Appendix V - Assembler listing

```
7500 A9 04              LDA #&04            \Make cursor keys return ASCII codes
7502 A2 01              LDX #&01
7504 20 F4 FF           JSR osbyte
7507 A9 19              LDA #break MOD 256  \Point BRK vector to new BRK handler
7509 8D 02 02           STA brkv
750C A9 78              LDA #break DIV 256
750E 8D 03 02           STA brkv+1
7511 A9 83              LDA #&83            \Calculate OS high water mark (OSHWM)
7513 20 F4 FF           JSR osbyte
7516 86 70              STX textstart       \and store it in textstart and textend
7518 84 71              STY textstart+1
751A 86 72              STX textend
751C 84 73              STY textend+1
751E A5 70      .textptrs LDA textstart     \Define curpos and mempos
7520 85 7E              STA curpos
7522 85 74              STA mempos
7524 A5 71              LDA textstart+1
7526 85 7F              STA curpos+1
7528 85 75              STA mempos+1
752A A9 00              LDA #&00            \Set cursor to top left
752C 85 7C              STA hpos
752E 85 7D              STA vpos
7530 A9 16      .menu   LDA #&16            \Select MODE 7 for screen mode
7532 20 EE FF           JSR oswrch
7535 A9 07              LDA #&07
7537 20 EE FF           JSR oswrch
753A A2 07              LDX #&07            \7 lines to be printed
753C A0 00              LDY #&00            \Start count at zero
753E B9 88 75   .mloop  LDA mtext,Y
7541 20 E3 FF           JSR osasci          \Print menu
7544 C8                 INY
7545 C9 0D              CMP #&0D            \End of a menu item?
7547 D0 F5              BNE mloop           \No, get remaining letters
7549 CA                 DEX                 \Update counter
754A D0 F2              BNE mloop
754C 20 E0 FF   .mloop2 JSR osrdch          \Input a character from keyboard
754F 90 0C              BCC mnoerr          \Check if ESCAPE was pressed
7551 C9 1B              CMP #&1B
7553 D0 08              BNE mnoerr
7555 A9 7E              LDA #&7E            \Acknowledge ESCAPE condition
7557 20 F4 FF           JSR osbyte
755A 4C 4C 75           JMP mloop2          \and try again
755D C9 2A      .mnoerr CMP #ASC("*")       \Validate input (1-4 or *)
755F F0 08              BEQ mcase
7561 C9 31              CMP #ASC("1")
7563 90 E7              BCC mloop2
7565 C9 35              CMP #ASC("5")
7567 B0 E3              BCS mloop2
7569 C9 31      .mcase  CMP #ASC("1")       \Case A of ...
756B D0 03              BNE mnotload
756D 4C EC 75           JMP load
7570 C9 32      .mnotload CMP #ASC("2")
7572 D0 03              BNE mnotsave
7574 4C 52 76           JMP save
7577 C9 33      .mnotsave CMP #ASC("3")
```

```
7579 D0 03              BNE mnotedit
757B 4C CD 76           JMP edit
757E C9 34      .mnotedit  CMP #ASC("4")
7580 D0 03              BNE mnotprint
7582 4C 3B 77           JMP print
7585 4C A3 77   .mnotprint JMP oscom
7588 20 20 20
     20 20 20
     20 20 20
     20 20 20
     20 20 20
     20         .mtext     EQUS "                "
7598 57 4F 52
     44 50 52
     4F                    EQUS "WORDPRO"
759F 0D                    EQUB &0D
75A0 0D                    EQUB &0D
75A1 31 2E 20
     4C 6F 61
     64 20 61
     20 66 69
     6C 65                 EQUS "1, Load a file"
75AF 0D                    EQUB &0D
75B0 32 2E 20
     53 61 76
     65 20 61
     20 66 69
     6C 65                 EQUS "2, Save a file"
75BE 0D                    EQUB &0D
75BF 33 2E 20
     45 64 69
     74 20 61
     20 66 69
     6C 65                 EQUS "3, Edit a file"
75CD 0D                    EQUB &0D
75CE 34 2E 20
     50 72 69
     6E 74 20
     61 20 66
     69 6C 65              EQUS "4, Print a file"
75DD 0D                    EQUB &0D
75DE 2A 2E 20
     4F 53 20
     63 6F 6D
     6D 61 6E
     64                    EQUS "*, OS command"
75EB 0D                    EQUB &0D
75EC 20 E7 FF   .load      JSR osnewl          \Output a carriage return
75EF A2 00                 LDX #&00            \Prompt user for input
75F1 BD 43 76   .lloop     LDA ltext,X
75F4 C9 0D                 CMP #&0D
75F6 F0 07                 BEQ lout
75F8 20 E3 FF              JSR osasci
75FB E8                    INX
