

Auto numerazione per Basic Apple

I computer hanno eliminato molti dei lavori noiosi e ripetitivi; e quale lavoro è più noioso che scrivere ogni volta il numero di riga delle istruzioni del Basic?

Dal momento che il nostro Apple è in grado sia di contare che di scrivere facciamo in modo che sia lui a fare per noi anche questo lavoro.

Il trucco è semplice: intercettiamo la routine che preleva i caratteri dalla tastiera e vediamo se il primo carattere è uno spazio. Se sì, simuliamo l'entrata del numero di riga, altrimenti torniamo alla normale routine di input del Monitor.

Per caricare il programma converrebbe avere un assembler, anche il miniassembler dell'Apple andrebbe benissimo, oppure, con molta pazienza, dopo essere passati al Monitor, col solito CALL-151, copiarsi il listato di figura 2 facendo attenzione agli 8 e alle B che si somigliano molto.

Come si vede dal disassemblato di figura 1 il programma risiede nella zona alta della memoria, appena sotto il DOS (con MAXFILES = 3) e si auto protegge spostando HIMEM verso il basso.

Chi volesse rilocare il programma, perché non dispone di 48K o ha già occupato con altri programmi la pagina Hex 95, tro-

va in fig. 2a il codice oggetto da inserire a partire dalla locazione Hex 300.

Per sicurezza dopo aver caricato il programma, disassemblarlo battendo 953 AL o 300L e confrontarlo con la figura 1 (a parte le prime sette istruzioni e le locazioni dei dati, i due programmi sono pressoché identici).

Salvate il programma su disco battendo BSAVE AUTO.NUM, A\$953A, L\$C3 o BSAVE AUTO.NUN, A\$300, L\$BD.

Le ultime cinque locazioni del programma contengono i dati le prime due il numero di riga (parte bassa, parte alta) le due successive sono l'incremento. In tabella 1 sono riportati i valori decimali ed esadecimali delle locazioni utili di ambedue le versioni.

Dato che i dati sono caricati insieme al programma il loro valore al momento del salvataggio diventa il valore di default. Non fate quindi girare il programma prima di salvarlo su disco o, se lo fate, ricordatevi di rimettere a posto il valore del numero di riga con cui volete iniziare. Nel nostro caso i due valori sono: 100 per la prima riga e 10 per l'incremento.

Per uscire dall'autonumerazione si può battere IN#0 o chiamare la routine MANUAL facendo un CALL 38218 o 778 a seconda della versione usata.

953A-	A9	39	85	73	A9	51			
9540-	85	38	A9	95	85	74	85	39	
9548-	D0	03	20	89	FE	20	EA	03	
9550-	60	20	1B	FD	E0	00	D0	47	
9558-	C9	A0	D0	43	AE	F8	95	4C	
9560-	F9	95	20	D5	95	A2	01	8E	
9568-	FC	95	86	1E	CA	86	1F	A0	
9570-	02	B9	17	00	46	1E	B0	06	
9578-	E6	1E	4A	4A	4A	29	0F		
9580-	C5	1F	F0	0B	C6	1F	09	B0	
9588-	9D	00	02	E8	20	ED	FD	A5	
9590-	1E	D0	DE	88	10	DB	A5	1F	
9598-	F0	EA	A9	A0	A4	24	60	C9	
95A0-	98	D0	04	4E	FC	95	60	C9	
95A8-	8D	D0	29	4E	FC	95	90	24	
95B0-	18	AD	F8	95	6D	FA	95	8D	
95B8-	F8	95	AD	F9	95	6D	FB	95	
95C0-	8D	F9	95	C9	FA	90	0B	86	
95C8-	1E	84	1F	20	4A	95	A6	1E	
95D0-	A4	1F	A9	8D	60	86	1E	84	
95D8-	1F	A2	00	86	17	86	18	86	
95E0-	19	A0	0F	06	1E	26	1F	F8	
95E8-	A2	FD	B5	1A	75	1A	95	1A	
95F0-	E8	30	F7	D8	88	10	EC	60	
95F8-	64	00	0A	00	00				

Figura 2 - Codice oggetto del programma di figura 1.

