ötlet

PROGRAM AJÁNLAT

VC 20
F billentyűk programozása

Cikkeinkben gyakran hivatkozunk lapunk alapelvére, miszerint a szerkesztő azért van ... stb. Nos, ez az alapelv sokszor úgy érvényesül, hogy az olvasók által beküldött anyagok adják lapszámunk oldalainak egy részét, másszor meg úgy, hogy az olvasók kérdéseire adott válaszok, a kívánságok kielégítése tölt ki lapot, lapokat. Annak idején elhatároztuk, hogy a VC 20-as géppel nem nagyon foglalkozunk mert már ,,kiment a divatból'', azután a hozzánk érkezett levelek, telefonhívások hatására megváltoztattuk véleményünket, s előbb vallatót, majd VC 20 prolongálva címmel cikkeket közöltünk a Commodore-ok doyenjével kapcsolatban. Elég régóta hallgatunk e gépről, s most egy ifjú olvasó kívánságát kielégítve ismét a VC 20-asok táborának kedvezünk. Az olvasó az alig 13 éves Melich Krisztián azt kérdezte tőlünk levelében, hogy hogyan lehetne programozni a gép F1–F8 billentyűit. Egy másik törzsolvasónk **Tóth Géza,** aki alig idősebb Krisztiánnál most rövid gépi kódú programmal válaszol a kérdésre. Ez a probléma ugyanis kicsit bonyolultabb, mint azt Krisztián hitte, efféle gépi kódú program nélkül ugyanis a dolog megoldhatatlan. Reméljük, a program más VC 20 tulajdonosoknak is hasznos, a gép rejtelmeibe beavatottaknak pedig tartalmaz némi programozói tanulságot.

A program bővített vagy bővítetlen VC 20-szal egyaránt használható. Betöltés után használata a következő: RUN-nal indítjuk, mire megjelenik a képernyőn egy kérdés:

Hány betűt tároljanak a billentyűk? (30–110)

Ez értelemszerűen azt jelenti, hogy a nyolc billentyű összesen 110 betűt képes tárolni és megjeleníteni. A válasz megadása után megjelenik a READY felirat és megkezdhetjük a billentyűk programozását. Ehhez az alábbi formátumot kell használnunk:

@ 1,''Tóth Géza programja''

A ,,kukac'' mellett álló szám természetesen a billentyű száma. Ha netán a szöveg nem fér ki már (tehát túléptük a megadott karakterszámot), akkor ILLEGAL QUANTITY hibajelzést kapunk, s próbálkozhatunk újra annak a billentyűnek a programozásával amelynél éppen tartottunk.

Ha elvégeztük a billentyűk programozását, akkor a következőképpen használhatjuk őket:

!ON – a programozott billentyűk bekapcsolása

!END – a programozott billentyűk kikapcsolása (de a RUN/STOP+RESTORE is ezt a hatást váltja ki)

Egy konkrét példa, amelyből egyéb érdekes lehetőségek is kiderülnek. Beírhatjuk például egy billentyűre az alábbit is:

@ 1,''LOAD''+CHR$(13)

Ha ezután !ON-t írunk, s megnyomjuk az F1-et, a gép úgy veszi mintha a LOAD parancsot írtuk volna a gépbe, s megnyomtuk volna a RETURN-t is, azaz elindul a töltés. (Hiszen a CHR$(13)= RETURN)

Ha a program beírásakor hibázunk, akkor futtatásnál ,,HIBA A PROGRAMBAN'' hibajelzést kapunk. Ha az adatokban (DATA-kban) hibáztunk, akkor pedig ennek megfelelő hibajelzést.

A program működéséről:

A program miután megkérdezi a puffer méretét, kiszámítja mennyivel kell a BASIC felső határát lejjebb tolni, hogy a program és a puffer elférjen.

Az első FOR ciklus a gépi kódú programot tölti be (a ''REM PROGRAM''-tól a ''REM-ADATOK''-ig terjedő rész).

