388 INPUT"KIIND. HOSSZ=";L
390 IF L<200RL>130:THEN GOTO388:ELSE:RETURN
392 REM PROGRAM BEIRAS
394 M0=0:N0=20:Z=0
396 PRINT"J N   G   X   Z";
398 PRINT"          F   MEGJ.N"
400 FORH=MOTONG:T$="00"+MID$(STR$(H),2)
402 H$=RIGHT$(T$,2):PRINTAT(0,N+2)H$:
404 G$="":S=8:H=1:U=0
406 PRINTAT(S,N+2)G$+"?":C=133
408 GETS$:IFS$="":THEN GOTO408
410 IFS$=CHR$(13):ANDNZ=N-1:GOTO408
412 IFS$=CHR$(13):THEN GOSUB496:GOTO408
414 IFNZGANDG$="":THEN G$=G$(N-1)
416 IFLEN(G$)<2:THEN GOTO406
418 PRINTAT(S,N+2)G$+" ";G$(N)=G$
420 IFG$="22":THENZ=N:GOTO408
422 IFG$="20":ORB$="21":THEN GOTO476
424 G$="":S=11:H=4:U=0
426 PRINTAT(S,N+2)G$+"?":REM X
428 GETS$:IFS$="":THEN GOTO428
430 IFS$<>CHR$(13):THEN GOSUB496:GOTO428
432 IF U <10RABS(VAL(G$))>6500:THEN 426
434 IFRIGHT$(G$,1)="-":THEN GOSUB492
436 GOSUB506:X$(N)=G$
438 IFVAL(G$)<0ANDG$(N)="00":THEN 476
440 IFG$(N)="02":OR G$(N)="03":THEN 476
442 IFG$(N)="84":ANDVAL(X$(N))=0:THEN 424
444 G$="":S=18:H=4:U=0
446 PRINT AT(S,N+2)G$+"?":REM Z
448 GETS$:IFS$="":THEN GOTO448
450 IFS$<>CHR$(13):THEN GOSUB496:GOTO448
452 IF U<10RABS(VAL(G$))>9999:THEN 446
454 IFRIGHT$(G$,1)="-":THEN GOSUB492
456 GOSUB506:Z$(N)=G$
458 IFG$(N)="84":ANDVAL(Z$(N))=0:THEN 444
460 IFG$(N)="00":THEN GOTO476
462 G$="":S=25:H=2:U=0
464 PRINT AT(S,N+2)G$+"?":REM F
466 GETS$:IFS$="":THEN GOTO466
468 IFS$<>CHR$(13):THEN GOSUB496:GOTO466
470 IFNDGANDG$="":THEN G$=F$(N-1)
472 IFVAL(G$)<0:ORVAL(G$)>499:THEN 464
474 GOSUB506:F$(N)=G$
476 G$="":S=32:U=0:H=5
478 PRINT AT(S,N+2)G$+"?":REM M
480 GETS$:IFS$="":THEN GOTO480
482 IFU>5ANDS$<>CHR$(13):THEN GOTO480
484 IFS$<>CHR$(13):THEN GOSUB496:GOTO478
486 GOSUB506:M$(N)=G$:PRINT:NEXT
488 REM PROGRAMOZAS VEGE
490 FORM=1TO1:NEXT:RETURN
492 G$="":LEFT$(G$,U-1):RETURN
494 G$=LEFT$(G$,U-1):GOTO502
496 IFU>0AND(S$=CHR$(20)ORS$="||")THEN494
498 IF U>H OR S$<"-":ORS$>"9":THEN:RETURN
500 G$=G$+S$:
502 U=LEN(G$)
504 PRINTAT(S,N+2)G$+"?":RETURN
506 PRINTAT(S,N+2)RIGHT$(G$,"+G$,5)
508 PRINT":":RETURN

```

Megmunkálóprogram





## MC-PROGRAMOK

```

314 REM KIINDULAS FELRAJZOLASA
315 HIRES0,1:DESIGN 0,18*64+49152
318 @B... .
320 @BBBB... .
322 @BBBBBBB... .
324 @BBBBBBBBB... .
326 @BBBBBBBBBBB... .
328 @BBBBBBBB... B
330 @BBBBBBBB... B
332 @.BBBBBB... B
334 @.B... BBB... B
336 @.B... .B... B
338 @.B... .B... B
340 @.B... .B... B
342 @.B... .B... B
344 @.B... .B... B
346 @.B... .B... B
348 @.B... .B... B
350 @.B... .B... B
352 @.B... .B... B
354 @.B... .B... B
356 @.B... .B... B
358 @.B... .B... B
360 BLOCK0,94,8,106,1
362 BLOCK9,88-D,16,112+D,1
364 BLOCK16,88-D,25,98-D,1
366 BLOCK16,102+D,25,112+D,1:C=100
368 FORX=0TO10:LINEX*32,C,X*32+15,C,1
370 LINEK*32+22,100,X*32+24,100,1:NEXT
372 RED 20,100-D,2*L,2*D,1
374 MOB SET1,18,0,0,0:X0=55+2*L
376 Y0=160+D:MMOB1,X0,Y0,X0,Y0,0,0
378 XK=X0:YK=Y0:PLOTX0-24,Y0-50,1:RETURN

```

## 1. program

```

510 REM PROGRAM KIIRASA/NYOMTATASA
512 PRINT"J N G X Z ";;
514 PRINT" F MEGJ.N" :FORN=0TOZ
516 N$=RIGHT$("00"+MID$(STR$(N),2),2)
518 PRINTN$" ";
520 G$=G$(N):S=3:GOSUB506
522 G$=X$(N):S=11:GOSUB506
524 G$=Z$(N):S=18:GOSUB506
526 G$=F$(N):S=25:GOSUB506
528 G$=M$(N):S=32:GOSUB506:PRINT:NEXT
530 IFW$="M":ORW$="B":THEN RETURN:/
532 PRINTAT(0,24)"J=JAVITAS ";
534 PRINT"P=NYOMTATAS M=MENU";
536 GETV$:IFV$=CHR$(133):THEN GOTO284
538 IFV$="M":THEN GOTO284
540 IFV$="P":THEN GOSUB506:HRDCPY:GOTO284
542 IFV$>"J":THEN GOTO582
544 PRINTAT(0,24)"M=MODOSITAS ";
546 PRINT" T=SOR TORLES B=BESZURAS ";
548 GETW$:IFW$="M":ORW$="T":THEN GOT0572
550 IFW$=CHR$(133):THEN GOTO284
552 IFW$>"B": THEN GOTO544
554 GOSUB506
556 INPUT"HANYADIK UTAN":J:IFJ>ZTHEN554
558 J=J+1:FORC=ZTOJ STEP-1
560 G$(C+1)=G$(C):X$(C+1)=X$(C)
562 Z$(C+1)=Z$(C):F$(C+1)=F$(C)
564 NEXT:Z=Z+1
566 G$(J)="" X$(J)="" Z$(J)="" F$(J)=""
568 GOSUB512:ZZ=Z:MO=J:HO=J
570 GOSUB400:Z=ZZ: GOTO284
572 GOSUB506:INPUT"MELYIKET":J
574 IFJ<0 OR J>Z:THEN GOTO572
576 IFW$="M": THEN GOTO566
578 FORC=JTOZ
580 G$(C)=G$(C+1):X$(C)=X$(C+1)
582 Z$(C)=Z$(C+1):F$(C)=F$(C+1)
584 NEXT:Z=Z-1: GOTO510
586 PRINTAT(0,24)" ";
588 PRINT" ";
590 PRINTAT(0,24)""; RETURN

```

## 2. program

## CNC-ESZTERGA SZIMULÁCIÓJA

**2** A forgácsoló program, ha azt a szerszámgép (itt most a szimulációs program) elfogadta, végrehajtásra kerülhet. Az elmozdulások irányá és nagysága a kész munkadarab méretét meghatározzák. Hibás értékek nem megfelelő alakú és méretű munkadarabot eredményeznek, adott esetben a szerszám vagy a szerszámgép károsodásához vezethetnek. A hibák zöme csak forgácsolás közben derül ki.

Lássuk a forgácsolást. A rajzi megjelenítéshez definíálni kell (itt most TO 1) az esztergakés ábráját (314–358 sorok) és fel kell rajzolni a kiindulási állapotot (360–378) sorok).

A program léptékhez ábrát készít a munkadarabról és a szerszám csúcását a nyers méret kontúrjától X, Y irányban 5-5 mm-re helyezi.

A „forgácsolást”, a késmezgás rajzi megjelenítését a 116–282 sorok végezik.

A fulajdonképpen forgácsolás végrehajtását vezérlő (116–216 sor) programrész minden egyes művelet végrehajtása alatt kirja az aktuális mondatot a képernyón.

A „kés leválasztja a forgácsot” és az új kontúroknak megfelelő munkadarab rajza marad a képernyón.

A program nem hajt végre olyan mozgást, ami „géptöréshez” vezetne. (Igaz, a túl nagy fogásvétel okozta túlterhelést nem érzékel.) A mozgási határon való túlhaladást eredményez

utasítások esetében — még ha nem is a szerszámgépen megtalálható formában — szimulációs programunk hibajelzést ad.

A listában a szükséges javítás: törlés, beszűrás, módosítás elvégezhető (510–590 sorok). A módosított programmal újra elvégezhető a „forgácsolás”.

A hiba nélküli elvégzett „megmunkálás” után a kész megmunkálóprogramot a listázási ág után az 540. sor segítségével akár ki is nyomtathatjuk.

Dr. Sz. Lukács János

```

116 REM FÖRGACSLAS
118 IFG$(Z)="22":THEN GOTO 124
120 TEXT24,16,"ZAROKOD HIANYZIK",1,0,8
122 GOSUB 280 :GOTO 210
124 FORN=0 TO Z:TEXT8,188,A$,0,0,8
126 A$="H"+RIGHT$("0"+MID$(STR$(N),2),2)
128 A$=A$+": G"+G$(N)
130 A$=A$+" X"+LEFT$(X$(N))+": ,5"
132 A$=A$+" Z"+LEFT$(Z$(N))+": ,5"
134 A$=A$+" F"+LEFT$(F$(N))+": ,3"
136 A$=A$+" " +LEFT$(M$(N))+": ,7"
138 TEXT8,188,A$,1,0,8
140 GETS$:IFS$="@":THEN GOTO 210
142 F=VAL(X$(N))/50:V=VAL(Z$(N))/50
144 IFG$(N)="22":THEN GOTO 208
146 IFG$(N)="21":THEN GOTO 206
148 IFG$(N)>"20":THEN GOTO 154
150 TEXT8,8,"STOP =@",0,0,8
152 GOSUB 270 :GOTO 206
154 XK=X0:YK=Y0:XV=KK+V:YY=YK+F
156 IFG$(N)>"00":THEN GOTO 166
158 W=0
160 IFV=0:THEN GOSUB 220
162 IFF=0:THEN GOSUB 228
164 X0=XV:Y0=YY:GOTO 206
166 IFG$(N)>"01":THEN GOTO 178
168 W=1:IFF=0:ORV=0:THEN GOTO 160
170 SV=0
172 FORX=X0TOXVSTEP(XV-KK)/ABS(XK-XV)
174 YQ=Y0+(X-X0)*F/V:GOSUB 244:NEXT
176 X0=XV:Y0=YY:GOTO 196
178 IFG$(N)>"02":ANDG$(N)>"03":THEN 198
180 C=251:SV=0:W=1:FORA=0TO1.57STEP0.03
182 J=Y0+F:IFG$(N)="03":THEN GOTO190
184 IFF>0:THEN X=X0-F*SIN(A):YQ=J-F*COS(A)
186 IFF<0:THEN X=X0+F-F*COS(A):YQ=J*SIN(A)
188 GOTO194
190 IFF>0:THEN X=X0-F+F*COS(A):YQ=J*SIN(A)
192 IFF<0:THEN X=X0+F+F*SIN(A):YQ=J-F*COS(A)
194 YY=YQ:GOSUB244:NEXT:X0=X0-F:Y0=YQ
196 IFSV=0:THEN LINEX-25,Y0-51,X-25,C-YQ,W
198 W=0:IFG$(N)>"84":THEN GOTO 206
200 X2=X0:Y2=Y0:GOSUB220:W=1:GOSUB 228
202 W=0:Y0=YY:YY=Y2:X0=XV:GOSUB 220
204 X0=XV:XV=X2:GOSUB228:X0=X2:Y0=Y2
206 NEXT
208 REM FORG VEGE
210 TEXT8,8,"STOP = @",0,0,8
212 TEXT8,8,"M = MENUHOZ",1,0,8
214 GETI$:IF I$>"M":THEN GOTO 214
216 MOB OFF1:HRM:GOTO 284
218 REM FORG RAJZA
220 S=1:IFYV<0:THEN S=-1
222 FORY=Y0TOYVSTEP:MMOB1,X0,Y,X0,Y,0,2
224 IFY<1500:RY>243:THEN GOSUB280:GOTO 210
226 NEXT:RETURN
228 S=1:IFXV<X0:THEN S=-1
230 FORX=X0TOXVSTEP:MMOB1,X,YY,X,YY,0,2
232 IFW=1:AND(X<2*L+46):THEN YQ=Y:GOSUB250
234 IFX<=520:RY>330:THEN GOSUB280:GOTO 210
236 NEXT
238 FORQ=99-DT099:A=TEST(X-25,Q)
240 IFA=1:THEN K=Q:Q=500
242 NEXT:LINEX-25,K,X-25,200-K,W:RETURN
244 MMOB1,X,Y0,X,Y0,0,150:O-YQ-50
246 P1=TEST(X-26,0):P2=TEST(X-51,0)
248 IFF1=1:ANDP2=1:ANDSV<1:THEN GOSUB266
250 IF W=0:THEN GOTO262
252 LINEX-25,100+D,X-25,YQ-50,0
254 LINEX-25,-97-D,X-25,250-YQ,0
256 IF X= X0:THEN GOTO262
258 LINEX-26,Y0-51,X-26,101,0
260 LINEX-26,99,X-26,251-YQ,0
262 PLOTX-25,YQ-50,W:PLOTX-25,250-YQ,W
264 RETURN
266 W=0:LINEX-25,Y0-51,X-25,251-YQ,1
268 SV=1:RETURN
270 REM INDITASRA VAR
272 TEXT8,8,"START = *",1,0,8:C=133
274 GETI$:IFI$<">*"ANDI$<>CHR$(C):THEN274
276 TEXT8,8,"START = *",0,0,8:X0=KK
278 TEXT8,8,"STOP = @",1,0,8:Y0=YK:RETURN
280 FORY=1TO1:NEXT:FORX=1TO1:NEXT
282 TEXT200,16,"BAJ VAN!",1,1,8:RETURN.

```

## 3. program

# ZENE-BONA SZÁMÍTÓGÉPEKKEL

TECHNISKOLA

EGY'SOROS "ZOMGORÁ"

