

# APPLE Memo

Nicole Bréaud - Poullquen



GRUPPO  
EDITORIALE  
JACKSON

EDIZIONE  
ITALIANA



# **APPLE**

# **Memo**

**Nicole Bréaud - Pouliquen**



**GRUPPO  
EDITORIALE  
JACKSON**  
Via Rossetini, 12  
20124 Milano



© Copyright per l'edizione originale Edizione da P.S.I. 1981

© Copyright per l'edizione italiana Gruppo Editoriale Jackson - 1984

Il Gruppo Editoriale Jackson ringrazia per il prezioso lavoro svolto nella stesura dell'edizione italiana la signora Francesca Di Fiore, e l'ing. Roberto Pancaldi.

Traduzione italiana a cura di Piero Dell'Orco

Tutti i diritti sono riservati. Stampato in Italia. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

Fotocomposizione: Lineacomp S.r.l. - Via Rosellini, 12 - 20124 Milano

Stampato in Italia da:

S.p.A. Alberto Matarrelli - Milano - Stabilimento Grafico

## PRESENTAZIONE

*Questo libro è destinato a rimanere, in permanenza, a fianco del vostro Apple quando lo utilizzate. Esso vi ricorda tutte le informazioni relative a riferimenti che potranno essere reperite velocemente: sintassi dei comandi, codici dei caratteri, messaggi di errore, linguaggio macchina, connessioni e indirizzi utili.*

*Le informazioni sono date senza eccessivi dettagli poiché lo scopo principale di questo libro è quello di fornire un rapido accesso alle informazioni; per una esposizione più introduttiva e completa, si potrà far riferimento ai libri "Alla scoperta dell'Apple II" e "Apple II - Guida all'uso".*

*Si termina con una raccolta di "trucchi" di differenti livelli, ma tutti utili, i "come...?" sono disposti senza un ordine particolare, è comunque possibile il loro rapido reperimento grazie ad un apposito indice.*

*Saremo riconoscenti a tutti i lettori per i suggerimenti e le informazioni supplementari che potranno essere inseriti in una prossima edizione.*





# SOMMARIO

CAPITOLI	Pagina
I - COMANDI	
Funzioni del BASIC Applesoft .....	1
Istruzioni del BASIC Applesoft .....	5
Operatori BASIC .....	11
Altri comandi .....	12
Applesoft .....	13
Funzioni e istruzioni dell'Integer BASIC .....	15
Funzioni Integer BASIC .....	16
Istruzioni Integer BASIC .....	18
Sistema operativo Pascal-UCSD .....	21
Editor Pascal-UCSD .....	23
Sistema operativo Pascal-UCSD: gestione dei file .....	24
I comandi del Monitor .....	28
Mini-assemblatore .....	31
I comandi del sistema operativo DOS .....	33
II - CARATTERI	
Conversione esadecimale/decimale/esadecimale .....	40
I codici della tastiera .....	41
I codici dello schermo .....	42
ASCII (positivo) (bit 7 = 0) .....	43
ASCII (negativo) (bit 7 = 1) .....	46
Conversione esadecimale/decimale .....	47
III - MESSAGGI DI ERRORE	
Applesoft .....	49
Messaggi di errore relativi ai file .....	56
Messaggi di errore relativi ai file DOS .....	57
IV - IL LINGUAGGIO MACCHINA	
I registri interni del 6502 .....	60
Il set di istruzioni del 6502 .....	61
V - I COME...?	
I Come...? .....	70

# INDICE

Indirizzi del Monitor .....	86
Indirizzi del Monitor e della ROM Autostart .....	88
Indirizzi di sistema .....	89
Indirizzi di sistema con la Language Card .....	92
Indirizzi di sistema - schede di interfaccia .....	94
Indirizzi memoria ROM .....	95
Indirizzi del Monitor .....	96
Indirizzi fondamentali .....	103
Applesoft - puntatori fondamentali .....	105
Applesoft - esempio n.1 .....	107
Applesoft - esempio n.2 .....	110
Applesoft .....	112
Indirizzi dell'interprete Applesoft .....	115
Integer BASIC - puntatori fondamentali .....	122
Integer BASIC - esempio .....	123
Indirizzi Integer BASIC .....	129
ROM: indirizzi dischetto .....	131
Comandi ROM .....	133
ROM: indirizzi memoria RAM .....	134
Routine RWTS .....	135
ROM: indirizzi pagina 3 .....	136
ROM: programmi di utilità .....	137
ROM - esempio .....	138

# COMANDI

## FUNZIONI DEL BASIC APPLESOFT

Una funzione chiede un argomento (oppure più argomenti) e restituisce un valore che è il risultato dell'applicazione di tale funzione al valore dell'argomento.

### Funzioni matematiche

- ABS** Valore assoluto dell'argomento posto tra parentesi.
- ASN** Arcosenante - il risultato è dato in radianti, compreso tra  $-\pi/2$  e  $\pi/2$ .
- COS** Coseno - l'argomento dev'essere in radianti. Esempio: cos(10 gradi) => COS(PI\*10/180).
- EXP** Esponenziale e<sup>x</sup>. L'argomento dev'essere minore o uguale a 99, in caso contrario si produce un supero di capacità.
- INT** Parte intera, o meglio la più grande parte intera inferiore o uguale all'argomento: INT(0.5) vale -0; INT(-2) vale -3.
- LOG** Logaritmo naturale (iperbolico) o in base e. Per ottenere il logaritmo di 8 in base 2, utilizzare LOG(8)/LOG(2). Esempio: logaritmo decimale di 2 => LOG(2)/LOG(10).
- RND** Fornisce, con un argomento positivo, un numero pseudo-casuale superiore o uguale a 0 e inferiore a 1. Esempio: PRINT RND(1) => .103112571. Se la chiamata alla funzione si succedono, i risultati costituiranno sempre la stessa serie di numeri pseudo-casuali. Una chiamata della funzione con un argomento negativo permette di uscire da una particolare serie. RND(0) restituisce l'ultimo numero generato.
- SIG** Funzione "segno": 1 se >0, -1 se <0 e 0 se =0.
- SIN** Seno - l'argomento è supposto in radianti.
- SQR** Radice quadrata - l'argomento dev'essere superiore o uguale a 0.
- TAN** Tangente - l'argomento è supposto in radianti.

