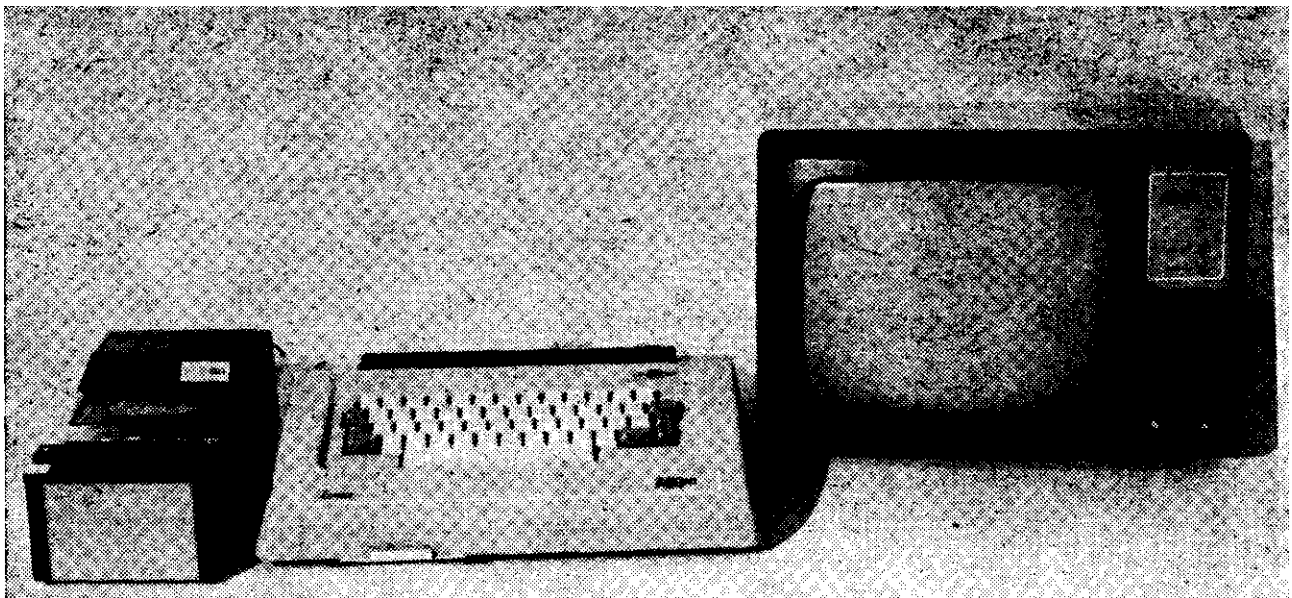


## Az ABC/80 számítógép BASIC–nyelvének ismertetése II. (Ciklusok, változók, függvények)

Ciklusnak nevezünk egy olyan programrészt, melyet többször egymásután szeretnénk végrehajtatni. Az eddigi ismereteink alapján a következőképpen tudunk ciklust készíteni (bal oldali program):

10 T=0	
20 X =0	X=0
30 IF T>10 THEN 80	30 FOR T=0 TO 100 STEP 1
40 X=X+2*T+1	40X=X+2*T+1
50 PRINT X	50 PRINT X
60 T=T+1	
70 GOTO 30	70 NEXT T
80 PRINT	80 PRINT

Ez a programrészlet az egész számok négyzetét írja ki  $1^2$ -től egészen  $100^2$ -ig. Először a 10-es címkéjű sorban a ciklusváltozó, T kezdőértékét állítottuk be. A 30 címkéjű sorban a végértéket határoztuk meg, ha ezt az értéket (100-at) a ciklusváltozó értéke nem haladja meg, akkor a program az 50 címkéjű sorban a kívánt szám négyzetét kiírja. A 60-as sorban mondjuk meg, hogy mennyivel növekedjen a ciklusváltozó (ezt *step*-nek, lépésnek nevezzük). Míg fut a program, állandóan vizsgáljuk, hogy T értéke meghaladta-e már a 100-at. Ha igen, akkor a ciklus befejeződik, és a program egy sor emelése után folytatódhat.



Az ABC/80 számítógép

A BASIC nyelv lehetőséget ad a ciklusok egyszerűbb megírására a FOR–NEXT utasításpár segítségével. Példánkban az előbbi programrészlet ciklusát ezzel az utasításpárral a jobb oldali oszlopba írtuk. A STEP szó után kell írni a lépésközt, ha ezt nem írjuk ki, akkor a lépésköz mindig +1.