```
1 A$="00770798810834854864881897911":VOL  
6:GETKEYH:SOUND1,VAL(MID$(A$,H*3,3)),12:  
GOT01
```

## 1. program

```

10. REM : HANGASZ V 1/2
20 F=123.5 :D=EXP(LOG(2)/12) :VOL6
30 FORJ=1TO3 :RESTORE
40 ::FORL=1TO12 :READH$ :PRINTH$,
50 : F=F*D:R=INT(1024-111840.781/F+.4)
60 : PRINTUSING"####.HZ K00: #### ";F,R
65 ::SOUND1,R,10:FORT=1TO400:NEXT
70 NEXT L,J :GETKEYA$
80 DATA C,C#,D,D#,E,E#,F,F#,G,G#,A,A#,H

```

## 2. program

### 3 program

Az 1. program C++-en futtatható, úgynevezett egysoros.

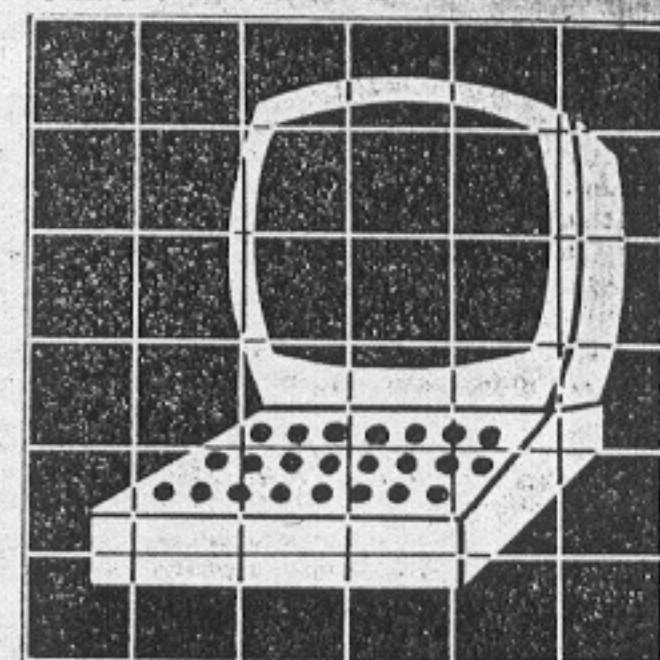
Az AS tartalmazza a különböző hangok megszólaltatásához szükséges értékeket. Az 1-9 billentyük lenyomására megszólal a megfelelő hang.

Amenyiben a félhangokat vagy egy adott frekvenciájú hangot akarunk megszólaltatni, akkor vagy megfelelő táblázatot vagy a hangsorképzés szabályát kell felhasználni.

A sok és unalmas DATA sor beírása helyett jó lenne utánanézni a hangsor-felépítés törvényszerűségeinek. A hangsorban az egymást követő hangok rezgésszáma mértani sor szerint növekszik. A két egymást követő rezgésszám hányadosát a 2. programban a D változó értéke adja. Ez a program így C+4 gépen futatható.

Ha már ilyen szépen megismerkedtünk a hangok programozásával, akkor begépelhetjük a C+4 számítógépbe a 3. programot, amelynek segítségével két oktáv terjedelmű dallam játszható le. A 100—200 sorok a képernyőre felrajzolják a billentyűzetet, és amíg a QWERTY soron, illetve felette a megfelelő helyen lévő számjegybillentyűt lenyomva tartjuk, addig szól a megnyomott billentyűnek megfelelő hang.

A 240-290 sorok biztosítják a billentyűzet figyelését, jelölik a képernyön lé-



**MC-PROGRAMOK**

vő klaviatúrán a megfelelő billentyűt, és biztosítják a hang megszólalását.

A 4. program a hangkeltést csak egyetlen sorban használja fel, amennyiben a morzejelek titáját hallhatóvá teszi. A program biztosítja a betűnként való gyakorlást, de az 54—62 programsorok lehetővé teszik azt, hogy egy beírt szöveg folyamatosan kerüljön „leadásra”. A képernyőn a latin betű mellett a megfelelő morzejel is látható. Az adás ütemét, a titáj és a táj jelek hosszának arányát a program 86. és 90. sorának átírásával változtathatjuk meg.

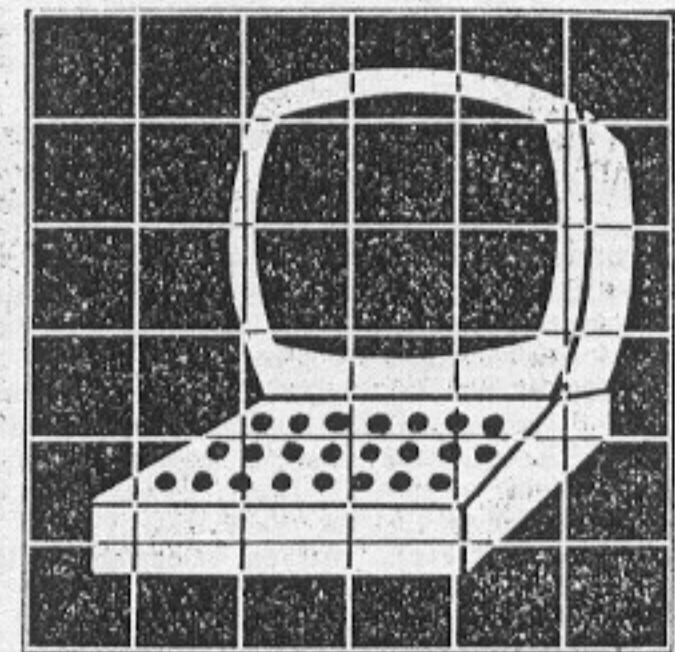
Dr. Sz. Lukács János

```
10 REM           MORZE C16/+4...V. 1
12 :
14 DIM M$(600):FORL=0TO57:READM$(L):NEXT
16 DATA-----,-----,-----,-----,-----,-----,
18 DATA -----,-----,-----,-----,-----,-----,
20 DATA -----,-----,-----,-----,-----,-----,
22 DATA -----,-----,-----,-----,-----,-----,
24 DATA -----,-----,-----,-----,-----,-----,
26 DATA -----,-----,-----,-----,-----,-----,
28 DATA -----,-----,-----,-----,-----,-----,
30 DATA -----,-----,-----,-----,-----,-----,
32 DATA -----,-----,-----,-----,-----,-----,
34 DATA -----,-----,-----,-----,-----,-----,
36 DATA -----,-----,-----,-----,-----,-----,
38 DATA -----,-----,-----,-----,-----,-----,
40 SCNCLR:PRINT"VRLASZTEK":PRINT:PRINT
42 PRINT"B" BETUNKENT":PRINT
44 PRINT"P" PUFFERELT":PRINT
46 GET V$:IF V$<>"B"ANDV$<>"P"THEN 46
48 IF V$="P"THEN GOTO 54
50 PRINT"> ";:GOSUB 64:T$=B$
52 GOSUB 72:GOTO 50
54 PRINT:PRINT"? ";:T$="";B$=""
56 DO
58   PRINT B$;:T$=T$+B$:GOSUB 64
60 LOOP UNTIL B$ =CHR$(13)
62 PRINT:GOSUB 72:GOTO 54
64 GETB$:
66 : IF V$="P"ANDB$=CHR$(13)THENGOTO 70
68 : IF B$<" " ORB$>"Z"    THENGOTO 64
70 RETURN
72 FORL=1 TO LEN(T$)
74   A=ASC(MID$(T$,L,1))-33
76   : IF A>0THEN GOTO 80
78   : PRINT" ";FORI=1TO400:NEXT: GOTO 98
80   : PRINT(MID$(T$,L,1));" ";M$(A)
82   : FORJ=1 TO LEN(M$(A))
84   :   S$=MID$(M$(A),J,1)
86   :   H=-10*(S$=".")-30*(S$="-")
88   :   VOL$:=SOUND1,770,H
90   :   FOR I=0 TO 10*H:NEXT:VOL0
92   :   FOR I=0 TO 40:NEXT
94   :   NEXTJ :VOL0
96   : FORI=0TO50:NEXT
98 NEXT L :RETURN
```

#### 4. PREDICTION

# Rajzolunk!

TECHNISKOLA



```

100 REM C+4 ZENEGER
110 DIM L(12):FOR J=1 TO 12:READL(J):NEXT
120 FOR L=329 TO 359:READJ:POKE L,J:NEXT
130 SV=PEEK(65005):VOLG :H=4:GRAPHIC1,1
140 CIROLE1,5,43,3,3:PRINT1,5,13
150 DRAW,9,10 TO 9,1:SHAPEA$,2,1,9,16
160 DRAW,0,13 TO 12,13:SHAPEB$,0,1,12,19
170 SCNCLR:S=29:FOR Y=60 TO 124 STEP 16
180 DRAW,10,Y TO 310,Y:NEXT
190 DO:SYS 828:POKE 239,0:IF J=13 THEN EXIT
200 : J=PEEK(869):Z=8*(17-J)
210 : K#=R$: IF J=1 THEN K#=B$
220 : GSHAPEK$,S,Z,2
230 : SOUND1GL(J),10:SOUND2,L(J)+H,10
240 : S=S+16:IF S>320 THEN S=20:SCNCLR
250 LOOP
260 END
270 DATA 596,643,685,704,723,770
280 DATA 798,810,834,854,864,881
290 DATA 32,159,255,32,228,255,201,800
300 DATA 240,246,162,13,221,87,3,248
310 DATA 6,202,208,248,76,60,3,142
320 DATA 101,3,96,6,81,87,69,82
330 DATA 34,89,85,73,79,80,61,43
340 DATA 32,6

```

1. program