## FUNZIONI DEL BASIC APPLESOFT

### Funzioni di tabulazione

- POS** Fornisce la posizione libera successiva sulla linea dello schermo (posizione orizzontale del cursore).
- SFC** Questa funzione può essere impiegata solo nell'istruzione PRINT. PRINT SFC(X) stampa X spazi. X dev'essere un numero intero compreso tra 0 e 255.
- TAB** Questa funzione può essere impiegata solo nell'istruzione PRINT. PRINT TAB(X) sposta il cursore alla posizione di stampa X (X è la posizione più a sinistra di una linea, 40 la più a destra). X dev'essere compreso tra 1 e 255. TAB(0) sposta il cursore alla posizione 256. Se X è minore dell'attuale posizione del cursore, non accade nulla. TAB non sposta mai il cursore verso sinistra.
- HTAB** Posiziona il cursore orizzontalmente prima dell'istruzione PRINT. HTAB(X) corrisponde alla posizione più a sinistra sullo schermo. La posizione estrema vale 255 (5 linee più lontano).
- VTAB** Posiziona il cursore verticalmente prima dell'istruzione PRINT. VTAB(X) corrisponde alla linea superiore sullo schermo. VTAB(24) posiziona il cursore alla linea inferiore sullo schermo. Se l'argomento è superiore, rispetto alla linea più bassa sullo schermo, la visualizzazione potrà essere effettuata solo alla linea "puntata" dall'argomento per ogni successiva istruzione PRINT.

### Funzioni di sistema

- PEE** Qualunque sia il valore dell'argomento, fornisce il numero di byte disponibili in memoria. Realizza anche la pulizia delle stringhe abbandonate.
- PEEK** Fornisce il contenuto (compreso tra 0 e 255) della locazione di memoria il cui indirizzo è uguale al suo argomento (che dev'essere un numero intero compreso tra 0 e 65535).
- USR** Chiama di un programma utente in linguaggio macchina. L'unico argomento è trasmesso l'accumulatore. L'indirizzo del sottoprogramma dev'essere preregistrato in R0R e R0C con JMP (R4C) in R0R. Il risultato è posto nell'accumulatore.

## Funzioni delle stringhe di caratteri

LEN(X)	Longhezza (da 0 a 255).
LEFT(X,N)	Estrazione degli N caratteri più a sinistra.
RIGHT(X,N)	Estrazione degli N caratteri più a destra.
MID(X,N,K)	oppure MID(X,N,K,M) estrazione fra tutti i caratteri o di N caratteri a partire dalla N-esima posizione. K dev'essere maggiore o uguale a 1.

## Funzione di conversione

ASC(X)	Restituisce il codice ASCII del primo carattere della stringa X. ASC("A") vale 65.
CHR(X)	Restituisce il carattere il cui codice ASCII vale X. CHR(64) è <u>CTRL-D</u> .
STR(X)	Fornisce la rappresentazione di un numero in stringa di caratteri a partire dal suo valore numerico X.
VAL(X)	Fornisce il valore numerico rappresentato dalla stringa X.

## Funzioni grafiche (bassa risoluzione)

COLOR	Fornisce un colore (da 0 a 15) per un successivo tratto in bassa risoluzione.
PLT X,Y	Disegna un piccolo quadrato all'ascissa X e all'ordinata Y. X e Y devono essere comprese fra 0 e 39. 0,0 è l'estrema posizione in alto a sinistra.
HLEN X1,X2 AT Y	Traccia una linea orizzontale tra X1 e X2 all'ordinata Y.
VLEN Y1,Y2 AT X	Traccia una linea verticale tra Y1 e Y2 all'ascissa X.
SCAN(X,Y)	Restituisce il colore del quadratino tracciato in X,Y.

## FUNZIONI DEL BASIC APPLESOFT

### Funzioni grafiche (alta risoluzione)

<b>HCOLOR=</b>	Fissa il colore (0, 1, 2, 3) del successivo punto da tracciare.
<b>HPLOT X,Y</b>	Disegna un punto all'ascissa X e all'ordinata Y. X dev'essere compreso tra 0 e 279. Y dev'essere compreso tra 0 e 159 (HBR) oppure tra 0 e 191 (HBR2). 0,0 è l'estremo in alto a sinistra.
<b>HPLOT X1,Y1 TO X2,Y2</b>	Traccia una linea tra due punti, il comando può essere esteso ad altri punti ... TO Xn,Yn.
<b>DRAW F AT X,Y</b>	Disegna la figura n. F della tabella delle figure (shapes) partendo dal punto X,Y.
<b>XDRAW F AT X,Y</b>	Disegna la figura n. F della tabella delle figure (shapes), il colore di ogni punto è il complemento dell'attuale colore del punto visualizzato sullo schermo.
<b>ROT=</b>	L'argomento è proporzionale all'angolo di rotazione che si desidera dare alla figura da disegnare con la funzione DRAW. ROT=0 corrisponde ad una rotazione di 90°.
<b>SCALE=</b>	Fornisce un valore d'ingrandimento alla figura da disegnare compreso tra 1 e 255.

### Funzioni relative ai comandi per i giochi (paddle)

<b>POL(X)</b>	Restituisce un numero da 0 a 255 proporzionale alla posizione angolare del potenziometro di cui è dotato il comando (paddle). 0 vale 0, 1, 2 oppure 3.
<b>PEEK(X-16387)</b>	Restituisce un risultato maggiore di 127 se il pulsante sul comando X è stato premuto. X vale 0, 1 oppure 2.