Ciklusok jól alkalmazhatók például különböző függvénytáblázatok készítésére. A túloldali program koszinusz-táblázatot készít, a függvényértékek 0 radiántól 4 radiánig terjednek 0,2-es lépésközzel.

Az ABC/80 típusú géppel nemcsak a koszinusz függvényt, hanem más függvényeket is ki lehet írni. Az alábbi táblázatban a felhasználható függvényeket, illetve azok gépi megfelelőit tüntettük fel.

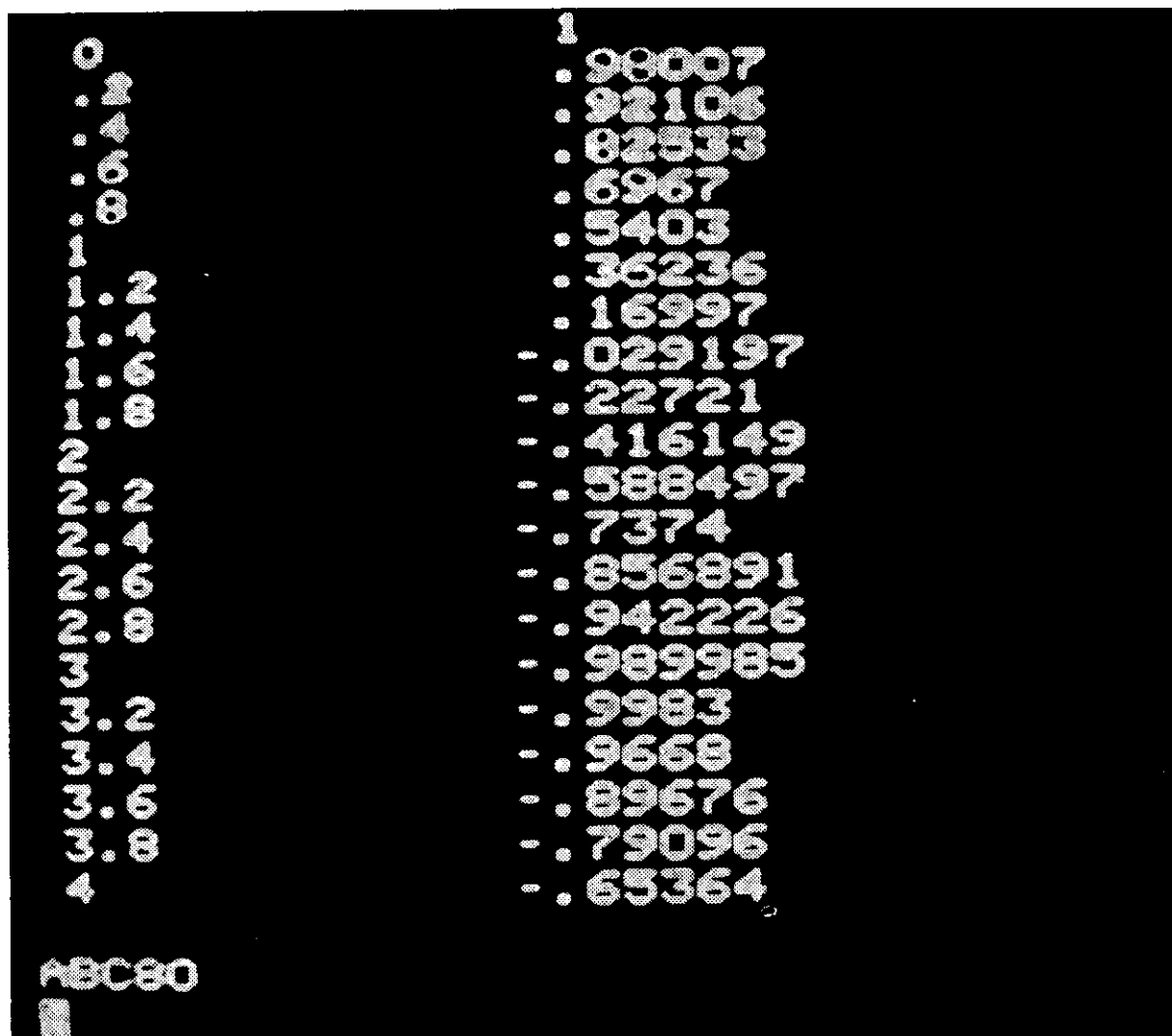
Függvény	Gépi megfelelője	Megjegyzés
$\sin x$	SIN(X)	az argumentum radiánban értendő
$\cos x$	COS(X)	az argumentum radiánban értendő
$\tan x$	TAN(X)	az argumentum radiánban értendő
$\arctan x$	ATN(X)	a $-\pi$ és $+\pi$ közti főértéket adja
$\ln x$	LOG(X)	természetes alapú logaritmus
$\lg x$	LOG10(X)	tíztes alapú logaritmus
$e^x$	EXP(X)	$e = 2,71\dots$
$\sqrt{x}$	SQR(X)	
$[x]$	INT(X)	$x$ egész része, $[-2, 1] = -3$
$ x $	ABS(X)	
sign $x$	SGN(X)	$x$ előjele, -1 ha $x < 0$ , 0 ha $x = 0$ , +1 ha $x > 0$ .
$x$ gépi egész része	FIX(X)	értéke sign $(x) \cdot [x]$ .

```

10 PRINT "KOSZINUSZ FÜGGVÉNY"
20 FOR X=0 TO 4 STEP .2
30 PRINT X, COS(X)
40 NEXT X
50 END

