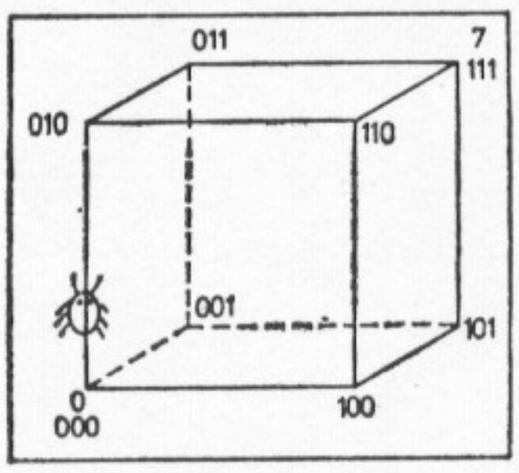
µC Iskolaszámítógép-programok

A VÉLETLEN

A statisztikus fizika, az élet jelenségeinek modellezésére, szimulálására, a játékok változatosabbá tételére a számítógépek nélkülözhetetlen kelléke a véletlenszám-generátor. Az iskolaszámítógép ezzel kapcsolatos utasításai a RANDOM (véletlen) és az RND. Vizsgáljuk az 1. program futtatásával, hogy valóban véletlen számokat állít-e elő számítógépünk.

Ha programunkat egymás után háromszor-négyszer futtatjuk, azt látjuk, hogy a kapott 1 és 100 közötti számokból álló számtízesek különbözők. Megnyugodhatnánk, hogy valóban véletlen számokat kaptunk. Jegyezzük azonban le a kapott számtízeseket és kapcsoljuk ki gépünket 15...20 másodpercre! A gép bekapcsolása után írjuk be újra a fenti programot és futtassuk újra háromszor-négyszer! Meglepődve látjuk, hogy ugyanazokat a számtízeseket kapjuk, mint az előbb. Pseudo-random — ál-véletlenszámoknak nevezik a számítógépek által így előállított



1. ábra: Kocka élein véletlenszerűen bolyongó bogár

számokat, mert azok a legnagyobb számítógépekben is bizonyos törvények szerint képződnek, sőt bizonyos lépésszám után (10^{10}...10^{15}) ismétlődnek is. Az említett céloknak azonban nagyon is jól megfelelnek.

Hogy mennyire jó a véletlenszám-generátorunk, azon akkor lepődünk meg, ha pl. a kockadobálás szimulálásával ellenőrizzük. A 2. program futtatásakor a gép kérésére 6-ot, 60-at, majd 600-at beadva látjuk, hogy főleg az utolsó esetben már csak néhány százalékos az eltérés a kapott oldalszámok között.

Természetesen azt nem is várhatjuk, hogy eltérés ne legyen, hisz véletlenről van szó; szerencsejátékról. Amennyiben az R = RND(6) utasítás helyébe az R = INT(2xRND(0) vagy az R = INT (RND(0) + 0,5-et írunk és a T(0) és (T) tárolók tartalmát íratjuk ki, akkor a pénzfeldobás szimulálásával vizsgáltuk iskolaszámítógépünk véletlenszám-generátorának a jóságát. Az eredmény így is megnyugtató.

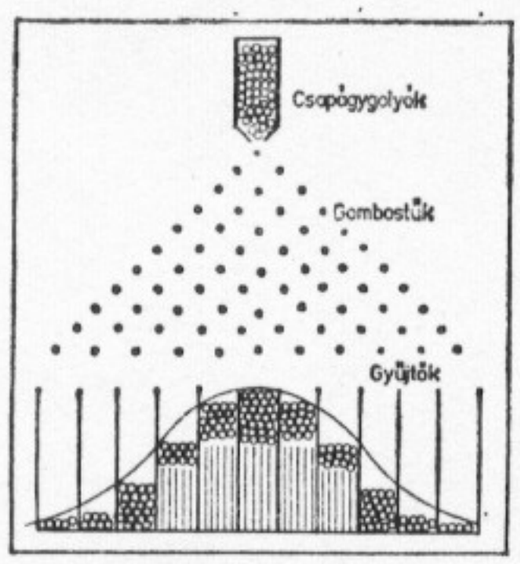
```
10 FOR I=1 TO 10
20 PRINT RND (100)
30 NEXT I
```

1. program

```
10 '   KOCKADOBÁLÁS

15 CLS
16 DIM T(100)
20 INPUT "FANYSZOR DOBJAK";N
30 FOR I=1 TO N: T(I)=0: NEXT I
40 FOR I=1 TO N
50 R=RND(6)
60 T(R)=T(R)+1
70 NEXT
80 FOR I=1 TO 6
90 PRINT T(I)
100 NEXT
110 GOTO 20
```

2. program



2. ábra: Kísérlet Galton-deszkával

Ha 0 és 1 közötti (tizedestört) véletlen számot kérünk az iskolaszámítógéptől, akkor az RND-utasításba argumentumnak 0-t kell beírnunk. Más számítógépek esetén ez a 0 esetleg el is hagyható. Vannak számítógépek, amelyek csak ilyen 0 és 1 közötti véletlen számot tudnak generálni. Ezekkel pl. 1 és 6 közötti véletlenszámokat így állítunk elő: INT(6xRND) +1. Ha pedig iskolaszámítógépünkkel olyan véletlen egész számokat akarunk generálni, amelyek között a 0 is szerepelhet, akkor pl. így kérjük a véletlen számot: X = INT(128)—1 vagy Y = RND(48)—1.

Nézzünk néhány programot a megbeszéltek felhasználására.

Engedjünk szabadon a levegőben illatszert, pl. a képernyő közepén 20...30 molekulát. Ezek a levegőmolekulákkal és egymással véletlenszerűen ütközve terjednek, azaz a levegővel diffundálnak. Vizsgáljuk meg, hogy milyen messzire jutnak el egy bizonyos idő alatt vagy pl. mikor ér el az első a képernyő határáig.

A matematikusok ezt a problémát a síkon való bolyongásnak nevezik igen szemléletesen. A gázok természetesen a térben terjednek, de a kapott kép jó modellje a valóságnak (3. program).

Nehezebb feladat a következő. Egy kocka élein bolyong véletlenszerűen egy bogár (1. ábra). Kérdés, hogy átlagosan hány élt jár be, amíg az egyik csúcsból eljut a szemközti csúcsba, pl. a 0 jelzésűből a 7. jelzésűbe. Próbáljuk magunktól megoldani a feladatot. Bizony izzasztó, mert vagy magasabb matematika (gráfos felrajzolás és valószínűségszámítás), vagy valami ügyes ötlet kell hozzá. Válasszuk A. Engel: Computerorientierte Mathematik c. könyve nyomán a csúcsok sorrendjét az ábrán megadottnak, sorszámukat pedig kettes számrendszerben írjuk fel. Ekkor ugyanis az egyes csúcsok sorszáma bármelyik szomszédos csúcs sorszámától csak egyetlen számjegyben különbözik, tehát egyetlen véletlen szám „dobásával" elérhető. A programban a következő jelöléseket használjuk (4. program): N jelenti az összes túrák számát, M az egy túrán belül várható maximális lépésszámot. Ez utóbbi a tapasztalat szerint a 100-at rendkívül ritkán haladja meg. A lépések teljes számát az S tárolóban való összegzéssel kapjuk. A legkisebb lépésszám 3 és a lépések száma egy túrán belül csak páratlan lehet. A program végén csak azon tárolóknak a sorszámát (I) és tartalmát íratjuk ki, amelyek valóban szerepeltek. És természetesen kiíratjuk a keresett átlagot is: S/N.

Igen tanulságos a fenti szép feladatot kézzel történő kockadobással is végigjátszani néhányszor. A kocka lapjait ekkor természetesen pl. így kell számoznunk: 1 és 2 — I; 3 és 4 — II.; 5 és 6 — III.

Az RND-utasítás játékokban való alkalmazására találunk példát az iskolaszámítógéphez mellékelt kazetta játékai között is (pl. Céllövés).

A radioaktív bomlás modellezése tipikus feladat a véletlen számok felhasználásában. Sokféle megoldása van. Közös vonásuk, hogy a radioaktív anyagra jellemző bomlási állandót, amely 1-nél kisebb szám, az atom elbomlási valószínűségének tekintjük. Minden atom fölött „sorsot vetünk". Ha a kapott véletlen szám nem haladja meg a bomlási állandót, akkor az atom elbomlott. Természetesen a bomlási állandót a választott időtartammal (DT) is meg kell még szoroznunk. Az 5. program a megmaradt atomok számát ábrázolja az idő függvényében.

Komoly programozói munka eredménye a Gauss-féle, normális eloszlást bemutató 6. programunk. Ezt egy második gimnazista diák készítette. A diákok a feladathoz a kísérleti modellt (2. ábra) is megépítették, ezt Galton-deszkának hívják. A sima deszkába bevert gombostűk elrendezése az ábra szerinti. A ferde deszkán fentről egy pontból apró csapágygolyókat indítunk. Ezek a gombostűkön való véletlen ütközések után az alsó gyűjtőrekeszekbe hullanak. Sok golyó esetén felületük szép „haranggörbét" alkot. Számítógépes szimulálása sem kisebb munka, mint kísérleti megvalósítás. Ha a program megvan, akkor természetesen már nagyon egyszerű.

Kovács Mihály

•••

Figyelem! Számítógép-programozási feladat található a keresztrejtvény mellett, a 24. oldalon.

```
10 ' TELJES TÚRA EGY KOCKA ÉLEIN

20 READ S,N,M: DIM L(M)
30 FOR I=3 TO M
40     L(I)=0
50 NEXT I
60 FOR K=1 TO N
70     Y(1)=0:Y(2)=0:Y(3)=0:I=0
80     D=INT(3*RND(0))+1
90     Y(D)=1-Y(D): I=I+1
100    IF  Y(1)+Y(2)+Y(3)<3 THEN 80
110    L(I)=L(I)+1: S=S+I
130 NEXT K
140 FOR I=3 TO M STEP 2
150    IF L(I)=0 THEN 170
160    PRINT I, L(I)
170 NEXT I
180 PRINT: PRINT S/N
190 DATA 0,100,99
```

4. program

```
2 '              Molekulák diffundálása

4 ' 30 MOLEKULÁT ENGEDÜNK SZÉT A KÉPERNYŐ KÖZEPÉN.
      MENNYIRE DIFFUNDÁLNAK SZÉT T IDŐ ALATT?
5 CLS
10 DIM X(50),Y(50)
20 FOR I=0 TO 30
30 X(I)=64: Y(I)=24
40 NEXT I
50 SET(64,24)
60 T=0
65 ON ERROR GOTO 160
70 FOR I=0 TO 30
80 RESET(X(I),Y(I))
90 X(I)=X(I)+(2*INT(2*RND(0))-1)*2
100 Y(I)=Y(I)+2*INT(2*RND(0))-1
110 SET(X(I),Y(I))
120 NEXT I
130 T=T+1
140 PRINT@10,"T= ";T
150 GOTO 70
160 PRINT@ 962,"A LEGFRISSEBB MOLEKULA ";SQR((X(I)-64)*(X(I)-64)+
        (Y(I)-24)*Y(I)-24));" EGYSEGNYIRE JUTOTT EL"
170 END
```

3. program

```
5 '          RADIOAKTIV BOMLÁS
8 CLS
10 INPUT"ATOMOK SZÁMA, BOMLÁSI ÁLLANDÓ,IDŐTARTAM,LÉPÉS;
      PL.:100;.02,100,1";N,L,T,DT
20 N1=N:N2=N:PRINT:PRINT"IDO      ATOMOK SZAMA     PONTOS ERTEK"
30 PRINT@ 385,"ATOMOK SZAMA":PRINT@ 1019,"IDO";: FOR E=17 TO 47:
  SET(0,E): SET(1,E): NEXT: FOR E=0 TO 115: SET(E,47):NEXT
40 FOR K=1 TO T/DT
50 FOR Y=1 TO N1
60 R=RND(0)
70 IF R<=L*DT THEN N2=N2-1
80 NEXT Y
90 IF K/10=INT(K/10) THEN PRINT@190,K*DT;"       ";N2;"          ";N*
EXP(-L*K*DT);: SET(100*K*DT/T, 47-N2*30/N)
100 N1=N2
110 IF N2<=N/2 AND Z=0 THEN PRINT@280, "A FELEZTESI IDO:"; K*DT;
  "SEC": Z=1
120 NEXT K
130 GOTO 130
```

5. program

```
5 '        G A L T O N - D E S Z K A
6 '                                  SZEBEDY VAJK :11.o.
7 '
10 CLS:DIM T(11):RANDOM
20 A=68:B=56:KT=6
30 FORY=9TO36STEP3
40 A=A-6:B=B+6
50 FORX=ATOBSTEP12
60 SET(X,Y):SET(X+1,Y)
70 NEXTX,Y
80 FORA=0TO59:READB:NEXT
90 POKE15391,176
100 READB,C,D,E,F
110 FORA=BTOCSTEPD
120 POKEA,E
130 NEXT
140 ONFGOTO100,150
150 FORA=0TO100:NEXT:POKE15391,128
160 RESTORE
170 FORA=15452TO15458
180 READB
190 POKEA,B
200 NEXT
210 READC,D,E
220 POKE15519,C
230 FORA=0TOD:NEXT
240 ONEGOTO170,245
245 X=62:Y=8:K=0:P=0
250 A=RND(2)
260 IFA=1THENK=K+0.5ELSEK=K-0.5
270 S=0:H=4
280 IFY=38THENS=0:H=3:POKE15391,176
290 S=S+1:IFS=HONHGOTO0,0,390,250,0,410
300 Y=Y+1:XT=X
310 ONAGOTO320,330
320 X=X+2:GOTO340
330 X=X-2
340 IFX<0GOTO360
350 SET(X,Y):SET(X+1,Y)
360 IFXT<0GOTO380
370 RESET(XT,Y-1):RESET(XT+1,Y-1)
380 GOTO280
390 IFA=1THENA=2ELSEA=1
400 S=0:H=6:GOTO290
410 I=KT+K:T(I)=T(I)+1
415 X=100+2*I:Y=23-T(I)
420 IFY<0GOTO430
425 SET(X,Y):SET(X+1,Y)
430 IFT(I)>99THENP=1ELSEP=0
440 PRINT@890+I*6-P,T(I);
450 GOTO150
460 DATA131,140,176,131,176,140,131,128,100,1,191,128,128,140,128,
128,191,128,50,1,191,128,128,176,128,128,191,128,25,1,191,128,128,
128,128,128,191,131,12,1,131,140,176,128,176,140,131,140,6,1,131,1
31,131,131,131,131,131,131,176,3,2
470 DATA15452,15458,1,131,1,16196,16250,6,188,1,16260,16314,6,191,
1,16320,16382,1,191,1,15858,15870,1,176,2
```

6. program

Iskolaszámítógép-programok

Az előző szám 6. programfeladatának egy másodikos gimnazista elkészítette egy másik változatát. Ez a mostani 1. programunk.

A számítógépes képernyőre rajzolás témakörhöz még azt jegyezzük még, hogy művészi rajzok, képek számítógépes reprodukálása nem tartozik a lélekemelő programozási feladatok közé. Általában sok adat kell hozzájuk, mint azt a 2. program is mutatja.

Előző számunkban a véletlen témakörével foglalkozva 5. programunk a radioaktív bomlást tárgyalta. Ugyanezt a problémát tárgyalja most 3. programunk, két generáció esetére. **Kovács Mihály**

```
0                    SZÖGES TÁBLA
                                        ZSEMBERY A. II.o.
10 CLS:PRINTCHR$(23):PRINT@456,"S Z O G E S  .T A B L A "
20 IFINKEY$=""THEN20ELSECLS
40 CLEAR200:B= 0:RANDOM: DIMT(10),P(10),X(10),Y(10)
45 INPUT"HANY GOLYOT ERESZEK LE";G
50 FORW=0TO10:READP(W):NEXT
60 FORW=0TO10:READX(W):Y(W)=47:NEXT
70 S=20/(G*0.3):IFS>1THENS=1
100 CLS:FORW=1TO3:READZ:X$=X$+CHR$(Z):NEXT:FORW=1TO3:READZ:Y$=Y$+C
HR$(Z): NEXT:FORW=1TO3:READZ:Z$=Z$+CHR$(Z):NEXT:FOFW=1TO3:READZ:V$
=V$+CHR$(Z):NEXT
110 PRINT@29,CHR$(131);CHR$(175);CHR$(139);CHR$(171);CHR$(135);CHR
$(129):PRINT@95,CHR$(161);CHR$(129)
120 Z=155:T$=X$+Y$+X$:PRINT@Z,T$:T$="":Z=213
130 T$=Z$:FORW=1TO3:T$=T$+V$+Z$:NEXT:GOSUB1000
140 T$=X$:FORW=1TO4:T$=T$+Y$+X$:NEXT:GOSUB1000
150 T$=Z$:FORW=1TO6:T$=T$+V$+Z$:NEXT:GOSUB1000
160 T$=X$:FORW=1TO7:T$=T$+Y$+X$:NEXT:GOSUB1000
170 T$=Z$:FORW=1TO9:T$=T$+V$+Z$:NEXT:GOSUB1000
180 T$=X$:FORW=1TO10:T$=T$+Y$+X$:NEXT:GOSUB1000
190 FORW=0TO127:SET(W,47):NEXT
210 FORI=1TOG:PRINT@0,I-1;:PRINT@56,G;
220 X=63:Y=2:SET(X,Y):FORW=0TO  B:NEXT:Y=3:RESET(63,2):SET(63,Y):F
ORW=0TO  B:NEXT:Y=4:RESET(63,3):SET(63,Y):FORW=0TO  B:NEXT:RESET(6
3,4)
250 V=RND(2):ONVGOTO260,280
260 FORH=1TO2:X=X-3:Y=Y+1:RESET(X+3,Y-1):IFY= 24 THEN 300
270 SET(X,Y):FORW=0TO B:NEXTW,H:GOTO250
280 FORE=1TO2:X=X+3:Y=Y+1:RESET(X-3,Y-1):IFY= 24 THEN 300
290 GOTO270
300 FORJ=0TO10:SET(X,Y):FORW=0TO10:NEXT:RESET(X,Y):NEXT:X=(X-3)/12
:T(X)=T(X)+1:PRINT@P(X),T(X);:Y(X)=Y(X)-S:SET(X(X),Y(X)):SET(X(X)+
1,Y(X)):NEXTI
999 PRINT@1018,:END
1000 PRINT@Z,T$:T$="":READZ:RETURN
1900 DATA448,390,268,210,150,35,165,235,304,439,509
1902 DATA2,14,26,38,50,62,74,86,98,110,122
1905 DATA160,152,176,130,129,131,128,160,128,136,134,140
1910 DATA274,332,393,451,512,0
```

1. program

```
10 '              MACKÓFEJ RAJZOLÁSA
11 '                                  DRASKOVITS IMRE II.O.
12 CLS:FOR X=0 TO 63 STEP 2
13 SET(X,0):NEXT
14 FOR Y=0 TO 47 STEP 2
15 SET(63,Y):NEXT
20 FOR X=9 TO 53
30 DATA 16,14,12,11,10,9,8,8,7,7,7,7,8,8,9,10,12,12,12,12,11,11,10
,10,9,9,8,8,8,9,9,10,11,12,14,23,22,22,22,20,20,20,20,22
35 '************************************************************
40 READ Y:SET(X,Y):NEXT
50 FOR X=9 TO 52
55 '************************************
60 DATA 17,15,13,12,16,15,14,13,12,11,11,11,11,12,9;12,11,18,18,19
,35,12,13,13,13,12,12,11,11,12,12,12,13,21,13,15,34,33,32,31,28,26
,26,25
65 '******************************************
70 READ Y:SET(X,Y):NEXT
80 FOR X=9 TO 44
90 DATA 18,25,27,29,30,30,31,32,36,37,38,39,39,40,40,40,41,41,41,4
0,40,40,40,39,38,38,38,38,38,38,38,38,37,36,36,35
100 READ Y:SET(X,Y):NEXTX
110 FOR X=9 TO 44
120 '****************************************
130 DATA 9,26,28,18,17,23,24,25,18,17,16,15,14,13,13,19,12,21,23,3
5,37,37,37,42,37,35,34,32,31,34,32,31,34,27,25,25,
140 '************************************************************
150 READY:SET(X,Y):NEXT
160 FOR X=31 TO 44
170 DATA 22,37,36,26,27,24,35,33,39,37,23,26,23,24
180 READY:SET(X,Y):NEXT
190 FOR X=32 TO 44
200 DATA 38,26,24,33,22,41,24,42,36,20,23,20,23
210 READY:SET(X,Y):NEXT
220 FOR X=32 TO43
230 DATA 41,24,21,25,21,40,17,41,35,19,22,19
240 READY:SET(X,Y):NEXT
250 FOR Y=12 TO 25
260 DATA 37,34,41,41,41,41,25,12,9,9,9,9,9,24
270 READX:SET(X,Y):NEXT
280 FOR Y=14 TO 25
290 DATA 34,36,37,43,39,40,13,13,13,23,23,49
300 READX:SET(X,Y):NEXT
310 FOR Y=18 TO 25
320 DATA 34,16,16,23,23,28,37,25
330 READX:SET(X,Y):NEXT
340 FOR Y=18 TO 24
350 DATA 35,33,24,28,28,32,26
360 READX:SET(X,Y):NEXT
370 FOR Y=18 TO 24
380 DATA 41,36,28,32,51,35,49
390 READX:SET(X,Y):NEXT
400 FOR Y=33 TO 37
410 DATA 16,16,16,37,17
420 READX:SET(X,Y):NEXT
430 DATA 16,9,27,35,27,36,28,36,32,20,32,40,35,14,36,20,37,37,37,3
9,39,23,39,40,40,23,43,16,43,18,44,16,48,23,48,29,48,30,49,21,49,2
7,52,21,52,23,52,24,0,0
440 READX,Y
450 IFX=0AND Y=0,GOTO471
460 SET(X,Y)
470 GOTO 440
471 RESET(44,9):RESET(53,17):RESET(52,18)
480 GOTO 480
```

2. program

```
5 '     RADIOAKTIV BOMLÁS  KÉT GENERÁCIÓ ESETÉN
10 CLS:INPUT"BOMLASI ALLANDOK PL.:  0.02;  0.03";LA,LB:PRINT
15 PRINT"   IDO  ";CHR$(34);"A";CHR$(34);" ATOMOK SZAMA   ",
17 PRINTCHR$(34);"B";CHR$(34);" ATOMOK SZAMA   ELMELETI ERTEKEK":P
RINT
20 DATA 100,0,1,100
30 READ NA,NB,DT,T
40 MA=NA:SA=NA:MB=NB:SB=NB:FORK=1TOT/DT
50 FORI=1TOMB:R=RND(0)
60 IFR<=LB*DTTHENSB=SB-1
70 NEXTI
80 FORI=1TOMA
90 R=RND(0)
100 IFR<=LA*DTTHENSA=SA-1:SB=SB+1
110 NEXTI
120 IFK/10<>INT(K/10)THENGOTO150
130 PRINT" ";K*DT,SA;  ,SB;"        ";NA*EXP(-LA*K*DT);
140 PRINT" ";NA*LA/(LB-LA)*(EXP(-LA*K*DT)-EXP(-LB*K*DT))
150 MA=SA:MB=SB
160 NEXTK
```

3. program

ABC 80-program: pozitív egész számok primtényezős felbontása

A pozitív egész számok primtényezős felbontása része a középiskolák első osztályai matematika anyagának. Az itt közölt program segítségével is ezt a feladatot oldhatjuk meg.