### Funzione altoparlante

<b>PEEK(-16336)</b>	Emette un "click" dall'altoparlante.
---------------------	--------------------------------------

Basta obbligatorio	Parola chiave	Definizione - Esempio
	<b>G</b>	Ritarda l'esecuzione di un sottoprogramma in linguaggio macchina il cui indirizzo si trova alla locazione \$3F4; \$3F7 con JMP(\$AC) alla locazione \$3F5. 164 → 10 se il sottoprogramma è convertito da decimale a esadecimale.
	<b>CALL</b>	Ritarda l'esecuzione di un sottoprogramma in linguaggio macchina all'indirizzo specificato. CALL-151 Un argomento negativo equivale al complemento di \$334 dell'indirizzo cercato.
	<b>CLEAR</b>	Rimette a zero tutte le variabili. Le stringhe hanno tutte lunghezza nulla.
Diretto	<b>CONT</b>	Prosegue l'esecuzione di un programma interrotto.
Diretto	<b>CTRL-C</b>	Arresta l'esecuzione di un programma in corso. Il programma resta intatto.
	<b>CTRL-D</b>	Inizia un comando EOF dev'essere preceduto da "PRMT".
Diretto	<b>CTRL-E</b>	Sospende la visualizzazione, l'immagine resta fissa fino a quando non viene premuto un tasto qualsiasi.
Diretto	<b>CTRL-F</b>	Annulla le impostazioni di una linea oppure un dato appena impostato.
	<b>DATA</b>	Definisce un elenco di costanti che saranno lette dall'istruzione READ 10 DATA ABC,5,0.15
	<b>DEL</b>	Con due argomenti separati da una virgola, delimita una parte di programma da cancellare DEL 10,30 sopprime le istruzioni da 10 a 30



Modo obbligatorio	Parola chiave	Definizione - Esempio
Programmato	<b>DEF FN</b>	Definisce una funzione dell'utente con un solo argomento: 10 DEF FN F(X)=0-354*INT(X/354)
	<b>DIM</b>	Dimensionamento di una matrice (fissa il numero e i valori massimi degli indici). 10 DIM A(100),B(500),C(10) 20 DIM T(8) 30 DIM M(10,10,10) 88 è il numero massimo di indici
	<b>END</b>	Arresta l'esecuzione relativa alla serie di istruzioni.
	<b>FOR</b>	Introduce un ciclo (loop): tutte le istruzioni comprese tra FOR I=A TO B STEP C e NEXT I saranno ripetute per tutti i valori di I compresi tra A e B, con passo C. 10 FOR I=1 TO 1000 20 FOR J=1.5 TO 2 STEP .1 30 FOR K=0 TO -8 STEP -2 Se più cicli si succedono con il medesimo indice, non è possibile interrompere la progressione dell'indice fino al suo valore massimo. 10 FOR I=1 TO 100 20 IF M(I)=0 THEN I=1.1*100 30 NEXT I 40 IF I=0 THEN PRINT"NON TROVATO":END 50 PRINT"TROVATO IN",I
	<b>FLASH</b>	Visualizza i caratteri in modo lampeggiante. Questo modo non può essere neutralizzato da RESET. Il tasto → modifica i caratteri sullo schermo. Battere NORMAL per ristabilire la situazione.

<i>Modo obbligatorio</i>	<i>Parola chiave</i>	<i>Definizione - Esempio</i>
Programmato	<b>GET</b>	Attende un carattere da tastiera, che poi non viene visualizzato. CTRL-C non ha alcun effetto 10 GET A\$ non è raccomandato con istruzioni del DOS nel programma, salvo che CTRL-D sia preceduto da un RETURN 20-CHAR(11)+CHAR(12)
	<b>GOSUB</b>	Chiamata ad un sottoprogramma 10 GOSUB 1000
	<b>GOTO</b>	Salta ad un'altra istruzione numerata 10 GO TO 50
	<b>GR</b>	Fare una parte dello schermo in visualizzazione grafica da 40x40 quadratini. Lascia 4 linee di testo in basso.
	<b>GRB</b>	Fare una parte dello schermo in visualizzazione grafica da 200x160 punti. Lascia 4 linee di testo.
	<b>GRB2</b>	Fare tutto lo schermo in visualizzazione grafica da 200x192. Il cursore non viene visualizzato.
	<b>HEMEM</b>	Specifica la più alta posizione di memoria RAM utilizzabile dal programma.
	<b>HORE</b>	Pulisce lo schermo e posiziona il cursore in alto a sinistra. Preceduto da TEXT: pulisce tutto lo schermo.
	<b>IF</b>	Salto condizionato, nella forma IF (condizione) THEN (istruzione). Se la condizione non è soddisfatta (risultato falso o uguale a 0) si passa alla linea seguente; se la condizione è soddisfatta viene eseguita l'istruzione che segue THEN IF C THEN GOTO n oppure

# ISTRUZIONI DEL BASIC APPLESOFT

Bando obbligatorio	Parola chiave	Definizione - Esempio
Programmato	<b>INPUT</b>	<p>IF C THEN a oppure ancora  IF C GOTO a  10 IF A=0 THEN T=0  20 IF A=-1 THEN S  30 IF A=00 OR A=100 THEN 100</p> <p>Acquisizione di un dato da tastiera  10 INPUT A  20 INPUT A,B,C,D  30 INPUT "Il primo nome ?";N1</p>
	<b>IN</b>	<p>Connette in entrata la periferica collegata al connettore indicato nell'argomento.</p>
	<b>INVERSE</b>	<p>Provoca la visualizzazione dei caratteri in nero su bianco. Per ritornare alla consueta visualizzazione battere NORMAL.</p>
	<b>LET</b>	<p>Istruzione di assegnazione di un valore ad una variabile. Non è obbligatoria (IF X=-1000000)</p>
	<b>LIST</b>	<p>Lista un programma  LIST tutto il programma  LIST 10,100 da 10 a 100  LIST 100, da 100 alla fine  LIST ,10 fine a 10  La virgola può essere sostituita con -</p>
Diretto	<b>LOAD</b>	<p>Carica un programma da cassetta in memoria RAM.</p>
	<b>LOMEM:</b>	<p>Specifica la posizione più bassa di memoria in RAM disponibile per le variabili del programma.</p>
	<b>NEW</b>	<p>Cancella il programma attualmente in memoria RAM. I 2 puntatori sono posti a zero.</p>
	<b>NEXT</b>	<p>Provvede alla successiva iterazione del ciclo FOR  NEXT I    NEXT J,I    NEXT</p>