```

10 REM C 64 SPRITE SZERKESZTO
11 PRINT "J": J=50000
12 PRINT "SPRITE SZERKESZTO":PRINT
13 PRINT "K RETURN = KESZ/TOVABB"
14 PRINT "< CLR > = UJRA KEZDES"
15 PRINT "<SPACE > = PONT IR/TOROL"
16 PRINT "PONT MOZGATAS KURZOR MOZGATOVAL"
17 GETA$: IF A$="" THEN 17
18 PRINT "J"
19 FOR X=1 TO 21
20 :PRINT "....."
21 NEXT
22 P=1024:X=1:Y=1:W=46
23 POKE P+X+40*Y,42
24 GETA$: IF A$="" THEN 24
25 IFA$="":THEN 39
26 IFA$=CHR$(13) THEN 42
27 IFA$="J":THEN RUN
28 POKE P+X+40*Y,W
29 X=X-(A$="M")+(A$="D")
30 Y=Y-(A$="U")+(A$="D")
31 IF X<1 THEN Y=Y-1:X=24
32 IF X>24 THEN Y=Y+1:X=1
33 IF Y<1 THEN Y=Y+1
34 IF Y>21 THEN Y=Y-1
35 W=PEEK(P+X+40*Y)
36 IF W=46 THEN POKE P+X+40*Y,42
37 IF W=224 THEN POKE P+X+40*Y,170
38 GOTO 24
39 IF W=224 THEN 41
40 POKE P+X+40*Y,224:W=224:GOTO 36
41 POKE P+X+40*Y,46:W=46:GOTO 36
42 POKE P+X+40*Y,W
43 FOR F=0 TO 20
44 : FOR X1=0 TO 2
45 : Q=0
46 : FOR X2=0 TO 7
47 : P=1065+40*F+8*X1+X2
48 : IF PEEK(P)=224 THEN Q=Q+2^(7-X2)
49 : NEXT
50 : POKE J+3*F+X1,Q
51 : NEXT
52 NEXT
53 PRINT CHR$(19); " "
54 FOR Y=0 TO 20:FOR X=0 TO 2
55 : PRINT "T"; TAB(26+4*X):PEEK(J+3*Y+X)
56 : NEXT:PRINT:NEXT
57 GETA$: IF A$="" THEN 57
58 PRINT CHR$(19)

```

2. program

A korábbiakban a felrajzolt zongorabillentyűzet mellett a hang előállításáról beszélünk. Most mintegy folytatásként nézzük meg az 1. programot, amely a +4 számítógép esetében egy oktav terjedelmű dal lejátszását teszi lehetővé a QWERTY billentyűk segítségével. A megszólaló hang mellett felrajzolja az ötvonalas kotta megfelelő helyére a hangjegyeket, egy-egy SHAPE-t.

A játékprogramról a +4 hiányosságának tekintik, hogy nincs sprite alkalmazására lehetőség. Egy sprite összeállítása C64 számítógépen sem megy gyorsan. Aki már készített ilyet, az igen sok összesfirkált vázlatrajzot tud mutatni, míg végre elköszült a megfelelő ábra és a kívánt DATA-ba irányadatos.

Ezt a munkát segítheti, könnyítheti meg a C64-en futtatható 2. program. A program segít nagyítva megrajzolni, megszerkeszteni a sprite-ot. Felrajzolja normál méretben is, és ún. kvázi direkt üzemmódban elkészíti a sprite felépítéséhez szükséges DATA sorokat. Érdemes tanulmányozni a programból vezérelt parancs üzemmódot és a billentyű buffer használatának lehetőségét a 77-79. sorokon. A 82. sor pedig elvégzi a Delete 10-82 parancsot, és a programlista így csak a sprite adatait tartalmazó DATA sorokból fog állni.

Ne felejtkezzük el a szerényebb grafikai lehetőségekkel ellátott számítógé-

## MC-PROGRAMOK

pekről sem. Hiszen ezek képernyőjén nehezebb látványos dolgokat művelni. A 3. program a HT képernyőjének tartalmát tükrözi a középpontra. A 20-120. sorokon levő rutin jól felhasználható középpontos tükrözés szemléltetésére, de látványos helyeseréket is be lehet építeni a programba ennek a gépi kódú rutinnak a felhasználásával.

Dr. Sz. Lukács János

59 FORM=1 TO 21

A 2. program folytatása

```

60 : PRINT TAB(26)
61 NEXT
62 V=53248:POKE V+21,4:POKE 2042,13
63 FORM=0 TO 62
64 :POKE 832+N,PEEK(J+H)
65 NEXT
66 POKE V+4,255:POKE V+5,136
67 POKE V+23,12:POKE V+29,12
68 GETA$: IF A$="" THEN 68
69 IFA$=CHR$(13) THEN 71
70 POKE V+21,0:GOT023
71 POKE 2,1:POKE V+21,0
72 PRINT "J":PRINT PEEK(2)," DATA";
73 FOR J=0 TO 6
74 :P=PEEK(49993+PEEK(2)*7+3)
75 :PRINT RIGHT$( "+STR$(P)+",4);
76 NEXT
77 A=A+1:PRINT "M":PRINT "GOT030"
78 POKE 631,19:POKE 632,17:POKE 633,13
79 POKE 634,13:POKE 198,14:END
80 IF PEEK(2)=9 THEN GOT082
81 POKE 2,PEEK(2)+1:GOT072
82 REM POKE 2355,0:POKE 2356,0

```

```

10 REM GRAFIKIT KÉZÖPPONTRA T1KRÉZ HT
20 FOR I=25600 TO 25676:READA:POKE I,A:NEXT
30 DATA 17,0,60,33,255,63,14,2,6,0,126,245
40 DATA 26,205,30,100,119,241,205,30,100
50 DATA 18,19,43,16,240,13,32,235,201
60 DATA 254,128,216
70 DATA 197,6,0,203,111,40,2,203,192,203
80 DATA 103,40,2,203,200,203,95,40,2,203
90 DATA 208,203,87,40,2,203,216,203,79,40
100 DATA 2,203,224,203,71,40,2,203,232
110 DATA 203,248,120,193,201
120 POKE 16526,0:POKE 16527,100
130 CLS:PRINT @479,CHR$(160)CHR$(144);
140 PRINT @543,CHR$(130)CHR$(129);
150 FOR I=0 TO 40:SET(0,I):SET(1,0):NEXT
160 FOR I=0 TO 15:SET(40,I):SET(1,40):NEXT
170 FOR I=15 TO 40:SET(15,I):SET(1,15):NEXT
180 PRINT @322,"SAMU";
190 FOR I=0 TO 80:SET(1,I/2):NEXT
200 H#=INKEY$:IF H$="" THEN 200
210 A=USR(0):GOT0200

```

3. program

# MŰHELYSAROK

Szerencsére (vagy sajnos), egyre nagyobb memóriakapacitású gépek vannak a számítógép-kedvelők kezében, amelyek típusa egyre szaporodik. A nagy szabad memóriaterület kevésbé kényeszeríti a programozót arra, hogy tömör, jól átgondolt, jól szerkesztett programot írjon. Pédig a profi programozók egy-egy ROM-ban a byte-okért harcolnak.

A program áttekinthetőségét növelhetjük REM sorok beírásával. Ez ugyan növeli a helyigényt, de a program elkészülése után, ha indokolt, törölhetjük azokat. A Spectrum számítógépen a REM után kiterjesztett üzemmódban (E) beírt Shift +9 karaktersorral akár villogó megjegyzéssort is készíthetünk. A villogás kikapcsolása E+Shift+8. A C64 (1. program) és C16 (2. program) számítógépeken inverz REM sorokat hozhatunk létre, ha az itt közölt programokat beírjuk futtatjuk majd betölt-

jük, vagy megírjuk azt a programot, amelyben a REM sorokat majd ki akarjuk emelni. Egyetlen feltétel, hogy a REM kulcsszó után legalább egy szóközt kell beírni. Ugyanis SYS 850 után a segédprogram megkeresi a REM kulcssqlakat, és ha van utána egy szóköz, azt lecseréli vezérlő karakterre.

A C16 tulajdonosok tudják, hogy grafika mellett alig marad szabad memóriaterület. A legrövidebb programsor (pl. 10 PRINT) 6 byte hosszú, a leghosszabb pedig 255 byte lehet. Az ilyen irdatlan hosszú sorok előállításának sok buktatója van, de gyakran jól jön az összevonható 6 programsor esetében megnyerhető 30 byte.

A 3. program hosszú egyensorokat igyekszik előállítani, nézzük hogyan? Írjuk be a programunkat, azt amelynek a sorait majd össze akarjuk vonni. Ha működőképes programot akarunk a késsőbbieken, akkor gondolunk arra,

hogy a programsor REM vagy az IE utáni része nem, illetve nem minden kerül végrehajtásra. Ha elkészültünk a programírással, akkor Merge-hez, azaz egy másik BASIC program utántöltéséhez készítsük elő a gépet POKE 43, PEEK (45)-2, POKE 44, PEEK (46). Ezután LIST-re úgy viselkedik a gépünk, mintha nem lenne benne program.

Töltsük be a 3. programot és indítsuk el azt. A tömörítő program annyi sor fog összevonni, egy sorba, hogy annak összhossza ne legyen nagyobb, mint 255 byte.

A program lefutása után óvatosan varázsoljuk elő az eredeti, de most már „hosszúsorosított” változatot. De hogyan? Ezt a 4. program kapcsán beszéljük meg.

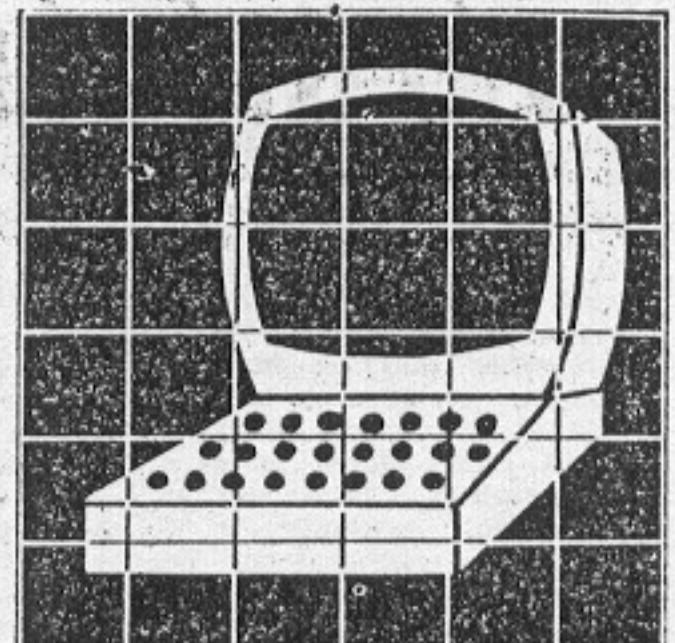
A 4. programot hasonlóan használhatjuk, mint az előzőt, de több a felelősségünk, úgy hogy a tömörítendő programot kezelés előtt célszerű kimenteni. Itt is a beírt és kimentett programot POKE 43, PEEK (45)—2, POKE 44, PEEK (46) parancssal eldugjuk, de előtte (!) jegyezzük fel, hogy hány sort akarunk összevonni. Pl. összevonható a 2. program esetében a 10. és 15. sor, azaz két sor.

Ugyeljünk a sorszám-hivatkozásokra, az összevonás után a kezdősorok megtartják sorszámukat. Töltsük be az 5. programot és az 50. DATA sorba írjuk be a példa szerinti DATA 2, 1, 3, 6, 2, 0 értékeket. Az összevonás yégét a DATA sor utolsó elemeként besírt 0 jelzi a programnak. Most már indítható a program.

A tömörítés elvégzése után a prog-

ram klírja a teendőket. Ugyanezt kell tenni a 3. program esetében is. A viszszahozott program átsorszámozható (ha minden jól csináltunk), futtatható, de már nem javítható a szokásos editálással.

## TECHNISKOLA



## HC-PROGRAMOK

Megjegyzés a 66. sorhoz; ha „kinyúvasztottunk” egy programot, azt monitor segítségével általában egyszerűen visszahozhatjuk. A Bitlet korábbi ötlejtést bonekolgatva a Renumber rutinlépései alapján a mi ötleünk C16 (C+4 esetében, ha még nem voltunk grafikus üzemmódban) NEW után a program visszahozható POKE 4097.1 SYS34840: SYS 34891, vagy a programban ajánlott POKE 4097: DELETE 1 parancssal.

No igen, de mit csináljon egy szegény C64 tulajdonos, ha véletlenül kiadta a NEW parancsot. Erre gondolva vegye fel jó előre az 5. programot kazettára vagy lemezre és a NEW után kéresse meg ezt a jól használható segédprogramot. Majd LOAD „OLD”. 8. 1. másodlagos cím beírásával gyorsan töltse be. Ez a kis program automatikusan elindul és visszaállítja az eredeti programot a tárban.

Dr. Sz. Lukács János