Az alkalmazott algoritmus eltér az iskolákban tanítottól. Mivel a számítógép a primszámok ismeretére csak külön programrész beiktatása révén válna képessé, a primosztók vizsgálata helyett csak osztókat keresünk. Ezzel nyilvánvalóan megnöveljük a vizsgálati lépések számát, de a számítógépünk viszonylag nagy műveleti sebessége miatt ez nem okoz problémát, így válik a program egyszerűbbé, kezdők számára is elkészíthetővé.

Az 1. számmal külön kell foglalkozni, hiszen az sem nem prim és sem nem összetett (70. sor). A kiiratásnál a megszokott hatványtényezős alak helyett az elsőfokú tényezők szorzatát alkalmaztuk. Ennek oka az, hogy áttekinthetőbb. Védjük a programunkat az elgépelés (ONERRORGOTO), valamint a nem megfelelő számok beadása (80-90. sor) ellen is.

A program felhasználását a számítástechnika tanításában elsősorban didaktikai szempontok indokolják. Jól szemléltethető vele, hogy az adott feladat megoldására az általában rendelkezésre álló több algoritmus közül azt választjuk, amely a céljainknak legjobban megfelel.

Tarcsai Tamás
gyak. gimn. tanár,
Szeged

```
10 DIM V(100)
20 ? CHR$(12);" POZITÍV EGÉSZ SZÁMOK PRIM-";: ? "   TÉNYEZŐS FELBONTÁSA"
30 ? "**********************************"
40 FOR I=1 TO 1500 : NEXT I
50 ONERRORGOTO 50 : ? "KÉREK EGY POZITÍV EGÉSZ SZÁMOT!!!"; : INPUT A
60 ONERRORGOTO 0
70 IF A=1 THEN ? "1" : GOTO 210
80 IF A<=0 THEN 100
90 IF A=INT(A) THEN 110
100 ? "NEM ILYEN SZÁMOT KÉRTEM." : GOTO 50
110 X=A : Q=0 : FOR I=1 TO 100 : V(I)=0 : NEXT I
120 B=INT(A/2)+1
130 B=B-1 : C=A/B
140 IF INT(C)<>C THEN 130
150 Q=Q+1 : V(Q)=C
160 IF B<>1 THEN A=B : GOTO 120
170 IF V(1)=X THEN ? X;" PRIMSZÁM." : GOTO 210
180 ? X;"=";
190 FOR I=1 TO Q-1
200 ? V(I);"*"; : NEXT I : ? V(Q)
210 ? "FOLYTASSUK?" : GET V$
220 IF V$="I" THEN 50
230 ? "VISZONTLÁTÁSRA!" : STOP : END
```

MIKROSZÁMÍTÓGÉP-PROGRAMOZÁSI FELADAT

Írjon ki az iskolaszámítógép képernyőjére a 26 nagybetűből véletlenszerűen egyet. Mi — amilyen gyorsan csak tudjuk — lenyomjuk a megfelelő betű billentyűjét. A gép ekkor írja ki a betű kiírása és a megfelelő billentyű lenyomása között eltelt időt (a befutott legrövidebb ciklusok számát). Készítsük el a játék programját.

A januári számunkban közölt programozási feladat egy lehetséges megoldása:

```
5 REM       "P E P I T A"
10 CLS
20 FOR Y=0 TO 23 STEP 4
30     FOR X=0 TO 31 STEP 4
40 GOSUB 110
50 NEXT X: NEXT Y
60 FOR Y=24 TO 47 STEP 4
70     FOR X=32 TO 63 STEP 4
80 GOSUB 110
90 NEXT X: NEXT Y
100 GOTO 100
110     SET(X,Y): SET(X,Y+1)
120     SET(X+1,Y): SET(X+1,Y+1)
130     SET(X+2,Y+2): SET(X+2,Y+3)
140     SET(X+3,Y+2): SET(X+3,Y+3)
150 RETURN
```

µC Iskolaszámítógép-programok

Előző közleményeinkben már elég sok, viszonylag egyszerű programot mutattunk be. Most azt szeretnénk bemutatni, hogy az ilyen — és az élet által kívánt nehezebb — programokat hogyan kell módszeresen megtervezni, majd megírni. Válasszuk mintapéldának a már szerepelt, de a gyakorlatban nagyon fontos „rendezési feladatot", hisz a szakemberek véleménye szerint a hasznos gépidő több mint egynegyed része rendezésre fordítódik.

Hogyan készül a program?

Az **első teendőnk** a feladat szabatos megfogalmazása, mégpedig — kezdők számára — a legegyszerűbb alakban. A jelen esetben pl. így: az A és a B változók egy-egy számot jelentenek. Rendezzük őket úgy, hogy a B legyen a nagyobb szám, és utána nyomtassuk is ki őket növekvő sorrendben.

Az alábbi kis program nyilván helyesen megoldja ezt az egyszerű feladatot:

```
10 INPUT A,B
20 IF A<=B THEN 40
30 S=A: A=B: B=S
40 PRINT A,B
```

A kezdő számára egyáltalán nem magától értetődő, hogy ha csere szükséges (lásd a 30-as sort!), akkor egy segédváltozó (S) bevezetésére is szükségünk van.

Ha három, négy vagy több szám esetén is a fenti módon akarunk eljárni, könnyen belebonyolódunk a dologba. (Próbáljuk meg!) Ilyenkor fontos a **második teendő**: a feladat alapos elemzése. Pl. a jelen esetben próbáljuk papíron rendezni az első öt számot. Többféle módszer is van rá. Mi maradjunk a cserélgetős vagy buborékoltató módszer (Bubble-Sort) mellett, vagyis két-két szomszédos számot megvizsgálva cseréljük fel őket ha szükséges, ha pedig nem, akkor menjünk tovább.

Válasszuk először a legrosszabb esetet, amikor az öt szám éppen fordított sorrendben van megadva:

```
5 4 3 2 1     c c c c
4 3 2 1 5     c c c m
3 2 1 4 5     c c m m
2 1 3 4 5     c m m m
1 2 3 4 5     m m m m
```

c: cserére volt szükség
m: maradt az eredeti sorrend

Látjuk, hogy az eljárás honnan kapta a nevét: a soron végighaladva az éppen legnagyobb szám a helyére, „legfelülre" ment, mint a buborék a folyadékban.

Tapasztalatok:

1. N elem esetén egy sorban N—1 vizsgálatra van csak szükség,

2. a második, harmadik stb. sorokban mindig egy-egy vizsgálattal kevesebb is elegendő, mert a többi elem már sorrendben van,

3. elég N—1 sorozatvizsgálatot végezni, mert az utolsó elem már így is a helyére kerül. Ezeket a tapasztalatokat figyelembe véve látjuk, hogy a fenti 4×5 vizsgálat helyett éppen a fele is elegendő.

Vegyünk most egy átlagos esetet, hátha újabb lépéseket takaríthatunk meg.

```
Volt: 2 3 1 4 5     m c m m
      2 1 3 4 5     c m m m
      1 2 3 4 5     m m-m m (lett)
```

Végül a legjobb eset:

```
Volt: 1 2 3 4 5     m m m m (maradt)
```

A 4. tapasztalat: ha valahogyan észlelni tudná a gép, hogy egy sor vizsgálata alkalmával már nem volt szükség cserére, akkor abba is hagyhatja a vizsgálatot.

Az alapos feladatelemzésnek köszönhetjük gép futási idejének lényeges rövidülését. Első próbálkozásra N adat esetén N×(N—1) vizsgálatot végeztünk volna. Látjuk, hogy a legrosszabb esetben is elég ennek a fele, a legjobb esetben pedig csak N—1 kell. Sok adat esetén ez jelentős nyereség.

Az alábbi rendező programunkban a tulajdonképpeni rendezés a 90—170. sorokban történik. A 3. tapasztalatunkat a 90-es, a másodikat pedig a 110-es sorban használtuk fel. A 4. tapasztalat reményét a 100-as és 140-es sorokkal valósítottuk meg. Az egyes sorok vizsgálata előtt „felhúztuk" a Z zászlót (Z=1). Ha egy sorban volt csere, akkor „levontuk" a zászlót: Z=0. Így a gép el tudja dönteni, hogy van-e további vizsgálatra szükség vagy nincs.

Az adatok bekérése a 20—70-es sorokkal történt, a kívánt sorrendbe rendezett adatok kinyomtatása pedig a 180—190-es sorokkal.

A kezdő programozó azt gondolhatná, hogy ezzel kész is a program, hisz fut is néhány esetre. Pedig még ezután jönnek a finomságok.

Mi ABC-rendbe akartunk rakatni szavakat vagy neveket. Ezért volt szükség a 20-as sor végén az A és az S változók Stringként történő definiálására. Természetesen tudjuk, hogy a gép a kódszámaik szerint rendezi a betűket, mert azokat ABC-rendben rövekvő kódszámokkal definiálták. Ha sok nevet akarunk rendeztetni, akkor gépünk hamarosan jelzi, hogy kifogyott a Stringek számára fenntartott helyekből (OS = Out of String space). Ekkor a program elejére pl. 5-ös sornak be kell vennünk a CLEAR n utasítást, ahol n megfelelő, 50-nél nagyobb szám. Ha számokat akarunk rendeztetni programunkkal, akkor a DEFSTR utasítást el kell hagynunk, mert különben a gép a számokat számjegyként kezeli. Ekkor a CLEAR-utasításra sincs szükség.

Nem is említettük még, pedig a kezdőnek az sem természetes, hogy a változókat vektorba helyeztük el, hisz különben sok adat esetén kifogynánk a lehetőségekből, és a kezelésük is bonyolultabb lenne. A vektor méretét a 40-es sor automatikusan biztosítja.

A jó programnak áttekinthetőnek kell lennie. Ezt segíti elő, ha egy sorban nincs sok utasítás. Két, három, főleg logikailag összetartozó utasítást azonban bátran tehetünk egy sorba. Ilyenek pl. a 130. vagy a 190. sorok. Segíti az áttekinthetőséget, ha a „beágyazott" ciklusokat valamivel beljebb írjuk (110—150. sorok).

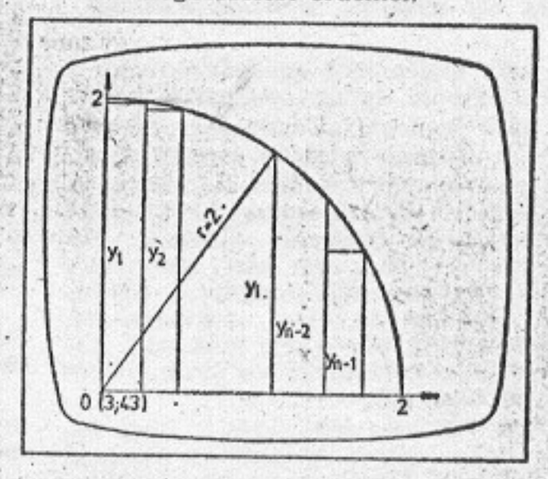
Ha programunk olyanok számára készült, akik a programozásban járatlanok, akkor jó, ha a hibás használat ellen is védjük. Ilyenkor a program futása hibás adatok beadása esetén sem áll le. Pl. ha a 30-as sor beadási kérésére a felhasználó tört számmal felelne, akkor az FC, hibajelzéses leállást pl. a következő utasítással lehet megelőzni:

```
35 IF INT(N)><N THEN 30
```

Ekkor gépünk leállás helyett újra kéri a beadást. A hibás használat elleni védelem nagy körültekintést kíván, de sokszor elhagyhatjuk, mert feltételezhető a program helyes használata.

Kár lenne azonban takarékoskodnunk abban, hogy programunk „barátságos" legyen. Ez röviden azt jelenti, hogy a programozni tudók számára könnyen érthető legyen. Adjunk mindig nevet a programnak és ezt a legelején megjegyzésként vagy kinyomtatandónak írjuk be a programba. Az INPUT-oknál adjuk meg a beadandó adat nevét, ha kell, mértékegységét és az adat értelmes alsó és felső határát. Tagoljuk a programot azáltal, hogy az egyes részek szerepét rövid megjegyzésként a programrészek elejére írjuk.

Második mintapéldának vegyünk egy klasszikus feladatot: a pí közelítő értékének kiszámítását. Mind a matematikai alapgondolat, mind a hozzá szükséges algoritmikus számolóprogram roppant egyszerű és rövid. A programot látványossá tevő rajzoltatás programozása azonban megbeszélést érdemel.



Az ábráról leolvashatjuk az alapötletet: a 2 egység sugarú negyedkör területének a számértéke egyenlő a pí számértékével. Ez a terület azonban az ábra szerint beírt téglalapok területének az összegével jól megközelíthető, ha elég sok téglalapra bontjuk. Ha a 2 egységnyi sugarat n részre osztottuk, akkor n—1 téglalapunk lesz. Valamennyinek az alapja 2/n, az i-ediknek a magassága pedig:

$$y_i=\sqrt{2^2-\left(i\cdot\frac{2}{n}\right)^2}$$

Elegendő a magasságokat összegeznünk és az összeget szoroznunk a közös alappal. 100 részre való osztás esetén már elég jó közelítést kapunk a pí-re. A feladat lényegének teljes megoldása a 2. program 70., 180., 190. és 270. sorában található. Nézzük most azonban az egyáltalán nem lényegtelen szemléltetést, a rajzoltatást.

A felnagyítható fél képernyő nagy lehetősége az iskolaszámítógépnek. Az írás ilyenkor messzebbről is jól olvasható, a rajzoláshoz használt „elemi pont" négyzet alakú. Erre kéri az áttérést programunk 30. sora. Az 50. sor szövege az alapötlet és az eljárás lényegét ismerteti a felhasználóval.

A vázlat alapján (lásd az ábrát!) eldöntjük, hogy az origó a képernyő (3;43) pontjában lesz, az „egység" pedig 20 képpont. Ekkor már szinte adódik a tengelyek, a beosztás és a körív megrajzoltatása (90—150. sorok).

A legnehezebb programozási feladat a megadott számú beírható téglalap kifestése. Három egymásba skatulyázott ciklussal érjük el, hogy: 1. minden téglalapot sorra vegyen, 2. azokat az alap egyik végétől a másikig, 3. a tetejüktől a talpukig kifehérítse (170—250. sorok).

A 270-es sorral beíratjuk a terület közepébe a kifestett terület mértékszámát, a 260-as sorral pedig a negyedkör mellé a pí értékét négy tizedes pontossággal. A 40-es, 60-as és a 280-as sorokkal azt érjük el, hogy várakozzék a gép, amíg valamelyik billentyű lenyomásával nem kérjük a program folytatását. Ha új adat beadását kérő sorra ugrunk vissza, mint itt az utolsó sorban tettük, akkor vigyáznunk kell arra, hogy a felhasznált változókat nullázzuk. Itt ezt az SY-változóval kellett megtennünk, a 160-as sorban.

Matematikailag nézve a feladat lényege: a terület kiszámítása szinte mellékesen történt meg a rajz kifestése, illetve az eredmény kinyomtatása közben (180., 190., ill. 270. sor).

Programunk elég lassú közelítéssel dolgozik. Most inkább a könnyű érthetőségre és főleg a szemléletességre törekedtünk. Készítsünk programot, amely a beírt téglalapok helyett a beírt derékszögű trapézokkal dolgozik. Ha vázlatot készítünk, már ránézéssel is látjuk, hogy ez sokkal gyorsabb közelítést ad, hiszen nem n—1, hanem n trapézunk van, mert az utolsó háromszöget is trapézként kezelhetjük.

Kovács Mihály

```
10 REM    ** NEVEK, SZAVAK RENDEZESE **
20 CLS: DEFSTR A,S
30 INPUT "HANY ADATOT RENDEZZEK"; N
40 DIM A(N)
50 FOR I=1 TO N
60 PRINT I;". ";:INPUT A(I)
70 NEXT I
80 REM  *****  R E N D E Z E S  *****
90 FOR I=1 TO N-1
100 Z=1
110     FOR J=1 TO N-I
120     IF A(J)<=A(J+1) THEN 150
130     S=A(J):A(J)=A(J+1):A(J+1)=S
140     Z=0
150     NEXT J
160 IF Z=1 THEN 180
170 NEXT I
180 PRINT: PRINT"A RENDEZETT ADATOR:"
190 FOR I=1 TO N: PRINT A(I): NEXT I
200 IF INKEY$="" THEN 200 ELSE RUN
```

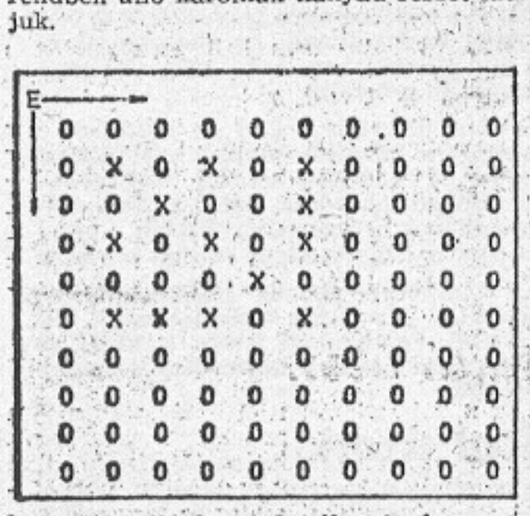
```
10 CLS
20 PRINT"     PI KISZAMITASA
30 PRINT: PRINT"TERJEN AT
A BALOLDALI FEL KEPERNYORE!"
40 IF INKEY$="" THEN 40
50 PRINT:PRINT"     AZ ALAPOTLET:
A 2 EGYSEGNYI SUGARU NEGYEDKOR
TERULETE EPPEN PI EGYSEG.
EZ A TERULET PEDIG BEIRT TEGLA-
LAPOK TERULETENEK OSSZEGEKENT
JO KOZELITESSEL SZAMOLHATO."
60 IF INKEY$="" THEN 60
70 CLS:INPUT"HANY RESZRE OSSZAM";N
80 REM A TENGELYEK ES A NEGYEDKOR MEGRAJZOLASA
90 FOR X=0 TO 50: SET(X,43): NEXT X
100 FOR Y=3 TO 47: SET(3,Y): NEXT Y
110 PRINT@64,"2";: PRINT@8*64,"1";
120 PRINT@15*64+12,"1";: PRINT@15*64+23,"2";
130  PRINT@15*64,"0";
140 FOR A=0 TO 1.58 STEP.2
150 SET(3+40*COS(A),43-40*SIN(A)): NEXT A
160 SY=0:REM **KIFEHERITES ES SZAMOLAS**
170 FOR I=1 TO N-1
180 X=I*40/N: Y=SQR(1600-X*X)
190 SY=SY+Y
200     FOR X1=3+X-40/N TO 3+X
210       FOR Y1=45-Y TO 43
220       SET(X1,Y1)
230       NEXT Y1
240   NEXT X1
250 NEXT I
260 PRINT@3*64+20,"PI=3.1415...";
270 PRINT@10*64+6,SY/20*2/N;
280 IF INKEY$="" THEN 280 ELSE 70
```

µC Iskolaszámítógép-programok

MATEMATIKAI MODELL

Előző számunkban A kutya és a csont címmel közöltünk példát és megoldást arra vonatkozóan, miként kezeljünk olyan esetet, amikor az életben felvetődött problémának először a matematikai modelljét készítjük el.

A következő feladatot szőlőskert feladatnak is nevezhetnénk. Szőlőskertünk bal felső sarkában állunk. Kérdés, hogy ebből a helyzetből a szőlőskert katonás rendben álló karóinak hanyad részét látjuk.



Az E pontból nézve a karók egy része nem látszik. A hatodik sorig és oszlopig a nem láthatókat X-szel jelöltük. Folytassuk a jelölést!

Jól bevált módszerünkkel, a próbálgatással igyekszünk itt is a matematikai modellt megalkotni és a számításhoz szükséges törvényszerűségeket megtalálni (a rajzon a nem látható karókat áthuzzuk).

Ha csak egy karóból áll az egész szőlőskert, akkor természetesen „valamennyi" karót látjuk. 2×2 karó esetén már csak három karót látunk, a negyediket az első eltakarja. A látható és az összes karók aránya 0,75. 6×6 karó esetén 23 karó látszik. Ez az összes karók 0,638 része. 10×10 karó esetén éppen 0,63 a látható és az összes karók aránya. Ekkor az ábrát jól megnézve talán már sikerül észrevennünk valamilyen törvényszerűséget. Ha nem, akkor járjunk el az alábbiak szerint.

Rajzoljunk koordináta-rendszert, amelynek a mi nézőpontunk az origója és az y tengely iránya mutasson lefelé. Ekkor valamennyi szőlőkaróhoz pozitív egész számú koordinátákat rendelhetünk. Az első sor és az első oszlop valamennyi pontja látszik. Ezek valamelyik koordinátája mindig 1. Az első vagyis az (1;1) koordinátájú pont által letakart pontok koordinátái: (2;2), (3;3), ... (n;n). Pl. a (3;1) pont által eltakart pontok koordinátái: (6;2), (9;3), ... (k×3; k×1). Ezt a tapasztalatot a matematika nyelvén úgy fogalmazhatjuk meg, hogy csak azok a pontok látszanak, amelyek koordinátái relatív prímszámok, vagyis legnagyobb közös osztójuk 1.

A szőlőskertből kiindulva megalkottuk annak matematikai modelljét. A feladat neve: a síkrács látható pontjai. A teendőnk pedig az, hogy n oldalú síkrács esetén kiválogassuk és összeszámláljuk azokat a pontokat, amelyek koordinátái relatív prímszámok és számukat elosszuk az összes pontok számával.

Két szám relatív prímszám, ha legnagyobb közös osztójuk 1. A legnagyobb közös osztót pedig a számítógép számára alkalmas euklideszi osztási algoritmussal határozzuk meg. A nagyobb számot addig osztjuk a kisebbel — a nagyobbat mindig a maradékkal helyettesítve —, amíg a maradék zérus nem lesz. Az utolsó osztó a két szám legnagyobb közös osztója. Legyen a két szám pl. 25 és 14, illetve 16 és 10. Papíron végezze az eljárást:

25	14 illetve	16	10
11	14	6	10
11	3	6	4
2	3	2	4
2	1	2	0
0	1		

25 és 14 relatív prímszámok. Az ilyen koordinátájú pont tehát látható. A 16 és 10 nem relatív prímszámok. Az ilyen koordinátájú pont tehát nem látható.

