Routine EDIT programmi Basic Applesoft

di Marco Merler - Gardolo (TN)

Una delle maggiori lacune riscontrabili nel Basic Applesoft è senz'altro la mancanza di un buon metodo di EDITING per la correzione degli errori. Correggere una riga diventa più difficile che fare un programma complesso fra copiare, correggere, aggiungere, copiare ancora, cancellare... Per non parlare di quando la riga è molto lunga o contiene delle stringhe o peggio ancora di quando è necessario inserire un comando all'interno di una riga. Molte volte è più conveniente riscrivere addirittura la riga per risparmiare tempo!!!

Ecco quindi una breve routine in ASSEMBLER che facilita notevolmente il lavoro di edit in un programma consentendo di risparmiare una notevole quantità di tempo. Se aggiungiamo che resta disponibile il vecchio sistema e che la routine "ruba" solo 512 byte di memoria RAM allora non si capisce veramente come mai non ci abbia pensato la Apple!!! Misteri dell'informatica ...

Istruzioni per il caricamento

Per utilizzare la routine è necessario entrare in monitor con l'istruzione CALL -151 e caricarla byte per byte ad iniziare dalla locazione $9400 come indicato in figura 1. Consiglio di caricare i byte a gruppi di 8 o 16 e controllare, mano a mano, che i dati inseriti siano esatti. Naturalmente questo lavoro si deve fare una sola volta. Terminata questa operazione è necessario salvare la routine, in linguaggio macchina con l'istruzione:

```
BSAVE EDIT. OBJ0, A$9400, L1FF
```

Quindi bisogna caricare il programma di inizializzazione di figura 2 (pag. 102) con lo stesso sistema, ad iniziare dalla locazione $341, e salvarlo con l'istruzione:

```
BSAVE EDINIT. OBJ0, A$340, L$80
```

Infine è necessario caricare il programma BASIC di figura 3 (pag. 102) che esegue un controllo sull'esattezza dei dati inseriti, e provvede all'inizializzazione della routi-

I minifloppy con i programmi per Apple II

Presso la redazione sono disponibili i minifloppy con alcuni dei programmi pubblicati nella rubrica di software per Apple II. Il prezzo è di 15.000 lire IVA compresa per ciascun minifloppy. Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla Technimedia srl, Via Valsolda 135, 00141 Roma