Modo obbligatorio	Parola chiave	Definizione - Esempio
	<b>NORMAL</b>	Ripristina sullo schermo la visualizzazione in bianco su nero.
	<b>NOTRACE</b>	Disabilita il modo TRACE
	<b>ON</b>	ON 1 GOTO 10,20,30 Se 1 vale 1, si va alla linea 10, se vale 2 si va alla linea 20, alla linea 30 se vale 3. Se 1 è nullo o falso, si passa all'istruzione successiva. ON 1 GOSUB 1000,2000 Se 1 vale 1, viene chiamato il sottoprogramma alla linea 1000, se vale 2 quello della linea 2000.
	<b>ONERR</b>	ONERR GOTO 500 Consente di intercettare un errore prima che questo provochi l'arresto dell'esecuzione del programma. Quando è riscontrato un errore il programma salta alla linea indicata.
	<b>POKE</b>	POKE a,b scrive il dato "b" all'indirizzo assoluto "a". "a" e "b" sono espressi in decimale. POKE 2000,45
	<b>POP</b>	Elimina dalla stack l'ultimo indirizzo di ritorno di un sottoprogramma. Il RETURN successivo porta all'istruzione che segue l'ultimo GOSUB eseguito.
	<b>PRINT</b>	Visualizza un risultato sullo schermo o sulla stampante PRINT A 10 PRINT A,B;J (se accanto all'altro) 20 PRINT "I";I 30 PRINT A,B,J (in zone fisse)
	<b>PPR</b>	Trasferisce l'uscita dei dati alla periferica collegata alla scheda nel connettore specificato nell'argomento PPRJ consente l'uscita sulla stampante se questa è collegata allo slot 1.

# ISTRUZIONI DEL BASIC APPLESOFT

Bando obbligatorio	Parola chiave	Definizione - Esempio
Programmato	<b>READ</b>	Lettura dei dati contenuti nell'istruzione DATA associata 10 READ A 20 READ B,C
Programmato	<b>RECALL</b>	Recupero dei dati numerici dalla cassetta verso la memoria RAM. L'argomento è costituito da una variabile specifica e correttamente dimensionata. 1 DIM B(100) 100 RECALL B
	<b>REM</b>	Introduce un commento nel listato di un programma.
	<b>RESET</b>	Questo tasto equivale a [CTRL-C] durante l'esecuzione di un programma. Il programma si arresta ma rimane intatto. Le periferiche in linea sono disattivate. Il programma torna all'interprete oppure all'indirizzo previsto in 43F2, 43F3 se PAPERUP è conforme, altrimenti il sistema riparte come all'accensione (COLDSTART).
	<b>RESTORE</b>	Torna all'inizio del DATA
	<b>RETURN</b>	Ritorno da un sottoprogramma 100 RETURN
	<b>RESUME</b>	Torna all'istruzione dove è stato riscontrato un errore trattato dal programma l'istruzione ONERR GOTO
	<b>RUN</b>	Ordina l'esecuzione di un programma. Aziona a zero tutte le variabili. RUN RUN 10
	<b>SAVE</b>	Salvataggio di un programma su cassetta.
	<b>SPEED=</b>	Modifica la velocità di visualizzazione sullo schermo da 1 (minimo) a 333 (standard).

Modo obbligatorio	Parola chiave	Definizione - Esempio
	<b>STEP</b>	Introduce il passo di incremento nel ciclo FOR NEXT.
	<b>STOP</b>	Arresta l'esecuzione di un programma. <b>10 STOP</b> visualizza il messaggio <b>***** 10 10</b> Si può proseguire con il comando CONT (se le istruzioni non sono state modificate).
	<b>STORE</b>	Salvataggio di una matrice numerica su cassetta. Non funziona direttamente con le stringhe di caratteri <b>STORE A</b>
	<b>TEXT</b>	Riporta la visualizzazione in modo testo dopo la visualizzazione in modo grafico. Ripristina i valori standard relativi alla visualizzazione: 40 caratteri per linea 34 linee per ogni "videata"
	<b>THEN</b>	Introduce l'istruzione da effettuare quando un IF è soddisfatto.
	<b>TO</b>	Introduce il valore finale nel ciclo FOR.
	<b>TRACE</b>	Modo di verifica e correzione di eventuali errori. Visualizza il numero di istruzioni eseguite senza RETURN, quindi tra le linee dei risultati del programma.
	<b>WAIT</b>	Passo condizionata in un programma. <b>WAIT A,B</b> Sospende l'esecuzione di un programma fino a quando ciò che è contenuto all'indirizzo A e 1bit per bit l'equivalente di B sia differente da 0 <b>WAIT -16384,128</b> corrisponde all'attesa della pressione di un tasto della tastiera.

## OPERATORI BASIC

+	Addizione di numeri oppure concatenazione di stringhe di caratteri
-	sottrazione
*	Moltiplicazione
/	Divisione
^	Elevazione a potenza
=	Uguale        <> diverso
<	Minore        > maggiore
<=	Minore o uguale
=<	Uguale o minore
>=	Maggiore o uguale
=>	Uguale o maggiore
NOT	Funzione logica NOT, agisce su un solo operando. Se A è vero NOT A è falso Se A è falso NOT A è vero
AND	Funzione logica AND su due operandi P AND Q non è vero solo se P e Q sono veri.
OR	Funzione logica OR su due operandi P OR Q non sono falsi solo se P e Q sono falsi.

