

MIT TUD ÖN AZ ON-RŐL? ÖN AZ ON-RŐL?



Múlt havi vallatónkban, amikor is az Aircomp 16 került a szigorú inkvizítorok kőpadjára, ígéretet tettünk, hogy e havi számunkban külön kis írásban foglalkozunk a gép BASIC-jének ON-jával

Az AIRCOMP-16 BASIC-jének egyik legszenálisabb „húzása”:

ON UTASÍTÁS

Az ON utasítás általában minden BASIC-ben megtalálható, a következő formában:

ON kifejezés GOTO 1. sorszám, 2. sorszám ...

Ezt a gép úgy hajtja végre, hogy kiszámítja a kifejezés értékét, s ha van a GOTO után annyiadik sorszám, amennyi a kifejezés értékének egész része, akkor a sorszám által meghatározott sorra ugrik, különben a futás a következő soron folytatódik. GOTO helyett GOSUB is írható.

Az AIRCOMP-16 BASIC ON utasítása viszont a következő alakú:

ON kifejezés '1. utasításcsoport '2. utasításcsoport ' ...

ahol az utasításcsoportok állhatnak egy, vagy több – kettősponttal elválasztott – utasításból. A hatás annyiban különbözik az előzőekben tárgyaltaktól, hogy a gép megkeresi az annyiadik aposztrófot, amennyi a kifejezés értékének egész része (ha nincs ennnyedik aposztróf, akkor a következő sorra megy), s ettől kezdve hajtja végre az utasításokat, amíg egy következő aposztrófot, vagy sorvéget nem talál.

Erre még bárki azt mondhatná, hogy jó, jó, de akkor itt a hagyományos ON utasítást:

ON kifejezés 'GOTO '1. sorszám 'GOTO 2. sorszám 'GOTO 3. sorszám ' ...

...kban kell írni? A válasz, hogy NEM! Ugyanis ha az aposztrófok között mindenhol ugyanaz az értékadástól eltérő utasítás szerepelne, akkor azt elég egyszer leírni az első aposztróf elé, tehát példánkban így:

ON kifejezés GOTO '1. sorszám '2. sorszám '3. sorszám ' ...

Lássunk most néhány példát az ON utasítás használatára:

1. A programban van 2 kapcsoló K és L, melyek értéke 0 vagy 1. Egy megfelelő helyen a négyféle kapcsolóállástól (K = 0; L = 0; K = 0, L = 1; K = 1, L = 0; K = 1, L = 1) függően négy különböző dolgot akarunk csinálni. Ezt a következőképpen tehetjük meg (persze az aposztrófok közé azokat az utasításokat kell írni, amiket az egyes esetekben végre akarunk hajtani).

```
100 ON 2*(K+1)-L*A=1: PRINT "L VAN BEKAPCSOLVA,";A=0: PRINT "MINKET K
APCSOLO KI VAN KAPCSOLVA,";A=1: PRINT "MINKET KAPCSOLO BE VAN KAPCSO
LVA,";A=10: PRINT "K VAN BEKAPCSOLVA,";
110 PRINT "ÍGY A ERTEKE: ";A
```

2. Egy játékprogram elejét sokszor így lehet szépen megírni:

```
20 INPUT "A VERSENYZOK SZAMA (1-6)? ";N
30 FOR I=1 TO N
40 ON I PRINT "AZ ELSO";"A MASODIK";"A HARMADIK";"A NEGYEDIK";"AZ
OTODIK";"A HATODIK";
50 INPUT "VERSENYZO NEVE: ";M$(I)
60 NEXT I
```

3. Nagyon jól ki lehet használni azt is, hogy a logikai műveletek (pl. az összehasonlítások, AND, OR) értéke 1 (igaz) vagy 0 (hamis). Tegyük fel, hogy egy játék végén P őrzi azt, hogy a játékosnak há-

nyadik próbálkozásra sikerült valamit kitalálni. P értékétől függően meg akarjuk dicsérni, vagy el akarjuk marasztalni. Ezt így tehetjük meg:

```
500 ON (P<11)+(P<13)+(P<17)+1 PRINT "ABSZOLUT NEM MEGY NEKED EZ A JATE
K, INKABB PROBALJ KI EGY MASIKAT:";"NEM REMEKELTEL TULSAGOSAN!";"EGESZ
JO, DE MEG LEHET JAVITANI!";"GRATULALOK, KIVALO A TELJESITMENYED:"
```

4. Szeretnénk készíteni egy rajzolóprogramot, aminek a következő dolgokat kell tudnia:

Először a képernyő közepén megjelenik egy pont. A pont az I, J, K, M billentyűk hatására mozogjon felfelé, balra, jobbra, ill. lefelé, és húzzon maga után vonalat. Persze vigyázni kell arra, hogy ne menjen ki a képernyőről, hiszen akkor hibajelzést kapnánk. Egy ilyen program a következő (a 10-es sor a képernyő grafikusra váltását és törlését, a 20-as sor a billentyűzet figyelését végzi):

```
10 CR=0:GL=200: PRINT CHR$(5):CR=1
15 X=160:Y=100
18 PLOT X,Y
20 A=USR(A): IF A<73 OR A>77 THEN GOTO 20
30 ON A-72:Y=(Y)*X=X*(X)*X=X*(X<319): GOTO 20:Y=Y+(Y<199)
40 GOTO 18
```

5. Végül még egy példa konkrét utasítások nélkül. Van egy 10 utasításból álló sorozatunk, melyet 6-szor kell egymás után végrehajtani úgy, hogy az 1-4. és 8-10. utasítások mind a 6 esetben megegyeznek, de az 5., 6. és 7. utasítások eltérőek. Ezt a következő szerkezettel programmal lehet megvalósítani:

```
10 FOR I=1 TO 6
20 utasitas-1
30 utasitas-2
40 utasitas-3
50 utasitas-4
60 ON I:ut.5/1:ut.6/1:ut.7/1:ut.5/2:ut.6/2:ut.7/2:ut.5/3:ut.6/3:ut.7/3
ut.5/4:ut.6/4:ut.7/4:ut.5/5:ut.6/5:ut.7/5:ut.5/6:ut.6/6:ut.7/6
70 utasitas-8
80 utasitas-9
90 utasitas-10
100 NEXT I
```

Aki ezek után a példák után még kételkedik az AIRCOMP-16 BASIC ON utasításának hatalmas előnyeiben, az próbálja megírni valamelyik másik BASIC-ben a 2., 3. és 4. programot.

Persze itt csak néhány „kapásból” kitalált példa van, akinek még vannak jó ötletei, az AIRCOMP-16 BASIC ON-jának felhasználására, kérjük, írja meg a szerkesztőségünkbe.

Király Zoltán

A szerkesztő azért van,

hogy a lap olyan legyen,

mint amilyenek az olvasói!