0300-	A9	11	85	38	A9	03	85	39	
0308-	D0	03	20	89	FE	20	EA	03	
0310-	60	20	1B	FD	E0	00	D0	47	
0318-	C9	A0	D0	43	AE	B8	03	AC	
0320-	B9	03	20	95	03	A2	01	8E	
0328-	BC	03	86	1E	CA	86	1F	A0	
0330-	02	B9	17	00	46	1E	B0	06	
0338-	E6	1E	4A	4A	4A	29	0F		
0340-	C5	1F	F0	0B	C6	1F	09	B0	
0348-	9D	00	02	E8	20	ED	FD	A5	
0350-	1E	D0	DE	88	10	DB	A5	1F	
0358-	F0	EA	A9	A0	A4	24	60	C9	
0360-	98	D0	04	4E	FC	95	60	C9	
0368-	8D	D0	29	4E	FC	95	90	24	
0370-	18	AD	B8	03	6D	BA	03	8D	
0378-	B8	03	AD	B9	03	6D	BB	03	
0380-	8D	B9	03	C9	FA	90	0B	86	
0388-	1E	84	1F	20	4A	95	A6	1E	
0390-	A4	1F	A9	8D	60	86	1E	84	
0398-	1F	A2	00	86	17	86	18	86	
03A0-	19	A0	0F	06	1E	26	1F	F8	
03A8-	A2	FD	B5	1A	75	1A	95	1A	
03B0-	E8	30	F7	D8	88	10	EC	60	
03B8-	64	00	0A	00	00				

Figura 2a - Codice oggetto da inserire a partire dalla locazione \$300 se la pagina \$95 non è disponibile.

Nome	pag.	3	rilocato
INIZIO	300	768	953A 38202
MANUAL	30A	778	954A 38218
HIMEM	=	=	9539 38201
RIGAL	3B8	952	95F8 38392
RIGAH	3B9	953	95F9 38393
INCRH	3BA	954	95FA 38394
INCRH	3BB	955	95FB 38395
COUNT	3BC	956	95FC 38396

Tabella delle locazioni di memoria usate dalle due versioni del programma di auto-numerazione.