```
10 REM  
15 REM  
20 REM  
25 :  
30 FORL=850. TO 910:READJ:POKEL,J:NEXT  
35 :  
40 REM  
45 REM  
50 REM  
55 :  
60 DATA 120,165, 43,133,251,165, 44,133  
65 DATA 252,160, 0,177,251,201,143,240  
70 DATA 27, 24,165,251,105, 1,133,251  
75 DATA 165,252,105, 0,133,252,165,252  
80 DATA 197, 46,208,231,165,251,197, 45  
85 DATA 208,225, 88, 96,200,177,251,136  
90 DATA 201, 32,208,221,200,169, 18,145  
95 DATA 251,136, 56,176,212,SAMU
```

## 1 program

```
10 REM
15 REM
20 FORL=850 TO 910:READJ:POKEI,J:NEXT
25 REM
30 REM
35 REM
40 DATA 120,165, 43,133,254,165, 44,133
45 DATA 255,160, 0,177,254,201,143,240
50 DATA 273,24,165,254,105, 1,133,254
55 DATA 165,255,105, 0,133,255,165,255
60 DATA 197, 46,208,231,165,254,197, 45
65 DATA 208,225, 88, 96,200,177,254,136
70 DATA 201, 32,208,221,200,169, 18,145
75 DATA 254,136, 56,176,212,86II
```

## 2. program

```

10 REM >>>TÖMÖRÍT A'LA PROKRUSTES<<
15 V=PEEK(43)+PEEK(44)*256-3
20 IF V<4100 THEN PRINT "IGY NEM JO":END
25 DIM K(20) : K(0)=4097 : KE=4101
30 FOR I=KE TO V : A=PEEK(I): IF A<>0 THEN 45
35 : K=K+1 : K(K)=PEEK(I+1)+256*PEEK(I+2)
40 : I=I+4: IF K(K)-K(0)>260 THEN GOTO 50
45 NEXT I : END
50 K(0)=K(K-1) : Z=0
55 FOR F=KE TO I-5 : L=F-Z*4: J=PEEK(F)
60 : IF J=0 THEN F=F+4: Z=Z+1 : J=58
65 : POKEL,J
70 NEXT F : POKEL+1,0
75 FOR I=F TO V : POKE(I-Z*4),PEEK(I):NEXT I
80 FOR P=I-Z*4 TO V : POKEP,0 :NEXT P
85 K=0 : KE=I+6 : V=V-Z*4 : GOTO 30

```

### 3. program

```

10 REM PROGRAM TÖMÖRÍTŐ / V. SOROKAT
12 V=PEEK(43)+PEEK(44)*256-3
14 IFV<4100THENPRINT"IGY NEM JO!" :END
16 DIM K(20) :K(0)=4097 :KE=4101
18 READS :IF S=0 THEN GOTO 52
20 FOR I=KETOI :A=PEEK(I):IF A>0THEN28
22 K=K+1:K(K)=PEEK(I+1)+256*PEEK(I+2)
24 IF K(K)-K(0)>250+K*4THENPRINT"BAJ VAN"
26 I=I+4:IF K=S THEN30
28 NEXT :END
30 K(0)=K(K-1) :Z=0
32 FOR F=KETOI-5 :L=F-Z*4:J=PEEK(F)
34 IF J=0 THEN F=F+4 :Z=Z+1 :J=58
36 POKE L,J
38 NEXT :POKEL+1,0
40 FOR I=FTOV:POKE(I-Z*4),PEEK(I):NEXT
42 FOR P=I-Z*4TOV :POKEP,0:NEXT
44 K=0 :KE=L+6 :V=V-Z*4:GOTO18
46 :
48 REM DATA SORBAN AZ ÖSSZEVONÁNDÓ SOROK
50 DATA PÉLDÁUL: 5,3,3,7,1,1,3,0
52 REM ITT A VÉGE/EREDETI PR VISSZA
60 PRINT"ADD KI A KÖVETKEZŐ PARANCSOKAT!"
62 PRINT
64 PRINT"POKE 43,1:POKE44,16:NEW"
66 PRINT"POKE4097,1:DELETE1"

```

#### 4. program

```
10 REM *****OLD /64
12 OPEN1,8,1,"OLD,P,W"
14 : FORL=0TO40
16 : READJ:PRINT#1,CHR$(J);
18 : NEXT
20 CLOSE1
22 DATA221, 2,165, 44,160, 1,145, 43
24 DATA 32, 51,165,165, 34, 24,105, 2
26 DATA133, 45,165, 35,105, 0,133, 46
28 DATA169,131,162,164,141, 2, 3,142
30 DATA 3, 3,193, 3, 3,139,222,221, 2
```

## 5. program

# SZÁMOLJUNK PONTOSABBAN

A számítógépek csak korlátozott pontossággal képesek számolni. Visszának tűnlök ez a megállapítás, hiszen a HT

dupla pontosságú aritmetikája 16 jegy pontossággal dolgozik.

De mit kell tenni akkor, ha a géptí-

pusok által biztosított lehetőségnél több tizedesjegyre van szükség?

Nézzük először az osztást.

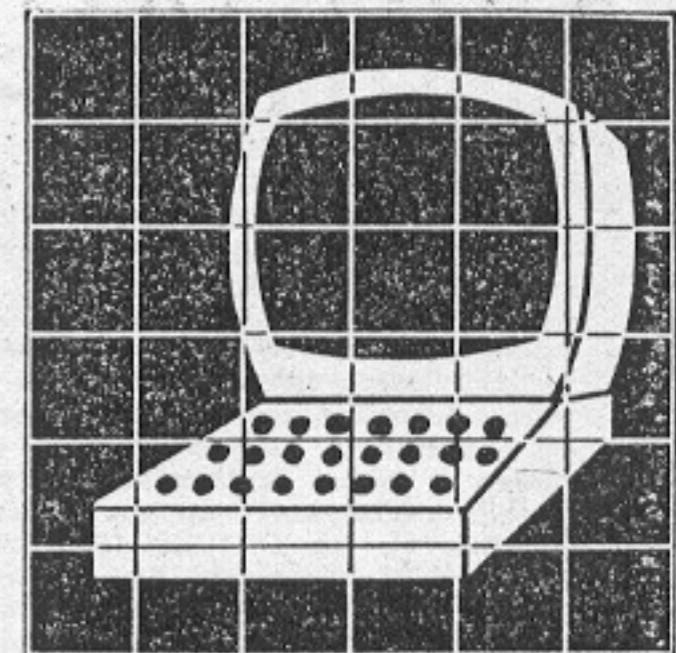
Programozzuk be a géphez azt az algoritmust, amely szerint annak idején mi magunk, papíron, ceruzával elvégeztük az osztást. Valahogy úgy kellett csinálni, hogy a maradék tízszeresét osztottuk az osztóval és ezt ismételtük a „végkimerülésig” (ábra). Ez a maradék elég körülmenyesen állítható elő a szokásos hazai számítógépeken. Az ENTERPRISE-on és a nagyobb gépeken már van maradék függvény is.

A feladat egy lehetséges megoldását az 1. program mutatja.

Az összeadás és kivonás 20 jegy pontosságú megvalósítását biztosítja a 2. és 3. program. Ezek a műveletek helyi értékenként elvégzik és kiírják az eredményt, de ha keletkezett átvitel, akkor azt is figyelembe veszik. Annak érdekében, hogy a műveleteket helyi értékenként el tudjuk végezni, a számot „szelőtelni” kellene. A szelőtelést stringek esetében könnyű elvégezni, ezért a számokat ebben a feladatban stringként kezeljük.

A 4. program a szorzás kivitelezésére mutat megoldást azzal a kikötéssel,

## TECHNIKOLA



## MC-PROGRAMOK

hogy itt max. tízjegyű szám 10 jegyűvel való szorzása végezhető el.

Ezek a programok C+4-re készültek.

Átírásuk más géptípusra nem okozhat nehézséget, de figyelembe kell venni a string szelőtelését és a képernyő adott helyére történő kiíratását, géptől függő sajátosságait.

Dr. Sz. Lukács János

```

10 REM STRING SZORZAS - EGESZ SZAMRA
12 N=10 :DIME$(2*N),V$(2*N)
14 N$="" :FOR I=1 TO N:N$=N$+"0":NEXT
16 INPUT A$:A$=RIGHT$(N$+A$,N) :S=0
18 INPUT B$:B$=RIGHT$(N$+B$,N) :S=0
20 CHAR1,3,5,A$+"*"+B$ :A=0
22 CHAR2,6,"-----"
24 FOR L=1 TO N :E$="":V=VAL(MID$(B$,L,1))
26 IF V=0 THEN E$="0000000000" :GOTO 40
28 FOR S=1 TO V :FOR I=N TO 1 STEP -1
30 E=VAL(MID$(E$,I,1))+VAL(MID$(A$,I,1))
32 E=A+E :IF E>9 THEN A=1 :ELSE A=0
34 E$(I)=RIGHT$(STR$(E),1)
36 CHAR,L+1+I,L+6,E$(I):NEXT
38 E$="" :FOR I=1 TO N :E$=E$+E$(I):NEXT :NEXT
40 CHAR,L+2,L+6,E$ :A=0
42 V$(L)=LEFT$(N$,L-1)+E$+LEFT$(N$,N-L)
44 NEXT :A=0 :CHAR2,6,"-----"
46 FOR I=N+2-1 TO 1 STEP -1 :S=A :FOR F=1 TO N
48 S=S+VAL(MID$(V$(F),I,1)) :S$=STR$(S)

```

4. program.

1. program.

1. program.

## C 16 tuning

A GOTO (aritmetikai kifejezés) alakú utasítást csak a Sinclair gépek fogadják el, igaz a többi géptípuson megtalálható az ON A GOTO utasítás.

A kiszámított GOTO sokszor előnyös lenne a Commodore gépeken is.

A C 16-hoz itt most egy ötletet mutatunk be.

```

10 REM KISZAMITOTT GOTO C16
20 DATA 174,0,4,134,20,174,1
30 DATA 4,134,21,76,80,141
40 FOR I=1026 TO 1038:READ A:POKE I,A:NEXT
50 REM PROBAFUTASOK
60 INPUT A:IF A=0 THEN END
70 POKE 1025 ,INT(A/256)
80 POKE 1024 ,A-256*INT(A/256)
90 SYS 1026 :REM EZ A 3 SOR A GOTO N
100 PRINT "HIBA":END
110 PRINT "E2100":GOTO 60
120 PRINT "E2200":GOTO 60
130 PRINT "E2300":GOTO 60
140 PRINT "E25120"
200 PRINT "E2100":GOTO 60
300 PRINT "E2200":GOTO 60
400 PRINT "E2300":GOTO 60
512 PRINT "E2512"

```

2. program.

3. program.