Minifloppy disponibili

codice	programma	MC n.
DA2/01	Motomuro	24
DA2/02	&DEBUG	28
DA3/03	EDIT Applesoft + INPUT senza INPUT	29

```
0000:            7 *
0000:            8 *ROUTINE DELL'INTERPRETE USATE NEL PROGRAMMA
0000:            9 *
FD0C:           10 RDK      EQU  $FD0C      ;ROUTINE STANDARD DI INPUT
FC10:           11 BS       EQU  $FC10      ;ARRETRA IL CURSORE DI 1 POSIZIONE
FBFD:           12 VIDOUT   EQU  $FBFD      ;FA L'OUTPUT SU VIDEO DEL CARATTERE
0000:           13 ;CONTENUTO NELL'ACCUMULATORE
FDED:           14 COUT     EQU  $FDED      ;ROUTINE STANDARD DI OUTPUT
FBF4:           15 ADV      EQU  $FBF4      ;FA AVANZARE IL CURSORE DI 1 POSIZ.
DD67:           16 FRMNUM   EQU  $DD67      ;VALUTA UNA FORMULA NUMERICA
E752:           17 GETADR   EQU  $E752
0000:           18 *
0000:           19 *LOCAZIONI RAM USATE DAL PROGRAMMA E DAL BASIC*
0000:           20 *
0050:           21 LINUML   EQU  $50         ;CONTIENE IL NUMERO DI RIGA TROVATO
0000:           22 ;DA GETADR
0006:           23 BUFF     EQU  $06         ;USATO COME MEMORIA DI LAVORO
0000:           24 ;INSIEME A $07 E $08
0240:           25 BUFFER   EQU  $240        ;INIZIO DEL BUFFER DELLA RIGA
0019:           26 NCHAR    EQU  $19         ;NUMERO CARATTERI DELLA RIGA
0038:           27 KSWL     EQU  $38         ;POINTER DELLA ROUTINE DI INPUT
0000:           28 ;STANDARD DEL BASIC
00F9:           29 YSAV1    EQU  $F9         ;MEMORIA PROVV. PER IL REGISTRO Y
0024:           30 CH       EQU  $24         ;INDIRIZZO ORIZZONTALE DEL CURSORE
0025:           31 CV       EQU  $25         ;INDIRIZZO VERTICALE DEL CURSORE
0341:           32 MESSAGE  EQU  $341        ;INIZIO DEL MESSAGGIO DI LINE NOT FOUND
----- NEXT OBJECT FILE NAME IS EDITDOC.OBJ0
9400:           34          ORG  $9400
                35 ********************************************
9400:           36 *INIZIO ROUTINE*
9400:           37 *1 RICERCA PARAMETRI*
9400:20 67 DD   38          JSR  FRMNUM     ;RICERCA NUMERO DI RIGA
9403:20 52 E7   39          JSR  GETADR     ;LO PORTA NELLE LOCAZIONI $50-51
9406:A9 00      40 RIC      LDA  #$00       ;INIZIALIZZA A ZERO IL
9408:85 19      41          STA  NCHAR      ;CONTATORE DI CARATTERI
940A:           42 *2 RICERCA RIGA*
940A:A5 67      43          LDA  $67
940C:85 06      44          STA  BUFF       ;CARICA IN $06-$07 L'INIZIO
940E:A5 68      45          LDA  $68        ;DEL PROGRAMMA BASIC
9410:85 07      46          STA  BUFF+1     ;CONTENUTO IN $67-$68
9412:A0 02      47 NEWLINE  LDY  #$02
9414:B1 06      48          LDA  (BUFF),Y   ;EFFETTUA LA SCANSIONE DEL
9416:C5 50      49          CMP  LINUML     ;PROGRAMMA PER CERCARE LA LINEA
9418:D0 07      50          BNE  NONTROVA
941A:C8         51          INY
941B:B1 06      52          LDA  (BUFF),Y
941D:C5 51      53          CMP  LINUML+1
941F:F0 22      54          BEQ  TROVATO    ;SALTA A TROVATO SE LA RICERCA E' OK
9421:A0 00      55 NONTROVA LDY  #$00       ;CONTROLLA SE CI SONO ALTRE LINEE
9423:B1 06      56          LDA  (BUFF),Y   ;DI PROGRAMMA
9425:85 08      57          STA  BUFF+2
9427:C8         58          INY
9428:B1 06      59          LDA  (BUFF),Y
942A:F0 09      60          BEQ  RIGAINES   ;SALTA A RIGAINES SE IL PROGRAMMA E'
942C:85 07      61          STA  BUFF+1     ;FINITO ALTRIMENTI ANALIZZA LA
942E:A5 08      62          LDA  BUFF+2     ;PROSSIMA RIGA (SALTO A NEWLINE)
9430:85 06      63          STA  BUFF
9432:4C 12 94   64          JMP  NEWLINE
9435:A0 00      65 RIGAINES LDY  #$00       ;EMETTE IL MESSAGGIO
9437:B9 41 03   66 NEWCHAR  LDA  MESSAGE,Y  ;DI LINE NOT FOUND SE IL PROGRAMMA
943A:20 ED FD   67          JSR  COUT       ;E' FINITO SENZA
943D:C8         68          INY             ;TROVARE LA RIGA RICHIESTA
943E:C0 11      69          CPY  #17
9440:D0 F5      70          BNE  NEWCHAR
9442:60         71          RTS
9443:           72 *3 COPIATURA RIGA IN BUFFER ASCII*
9443:A9 19      73 TROVATO  LDA  #>CHINBF    ;MODIFICA IL PUNTATORE DELLA
9445:85 36      74          STA  $36         ;ROUTINE DI OUTPUT DEL
9447:A9 95      75          LDA  #<CHINBF    ;MONITOR PER USARE LA ROUTINE
9449:85 37      76          STA  $37
944B:20 EA 03   77          JSR  $03EA
944E:A6 50      78          LDX  LINUML
9450:A5 51      79          LDA  LINUML+1

9452:20 24 ED   80          JSR  $ED24      ;APPLESOFT $ED24
9455:A9 F0      81          LDA  #$F0       ;CHE TRASFORMA IL NUMERO DI RIGA
9457:85 36      82          STA  $36        ;IN FORMATO DECIMALE E LO PORTA
9459:A9 FD      83          LDA  #$FD       ;NEL BUFFER DELLA ROUTINE
945B:85 37      84          STA  $37
945D:20 EA 03   85          JSR  $03EA
9460:A0 04      86          LDY  #$04
9462:84 F9      87 NEWBYTE  STY  YSAV1      ;ANALIZZA LA RIGA BYTE PER BYTE
9464:B1 06      88          LDA  (BUFF),Y
9466:F0 39      89          BEQ  EDIT       ;SE FINE RIGA SALTA A EDIT
9468:10 2E      90          BPL  CHAR       ;SE E' UN CARATTERE ASCII SALTA A CHAR
946A:29 7F      91          AND  #$7F       ;TRASFORMA IL CODICE DELLA PAROLA
946C:AA         92          TAX             ;CHIAVE BASIC IN UNA STRINGA ASCII
946D:A9 CE      93          LDA  #$CE       ;SFRUTTANDO LA TABELLA DELL'INTERPRETE

946F:85 81      94          STA  $81        ;BASIC PRESENTE IN MEMORIA
9471:A9 D0      95          LDA  #$D0
9473:85 82      96          STA  $82
9475:A0 00      97          LDY  #$00
9477:B1 81      98 NEW3     LDA  ($81),Y
9479:10 03      99          BPL  MZ1
947B:CA        100          DEX
947C:30 09     101          BMI  MZ2
947E:E6 81     102 MZ1      INC  $81
9480:D0 02     103          BNE  MZ3
9482:E6 82     104          INC  $82
9484:4C 77 94  105 MZ3      JMP  NEW3
9487:A9 A0     106 MZ2      LDA  #$A0       ;AGGIUNGE UNO SPAZIO PRIMA DI OGNI
9489:20 19 95  107          JSR  CHINBF     ;PAROLA CHIAVE COME PER IL LIST
948C:C8        108 NEW4     INY
948D:B1 81     109          LDA  ($81),Y
948F:48        110          PHA
9490:20 19 95  111          JSR  CHINBF
9493:68        112          PLA
9494:10 F6     113          BPL  NEW4
9496:A9 A0     114          LDA  #$A0       ;AGGIUNGE UNO SPAZIO DOPO OGNI
9498:20 19 95  115 CHAR     JSR  CHINBF     ;PAROLA CHIAVE
949B:A4 F9     116          LDY  YSAV1      ;MEMORIZZA I CARATTERI ASCII
949D:C8        117          INY             ;NEL BUFFER PER ESSERE MODIFICATI
949E:4C 62 94  118          JMP  NEWBYTE
94A1:A6 19     119 EDIT     LDX  NCHAR      ;MEMORIZZA UN CARATTERE ASCII NULLO
94A3:9D 40 02  120          STA  BUFFER,X   ;PER SEGNARE LA FINA DELLA RIGA
94A6:20 5D 95  121          JSR  RESTORE    ;CANCELLA LE ULTIME RIGHE DEL VIDEO
94A9:20 42 FC  122          JSR  $FC42
94AC:C6 19     123          DEC  NCHAR
94AE:20 67 95  124 EDITZ    JSR  LIST       ;STAMPA IL CONTENUTO DEL BUFFER SU
94B1:          125 *5 EDIT* ;VIDEO
94B1:F0 06     126          BEQ  CHANGE
94B3:20 25 95  127 MZ8      JSR  CHINBF1
94B6:20 67 95  128 CHANGE1  JSR  LIST
94B9:20 0C FD  129 CHANGE   JSR  RDK        ;LEGGE LA TASTIERA
94BC:C9 9B     130          CMP  #$9B       ;E' STATO PREMUTO L'ESC?
94BE:F0 1D     131          BEQ  INSERT     ;SI', SALTA A INSERT
94C0:C9 92     132          CMP  #'R'+128   ;E'STATO PREMUTO IL <CTRL R>?
94C2:D0 03     133          BNE  CONT
94C4:4C 06 94  134          JMP  RIC        ;SI', SALTA A RIC
94C7:C9 8D     135 CONT     CMP  #$8D       ;PREMUTO <RETURN>?
94C9:F0 61     136          BEQ  END        ;SALTA A END
94CB:48        137          PHA
94CC:20 DC 95  138          JSR  MOVIMENTI  ;GESTISCE GLI EVENTUALI MOVIMENTI
94CF:68        139          PLA             ;DEL CURSORE
94D0:C9 A0     140          CMP  #$A0       ;BLOCCA I CARATTERI DI CONTROLLO
94D2:30 E5     141          BMI  CHANGE     ;SE E' UN CARATTERE ASCII
94D4:E4 19     142          CPX  NCHAR      ;INSERISCE NEL BUFFER IL NUOVO
94D6:D0 DB     143          BNE  MZ8
94D8:9D 40 02  144          STA  BUFFER,X   ;CARATTERE
94DB:F0 D9     145          BEQ  CHANGE1
94DD:20 0C FD  146 INSERT   JSR  RDK        ;ROUTINE CHE GESTISCE L'INSERZIONE
94E0:C9 9B     147          CMP  #$9B       ;DI UN CARATTERE NEL BUFFER
94E2:F0 D5     148          BEQ  CHANGE
94E4:C9 88     149          CMP  #$88
94E6:D0 08     150          BNE  MZ9        ;SE E' STATA PREMUTA LA FRECCIA
94E8:20 C1 95  151          JSR  BACK       ;INDIETRO CANCELLA IL CARATTERE
```