- Posizionamento del cursore
- Visualizzazione di linee sullo schermo
- Copia di una linea in memoria RAM
- Soppressione di una linea dalla memoria RAM

Con il Monitor AUTOSTART ROM in memoria ROM i comandi di posizionamento del cursore sono ottenuti con i seguenti quattro tasti, preceduti dalla pressione del tasto ESC.



Utilizzando solo questi quattro tasti si rende in sede controllo cursore nelle quattro direzioni.

Per tornare al modo normale di inserimento e correzione, basta premere il tasto ESC.

Per cancellare, partendo dalla posizione del cursore, fino in fondo alla pagina premere ESC E.

Per cancellare, partendo dalla posizione del cursore, fino in fondo alla pagina premere ESC F.

Per cancellare tutto il contenuto dello schermo e posizionare il cursore nell'angolo in alto a sinistra, ESC (SHIFT/P).

Usare il tasto ← (freccia sinistra) per annullare l'ultimo carattere inserito.

Usare il tasto → (freccia destra) per reinserire in memoria RAM i caratteri posti sotto il cursore. Per reinserire una serie di caratteri utilizzare il tasto → insieme al tasto REPT, si procederà più velocemente.

Per sopprimere una linea d'istruzione dal programma in memoria RAM, battere il numero di linea seguito da RETRM.



# APPLESOFT

Le parole chiave in ordine alfabetico e i relativi codici esadecimali.

Ordine alfabetico dalla A alla Z

Parola chiave	Codice esad.	Parola chiave	Codice esad.	Parola chiave	Codice esad.
A	88F	OR	888	ON	884
ABB	884	HCOLOR=	892	ONERR	845
AND	8C0	HBR	891	OR	8CE
ASC	8E6	HIREM1	843	PBL	808
AT	8C5	HLIN	88E	PEEK	8E2
ATM	8E1	HONE	897	PLOT	885
CALL	88C	HPLDT	893	POKE	889
CHRR	8E7	HTAB	896	POP	841
CLEAR	888	IF	840	POS	809
COLOR	848	INH	88B	PRINT	884
CONT	88B	INPUT	884	PRN	884
COS	8DE	INT	883	READ	887
DATA	883	INVERSE	89E	RECALL	8A7
DEF	888	LEFTA	8EB	REM	882
DEL	885	LEN	8C3	RESTORE	84E
DIN	886	LET	84A	RESUME	846
DRAW	894	LIST	88C	RETURN	881
END	880	LOAD	886	RIGHTR	8E9
ESP	889	LOB	88C	RND	808
FLASH	89F	LOWER1	844	ROT	890
FN	8C2	RIGH	88A	RUN	8AC

Le parole chiave in ordine alfabetico e i relativi codici esadecimali.

(Segue)

Ordine alfabetico dalla F alla I

Parola chiave	Codice esad.	Parola chiave	Codice esad.	Parola chiave	Codice esad.
FOR	881	NEW	88F	SAVE	887
FRE	886	NEXT	882	SCALE	889
GET	88E	NORMAL	898	SCREEN	887
GOOSE	880	NOT	8C6	SEN	882
GO TO	848	NOTRACE	89C	SHLDR	894
SIN	88F	STR	8C4	USR	883
SPEC	8C3	TAB	8C0	VAL	8E3
SPEED=	889	TAN	8E0	VLEN	88F
SEE	8D4	TEST	889	VTAB	8A2
STEP	8C7	THEN	8C4	WAIT	885
STOP	883	TO	8C1	XPLOT	*
STORE	848	TRACE	898	XDRAM	895

\* XPLOT è codificato in: 838 "I" 880 "PLOT"

Non può essere utilizzato come nome di variabile.

# FUNZIONI E ISTRUZIONI DELL'INTEGER BASIC

Le parole chiave in ordine alfabetico dalla A alla V

Parola chiave	Codice esad.	Parola chiave	Codice esad.	Parola chiave	Codice esad.
ABS	831	INPUT	853 853 854	PRINT	861 862 863
AND	818				
ASC	83C	LEN	838	POS	87C
AT	868 86E	LET	85E	REN	850
AUTO	808	LIST	874 875 876	RETURN	858
CALL	858	LOAD	884	RND	82F
CLR	88C	LOWER	811	SAVE	805
COLOR*	866	RAN	88F	SCREEN	858
CON	866	MOD		SON	858
DEL	86F	NEW	888	STEP	858
DIR	84C	NEXT	85F	TAB	858
DSP	878 87C	NOT	837	TEXT	848
END	831	NOTRACE	874	THEN	824
FOR	833	NOESP	878 879	TO	857
GOON	85C	OR	81E	TRACE	878
GOTO	85F	PBL	832	VLEN	86C
GR	84C	PEEK	82E	VTAB	86F
HLEN	86F	PLOT	867		
IF	868	POKE	864		
INH	87F	POP	877		

## Funzioni matematiche

<b>ABS</b>	Valore assoluto.
<b>MOD</b>	Restituisce il resto della divisione tra il primo e il secondo operando. PRINT 15 MOD 4 restituisce il numero 3
<b>RND</b>	Genera un numero intero pseudo-casuale positivo e inferiore all'argomento PRINT RND(10): fornisce il numero 3

## Funzioni di tabulazione

<b>TAB</b>	Equivale al comando HTAB dell'AppleSoft. Posiziona il cursore, sullo schermo, in una posizione tra 1 e 255.
<b>VTAB</b>	Posiziona verticalmente il cursore (in assoluto) tra 1 e 24.
<b>VLIN Y1,Y2 AT I</b>	Traccia una linea verticale tra Y1 e Y2