```

100 REM OSZTAS Sok jegy pontossagra
110 INPUT "hany jegyig szamoljon";J
120 INPUT "osztando";A
130 INPUT "oszto";B
140 IF B=0 THEN GOTO 130
150 PRINT INT(A/B);".";
160 GOTO 180
170 PRINT INT(A/B);
180 LET M=A-INT(A/B)
190 LET A=10*M
200 LET J=J-1
210 IF J>0 THEN GOTO 170
220 END

```

```

10 REM STRING OSSZEADAS EGESZ SZAMRA
15 N$="000000000000000000000000":DIME$(20)
20 INPUT A$:A$=RIGHT$(N$+A$,20)
25 INPUT B$:B$=RIGHT$(N$+B$,20):S=0
30 CHAR,3,10,A$+CHAR,2,11,"+"+B$:A=0
35 CHAR,2,12,"-----"
40 FOR I=20 TO 1 STEP -1
45 E=VAL(MID$(A$,I,1))+VAL(MID$(B$,I,1))
50 E=A+E :IF E>9 THEN A=1 :ELSE A=0
55 E$(I)=RIGHT$(STR$(E),1)
60 CHAR,I+2,13,E$(I)
65 NEXT

```

```

10 REM STRING KIVONAS EGESZ SZAMRA
15 N$="000000000000000000000000":DIME$(20)
20 INPUT A$:A$=RIGHT$(N$+A$,20):E$=""
25 INPUT B$:B$=RIGHT$(N$+B$,20):S=0
30 CHAR,3,10,A$+CHAR,2,11,"-"+B$:A=0
35 CHAR,2,12,"-----"
40 IF B$>A$ THEN S$=A$:A$=B$:B$=S$:E$="-"
45 FOR I=20 TO 1 STEP -1
50 E=VAL(MID$(A$,I,1))-VAL(MID$(B$,I,1))
55 E=E-A :IF E<0 THEN E=E+10:A=1 :ELSE A=0
60 E$(I)=RIGHT$(STR$(E),1)
65 CHAR,I+2,13,E$(I)
70 NEXT:CHAR1,2,13,E$

```

# A SZÁMOLÁSI KÉSZSÉG FEJLESZTÉSE

Az itt közölt programok az általános iskolák alsó tagozatosai által használhatók, és több-kévesebb játékossággal biztosítják az alapműveletek begyakorlását.

A számkigyó című 1. program a C+4 képernyőjére felrajzol egy  $10 \times 10$ -es házat, és ebbe 100 db, 0 és 9 közé eső véletlen számot tölt be.

Feladat: úgy megtenni 25 lépést ezen a felületen, hogy az érintett számok összesze a lehető legnagyobb legyen.

Ehhez a kurzormozgató nyilakkal ki-

kell választani a megfelelő kezdőpontot. Ha kijelöltük a kezdőpontot, akkor a Return lenyomása után továbbra is mozoghatunk a táblán, de most már a gép számolja pontjainkat és jelzi a megtett utat. Vonalunk nem kereszthető önmagát és nem lehetünk le a mezőről. A 25. lépés megtétele után értékeli a teljesítményt.

Egyeszerű összeadás-, kivonás-, szorzás- és osztásgyakorló programot mutatunk be a 2. programmal.

```

10 REM SZAMKIGYÓ +4
11 COLOR1, 1, 0 : COLOR0, 3, 7 : COLOR4, 6, 5
12 W$=" "
13 POKE65303, 105 : POKE65302, 39
14 K=0 : SCNCLR : PEEK(65286)OR64
15 S=0 : CHAR, 2, 3, W$ : FORL=2TO20STEP2
16 FORJ=4TO22STEP2 : FORL=2TO20STEP2
17 CHAR, L, J, STR$(INT(RND(1)*10))
18 NEXT : CHAR, L, J, " "
19 CHAR, 2, J+1, W$ : NEXT : CHAR, 2, 23, W$+" "
20 X=0 : Y=0 : D=3235 : POKED, PEEK(D)-64
21 POKED-1024, 127 : W=25 : GOSUB52
22 CHAR, 8, 0, "VALASSZ KÉZDOPONTOT"
23 GETKEYA$ : DE=D : IF A$=CHR$(13) THEN 27
24 GOSUB47 : IF DE=D THEN 23
25 POKED, PEEK(DE)+64 : POKED, PEEK(D)-64
26 POKED-1024, 16 : POKED-1024, 127 : GOT023
27 CHAR, 8, 0, " "
28 S=S+PEEK(D)-112 : K=K+1 : GOSUB52
29 IF K=W THEN 41
30 DE=D : GETKEYA$ : GOSUB47 : IF DE=D THEN 30
31 IF PEEK(D)<127 THEN D=DE : GOT030
32 IF D=DE-80 THEN POKED-40, 96
33 IF D=DE+80 THEN POKED+40, 96
34 IF D=DE+ 2 THEN POKED+ 1, 96
35 IF D=DE- 2 THEN POKED- 1, 96
36 POKED, PEEK(D)-64 : POKED-1024, 127
37 IF Y>0 AND PEEK(D-80)>127 THEN 28
38 IF Y<9 AND PEEK(D+80)>127 THEN 28
39 IF X<9 AND PEEK(D+ 2)>127 THEN 28
40 IF X>8 AND PEEK(D- 2)>127 THEN 28
41 IF M<STHEN M=S : GOSUB52 : ELSE GOSUB52
42 CHAR, 8, 0, "UJ JATEK I/N ? "
43 POK239, 0 : GETKEYA$ : IF A$="I" THEN GOT014
44 IF A$<>"N" THEN GOT043
45 POK65286, PEEK(65286)AND191
46 CHAR, 8, 0, "AKOR VISZ LATASRA " : END
47 IF A$="I" AND X>0 THEN D=D-2 : X=X-1
48 IF A$="I" AND X<9 THEN D=D+2 : X=X+1
49 IF A$="O" AND Y>0 THEN D=D-80 : Y=Y-1
50 IF A$="O" AND Y<9 THEN D=D+80 : Y=Y+1
51 RETURN
52 CHAR, W, 4, "LEPES:" +STR$(K)
53 CHAR, W, 6, "EDDIG:" +STR$(S)
54 CHAR, W, 8, "MAX :" +STR$(M) : RETURN

```

## 1. program.

```

10 REM ALAPMUVELET GYAKOROLTATO +4
20 INPUT "A TAGOK/TENYEZOK MAX. ERTEKE" ; M
30 SCNCLR
40 A=INT(M*RND(0)+1) : B=INT(M*RND(0)+1)
50 PRINT A; "+" ; B; "=" ; INPUT E
60 REM ! VALTOZIK HA MAS MUVELET
70 IF E>A+B THEN PRINT "PONTOSABBAN" : GOT050
80 PRINT "UGYES" : GOT040

```

## 2. program.

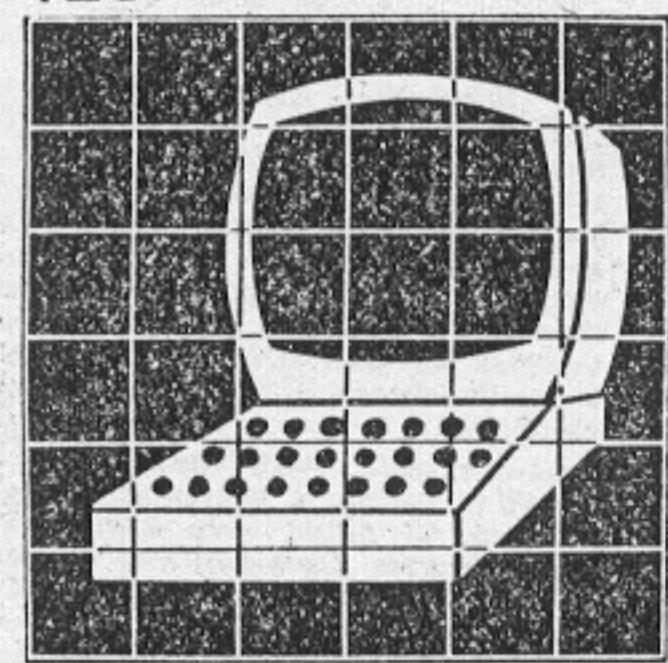
```

10 REM OSZTAS GYAKOROLTATO +4
20 INPUT "A TENYEZOK MAX. ERTEKE" ; M
30 SCNCLR
40 A=INT(M*RND(0)+1) : B=INT(M*RND(0)+1)
50 PRINT A*B; "/" ; B; "=" ; INPUT E
70 IF E>A THEN PRINT "PONTOSABBAN" : GOT050
80 PRINT "UGYES" : GOT040

```

## 3. program.

# TECHNIKOLA



## MIC-PROGRAMOK

tést. A program az egyes lépésekkel rendezi kiírja a képernyőre.

Dr. Sz. Lukács János

```

10 REM BUVOS-NEGYZET +4
11 A$="I   |   |   |" : B$="I   |   |   |" : C$="L   |   |   |" : SCNCLR : H=12
12 B$="I   |   |   |" : C$="L   |   |   |" : SCNCLR : H=12
13 C$="L   |   |   |" : SCNCLR : H=12
14 CHAR, H, 5, A$ : CHAR, H, 6, B$ : CHAR, H, 7, B$ : SCNCLR : H=12
15 CHAR, H, 8, A$ : CHAR, H, 9, B$ : CHAR, H, 10, B$ : SCNCLR : H=12
16 CHRR, H, 11, A$ : CHAR, H, 12, B$ : CHAR, H, 13, C$ : SCNCLR : H=12
17 F=(F-3*A)/3 : E=3*INT(A+F*RND(0)) : R=E/3
18 B=INT(E/2*RND(0)) : C=INT(E/2*RND(0))
19 R(2)=E-C-A : A(3)=E-C-B : R(4)=E-R-R(3)
20 A(1)=E-B-A : A(5)=E-B-A(2) : A(6)=E-C-A(1)
21 CHAR, 28, 9, "=" +STR$(E) : CHAR, 18, 9, STR$(R)
22 CHAR, 13, 6, STR$(B) : CHAR, 13, 12, STR$(C)
23 FORI=1TO6 : READ0, S : CHAR, 0+1, S, "?" : PRINT "?"
24 : CHAR, 0, 17, " "
25 : INPUTX : IF X>R(I) THEN GOT024
26 : CHAR, 0, S, STR$(X)
27 NEXT
28 CHAR, 0, 17, "MEG EGY JATEK I/N ?"
29 GETKEY A$ : IF A$="I" THEN RUN
30 PRINT "VISZONTLATASRA" : END
31 DATA23, 12, 23, 6, 13, 9, 23, 9, 18, 6, 18, 12

```

## 4. program.