A maradék meghatározása a számítástechnikában az ún. maradékfüggvénnyel történik. A és B számpár esetén ez így fest:

$$A=A-INT(A/B)\,B$$

Ezt természetesen így olvassuk: A és B osztásakor nyert maradék egyenlő az előző A, mínusz A/B egész részének és B-nek a szorzata.

Miután minden algebrai részletet megértettünk, érthetővé válik az elegáns rövid 1. program. A 40-120as sorok nemcsak kiválogatják azokat a pontokat, amelyeknek koordinátái relatív prímszámok, hanem a 110-es sorban össze is számolják, sőt a képernyőn ábrázolják is őket N=15 oldalhosszig. A 130-160-as sorok pedig szövegesen is kinyomtatják munkánk eredményét.

Szép matematikai probléma analizálására is alkalmassá válik programunk, ha a 30-as sort, amely az oldalszámot korlátozza, a 110-es és 120-as sorokból pedig a rajzolást végző SET utasításokat töröljük. 20-30-40 oldalszámok esetén futtatva programunkat azt tapasztaljuk, hogy a futás egyre hosszabb ideig tart, de mintha az arányszám 0,63 közelében stabilizálódna. Pedig a „józan megfontolás" mást sugall. Ha az oldalhosszat minden határon túl növeljük, akkor minden látható pont végtelen sok mögötte állót takar el. Azt gondolnánk tehát, hogy a pontok számának növelésével a látható és az összes pontok aránya csökken. Az eddigi néhány ezer pontra kapott eredmény még nem sokat bizonyít. Hogyan vizsgálhatnánk meg a feladatot pl. 10 000-es oldalhosszúság, vagyis százmillió pont esetére?

A fenti módszerrel aligha! Hisz pl. 900 pont esetére a futási időt lemérve, majd ebből számolva a szükséges futási időt, óriási számot kapunk. Van azonban egy más mód, amely célhoz vezet. Vegyünk 10 000 oldalhosszúságú síkrácsot és pontjai közül véletlenszerűen válasszunk ki ezret. Nagy valószínűséggel mondhatjuk, hogy ami erre a véletlenül kiválasztott 1000 pontra igaz, az valamennyire is igaz.

Az erre szolgáló 2. program még rövidebb is, mint az előző. Az eddigiek után már úgy gondoljuk, hogy bővebb magyarázatra nem is szorul. Mi csak 100 pontot dobattunk a türelmetlenebbek kedvéért, de 1000 pont jobb közelítést ad. Azt tapasztaljuk, hogy itt is 0,63 közelébe eső arányszámot kapunk, főleg, ha 20-as sornak beírjuk a véletlen számokat „véletlenebbé" tevő RANDOM utasítást is.

Eljárásunk nem matematikai értelemben vett bizonyítás ugyan, de olyan „valószínűvé tevő" eljárás, amelyre a számítógép nyitotta meg a lehetőséget.

Kovács Mihály

```
10 CLS: PRINT@10,"A SÍKRÁCS LÁTHATÓ PONTJAI"
                 ==========================
20 PRINT: INPUT"OLDALHOSSZ (0 < N < 16)"; N
30 IF N>15 THEN 20
40 CLS:FOR X=1 TO N
50     FOR Y=1 TO N
60     A=X: B=Y
70     A=A-INT(A/B)*B
80     IF A=0 THEN 110
90     B=B-INT(B/A)*A
100    IF B><0 THEN 70
110    IF A+B=1 THEN S=S+1: SET(5*X,3*Y)
120 NEXT Y,X: SET(0,0)
130 PRINT@240,"OLDALHOSSZ:";N;
140 PRINT@304,"LATHATO    :";S;
150 PRINT@368,"ARANYSZAM :";
160 PRINT@436,S/(N[2);
170 IF INKEY$="" THEN170 ELSE 20
```

1. program

```
5 CLS
10 PRINT@10,"A SÍKRÁCS LÁTHATÓ PONTJAI
             ==========================",
40 CLS:FOR X=1 TO 100
60 A=RND(10000): B=RND(10000)
70 A=A-INT(A/B)*B
80 IF A=0 THEN 110
90 B=B-INT(B/A)*A
100 IF B><0 THEN 70
110 IF A+B=1 THEN S=S+1
120 NEXT X
130 PRINT@240,"100-BOL"
140 PRINT@304,"LATHATO    :";S;
150 PRINT@368,"ARANYSZAM :";
160 PRINT@436,S/100
170 IF INKEY$="" THEN170 (ELSE 20)
```

2. program

ABC 80 PROGRAMOK

Lissajous-görbék

Az egymásra merőleges harmonikus rezgések összetevése a fizika látványos, a tanulók által kedvelt területe. Az így létrejövő Lissajous-görbéket a fizikaórákon oszcilloszkóp segítségével szokás bemutatni. Az itt közölt program nem helyettesíti ezt, de módot ad arra, hogy különböző, időben állandó paraméterek esetén tanulmányozzuk ezeket a görbéket.

A program a kapott adatokat transzformálja úgy, hogy az arányok megmaradjanak, de a képernyő elegendően nagy részét töltse ki az ábra (140. sor). Védekeztünk a hibás adatok beadása ellen is (ONERRORGOTO, 130. sor 170. sor, 210. sor).

A számítógép a görbe rajzolását akkor fejezi be, amikor már elegendően hosszú ideig nem talál még ki nem gyújtott fénypontot. A BASIC-nyelv interaktív tulajdonságát felhasználva a program az indítás után irányítja a felhasználót, a hanggenerátor alkalmazása kizárólag színesítést szolgál.

A következő táblázat olyan adatcsoportokat tartalmaz, amelyek megadása esetén tetszetős, nem túl bonyolult görbéket kapunk.

Sorszám	A	B	O_1	O_2	P/
1	1	1	1	2	2
2	1	1	1	2	1
3	1	1	1	2	15
4	2	1	1	1	1
5	2	1	2	2	2
6	2	1	1	1	3
7	1	1	3	4	2

```
10 ; CHR$(12)
20 ; '        LISSAJOUS-GÖRBÉK'
30 ; '        ****************'
40 ; '  (EGYMÁSRA MERŐLEGES HARMONIKUS'
50 ; '     REZGÉSEK ÖSSZETEVÉSE)'
60 FOR I=1 TO 2000 : NEXT I : ; : ;
70 ; 'X=A*SIN(O1*T)'
80 ; 'Y=B*SIN(O2*T+D)' : ; ' '
90 FOR I=1 TO 2000 : NEXT I
100 ; ' AZ ADATOKAT SI-BEN KEREM!' : ; '-'
110 ONERRORGOTO 110 : ; 'A='; : INPUT A
120 ONERRORGOTO 120 : ; 'B='; : INPUT B
130 IF A(=0 OR B(=0 THEN 110
140 IF 24/A)24/B THEN P=24/B ELSE P=24/A
150 ONERRORGOTO 150 : ; 'O1='; : INPUT O1
160 ONERRORGOTO 160 : ; 'O2='; : INPUT O2
170 IF O1(=0 OR O2(=0 THEN 150
180 IF FIX(O1)=0 OR FIX(O2)=0 THEN 210
190 IF FIX(O1)>FIX(O2) THEN O2=O2/FIX(O1)/2 : O1=O1/FIX(O1)/2 : GOTO 210
200 O1=O1/FIX(O2)/2 : O2=O2/FIX(O2)/2
210 ONERRORGOTO 210 : ; 'D=PI/'; : INPUT N : D=PI/N
220 FOR I=1 TO 24 : ; : ; CHR$(151); : NEXT I : I=Q : Q=0
230 I=I+1 : ONERRORGOTO 230
240 X=P*A*SIN(O1*I) : Y=P*B*SIN(O2*I+D)
250 IF DOT(Y+34,X+35)=0 THEN Q=0 ELSE Q=Q+1
260 IF Q=18 THEN FOR J=1 TO 3200 : NEXT J : GOTO 280
270 SETDOT Y+34,X+35 : OUT 6,FIX(X*Y) : GOTO 230
280 ; CHR$(12);'FOLYTASSUK?'; : GET V$ : ; ' '
290 IF V$='N' THEN ; 'VISZONTLÁTÁSRA!' : STOP : END
300 GOTO 110
```

Nagy számok, nagy pontosság

A tudományban és a technikában sokszor szükséges, hogy vagy nagy számokkal, vagy nagy pontossággal és még nagy sebességgel is végezzék a számításokat. Ez csak nagy, pl. 64 bites gépekkel vagy még ennél is nagyobbakkal érhető el igen nagy költséggel.

A mikroszámítógépek mindössze 8 bitesek, tehát közvetlenül csak a 0—255 közötti számokkal dolgoznak. A gépbe beépített programokkal, a sebesség rovására érik el, hogy a gyakorlatban szükséges, viszonylag nagy számot is tudnak kezelni, de csak korlátozott pontossággal.

Iskola-számítógépünk pl. 38 decimális jegyig végez számításokat, pontossága pedig a változók típusa szerint 5, 6, illetve 16 jegy lehet. (Lásd a BASIC kézikönyv 6-7. és 92. lapjait). Az *egész* típusú változó (integer) jele a változó neve után tett százalék jel (1. a ábra). Az ilyen változó értéke csak — 32 768 és + 32 767 közötti egész szám lehet. Ezek az ún. kettes komplemenses, előjeles kétbyte-os számok. A velük végzett műveletek teljes pontosságúak, de természetesen a műveletek eredménye is csak a fenti határok

```
a, Egész típusú változók:
   A%=548: B%=43
   PRINT A%*B%
   23564
b, Egyszeres pontosság:
   A!=123456
   PRINT A!
   123456
   B=123456789
   PRINT B
   1.234567E+06
c, Kétszeres pontosság:
   A#=12345678: B#=99999999
   PRINT A#*B#
   1234567787654322
d,  A#=1/3
   PRINT A#
   .3333333432674408
e,  A#=1/3#
   PRINT A
   .3333333333333333
```

1. ábra: Változótípusok

közötti ötjegyű szám lehet. Az *egyszeres* pontosság jele a változók után tett felkiáltójel. Ez a jel el is hagyható. A nagyság 38 decimális jegy, a pontosság pedig 6 jegy a hetedikről vett kerekítéssel (1. b ábra). A *kétszeres* pontosság jele a kettős kereszt (1. c ábra). Az ilyen változókkal végzett műveletek 16 jegyig pontosak. Vigyázni kell azonban arra, hogy minden adat kétszeres pontosságú legyen. Pl. az 1. d ábra osztása téves eredményt ad, az 1. e alakba átírva azonban jót.

A típusok jelét fárasztó minden egyes változó után kiírni. Egyszerűbb, ha a program elején a megfelelő változókat a programban használt típusúra definiáljuk. Pl. a DEFINT I, J, K utasítás a program valamennyi, ilyen betűvel kezdődő változóját egész típusúvá teszi.

A számok nagysága és a pontosság szempontjából tehát az iskola-számítógép határai: 38 decimális jegy nagyság és 16 jegy pontosság. Hogyan végezhetnénk el pl. egy osztást tetszés szerinti pontossággal, vagy kiszámíthatnánk-e pl. a 100 faktoriális értékét, amiről sejtjük, hogy 150 jegyűnél is nagyobb szám? Némi ügyeskedéssel és programozási fogásokkal ez is lehetséges.

Osztás tetszés szerinti pontossággal

Már általános iskolában megtanultuk, hogy írásban hogyan kell egy osztást tetszés szerinti pontossággal elvégezni. Így kapjuk pl. hogy 13 : 7 = 1, 857 142 857... A középiskolában azután beláttuk, hogy — a jelen esetben — a hatodik tizedesjegy után nem is érdemes tovább számolnunk, hisz ugyanez a hat jegy ismétlődik vég nélkül. A maradék ugyanis legfeljebb csak 6-féle lehet. Hasonlóképpen:

```
422:1665=0,25345...
```

Itt az utolsó három tizedesjegy ismétlődik vég nélkül, pedig a maradék elvileg 1664-féle lehetne. A matematikusok ezt szabatosan így mondják: két természetes szám osztási eredménye vagy véges szám, vagy végtelen, de mindig szakaszos tizedes tört. A szakasz hossza mindig kisebb, mint az osztó értéke, de a tényleges hosszát csak a művelet elvégzésével tudjuk megmondani.

Az 1. programunk teljes egészében a papíron végzett osztás algoritmusát követi. Két, tetszés szerinti szám osztása tetszés szerinti pontossággal végezhető el vele. Érdekes feladat a program kiegészítése olyanná, hogy a szakasz újrakezdésekor a program futása magától leálljon. Próbáljuk ki pl. 1:181 esetére. Előruljuk, hogy 172 jegyből áll a szakasz. A tárolókkal készüljünk fel rá.

Sokjegyű számok szorzása

Lényegében ezt is elvégezhetjük a számítógépen az általános iskolában tanult és begyakorolt algoritmus felhasználásával. Azonban a programnak a legegyszerűbb alakban való elkészítése is nagy figyelmet kíván. Nézzük először papíron a szorzást viszonylag kis számokkal. A törvényszerűségek így is megfigyelhetők és összegyűjthetők.

```
 6842·385
 34210
 54736
 20526
 2634170
```

A szorzást számjegyenként végeztük és a legkisebb helyértékű jegyekkel kezdtük. A szorzat annyi vagy eggyel kevesebb jegyű lesz, mint ahány jegyű a szorzandó és a szorzó együttvéve. Ha két számjegy szorzata nagyobb mint 9, akkor az átvitelről is gondoskodnunk kell. Erre főleg a bal oldali utolsó jegy szorzásakor ügyeljünk. Az új részletszorozatokat egy-egy jeggyel balra írtuk. Ez a részletszorzat 10-zel való szorzását jelenti.

Ajánlatos a részletszorzatokat nem a szorzás befejeztével egyszerre összeadni, hanem pl. az egyes részletszorzatok elkészülése után azonnal. — A legegyszerűbb minden számjegyet külön (indexes) tárolóba tenni.

A 2. program a fenti tapasztalatok felhasználásával készült. A program első harmada az adatok bekérése. A második harmad (150-től 280-ig) a szorzás elvégzése. A harmadik pedig a szorzat kinyomtatása. Figyeljük meg, hogy az első jegy kírását hogyan akadályozza meg a program, ha az 0.

Programunkban kezdő adatként szerepel a szorzandó és a szorzó számjegyeinek a száma: N és M. Az egyes számjegyeket külön-külön kéri be a program és helyezi el az A és a B vektorokban. A részletszorzat a C, a részletszorzatok összege pedig a D vektorba kerül. Feladatként próbáljuk megvalósítani, hogy kezdőadatként csak a két számot kelljen a gépbe beadnunk. A gép maga állapítsa meg, hogy hány jegyűek a számok, és helyezze el őket az A, illetve a B vektorokban. A vektorok dimenzionálását is végezze maga a gép. Négy vektor helyett legyen három elegendő a szorzás elvégzésére, és az átvitelek ügyének a rendezése az egy-egy jeggyel való szorzás végén történjék.

Faktoriális kiszámítása

Az n faktoriális (n!) az első N természetes szám szorzatát jelenti. N = 10 esetén már 7 jegyű, N = 100 esetén pedig már 156 jegyű a szám. Értékét meghatározni csak a szorzások elvégzésével lehet.

A végén levő nullák számát azonban a tényezőkben levő nulla jegyek és a tényezők végén levő 2-es 5-ös jegypárok számából kiokoskodhatjuk.

A kiszámítást végző rövid és viszonylag gyors 3. programot egy diák készítette évekkel ezelőtt. Mesteri példája annak, hogy a String-függvényeket hogyan lehet bonyolult feladatok végzésére, számításokra is felhasználni. A program működésének a magyarázata azok számára, akik a String-függvények kezelésében nem eléggé járatosak, hosszadalmas lenne. Ehelyett most inkább azt mutatjuk be, hogy a mások által készített nehéz

(Folytatás a 11. oldalon.)

```
10 REM  SOKJEGYŰ SZÁMOK SZORZÁSA
        ========================
20 CLS: DEFINTA-Z
30 DIM A(25),B(25),C(50),D(50)
40 INPUT"HÁNY JEGYŰ A SZORZANDÓ";N
50 PRINT"KÉREM A SZORZANDÓT:"
60 FOR I=N-1 TO 0 STEP-1
70   INPUT A(I)
80 NEXT
90 INPUT"HÁNY JEGYŰ A SZORZÓ";M
100 PRINT"KÉREM A SZORZÓT:"
110 FOR I=M-1 TO 0 STEP-1
120   INPUT B(I)
130 NEXT I: CLS
140 '**************************
150 FOR I=0 TO M-1
160    FOR J=0 TO N-1
170      S=A(J)*B(I)+AV
180      AV=INT(S/10): S=S-AV*10
190      C(J+I)=S
200    NEXT J
210    C(J+I)=AV: AV=0
220    IF I>0 THEN C(I-1)=0
230    FOR K=0 TO N+M-1
240      D(K)=D(K)+C(K)+AV
250      AV=INT(D(K)/10)
260      D(K)=D(K)-AV*10
270    NEXT K
280 NEXT I
290 '**************************
300 I=N+M-1
310 IF D(I)>0 THEN J=I: GOTO 330
320 I=I-1: GOTO 310
330 PRINT D(J);
340 J=J-1
350 IF J>=0 GOTO 330
```

2. program

```
10 REM    n! számítása
          ============
                        Kertész L.IV.o.
20 DEFINT N,I,J,A,B,Z:DIM A(200):A(1)=1
30 INPUT"MEDDIG SZÁMOLJAM"; N:CLS
40 FOR I=1 TO N
50    FOR J=1 TO INT(1.75*I)
60       A$=STR$(A(J-1))
70       B=A(J)*I
80       A(J)=B+VAL(LEFT$(A$,LEN(A$)-1))
90       A(J-1)=VAL(RIGHT$(A$,1))
100    NEXT J: PRINT I
110 NEXT I
120 PRINT: PRINT I-1;"! = ";
130 Z=INT(1.75*I)
140 IF A(Z)>0 THEN B=Z:GOTO 160
150 Z=Z-1:GOTO 140
160 PRINT A(B);
170 B=B-1
180 IF B>0 THEN 160
190 GOTO 190

 RUN
  30! = 265252859812191058636308480000000
```

3. program

```
5 CLS:     Osztás
  tetszés szerinti pontossággal
10 INPUT"Osztandó,Osztó:";A,B
20 E=INT(A/B)
30 PRINT E;",";
40 A=(A-B*E)*10
50 H=INT(A/B)
60 M=A-B*H
70 PRINT H;
80 A=M*10
90 GOTO 50
```

1. program

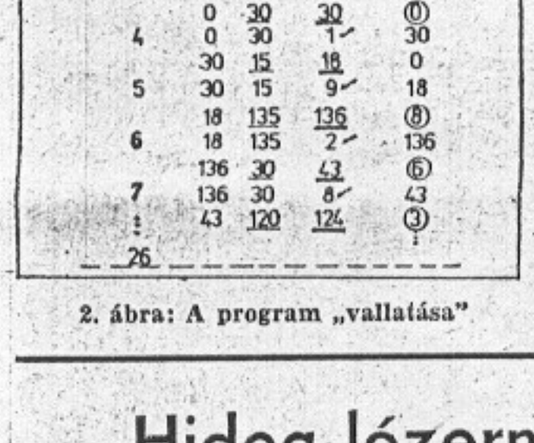
ISKOLASZÁMÍTÓGÉP-PROGRAMOK

(Folytatás a 10. oldalról.)

programokat hogyan kell „vallatnunk", hogy azután a működésüket könnyebben megérthessük.

A program lényege nyilván a 40—110. sorokban rejtőzik. Töröljük a 100-as sorban a PRINT utasítást, és iktassuk be programunkba a következő „vallató" sorokat:

```
45 PRINT I
55 PRINT " ";J;A$; B;A(J);A(J-1)
95 PRINT " ";" " A$;B;A(J);A(J-1)
```



2. ábra: A program „vallatása"

Futtassuk programunkat pl. N = 50 esetén, és időnként állítsuk le a program futását a SHIFT és az (á) gombok együttes lenyomásával. A 2. ábrán látjuk a kapott nyomtatási képből azokat a részleteket, amikor a gép 1-gyel, 2-vel, 3-mal, 4-gyel majd 15-tel szoroz. Figyeljük meg gondosan ezt a táblázatot.

A mindenkori szorzó (a bemutatott esetekben 1, 2, 3, 4, illetve 15) természetesen az I oszlopban jelenik meg. A szorzandó éppen soron következő jegye az A(J) oszlop felső soraiban, pipákkal jelöltük meg őket. A szorzatokat a B oszlopban látjuk és aláhúzással jelöltük. A szorzat és az áthozat összegét szintén az A(J) oszlopban látjuk, de az alsó sorokban, kétszeres aláhúzással. A szorzat számjegyei az A-string oszlopából kerülnek a végleges helyükre, az A(J-1) oszlopba. A gép végül is a program nyomtató részének hatására (120—180. sorok) az A vektorból nyomtatja ki az eredményt a legmagasabb helyértéktől kiindulva és vigyázva arra, hogy az A vektor legmagasabb helyein levő, előre meg nem határozható számú felesleges nullákat ki ne nyomtassa. Az eredmény végén álló, és így az A vektor elején levő értékes nullákat természetesen ki kell nyomtatnia.

Ha már ennyire látjuk a szorzást végző algoritmusunk munkájának az eredményét, akkor már a győzelem reményében foghatunk hozzá az algoritmus működésének a megértéséhez. Az eljárás kigondolásához természetesen találékonyságra is szükség volt.

Bemutatott példáinkon látjuk, hogy a mikroszámítógép is alkalmas szinte tetszés szerinti pontosság elérésére. Csak lelemyességgel és a futtatási idővel kell pótolnunk gépünk olcsóságát.

A fenti programmal gépünk a 156 jegyű 100 faktoriálisát kb. 8 perc alatt számolja ki. Közben annak a jelzésére, hogy dolgozik, kinyomtatja azt a tényezőt, amelyikkel éppen szoroz, ahol a munkában tart. Az erre szolgáló utasítás a 100-as sorban található. A vallatáskor ezt kitöröltük, mert akkor zavart volna bennünket. Ha a vallatás után mindjárt futtatni akarunk, akkor a vallató sorokat töröljük ki, ezt az utasítást pedig írjuk vissza.

A program némi módosításával 100!-nál tovább is számoltathatunk egészen gépünk memóriájának, de még inkább türelmünk határáig.

Kovács Mihály

ABC 80 PROGRAM: MUNKABÉR TELJESÍTMÉNY SZERINTI ELOSZTÁSA

A program egy csoport által keresett munkabér teljesítmény szerinti elosztását végzi. Egyszerű példa az arányos osztási feladatok számítógépes megvalósítására. Alkalmazható a középiskolások által termelőmunkán keresett pénz felosztására is.

Az interaktív üzemmód segítségével a számítógép irányítja a felhasználót. Védekezik a hibás adatok beadása ellen is (ONERRORGOTO, 50, 160, 220 sorok).

A program elkészítése egyszerű, gyakorlati alkalmazhatósága nyilvánvaló, jól alkalmazható a számítástechnika oktatásában is.