ne. Attenzione: il controllo avviene sommando le varie locazioni di memoria della routine. Può perciò succedere, come per la "famosa" prova del 9, che due errori si annullino a vicenda.

A questo punto si dà il run e, se non succede niente, la routine è già pronta per l'uso, altrimenti il programma emette un messaggio d'errore, e la routine deve essere corretta o ricaricata. A questo punto si possono cancellare le righe dal numero 20 al numero 40 e salvare il programma Basic con il nome p.e. di EDIT.

```
SAVE EDIT
```

Se tutto è andato bene, per attivare la routine è adesso sufficiente battere:

```
RUN EDIT
```

È possibile naturalmente inserire le istruzioni Basic sopracitate nel programma che il DOS esegue al bootstrap in modo che l'Apple II le esegua automaticamente all'accensione.

Istruzioni per l'uso della routine

Per editare una riga è sufficiente battere l'istruzione:

```
& NN
```

dove NN è il numero di riga da correggere. A questo punto la riga appare sul video, come per un normale LIST, tranne per i caratteri di controllo che sono scritti in negativo, e con il cursore che si posiziona sul primo carattere. Per muovere il cursore e posizionarlo nel punto voluto, si usano le solite frecce e i caratteri ‹CTRL A› e ‹CTRL Z› rispettivamente per salire e per scendere. Dopo essersi posizionati sul punto desiderato, è possibile manipolare la riga con i seguenti comandi:

CHANGE: per sostituire uno o più caratteri all'interno della riga è sufficiente semplicemente posizionarsi sul carattere errato e ribattere quello corretto. Attenzione: se una volta giunti a fine riga si continua ad inserire dei caratteri, questi continuano a sostituirsi all'ultimo premuto.

INSERT: per inserire dei caratteri bisogna posizionarsi sul punto dove si desidera fare l'inserimento e premere ‹ESC› per entrare nel modo Insert.

Attenzione: nel modo Insert, l'unico carattere di controllo cursore disponibile è la freccia indietro che cancella l'ultimo carattere inserito.

Per uscire dal modo Insert e tornare nel modo Edit normale premere nuovamente ‹ESC›.

DELETE: per cancellare un carattere è sufficiente posizionarsi sul carattere da cancellare e premere ‹CTRL D›

RESTORE: premendo ‹CTRL R› si cancellano tutte le modifiche effettuate, e la riga viene riportata nel buffer di edit nella sua forma originaria pronta per essere corretta. Premere questo tasto quindi, quando si sono fatte delle modifiche non volute.

END: per terminare l'edit e riportare la riga corretta nel programma Basic premere ‹RETURN›.

Come funziona

Come non è necessario sapere come funziona al suo interno un computer per usarlo, così alla maggior parte degli utilizzatori di questa routine non interesserà minimamente sapere come funziona.

Tuttavia per chi avesse intenzione di trascorrere alcune notti in bianco a studiare il listato sorgente pubblicato in queste pagine (poveretti!!) per scoprire come funziona la routine, e magari modificarla, dirò che essa è composta essenzialmente di 3 parti:

1) preleva la riga da correggere dal programma Basic in memoria, e la porta nel "buffer" di correzione situato a partire dalla locazione $240, trasformandola opportunamente in una stringa di caratteri ASCII. Ricordo infatti che in memoria le istruzioni Basic sono "tokenizzate", cioè sostituite da un numero maggiore di 127, come riportato dall'appendice F del manuale di riferimento Applesoft, ed è perciò necessario provvedere alla loro decodificazione ($946A - $9496).

```
94EB:20 8C 95  152          JSR  DELETE    ;PRECEDENTEMENTE INSERITO
94EE:A9 80     153          LDA  #$80
94F0:C9 A0     154 MZ9      CMP  #$A0
94F2:30 E9     155          BMI  INSERT
94F4:85 08     156          STA  BUFF+2
94F6:E6 19     157          INC  NCHAR     ;SPOSTA IN AVANTI TUTTI I CARATTERI
94F8:E6 19     158          INC  NCHAR
94FA:86 09     159          STX  BUFF+3    ;FINO AL FINE RIGA PER FAR POSTO
94FC:A4 09     160          LDY  BUFF+3    ;AL CARATTERE DA INSERIRE
94FE:B9 40 02  161 OTHER    LDA  BUFFER,Y
9501:48        162          PHA
9502:A5 08     163          LDA  BUFF+2
9504:99 40 02  164          STA  BUFFER,Y
9507:68        165          PLA
9508:85 08     166          STA  BUFF+2
950A:C8        167          INY
950B:C4 19     168          CPY  NCHAR
950D:F0 EF     169          BEQ  OTHER
950F:90 ED     170          BCC  OTHER
9511:E8        171          INX            ;SPOSTA IN AVANTI IL CURSORE
9512:20 67 95  172          JSR  LIST      ;STAMPA SU VIDEO LA RIGA COSI'
9515:C6 19     173          DEC  NCHAR
9517:D0 C4     174          BNE  INSERT    ;MODIFICATA
9519:          175 *ROUTINE METTE CHAR IN BUFF*
9519:86 09     176 CHINBF   STX  BUFF+3    ;SUBROUTINE CHE METTE NEL BUFFER IL
951B:A6 19     177          LDX  NCHAR     ;CARATTERE PRESENTE NELL'ACCUMULATORE

951D:20 25 95  178          JSR  CHINBF1   ;AGGIORNANDO IL CONTATORE DI CARATTERI

9520:86 19     179          STX  NCHAR     ;E LASCIANDO INALTERATO IL REGISTRO X

9522:A6 09     180          LDX  BUFF+3
9524:60        181 RTS2     RTS
9525:09 80     182 CHINBF1  ORA  #$80
9527:9D 40 02  183          STA  BUFFER,X  ;IDEM COME SOPRA TRANNE CHE MODIFICA
952A:E8        184          INX            ;IL REGISTRO X
952B:60        185          RTS
952C:20 5D 95  186 END      JSR  RESTORE   ;ROUTINE CHE PROVVEDE ALL'INSERIMENTO
DELLA
952F:A9 3F     187          LDA  #>INPUT   ;DELLA RIGA CORRETTA NEL PROGRAMMA
9531:85 38     188          STA  KSWL      ;BASIC MODIFICANDO IL PUNTATORE DELLA

9533:A9 95     189          LDA  #<INPUT   ;ROUTINE DI INPUT ($38-$39) PERCHE'
9535:85 39     190          STA  KSWL+1    ;ACCETTI L'INPUT DALLA SEGUENTE
9537:20 EA 03  191          JSR  $03EA
953A:A9 00     192          LDA  #$00
953C:85 06     193          STA  BUFF
953E:60        194          RTS
953F:86 07     195 INPUT    STX  BUFF+1    ;ROUTINE CHE INVIA ALL'INTERPRETE
9541:A6 06     196          LDX  BUFF      ;BASIC I CARATTERI COMPONENTI LA RIGA

9543:BD 40 02  197          LDA  BUFFER,X  ;QUANDO LA RIGA E' FINITA
9546:C9 80     198          CMP  #$80      ;SPEDISCE UN CR E RIPRISTINA I
9548:D0 0D     199          BNE  NONCR     ;PUNTATORI DELLA ROUTINE DI INPUT
954A:A9 1B     200          LDA  #$1B      ;ALLA ROUTINE CHE LEGGE LA TASTIERA
954C:85 38     201          STA  KSWL
954E:A9 FD     202          LDA  #$FD
9550:85 39     203          STA  KSWL+1
9552:20 EA 03  204          JSR  $03EA
9555:A9 8D     205          LDA  #$8D
9557:E8        206 NONCR    INX
9558:86 06     207          STX  BUFF
955A:A6 07     208          LDX  BUFF+1
955C:60        209          RTS
955D:A9 11     210 RESTORE  LDA  #17       ;PORTA IL CURSORE A CAPO DELLA 17 RIGA

955F:85 25     211          STA  CV
9561:A9 00     212          LDA  #$00
9563:85 24     213          STA  CH
9565:AA        214          TAX
9566:60        215          RTS
9567:          216 *LIST*   ;STAMPA SU VIDEO IL CONTENUTO DEL BUFFER
9567:86 08     217 LIST     STX  BUFF+2    ;LASCIANDO INALTERATO IL REGISTRO X

9569:20 5D 95  218          JSR  RESTORE   ;E LA POSIZIONE DEL CURSORE
956C:20 62 FC  219          JSR  $FC62
956F:BD 40 02  220 LOOP1    LDA  BUFFER,X
9572:F0 0C     221          BEQ  ENDL      ;CONTROLLA LA FINE DELLA RIGA
9574:C9 A0     222          CMP  #$A0
9576:10 02     223          BPL  VIDEO
9578:29 7F     224          AND  #$7F      ;STAMPA IN NEGATIVO I CARATTERI DI
957A:20 FD FB  225 VIDEO    JSR  VIDOUT    ;CONTROLLO E IN POSITIVO GLI ALTRI
957D:E8        226          INX
957E:D0 EF     227          BNE  LOOP1
9580:20 9C FC  228 ENDL     JSR  $FC9C
9583:20 10 FC  229 BX       JSR  BS        ;RIPORTA IL CURSORE AL SUO POSTO
9586:CA        230          DEX
9587:E4 08     231          CPX  BUFF+2