## Funzioni di sistema

<b>peek</b>	Legge il contenuto di una posizione di memoria all'indirizzo indicato dall'argomento. L'argomento è compreso tra -32768 e +32767.
-------------	---

## Funzioni di conversione

<b>ASC("C")</b>	Restituisce il codice ASCII del carattere.
-----------------	--

## Funzioni grafiche

<b>COLOR</b>	Fornisce un colore (da 0 a 15) per il successivo tracciamento in bassa risoluzione.
<b>PLOT X,Y</b>	Disegna un piccolo quadrato nella posizione dell'ascissa X e della coordinata Y. X e Y devono essere comprese tra 0 e 19.9,0,0 corrisponde alla posizione in alto a sinistra.
<b>HLIN X1,X2 AT Y</b>	Traccia una linea orizzontale tra X1 e X2 alla coordinata Y.

## **FUNZIONI DELL'INTEGER BASIC**

- VLEN Y1,Y2 AT X** Traccia una linea verticale tra Y1 e Y2 all'ascissa X.
- SCREEN,X1** Restituisce il colore del quadratino tracciato alla posizione X,1.

### **Funzioni relative ai comandi dei giochi (Paddle)**

- POL(X)** Restituisce un numero tra 0 e 355 proporzionale alla posizione angolare del potenziometro. 1 vale 0,1 oppure 2.
- PDDR(X-14287)** Restituisce un valore maggiore di 127 se è premuto il tasto della paddle 1. 1 vale 0,1 oppure 2.

### **Funzioni dell'altoparlante**

- PDDR(-14334)** Produce l'emissione di un 'click' dall'altoparlante.

Parola chiave	Definizione - Esempio
<b>AUTO</b>	Numerazione automatica delle linee a partire dal numero indicato nell'argomento. <b>AUTO 100,5</b> : numerazione di 5 in 5 partendo dalla linea 100. Se il secondo argomento non è specificato la numerazione procede di 10 in 10.
<b>CALL</b>	Esegue un sottoprogramma che comincia all'indirizzo indicato (esempio di 32767). <b>CALL-919</b> pulisce lo schermo.
<b>CLR</b>	Risette a zero tutte le variabili.
<b>CON</b>	Presegue il programma dopo una interruzione.
<b>CTRL-C</b>	Arresta un programma.
<b>DEL</b>	Supprime le linee di istruzioni <b>DEL 10,100</b>
<b>DIM</b>	- Dimensionamento di una matrice ad una sola dimensione. - Dimensionamento della lunghezza massima di una variabile stringa di caratteri <b>10 DIM A(20)</b> (obbligatoria per tutte le variabili stringhe di caratteri).
<b>DSP</b>	Visualizza i nuovi valori assunti dalla variabile specificata durante l'esecuzione di un programma <b>10 X=ABS(10)</b> <b>15 DSP X</b> <b>20 STOP 10</b> <b>AAA → 0 10 X=3</b> <b>0 10 X=7</b> etc.
<b>END</b>	Ultima istruzione di un programma BASIC. Se manca tale istruzione viene prodotto un messaggio di errore.
<b>FOR</b>	Ciclo <b>FOR-NEXT</b> che ripete, per ogni ciclo, le istruzioni comprese tra <b>FOR</b> e <b>NEXT</b> , oltre al conteggio e alla verifica della fine ciclo.
<b>GOSUB</b>	Chiamata di un sottoprogramma ad un indirizzo che può essere rappresentato da una espressione aritmetica oppure da una variabile <b>GOSUB 100:10</b>

# ISTRUZIONI DELL'INTEGER BASIC

Parola chiave	Definizione - Esempio
<b>GOTO</b>	Salta ad una linea di istruzione il cui numero può essere calcolato nel programma <code>GOTO N=10</code>
<b>GR</b>	Modo grafico con risoluzione 40x40
<b>IF</b>	Salto condizionale nella forma IF condizione THEN istruzione V : istruzione F. Se la condizione non è soddisfatta l'istruzione V è ignorata, si passa quindi all'istruzione F. Se la condizione è soddisfatta l'istruzione V viene eseguita prima dell'istruzione F.
<b>INPUT</b>	Acquisizione di un dato da tastiera. E' necessario separare il messaggio dalla variabile con una virgola. <code>IF INPUT "QUAL'E' IL Tuo nome ? ",N</code> Durante l'esecuzione viene visualizzato un punto di domanda davanti al cursore se non è previsto un messaggio oppure se la variabile è numerica.
<b>INH</b>	Esceva i dati da una periferica. L'argomento è il numero del connettore (da 1 a 7) corrispondente al collegamento della periferica.
<b>LET</b>	Assegna un valore ad una variabile.
<b>LOAD</b>	Caricamento di un programma da cassetta in memoria RAM.
<b>LOCN:</b>	Modifica in memoria il posizionamento delle variabili.
<b>MAN</b>	Termina il modo automatico di numerazione delle linee. Quando appare un nuovo numero di linea spostare il cursore fino a > quindi battere <code>MAN</code> .
<b>NEW</b>	Cancella il programma in memoria RAM.
<b>NEXT</b>	Comanda il passaggio al successivo ciclo FOR.
<b>NOTRACE</b>	Termina la funzione TRACE.
<b>NDOSP</b>	Termina la funzione di visualizzazione dei cambiamenti relativi ai valori delle variabili chiesti con l'istruzione <code>DSP</code> .