953A-	A9	39	LDA	##39	
953C-	85	73	STA	##73	
953E-	A9	51	LDA	##51	
9540-	85	38	STA	##38	
9542-	A9	95	LDA	##95	
9544-	85	74	STA	##74	
9546-	85	39	STA	##39	
9548-	D0	03	BNE	##954D	
954A-	20	89	JSR	##FE89	
954D-	20	EA	JSR	##03EA	
9550-	60		RTS		
9551-	20	1B	FD	JSR	##FD1B
9554-	E0	00	CPX	##E000	
9556-	D0	47	BNE	##955F	
9558-	C9	A0	CMP	##C9A0	
955A-	D0	43	BNE	##955F	
955C-	AE	F8	95	LDX	##95F8
955F-	AC	F9	95	LDY	##95F9
9562-	20	D5	95	JSR	##95D5
9565-	A2	01	LDX	##A201	
9567-	8E	FC	95	STX	##95FC
956A-	86	1E	STX	##861E	
956C-			CA	DEX	
956D-	86	1F	STX	##861F	
956F-	A0	02	LDY	##A002	
9571-	B9	17	00	LDA	##9017, Y
9574-	46	1E	LDX	##461E	
9576-	B0	06	BCC	##957E	
9578-	E6	1E	INC	##E61E	
957A-	4A		LDX	##4A	
957B-	4A		LDX	##4A	
957C-	4A		LDX	##4A	
957D-	4A		LDX	##4A	
957E-	29	0F	AND	##290F	
9580-	C5	1F	CMP	##C51F	
9582-	F0	0B	BEQ	##958F	
9584-	C6	1F	DEC	##C61F	
9586-	09	B0	ORA	##09B0	
9588-	9D	00	02	STA	##0200, X
958B-	E8		INX		
958C-	20	ED	FD	JSR	##FED0
958F-	A5	1E	LDA	##A51E	
9591-	D0	DE	BNE	##9571	
9593-	8E		DEY		
9594-	10	DB	BPL	##9571	
9596-	A5	1F	LDA	##A51F	
9598-	F0	EA	BEQ	##9584	
959A-	A9	A0	LDA	##A9A0	
959C-	A4	24	LDY	##A424	
959E-	60		RTS		
959F-	C9	98	CMP	##C998	
95A1-	D0	04	BNE	##95A7	
95A3-	4E	FC	95	LSR	##95FC
95A6-	60		RTS		
95A7-	C9	8D	CMP	##C98D	
95A9-	D0	29	BNE	##95D4	
95AB-	4E	FC	95	LSR	##95FC
95AE-	90	24	BCC	##95D4	
95B0-	18		CLC		
95B1-	AD	F8	95	LDA	##95F8
95B4-	6D	FA	95	ADC	##95FA
95B7-	8D	F8	95	STA	##95F8
95BA-	AD	F9	95	LDA	##95F9
95BD-	6D	FB	95	ADC	##95FB
95C0-	8D	F9	95	STA	##95F9
95C3-	C9	FA	CMP	##C9FA	
95C5-	90	0B	BCC	##95D2	
95C7-	86	1E	STX	##861E	
95C9-	84	1F	STY	##841F	
95CB-	20	4A	95	JSR	##954A
95CE-	A6	1E	LDX	##A61E	
95D0-	A4	1F	LDY	##A41F	
95D2-	A9	8D	LDA	##A98D	
95D4-	60		RTS		
95D5-	86	1E	STX	##861E	
95D7-	84	1F	STY	##841F	
95D9-	A2	00	LDX	##A200	
95DB-	86	17	STX	##8617	
95DD-	86	18	STX	##8618	
95DF-	86	19	STX	##8619	
95E1-	A0	0F	LDY	##A00F	
95E3-	06	1E	ASL	##061E	
95E5-	26	1F	ROL	##261F	
95E7-	F8		SED		
95E8-	A2	FD	LDX	##A2FD	
95EA-	B5	1A	LDA	##B51A, X	
95EC-	75	1A	ADC	##751A, X	
95EE-	95	1A	STA	##951A, X	
95F0-	E8		INX		
95F1-	30	F7	BMI	##95EA	
95F3-	D8		CLD		
95F4-	88		DEY		
95F5-	10	EC	BPL	##95E3	
95F7-	60		RTS		
95F8-	64		???		
95F9-	00		BRK		
95FA-	0A		ASL		
95FB-	00		BRK		
95FC-	00		BRK		

Figura 1 - Listato del programma di autonumerazione delle righe BASIC.

Cross-reference dictionary in Basic

Nei grossi elaboratori tipo IBM 360/370, a richiesta del programmatore, i compilatori forniscono una tabella con l'elenco delle variabili usate nel programma e il numero di riga in cui sono state usate. È anche possibile sapere in quali locazioni di memoria sono state messe e quale nome ha loro assegnato il compilatore.

Nell'APPLE esiste una tabella con i nomi delle variabili e i loro puntatori, ma dato che il programma non è compilato, se una variabile non viene usata realmente dal programma — ad esempio una uscita da una IF che non viene attivata — quella variabile naturalmente non compare nella lista. Quindi non possiamo usare questa tabella per costruire il nostro cross reference.

Se vogliamo allora una lista completa delle variabili che compaiono nel nostro programma, dobbiamo cercarcele direttamente nella memoria del programma. Partiamo quindi dalla prima locazione

(2048) e scorriamo tutto il programma alla ricerca delle variabili. Queste sono scritte in memoria nel solito codice ASCII; facile quindi distinguerle dalle istruzioni che l'APPLE memorizza in un solo Byte usando un codice speciale che già abbiamo presentato in un altro articolo. Appena appena più difficile è riconoscere un numero che appartiene ad una variabile da uno isolato — ad esempio distinguere 3 da A3 — ma dato che nessuna variabile inizia con un numero, basta un controllo al carattere precedente per eliminare ogni dubbio.