75FC 4C F1 75              JMP lloop
75FF A9 0C      .lout      LDA #&0C            \Input parameter - max. line length
7601 20 C0 77              JSR input
```

```
7604 90 03              BCC lover
7606 4C 30 75           JMP menu
7609 A9 33      .lover  LDA #buffer MOD 256     \Set up a block for loading
760B 8D 00 01           STA block               \Block 0-1 address of filename
760E A9 7B              LDA #buffer DIV 256
7610 8D 01 01           STA block+1
7613 A5 70              LDA textstart           \Block 2-5 load address
7615 8D 02 01           STA block+2
7618 A5 71              LDA textstart+1
761A 8D 03 01           STA block+3
761D A9 00              LDA #&00
761F 8D 04 01           STA block+4
7622 8D 05 01           STA block+5
7625 8D 06 01           STA block+6             \Block 6 load address flag
7628 A9 FF              LDA #&FF                \OSFILE command (load file)
762A A2 00              LDX #block MOD 256
762C A0 01              LDY #block DIV 256
762E 20 DD FF           JSR osfile
7631 AD 0A 01           LDA block+10            \Calculate new textend
7634 18                 CLC
7635 65 70              ADC textstart
7637 85 72              STA textend
7639 AD 0B 01           LDA block+11
763C 65 71              ADC textstart+1
763E 85 73              STA textend+1
7640 4C 1E 75           JMP textptrs
7643 46 69 6C
     65 20 74
     6F 20 6C
     6F 61 64
     3A 20      .ltext  EQUS "File to load; "
7651 0D                 EQUB &0D
7652 20 E7 FF   .save   JSR osnewl              \Output a carriage return
7655 A2 00              LDX #&00                \Prompt user for input
7657 BD C3 76   .sloop  LDA stext,X
765A C9 0D              CMP #&0D
765C F0 07              BEQ sout
765E 20 E3 FF           JSR osasci
7661 E8                 INX
7662 4C 57 76           JMP sloop
7665 A9 0C      .sout   LDA #&0C                \Input parameter - max. line length
7667 20 C0 77           JSR input
766A 90 03              BCC sover
766C 4C 30 75           JMP menu
766F A9 33      .sover  LDA #buffer MOD 256     \Set up a block for saving
7671 8D 00 01           STA block               \Block 0-1 address of filename
7674 A9 7B              LDA #buffer DIV 256
7676 8D 01 01           STA block+1
7679 A9 00              LDA #&00                \Block 2-5 load address
767B 8D 02 01           STA block+2
767E 8D 03 01           STA block+3
7681 8D 04 01           STA block+4
7684 8D 05 01           STA block+5
7687 8D 06 01           STA block+6             \Block 6-9 execution address
768A 8D 07 01           STA block+7
768D 8D 08 01           STA block+8
7690 8D 09 01           STA block+9
```

```
7693 A5 70              LDA textstart          \Block 10-13 start of data to save
7695 8D 0A 01           STA block+10
7698 A5 71              LDA textstart+1
769A 8D 0B 01           STA block+11
769D A9 00              LDA #&00
769F 8D 0C 01           STA block+12
76A2 8D 0D 01           STA block+13
76A5 A5 72              LDA textend            \Block 14-17 byte after end of data
76A7 8D 0E 01           STA block+14
76AA A5 73              LDA textend+1
76AC 8D 0F 01           STA block+15
76AF A9 00              LDA #&00
76B1 8D 10 01           STA block+16
76B4 8D 11 01           STA block+17
76B7 A9 00              LDA #&00               \OSFILE command (save file)
76B9 A2 00              LDX #block MOD 256
76BB A0 01              LDY #block DIV 256
76BD 20 DD FF           JSR osfile
76C0 4C 30 75           JMP menu
76C3 53 61 76
     65 20 61
     73 3A 20   .stext  EQUS "Save as: "
76CC 0D                 EQUB &0D
76CD A9 FF         .edit LDA #&FF               \Mark textend with rogue value &FF
76CF A0 00              LDY #&00
76D1 91 72              STA (textend),Y
76D3 20 41 78      .estart JSR display
76D6 A9 1F              LDA #&1F               \Display cursor
76D8 20 EE FF           JSR oswrch
76DB A5 7C              LDA hpos
76DD 20 EE FF           JSR oswrch
76E0 A5 7D              LDA vpos
76E2 20 EE FF           JSR oswrch
76E5 20 E0 FF           JSR osrdch             \Input a character from keyboard
76E8 C9 88              CMP #&88               \Case A of ...