```

10 REM MUVELETEK TORTEKKEL
11 INPUT "ELSO TORT SZAMLAJOJA" ; S1%
12 INPUT "ELSO TORT NEVEZOJE" ; N1%
13 INPUT "MASODIK TORT SZAMLAJOJA" ; S2%
14 INPUT "MASODIK TORT NEVEZOJE" ; N2%
15 S1=S1% : S2=S2% : N1=N1% : N2=N2%
16 IF N1=0 OR N2=0 OR S2=0 THEN PRINT "RUN"
17 INPUT "MUVELET" ; M$ : PRINT
18 L=S1 : J=N1 : GOSUB38 : L=S2 : J=N2 : GOSUB40
19 PRINT : IF M$="+" OR M$="-" THEN GOT024
20 L=S1 : IF M$="*" THEN PRINT : GOT023
21 IF M$<>/"/" THEN GOT017
22 S=N2 : N2=S2 : S2=S : M$="*" : PRINT
23 J=N1 : GOSUB38 : L=S2 : J=N2 : GOSUB40 : GOT030
24 N=N1 : PRINT --REM LKKT
25 IF N2<0 INT(N/N2) THEN N=N+N2 : GOT025
26 K=N : R=K/N1 : B=K/N2 : S1=S1*R : S2=S2*B
27 L=S1 : J=N1 : GOSUB38 : L=S2 : J=N2 : GOSUB40
28 U=S1+S2*(M$="-")-S2*(M$="+")
29 PRINT U//K : GOT031
30 U=S1*S2 : K=N1*N2 : PRINT U//K
31 REM LNK0
32 PRINT : O=1 : FORI=2TOK : R=U/I : B=K/I
33 IF A=INT(R) AND B=INT(K) THEN O=I
34 NEXT : IF M$<>"*" THEN GOT036 : ELSE S1=J=N1
35 GOSUB38 : L=S2 : J=N2 : GOSUB40 : GOT037
36 L=S1 : J=N1 : GOSUB38 : L=S2 : J=N2 : GOSUB40
37 PRINT U//K//O//K/O : PRINT : PRINT : RUN
38 PRINT USING "### /###" ; L, J : PRINT M$;
39 RETURN
40 PRINT USING "### /###" ; L, J : RETURN

```

## 5. program.

# OTTHONI SZÁMÍTÓGÉPEK

Immár több mint hat éve juthat hozzá az érdeklődő többé-kevésbé megfizethető áron olyan masinához, amelyek a Neumann-féle ismérveket kielégítve számítógépnek tekinthetők. Természetesen — mint lépten nyomon hallhatjuk és tapasztalhatjuk — e téren a fejlődés rohamleptékkel történt és történik.

Mivel minden vannak új és újabb vázárlók, azok kedvéért nézzük meg egy csomorra való szempontot, ami alapján döntethetnek, melyik számítógépet veszik meg a családnak. A családnak, sőt a gyerekeknek, mert általában, ahogy korábban a villanyvasutat is a gyerekek kapta ajándékba, azért a szülők is eljátszottak vele. Ugyanúgy vagyunk manapság a számítógép megvásárlásával is.

A számítógép kiválasztása előtt néhány kérdést tisztázni kell.

- játékprogramokat akarok futtatni,
- meg alkarrak tanulni alepfokon programozni,
- szeretnék betekinteni a gép működésébe, gépi kódú programot akarok írni,
- minden nap munkában tervezésre, adatnyilvántartásra, szövegszerkesztésre akarom használni,
- vezérlésre, szabályozásra.

## Játszunk

A számítógépprogram-gyártók tudják a legjobban, hogy egy-egy programot aránylag hamar megvannak a felhasználók, valamint hogy a hivatalos csatornán eladtott programszám és a közékkörben forgó programok száma között nagyságrendi eltérés lehet. Ezért, hogy a piaci forgalmat fenn tudják tartani, állandóan új és újabb programokat dobnak piacra. Ezt a stratégiát a két klasszikusnak tekintető cég, a Sinclair és a Commodore alkalmazta a leghatékonyabban. És mert ők voltak az úttörők, az eredeti ötleteket tartalmazó játékprogramok döntő hányadát is a Sinclair Spectrumra és a Commodore 64-re írták.

A későbbi született számítógép-típusokra általában az e két „nagy öreg” által felismert és elkészített játékprogramokat átírták. Sajnos nemelyik cég az üzleti sikert szem előtt tartva az agresszívítás, az izléstelenség irányába eltolódó softvereket is piacra dob.

A hazai piacón és a számítógép-klubokban a legnagyobb játékprogram-választék a C-64, a Spectrum, a C+4 és az Enterprise gépekre van. Ezekhez a játékprogramokhoz különöző gyűjteményekben magyar nyelvű leírás is kapható (LSI, Novotrade kiadványai). Természetesen ha játékprogramokat akarunk futtatni a gépen, akkor feltétlenül be kell szerezni valami „strapabíró” bot-kormányt (joystick). A játékprogramok egy része ugyan irányítható billentyűzetről, de a játék kezelése nehézkessé válik, és bizony elég sok sérült billentyűtől találkozik az ember, főleg a Sinclair gépeken. A hiba általában a földi a „TÚZ” billentyű alatt tapasztalható.

## Tanuljunk programozni

A személyi számítógépek mindegyike nélküli keresztmetszetben BASIC nyelven programozható. Ez a programozási nyelv az angol betűszó rövidítésére a kezdők általános célú szimbolikus nyelv. Mint ilyen általános célú „barkácskészlet”, sok feladat elvégzésére alkalmas, de ha

váltaki egyszer komolyan akar dolgoznál, ekkor jobb, ha célszerszámot alkalmaz, azaz a feladat megoldására orientált programozási nyelvet.

Természetesen a kezdő programozó esetében igen sok helyre kell figyelni. Olyan ez, mint a gépkocsivezetés. Mennyivel egyszerűbb egy automata sebeségváltóval szerelt gépkocsin tanulni a jármű vezetését, abban több idő marad a szabályok, a környezet figyelemre, nem kell bajlódni a kuplunggal és a sebeségváltóval.

A mi esetünkben a kezdőnek sajnos alapvető problémája van a kifejezésmóddal (a kulcsszavak ismeretével), és az írógép billentyűtől is nehezen találja meg. Ezt is figyelembe vették a Sinclair gépek szerkesztői. Egyetlen billentyű lenyomásával egy egész kulcsszó „frödik” be a programba.

A billentyűlenyomások számának csökkenését biztosítja a Commodore gépeken a kulcsszavak rövidítésének lehetősége. A szép, szabatos programírára kényszerítő a gép előtt ülöt az Enterprise. Ez a kényszerítés azonban több programozási (helyesírási) szabály ismeretét követeli meg.

A barátságos BASIC nyelvjárások alig követelnek meg számítástechnikai (angol nyelvtudásbeli) ismereteket. Ezek olyan igénytelenek, mint a gépjárművek közül a Trabant. A konforthatabb BASIC-ek kulcsszókészlete választékban több lehetőséget biztosít, de a programozónak is többet kell tanulnia, hogy a gép lehetőségeit (BASIC-kel) ki tudja használni. Véleményem szerint a legbarátosabb a Spectrum, a legkomfortabban az Enterprise BASIC-je a személyi számítógépek kategóriájában.

A programírás során sok szempontot kell figyelembe venni. Pl.: Mit szeretnék a számítógéppel megoldani? Rajzolni, színeket alkalmazni, mozgásokat bemutatni, zenélni, vagy „csak” feketén-fehér feladatokat megoldani?

## Rajzoljunk

A személyi számítógépen a magunk szórakoztatására készítjük a programot. Elég közelről személjük a képernyőt ahhoz, hogy ne a fantáziára kelljen bízní, hogy mit látunk az pl. valóban kör-e? Pedig így volt ez a ZX 81 és a HT számítógépekkel. Ma már joggal vártuk el számítógépünkön, hogy a képernyőt legalább  $256 \times 196$  képponton címezhetük, akár BASIC-kel is.

A fiatabb korosztály szívesen rajzol a képernyőre. Egyszerűen alkalmazható grafikus utasításkészlete van a Spectrum, a C+4, a Videoton és az Enterprise számítógépeknek már alapkiépítésben.

A C-64 esetében BASIC utasítással egyszerűen csak valamelyen segédprogram betöltése után használhatjuk rajzok készítésére a számítógépet. Grafika kezelése szempontjából ma a legsokoldalúbb lehetőséget az Enterprise biztosítja.

A színek kezelése szempontjából a szűk keresztmetszet nem annyira a számítógép, mint inkább az alkalmazott televízió jelenti. Ha a C+4 és Enterprise számítógépek teljes színskáláját élvezni akarjuk, akkor bizony már színes monitor használatát is fontolhatjuk kell venni. A színek előállításához az RGB (vörös, zöld, kék) bemenettel elláttott monitor használata célszerű pl. Sinclair QL és Enterprise esetében.

Mind játékprogramok, mind oktatóprogramok esetében fontos szerepe lehet a mozgások bemutatásának, érzékelhetősének. E szempont alapján a C-64 spritekezelése (főleg, ha pl. Simon's Basic bővíttető használunk) egyszerű és nagyszerű hatások elérését teszi lehetővé. Hasonlóan segíti a Spectrum gépek esetében a képernyő vagy részeinek mozgatását az ott használható Beta Basic bővítmény. A képernyő (vagy részeinek) gyors váltását biztosítja — és ehhez elegendő memoriája is van — az Enterprise számítógép.

## Hanghatások, zene

Könnyen belátható, hogy a hang minősége és programozásának bonyolultsága között fordított arányosság áll fent. Akár mi magunk is írhatsunk HiFi-minőségen megszólaló zenét C-64 számítógépen, de bizony ehhez igencsak ismesni kell a SID programozásának összetett szabályait.

A HT, Primo, Spectrum, Videoton TVC, C+4 számítógépeken egyszerűen szólaltathatunk meg hangokat, de ezek minősége BASIC szintű programozás esetében inkább csak „saját fogyasztásra” felel meg.