Tarcsay Tamás

```
10 ; CHR$(12)
20 ; "HANYAN VETTEK RESZT A MUNKABAN?"
30 ONERRORGOTO 20
40 INPUT N
50 IF N<=0 OR N<>INT(N) THEN 10
60 DIM T$(N),A(N)
70 ; CHR$(12)
80 ; "KEREM A DOLGOZOK NEVET ES TELJESITMENYET!"
90 FOR I=1 TO 1500 : NEXT I
100 O=0
110 FOR I=1 TO N
120 ; CHR$(12)
130 ; I" .DOLGOZO NEVE: "; : INPUT T$(I)
140 ONERRORGOTO 140
150 ; I" .DOLGOZO TELJESITMENYE: "; : INPUT A(I)
160 IF A(I)<0 THEN 140
170 O=O+A(I)
180 NEXT I
190 ; "MENNYI AZ ELOSZTHATO MUNKABER?"
200 ONERRORGOTO 190
210 INPUT P
220 IF P<=0 THEN 190
230 E=P/O
240 ; CHR$(12)
250 FOR I=1 TO N
260 ; T$(I);" MUNKABERE: "E*A(I);" FT." : ;
270 GET V$
280 NEXT I
```

Hideg lézermaratás

Új lézermaratási eljárást dolgoztak ki az amerikai IBM cégnél. A speciális maratási eljárás előnyösen alkalmazható műanyagok, szerves anyagok (csontok és fogak) megmunkálásában az infravörös lézerek esetében szükséges óvintézkedések nélkül.

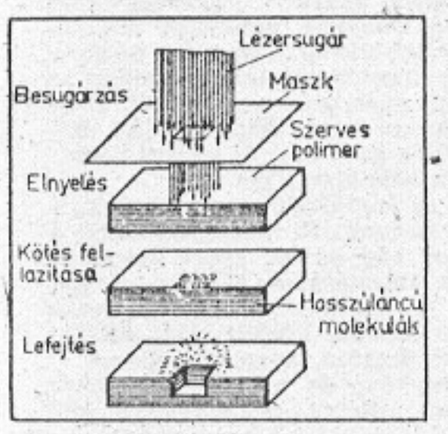
Az új technika, amit lefejtő fotokompenzációnak neveznek, új lehetőséget teremt az integrált áramkörök gyártásában, az orvosi gyakorlatban a műtétek és fogászati kezelések végrehajtásában.

Az eljárásban alkalmazott lézersugár hullámhossza az ultraibolya tartományba esik (a fejlesztők 193 nanométeres argon-fluor lézert alkalmaztak.) Ezek a lézersugarak a hősugaraknál jobban fellazítják az anyag molekuláinak kémiai kötését és magukkal ragadják az eltávolítandó anyagot. A fejlesztők szerint az eljárás kapcsán egy miniatűr robbanás következik be a besugárzott anyag felületén, és noha a lézersugár maga nem látható, de hatása — a felületről eltávozó leszakadt molekulák révén gyenge parázs fény formájában — jól megfigyelhető.

Az IBM fejlesztői úgy találták, hogy a 200×10^{-9} m-nél rövidebb hullámhosszú ultraibolya sugarakat a legtöbb szerves anyag jól elnyeli és az elnyelés csak csekély mélységű bemaródást eredményez. Megállapították továbbá a vékony műanyagok lézerrel való kezelésekor, hogy minden szerves anyagnak létezik egy intezitási küszöbértéke a lézersugarak behatolásával szemben. Ha ezt a küszöbértéket túllépjük, a besugárzott anyag felületéről igen sok molekula leválik. Ez a hatás nemcsak a lézersugarak intenzitásától függ, hanem a leváló molekulák kémiai kötéslazító hatásától is, ami alacsony hőmérsékletű párolgást eredményez. Természetesen a leírt jelenséghez a lézersugár szolgáltatja az energiát.

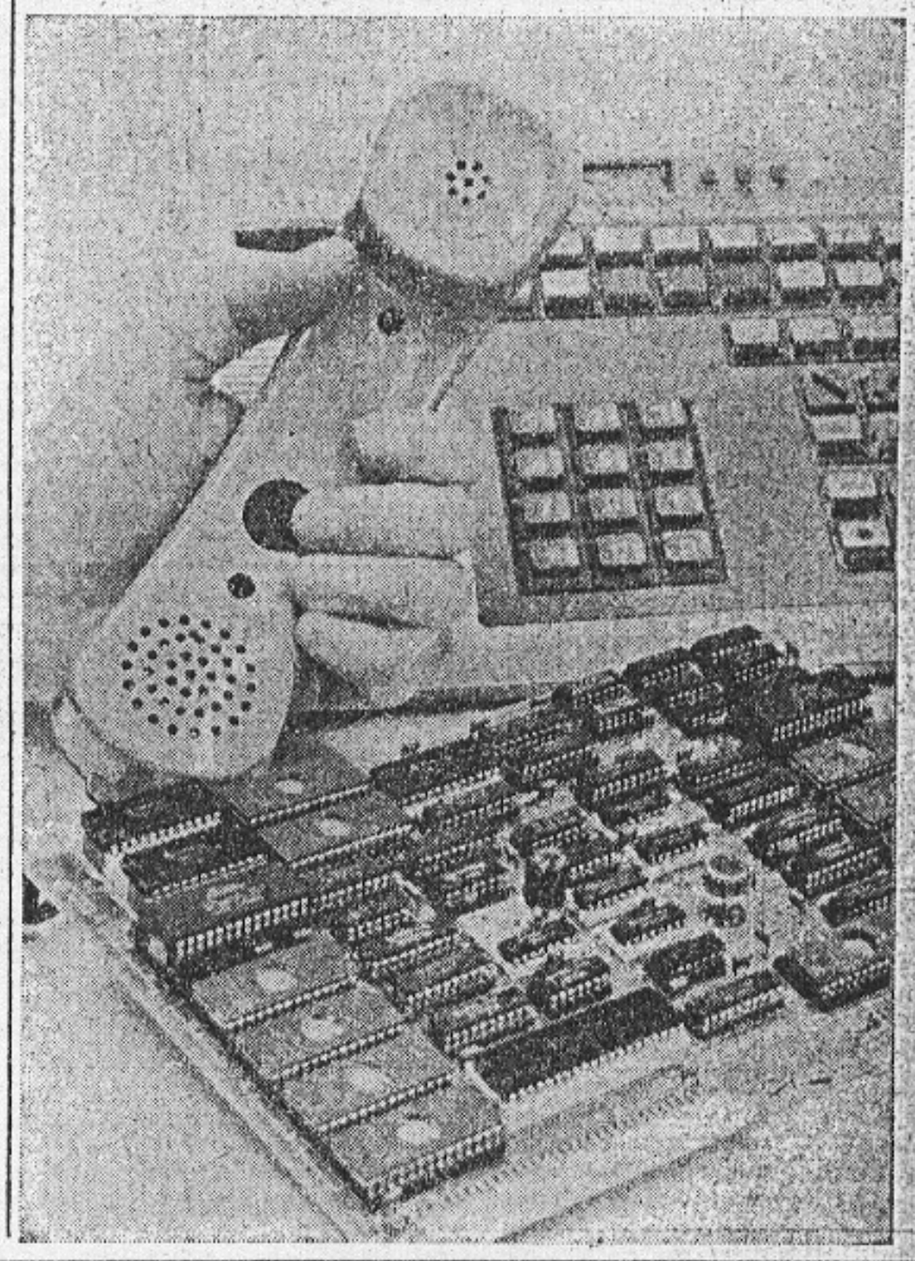
Az alkalmazási területről szólva a kutatók úgy vélik, hogy az elvet remekül lehet alkalmazni az űrrepülőgépeken. Amennyiben azokat bevonják egy elégetendő réteggel, azok az atmoszférába belépve a keletkezett hőt le tudják adni.

A lézeres lefejtő eljárás jól alkalmazható a fotolitográfiában és alapvető eljárás lehet a mikroáramkörök gyártásában. A mikroáramkörök gyártásakor az áramkört sugárzással marják ki a sugárzásra érzékeny felületből. A lézersugaras maratás sokkal pontosabb a fény-, a röntgen-, az elektronsugaras és a kémiai maratási eljárásoknál.



Nyelvet értő számítógép

Gépek és számítógépek mikrofonon keresztül felvett vezényszavakkal vezérelhetők a digitális technikai és a félvezető eszközök gyors fejlődésének eredményeként. A Siemens cég utazási irodák céljaira fejlesztett ki olyan számítógépet, amely meghatározott szókészletet le tud fordítani. Ez a számítógép előnyösen használható pl. utazási megrendelő űrlapok kitöltésére, ami rendkívül időrabló, különösen akkor, ha a megrendelés szövegét idegen nyelvre le kell fordítani.



Egy másik nagy alkalmazási területet jelent a távbeszélő hálózat, amire a vasúti és légi közlekedés információs központjai, a bankok, az áruházak és a könyvtárak is rákapcsolódnak. A jövőben ezek szolgáltatásait beszéddel lehet majd lehívni. Addig azonban még jelentős kutatómunkát kell elvégezni ahhoz, hogy a számítógép minden telefonáló személy beszédét megértse.

A Siemens által gyártott beszédfelismerő telefonközponti munkahely a tárcsázást helyettesítő gombnyomásokat a megfelelő szavak vételével helyettesíti és ennek megfelelően valósítja meg a kapcsolást. A rendszer egy meghatározott személy 40 szavát tudja azonosítani és az azokhoz rendelt tevékenységeket kiváltani. Ebben az eljárásban a digitalizált beszédjelből olyan jellemzőket származtatnak, amelyeket összehasonlítanak az ugyanazon személy által már előzőleg bemondott szókészlettel. A beszédfelismerő rész hardverje három olyan jelfeldolgozóból áll, amelyek 4 millió művelet/s feldolgozási sebességgel működnek. Háttértárolóként mágneses tárat alkalmaznak.

Az ábrán a beszédre kapcsoló telefonközponti készülék és annak szerelt panelje látható.

µC Iskolaszámítógép-programok

1	2	3	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	(25)	~~26~~	~~27~~	~~28~~	29	~~30~~
31	32	~~33~~	34	(35)	~~36~~	37	~~38~~	~~39~~	~~40~~
41	~~42~~	43	~~44~~	~~45~~	~~46~~	47	~~48~~	(49)	~~50~~
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

2. ábra: A törzsszámok kiválogatása a „szitálás" módszerével

Folytassuk előző számunkban közölt „lazító" programunkat két szimulációs programmal.

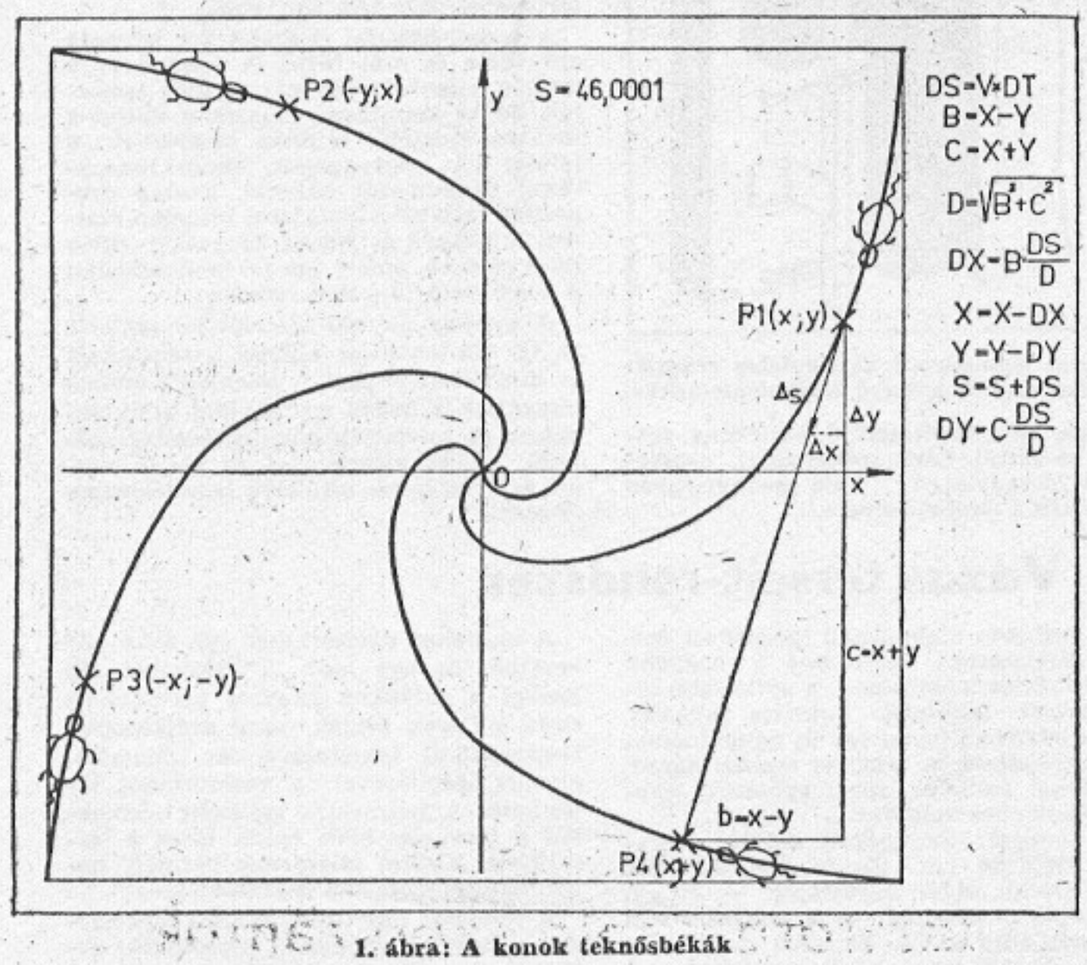
A Gellérthegy magasságú Apolló űrhajóról, amelynek segítségével 15 évvel ezelőtt az első ember eljutott a Holdra, a fiatalabbak már csak hírből tudnak. Ennek mind az öt fokozatát számítógépes szimulációval tervezték meg. Az 1. programunk az első fokozat működését szimulálja a tényleges adatokkal. Ha READ—DATA sorokban (220—210.) a DM-adatot finoman változtatjuk, különböző lehetőségeket kapunk, amelyek közül nyilván a műszakilag és biológiailag a legmegfelelőbbet választották. A program futtatása annyira élményszerű, az elért magasság, sebesség és gyorsulás grafikus ábrázolása annyira szemléletes, hogy szinte az űrhajóban érezzük magunkat. A testünkre ható nehézségi erő ötszörösére növekedését szerencsére nem tapasztaljuk.

A 2. programot izgalmas játéknak szántuk. Az autózást lehet vele gyakorolni vezetői engedély nélkül. Még a karambol sem veszedelmes ebben az esetben. Igyekezzünk azonban minél kevesebb autót „összetörni", mert a cél az, hogy közeledjünk a baleset nélküli vezetés felé. Ez itt nem is olyan könnyű, mert a pálya igen kanyargós. A programot egy tanuló készítette egyévi BASIC-tanulás után. (A BASIC-bővítést ne kapcsoljuk be!)

Következzen talán egy egyszerű, de igen izgalmas játék (3. program). A képernyő bal felső sarkában reszket egy kis levelibéka, mert — fél képernyő esetén — a jobb alsó sarokban leselkedik rá egy hatalmas kígyó. Már el is indul a béka felé. A békát a 4, nyíllal jelölt nyomógombbal minden irányban mozgathatjuk. A képernyőről egyikük se tud kilépni. Segítsük a pici békát! Megmenekülhet, mert a kígyó egy bizonyos időn után kifárad. Ha a programot jól megfigyeljük, akkor a kígyó hosszát, sőt a sebességét is változtatni tudjuk.

A 4. program futtatása igen látványos.

(Folytatás a 11. oldalon.)



1. ábra: A konok teknősbékák

```
10 CLS:PRINT"                Az APOLLO holdrakéta
                  ====================="
20 PRINT"TERJEN AT A BALOLDALI
FEL KEPERNYORE!"
30 IF INKEY$="" THEN 30
40 FOR FI=.2 TO 3.2 STEP.3
50 SET(25*COS(FI)+32,-5*SIN(FI)+45): NEXT FI
60 PRINT@974,"F O L D";
70 A$=CHR$(158)+CHR$(173)
80 FOR G=783 TO 15 STEP-64
90 PRINT@G,A$;
100     FOR I=0 TO 100:NEXT
110 PRINT@G,"  ";
120 NEXT
130 IF INKEY$="" THEN 130
140 PRINT:CLS
150 PRINT"AZ APOLLO HOLDRAKETA INDULASI
TOMEGE 2995 TONNA VOLT.
ELSO FOKOZATANAK HAJTOMUVE 15
TONNA UZEMANYAGOT EGETETT EL
MP-ENKENT. IGY ALLANDOAN 35
MILLIO N TOLOEROT SZOLGALTATOTT."
160 PRINT"AZ ELSO FOKOZAT MUKODESI
IDEJE MINDOSSZE 2.5 PERC."
170 PRINT"SZAMITSUK KI A RAKETA 10 mp-
ENKENT ELERT GYORSULASAT,
SEBESSEGET ES MAGASSAGAT!
  KESZITSUNK GRAFIKONOKAT IS!"
180 IF INKEY$="" THEN 180
190 CLS:PRINT"TERJEN VISSZA A TELJES KEPERNYORE!
200 IF INKEY$="" THEN 200
210 CLS:DATA 2.985E6,3.5E7,1.5E3,9.806
220 READ M,F,DM,G
230 PRINT@986,"IDO                              IDO"
240 FOR E=4 TO 57: SET(E,47): NEXT
250 FOR E=67 TO 120: SET(E,47): NEXT E
260 PRINT@516,"SEBESSEG                GYORSULAS"
270 FOR E=24 TO 47: SET(4,E): SET(5,E)
280 SET(67,E): SET(68,E):NEXT E
290 PRINT@80,"IDO          (sec):
                MAGASSAG      (m):
                SEBESSEG    (m/s):
                GYORSULAS (m/s[2):"
300 FOR T=0 TO 150 STEP 10
310 SET(0,47-S/2000): SET(T/3+4,47-V/100)
320 SET(T/3+67,47-.4*A)
330 PRINT@102,T;:PRINT@166,S;"    ";
340 PRINT@230,V;"    ";:PRINT@294,A;
350 FOR DT=1 TO 100
360 M=M-DM
370 A=F/M-G
380 V=V+A*.1
390 S=S+V*.1
400 NEXT DT,T
410 PRINT@5*64+20,"AZ ELSO FOKOZAT LEVALT!
                        K E S Z !";
420 IF INKEY$="" THEN 420
```

1. program

```
10 CLS:PRINT"          AUTÓVEZETÉS
                              Fábry G.":    PRINT
20 PRINT"    VEZETES A 4-ES, 5-OS ES 6-OS NYOMOGOMBOKKAL
           (BALRA, EGYENESEN, JOBBRA)
    A SZOKOZBILLENTYU A GAZPEDAL"
30 PRINT"    250 IDOEGYSEGIG A TIED A KOCSI"
40 PRINT: PRINT"HA A VEZETESRRE KESZEN ALLSZ,
NYOMJ LE EGY BILLENTYUT!"
50 IF INKEY$="" THEN 50
60 D=25:CLS:G=D+8: CLS
70 A=RND(5)-3:B=RND(6)
80 FOR C=0 TO B
90 H=H+1:IFH>200THEN310
100 PRINT@58,INT(H);: PRINT@0,K;
110 IFM=1ANDN=1THENN=0:H=H-1.5:GOTO170ELSEN=1
120 D=D+A: IF D<1THENA=2:D=0
130 IF D>44 THEN A=2:D=45
140 PRINT@D+960,"$$$          $$$"
150 L=RND(58)+3:IFL<D+20RL>D+13THEN170ELSEPRINT@L+960,"*";
160 L=RND(58):IFL<D+20RL>D+13THEN170ELSEPRINT@L+896,"*":
170 E=PEEK(14416)
180 IF EAND16THENF=F-1
190 IF EAND64THENF=F+1
200 IF EAND128THENM=0ELSEM=1
210 IFEAND32THENF=0
220 F=SGN(F): GE=G: G=G+F
230 IF PEEK(15552+G)=36 THEN :FORI=0TO30:PRINT@G+192,"#";:FORJ=0
TO30-3*I:NEXT:PRINT@G+192," ";:FORJ=0TO30-3*I:NEXT:NEXT:K=K+1:F=
0:R=15552ELSE260
240 IFPEEK(R)=36 OR PEEK(R)=35THENPRINT@R-15360,"$$$     #     $
$$";:G=R-15544:FORJ=0TO250:NEXTJ:GOTO260
250 R=R+1:GOTO240
260 PRINT@G+192,"#";
270 IFN=0THENPRINT@GE+192," ";:GOTO290
280 PRINT@GE+128," ";
290 NEXT C
300 GOTO70
310 FOR L=0 TO 300:NEXT L
320 FOR KI=0 TO 10:CLS:PRINTCHR$(23):FOR L=0 TO 10:NEXT L:PRINT@
470,"I D.O ! ! !":FOR L=0 TO 10:NEXT L:NEXT KI
330 PRINT"OSSZESEN"K"AUTOT TORTEL OSSZE!"
340 PRINT"MASKOR MAJD JOBBAN SIKERUL!"
350 INPUT"JATSZOL MEG EGYET";I$:IFI$="NEM"THENSTOPELSERUN
```

INKEY$ helyett

2. program

```
10 CLS: REM     A KIGYÓ ÉS A BÉKA
15 '            *****************
20 DIMA(25):DIMB(25)
30 X1=0:Y1=0:X=63:Y=46
40 FOR I=0 TO 25
50   N=N+1:A=PEEK(14400)
60   IF A=8 AND Y1>0 THEN Y1=Y1-1.5
70   IF A=16 AND Y1<46 THEN Y1=Y1+1.5
80   IF A=32 AND X1>0 THEN X1=X1-1.5
90   IF A=64 AND X1<63 THEN X1=X1+1.5
100  X=X+SGN(X1-X):Y=Y+SGN(Y1-Y)
110  RESET (A(I),B(I))
120  SET (X1,Y1): SET(X,Y)
130  A(I)=X:B(I)=Y
140  IF X=INT(X1) AND Y=INT(Y1) THEN 190
150  RESET (X1,Y1)
160 IF N>500 THEN 210
170 NEXT I
180 GOTO 40
190 CLS:PRINT@512,"AZ ELERT PONTSZAM:";N
200 IF INKEY$=" " THEN RUN ELSE 200
210 CLS:PRINT@520,"POMPAS EREDMENY!
     M E G M E N E K U L T E L !"
220 GOTO 200
```

3. program

```
10 CLS
20 PRINT"          A KONOK TEKNŐSBÉKÁK"
30 PRINT"          ====================="
40 PRINT"          (Archimédeszi spirál)"
50 PRINT"          ----------------------"
60 PRINT"EGY NEGYZET SARKAIBOL ELINDUL NEGY TEKNOC.
MINDEGYIK A TOLE JOBBRA LEVO FELE TOREKSZIK.
RAJZOLTASSUK MEG UTJAIKAT.
SZAMITSUK KI AZ UTJUK HOSSZAT!"
70 A$=INKEY$:IF A$="" THEN GOTO 70
80 CLS
90 X=23:Y=23:V=1:DT=.2:S=0
100 FOR I=1 TO 10
110    DS=V*DT:S=S+DS
120    B=X-Y:C=X+Y:D=SQR(B[2+C[2):K=DS/D
130    DX=B*K:DY=C*K:X=X-DX:Y=Y-DY
140 NEXT I
150 SET(8*X/3+64,-Y+24):SET(8*Y/3+64,X+24)
160 SET(8*(-X)/3+64,Y+24):SET(8*(-Y)/3+64,-X+24)
170 IF ABS(X)>.5 OR ABS(Y)>.5 THEN GOTO 100
180 PRINT@100,  "S=";S;
190 GOTO 190
```

4. program

µC Iskolaszámítógép-programok

(Folytatás a 10. oldalról.)