76EA D0 06              BNE enotleft
76EC 20 98 78           JSR cleft
76EF 4C D3 76           JMP estart
76F2 C9 89         .enotleft CMP #&89
76F4 D0 06              BNE enotright
76F6 20 DE 78           JSR cright
76F9 4C D3 76           JMP estart
76FC C9 8A         .enotright CMP #&8A
76FE D0 06              BNE enotdown
7700 20 12 79           JSR cdown
7703 4C D3 76           JMP estart
7706 C9 8B         .enotdown CMP #&8B
7708 D0 06              BNE enotup
770A 20 79 79           JSR cup
770D 4C D3 76           JMP estart
7710 C9 7F         .enotup CMP #&7F
7712 D0 06              BNE enotrub
7714 20 27 7A           JSR rubout
7717 4C D3 76           JMP estart
771A C9 0D         .enotrub CMP #&0D
771C F0 08              BEQ eover
771E C9 20              CMP #&20
```

```
7720 90 0A              BCC enotwrite
7722 C9 7F              CMP #&7F
7724 B0 06              BCS enotwrite
7726 20 74 7A   .eover  JSR write
7729 4C D3 76           JMP estart
772C C9 1B      .enotwrite CMP #&1B
772E D0 08              BNE enoerr
7730 A9 7E              LDA #&7E            \Acknowledge ESCAPE condition
7732 20 F4 FF           JSR osbyte
7735 4C 30 75           JMP menu
7738 4C D3 76   .enoerr JMP estart
773B A9 0C      .print  LDA #&0C            \Clear the screen
773D 20 EE FF           JSR oswrch
7740 A9 02              LDA #&02            \Turn the printer on
7742 20 EE FF           JSR oswrch
7745 A5 70              LDA textstart       \Set up text pointers
7747 85 76              STA temp
7749 A5 71              LDA textstart+1
774B 85 77              STA temp+1
774D A5 72              LDA textend
774F 85 78              STA temp+2
7751 A5 73              LDA textend+1
7753 85 79              STA temp+3
7755 A0 00              LDY #&00
7757 8A         .ploop  TXA                 \Store registers
7758 48                 PHA
7759 98                 TYA
775A 48                 PHA
775B A9 79              LDA #&79            \Check if ESCAPE is being pressed
775D A2 F0              LDX #&F0
775F 20 F4 FF           JSR osbyte
7762 E0 00              CPX #&00
7764 10 11              BPL pok
7766 A9 03              LDA #&03            \It is, so turn the printer off
7768 20 EE FF           JSR oswrch
776B A9 15              LDA #&15            \and flush the printer buffer
776D A2 03              LDX #&03
776F 20 F4 FF           JSR osbyte
7772 68                 PLA                 \Clean up the stack
7773 68                 PLA
7774 4C 30 75           JMP menu
7777 68         .pok    PLA                 \Retrieve registers
7778 A8                 TAY
7779 68                 PLA
777A AA                 TAX
777B A5 76              LDA temp            \Have we reached textend?