Az Enterprise hangja attól függően változik, hogy saját hangszerű vagy erősítőt alkalmazunk. Vele is elérhető a HiFi-minőség, de inkább a hardverkiegészítés szab határt az elérhető eredménynek.

A számítógép forgalmazói felismerők a hangprogramozásban rejlő nehézségeket, így egy sor zenei program frását segítő utility program kapható a számítógéphez, és mert a programelláttság a C-64-re a legszélesebb körű, e ténben is itt található a legtöbb.

## Gépi kódú programozás

A mai házi használatú számítógépek központi processzora 8 bites, Z 80 vagy 6510, illetve ezek továbbfejlesztett változatai. E processzorok programozásához a gépi kódú utasításkészletet kell alaposan ismerni. Ezen ismeretek elsajátításához az LSI, a Novotrade, az IPIK és a Műszaki Könyvkiadó kezelésében bőséges irodalom áll rendelkezésre.

A gépi programok beírása a számítógéphez már feltételez bizonys ismereteket, illetve megfelelő Monitorprogramokat. E szempontból a legegyszerűbb a dolgunk C+4 (C-16, C-116) számítógép esetében, hiszen itt rendelkezésre áll a beépített monitorprogram.

A többi számítógép esetében azonban ezeket a monitorprogramokat be kell tölteni a számítógépekre.

A gépi kódú programozáshoz sok esetben célszerű az adott számítógép belső — ROM-ban tárolt — programját vagy annak részleteit felhasználni.

## A számítógép alkalmazása

A itthon általában használt személyi számítógépek ROM-fejései beszerezhetők. Tehát a Spectrumhoz és a HT-hez az Ipari Informatika, a Primo-hoz használható a HT ROM-fejése, a C-64-hez Ipari Informatika, LSI, Novotrade, a C+4 (C-16)-hoz LSI, Novotrade, a Videotonhoz a Novotrade kiadása kapható,

míg az Enterprise-hoz a Novotrade munkája szerkesztés alatt van.

Nyilvánvaló, hogy az emberek zöme nem programot akar írni, hanem egy bizonyos feladatának megoldására akarja felhasználni a gép hardver és szoftver adta lehetőségeit. Ebből a szempontból a legátigondolatból és legsikeresebb munkát a C-64 esetében végeztek a forgalmazók. Ugyan e gépet még új korában sem lett volna szabad nagyvállalati szinten számítógépnek tekinteni, de háztartási szinten, pár fős gmk szintjén valóban alkalmas lehet a gép adminisztrációs, adatnyilvántartási feladatok megoldására. A tartós piaci sikér titka a megbízható hardver mellett a sokféle szoftver.

Sajnos még az oly jó ellátott C-64 esetében is az egyes programokat a konkret feladathoz adaptálni kellene, és ez nem meg egyszerűen. A többi személyi számítógép konkrét felhasználói programválaszték sajnos szegényes. Az adatnyilvántartás, levelezés számítógépes intézése megfelelő periférielláttságot telez fel.

Ma már a hajlékonylemezes floppy és a nyomtató is természetes tartozéka kell legyen egy számítógépnek, még akkor is, ha csak „háztájiban” használjuk. Ez a gyártói és forgalmazói oldalról biztosított.

A felhasználó részben anyagi okok miatt néha még kénytelen háttértárolóként magnót használni. Ennek azonban sem a sebessége, sem a megbízhatósága nem elégítheti ki a munkavégző igényét.

Azok (a boldog emberek), aik főhivatalukat munkájukat otthon hobbiként tudják folytatni, célszerű, ha olyan számítógéppel, legalább lemezformátummal dolgoznak, amely a munkahelyi AT, XT és az otthoni kis gép közötti adatforgalmat biztosítani tudja. E szempontból szinte korlátlan lehetőséget biztosít az Enterprise IS-DOS operációs rendszere.

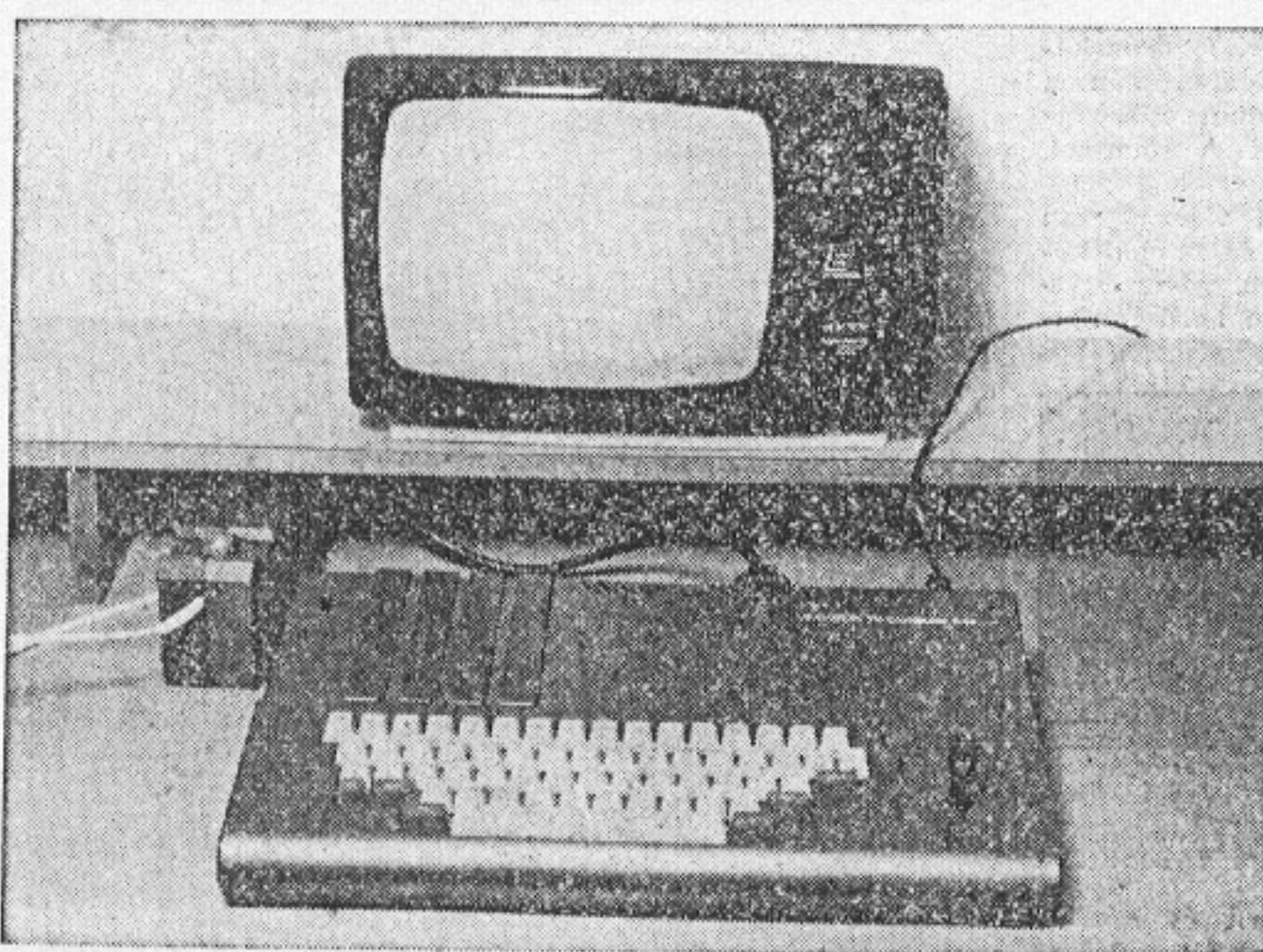
## Vezérlés, szabályozás, hálózat

Ezzel kapcsolatban egyrészt sokkal több feladat elvégzésére készítik fel a számítógépeket, másrészt BASIC-kel nem is minden valósítható meg e téren. Sokkal olcsóbb megoldások is elégégesek lehetnek szabályozási feladatok megoldására. A bővítőhöz és a felhasználói porthoz általában ne nyújtanak célszerű, a kezdők. A hálózat több számítógép közös üzemeltetését feltételezi, s azzal képest, hogy az első megvásárlásán töprengünk, talán korai erről beszélni. A legegyszerűbben kapcsolhatók össze az Enterprise és a Spectrum számítógépek, igaz ez utóbbi feltételezi az Interface I. meglétét.

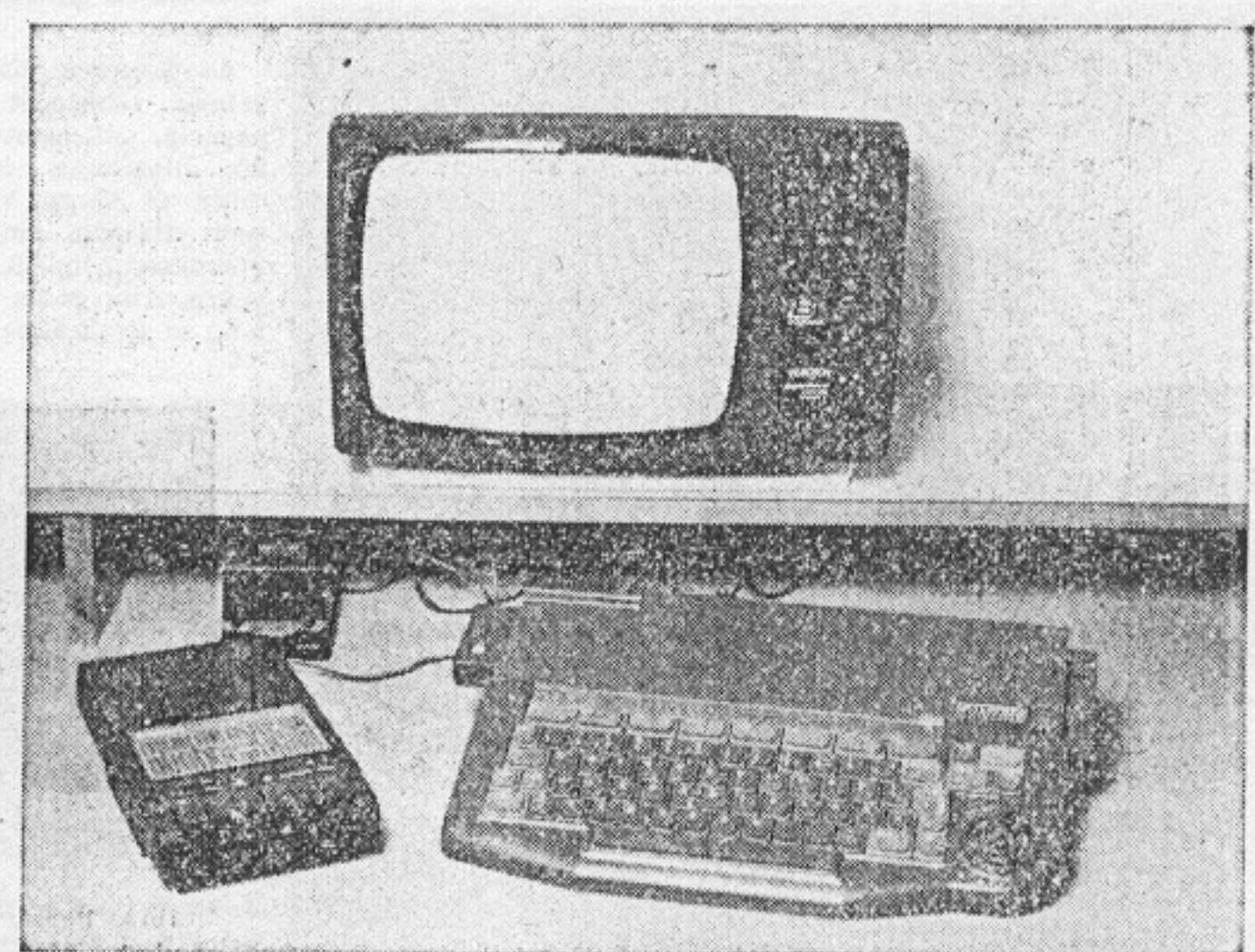
Hát aktor melyiket válasszam? Mint hogy általában nem egy-egy kiragadott cél érdekében akarjuk a számítógépünket alkalmazni, nehéz erre egyéretermi tipust megjelölni. Az itt többször említett tipusok a hazai gyakorlatban a legelterjedtebbek. Vannak jobb típusok korszerűbb gépek, de ezekhez nehéz a márkatakákat fellelni, az üzemeléshez tapasztalatokat, tanácsokat, programokat beszerezni.

Az, hogy melyik a legjobb, megint csak nehezen ítéhető meg, hiszen ezen a szinten a gépek hardverlehetőségei ki-egyenlítősek. A használt program pedig a programkészítő ügyességtől függ.

Dr. Sz. Lukács János



1. ábra: A Videoton TV-Computer RAM-ja 64k-s



2. ábra: A feltörekvő Enterprise

# RAJZI LEHETŐSÉGEK

A személyi számítógépek zöme ma már elfogadható pontosságú grafikus ábrázolást tesz lehetővé. A műszaki életben egy sor olyan görbékkel találkozhatunk, mely a hagyományos euklideszi szerkesztés fogalma szerint nem szerkeszthető meg, csak pontjait tudjuk megállapítani. Ezért meg szoktunk elégdeni egy-egy evolvens (ciklois stb.) görbe megnajzolásával. Pedig az ipari görbék egyenleteit ismert, különböző paramétereik változtatásával érdekes határesetekre („elfajlásokhoz”) juthatunk.

Ezeket az időigényes szerkesztéseket a számítógép igen hamar elvégzi. Végünk sorra egy csokorral valót ezekből az ipari gyakorlatban használt görbeprofilokból, illetve mozgáspályákból.

Nézzük először az evolvenset, amit fogaskerekek fogprofiljaként ismerünk (1. program).

Az archimedesi spirális (az evolvens egy speciális esetként fogható fel) megvalósítható a tokmányopoda mereszítőarcáján, vagy az Abbe-féle hosszmérő nóniúszán (2. program).

A cikloisok különböző változatát állíthatjuk elő attól függően, hogy az a bizonyos kör egyenesen, körön belül vagy kívül gördül-e le, és hogy a vizsgált pontot a legörökült kör sugarán hol figyeljük. Ciklois görbét ír le a kombinált tollalapátja, a marcszerszám fogja. Ciklois alakú lehet a fogaskerek fogprofilja, de egyes finomfelületi megmunkálások során a csiszolószemcse vagy a mun-

kadarab mozgáspályája is ezzel a görbével írható le. Attól függően, hogy a P értékét hogyan válfotítsuk, a képernyőre a program nyújtott, csúcsos ( $R=P$ ) vagy hurkolt ciklost rajzol a futtatás eredményeként (3. program).

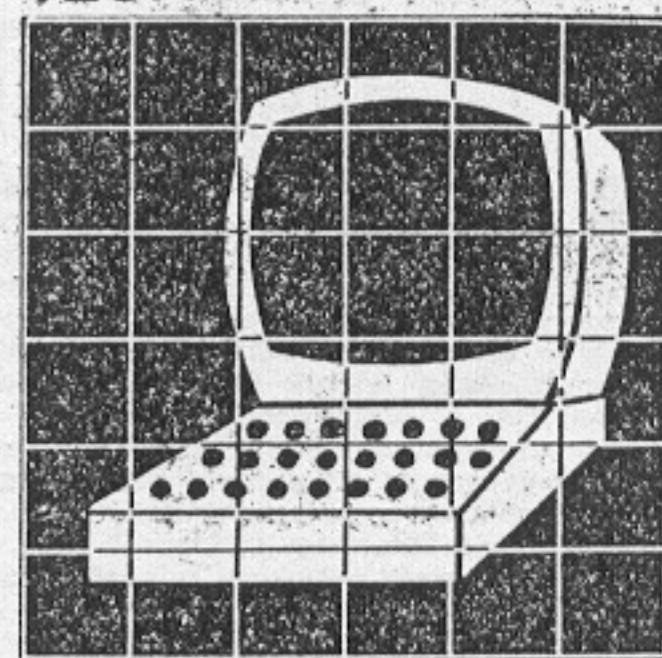
A 4. program különböző epi- és hipociklois megrajzolását biztosítja. A változkoz kedvű programozók ezekből az egyenletekből nemcsak a praktikus bemutatás céljára, hanem akár látványnak is érdekes ábrákat tudnak előállítani.

A különböző ipari görbék mellett egy szép ellipszis megrajzolása is hosszadalmas munkát igényelt. Ezt a felületet egy egyenes körhenger elmetszésével kaphatjuk, ha a metszősfik a henger tengelyével 0 és 90° közötti szöget zár be.

A következő program C+4 számítógép esetén biztosítja, hogy a körhenger sugarának és a metszősfik hajlásszögének beírása után rajzos és táblázatos formában megkapjuk a henger palástjának megrajzolásához szükséges adatokat (5. program). A program a kör kerületét 32 részre osztja, az újabb adatokat minden SPACE lenyomásával kérhetjük a képernyőre.

A praktikus felhasználás után még egy látványos ábra és annak előállítására szolgáló program. A térgörbe megrajzolásakor olyan logikát használunk, hogy előbb a hozzá közelebb lévő vonalakat

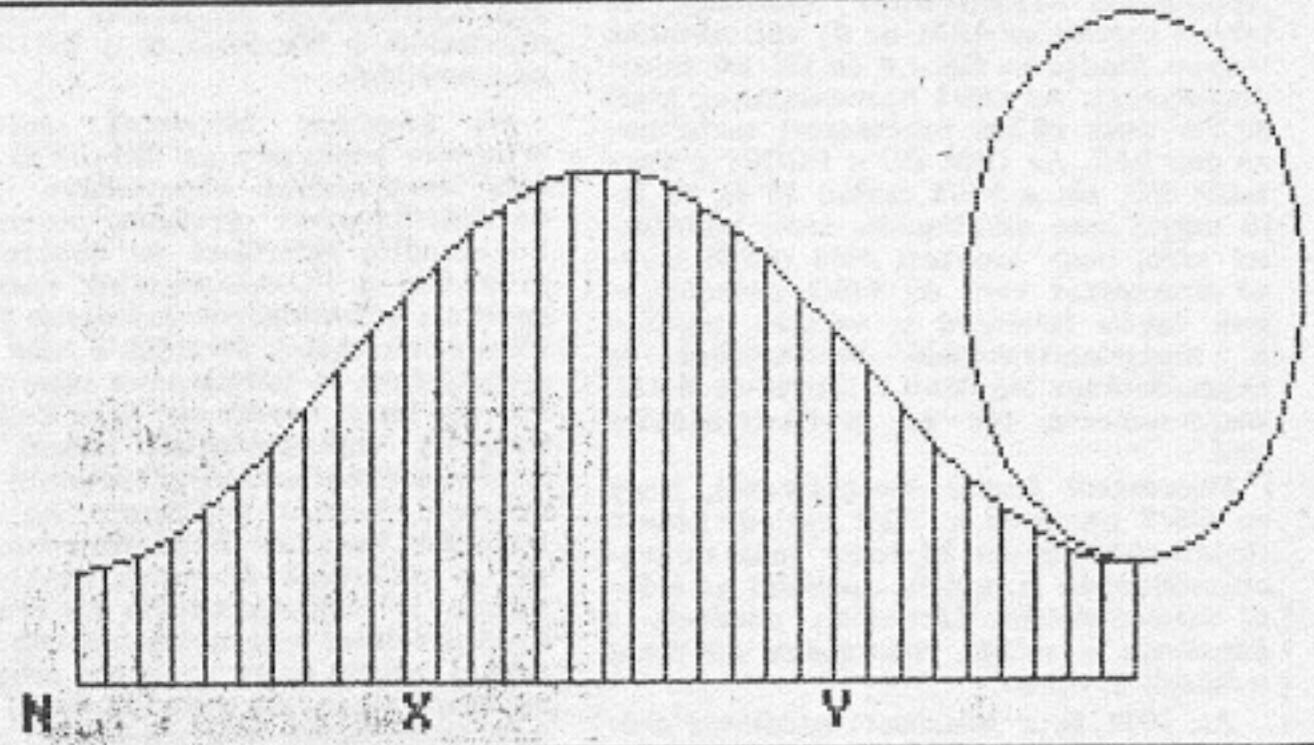
## TECHNIKOLA



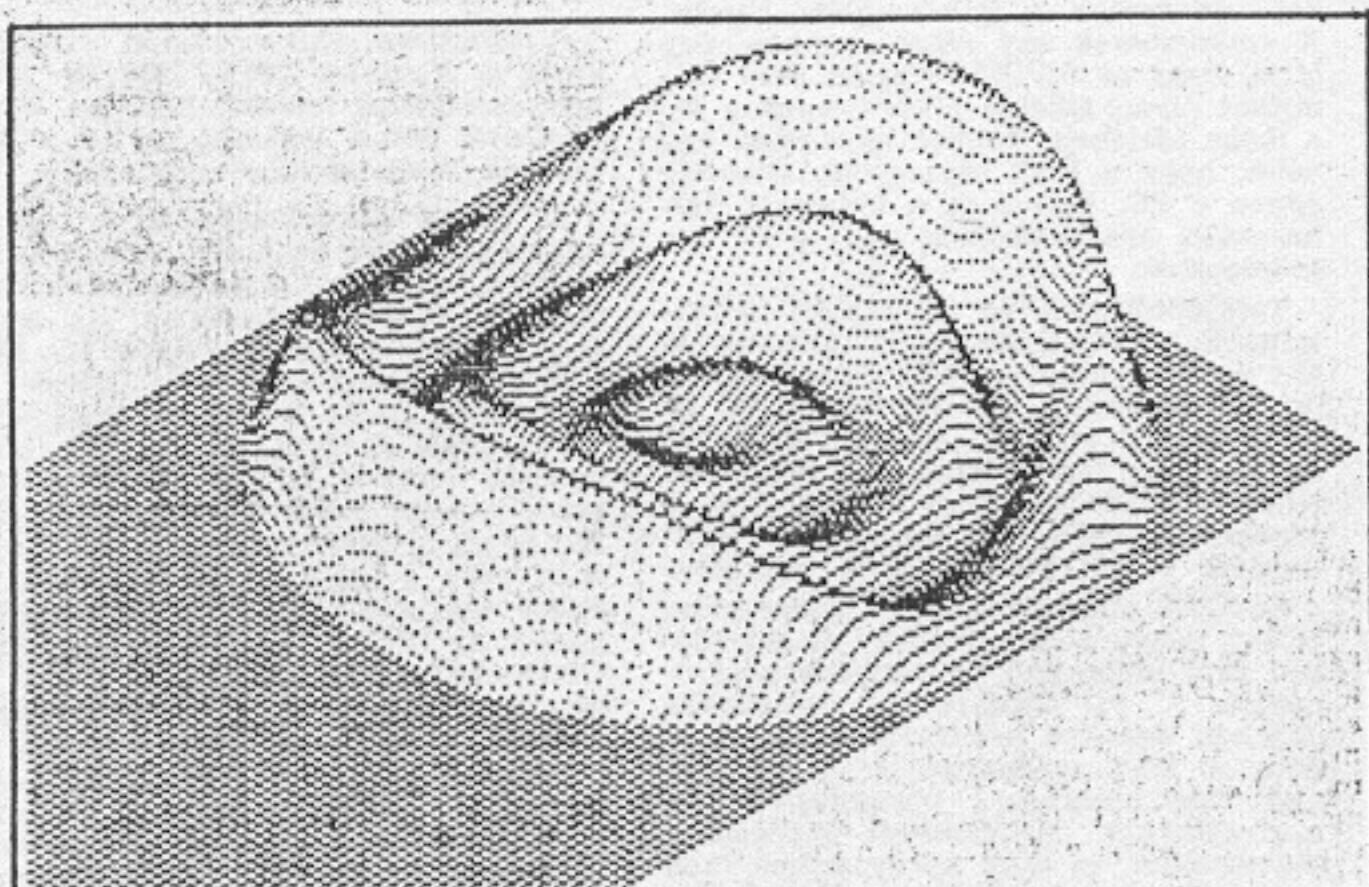
### MC-PROGRAMOK

rajzoljuk meg, és az újabb vonal megrajzolásakor a láthatóság figyelembevételével a takarásba kerülő programokat nem rajzoljuk ki. Ezt az Y tömbbe tárolt érték figyelésével biztosíthatjuk (28-20 sor). Ezt a módszert más hasonló problémá megoldására is fel lehet használni (6. program).

Dr. Sz. Lukács János



1. ábra: Ferdén metszett henger palástja



2. ábra: Kalap ábrázolása a láthatóság figyelembevételével