A négy teknőc más-más irányba megy, és a végén mégis találkoznak. Az általuk bejárt görbét archimédeszi spirálnak hívják. De talán a futtatásnál is érdekesebb, hogy jóllehet a feladat matematikai megoldásához egyetemi végzettség szükséges, a számítógépes eljárást már a 14 évesek is megérthetik. Az 1. ábra koordinátarendszerének az origója a kép közepén van. Ha az első teknőc pillanatnyi helyzete és ennek koordinátái: P1 (x; y), akkor a másik háromé: P2 (—y; x), P3 (—x; —y) és P4 (y, —x). Ha a teknőcök sebessége V, akkor a kicsiny DT-idő alatt befutott DS = V×DT út egyenes darabnak tekinthető. Az 1. ábra hasonló háromszögeiből a kis koordinátaváltozások, majd az új koordináták számíthatók. Ezt ismételtetjük a géppel nagyon sokszor. Így a megtett út hosszára igen jó eredményt kapunk. Érdekessége a feladatnak, hogy a megtett út hossza éppen egyezik a teknőcök kezdeti távolságával, amely jelen esetben 46 egység.

A törzsszámok problémája több évezrede izgatja az embereket. Egy 2100 éves módszert mutatunk be a törzsszámok kiválogatására (Eratoszthenész szitája, 2. ábra). A 2 törzsszám, a többszöröseit töröljük. A 3 törzsszám, ennek a többszöröseit is töröljük. Ugyanezt tesszük 5 és 7 többszöröseivel is. Így 50-ig már meg is kaptuk a törzsszámokat. Folytassuk a táblázatban megadott 100-ig, esetleg még tovább is. Próbáljunk törvényszerűségeket észrevenni a „szitálásban". Ezeket a törvényszerűségeket használja fel az az aránylag gyors és látványos 5. programunk.

A 6. programunk már igazi „profi" játékprogram. Az egyik diák készítette évekkel ezelőtt, miután már két évet tanult programozni. A játékokat gyorssá és változatossá a gépi kód teszi. A sok adat a programban levő gépi kódú részprogramokhoz szükséges. Elveinkkel ellentétben a program működését most nem szándékozunk megértetni, mert ez nem nyárra való feladat. A program begépelése nem különösebb fáradság, ha valaki értelmesen diktálja. Ha pedig fut, akkor már szalagra lehet venni, és legközelebb már a géppel gyorsan be lehet olvastatni.

Jó játékot kívánunk a szünetre. Szeptembertől majd jöhet ismét a tanulás a számítástechnikában is.

Kovács Mihály

```
10 REM        ERATOSZTHENÉSZ SZITÁJA
              **********************
20 CLS : DEFINT A-Z
25 INPUT "MEDDIG SZITALJAM KI A PRIMSZAMOKAT";N
27 DIMA(N)
30 FOR I=2 TO N : PRINT I; : NEXT I : PRINT
40 FOR I=2 TO SQR(N)
50 IF A(I)=0 THEN PRINT I : FOR K=I*I TO N STEP I:
    A(K)=1 : NEXT K ELSE GOTO 90
60 FOR K=2 TO N
70 IF A(K)=0 THEN PRINT K; ELSE PRINT ".";
80 NEXT K : PRINT: PRINT
90 NEXT I
```

5. program

```
10 CLS:PRINT"          F A L T Ö R Ő - J Á T É K"
15 PRINT"          ----------------------------------"
16 PRINT"                                (C): Fábry Géza"
20 PRINT:PRINT:PRINT"          H A S Z N A L A T I   U T A S I T A S"
25 PRINT:PRINT"A JATEK CELJA A BALOLDALI FALAT TELJESEN LEBONTANI EGY
LABDAVAL.";
30 PRINT"AZ UTO A 6-OS GOMBBAL FELFELE, A 3-ASSAL LEFELE MOZGATHATO."
35 PRINT:PRINT:PRINT"J O     S Z O R A K O Z A S T ! ! !
":PRINT:PRINT
90 '              *** F O P R O G R A M ***
100 DATA33,241,127,126,43,134,214,2,202,160,125,254,120,242,170,125,25
4,104,202,176,125,0,0,0,0,0,0,0,0,0,0,0,33,243,127,126,43,134,214,1,25
4,8,202,208,125,254,45,202,217,125,0,0,0,0,0,0,0,0,33,240,127,70,35,35
,78,151,205,0,127,4,151,205,0,127
110 DATA126,254,128,196,224,125,33,243,127,126,43,134,61,119,79,43,126
,43,134,214,2,119,71,62,2,205,0,127,4,62,2,205,0,127,0,0,0,0,33,80,56,
126,230,64,254,64,204,64,126,126,230,8,254,8,204,0,126,201,0,125
140 '              *** A L P R O G R A M O K ***
150 DATA62,4,50,241,127,195,32,125,0,0,33,255,255,195,154,10,33,245,12
7,86,43,43,0,43,126,0,186,250,32,125,214,5,186,242,32,125,151,50,241,1
27,195,32,125,0,0,0,0,0,62,2,50,243,127,195,58,125,0,151,50,243,127,19
5,58,125
160 DATA54,128,33,241,127,126,254,2,242,238,125,54,4,201,54,0,201
190 '              *** U T O M O Z G A T A S  ***
200 DATA33,245,127,126,254,42,200,6,104,78,52,52,151,205,0,127,12,151,
205,0,127,4,151,205,0,127,13,151,205,0,127,58,245,127,198,3,79,62,2,20
5,0,127,12,62,2,205,0,127,5,62,2,205,0,127,13,62,2,205,0,127,201,0,0,0
210 DATA33,245,127,126,254,4,200,6,104,53,53,78,62,2,205,0,127,12,62,2
,205,0,127,4,62,2,205,0,127,13,62,2,205,0,127,12,12,12,12,12,151,205,0
,127,5,151,205,0,127,12,151,205,0,127,4,151,205,0,127,201
240 '           *** S E T - S U B R O U T I N E  *** /
250 DATA197,213,0,245,33,0,0,121,254,3,250,19,127,214,3,35,195,8,127,4
1,41,41,41,41,41,41,135,17,0,60,25,87,120,31,220,92,127,95,122,22,0,25,13
5,135,135,87,241,254,1,122,250,69,127,242,79,127,246,70,50,62,127,151,
203,70,204,89
260 DATA127,209,193,201,246,134,50,75,127,203,134,195,66,127,246,198,5
0,85,127,203,198,195,66,127,62,1,201,20,201
290 '                *** B E T O L T E S  ***
300 FORA=32000TO32133:READJ:POKEA,J:NEXT
310 FORA=32160TO32240:READJ:POKEA,J:NEXT
320 FORA=32256TO32379:READJ:POKEA,J:NEXT
330 FORA=32512TO32605:READJ:POKEA,J:NEXT
340 POKE16526,0:POKE16527,125
400 CLEAR500:INPUT"A LABDA SEBESSEGE=? (5-30) ";Q:Q=Q-5:B$=CHR$(128):F
ORA=1TO6:B$=B$+B$:NEXT:IFQ>25THENPOKE32131,195ELSEPOKE32131,201
410 PRINT@0,;:FORA=0TO18:PRINTB$;:NEXT
500 A$=CHR$(191):A$=A$+A$:A$=A$+A$:A$=A$+CHR$(128)+CHR$(128)+A$
510 FORA=8TO44:SET(0,A):SET(1,A):NEXT
520 FORA=0TO118:SET(A,45):SET(A,8):NEXT:RESET(104,8):RESET(105,8):RESE
T(104,45):RESET(105,45)
530 FORA=196TO958STEP64:PRINT@A,A$;:NEXT
540 POKE32757,26:FORA=26TO30:SET(104,A):SET(105,A):NEXT:POKE32756,0
550 POKE32752,102:POKE32753,0:POKE32754,RND(36)+8:POKE32755,(RND(2)-1.
5)*2+1
560 U=USR(9):IFU<0THEN700
570 FORA=0TO250-10*QSTEP5:NEXT
580 GOTO560
700 B=PEEK(32757):FORA=8TO8+4:SET(104,A):SET(105,A):NEXT:FORK=0TO5:FOR
KK=0TO30-7*K:NEXT:PRINT@0,"K I M E N T ! ! !";:FORKK=0TO30-7*K:NEXT:PR
INT@0,B$;:NEXT:PRINT@0,"K I M E N T ! ! !";:FORA=0TO200:NEXT
710 L=L+1:IFL=5THEN800ELSEPRINT@0,"EDDIG 5 LABDABOL"L"MENT KI.";:FORA=
0TO47:RESET(118,A):RESET(119,A):NEXT:FORA=0TO200:NEXT:IFQ>25THENPRINT@
0,"E L E G   G Y Ö R S ? ? ?";:FORA=0TO300:NEXT
720 PRINT@0,B$;B$;:GOTO550
800 PRINT@0,"E L F O G Y T A K  A  L A B D A I D ! ! !";:IFQ>25THENP
RINT@480,"( H E - H E - H E )";
805 A$=INKEY$
810 IFINKEY$=""THEN810ELSESTOP
999 STOP
1000 INPUTIN,NX:FORA=INTONX:PRINTA,PEEK(A):NEXT:GOTO1000
```

6. program

Membrán tengelykapcsolók

A tengelykapcsolókkal szemben a felhasználók fokozott követelményeket támasztanak az üzembiztonság tekintetében. A problémák akkor jelentkeznek, ha nagy forgatónyomatékot kell átvinni nagy fordulatszámon. Ilyen célokra fejlesztett ki különféle tengelykapcsolókat az angol Flexibox International Group. Termékei megfelelnek a nagy teljesítményű berendezésekhez, max 25 000/min fordulatszámig és 24 mkN nyomatékig.

A fémmembrán tengelykapcsolók különösen olyan berendezésekben alkalmasak, amelyekben előre pontosan meg kell határozni a merevségi jellemzőket, és azoknak a tengelykapcsoló egész élettartama során közel állandónak kell maradni. E tekintetben az ilyen típusú tengelykapcsolók megfelelnek a legtöbb gyakorlati követelménynek. A fémmembránokban nagyon kevés alakváltozási energia halmozódik fel, így azok hiszterézisvesztesége elhanyagolható. Nem jelentkeznek a hő okozta problémák, amelyek nagy fordulatszámon nagy nyomaték hatására más típusú rugalmas tengelykapcsolókban fennállnak.

A nagy teljesítményű tengelykapcsolókhoz korrózióálló acélból készítik az egymás mellé helyezett gyűrűs membránokat. A csapok elrendezése olyan, hogy a forgatónyomaték a membrángyűrűkben túlnyomórészt húzó igénybevételt keltsen. A membrángyűrűket csavarokkal szorítják össze váltakozó irányban elhelyezett perselyekkel, két alátét között. Az alátétek összeszorító felülete domború, így a tengelyközépvonalak radiális, axiális vagy szögeltérései következtében a membrángyűrűben fellépő hajlítófeszültségek minimálisak az alátétek körüli kritikus zónában. A gyűrűk alakja ezenkívül megfelelő rugalmasságot biztosít viszonylag kis átmérők esetében is, és a centrifugális erők csökkenése következtében nagy fordulatszámon is használhatók.

A csavarok által létesített nyomófeszültség csökkenti a membrángyűrűk közötti súrlódás lehetőségét és az ebből eredő, kifáradás okozta meghibásodást. A csavarfej kialakítása lehetővé teszi a tengelykapcsoló tömegének kis értéken tartását. A csavarokat és anyákat nagy szakítószilárdságú acélból készítik.

A tengelykapcsoló fejlesztési munkáiban sokféle próbapadot használnak. Egyik különösen fontos vizsgálóberendezés a fárasztógép, amellyel a membránanyagok kifáradási határát vizsgálják. Fontos a különböző anyagok élettartamának vizsgálata váltakozó irányú terhelés esetében. A vizsgált anyagok között nem szikrázó anyagokat (pl. monelfémet) is találunk. Egy másik vizsgálati módszer a membránt terhelő hajlítófeszültségnek a kifáradási élettartamra gyakorolt hatása.

Fokozott igénybevételkor lényeges, hogy a központosítási hibák okozta behajlások által keltett feszültségek a lehető legkisebbek legyenek. Ezt nagymértékben befolyásolja az alátétgyűrű összeszorító felületének sugara. Ha a sugár túl kicsi, akkor az a membránban a feszültségeket igen nagymértékben megnöveli. Viszont a túl nagy sugárnak vagy — szélső esetben — a teljesen sík alátétfelületnek ugyanez a hatása, mert a membránban az alátét külső átmérője mentén feszültségcsúcsok keletkeznek.

A véges elemek módszerével végzett kísérleti programot számítógéppel értékelik ki annak a meghatározására, hogy a tengelykapcsolóban hol koncetrálódnak a feszültségek. Az ilyen feszültséganalízis egyik lehetséges területe a membránok vizsgálata. Próbapadokkal egyéb alkatrészeket is vizsgálnak, pl. a tömegcsökkentés vizsgálatára vagy annak a megállapítására, hogy egy meglevő tengelykapcsoló mekkora fordulatszámmal és nyomatékkal üzemelhet.

A cég által gyártott TLHW típusú tengelykapcsolóval pl. turbina és kompresszor között az átvitt teljesítmény 3,25 MW lehet 13 820/min fordulatszámon, a megengedett max. fordulatszám 15 200/min.

Ragasztott fémforgácsoló szerszámok

A keményfém betétes forgácsoló szerszámok betéteinek rögzítését forrasztással, hegesztéssel vagy mechanikus módon oldják meg. A forrasztás vagy hegesztés során a keményfém vágólapkák anyagában a belső feszültségek hatására gyakran mikrorepedések keletkeznek. A gyártásban a selejt értéke 10 ... 20%, bizonyos szerszámok esetében nem ritkán még a 40%-ot is elérheti. Különösen érzékenyen érinti ez a selejtveszteség a gyártókat, mivel a jelenleg alkalmazott keményfém betétes szerszámok 90%-a forrasztással készül.

A hetvenes években a Szovjetunióban kifejlesztett szintetikus szuperkemény anyagok (Elbor—R, Hexanit—R) széles körű elterjedését gátolja a megfelelő rögzítési módszer hiánya.

Rigai szakemberek a keményfém lapkák és betétek szerszámszárhoz való rögzítésére a ragasztást javasolják. A ragasztást leginkább a nagyméretű szerszámok esetében célszerű alkalmazni, például turbina- és kompresszorlapátok hornyainak kimunkálására szolgáló üregelő szerszámokon, amelyeket eddig teljes keresztmetszetben gyorsacélból állítottak elő. Így 40 ... 85% drága ötvözőket tartalmazó gyorsacél-megtakarítást lehet elérni. A bonyolult geometriájú és elkészítési szempontból munkaigényes lefejtőmarók gyártásakor átlagosan 100 gramm keményfémet lehet megtakarítani a ragasztás alkalmazásával, ami a szerszám tömegének kétharmadát teszi ki. Egy vizsgálat alkalmával a ragasztott keményfém betétes dörzsárak 15 %-kal hosszabb élettartamot mutattak, mint a forrasztott kivitelűek.

A ragasztás alkalmazása minden egyes szerszámtípus esetén alapos megfontolást, gyakran szerkezeti változtatásokat igényel. Fontos a megfelelő fizikai, mechanikai és technológiai tulajdonságokkal rendelkező ragasztóanyag helyes megválasztása.

A szerszámragasztásra szolgáló szovjet Instrumentol és UP5—207 típusú ragasztók átlagos nyírószilárdsága 61,8 ± 9,5 és 59,3 ± 5,8 N/mm². A ragasztóanyagok hőállósága 200 °C. A gyakorlati alkalmazás során a keményfém ragasztásának a következő előnyei jelentkeztek a hagyományos, forrasztásos vagy hegesztéses eljárással szemben:

- egyszerűsödik a szerszám gyártástechnológiája, mivel néhány részművelet elmarad,
- elmaradnak az illesztés környezetében a hőhatás okozta hibák, nem jelentkeznek a mikrorepedések, nem csökken a befogószár keménysége a forrasztási helyen, nem ég ki annak az anyaga a hegesztés során,
- a betétek anyaga megőrzi eredeti szerkezetét és mechanikai tulajdonságait,
- a tartószárak többször felhasználhatók.

A forrasztás és hegesztés ragasztással történő kiváltása javítja az így gyártott szerszámok minőségét, 1,2 ... 2-szeresére növeli az előírt forgácsolási sebességeken elérhető szerszámélettartamot.

Személyi robotok

A személyi számítógépek iránti érdeklődés fellendülése után most a „robothullám" került a figyelem középpontjába. Az USA-ban az R. B. Roboter Corp. kezdte el gyártani az első fürge „emberkéket". Az acélból és elektronikából összeállított „lény" több mint túlméretezett számítógépes játék, mert a még ez évben piacra kerülő robot a személyi számítógép minden jó tulajdonságával is rendelkezik. A költségvetés megtervezésétől egészen a határidő ellenőrzéséig minden műveletet gyorsan elvégez. A Toby névre keresztelt robot zenél, önállóan mozog, sőt BASIC nyelven programozható, egyidejűleg személyi számítógépként is használható.

A személyi robot szenzoraival tájékozódik az akadályok között. Teljesíthet az otthonokban biztonsági feladatokat, tartalmazhat füstjelzőt, mozgásjelzőt (pl. betörési kísérlet megakadályozására) és egy kart pl. virágok öntözésére.

Ahogy az autó megváltoztatta a környezetet, ugyanígy várható, hogy a személyi robot átalakítja a háztartásokat, hiszen a készülékek nem úgy „járnak", mint az emberek, hanem görgőkön közlekednek. A robotot vezérlő személyi számítógép agya a Toby robotban INS 8073 típusú mikroprocesszor.

ISKOLASZÁMÍTÓGÉP-PROGRAMOK

FIZIKAI FELADATOK

Sok szép fizikai feladat van, amelyekkel a középiskolában a kellő matematikai eszközök hiánya miatt régebben nem foglalkozhattunk. A számítógépek megjelenése azonban új helyzetet teremtett ezen a téren is. Sok feladat algoritmikus módszerrel jó közelítéssel számítható, így a középiskolában, sőt már az általános iskola felső tagozatában is megoldható. A következőkben ezekből közlünk néhányat.

A számítógépek egyelőre nem ismerik a görög ábécét. Az iskola-számítógéppel a kis- és a nagybetűk vegyes használata is körülményes. Ezért a jelölésekben kissé eltérünk a fizikában megszokott jelölési módtól. A fizikai mennyiségek jelölésében általában nagybetűket használunk, a görög betűk helyett pedig a megfelelő latin, illetve magyar betűket. Nézzük ezen megjegyzések után az első feladatot.

A légnyomás számítása

A levegő sűrűsége a föld felszínén $R_0=1{,}293$ kg/m^3, a légnyomás pedig $P_0=10^5$ Pa. Felfelé haladva mindkettő csökken. Mekkora a légnyomás pl. 5500 vagy 11 000 m magasságban, ha a levegő egyéb adatai (pl. hőmérséklet, páratartalom stb.) közben változatlanok maradnak?

Folyadék esetén egyszerűen a hidrostatikai nyomás ismert törvényét alkalmaznánk. Gázok esetén azonban a nyomással együtt a sűrűség is rohamosan csökken. Így nagyobb magasságok esetén más módszert kell keresnünk. Algoritmikus eljárásunk a következő. A kérdezett H1 magasságot nem túlságosan nagy DH-szakaszokra bontjuk. Ezeken belül a sűrűséget állandónak vesszük és így számolunk. Az ebből származó pontatlanság nyilván annál kisebb, minél kisebbek a DH-szakaszok.

A légnyomás számértéke egyenlő az 1 m^2 terület feletti levegőoszlop tömegével. Egyszerűen belátható az, hogy DH m-rel magasabban a légnyomás annyival kisebb (DP), mint a DH m magasságú, egységnyi keresztmetszetű levegőoszlop tömegének számértéke:

$$DP = 1 \cdot DH \cdot R \cdot G$$

ahol R a levegő sűrűsége, G pedig a nehézségi gyorsulás, vagy ahogy az újabb tankönyvek nevezik: gravitációs térerősség. Ez a szóba jöhető magasságokon belül állandónak tekinthető. Így az új légnyomás:

$$P_{új} = P_{előző} - DP$$

Az új magasságban a levegő új sűrűsége a Boyle—Mariotte-törvényből az új nyomás (P), a kezdeti nyomás (P_0) és a kezdeti sűrűség (R_0) felhasználásával kiszámítható:

$$\frac{R}{R_0}=\frac{P}{P_0} \longrightarrow R_{új}=\frac{P}{P_0}\cdot R_0$$

Az új sűrűségből az újabb nyomáscsökkenés és nyomás, majd az újabb sűrűség számítható. És így tovább, amíg a kívánt H1 magasságot el nem értük.

Az 1. program 50—90. soraiban találjuk a számítógépes BASIC-program lényegét. A többi „csak" a feladat — egyáltalán nem lényegtelen — „felöltöztetését" szolgálja. Mint a próbafuttatásokból látjuk, a felsőbb matematikával kapható nyomásértékeket is kiszámíttatjuk és kiíratjuk gépünkkel, összehasonlítás céljából. A gyakorlati életben ez a „pontos" érték egyáltalán nem jelentős, mert a tényleges fenti légnyomás annyi mindentől függ, hogy elvi pontossággal úgysem számítható.

A 200—280-as sorokban tulajdonképpen új szempont szerint dolgoztuk fel a feladatot. Szemléletesen is bemutatjuk vele, hogy a légnyomás a magassággal exponenciálisan csökken. A 110-es és a 250-es sorok azonosak. Próbáljuk őket felírni a fizika-, illetve a matematikaórán megszokott alakban. Egyébként a formula bármely alapfokú egyetemi fizika tankönyvben megtalálható.

A fenti fizikai témájú programot tökéletes „oktató programmá" alakíthatjuk, ha még egy ugyanilyen hosszúságú bevezető programrésszel is ellátjuk. Ebbe vegyünk be pl. a képernyő bal oldalára egy ábrát, a jobb oldalára pedig a feladat és az elvi megoldás szövegének a lényegét. Ha a szöveg számára a hely kevésnek bizonyulna, akkor áttérhetünk egy egész újabb oldalra is. Így programunk külön magyarázó szöveg nélkül is kerek egész, önálló tanulásra is felhasználható lesz. Ennek a kiegészítésnek minden gondját, de örömét is a kedves olvasónak hagyjuk meg.