777D C5 78              CMP temp+2
777F D0 06              BNE pover
7781 A5 77              LDA temp+1
7783 C5 79              CMP temp+3
7785 F0 0E              BEQ pout            \Yes, so return
7787 B1 76      .pover  LDA (temp),Y
7789 20 E3 FF           JSR osasci          \Print the character
778C E6 76              INC temp            \Move to next character
778E D0 02              BNE pnoinc
7790 E6 77              INC temp+1
7792 4C 57 77   .pnoinc JMP ploop
```

```
7795 A9 03      .pout     LDA #&03              \Turn the printer off
7797 20 EE FF             JSR oswrch
779A 20 E7 FF             JSR osnewl
779D 20 ED 77             JSR getkey            \Wait for a key press
77A0 4C 30 75             JMP menu
77A3 20 E7 FF   .oscom    JSR osnewl            \Output a carriage return
77A6 A9 2A                LDA #ASC("*")         \and an asterisk as a prompt
77A8 20 E3 FF             JSR osasci
77AB A9 FF                LDA #&FF              \Input parameter - max, line length
77AD 20 C0 77             JSR input
77B0 A2 33                LDX #buffer MOD 256   \OSCLI command
77B2 A0 7B                LDY #buffer DIV 256
77B4 20 F7 FF             JSR oscli
77B7 20 E7 FF             JSR osnewl
77BA 20 ED 77             JSR getkey            \Wait for a key press
77BD 4C 30 75             JMP menu
77C0 48         .input    PHA                   \Save parameter on the stack
77C1 A9 33                LDA #buffer MOD 256   \Set up a block for input
77C3 8D 00 01             STA block             \Block 0-1 address of buffer
77C6 A9 7B                LDA #buffer DIV 256
77C8 8D 01 01             STA block+1
77CB 68                   PLA                   \Block 2 maximum line length
77CC 8D 02 01             STA block+2
77CF A9 20                LDA #ASC(" ")         \Block 3 min, acceptable ASCII value
77D1 8D 03 01             STA block+3
77D4 A9 7F                LDA #&7F              \Block 4 max, acceptable ASCII value
77D6 8D 04 01             STA block+4
77D9 A9 00                LDA #&00              \OSWORD command (input string)
77DB A2 00                LDX #block MOD 256
77DD A0 01                LDY #block DIV 256
77DF 20 F1 FF             JSR osword
77E2 90 08                BCC iout              \Check if ESCAPE was pressed
77E4 A9 7E                LDA #&7E              \Acknowledge ESCAPE condition
77E6 20 F4 FF             JSR osbyte
77E9 4C 30 75             JMP menu
77EC 60         .iout     RTS
77ED A2 00      .getkey   LDX #&00
77EF BD 0A 78   .gloop    LDA gtext,X           \Prompt user for a key press
77F2 F0 07                BEQ gout
77F4 20 E3 FF             JSR osasci
77F7 E8                   INX
77F8 4C EF 77             JMP gloop
77FB 20 E0 FF   .gout     JSR osrdch            \Wait for a key press
77FE 90 09                BCC gout2             \Check if ESCAPE was pressed
7800 C9 1B                CMP #&1B
7802 D0 05                BNE gout2
7804 A9 7E                LDA #&7E              \Acknowledge ESCAPE condition
7806 20 F4 FF             JSR osbyte
7809 60         .gout2    RTS
780A 50 72 65
     73 73 20
     61 6E 79
     20 6B 65
     79 2E      .gtext    EQUS "Press any key,"
7818 00                   EQUB &00
7819 20 E7 FF   .break    JSR osnewl            \Output a carriage return
781C A0 01                LDY #&01              \Output error message
```

```
781E B1 FD      .bloop    LDA (retaddr),Y
7820 F0 07                BEQ bout
7822 C8                   INY
7823 20 E3 FF             JSR osasci
7826 4C 1E 78             JMP bloop
7829 A9 20      .bout     LDA #ASC(" ")          \Output a dash
782B 20 E3 FF             JSR osasci
782E A9 2D                LDA #ASC("-")
7830 20 E3 FF             JSR osasci
7833 A9 20                LDA #ASC(" ")
7835 20 E3 FF             JSR osasci
7838 20 ED 77             JSR getkey             \Wait for a key press
783B 68                   PLA                    \Discard unwanted return address
783C 68                   PLA
783D 68                   PLA                    \and status register
783E 4C 30 75             JMP menu
7841 A5 74      .display  LDA mempos             \Start displaying from mempos
7843 85 76                STA temp
7845 A5 75                LDA mempos+1
7847 85 77                STA temp+1
7849 A9 00                LDA #&00               \Set end of text flag to false
784B 85 78                STA temp+2
784D A9 19                LDA #lines             \Set line count to lines
784F 85 79                STA temp+3
7851 A9 00                LDA #screenst MOD 256  \Define screen pointer
7853 85 7A                STA temp+4
7855 A9 7C                LDA #screenst DIV 256
7857 85 7B                STA temp+5
7859 A0 00                LDY #&00
785B A2 28      .dloop    LDX #length            \Set row count to length
785D B1 76      .dloop2   LDA (temp),Y           \Get character from text
785F C9 FF                CMP #&FF               \Text finished? (&FF is a rogue value)
7861 D0 05                BNE dnotend
7863 85 78                STA temp+2             \Yes, so set end of text flag to true
7865 4C 70 78             JMP dspaces
7868 C9 0D      .dnotend  CMP #&0D               \Is there a carriage return in text?
786A F0 04                BEQ dspaces
786C 24 78                BIT temp+2             \Is the end of text flag set?