Il programma parte dal numero di riga 60000 per poter essere accodato a quello su cui deve lavorare. E si ferma da solo appena incontra il numero di riga 60000. Per usarlo, dopo averlo accodato col RE-NUMBER/ MERGE, battere RUN 60000. Dato che la lista può essere abbastanza lunga è consigliabile uscire direttamente sulla stampante. Oltre ai nomi delle variabili ed ai rispettivi numeri di linea, vengono listati anche tutti i REM.

```
50000 I = PEEK (103) - 1 + PEEK (104) * 256:
HOME = DIM V$(100),N$(100)
60010 Q = PEEK (I)
60020 IF Q = 34 THEN F = 1 - F
60030 IF F THEN 60110
60040 IF Q = 0 THEN 60120
60050 IF Q > 90 OR Q = 58 OR Q = 59 OR Q = 44
THEN 60120
60060 IF (Q < 59 OR Q = 40) AND A$ = "" THEN 6
0110
60070 IF Q < 32 THEN 60120
60080 IF Q = 41 THEN 60120
60090 A$ = A$ + CHR$(Q)
60100 IF Q = 40 THEN 60120
60110 I = I + 1: GOTO 60010
60120 IF A$ = "" THEN 60170
60130 FOR H = 0 TO J
60140 IF V$(H) = A$ THEN N$(H) = N$(H) + " " +
STR$(A): GOTO 60170
60150 NEXT
60160 IF A$ < > "" THEN J = J + 1: V$(J) = A$:
N$(J) = N$(J) + " " + STR$(A)
60170 A$ = "": IF Q THEN 60110
60180 I = I + 3: A = PEEK (I) + PEEK (I + 1) *
256
60190 IF A = 60000 THEN 60230
60200 VTAB 1: PRINT A
60210 A$ = ""
60220 I = I + 1: GOTO 60110
60230 VTAB 1: INPUT "SLOT DELLA STAMPANTE ?": S
L
60240 IF PEEK (994) + PEEK (1001) = 192 THEN
PRINT CHR$(4): "PR#": SL: GOTO 60260
60250 PR# SL
60260 FOR H = 0 TO J
60270 PRINT V$(H): " # "N$(H)
60280 NEXT
```

Incolonniamo i numeri!

È evidente che il BASIC, seppure bravo in aritmetica, non è andato a scuola! Infatti già dalle elementari ci viene spiegato che i numeri si incolonnano in base alla posizione del punto decimale, mentre il nostro computer si ostina ad incolonnarli a sinistra; con un effetto grafico penoso e una leggibilità orripilante! Non parliamo poi del caso in cui, raggiunto il margine destro, il povero numero viene addirittura mandato a capo!!

Per fortuna è abbastanza semplice fare un programmino che, tutte le volte che ci occorre un incolonnamento, provveda a darci una mano. Il programma è in figura 1 e si commenta da solo, vi do comunque alcune spiegazioni aggiuntive.

Le variabili usate sono:

F - contiene il numero di caratteri destinati al numero da stampare compresi i decimali, il punto, e gli spazi prece-

```
4995 REM CONVERTE R IN STRINGA
5000 R$ = STR$(R): DP = C
5005 REM CERCA IL PUNTO DECIMALE
5010 FOR SP = 1 TO LEN (R$): M$ = MID$(R$,SP,1)
5020 IF M$ = "." THEN DP = SP
5025 REM CONTROLLA CHE NON
5030 REM CI SIA L'ESPOSANTE
5035 IF M$ = "E" THEN SP = 0: GOTO 5090
5040 NEXT
5045 REM AGGIUNGE IL PUNTO
5050 REM AI NUMERI INTERI
5055 IF NOT DP THEN R$ = R$ + "."
5060 REM TRONCA I DECIMALI
5065 REM IN ECCEDEZZA
5070 R$ = LEFT$(R$,DP + D)
5075 REM AGGIUNGE GLI ZERI DOPO
5080 REM LE CIFRE SIGNIFICATIVE
5090 IF LEN (R$) < DP + D THEN R$ = R$ + "0":
GOTO 5070
5095 REM CALCOLA GLI SPAZI
5100 SP = F - D - DP
5105 REM CONTROLLA L'OVERFLOW
5110 IF SP < 1 THEN SP = F - 6: R$ = " -OVER-"
5115 REM *** STAMPA ***
5120 PRINT SPC( SP): R$:
5130 REM CONTROLLA IL NUMERO DI CARATTERI
5140 REM PER RIGA ED ESEGUE IL < C,R, >
5150 C = C + F: IF C > L - F THEN C = 0: PRINT
5120 RETURN
```