9589:D0 F8     232          BNE  BX
958B:60        233          RTS
958C:86 09     234 DELETE   STX  BUFF+3    ;GESTISCE LA CANCELLAZIONE DI UN
958E:E4 19     235          CPX  NCHAR     ;CARATTERE DAL BUFFER SPOSTANDO
9590:F0 1F     236          BEQ  LAST      ;ALL'INSIETRO TUTTI I CARATTERI RESTAN
TI
9592:A4 19     237          LDY  NCHAR
9594:A9 00     238          LDA  #$00
9596:85 08     239          STA  BUFF+2
9598:B9 40 02  240 LOOP5    LDA  BUFFER,Y
959B:48        241          PHA
959C:A5 08     242          LDA  BUFF+2
959E:99 40 02  243          STA  BUFFER,Y
95A1:68        244          PLA
95A2:85 08     245          STA  BUFF+2
95A4:88        246          DEY
95A5:F0 04     247          BEQ  DECN
95A7:C4 09     248          CPY  BUFF+3
95A9:B0 ED     249          BCS  LOOP5
95AB:C6 19     250 DECN     DEC  NCHAR
95AD:20 67 95  251 EDIT3    JSR  LIST      ;STAMPA LA RIGA COSI' OTTENUTA
95B0:60        252          RTS
95B1:A9 A0     253 LAST     LDA  #$A0      ;SE IL CARATTERE E' L'ULTIMO INVECE
95B3:9D 40 02  254          STA  BUFFER,X  ;DI CANCELLARLO LO SOSTITUISCE
95B6:D0 F5     255          BNE  EDIT3     ;CON UNO SPAZIO
95B8:E4 19     256 ADVANCE  CPX  NCHAR     ;GESTISCE L'AVANZAMENTO DEL CURSORE
95BA:F0 04     257          BEQ  RTS3      ;DI UNA POSIZIONE
95BC:20 F4 FB  258          JSR  ADV       ;CONTROLLANDO LA FINE DELLA RIGA
95BF:E8        259          INX
95C0:60        260 RTS3     RTS
95C1:E0 00     261 BACK     CPX  #$00      ;GESTISCE L'ARRETRAMENTO DEL CURSORE
95C3:F0 FB     262          BEQ  RTS3      ;DI UNA POSIZIONE CONTROLLANDO
95C5:20 10 FC  263          JSR  BS        ;L'INIZIO RIGA
95C8:CA        264          DEX
95C9:60        265          RTS
95CA:A0 28     266 UP       LDY  #40       ;SPOSTA IL CURSORE VERSO L'ALTO DI UNA

95CC:20 C1 95  267 MZ5      JSR  BACK      ;RIGA, SE NON E' POSSIBILE LO PORTA
95CF:88        268          DEY            ;A INIZIORIGA
95D0:D0 FA     269          BNE  MZ5
95D2:60        270          RTS
95D3:A0 28     271 DOWN     LDY  #40       ;SPOSTA IL CURSORE VERSO IL BASSO DI
95D5:20 B8 95  272 MZ6      JSR  ADVANCE   ;UNA RIGA SE NON E' POSSIBILE LO PORTA

95D8:88        273          DEY            ;A FINE RIGA
95D9:D0 FA     274          BNE  MZ6
95DB:60        275          RTS
95DC:C9 95     276 MOVIMENTI CMP #21+128   ;CONTROLLA SE SONO STATI PREMUTI I
95DE:F0 D8     277          BEQ  ADVANCE   ;TASTI RELATIVI AI VARI MOVIMENTI DEL

95E0:C9 88     278          CMP  #08+128   ;CURSORE E LI ESEGUE DI CONSEGUENZA
95E2:F0 DD     279          BEQ  BACK
95E4:C9 81     280          CMP  #'A'-64
95E6:F0 E2     281          BEQ  UP
95E8:C9 9A     282          CMP  #'Z'-64
95EA:F0 E7     283          BEQ  DOWN
95EC:C9 84     284          CMP  #'D'-64
95EE:F0 9C     285          BEQ  DELETE
95F0:60        286          RTS
```

```
9400- 20 67 DD 20 52 E7 A9 00
9408- 85 19 A5 67 85 06 A5 68
9410- 85 07 A0 02 B1 06 C5 50
9418- D0 07 C8 B1 06 C5 51 F0
9420- 22 A0 00 B1 06 85 08 C8
9428- B1 06 F0 09 85 07 A5 08
9430- 85 06 4C 12 94 A0 00 B9
9438- 41 03 20 ED FD C8 C0 17
9440- D0 F5 60 A9 19 85 36 A9
9448- 95 85 37 20 EA 03 A6 50
9450- A5 51 20 24 ED A9 F0 85
9458- 36 A9 FD 85 37 20 EA 03
9460- A0 04 84 F9 B1 06 F0 39
9468- 10 2E 29 7F AA A9 CE 85
9470- 81 A9 D0 85 82 A0 00 B1
9478- 81 10 03 CA 30 09 E6 81
9480- D0 02 E6 82 4C 77 94 A9
9488- A0 20 19 95 C8 B1 81 48
9490- 20 19 95 68 10 F6 A9 A0
9498- 20 19 95 A4 F9 C8 4C 62
94A0- 94 A6 19 9D 40 02 20 5D
94A8- 95 20 42 FC C6 19 20 67
94B0- 95 F0 06 20 25 95 20 67
94B8- 95 20 0C FD C9 9B F0 1D
94C0- C9 92 D0 03 4C 06 94 C9
94C8- 8D F0 61 48 20 DC 95 68
94D0- C9 A0 30 E5 E4 19 D0 DB
94D8- 9D 40 02 F0 D9 20 0C FD
94E0- C9 9B F0 D5 C9 88 D0 08
94E8- 20 C1 95 20 8C 95 A9 80
94F0- C9 A0 30 E9 85 08 E6 19
94F8- E6 19 86 09 A4 09 B9 40
9500- 02 48 A5 08 99 40 02 68
9508- 85 08 C8 C4 19 F0 EF 90
9510- ED E8 20 67 95 C6 19 D0
9518- C4 86 09 A6 19 20 25 95
9520- 86 19 A6 09 60 09 80 9D
9528- 40 02 E8 60 20 5D 95 A9
9530- 3F 85 38 A9 95 85 39 20
9538- EA 03 A9 00 85 06 60 86
9540- 07 A6 06 BD 40 02 C9 00
9548- D0 0D A9 1B 85 38 A9 FD
9550- 85 39 20 EA 03 A9 8D E8
9558- 86 06 A6 07 60 A9 11 85
9560- 25 A9 00 85 24 AA 60 86
9568- 08 20 5D 95 20 62 FC BD
9570- 40 02 F0 0C C9 A0 10 02
9578- 29 7F 20 FD FB E8 D0 EF
9580- 20 9C FC 20 10 FC CA E4
9588- 08 D0 F8 60 86 09 E4 19
9590- F0 1F A4 19 A9 00 85 08
9598- B9 40 02 48 A5 08 99 40
95A0- 02 68 85 08 88 F0 04 C4
95A8- 09 B0 ED C6 19 20 67 95
95B0- 60 A9 A0 9D 40 02 D0 F5
95B8- E4 19 F0 04 20 F4 FB E8
95C0- 60 E0 00 F0 FB 20 10 FC
95C8- CA 60 A0 28 20 C1 95 88
95D0- D0 FA 60 A0 28 20 B8 95
95D8- 88 D0 FA 60 C9 95 F0 D8
95E0- C9 88 F0 DD C9 81 F0 E2
95E8- C9 9A F0 E7 C9 84 F0 9C
95F0- 60
```