786E F0 10                BEQ dover              \No, so skip dspaces
7870 A9 20      .dspaces  LDA #ASC(" ")          \Print spaces to end of line
7872 91 7A      .dsloop   STA (temp+4),Y
7874 E6 7A                INC temp+4             \Update counters
7876 D0 02                BNE dsnoinc
7878 E6 7B                INC temp+5
787A CA         .dsnoinc  DEX
787B D0 F5                BNE dsloop
787D 4C 89 78             JMP dout
7880 91 7A      .dover    STA (temp+4),Y         \Print current character
7882 CA                   DEX                    \Update counters
7883 E6 7A                INC temp+4
7885 D0 02                BNE dout
7887 E6 7B                INC temp+5
7889 E6 76      .dout     INC temp
788B D0 02                BNE dnoinc
788D E6 77                INC temp+1
788F E0 00      .dnoinc   CPX #&00               \Have we printed a whole line yet?
7891 D0 CA                BNE dloop2             \No, so do next character
```

```
9                DEC temp+3             \Have we finished yet?
4                BNE dloop              \No, so do next line
                 RTS
E      .cleft    LDA curpos             \Are we at textstart?
0                CMP textstart
7                BNE clok
F                LDA curpos+1
1                CMP textstart+1
1                BNE clok
                 RTS                    \Yes, so return
E      .clok     LDA curpos             \Decrement curpos
                 SEC
1                SBC #&01
E                STA curpos
F                LDA curpos+1
0                SBC #&00
F                STA curpos+1
C                LDA hpos               \Are we in the top left hand corner?
E                BNE clover2
D                LDA vpos
0                CMP #&00
6                BNE clover
1 79             JSR sdown              \Yes, so scroll down
4 78             JMP clover2
D      .clover   DEC vpos               \Decrement vpos
0      .clover2  LDY #&00               \Is there a carriage return at curpos?
E                LDA (curpos),Y
D                CMP #&0D
6                BNE clnocr
2 7A             JSR nlcalc             \Yes, so get correct value for hpos
C                STA hpos
                 RTS
C      .clnocr   LDA hpos               \Are we in the leftmost column?
3                BEQ clover3
C                DEC hpos               \No, so decrement hpos
                 RTS
7      .clover3  LDA #length-1          \Set hpos to far right
C                STA hpos
                 RTS
0      .cright   LDY #&00               \Are we at the end of text?
E                LDA (curpos),Y
F                CMP #&FF
1                BNE crok
                 RTS                    \Yes, so return
E      .crok     INC curpos             \Increment curpos
2                BNE crnoinc
F                INC curpos+1
D      .crnoinc  CMP #&0D               \Is there a carriage return at curpos?
7                BNE crnocr
0                LDA #&00               \Yes, so set hpos to far left
C                STA hpos
5 79             JMP crscroll           \Scroll up if necessary
C      .crnocr   LDA hpos               \Are we in the rightmost column?
7                CMP #length-1
3                BEQ crover2
C                INC hpos               \No, so increment hpos
                 RTS
```

```
7901 A9 00      .crover2   LDA #&00            \Set hpos to far left
7903 85 7C                 STA hpos
7905 A5 7D      .crscroll  LDA vpos            \Are we on the bottom line?
7907 C9 18                 CMP #lines-1
7909 D0 04                 BNE crover3
790B 20 0D 7A              JSR sup             \Yes, so scroll up
790E 60                    RTS
790F E6 7D      .crover3   INC vpos            \Increment vpos
7911 60                    RTS
7912 A9 28      .cdown     LDA #length         \Set limit to length-hpos
7914 38                    SEC
7915 E5 7C                 SBC hpos
7917 85 76                 STA temp
7919 A0 00                 LDY #&00            \Set count to zero
791B B1 7E      .cdloop    LDA (curpos),Y      \Is there a carriage return at curpos?