Az asztalon pattogó labda

Az előzőnél lényegesen nehezebb feladat a következő. Egy vízszintes asztal fölött M m magasságban vízszintes irányban VV m/s-os kezdősebességgel eldobunk egy rugalmas labdát. A pattanási együttható (C) egynél kisebb szám. Ez azt jelenti, hogy a labda az asztalról kisebb sebességgel pattan vissza, mint ahogyan oda érkezett. Számítsuk ki kis DT-időközökben a labda X, Y koordinátáit, sebességének függőleges összetevőjét, Z-t. Számítsuk ki továbbá az N számú pattanás alatt eltelt időt, az ezalatt vízszintes irányban megtett utat. A koordinátarendszer origója az asztalon van, a labda indulási helye alatti pontban. Végül szimuláljuk a labda pattogását, vagyis rajzoltassuk fel valahogyan a képernyőre a labda pályáját.

A feladat megoldásának programját (2. program) némi átalakítással Byron S. Gottfried: Programming with BASIC c. könyvéből vettük, de megértése sem könnyű feladat. A feladat megoldása igen szellemes, futtatása látványos. Nézzük most a program egyes részeit.

A 20—100 sorokban a kezdő adatok bekérése és a változókba való elhelyezése történik meg. Legyenek ezek pl.: H = 0,6 m, VV = 0,4 m/s, N = 3, C = 0,8 és DT = 0,05 s. Ezekből az idő, a helykoordináták és a sebesség-összetevők kezdő értékei már következnek.

Az új adatok számítása a 120—230-as sorokban történik a 450—500-as sorokban levő szubrutin felhasználásával. Az új adatok kiszámításakor a program két, illetve három esetet különböztet meg aszerint, hogy a DT-időtartam végén a labda még az asztal fölött van-e — ez az átlagos eset —, vagy a labda már az asztal alatt lenne, ez az ütközések esete. Az új adatok számítása az átlagos esetben a 130-as sorban, illetve a szubrutinban történik. A közbülső ütközések esetén a szubrutinban történő számítások után egyrészt korrekcióra van szükség, hogy a DT helyett a valóságos időközt, a D1-et, a valóságos magasságot és a valóságos sebességet kapjuk meg, másrészt, hogy a pattanás miatt beálló sebességveszteséget és a sebesség irányának a megfordulását is figyelembe vegyük. Mindez a 160—190. sorokban történik meg. Az utolsó ütközés alkalmával a program 220-as sora kiszámítja a tényleges időközt, és ennek felhasználásával számíttatja ki a szubrutinnal az utolsó intervallum többi adatát. A három számítási eset szétválasztását a program 140—150. sorai végzik.

A számított adatok kinyomtatása a program 250—300. sorainak a feladata. A gép a számított adatokat négy vektorban tárolta (lásd a 20-as sort). Ez növeli a felhasznált memóriaterületet, de gyorsabbá teszi a program futását. A külső nyomtató memóriájába ugyanis gyorsan kiküldhetők az adatok és a nyomtatás ideje alatt a gép dolgozhat a program további részén.

A program különálló része a pattogás szimulálása, vagyis a labda pályájának a képernyőre, vagy ha lehetőség van rá, papírra történő megrajzolása (320—440. sorok). Az ábrázoláshoz a nagy gépeknél szokásos TAB-utasítást használjuk. Ennek következménye a koordinátarendszer negatív irányba történő 90°-os elfordítása. Ez papíron való ábrázolás esetén nem zavar, mert a papírost egyszerűen 90°-kal visszafelé fordítva nézzük. A képernyőn való ábrázoláskor erre nincs módunk, de a képernyő futását tetszés szerinti helyen megállíthatjuk.

A program 320—350-es sorai a vízszintes tengelyt rajzolják meg. A függőleges X tengely megrajzolása a függvény megfelelő pontjaival egyidőben történik. A 300-as sorban az Y (1), vagyis a legnagyobb függvényértékkel való osztással azt érjük el, hogy a grafikon nagy számok esetén sem fut ki a képernyőről.

Második fizikai programunk is sok tanulásra és kísérletezésre nyújt lehetőséget. Az ütközések alkalmával szükséges korrekciókat aligha értjük meg első olvasásra. Készítsünk felnagyított, konkrét adatokkal ellátott rajzot az eseményről. Így érthetjük csak meg a felhasznált matematikai formulákat. Ezek nélkül a korrekciók nélkül a feladat nem oldható meg. Változtassuk az ajánlott kezdő adatok nagyságát és figyeljük meg a változtatások hatását. Ha a pattanások számát 3-nál nagyobbra vesszük, vagy a lépéshosszt 0,05 s-nál lényegesen kisebbre, akkor esetleg növelnünk kell a tárolóvektorok méretét. A számítógéppel bizonyos értelemben kísérletezni is lehet. De ha pl. a pattanási együtthatót 1-nél nagyobbnak vesszük, akkor a visszapattanási sebesség egyre nő. Ez a valóságban nincs így.

Kovács Mihály

```
10 CLS:PRINT@70,"      A LEGNYOMÁS SZÁMÍTÁSA"
15 '                   **********************
20 INPUT"MAGASSAGNOVELES ES A KIVANT MAGASSAG m-BEN";DH,H1
30 P0=100000: R0=1.293: G=9.81: P=P0: R=R0: PRINT
40 PRINT" A LEGNYOMAS IDELENT:                    ";P;"PA"
50 FOR H=0 TO H1 STEP DH
60     DP = R*G*DH
70     P = P - DP
80     R = R0*P/P0
90 NEXT H
100 PRINT H1;"m MAGASBAN";DH;"m-ENKENT SZAMITVA*:";P;"PA"
110 PP=P0*EXP(-H1*R0*G/P0)
120 PRINT" MAGASABB MATEMATIKAVAL SZAMITVA        ";PP;"PA"
130 IF INKEY$=""THEN130
195 CLS
200 PRINT" A LEGNYOMAS ABRAZOLASA A MAGASSAG FUGGVENYEBEN"
205 '      ***************************************
210 FOR I=6 TO 46: SET(2,I): NEXT I
220 FOR I=2 TO 110: SET(I,46): NEXT I
230 PRINT"NYOMAS:": PRINT@1015,"MAGASSAG";
235 PRINT@131,"100000 PA";:SET(12,47):PRINT@902,"5500 m";
240 FOR H=0 TO 55000 STEP 2750
250     PP=100000*EXP(-H*1.293*9.81/100000)
260     SET(H/550+3,-PP/2500+46)
270 NEXT H
280 IF INKEY$=""THEN 280


RUN


 A LEGNYOMAS IDELENT:                    100000 PA
 5500 m MAGASBAN 500 m-ENKENT SZAMITVA*: 45554.3 PA
 MAGASABB MATEMATIKAVAL SZAMITVA         49776 PA

 A LEGNYOMAS IDELENT:                    100000 PA
 5500 m MAGASBAN 100 m-ENKENT SZAMITVA*: 48925.8 PA
 MAGASABB MATEMATIKAVAL SZAMITVA         49776 PA

 A LEGNYOMAS IDELENT:                    100000 PA
 5500 m MAGASBAN 10 m-ENKENT SZAMITVA*:  49690.8 PA
 MAGASABB MATEMATIKAVAL SZAMITVA         49776 PA
```

1. program

```
10 CLS:REM   A PATTOGÓ LABDA
             ===============
20 DIMX(100),Y(100),Z(100),T(100)
30 INPUT"A KEZDETI MAGASSAGA (m-BEN)";H
40 IF H=0 THEN END
50 INPUT"A VIZSZINTES SEBESSEG (m/sec)";VV
60 INPUT"A PATTANASOK SZAMA";N
70 INPUT"PATTANASI EGYUTTHATO";C
80 INPUT"IDOKOZOK HOSSZA (sec)";DT
90 T(1)=X(1)=Z(1)=0:B=0:PRINT
100 Y(1)=H: G=9.81
110 '
120 FOR I=1 TO 99
130   GOSUB 450
140   IF Y(I+1)>0 THEN 210
150   IF B=N THEN 220
160   D1=DT*Y(1)/(Y(1)-Y(I+1))
170   Z1=-C*(Z(I)-G*D1)
180   Z(I+1)=Z1-G*(DT-D1)
190   Y(I+1)=.5*(Z1+Z(I+1))*(DT-D1)
200   B=B+1
210 NEXT I: GOTO 250
220 DT=DT*Y(I)/(Y(I)-Y(I+1))
230 GOSUB 450
240 '
250 I1=I+1: T1=T(I1): X1=X(I1)
260 PRINT"VIZSZINTESEN MEGTETT UT=";X1;"M"
270 PRINT"ELTELT IDO=";T1;"SEC": PRINT
280 FOR I=1 TO I1
290   PRINT"T=";T(I),"X=";X(I),"Y=";Y(I),"Z=";Z(I)
300 NEXT I: PRINT
310 '
320 PRINT"A FELADAT GRAFIKUS MEGOLDASA": PRINT
330 FOR J=0 TO 66
340   PRINTTAB(J);".";
350 NEXT J
360 PRINTTAB(61)"*": PRINT"."
370 FOR I=2 TO I1
380 . J=INT(61*Y(I)/Y(1))
390   IF J=0 THEN 410
400   PRINT".";TAB(J)"*": GOTO 420
410   PRINT"*"
420 PRINT"."
430 NEXT I
440 GOTO 30
450       REM       *** SZUBRUTIN ***
460       T(I+1)=T(I)+DT
470       X(I+1)=X(I)+VV*DT
480       Z(I+1)=Z(I)-G*DT
490       Y(I+1)=Y(I)+.5*(Z(I+1)+Z(I))*DT
500       RETURN
```

2. program

TECHNISKOLA

µC-Iskolaszámítógép-programok

(2) Már az első fizikai feladatunknál is láttuk, hogy a tulajdonképpeni számítást végző programrész viszonylag egyszerű és rövid volt. A megoldás megértéséhez azonban egyrészt szabatosan meg kell fogalmaznunk a szükséges fizikai fogalmakat, másrészt igyekeznünk kell a programot jól és könnyen kezelhetővé, „barátságossá" is tenni. Ehhez viszont alapos számítástechnikai felkészültség is szükséges. A mostani két feladatunk is ezen elvek alkalmazására lesz jó példa.

Emelési munka

Mekkora munka árán juttathatunk fel 1 kg tömeget a Föld felszínéről tetszés szerinti H km magasságra, pl. földsugárnyira vagy gyakorlatilag a végtelenbe?

Tudjuk, hogy ha 1 kg tömeget a Föld felszínéről 1 m-rel magasabbra emelünk, akkor a végzett munka: W=1 kg×g_0×1 m vagyis 9,81 joule, ahol g_0 a Föld felszínén a gravitációs térerősség vagy — ahogy régebben mondták — a nehézségi gyorsulás. Ha pedig az 1 kg-os testet a Föld felszínéről 1 km-rel visszük magasabbra, akkor a végzett munka kb. 10 kjoule. Hát ez a számítás igazán egyszerű. Kár, hogy nagyobb magasságok esetén nem használható, mert a gravitációs térerősség nem állandó, hanem a Föld középpontjától számított távolság négyzetével fordított arányban csökken.

$$\frac{g}{g_0} = \frac{R^2}{(R+H)^2} \longrightarrow g = g_0 \frac{R^2}{(R+H)^2}$$

A képletben R a Föld sugara, g pedig a gravitációs térerősség a Föld felszínétől számított H magasságban. Pl. 1000 km magasságban a gravitációs térerősség már csak 75%-a a Föld felszínén mért térerősségnek, 6370 km, tehát földsugárnyi „magasságban" pedig már csak 25%-a, vagyis egynegyede.

A gravitációs térerőt az előző módszerrel 10 km magasságban számítva azt kapjuk, hogy az csak 0,3%-kal kisebb, mint a Föld felszínén. Ebből adódik az ötlet, hogy ilyen lépésekben haladva számítsuk az emelési munkát, és ekkor az elvi értéket jól megközelítjük. Mai űrlaboratóriumaink azonban szinte már a „végtelenbe" is eljutnak. Szeretnénk az emelési munkát ezek esetére is kiszámítani. Egy elvi és egy gyakorlati akadálya van ennek a számításnak. Lépésenként haladva a végtelenbe soha sem juthatunk el. Ez azonban nem is szükséges. Pl. 100 millió km távolságban a Föld gravitációs térerőssége már olyan kicsi (4×10^{-9} g_0), hogy az innen számított további munka már nyugodtan elhanyagolható. A másik nehézség az, hogy ha 10^8 km-ig 10 km-es lépésekben haladnánk, ez 10^7 lépést jelentene. Ez még gyors számítógépek esetén is fölöslegesen sok gépidőt venne igénybe. Ezt a nehézséget is leküzdhetjük, ha a következőképpen járunk el.

Legyenek lépéseink kb. egy földsugárnyi távolságig 10 km-esek, 6400 km-től 10 000 km-ig 100 km-esek, 100 000 km-ig 1000 km-esek, és így tovább. Az utolsó szakaszban, 10^7 km-től 10^8 km-ig 10^6 km-esek. Így kb. 1000 lépésben az emelési munkát jó közelítéssel gyakorlatilag a végtelenig kiszámíthatjuk.

A számítás programját az 1. programon látjuk a próbafuttatással együtt. A számítási algoritmus a 100—150-es sorokban található. A kinyomtatási és a léptékváltási helyeket a 60—90-es sorokban levő feltételek szabják meg. A 160—170-es sorokban két nyomtató utasításra azért van szükség, mert az első és a második nyomtatásban nincs lépésváltás, a többinél pedig van.

A fenti egyszerű programból, hogyan készíthetünk még értékesebb fizikai oktató programot pl. a gimnáziumi II. osztályos Fizika tankönyv megfelelő fejezeteinek a tanításához? (Fizika II., 61., 113. és 177. lapjai.) Bevezetőnek igen jól fest egy filmszerű kép, amint egy 1 kg-os tömeg a Föld felszínéről emelkedik felfelé, és eltűnik a képernyő tetején. Ilyen programot már megadtunk az Apolló űrhajó indítása c. szimulációnk elején. Egészítsük ki programunkat néhány matematikai vonatkozással is. Pl. a Fizika II. 170. ábrájához hasonlóan rajzoltassuk fel a számítógéppel a gravitációs erő függvényét, a másodfokú (!) hiperbola képét, és mutassunk rá arra, hogy — matematikai szemszögből nézve — feladatunkban, ezen másodfokú hiperbola alatti területet számítottuk ki „körülírható téglalapokkal" való közelítést használva. A Fizika II. 171. ábráját a mi 1. ábránk módjára rajzoljuk fel, mert ez az általunk követett gondolatmenethez alkalmazkodik. A „végtelenbe" való eljutáshoz tetemes nagyság, de mégis véges energia szükséges. Ezért tudtunk űrkutató rakétát küldeni naprendszerünkön túlra is. Igaz, hogy a legközelebbi csillag közelébe kb. 800 000 év múlva ér el.

Fizikai szempontból érdekes eredmény még, hogy ha a testet a Földtől földsugárnyira elvittük, akkor a végtelenbe való elvitelhez szükséges munka felét már elvégeztük. Az elvitelkor végzett munka a test helyzeti energiáját növeli. A maximális helyzeti energiából kiszámítható, hogy elvileg mekkora sebességgel kellene a testet a Földről függőleges irányba elindítani, hogy oda többé ne térjen vissza (a második kozmikus sebesség: 11,2 km/s).

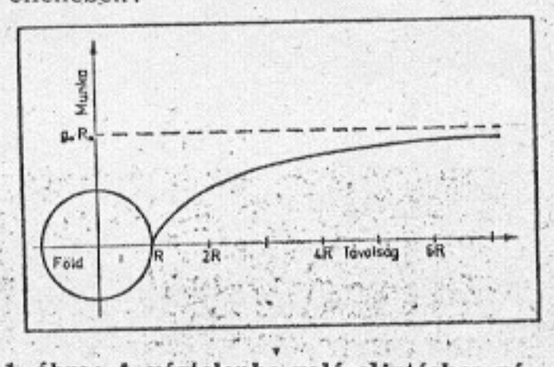
Egyszerű feladatunk megoldása gyakorlatilag és elvileg is fontos eredményekkel szolgált. Beláttuk, hogy egy test felvitele a Földről tetszés szerinti magasságra véges munkával megoldható. Ezt a munkát algoritmikus eljárással elemi úton is számítani tudjuk. Matematikailag pedig azt láttuk be, hogy az

$$y = \frac{k^2}{(x+k)^2}$$

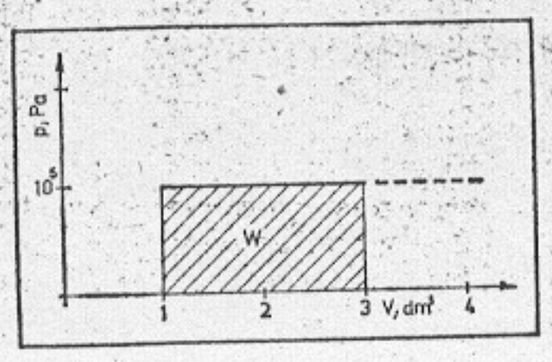
másodfokú hiperbola alatti, nulla és végtelen közötti terület véges. Pl k = 3 esetén 3 egység.

Tágulási munka

Van 1 dm³ 0 °C-os levegőnk, amelynek nyomása 10^5 Pa. Felmelegítjük 546 °C-ra, a térfogata pedig 3 dm³ lesz. Mekkora munkavégzés történt a külső légnyomás ellenében?



1. ábra: A végtelenbe való eljutáshoz véges energia szükséges



2. ábra: Munkavégzés a külső légnyomás ellenében, amennyiben a gáz nyomása melegítés közben állandó

a) Amennyiben a gáz nyomása melegítés közben állandó maradt, akkor könnyű a dolgunk (2. ábra). Tudjuk, hogy a táguláskor végzett munka a nyomás és a térfogatváltozás szorzata: W=p×dV. A p—V diagram alapján ez éppen a „görbe" 1—3 szakasza alatti terület mértékszáma:

$$W = p \cdot dV = 10^5 \frac{N}{m^2} \cdot 2 \cdot 10^{-3} m^3 = 200 \text{ joule}$$

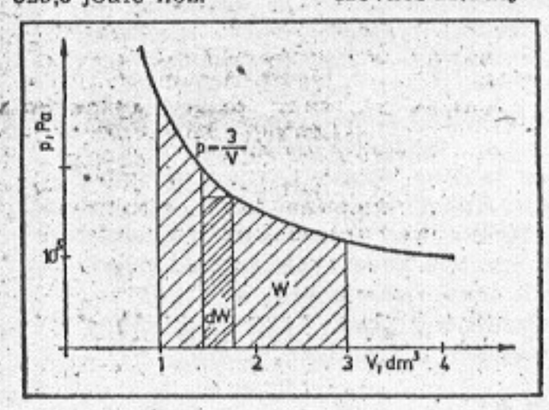
b) Nehezebb a helyzetünk, ha a gázt először állandó térfogaton melegítjük fel 0 °C-ról 546 °C-ra, majd állandó energiaközléssel ezen a hőmérsékleten tartva hagyjuk kitágulni és munkát végezni (3. ábra). A munka ekkor is a p—V görbe alatti terület mértékszáma a V=1-től a V=3-ig. A mindenkori nyomást a Boyle-Mariotte törvény alapján kapjuk meg:

$$p \cdot V = 3 \cdot 1 \longrightarrow p = \frac{3}{V}$$

Bontsuk keskeny „beírt" téglalapokra a kiszámítandó területet, vagyis egy-egy keskeny téglalapon belül tekintsük a nyomást állandónak. Ha elég keskenyek a téglalapok, akkor jó közelítést kapunk a területre, vagyis a vele arányos munkára.

A 2. program sokkal többet ad, mint a feladat egyszerű megoldását. Komplett számítógépes oktatási programnak tekinthető, amely a fizika és a matematika órákon egyaránt használható a középiskolákban. A 40-es sorban kéri a „felbontást", vagyis hogy a V=1-től V=3-ig terjedő területet hány téglalapra akarjuk felbontani. Az 50—140. sorokban ábrázolja és „fehérre festi" a választott téglalapok területét. A 85. sorban „mellékesen" kiszámolja a téglalapok magasságait és azok összegét (Y, illetve S). A 145. és a 150. sorban pedig kiszámítja és kinyomtatja a végzett munkára kapott közelítő értéket. Bármely gomb lenyomására a program újra fut és kéri az új felbontást. Pl. 4, 8, 16, 50 felbontást beadva az értékek egyre közelednek az elvileg kapható 329,5 joule-hoz.