Figura 1 - Codice oggetto del programma EDIT.

```
0340- 60 8D CC C9 CE C5 A0 CE
0348- CF D4 A0 C6 CF D5 CE C4
0350- A0 C5 D2 D2 CF D2 87 8D
0358- 8D C1 D0 D0 CC C5 D3 CF
0360- C6 D4 A0 C5 C4 C9 D4 CF
0368- D2 A0 D2 CF D5 D4 C9 CE
0370- C5 A0 B1 AE B0 8D A8 C3
0378- A9 A0 B1 B9 B8 B3 A0 C2
0380- D9 A0 CD C5 D2 CC C5 D2
0388- A0 CD C1 D2 C3 CF 8D 00
0390- A9 4C 8D F5 03 A9 94 8D
0398- F7 03 A9 00 8D F6 03 A8
03A0- B9 58 03 F0 06 20 ED FD
03A8- C8 D0 F5 60
```

Figura 2

2) provvede alla correzione della riga nel buffer seguendo i comandi dell'operatore indicati sopra.

3) quando l'operatore preme ‹return› riporta la riga corretta nel programma Basic sostituendola alla precedente.

Per evitare una gestione del video complessa la routine modifica esclusivamente il buffer e provvede a ristamparlo, completamente, dopo ogni modifica. Viene inoltre mantenuto un cursore interno che indica il carattere sul quale è posizionato il cursore video ed è rappresentato dal registro X. Tutte le modifiche (Change, Insert, Delete etc..) vengono eseguite sul carattere indicato da tale registro. La breve routine situata a partire dalla locazione $340 serve a stampare i messaggi di errore, di copyright e a collegare la routine al simbolo &.

Detto questo penso che il listato dovrebbe essere comprensibile, per la maggior parte delle istruzioni, a chiunque abbia un minimo di conoscenza del linguaggio macchina del 6502.

AVVERTENZE

1) I caratteri di controllo vengono stampati sul video in negativo. Possono venire cancellati o sostituiti con caratteri normali, ma *né nel modo normale, né nel modo insert possono essere inseriti caratteri di controllo nella linea da editare*. La routine li rifiuterà automaticamente.

2) Per motivi che è difficile spiegare, è impossibile cancellare il primo carattere con il tasto ‹CTRL D›. Pertanto posizionandosi con il cursore sul primo carattere, e premendo tale tasto *verrà cancellato il secondo carattere*. È però possibile cancellare il primo carattere, sostituendo ad esso uno spazio.

3) Con questa versione non è possibile editare righe di più di 240 caratteri. Per superare tale limite è sufficiente dare il comando POKE 38238,16. In ogni modo però è necessario ricordarsi di non superare i 255 caratteri, perché la routine non esegue alcun controllo sulla lunghezza, e i caratteri in più andrebbero persi con risultati imprevedibili.

L'input è meglio senza INPUT

Se avete provato ad usare la INPUT del Basic per creare un file EXEC o per inserire una lista di nomi e numeri in un programma di archivio, per esempio una rubrica, vi sarà certamente capitato di veder apparire la scritta EXTRA IGNORED alla pressione del tasto di Return. Questo succede quando la riga digitata contiene una virgola, che la funzione INPUT utilizza come separatore tra gli argomenti.

Per ovviare all'inconveniente si deve fare ricorso ad una complicata e pesante gestione dell'input da tastiera mediante l'uso della GET e il successivo concatenamento alla stringa in uso.

La routine che vi presentiamo consente, con una semplice CALL, di prelevare da tastiera una riga qualsiasi e di assegnarla ad una variabile stringa a piacere.

La routine è lunga più di 200 byte e questo ci impedisce di metterla nella solita pagina $3 a partire dalla locazione 768; così si è deciso di metterla in una zona di memoria che nessun programma andrà mai a sporcare: tra il DOS e i suoi BUFFER.

In realtà tra il DOS e i Buffer ci sono solo sette byte liberi, ma basta spostare i puntatori al buffer di 255 indietro per creare lo spazio necessario alla nostra routine. Il DOS, per trovare l'inizio del buffer, legge il valore contenuto nelle locazioni $9D00 e $9D01, la prima locazione contiene la parte bassa, la seconda la parte alta. La parte bassa non ci interessa in quanto la nostra routine entra esattamente in una pagina, ci basta quindi decrementare di uno la parte alta (locazione $9D01) per far indietreggiare di 255 byte i BUFFER del DOS. Una volta cambiato il valore si farà un JMP $3D3 per eseguire un COLDSTART del DOS e fargli accettare la modifica.

Non è purtroppo possibile lanciare in esecuzione una routine che si sovrappone ai buffer del DOS prima che si sia riservato lo spazio opportuno. Perciò la routine viene caricata a partire dalla locazione $3FED e una piccola routine iniziale provvede a spostare i puntatori dei buffer e trasferire il resto della routine nella zona da $9C00 a $9CFF. Attenzione al fatto che il JMP $3D3, che effettua il Coldstart del DOS, effettua anche un New con la perdita dei programmi in BASIC.

```
10  HIMEM: 37887: PRINT  CHR$ (4)"BLOAD EDIT.OBJ0": PRINT  CHR$ (4)"BLOAD
      EDINIT.OBJ0":TT = 0
20  FOR X = 37888 TO 38384:TT = TT +  PEEK (X): NEXT X: IF TT <  > 60004 THEN
      PRINT "ERRORE IN EDIT.OBJ0": CHR$ (7): END
25 TT = 0
30  FOR X = 833 TO 939:TT = TT +  PEEK (X): NEXT X: IF TT <  > 18724 THEN
      PRINT "ERRORE IN EDINIT.OBJ0": CHR$ (7): END
40  PRINT "ROUTINE O.K."
50  CALL 912
60  NEW
```