791D C9 0D                 CMP #&0D
791F F0 0D                 BEQ cdout2          \Yes, so exit loop
7921 C9 FF                 CMP #&FF            \Have we reached textend?
7923 F0 08                 BEQ cdout           \Yes, so return
7925 C8                    INY                 \Update counter
7926 C4 76                 CPY temp            \Has count reached limit?
7928 F0 04                 BEQ cdout2          \Yes, so exit both loops
792A 4C 1B 79              JMP cdloop
792D 60         .cdout     RTS
792E A5 7D      .cdout2    LDA vpos            \Are we on the bottom line?
7930 C9 18                 CMP #lines-1
7932 D0 06                 BNE cdover
7934 20 0D 7A              JSR sup             \Yes, so scroll up
7937 4C 3C 79              JMP cdover2
793A E6 7D      .cdover    INC vpos            \Increment vpos
793C A9 28      .cdover2   LDA #length         \Set count to length-hpos
793E 38                    SEC
793F E5 7C                 SBC hpos
7941 AA                    TAX
7942 A0 00                 LDY #&00
7944 B1 7E      .cdloop2   LDA (curpos),Y      \Is there a carriage return at curpos?
7946 C9 0D                 CMP #&0D
7948 F0 0C                 BEQ cdout3          \Yes, so exit loop
794A E6 7E                 INC curpos          \Increment curpos
794C D0 02                 BNE cdnoinc
794E E6 7F                 INC curpos+1
7950 CA         .cdnoinc   DEX                 \Update counter
7951 F0 0B                 BEQ cdloop3         \and exit loop if finished
7953 4C 44 79              JMP cdloop2
7956 A2 00      .cdout3    LDX #&00            \Set count to zero
7958 E6 7E                 INC curpos          \and increment curpos
795A D0 02                 BNE cdloop3
795C E6 7F                 INC curpos+1
795E E4 7C      .cdloop3   CPX hpos            \Has count reached hpos?
7960 F0 16                 BEQ cdout5          \Yes, so exit loop
7962 B1 7E                 LDA (curpos),Y      \Is there a carriage return at curpos?
7964 C9 0D                 CMP #&0D
7966 F0 0E                 BEQ cdout4          \Yes, so exit loop
7968 C9 FF                 CMP #&FF            \Have we reached textend?
796A F0 0A                 BEQ cdout4          \Yes, so exit loop
796C E6 7E                 INC curpos          \Increment curpos
796E D0 02                 BNE cdnoinc2
```

```
                INC curpos+1
    .cdnoinc2   INX                 \Update counter
79              JMP cdloop3
    .cdout4     STX hpos            \Set hpos to count
    .cdout5     RTS
    .cup        LDA mempos          \Are we on the top line?
                CMP textstart
                BNE cuok
                LDA mempos+1
                CMP textstart+1
                BNE cuok
                LDA vpos
                BNE cuok
                RTS                 \Yes, so return
    .cuok       LDX #length         \Set count to length
                LDY #&00
    .culoop     LDA curpos          \Decrement curpos
                SEC
                SBC #&01
                STA curpos
                LDA curpos+1
                SBC #&00
                STA curpos+1
                DEX                 \Update counter
                BEQ cunocr          \and exit loop if finished
                LDA (curpos),Y      \Is there a carriage return at curpos?
                CMP #&0D
                BNE culoop          \No, so loop back
7A              JSR nlcalc          \Get column number of carriage return
                CMP hpos            \Is it greater than hpos?
                BCC cuover
                BEQ cuover
                SEC                 \Yes, so calculate new value for count
                SBC hpos
                TAX
79              JMP culoop          \and loop back
    .cuover     STA hpos            \Store correct value in hpos
    .cunocr     LDA vpos            \Are we on the top line?
                BNE cuover2
79              JSR sdown           \Yes, so scroll down
                RTS
    .cuover2    DEC vpos            \Decrement vpos
                RTS
    .sdown      LDA mempos          \Decrement mempos
                SEC
                SBC #&01
                STA mempos
                LDA mempos+1
                SBC #&00
                STA mempos+1
                LDY #&00
                LDA (mempos),Y      \Is there a carriage return at mempos?