Kovács Mihály



3. ábra: Munkavégzés abban az esetben, ha a gázt állandó térfogaton melegítjük fel, majd azon a hőmérsékleten tartva hagyjuk kitágulni

```
10 '      EMELESI MUNKA
          **************
20 CLS:DH=10:G0=9.81:R0=6370
30 G = G0: R = R0
40 PRINT"MAGASSAG              MUNKA"
50 PRINT"    KM                 KJ":PRINT
60 IF H>1E8 THEN 180
70 IF H=1E8 OR H=1E7 OR H=1E6
    OR H=1E5 OR H=1E4 THEN160
80 IF H=6400 THEN DH=DH*10:GOTO100
90 IF H=6370 OR H=1000 THEN170
100 H = H + DH
110 R = R + DH
120 G = G0*R0[2/R[2
130 DW = G*DH
140 W = W + DW
150 GOTO60
160 PRINTH,W:DH=10*DH:GOTO100
170 PRINTH,W:GOTO100
180 PRINT:PRINT"A 'VEGTELEN'-BE VALO
ELVITELHEZ SZUKSEGES MUNKA:
      G0*R0 = 62489 KJ
ANNYI, MINTHA FOLDSUGARNYI MELY
KUTBOL EMELTUK VOLNA KI A TESTET"

RUN
MAGASSAG          MUNKA
   KM               KJ

 1000             8466.54
 6370             31208.1
 10000            38088.9
 100000           57953
 1E+06            61131.3
 1E+07            61468.2
 1E+08            61502.1

A 'VEGTELEN'-BE VALO
ELVITELHEZ SZUKSEGES MUNKA:
      G0*R0 = 62489 KJ
ANNYI, MINTHA FOLDSUGARNYI MELY
KUTBOL EMELTUK VOLNA KI A TESTET
```

1. program

```
10 CLS:PRINT"       TAGULASI MUNKA"
20     PRINT"       ***************".
30 FOR X=0 TO 1000: NEXT X
40 CLS:INPUT"  FELBONTAS";N
50 FOR Y=0 TO 47: SET(0,Y): SET(1,Y): NEXT
60 FOR X=0 TO 127: SET(X,47): NEXT
70 S=0
80 FORI=1 TO N
85    X=1+I*2/N: Y=3/X: S=S+Y
90    FOR X1=40*X TO 40*(X-2/N) STEP-1
100       FOR Y1=47-10*Y TO 47
110         SET(X1,Y1)
120       NEXT Y1
130    NEXT X1
140 NEXT I
145 W=S*2/N
150 PRINT@86,"A VEGZETT MUNKA:";W*100;"J";
160 A$=INKEY$:IF A$=""THEN160 ELSE RUN
200 CLS:PRINT"    A TERFOGAT NOVELESEVEL
    A VEGZETT MUNKA EGYRE NOVEKSZIK:"
210 IF INKEY$="" THEN 210
220 FOR Y=5 TO 47: SET(3,Y): SET(4,Y):NEXT
230 FOR X=0 TO 120: SET(X,44): NEXT:SET(2,28)
232 SET(2,27): PRINT@9*64+3,"329,6 J";
234 PRINT@15*64+7,"1";: PRINT@15*64+17,"3";
235 PRINT@128,"W";: PRINT@1022,"V";: SET(14,45)
236 FOR K=1 TO 10: SET(K*10+4,45): NEXT K
240 FOR X=1 TO 10 STEP .4
250   SET(10*X+4,-15*LOG(X)+44)
260 NEXT X
270 IF INKEY$="" THEN 270
280 CLS:PRINT@7*64,"A VEGZETT MUNKA ALLANDO NOVEKEDESET"
290 PRINT@9*64,"   AZ ALLANDO HOKOZLES MAGYARAZZA"
300 IF INKEY$="" THEN 300
```

2. program

TECHNISKOLA

µC-Iskolaszámítógép-programok

Műszaki feladatok

Műszaki feladatok megoldásának segítésére, sőt a feladatok végrehajtására is mind eredményesebben használják a számítógépet, vagy egyszerűbb esetekben a számítógép leglényegesebb építőelemét, a mikroprocesszort. Az első holdutazást számítógépen „játszották végig", szimulálták szinte végtelen sokszor, és így választották ki a legmegfelelőbbnek látszó változatot. A műholdak indítása, az automata űrszondák vezérlése is számítógépekkel történik. A feladat eredményes végrehajtásához a jó szoftver, program legalább annyira szükséges, mint a jó számítógép vagy mikroprocesszor. A következőkben néhány műszaki célokat szolgáló egyszerű programot mutatunk be.

Termosztát működésének szimulálása

A termosztát a legegyszerűbb esetben egy olyan készülék, amelyben egy zárt edény belső hőmérsékletét a környezetnél magasabb, közel állandó hőmérsékleten tudjuk tartani. A hűtőgéphez hasonlít működésében, de bizonyos szempontból nézve annak éppen a fordítottja. Működtetéséhez csak egy fűtőtest és megfelelő szabályozóberendezés szükséges. Mivel a környezeténél magasabb hőmérsékletű, ezért a leggondosabb hőszigetelés ellenére is állandóan energiát ad le a környezetének.

A környezetnek leadott teljesítmény nyilván arányos az edény felületével (F), a belső és a külső hőmérséklet különbségével (HT—HK), függ az edényfal anyagának hővezetési tényezőjétől és fordítva arányos az edény falvastagságával (D):

$$P1 = \frac{F \cdot (HT - HK) \cdot L}{D}$$

Ha a fűtőtest teljesítménye P, akkor DT idő alatt a termosztát belső hőmérsékletének a változása:

$$DH = \frac{(P - P1) \cdot DT}{M \cdot Cp}$$

ahol a nevező az edény belsejében levő anyagok tömegének és fajhőjének a szorzata.

A DH hőmérsékletváltozás ismeretében a régi hőmérsékletből az új már számítható. Az eljárás egymás utáni ismétlésével a termosztát működése követhető.

Programunk lényegében az ismertetett eljárást követi és a felmelegedés-lehűlés folyamatát és a hőmérséklet-szabályozáskor lejátszódó eseményeket szemlélteti. A Praxis der Naturwissenschaften, Physik c. didaktikai lap 1984/4. számának egyik cikke alapján készült.

A berendezésen még további, de a lényeget nem érintő egyszerűsítéseket végzünk. Az edény kocka alakú, amelynek az éle A. Belsejében az elhanyagolható tömegű érzékelőn és elektromos melegítőn kívül csak levegő van. A mintavétel időközeinek a hosszát (DT) általában 1 másodpercnek vesszük. A belső hőmérséklet ingadozása a jobb láthatóság kedvéért ±2 K. A berendezés vázlatos elvi rajzát a 2. ábra felső részén látjuk. Az ábra bal oldali részén van az edény az E érzékelővel és a fűtőtesttel. Az U feszültségű áramforrás áramkörét külső kapcsoló zárja és nyitja. A tényleges és az előírt hőmérséklet összevetésével a komparátor dönti el, hogy az áramkört zárni vagy nyitni kell.

A program annyira áttekinthető, hogy nem sok magyarázat szükséges hozzá. Az adatok bekérését a 30—90. sorok végzik. A 100—110. sorok döntik el választásunk alapján, hogy a két szimuláció közül melyik következzék. A felmelegedés-lehűlés szimulálása a 120—220-as sorok feladata. A hőmérséklet-szabályozást pedig a 230—520. sorok mutatják be. A 140. és a 330. sorokban a programok lényege az egyszerűség céljából ismétlődik.

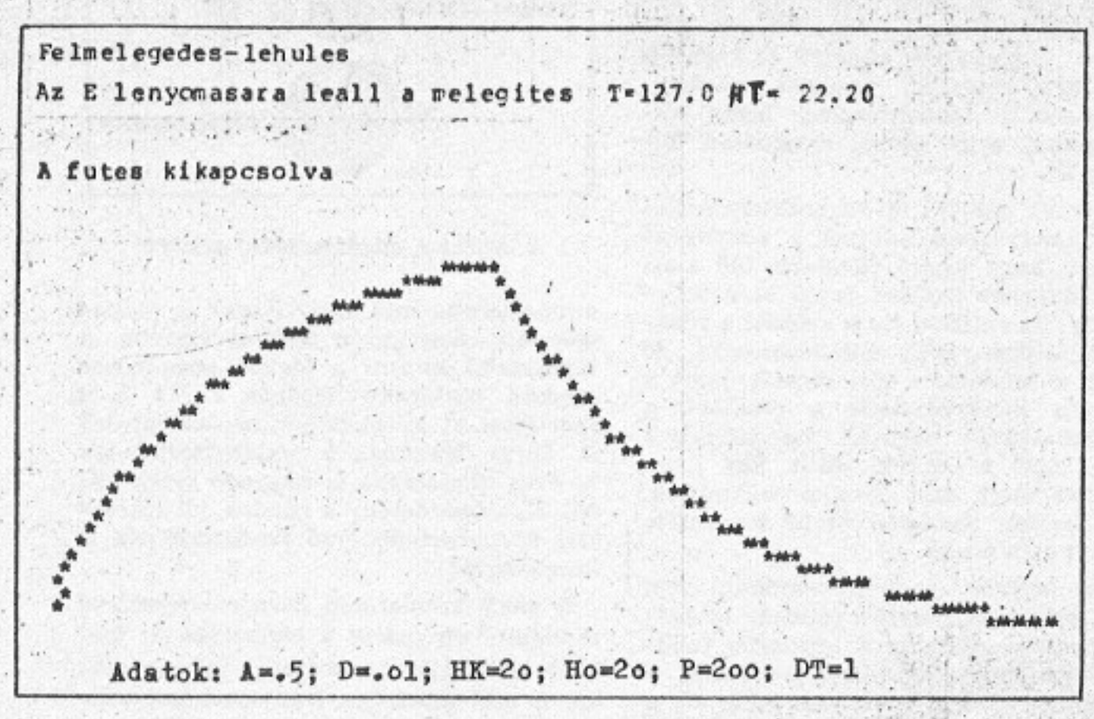
A program futtatása

A feladatban annyi az általunk megadható változó, hogy a program megvédése a lehetetlen vagy a nem látványos adatok bevitele ellen igen nagy apparátust követelne. Ezért inkább az 1. és 2. ábrán megadtuk azokat az adatokat, amelyekkel mi futtatjuk a programot. Eleinte ezekhez közelálló adatokkal próbálkozzunk. De később bátran „kísérletezzünk" más adatokkal is. Érdemes előre jól átgondolni, hogy mit kell kapnunk.

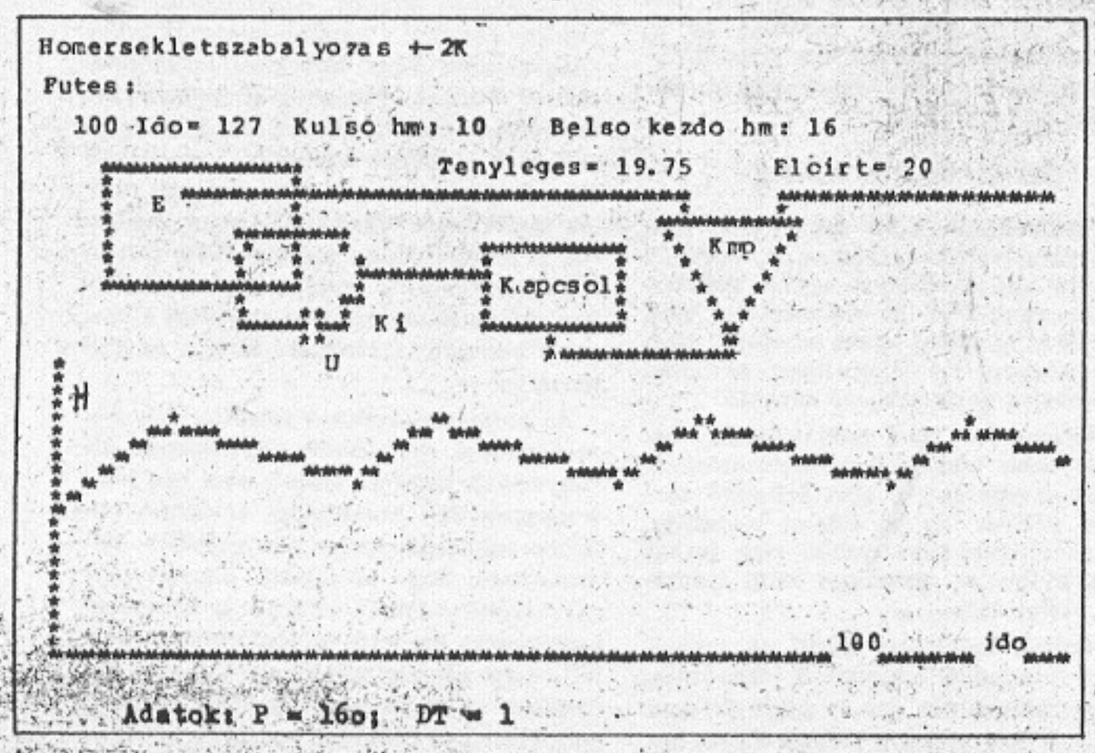
A melegedés-hűlés programrész futtatásakor bármikor és bármely nyomógomb lenyomásával áttérhetünk a melegedésről a lehűlésre. Az 1. ábrához hasonló képet kapunk. Az eltelt időt és a tényleges hőmérsékletet minden időköz végén kiírja a gép. A lehűlésre való áttéréskor nyomtatja ki a gép a „Fűtés kikapcsolva" feliratot. Rajzolási nehézségek miatt a gép nem áll le. Erről a 10-es sor hibavédelme gondoskodik.

A hőmérséklet-szabályozás választása esetén a gép először a modell működő rajzát vázolja fel, majd az idő-hőmérséklet tengelyeket, és csak azután indul meg a folyamat (2. ábra). Az idő és a tényleges hőmérséklet adatai minden időköz végén felfrissülnek. Az U feszültséggel táplált fűtőáramkört a hőmérséklet megengedett alsó és felső határainak elérésekor zárja és nyitja a modell. Az előírt hőmérséklet és az E érzékelővel mért tényleges hőmérséklet adatai a komparátorba futnak be. Ez összehasonlítja a két adatot. A hőmérséklethatárok elérésekor jelet küld a kapcsolóba, és az átkapcsolás megtörténik. A program igen alkalmas a felsorolt szabályozástechnikai fogalmak megértésére.

Dr. Kovács Mihály



1. ábra: Felmelegedés és lehűlés



2. ábra: A termosztát vázlatos elvi rajza és egy program a működésre

```
10 CLS: ON ERROR GOTO 530
20 REM FELMELEGEDES-LEHULES
       HOMERSEKLET-SZABALYOZAS

30 INPUT "A kocka ele (m=ben)";A
40 INPUT"A kocka falvastagsaga (m-ben)";D: PRINT
50 PRINT"Afal stynoporbol van (.04 W/mK)"; L=.04
60 PRINT:INPUT "A kulso homerseklet ";HK
70 INPUT"A kezdeti belso homerseklet";H0
80 INPUT "A futesi teljesitmeny (W-okban)";P
90 INPUT"Lepeshossz";DT: T=0
100 INPUT "   Egyszeri felmelegites-lehules (1)
   Homersekletszabalyozas (2)"; M
110 ON M GOTO 120,230

120 CLS:PRINT"Felmelegedes-lehules"
130 PRINT"A U lenyomasara leall a melegites";:H1=H0
140 HT=H1+(P-6*A*A*L*(H1-HK)/D)*DT/(1293*A*A*A):T=T+DT
150 X=T: Y=47+HK-HT:SET(X,Y):PRINT@100,"T=";USING"###.#";T;
160  PRINT" HT=";USING"###.##";HT;
170 H1=HT
180 IF P=0 THEN GOTO 210
190 A$=INKEY$:IF A$="" THEN GOTO 140
200 PRINT@192,"A futes kikapcsolva";:P=0
210 B$=INKEY$:IF B$=""THEN GOTO 140 ELSE GOTO 100
220 GOTO 100

230 REM Homersekletszabalyozas
240 INPUT"Az eloirt ertek (0-25) ";HE: H1=H0
250 PRINT"A LEGN. ELTERES AZ ELOIRT ERTEKTOL 2 K LEHET!":MA=2
260 FOR R=1 TO 1600:NEXT:T=0
270 CLS:PRINT"Homersekletszabalyozas +-2K":PRINT"Futes:";
 P,"Kulso hm:";HK,"Belso kezdo hm:";H1
280 GOSUB 390
290 GOSUB 330
300 IF H1<HE-MA,PH=P
310 IF H1>HE+MA,PH=0
320 GOTO 290
330 HT=H1+(PH-6*A*A*L*(H1-HK)/D)*DT/(1293*A*A*A)
340 T=T+DT: H1=HT: X=T:Y=42+HK-H1:SET(X,Y)
350 PRINT@134,"Ido=";T;:PRINT@227,USING"###.##";H1;
360 IF PH<>0,PRINT@469,"Be";:SET(39,17):SET(39,18):SET(39,19)
370 IF PH=0,PRINT@469,"Ki";:RESET(39,17):RESET(39,18):
    RESET(39,19)
380 RETURN

390 REM rajzolas
400 FOR X=8 TO 32:SET(X,10):NEXT:FOR X=18 TO 81:SET(X,12):NEXT:
  FOR X=93 TO 127:SET(X,12):NEXT:FOR X=78 TO 95:SET(X,14):NEXT
410 SET(81,13):SET(93,13)
420 FOR X=25 TO 38:SET(X,15):NEXT:FOR X=56 TO 73:SET(X,16):NEXT
  :SET(26,16):FOR X=56 TO 73:SET(X,16):NEXT
430 FOR X=40 TO 56:SET(X,18):NEXT:FORX=8 TO 32:SET(X,19):NEXT:
  FOR X=25 TO 33:SET(X,22):NEXT:FOR X=35 TO 38:SET(X,22):NEXT
440 FOR X=56 TO 73:SET(X,22):NEXT:FORX=64TO87:SET(X,24):NEXT:
  SET(33,21):SET(35,21):SET(33,23):SET(35,23):SET(64,23):
  SET(86,23):SET(87,23)
450 FOR Y=10TO19:SET(8,Y):NEXT:FORY=17TO 22:SET(25,Y):NEXT:
  SET(38,16):FORY=10TO19:SET(32,Y):NEXT
460 FOR Y=20TO22:SET(38,Y):NEXT:FOR Y=17TO20:SET(40,Y):NEXT
470 SET(39,20):FOR Y=16 TO 22:SET(56,Y):SET(73,Y):NEXT:
  FOR N=0 TO 7:SET(79+N,15+N):SET(94-N,15+N):NEXT
480 PRINT@217,"Tenyleges=";:PRINT@238,"Eloirt=";
490 PRINT@263,"E";:PRINT@413,"Kapcsol";:PRINT@529,"U";:
  PRINT@245,HE;:PRINT@362,"Kmp";
500 FORX=0TO127:SET(X,47):NEXT:FOR Y=25 TO 47:SET(0,Y):NEXT
510 PRINT@1009,"100";:PRINT@1018,"ido";:PRINT@577,"H";
520 RETURN
530 IF INKEY$=" " THEN CLS ELSE 530
540 PRINT: PRINT"Uj szimulalas": RESUME 50
```

TECHNISKOLA

µC-Iskolaszámítógép-programok

Szimulációs program

Közel másfél éves programsorozatunk befejezéseként egy nevezetes műszaki feladat számítógépes szimulálását mutatjuk be. A Voyager—2 űrszondát 1977. augusztus 20-án indították el

A költséget és időt az csökkenti, hogy a bolygók a jól kiválasztott indulási időpont és irány következtében a mellettük elhaladó űrszonda sebességét jelentős mértékben növelik és haladási irányát célszerűen megváltoztatják. A következőkben annak a kb. két napnak az eseményeit szimuláljuk, amikor az űrszonda a Jupiter közelében elhaladt, sebessége jó 2 km/s-mal nőtt, sebességének az iránya pedig olyan lett, hogy két év múlva a Saturnus közelébe ért, hogy ott megismételje bravúrját.

A szondát a Naphoz rögzített koordinátarendszerben 42,2 km/s-os sebességgel kellett indítani. Az alkalmas indítási irány és hely következtében ebből 30 km/s-os sebességet a Földnek a Nap körüli sebessége. 0,4 km/s-ot pedig a Föld forgási sebessége adott. Így a hajtórakétáknak csak a fennmaradt 11,8 km/s-os sebességet kellett szolgáltatniok. A Jupiter közelébe érve a szonda sebessége már 18,74 km/s-ra csökkent a Naphoz rögzített koordinátarendszerben.

A 2. ábrán látjuk az 1. ábrához képest kb. 20-szoros nagyításban a Jupiter közelében lejátszódó eseményeket. Tegyük fel, hogy az AO-pontban levő szonda a Jupiter pályájára éppen merőleges irányban halad a JO-pontban levő Jupiter felé. Távolsága a Jupitertől 1,6 millió km. Vegyük fel a koordinátarendszer kezdőpontját a JO-pontban és az X tengelyt a Jupiter mozgásirányában. A Jupiter sebessége 13,6 km/s. A szonda pl. DT= 30 000 másodperc 8,33 óra idő alatt az A1 pontba jutna. De a Jupiter ez alatt az idő alatt kb. 0,4 millió km-rel jobbra, a J1 helyzetbe kerül. Így a szondát egyrészt gyorsítja, másrészt jobbra téríti. Így az a B1 helyzetbe kerül. A most következő szakaszon mutatjuk be, hogy hogyan lehetne a pályát megszerkeszteni, de a programban felhasznált számítás és rajzolás gondolatmenete is ugyanez.

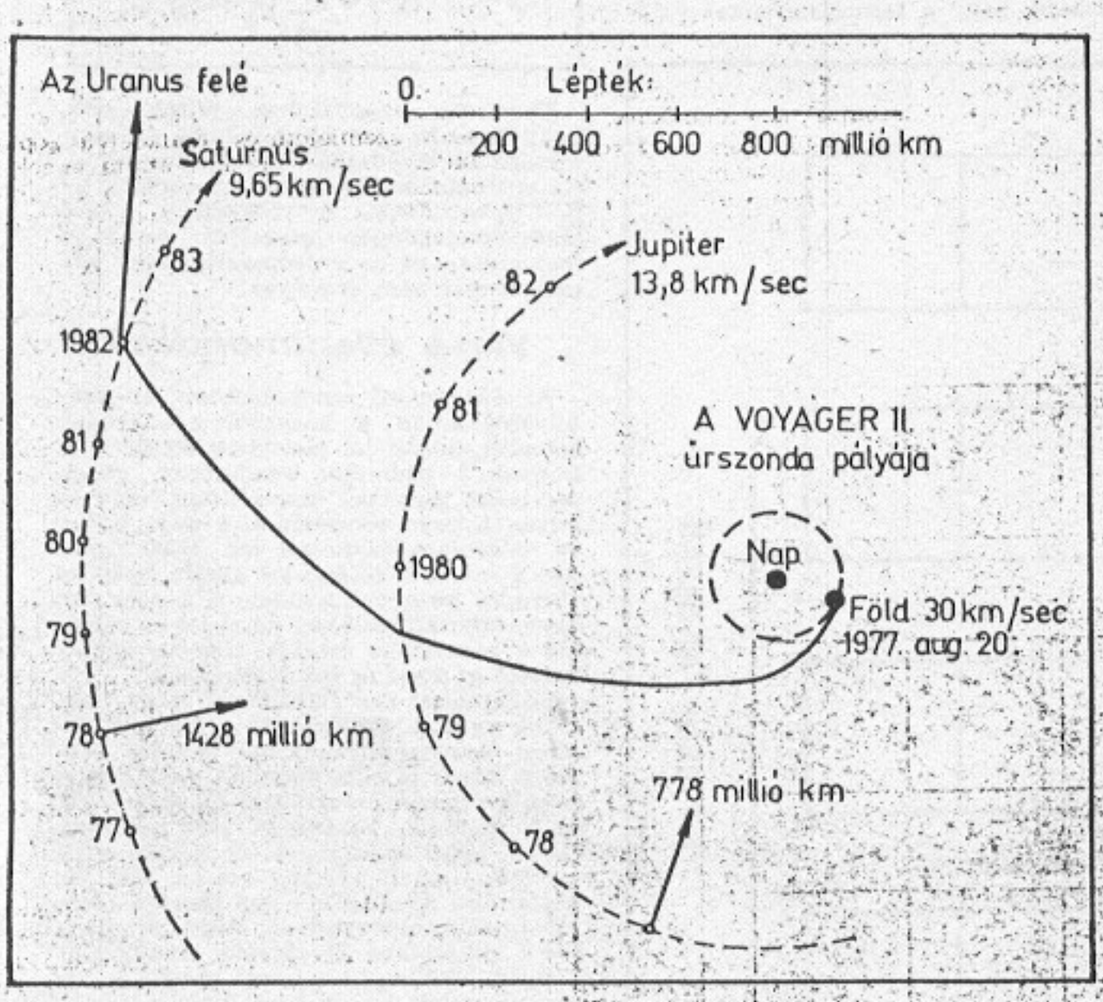
A kiválasztott DT időtartam alatt a Jupiter a J1 helyzetből a J2 helyzetbe kerül. Mi úgy vesszük, mintha az egész időtartam alatt a középső, J(1—2) pontban lenne. A szonda és a Jupiter DT idő alatt állandónak vehető távolságából — B1—J(1—2) — a Jupiter tömegéből ($1{,}9\ 10^{27}$ kg) és a gravitációs állandóból ($f = 6{,}67 \cdot 10^{-11}$ Nm^2/kg^2) a szondának a J(1—2) pont felé történő gyorsulása, majd sebessége és végül elmozdulása számítható (B1—B2'). Ezt és a szondának az időtartam kezdetén vett sebességéből származó elmozdulást (B1—A2) vektorilag összegezve kapjuk a tényleges elmozdulást: B1—B2. A szonda a Jupiterhez legközelebb még a Jupiter pályájának az elérése előtt lesz. Ekkor a legnagyobb a ráható gyorsító erő. Sebessége még ezután is jó ideig nő. Amikor végre a Jupiter a szondát lassítani kezdi, akkor már olyan messze van tőle, hogy lassító hatása szinte elhanyagolható.

