

1: This is a short program for the UK101. It was inspired by an
 2: article in August's edition of PCW, which was a small Assembly
 3: program written for the BBC Micro. That program split multiple
 4: statement lines up into separate lines. Well with that program in
 5: mind, I went on to devise a program which not only splits up
 6: multi-statement lines but also emulates the BBC Micro by indent-
 7: ing FOR..NEXT loops.

8: The program is given in both Assembler (for those of you with
 9: the ENCODER Chip) and in Hexadecimal.

11: The program resides at the area of free RAM from \$0240 onwards
 12: and is protected against a cold start. To initialise it from
 13: BASIC, type on one line:

14: POKE538,64:POKE539,2

15: (This points BASIC's output vector to the start of the new rou-
 16: tine)

18: If PREMIER's TK II is used the routine will be disabled after
 19: any TK command is issued, and the routine must be re-initialised
 20: as above.

22: The routine, as listed above, ignores the existence of the TK II
 23: chip (Jumping straight to the CEGMON output routine at \$FF9B), but
 24: output may be sent through TK II by changing \$FF9B to \$807C.

26: Note : When saving a program to tape, this routine must be dis-
 27: abled by issuing the commands (on one line, in Immediate mode)

29: POKE538,155:POKE539,255

31: or if TK II is used:

33: POKE538,124:POKE539,128

35: Adaptation to other machines should not be a problem...any
 36: 6502-based computer which vectors its output through a RAM ad-
 37: dress should be suitable (although the ASCII values for '!', ' ',
 38: , FOR and NEXT may need to be changed if the machine has a non-
 39: standard character set, as will the address of the CR/LF routine).

```

41: 0240 C9 20      CMP #$20      ;Is is a space?
42: 0242 D0 27      BNE :NSPC
43: 0244 AD 00 02    LDA $0200      ;Cursor displacement
44: 0247 C9 00      CMP #$00      ;First space?
45: 0249 F0 1E      BEQ :SPC1      ;Then skip
46: 024B C9 07      CMP #$07      ;If char count >7
47: 024D 10 1A      BPL :SPC1      ;Then skip
48: 024F BD CF 02    STA :MARG      ;Store in margin
49: 0252 EE CF 02    INC :MARG
50: 0255 AD CE 02    LDA :NEST      ;If nesting depth=0
51: 0258 F0 0F      BEQ :SPC1      ;Then skip
52: 025A 8A         TXA
53: 025B 48         PHA            ;Save X-register
54: 025C AE CE 02    LDX :NEST      ;Counter
55: 025F A9 20      LDA #$20      ;Space char
56: 0261 20 CB 02 :LOOP JSR :OUT      ;Print a space
57: 0264 CA         DEX
58: 0265 D0 FA      BNE :LOOP      ;Until nesting depth printed
59: 0267 68         PLA
60: 0268 AA         TAX
61: 0269 A9 20      :SPC1 LDA #$20      ;Restore X-register
62: 026B 48         :NSPC PHA            ;Restore space
63: 026C AD D1 02    LDA :STAK+1
64: 026F 8D D0 02    STA :STAK
65: 0272 AD D2 02    LDA :STAK+2
66: 0275 8D D1 02    STA :STAK+1
67: 0278 68         PLA
68: 0279 8D D2 02    STA :STAK+2      ;Last 3 chars printed
69: 027C C9 52      CMP #$52      ;Char 'R'
70: 027E D0 11      BNE :NoF      ;If not, then not a 'FOR'
71: 0280 AD D1 02    LDA :STAK+1
72: 0283 C9 4F      CMP #$4F      ;Char 'O'
73: 0285 D0 0A      BNE :NoF
74: 0287 AD D0 02    LDA :STAK
75: 028A C9 46      CMP #$46      ;Char 'F'
76: 028C D0 03      BNE :NoF
77: 028E EE CE 02    INC :NEST      ;Another 'FOR' found
78: 0291 AD D0 02 :NoF LDA :STAK

```



```

79: 0294 C9 4E      CMP ##4E      ;Char 'N'
80: 0296 D0 11      BNE :NoN      ;Not a 'NEXT'
81: 0298 AD D1 02    LDA :STAK+1
82: 029B C9 45      CMP ##45      ;Char 'E'
83: 029D D0 0A      BNE :NoN
84: 029F AD D2 02    LDA :STAK+2
85: 02A2 C9 58      CMP ##58      ;Char 'X'
86: 02A4 D0 03      BNE :NoN
87: 02A6 CE CE 02    DEC :NEST      ;Another 'NEXT'
88: 02A9 AD D2 02 :NoN LDA :STAK+2
89: 02AC C9 3A      CMP ##3A      ;Char ':'
90: 02AE D0 1B      BNE :OUT      ;Exit if not a ':'
91: 02B0 20 CB 02    JSR :OUT
92: 02B3 20 F5 FB    JSR $FBF5      ;Print CR/LF (to divide line)
93: 02B6 8A          TXA
94: 02B7 48          PHA      ;Save X-register
95: 02B8 18          CLC      ;Clear carry for addition
96: 02B9 AD CF 02    LDA :MARG
97: 02BC 6D CE 02    ADC :NEST      ;Depth of indentation
98: 02BF AA          TAX      ;Into counter
99: 02C0 A9 20      LDA ##20      ;Space char
100: 02C2 20 CB 02 :loop JSR :OUT      ;Print it
101: 02C5 CA          DEX
102: 02C6 D0 FA      BNE :loop
103: 02C8 68          PLA
104: 02C9 AA          TAX
105: 02CA 60          RTS      ;Restore X-register
106:                                     ;Return
107: 02CB 4C 9B FF :OUT JMP $FF9B      ;BASIC output routine
108:
109: 02CE 00          :NEST ##00      ;Memory locations used.
110: 02CF 00          :MARG ##00
111: 02D0 00 00 00 :STAK ##00 ##00 ##00

```