Figura 1

deni;
D - è il numero delle cifre decimali;
L - massimo numero di caratteri per riga;
C - è il contatore dei caratteri stampati;

R - è il numero da stampare;

R\$ - contiene R in modo che possa essere elaborato;

SP - viene usata per vari scopi, ma alla fine contiene il numero di spazi da anteporre ad R per ottenere l'incolonnamento corretto. Può essere usata anche altrove senza problemi.

Per usare questa subroutine occorre definire F e D (ricordatevi che F contiene D), stabilire il numero di caratteri per riga, mettere in R il numero da stampare ed effettuare un GOSUB 5000. Per iniziare la stampa da una posizione diversa dal margine basta dare un HTAB dopo il PRINT della riga 5120.

Se dovete scrivere qualcosa tra un numero e l'altro ricordatevi di aggiornare il contatore dei caratteri C.

Per la cronaca, molti basic sono provvisti dell'istruzione PRINT USING che consente l'incolonnamento dei numeri e la loro stampa "formattata".

Scritte che scorrono

Capita a volte, mentre ci si trova in pagina grafica mista a testo, di dover visualizzare dei risultati o delle notizie molto lunghe. Ora, nell'APPLE, le righe destinate al testo sono solo quattro, delle quali la prima proprio attaccata alla zona grafica. Se poi, per agevolare la lettura, vogliamo lasciare una interlinea, le righe utili si riducono a due.

Questo programmino, usato come subroutine, consente di far scorrere, sempre sullo stesso rigo, un testo lungo fino a 215 caratteri. Il testo è contenuto nella stringa A\$, ad essa viene premessa la stringa B\$, che contiene 39 blank necessari per far partire la scritta dal fondo della riga.

Dopodiché tutto il lavoro viene svolto dalla potentissima istruzione MIDS. Que-

sta consente di estrarre uno o più caratteri di una stringa cominciando da un punto qualsiasi, ad esempio: MIDS(A\$, 5, 10) estrae dieci caratteri di A\$ a partire dal quinto.

Grazie a questa istruzione è quindi faci-

```
LISTATO 1
1000 HOME
1010 FOR I = 0 TO 18: BS = BS + " ":NEXT
1020 AS = BS + AS + " "
1030 FOR I = 1 TO LEN( AS)
1040 VTAB 23: PRINT MIDS( AS,I,40)
1050 FOR D = 0 TO PDL( 0) * 2: NEXT
1060 NEXT
1070 RETURN

LISTATO 2
3000 FOR I = 1 TO LEN( AS): L = 15
3010 A = ASC( MIDS( AS,I,1)): IF A = 64 THEN
I = I + 1: L = 0: GOTO 3010
3020 IF A < 58 OR A > 90 THEN L = 0
3030 MS = MS + CHR$( A + L)
3040 NEXT
3050 AS = MS: MS = "": RETURN
```

lissimo aprire una finestra di 40 caratteri all'interno di A\$. Spostando poi il punto di partenza della finestra è possibile percorrere tutto il contenuto della stringa in esame.

Le righe 1030 e 1040 della subroutine 1000 fanno proprio questo lavoro. Il contenuto della finestra viene poi stampato sempre sulla medesima riga, sicché si abbia l'illusione dello scorrimento.

La riga 1050 introduce un ritardo, regolabile tramite la paddle zero, onde permettere una agevole lettura del testo; non volendo usare le paddle, si può sostituire PDL(0) * 2 con un valore tra 100 e 400 secondo i propri gusti.

Per chi ha già installato la EPROM MINUS, che permette l'uso delle minuscole sul video, la subroutine del listato 2 converte in minuscolo il contenuto di A\$, tranne quei caratteri che fossero preceduti dalla @.