Parola chiave	Definizione - Esempio
<b>POKE</b>	POKE a,b scrive il dato "b" (inferiore a 256) nella cella di memoria che ha l'indirizzo "a" (da -32768 a 32767).
<b>POP</b>	Elimina un livello di "return" da un sottoprogramma.
<b>PRINT</b>	Visualizza un risultato sullo schermo.
<b>PR#</b>	Seleziona la periferica di uscita.
<b>REM</b>	Commento in una linea di programma.
<b>RETURN</b>	Comanda il ritorno da un sottoprogramma.
<b>RUN</b>	Esegue un programma, le variabili dimensionate non sono rimesse a zero.
<b>SAVE</b>	Salva un programma su cassetta.
<b>STEP</b>	Definisce il passo di incremento nel ciclo FOR.
<b>TEXT</b>	Riporta il calcolatore al modo testo dopo l'utilizzo in modo grafico. Ripristina il modo di visualizzazione standard.
<b>THEN</b>	Definisce l'istruzione da effettuare quando un IF è soddisfatto. La seconda istruzione dopo THEN sarà effettuata se la condizione è falsa.
<b>TRACE</b>	Se più istruzioni sono sulla medesima linea, la funzione TRACE indicherà solo il passaggio relativo alla prima istruzione della linea.

I nomi delle variabili in Integer BASIC sono conservati interamente quale che sia la loro lunghezza.

In Integer BASIC sono possibili solo operazioni intere.

Gli errori di sintassi sono riconosciuti subito dopo aver convalidato la linea con **RETURN**.

**Operatori**

- = - \* / (divisione intera)
- ^ (elevazione a potenza)
- = (uguale)
- > < ≠ (diverso da)
- NOT, AND, OR



## SISTEMA UCSD - PASCAL

### Configurazione standard

- + 48 Kbyte di memoria RAM
- + Language Card, con 16 Kbyte di memoria RAM, sullo slot 0
- + 2 unità per floppy disk Disk II
- + 1 controller per floppy disk con FROM P2A e P2B (16 settori) nello slot 6.

### I dischetti del sistema Pascal

APPLE II:     {   Sistema   }     Editor, Filer, Apple Library  
APPLE II:     {            }     Compiler, Linker, Assembler, etc.  
APPLE II:     Dimostrazione capacità grafiche, Formatter, Library  
BASIC 80:     Per il passaggio in BASIC (sotto DOS 3.2) e caricamen-  
to di un interprete nella Language Card.

### Formattazione dei dischetti vergini sotto Pascal

1 (esecuzione) del programma APPLE II: FORMATTER

Forma il dischetto vergine in D2 (drive 2)

Rispondere 5 alla domanda FORMAT WHICH DISK (4,5,9,...12) ? oppure RETURN per rinunciare alla formattazione.

### Copia dei dischetti:

P (gestione dei dischetti) seguito da TRANSFER:

TRANSFER ? nome del dischetto da copiare:

TO WHERE ? BLANK :

TRANSFER 200 BLOCKS ? Y

DESTROY BLANK : ? Y

### Comandi

<u>E</u> dit	Chiama l'editor.
<u>R</u> un	Chiama un programma sorgente (.TEXT), lo com- pila e lo esegue.
<u>F</u> iler	Gestione degli archivi e dei programmi su dischetto.
<u>C</u> omp	Compilazione.
<u>L</u> ink	Unisce due programmi già compilati.

<u>Execute</u>	Esecuzione di un programma oggetto (.CODE).
<u>Assem</u>	Chiama l'assembler.
<u>Diebug</u>	Non è utilizzato.
<u>H</u>	Rinizializza il sistema con un dischetto DOS 3.2 o con il dischetto BASICS (per il DOS 3.2).
<u>CTRL-B</u>	Visualizzazione dei caratteri dalla colonna 41 alla colonna 80 (parte destra dello schermo).
<u>CTRL-K</u>	Per l'impostazione del carattere K.
<u>SHIFT-M</u>	Per l'impostazione del carattere M.

## EDITOR UCSD - PASCAL

>EDIT    >ADJUST    >COPY    >DELETE    >FIND    >INSERT  
>JUMP    >REPLACE    >QUIT    >RECHANGE    >TAP    >SET  
>VERIFY

### Direzione di posizionamento del cursore nel testo

> scorre con  o  : CTRL-A visualizza la parte destra  
< indietro    CTRL-L posiziona verso il basso  
                  CTRL-O posiziona verso l'alto  
                  CTRL-B posiziona ad inizio linea  
                  CTRL-J visualizza seguendo il cursore

### Modifica della direzione del cursore

- oppure , per <

= oppure , per >

<BS> un carattere indietro con la freccia sinistra 

<EXT> fine delle operazioni con CTRL-C

<ESC> annullamento di operazione con il testo ESC

<DEL> annullamento di una linea di testo con CTRL-J

Impaginazione    >ADJUST : >LJUST    >RJUST    >CENTER  
                  <LEFT, RIGHT, UP, DOWN ARROWS> C <EXT> TO LEAVE J

L            posizionamento del testo sul margine sinistro  
R            allineamento del testo sul margine destro  
C            centratura del testo nella pagina  
LEFT        freccia ←    posizionamento del testo a sinistra  
RIGHT       freccia →    posizionamento del testo a destra  
UP           CTRL-O    allineamento della linea precedente  
DOWN        CTRL-L    allineamento della linea seguente  
<EXT>       CTRL-C    per lasciare la funzione

Copia            >COPY : >BUFFER    >FROM    >FILE    <ESC>

B            copia tampone (per esempio di un testo che sarà visualizzato).

F            copia di un file dal dischetto dove si trova il cursore.  
<ESC>        il testo ESC consente di lasciare la funzione

Cancellazione    >DELETE : <>MOVING COMMANDS C <EXT> TO DELETE,  
                  <ESC> TO ABORT J

freccia ← per cancellare un carattere  
 freccia → per riprendere il testo cancellato  
 <EXT> CTRL-E per convalidare la cancellazione  
 <ESC> ESC per annullare la cancellazione

Verifica >VERIFY Verifica dello schermo dopo le modifiche

Ricerca >Find E I J : LIT <TARGET> =>

La frase da cercare è compresa tra / e /. L<ITERAL se il testo è incluso nella frase, I n I alla n-esima occorrenza della frase. Questo numero dev'essere indicato prima di battere F.