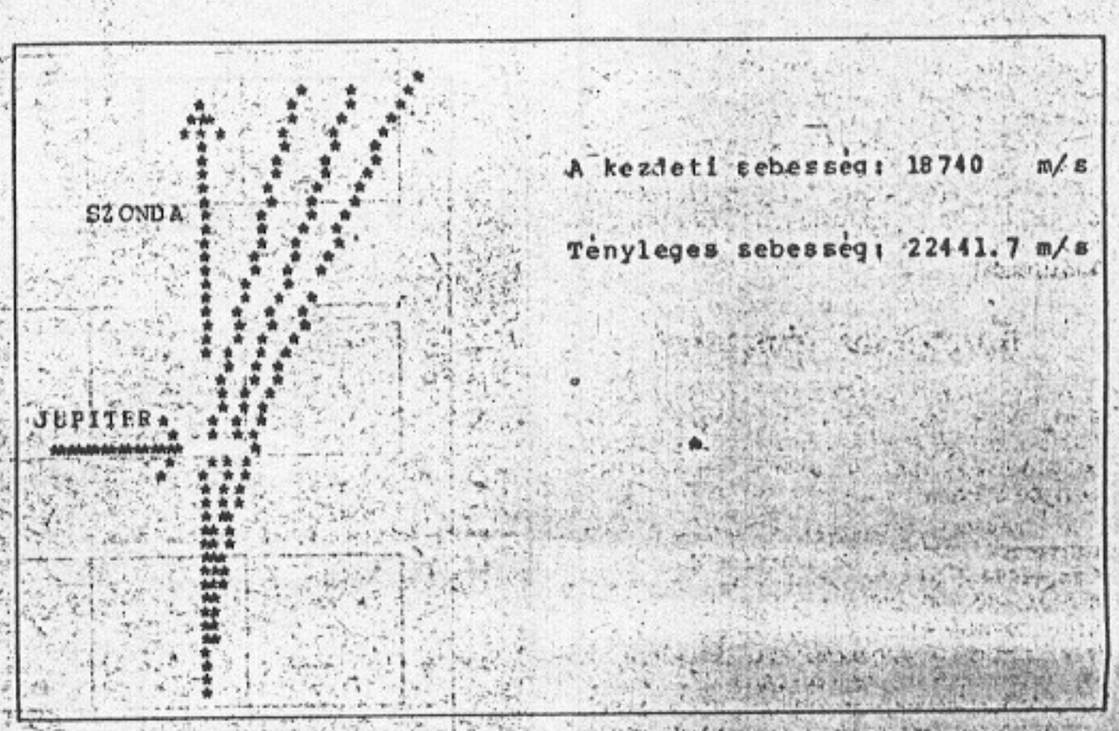
Lássuk a mozgás számításának és rajzoltatásának programját. A címfelirat után, a 60—160-as sorokban az adatok beadását és az igen hasznos segédleteket találjuk. V a szonda sebessége a megfigyelés kezdetén m/s-okban. A s-okban megadott DT lépéshossz valamivel több, mint egy óra, de ez csillagászati méretekben nem túlságosan nagy, gépünk viszonylag durva grafikája miatt pedig éppen megfelelő. A K, a Jupiter tömegének és az f gravitációs állandónak a szorzata. VJ, a Jupiter sebessége m/s-okban. DJ pedig a Jupiter elmozdulása a DT időegység alatt. XJ a Jupiter helykoordinátája (az YJ mindig nulla), XS és YS a szonda helykoordinátái. A 3. ábrán a vízszintes irányú nyíl a Jupiter állandó sebességének az irányát mutatja, a felfelé mutató nyíl pedig a szonda kezdeti sebességének az irányát. Ezeket rajzoltatjuk fel a 150—160-as sorokkal.

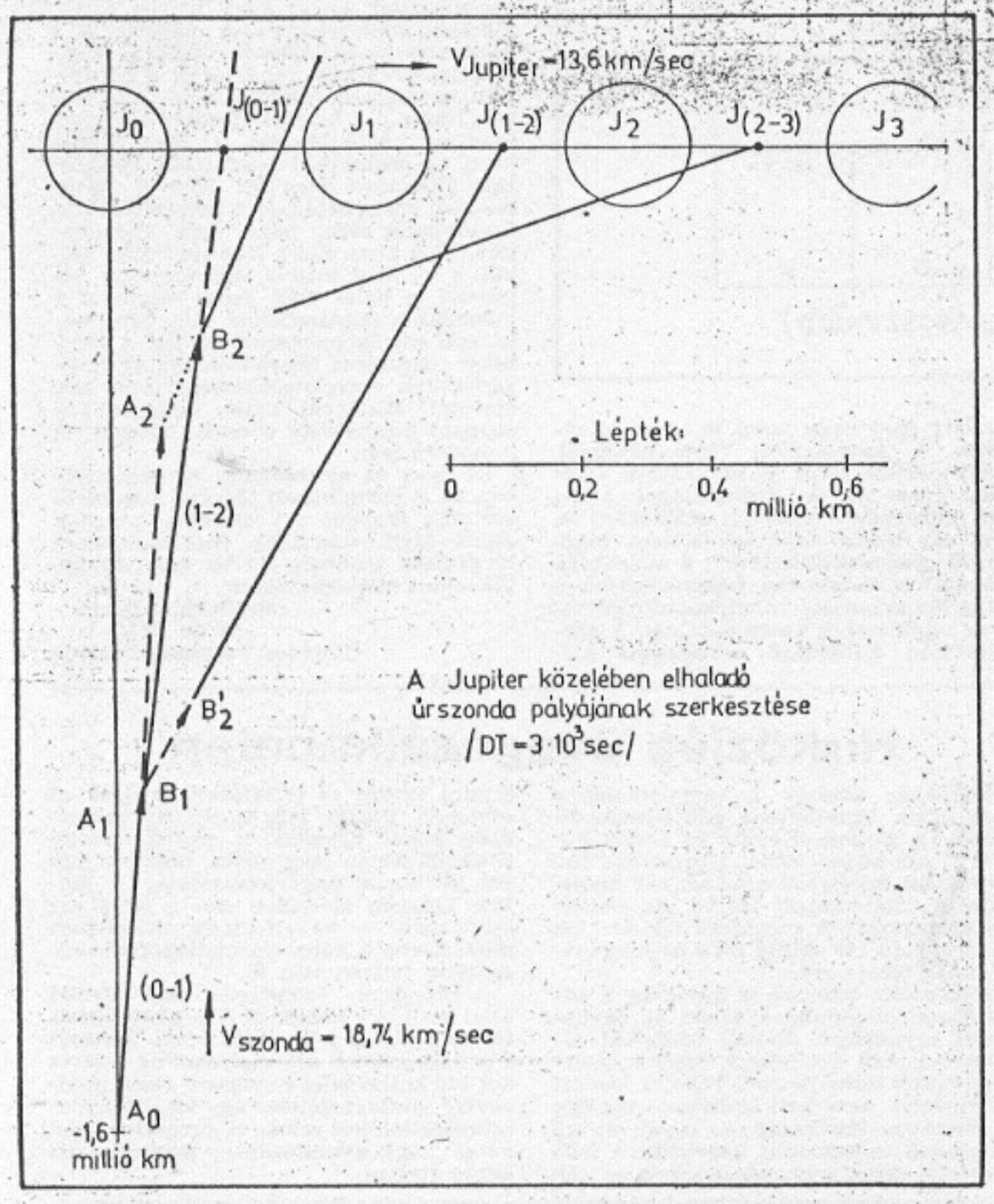
A 3. ábrán a képernyőről mutatunk be másolatot. Ez a program futásának a végén készült.

Bármely gomb lenyomására a program újra indul. A középső és a jobb oldali pályagörbe azoknak az eseteknek felel meg, amikor a szonda sebessége kezdetben nem a nyíl irányában mutat, hanem attól egy kissé eltér jobbra.

Kovács Mihály





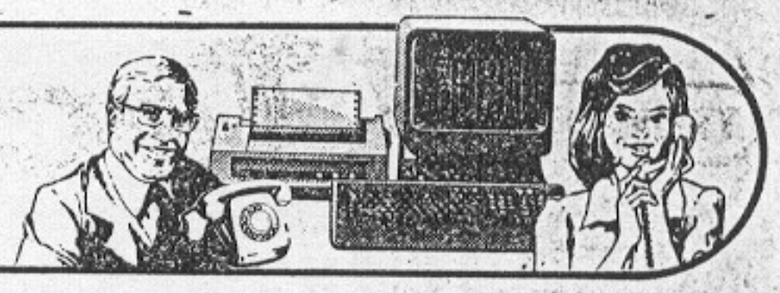



```
10 CLS
20 PRINT"   A VOYAGER-2 URSZONDA MOZGASA
      A JUPITER KÖZELÉBEN"

30 IF INKEY$="" THEN 30
40 VX=0
50 CLS
60 '          Adatok
70 V=18740: DT=4000: K=1.2673E17
80 VJ=13600: DJ=VJ*DT: XJ=DJ/2
90 XS=0: YS=-1.6E9: VY=V*DT
100 FOR Y=2 TO 20: SET(20,Y): NEXT
110 SET(19,3): SET(21,3): SET(18,4): SET(22,4)
120 FOR X=0 TO 17: SET(X,27): NEXT
130 SET(16,26): SET(16,28): SET(15,25): SET(15,29)
140 PRINT@512,"JUPITER";:PRINT@195,"SZONDA";
150 PRINT@2*64+32,"A kezdeti sebesseg: 18740   m/s";
160 PRINT@4*64+32,"Tenyleges sebesseg:         m/s";
170 '          Szamolas, rajzolas
180 FOR X=1 TO 45
190 B=XS: C=YS: E=XJ
200 D2=(XJ-XS)*(XJ-XS)+YS*YS: D=SQR(D2)
210 A=K/D2
220 W=A*DT*DT/2: WX=W*(XJ-XS)/D: WY=W*(-YS)/D
230 XS=XS+VX+WX: YS=YS+VY+WY: XJ=XJ+DJ
240 N=1.2*1E-8
250 RESET(2*N*E+20,27)
260 SET(2*N*XJ+20,27): SET(2*N*XS+20,-N*YS+27)
270 V2=(XS-B)*(XS-B)+ (YS-C)*(YS-C): V=SQR(V2)/DT
280 PRINT@4*64+51,V;
290 VX=XS-B: VY=YS-C
300 NEXT X: M=M+1
310 '          Ismetles uj adatokkal
320 IF INKEY$="" THEN 320
330 IF M=2 THEN 350
340 VX=5000000: GOTO 70
350 VX=10000000: GOTO 70
```

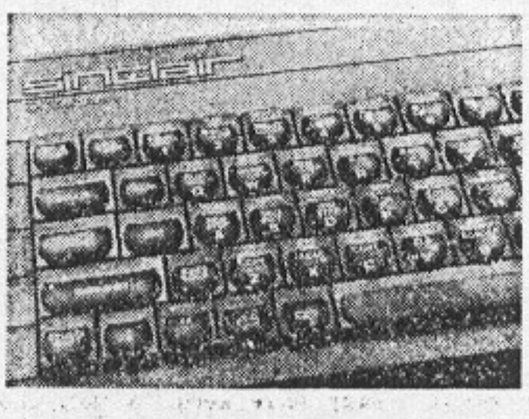
TECHNISKOLA

FÓRUM



SINCLAIR ZX SPECTRUM+

Angliában bemutatták az új „ZX Spectrum+" mikroszámítógépet, amely a + jel nélküli, hazánkban is közkedvelt modell továbbfejlesztett változata. Beépített 48 kByte-os memóriája van és írógép billentyűzete. A gép a régebbi modellel összeférő, így minden, a Spectrumhoz került szoftver és periféria használható hozzá. Továbbra is rendelkezésre áll a 8 szín, a nagy felbontású grafika, a 10 oktáv átfogású hang és az olcsó személyi számítógépek között a legnagyobb használható tárkapacitás.



A kemény műanyag billentyűk alatt egy hosszú léptetőbillentyű is van, számuk egyébként 17-tel nőtt. Ez lehetővé teszi számos művelet egyetlen gombnyomásra történő végrehajtását.

A Spectrum+ a ZX bővítőrendszerrel kombinálva a legkülönbözőbb nehézségi fokú személyi és professzionális számításokra, oktatási és szórakozási célokra is kiváló. A bővítés a hajlékonylemezes tárat helyettesítő ZX Microdrive-ból, ZX Interface I-ből és 4 Microdrive kazettából áll. Ábránkon a gép egy jellegzetes részlete látható.

HT 1080 Z-PROGRAM: KARÁCSONYI ÜDVÖZLET

Az alábbi program a karácsonyi ünnepek közeledtével válik aktuálissá. A program megjelenít egy fenyőfát, és kiírja a képernyőre az ünnepi jókívánságot. Azok a tanulók és tanárok, akik hozzáférnek iskolaszámító-géphez, kipróbálhatják, hogy az egyéb számításokon kívül ilyen célra is használható a készülék.

```
2 ' karácsonyi üdvözlet
4 '
6 ' karácsonyfa rajza
8 CLS:V=0
10 FORI=41TO7STEP-1:SET(31,I):NEXT
12 FORI=0TO8
14    FORK=0TOV
16       SET(30-I+K,9+4*I-K)
18       SET(32+I-K,9+4*I-K)
20    NEXTK
22    V=V+1
24 NEXTI
26 PRINT@79,"*";
28 ' szöveg balra léptetése
30 A$="KELLEMES KARÁCSONYI ÜNNEPEKET !
32 FORI=0TO31
34    PRINT@32-I,LEFT$(A$,I);
36    PRINT@992-I,LEFT$(A$,I);
38    FORK=1TO70:NEXT
40 NEXTI
42 ' "szilvorka"
44 FORK=1TO20
46 IFK>1THENPRINT@64*Y+X," ":PRINT@64*Z+22+X," ";
48 X=RND(9):Y=RND(14):Z=RND(14)
50 PRINT@64*Y+X,"*";:PRINT@64*Z+22+X,"*";
52 FORI=1TO50:NEXT
54 NEXTK
56 ' oh Tannenbaum !
58 OUT31,7:OUT30,254:OUT31,8:OUT30,15:OUT31,9:OUT30,15:OUT31,0
60 RESTORE
62 FORI=1TO47
64 READH,T
66 PRINT@79," ";
68 FORS=1TO6:OUT30,0:NEXT
70 PRINT@79,"*";
72 OUT30,H:FORS=1TO30*T:NEXT
74 NEXTI:OUT30,0
76 RUN
78 DATA208,8,155,6,155,2,155,12,138,4,123,6,123,2,123,12,123,4
80 DATA138,4,123,4,116,8,165,8,138,8,155,8,0,4,103,4,103,4,123,4
82 DATA92,12,103,4,103,4,116,4,116,12,116,4,116,4,138,4,103,12
84 DATA116,4,116,4,123,4,123,8,208,8,155,6,155,2,155,12,138,4,123
86 DATA6,123,2,123,12,123,4,138,4,123,4,116,8,165,8,138,8,155,8
```

VÁLASZOK AZ OKTÓBERI FIZIKAKÉRDÉSEKRE

Az októberi Fórum rovatban felvetett nyomozási-kutatási feladatok között voltak olyanok, amelyekben nagy szerepet játszott az elhanyagolás, a lényeges és lényegtelen szétválasztása.

Az egyik kérdés az volt, hogy a sötét hátterű ablaküvegben látjuk magunkat, miért nem látjuk magunkat a világos hátterű ablakban is?

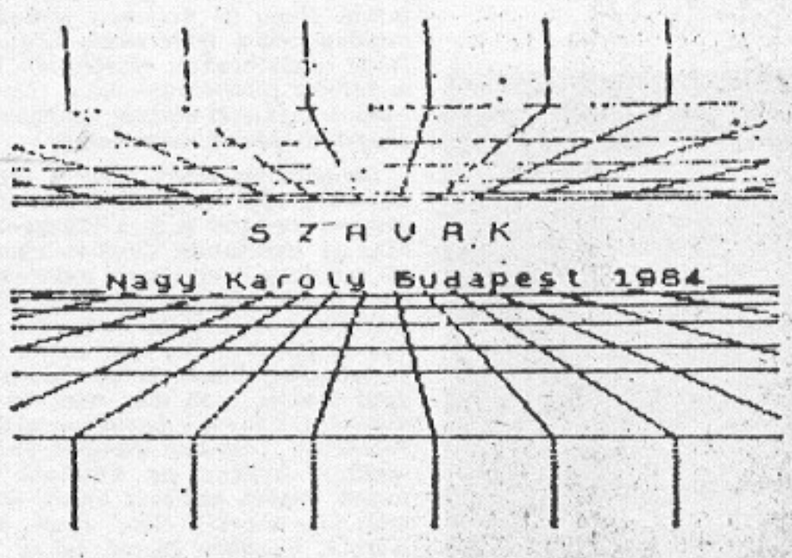
Aki már jártas az elhanyagolások „művészetében", könnyen válaszol: az ablaküveg a ráeső fényt mindkét oldalról nagyrészben átereszti, kisebb részben visszaveri (és elhanyagolható részben el is nyeli). A sötét hátterű ablaküvegre eső fény egy része átmegy az ablaküvegen, és a sötét háttéren elnyelődik, a másik része viszont visszaverődik az üvegről, és ezt látjuk. Ha sötét a háttér, a másik oldalról semmilyen fénysugárzás nem jön át. A világos hátterű ablaküveg viszont a világos háttérből is fényt ereszt át felénk, így azzal keveredik, az ablaküvegről visszavert, rólunk kiinduló fény. Most már csak azt kell meggondolni, mikor hanyagolható el a visszavert fény, és mikor nem.

Másik nyomozati kérdésünk a magas hegyekben honos hidegre vonatkozott. Itt az elhanyagolás másik oldalról jelentkezett. „Köztudomású — állítottuk —, hogy a meleg levegő felfelé száll." Ez igaz is, de csak akkor, ha közben a külső nyomás lényegesen nem változik meg! Vagyis abban az esetben, ha a légnyomásváltozás a magassággal jelentősen változik, állításunk nem igaz. Innen már csak egy rövid gondolat a magyarázat, mindenki megpróbálkozhat vele.

ZX SPECTRUM-PROGRAM: IDEGEN SZAVAK TANULÁSA

Júliusban közölt programunktól eltérően ez a program váltakozva véletlenszerűen kérdez magyar, illetve idegen szavakat. A szavakat a 200. sorban megadott „n" mennyiségű készletből válogatja ki véletlenszerűen. A 10—190. sorok a program szempontjából nem lényegesek, csak a bemutatott címet rajzolják meg. A 210—260. sorok tárolják be a szavakat (az a$ idegen, az m$ magyar szavakat jelent). A 320. sor kiválaszt egy szópárt, a 330. sor pedig kiválasztja, hogy magyarul vagy idegen nyelven kérdezze-e meg. Helytelen válasz esetén a 19. sorban a képernyőn kiírja az idegen szó = magyar szó jelentést. Úgy a helyes, mint a helytelen választ rövid hangjelzés kíséri.

Nagy Károly




```
  10 FOR i=1 TO 13
  20 READ a,b: LET c=254: LET d=
▼
  30 DATA 1,60,1,110,1,78,1,112,
1,74,1,68,1,116,1,60,1,122,1,52,
1,130,1,30,1,142
  40 PLOT a,b: DRAW c,d
  50 NEXT i
  60 FOR j=1 TO 12
  70 READ a,b: LET c=0: LET d=30
  80 DATA 20,0,20,142,60,0,60,14
2,100,0,100,142,140,0,140,142,18
0,0,180,142,220,0,220,142
  90 PLOT a,b: DRAW c,d
 100 NEXT j
 110 FOR i=1 TO 25
 120 READ x,y,a,b
 130 DATA 0,118,38,-8,0,130,68,-
20,20,142,70,-32,60,142,43,-32,1
00,142,17,-32,140,142,-10,-32,18
0,142,-38,-32,220,142,-66,-32,25
5,136,-83,-26,255,122,-66,-12,25
5,114,-25,-4,255,77,-17,3,255,70
,-52,10,255,57,-80,23,255,36,-97
,44,220,30,-74,50,180,30,-43,50,
140,30,-12,50,100,30,18,50,60,30
,49,50,20,30,80,50,0,40,89,40,0,
60,72,20,0,71,44,9,0,77,16,3
 140 PLOT x,y: DRAW a,b
 150 NEXT i
 160 PRINT AT 9,10;"S Z A V A K"
 170 PRINT AT 11,4;"Nagy Karoly
Budapest 1984"
 190 PAUSE 100
 195 CLS
 200 PRINT AT 5,1;"a gyakorlando
 szavak szama :": INPUT n: PRINT
 AT 5,30;n
 210 DIM a$(n,10)
 220 DIM m$(n,10)
 230 FOR i=1 TO n
 240 PRINT "a$ ";i;: INPUT a$(i)
PRINT TAB 5;a$(i)
 250 PRINT "m$ ";i: INPUT m$(i)
PRINT TAB 15;m$(i)
 260 NEXT i
 270 CLS
 310 DIM q$(1,10)
 320 LET s=INT (RND*((n+1)-1)+1)
 330 LET k=INT (RND*(3-1)+1)
 340 IF k=1 THEN PRINT AT 8,6;a$
(s)
 350 IF k=2 THEN PRINT AT 8,6;m$
(s)
 360 PRINT AT 15,6;"a valasz :"
 INPUT q$(1)
 370 IF q$(1)=a$(s) OR q$(1)=m$(
s) THEN PRINT AT 17,5;"O.K. a va
lasz helyes": BEEP .05,25: BEEP
.05,35: BEEP .05,40: BEEP .3,45.
 CLS : GO TO 320
 380 IF q$(1)<>a$(s) OR q$(1)<>m
$(s) THEN PRINT AT 17,5;"a valas
z nem helyes !": PRINT AT 19,5;a
$(s);"=";m$(s): PAUSE 100: BEEP
.05,0: BEEP .05,-2: BEEP .05,-5
 BEEP .3,-10: CLS : GO TO 320
```