```
100 REM EVOLVENS
110 GRAPHIC 1,1
111 R=20 : REM A ALAPKOR SUGARA
120 CIRCLE 1,160, 70,A : PAINT 1,160,70
130 FOR F = 0 TO 6.28 STEP 0.1
140 : X=A*(COS(F)+F*SIN(F))
150 : Y=A*(SIN(F)-F*COS(F))
160 : DRAW TO X+160 ,70-Y
170 NEXT
180 GETKEY A$:GRAPHIC 0
1. program
```

```
100 REM ARCHIMEDESI SPIRAL
110 GRAPHIC 1,1
120 FOR A = 1 TO 51 STEP 0.1
130 R=2*A : REM EGYUTTHATÓ VALTOZTATHATÓ
140 IF A=1 THEN DRAW 1,160,100
150 DRAW TO 160+R*COS(A),100+R*SIN(A)
160 NEXT
170 GETKEY A$:GRAPHIC 0
2. program
```

```
1 REM CIKLOISOK NYUJTOTT, CSUCSOS, HURKOLT
2 GRAPHIC 2,1
3 INPUT "GORDOLOKOK" R="";R
4 INPUT "OP TAVOLSAG" T="";P
5 IF R<5 OR R>40 OR P<5 OR P>70 THEN RUN
6 DRAW 0,120 TO 320,120:CIRCLE,R,120-R,R
10 FOR T=-1 TO 7 STEP .1
11 : X=R*T-P*SIN(T)
12 : Y=R -P*COS(T)
13 : DRAW1,X+R,120-Y
14 NEXT
3. program
```

```
100 REM EPI-, HIPO CIKLOIS
110 GRAPHIC 2,1
120 INPUT "ALAPKOR" R1="";P
130 INPUT "GORDOLOKOK" R2="";R
140 INPUT "EP VAGY HIPOCIKLOIS E/H " ;A$
150 IF A$<>"E" AND A$<>"H" THEN GOTO 140
160 CIRCLE1,160,100,P
170 N=R/P : J=0
180 FOR F=0 : TO 6.28/N STEP .1
190 : T=F: IF A$=="E" THEN T=-F
200 : IF J=1 THEN GOTO 220
210 : IF A$=="E" THEN N=-N:R=-R:J=1
220 : Q=N+1:W=N*T: E=R/N
230 : X=E*(Q*COS(W)-N*SIN(Q*T))
240 : Y=E*(Q*SIN(W)-N*COS(Q*T))
250 : DRAW1,X+160,100-Y
260 NEXT
4. program
```

```
100 REM FERDEN METSZETT HENGER PALASTJA
110 REM
120 GRAPHIC 2,1 : L=1 : DIM X(33),Y(33)
130 INPUT "ALAPKOR SUGARA";R
140 INPUT "M. SIK TENG.-EL BEZART SZOGE";A
150 IF A>90 OR A<0 THEN GOTO 140
160 A=A*0.01745
170 K=2*R*3.1415: IF K>300 THEN L=260/K
180 M=2*R/SIN(A): IF M>100 THEN L=100/M
190 CHAR1,0,0,"HENGER PALAST"
200 CHAR1,1,19,"N" X Y
210 FOR V=0 TO 6.28 STEP 0.19625
220 X=K*V/6.28: Z=R-R*COS(V): Y=Z/TAN(R)
230 PRINT J, INT(X), INT(Y)
240 J=J+1: X=L*X: Y=L*Y
250 DRAW1,20 , 150 TO 20+X,150
260 DRAW1,20+X,150 TO 20+X,130-Y
270 X(J)=20+X: Y(J)=130-Y
280 GETKEY A$
290 NEXT: PRINT
300 A=INT(2*R/COS(A))
310 PRINT "A=";A, "B=";2*R
320 DRAW,X(1),Y(1)
330 FOR I=1 TO 33:DRAW TO X(I),Y(I):NEXT
340 CIRCLE1,X(33),Y(33)-L*A/2,L*R,L*R/2
350 END
5. program
```

```
10 REM KALAP + LATHATOSAG C+4
12 DIM Y(320)
14 Q=.77: A=160 : D=190: W=100: F=3.14/0
16 FOR S=1 TO 320 : Y(S)=199 : NEXT
18 GRAPHIC 1,1 : TI$ ="0000000"
20 FOR X=1 TO R+154 : L=D-(1/2)*(X-A)
22 P=SQR(((X-A)/W-Q)^2+(R/W-Q)^2)
24 Y=10*(1-COS(6*F*P))*(P<=.76)
26 Y=L+Y*(1.2+COS(4.078*((X-A)/100)))
28 IF Y>Y(X) THEN 32
30 Y(X)=Y: DRAW 1,X,Y
32 IF X=A+154 THEN GOTO 36
34 NEXT
36 R=R+2, : A=160-R: D=D-1
38 IF A<6 THEN GOTO 42
40 GOTO 20
42 CHAR1,0,0,TI$
```

6. program