A ''REM ADATOK'' után következő DATA sorokban tároljuk a kezdőcímtől függő byte-ok relatív helyét a programban és a kezdőcímhez mért relatív tartalmukat. (Például JMP kezdőcím+580, a JMP után következő byte-okat úgy állítja be, hogy azok értéke K+580 legyen.)

```
0 PRINT"□  #FUNKCIOBILLENTYUK"
1 PRINT
2 PRINT"HANY BETUT TAROLJANAK A BILLENTYUK?(30-110)
3 INPUTP
4 IFP<30ORP>110ORP<>INT(P)GOTO3
10 P=P+8:VR=PEEK(56)*256+PEEK(55)
20 VU=VR-P-385
30 VH=INT(VU/256):VL=VU-VH*256
40 K=VU+1:W=0
50 FORN=KTOK+395:READA:POKEN,A:W=W+A:NEXT:IFW<>49065THENPRINT"□"HIBA A PROGRAMBAN!
":END
55 W=0
60 READN:IFN=-1GOTO100
65 READA:C=K+A:CH=INT(C/256):CL=C-CH*256
70 POKEN+K,CL:POKEN+K+1,CH:W=W+A+N
80 GOTO60
100 IFW<>8886THENPRINT"HIBA AZ ADATOKBAN!":END
105 KH=INT(K/256):KL=K-KH*256:POKEK+350,KL:POKEK+352,KH
110 S=K+385:SH=INT(S/256):SL=S-SH*256:POKEK+370,SL:POKEK+372,SH
115 POKEK+282,P
116 POKE56,VH:POKE55,VL
120 POKE251,0:SYSK+369:NEW
130 REM----PROGRAM----
510 DATA165,153,240,3,76,29,242,165,211,133,202,165,214,133
520 DATA201,152,72,138,72,165,200,240,6,76,87,230,32,66,231
530 DATA165,198,133,204,141,146,2,200,13,164,251,240,243,185
540 DATA0,82,240,238,169,1,208,236,120,165,207,240,12,165
550 DATA206,174,135,2,160,0,132,207,32,161,234,32,99,80,201
560 DATA131,208,16,162,9,120,134,198,169,243,237,157,118
570 DATA2,202,208,247,240,194,201,13,208,187,76,29,230,164
580 DATA251,240,7,185,0,82,240,2,208,28,32,207,229,201,133
590 DATA144,24,201,141,176,20,233,132,201,4,176,1,24,42,41
600 DATA7,170,32,144,80,185,0,82,200,132,251,170,24,96,169
610 DATA255,138,240,9,200,185,0,82,208,250,202,208,247,200
620 DATA96,32,158,215,202,224,8,176,118,134,10,32,253,206
630 DATA32,158,205,32,163,214,133,147,170,240,102,40,100
640 DATA166,10,32,144,80,132,13,166,10,232,32,144,80,165
650 DATA13,56,101,147,133,103,32,236,80,169,0,133,251,166,183
660 DATA202,157,0,82,164,147,240,12,136,202,177,34,240,57
670 DATA157,0,82,136,16,245,96,132,97,133,98,162,8,32,144
680 DATA80,132,99,165,97,56,166,97,164,98,229,98,40,17,240
690 DATA14,228,99,176,10,169,0,82,153,0,82,232,200,208,242
700 DATA96,152,24,229,97,24,101,99,201,127,144,3,76,72,210
710 DATA168,166,93,202,228,97,144,233,189,0,82,153,0,82,136
720 DATA208,242,32,115,0,201,64,208,9,32,115,0,32,160,80
730 DATA76,174,199,201,33,208,9,32,115,0,32,85,81,76,174
740 DATA199,32,121,0,76,231,199,201,128,240,16,169,145,32
750 DATA255,206,162,1,160,80,142,36,3,140,37,3,96,162,14
760 DATA32,115,0,160,242,208,240,162,50,160,81,142,8,3,140
770 DATA9,3,96
781 DATA133,0,134,0,135,0,136,0,137,0,138,0,139,0,140,0
800 REM-----ADATOK-----
810 DATA69,98,132,143,188,143,196,143,206,235,242,143,316,159,329,340
820 DATA43,300,103,300,135,300,150,300,216,300,229,300,264,300,297,300,300,300,2
67,300
1000 DATA-1
```