                CMP #&0D
                BNE sdover
                LDA curpos          \Yes, so store curpos
                PHA
                LDA curpos+1
```

```
79DB 48                 PHA
79DC A5 74              LDA mempos          \and transfer mempos into curpos
79DE 85 7E              STA curpos          \(nlcalc reads parameter from curpos)
79E0 A5 75              LDA mempos+1
79E2 85 7F              STA curpos+1
79E4 20 D2 7A           JSR nlcalc          \Get column number of carriage return
79E7 85 76              STA temp
79E9 68                 PLA                 \Restore curpos
79EA 85 7F              STA curpos+1
79EC 68                 PLA
79ED 85 7E              STA curpos
79EF A5 74              LDA mempos          \Subtract column number from mempos
79F1 38                 SEC
79F2 E5 76              SBC temp
79F4 85 74              STA mempos
79F6 A5 75              LDA mempos+1
79F8 E9 00              SBC #&00
79FA 85 75              STA mempos+1
79FC 4C 0C 7A           JMP sdout
79FF A5 74      .sdover LDA mempos          \Set mempos to mempos-(length-1)
7A01 38                 SEC
7A02 E9 27              SBC #length-1
7A04 85 74              STA mempos
7A06 A5 75              LDA mempos+1
7A08 E9 00              SBC #&00
7A0A 85 75              STA mempos+1
7A0C 60         .sdout  RTS
7A0D A2 27      .sup    LDX #length-1       \Set count to length-1
7A0F A0 00              LDY #&00
7A11 B1 74      .suloop LDA (mempos),Y      \Is there a carriage return at mempos?
7A13 C9 0D              CMP #&0D
7A15 F0 09              BEQ suout           \Yes, so exit loop
7A17 E6 74              INC mempos          \Increment mempos
7A19 D0 02              BNE suover
7A1B E6 75              INC mempos+1
7A1D CA         .suover DEX                 \Update counter
7A1E D0 F1              BNE suloop          \and exit loop if finished
7A20 E6 74      .suout  INC mempos          \Increment mempos
7A22 D0 02              BNE suout2
7A24 E6 75              INC mempos+1
7A26 60         .suout2 RTS
7A27 A5 7E      .rubout LDA curpos          \Are we at textstart?
7A29 C5 70              CMP textstart
7A2B D0 0C              BNE rok
7A2D A5 7F              LDA curpos+1
7A2F C5 71              CMP textstart+1
7A31 D0 06              BNE rok
7A33 A9 07              LDA #&07            \Yes, so beep and return
7A35 20 EE FF           JSR oswrch
7A38 60                 RTS
7A39 20 98 78   .rok    JSR cleft           \Move cursor left
7A3C A5 7E              LDA curpos          \Copy curpos into temp
7A3E 85 76              STA temp
7A40 18                 CLC                 \and copy curpos+1 into temp+2
7A41 69 01              ADC #&01
7A43 85 78              STA temp+2
7A45 A5 7F              LDA curpos+1
```

```
7A47 85 77              STA temp+1
7A49 69 00              ADC #&00
7A4B 85 79              STA temp+3
7A4D A0 00              LDY #&00
7A4F B1 78      .rloop  LDA (temp+2),Y      \Copy contents of temp+2 into temp
7A51 91 76              STA (temp),Y
7A53 C9 FF              CMP #&FF            \Have we reached textend?
7A55 F0 0F              BEQ rout
7A57 E6 76              INC temp            \Increment temp
7A59 D0 02              BNE rnoinc
7A5B E6 77              INC temp+1
7A5D E6 78      .rnoinc INC temp+2          \Increment temp+2
7A5F D0 02              BNE rnoinc2
7A61 E6 79              INC temp+3
7A63 4C 4F 7A   .rnoinc2 JMP rloop
7A66 A5 72      .rout   LDA textend         \Decrement textend
7A68 38                 SEC
7A69 E9 01              SBC #&01
7A6B 85 72              STA textend
7A6D A5 73              LDA textend+1
7A6F E9 00              SBC #&00
7A71 85 73              STA textend+1
7A73 60                 RTS
7A74 48         .write  PHA                 \Store A (holds value of key pressed)
7A75 A5 72              LDA textend         \Have we run out of memory?