Figura 3

Una volta lanciata la routine apparentemente non accade nulla, ma se durante un programma in BASIC effettuiamo una CALL 39936, A$, l'Apple si predispone ad accettare una riga in INPUT e a depositarla nella variabile A$ (o un'altra a piacere). La riga può essere composta da qualsiasi carattere ed essere lunga al massimo 240 caratteri, un Bip ci segnala gli ultimi dieci caratteri. La tastiera è tipo macchina da scrivere, cioè in minuscolo, e con lo shift si ottengono le maiuscole. I caratteri speciali che si trovano sopra la P, la N e la M si ottengono premendo CTRL + SHIFT + il tasto desiderato; nello stesso modo si ottengono le lettere accentate. La pressione del CTRL X annulla la riga scritta e ripropone il prompt senza tornare al programma. Solo il tasto di RETURN consente di uscire dalla routine. Per le modifiche si usano le due frecce come al solito. Nella locazione 40050 si trova il codice ASCII del prompt (di default è un punto interrogativo) che viene presentato dalla routine di input, se non lo si vuole si può fare una POKE 40050,128; qualunque altro valore verrà interpretato come prompt e stampato in testa al rigo.

Chi non avesse (ancora?) la nostra EPROM Apple-Minus per le minuscole può fare a meno di una gran parte di programma; la routine semplificata (vedi figura 5) entra comodamente nella pagina tre.

In figura 6 trovate il listato in Basic di un programmino che serve per creare dei File EXEC, quei file cioè che vengono eseguiti dall'Apple come se il loro contenuto fosse direttamente battuto da un operatore sulla tastiera del computer.

```
3FED                1           ORG $3FED
3FED                2           OBJ $3FED
3FED                3 ;
3FED                4 ; ROUTINE DI INPUT
3FED                5 ; BY VALTER DI DIO
3FED                6 ;
3FED                7 LOCO    EPZ $E
3FED                8 DIFF    EPZ $0D
3FED                9 CH      EPZ $24
3FED               10 BASL    EPZ $28
3FED               11 LNPRG   EPZ $75
3FED               12 NXTINL  EPZ $7F
3FED               13 NXTINH  EPZ $80
3FED               14 VARL    EPZ $85
3FED               15 VARH    EPZ $86
3FED               16 NWLNL   EPZ $87
3FED               17 NWLNH   EPZ $88
3FED               18 OCHRL   EPZ $8B
3FED               19 OCHRH   EPZ $B9
3FED               20 ;
3FED               21 GETCHR  EQU $B1
3FED               22 NXTCHR  EQU $B7
3FED               23 ;
3FED               24 IN      EQU $200
3FED               25 SHIFT   EQU $C063
3FED               26 BUFFER  EQU $9D00
3FED               27 ;
3FED               28 DOS     EQU $3D3
3FED               29 LET     EQU $DA7B
3FED               30 COMMA   EQU $DEBE
3FED               31 RECLN   EQU $D533
3FED               32 FINDV   EQU $DFE3
3FED               33 STRNG   EQU $E3ED
3FED               34 VALUE   EQU $E73D
3FED               35 KEYIN   EQU $FD1B
3FED               36 COUT    EQU $FDED
3FED               37 CROUT   EQU $FD8E
3FED               38 CLREOL  EQU $FC9C
3FED               39 BELL    EQU $FF3A
3FED               40 ;
3FED A200          41          LDX #$0
3FEF BD0040        42 LOOP     LDA $4000,X
3FF2 9D009C        43          STA $9C00,X
3FF5 CA            44          DEX
3FF6 D0F7          45          BNE LOOP
3FF8 A99B          46          LDA #$9B
3FFA 8D019D        47          STA BUFFER+1
3FFD 4CD303        48          JMP DOS
4000               49 ;
9C00               50          ORG $9C00
9C00               51          OBJ $4000
9C00               52 ;
9C00 20BEDE        53          JSR COMMA
9C03 20719C        54          JSR GETLN
9C06 2033D5        55          JSR RECLN
9C09 AD0002        56          LDA IN
9C0C 8575          57          STA LNPRG
9C0E 867F          58          STX NXTINL
9C10 8480          59          STY NXTINH
9C12 20E3DF        60          JSR FINDV
9C15 8585          61          STA VARL
9C17 8486          62          STY VARH
9C19 A5B8          61          LDA OCHRL
9C1B A4B9          62          LDY OCHRH
9C1D 8587          63          STA NWLNL
9C1F 8488          64          STY NWLNH
9C21 A67F          65          LDX NXTINL
9C23 A480          66          LDY NXTINH
9C25 86B8          67          STX OCHRL
9C27 84B9          68          STY OCHRH
9C29 20B100        69          JSR GETCHR
9C2C A5B8          70          LDA OCHRL
9C2E A4B9          71          LDY OCHRH
9C30 850E          72          STA LOCO
9C32 850D          73          STA DIFF
9C34 20EDE3        74          JSR STRNG
9C37 203DE7        75          JSR VALUE
9C3A 207BDA        76          JSR LET
9C3D 20B700        77          JSR NXTCHR
9C40 A5B8          78          LDA OCHRL
9C42 A4B9          79          LDY OCHRH
9C44 857F          80          STA NXTINL
9C46 8480          81          STY NXTINH
9C48 A587          82          LDA NWLNL
9C4A A488          83          LDY NWLNH
9C4C 85B8          84          STA OCHRL
9C4E 84B9          85          STY OCHRH
9C50 60            86 END      RTS

9C51 BD0002        87 NOTCR    LDA IN,X
9C54 20EDFD        88          JSR COUT
9C57 C988          89          CMP #$88
9C59 F01D          90          BEQ BCKSPC
9C5B C998          91          CMP #$98
9C5D F00A          92          BEQ CANCELL
9C5F E0E5          93          CPX #$E5
9C61 D003          94          BNE NOTCR1
9C63 203AFF        95          JSR BELL
9C66 E8            96 NOTCR1   INX
9C67 D013          97          BNE NXTCH
9C69 A9DC          98 CANCEL   LDA #"\"
9C6B 20EDFD        99          JSR COUT
9C6E 20BEFD       100 GETLNZ   JSR CROUT
9C71 A9BF         101 GETLN    LDA #"?"
9C73 20EDFD       102          JSR COUT
9C76 A201         103          LDX #$1
9C78 8A           104 BCKSPC   TXA
9C79 F0F3         105          BEQ GETLNZ
9C7B CA           106          DEX
9C7C A424         107 NXTCH    LDY CH
9C7E B128         108          LDA (BASL),Y
9C80 48           109          PHA
9C81 C9E0         110          CMP #$E0
9C83 9002         111          BCC NOINV
9C85 291F         112          AND #$1F
9C87 293F         113 NOINV    AND #$3F
9C89 0940         114          ORA #$40
9C8B 9128         115          STA (BASL),Y
9C8D 68           116          PLA
9C8E 20A49C       117          JSR GETCH
9C91 C995         118          CMP #$95
9C93 D002         119          BNE ADDINP
9C95 B128         120          LDA (BASL),Y
9C97 9D0002       121 ADDINP   STA IN,X
9C9A C98D         122          CMP #$8D
9C9C D0B3         123          BNE NOTCR
9C9E 209CFC       124          JSR CLREOL
9CA1 4C8EFD       125          JMP CROUT
9CA4 201BFD       126 GETCH    JSR KEYIN
9CA7 C9C0         127          CMP #$C0
9CA9 B039         128          BCS MINUS
9CAB C9A0         129          CMP #$A0
9CAD 9001         130          BCC CTRL
9CAF 60           131          RTS
9CB0 2C63C0       132 CTRL     BIT SHIFT
9CB3 1001         133          BPL RTS1+1
9CB5 60           134 RTS1     RTS
9CB6 C985         135          CMP #$85
9CB8 D002         136          BNE C1
9CBA A9FD         137          LDA #$FD
9CBC C98F         138 C1       CMP #$8F
9CBE D002         139          BNE C2
9CC0 A9FC         140          LDA #$FC
9CC2 C995         141 C2       CMP #$95
9CC4 D002         142          BNE C3
9CC6 A9E0         143          LDA #$E0
9CC8 C989         144 C3       CMP #$89
9CCA D002         145          BNE C4
9CCC A9FE         146          LDA #$FE
9CCE C981         147 C4       CMP #$81
9CD0 D002         148          BNE C5
9CD2 A9FB         149          LDA #$FB
9CD4 C9B0         150 C5       CMP #$B0
9CD6 F009         151          BEQ CONV
9CD8 C99E         152          CMP #$9E
9CDA F005         153          BEQ CONV
9CDC C99D         154          CMP #$9D
9CDE F001         155          BEQ CONV
9CE0 60           156          RTS
9CE1 0940         157 CONV     ORA #$40
9CE3 60           158          RTS
9CE4 C9DE         159 MINUS    CMP #"^"
9CE6 F004         160          BEQ CONV1
9CE8 C9DD         161          CMP #"]"
9CEA D005         162          BNE RTS2+1
9CEC 0910         163 CONV1    ORA #$10
9CEE 29EF         164          AND #$EF
9CF0 60           165 RTS2     RTS
9CF1 C9C0         166          CMP #"@"
9CF3 D003         167          BNE RTS3+1
9CF5 0910         168          ORA #$10
9CF7 60           169 RTS3     RTS
9CF8 2C63C0       170          BIT SHIFT
9CFB 1002         171          BPL RTS4
9CFD 0920         172          ORA #$20
9CFF 60           173 RTS4     RTS
                  174          END
```