```

A 30 címkéjű sorban az X és a COS(X) közti vessző biztosítja a táblázatos kiíratást [ lásd Lőcs –Sarkadi – Nagy – Szlankó: BASIC programozási nyelv, 68. oldal].



A ciklusok másik alkalmazási területe az indexes változók kezelése. Vektorok, mátrixok beolvasását is célszerű mindig ciklusban végezni. Az első részben megismert aritmetikai változókhoz hasonlóan a vektorok illetve mátrixok jelölésére is szimbolikus neveket használhatunk, ezek az ABC nagybetűinek és a 0–9 számjegyek kombinációiból állnak.

*Vektoron* rendezett számsorozatot értünk. Egy vektor adott elemét úgy kaphatjuk meg, hogy leírjuk a vektor nevét (azonosítóját), és utána zárójelben az elemhez tartozó indexet, például  $A(7)$ . A vektorok indexei 0-tól indulnak és egyesével nőnek.

*Mátrixon* olyan  $n \times m$ -es számtáblázatot értünk, mely  $n$  sorból és  $m$  oszlopból áll. Megadásakor a következőképpen járunk el: megadjuk a mátrix azonosítóját és utána zárójelben, vesszővel elválasztva először a sor- majd az oszlopindexet. Például  $B(7, 2)$ . A sorok illetve oszlopok indexei 0-tól indulnak.

A vektorokat és a mátrixokat a programozási gyakorlatban tömböknek nevezzük. A tömbök kijelölése a tárban a DIM utasítással történhet (lásd az idézett könyv 81. oldalán). A DIM  $B(3)$ ,  $C(2, 3)$  utasítással egy  $B$  nevű, 4 elemű vektort (az indexek 0, 1, 2 és 3), valamint egy  $C$  nevű  $3 \times 4$  méretű mátrixot definiálunk. Az alábbi, bal oldali programrész az  $A$  vektor 0, 1, 2, ...,  $N$  indexű elemeinek értékét megnöveli kettővel, majd kikeresi a megnövelt értékek közül a legnagyobbat.

```

10 M=A(0)           10 INPUT C, D
20 FOR X=0 TO N      20 FOR I=1 TO C STEP 20
30 A(X)=A(X)+2       30 FOR N=2 TO D STEP 2
40 IF M>A(X) THEN 60 40 PRINT I+N
50 M=A(X)           50 NEXT N
60 NEXT X            60 NEXT I
70 PRINT M           70 END

```

Gyakran előfordul, hogy egy cikluson belül újabb ciklust is beiktatunk. Ezt a ciklusok „egymásba skatulyázásának” nevezzük. Ez történt a fenti jobb oldalon látható programban is. Ha ezt a programot futtatjuk, a képernyőn a következők jelennek meg:

```

RUN
?, 6, 4

```

3 4 5 5 6 7 7 8 9

Az első sorban a RUN szó indítja a program futását. A kérdőjel utáni két szám a C illetve D értéke, azt mi adjuk meg. A „RETURN” gomb megnyomása után kapjuk a harmadik sort, ez a program végeredménye. Jól látható, hogy a ciklusok hogyan, milyen sorrendben futnak le.

\*

Az alábbi feladatok megoldását a következő címre lehet küldeni:

**Appel György tanár**

FPI Oktatástechnika

*Budapest*, Bródy Sándor u. 14. 1088.

3. *feladat.* Oldjuk meg a következő egyenletrendszert az ABC/80 segítségével:

$$3,5x + 0,3y = 15,4424, 3x - 48,5y = 53,2$$

4. *feladat.* Írjunk olyan BASIC-programot, amely kiszámítja és kiírja az 1000 és 5000 közötti prímszámokat.