7A77 C9 FF              CMP #last MOD 256
7A79 D0 0D              BNE wok
7A7B A5 73              LDA textend+1
7A7D C9 74              CMP #last DIV 256
7A7F D0 07              BNE wok
7A81 A9 07              LDA #&07            \Yes, so beep, retrieve A and return
7A83 20 EE FF           JSR oswrch
7A86 68                 PLA
7A87 60                 RTS
7A88 A5 72      .wok    LDA textend         \Copy textend into temp
7A8A 85 76              STA temp
7A8C 18                 CLC                 \and copy textend+1 into temp+2
7A8D 69 01              ADC #&01
7A8F 85 78              STA temp+2
7A91 A5 73              LDA textend+1
7A93 85 77              STA temp+1
7A95 69 00              ADC #&00
7A97 85 79              STA temp+3
7A99 A0 00              LDY #&00
7A9B B1 76      .wloop  LDA (temp),Y        \Copy contents of temp into temp+2
7A9D 91 78              STA (temp+2),Y
7A9F A5 76              LDA temp            \Decrement temp
7AA1 38                 SEC
7AA2 E9 01              SBC #&01
7AA4 85 76              STA temp
7AA6 A5 77              LDA temp+1
7AA8 E9 00              SBC #&00
7AAA 85 77              STA temp+1
7AAC A5 78              LDA temp+2          \Decrement temp+2
7AAE 38                 SEC
7AAF E9 01              SBC #&01
7AB1 85 78              STA temp+2
```

```
7AB3 A5 79              LDA temp+3
7AB5 E9 00              SBC #&00
7AB7 85 79              STA temp+3
7AB9 A5 78              LDA temp+2          \Have we reached curpos?
7ABB C5 7E              CMP curpos
7ABD D0 DC              BNE wloop
7ABF A5 79              LDA temp+3
7AC1 C5 7F              CMP curpos+1
7AC3 D0 D6              BNE wloop
7AC5 68                 PLA                 \Yes, so retrieve A
7AC6 91 7E              STA (curpos),Y      \Insert character
7AC8 E6 72              INC textend         \Increment textend
7ACA D0 02              BNE wnoinc
7ACC E6 73              INC textend+1
7ACE 20 DE 78  .wnoinc  JSR cright          \Move cursor right
7AD1 60                 RTS
7AD2 A9 00     .nlcalc  LDA #&00            \Start count at zero
7AD4 85 76              STA temp
7AD6 85 77              STA temp+1
7AD8 A5 7E              LDA curpos          \Copy curpos into temp+2
7ADA 85 78              STA temp+2
7ADC A5 7F              LDA curpos+1
7ADE 85 79              STA temp+3
7AE0 A5 70              LDA textstart       \Copy textstart-1 into temp+4
7AE2 38                 SEC
7AE3 E9 01              SBC #&01
7AE5 85 7A              STA temp+4
7AE7 A5 71              LDA textstart+1
7AE9 E9 00              SBC #&00
7AEB 85 7B              STA temp+5
7AED A0 00              LDY #&00
7AEF E6 76     .nloop   INC temp            \Update counter
7AF1 D0 02              BNE nnoinc
7AF3 E6 77              INC temp+1
7AF5 A5 78     .nnoinc  LDA temp+2          \Decrement copy of curpos
7AF7 38                 SEC
7AF8 E9 01              SBC #&01
7AFA 85 78              STA temp+2
7AFC A5 79              LDA temp+3
7AFE E9 00              SBC #&00
7B00 85 79              STA temp+3
7B02 B1 78              LDA (temp+2),Y      \Is there a carriage return there?
7B04 C9 0D              CMP #&0D
7B06 F0 0C              BEQ nout            \Yes, so exit loop
7B08 A5 78              LDA temp+2          \Has copy of curpos hit textstart-1?
7B0A C5 7A              CMP temp+4
7B0C D0 E1              BNE nloop           \No, so loop back
7B0E A5 79              LDA temp+3
7B10 C5 7B              CMP temp+5
7B12 D0 DB              BNE nloop
7B14 A5 77     .nout    LDA temp+1          \Is count>length+1?
7B16 D0 06              BNE nover
7B18 A5 76              LDA temp
7B1A C9 29              CMP #length+1
7B1C 90 10              BCC nout2
7B1E A5 76     .nover   LDA temp            \Yes, so subtract length from count
7B20 38                 SEC
```

```
7B21 E9 28              SBC #length
7B23 85 76              STA temp
7B25 A5 77              LDA temp+1
7B27 E9 00              SBC #&00
7B29 85 77              STA temp+1
7B2B 4C 14 7B           JMP nout
7B2E C6 76      .nout2  DEC temp                \Decrement temp
7B30 A5 76              LDA temp                \Return with wanted value
7B32 60                 RTS
```