Figura 4 - Sorgente LISA del programma di INPUT. Se non si dispone di un assemblatore caricare i codici della seconda colonna a partire dalla locazione $3FED e salvare con il nome di INPUT. PLUS, A$ 3FED, L$ 115.

```
10  DIM A$(100):I = 0
20 D$ =  CHR$ (4): REM  CTRL D
30  HOME : TEXT
40  IF  PEEK (39937) +  PEEK (399
    38) = 412 THEN 60
50  PRINT D$"BRUN INPUT.OBJ"
60 I = I + 1
70  CALL 39936,A$(I)
110  IF A$(I) <  > "" GOTO 60
120  PRINT
130  INPUT "WHAT FILE NAME? ";N$
140  PRINT D$;"OPEN ";N$
150  PRINT D$"DELETE ";N$
160  PRINT D$;"OPEN ";N$
170  PRINT D$;"WRITE ";N$
190  FOR J = 1 TO I - 1
200  PRINT A$(J)
210  NEXT J
220  PRINT D$;"CLOSE ";N$
```

Figura 5 - Dump esadecimale della routine semplificata per chi non dispone della EPROM delle minuscole. Caricare a partire dalla locazione $300 e salvare con BSAVE INPUT. PLUS, A$ 300, L$BO.

```
0300-   20 BE DE    JSR   $DEBE
0303-   20 2C D5    JSR   $D52C
0306-   AD 00 02    LDA   $0200
0309-   85 75       STA   $75
030B-   86 7F       STX   $7F
030D-   84 80       STY   $80
030F-   20 E3 DF    JSR   $DFE3
0312-   85 85       STA   $85
0314-   84 86       STY   $86
0316-   A5 B8       LDA   $B8
0318-   A4 B9       LDY   $B9
031A-   85 87       STA   $87
031C-   84 88       STY   $88
031E-   A6 7F       LDX   $7F
0320-   A4 80       LDY   $80
0322-   86 B8       STX   $B8
0324-   84 B9       STY   $B9
0326-   20 B1 00    JSR   $00B1
0329-   A5 B8       LDA   $B8
032B-   A4 B9       LDY   $B9
032D-   85 0E       STA   $0E
032F-   85 0D       STA   $0D
0331-   20 ED E3    JSR   $E3ED
0334-   20 3D E7    JSR   $E73D
0337-   20 7B DA    JSR   $DA7B
033A-   20 B7 00    JSR   $00B7
033D-   A5 B8       LDA   $B8
033F-   A4 B9       LDY   $B9
0341-   85 7F       STA   $7F
0343-   84 80       STY   $80
0345-   A5 87       LDA   $87
0347-   A4 88       LDY   $88
0349-   85 B8       STA   $B8
034B-   84 B9       STY   $B9
034D-   60          RTS
```

Figura 6 - Programma in Basic per la creazione di file EXEC. La routine di INPUT. PLUS deve già essere in memoria!