Inserzione >INSERT TEXT E <BS> A CHAR, <DEL> A LINE J  
 E <EXT> TO ACCEPTS, <ESC> ESCAPES J

Il testo da inserire è normalmente inserito correggendo con la freccia <- oppure <BS> per il carattere precedente o, ancora, CTRL-I per annullare una linea, <EXT> oppure CTRL-E per convalidare l'inserimento, <ESC> per annullare l'operazione.

Salto >GOTO : BEGINNING E:ND MARKER <ESC>

B Per posizionare il cursore all'inizio del testo.  
 E Per posizionare il cursore alla fine del testo.  
 M Per posizionare il cursore ad un marker prefissato (vedere SET a MARKER).  
 <ESC> Per lasciare la funzione.

Uscita dall'editor >QUIT : UPDATE THE WORKFILE AND LEAVE  
 EXIT WITHOUT UPDATING  
 RETURN TO THE EDITOR WITHOUT  
 UPDATING  
 WRITE TO A FILE NAME AND RETURN

U Aggiorna il file di lavoro chiamato SYSTEM.WRK.TEXT.  
 E Uscita senza aggiornamento del file.  
 R Ritorno all'editor senza salvare il file.  
 W Salvataggio del file specificato e ritorno all'editor.

Margini >SET : ENVIRONMENT MARKER <ESC>  
 ENVIRONMENT : E OPTIONS J <ESC> OR <SP>  
 TO LEAVE

AUTO INDENT TRUE indentazione automatica (allineamento sulla linea precedente).

## EDITOR UCSD - PASCAL

FILLING	FALSE	riempimento fino al margine destro.
LEFT MARGIN	0	margine sinistro.
RIGHT MARGIN	70	margine destro.
PARA MARGIN	5	margine relativo al paragrafo.
COMMAND CH	"	carattere di comando nel modo MARGIN
TICKON DEF	TRUE	presenta il modo FIND e REPLACE.

Nell'edizione di un programma scritto in Pascal, A deve restare TRUE e F dev'essere FALSE.

<SP> barra-spazio per uscire dalla funzione.

**Sostituzione** >REPLACE I n I : LIT VOFF <TARGET> <SUB> =>

Sostituzione del testo compreso tra / e / con un nuovo testo compreso tra / e / di lunghezza qualsiasi.

L Per sostituire una parte di testo.

n Numero delle operazioni di sostituzione da effettuare (dev'essere impostato prima del comando R).

**Cambio di caratteri** >CHANGE : TEXT I <S> & CHAR I  
I <ESC> ESCAPE, <EXT> ACCEPTS I

Il carattere sostituisce quello posto sotto il cursore.

<BS> freccia ← per il carattere precedente.

<ESC> Tasto ESC per annullare.

<ETD> CTRL-C per convalidare.

**Cancellazione** >DAP cancellazione dopo la corrente  
posizione cursore fino a quella d'inizio dell'ultimo testo trovato, sostituisce o inserisce.

# SISTEMA UCSD - PASCAL GESTIONE DEI FILE E PROGRAMMA CON IL FILEN

## COMANDI

FILEN GET SAVE NEW LIST DIRECTORY  
EXTENDED-DIRECTORY LIST RENAME CHANGE  
TRANSFER DATE QUIT VIEWER WHAT  
BAD-BLOCKS EXAMINE IDEAS PROFILE

- G** GET ? Nome del dischetto : Nome di programma  
Caricamento in memoria del programma specificato, sostituisce il SYSTEM.WRK.TEST (file di lavoro).  
TEXT FILE LOADED, l'operazione è stata realizzata.
- S** SAVE AS ? Nome del dischetto : Nome di programma  
Salvataggio di un file di lavoro sotto il nome specificato nel dischetto indicato.  
TEXT FILE SAVED, l'operazione è stata realizzata.
- N** Cancellazione di un file di lavoro in memoria RAM e nel dischetto. Il calcolatore risponde : WORKFILE CLEARED.
- L** DIR LISTING OF ? Nome del dischetto :  
Visualizza il contenuto del dischetto (catalogo).  
Far seguire il nome da ,PRINTER : per ottenere una stampa su carta.
- E** Visualizza del contenuto con informazioni diverse (come quelle presenti nelle zone inutilizzate).  
DIR LISTING OF ? Nome del dischetto :
- R** Soppressione di un file.
- C** Modifica il nome di un file sul dischetto.
- T** Trasferimento di un dischetto o di un file su un altro dischetto.  
TRANSFER ? Nome del dischetto : E Nome del programma 2  
TO WHERE ? Nome del dischetto : E Nome del programma 3  
Per stampare un programma sorgente rispondere PRINTER: alla domanda TO WHERE ?.
- D** Aggiornamento della data.  
Giorno - Mese (2 lettere) - Anno (2 cifre).
- Q** Permette di lasciare le funzioni del FILEN.
- V** Elenco dei volumi conosciuti dal sistema ordinati per numero e per nome.
- W** WHAT fornisce i nomi dei file di lavoro e indica se sono stati salvati oppure no.

## SISTEMA UCSD - PASCAL - FILES

- B Bad-blocks verifica i 288 blocchi di un dischetto e segnala i blocchi in cattivo stato.
- I Esame dei blocchi che sono risultati in cattivo stato e tentativo di ripristino. Se ciò non risulta possibile consente di marcare i blocchi danneggiati (operazione utile prima dell'utilizzazione di un dischetto vergine).
- Z Zero, cancella la DIRECTORY (elenco degli archivi).
- P Prefixes, consente la modifica del nome relativo al volume corrente assunto per default battendo soltanto "I".

Il "prog" (pronto) è rappresentato dal segno #.  
I dati sono forniti in notazione esadecimale.  
Gli indirizzi sono forniti con 4 cifre esadecimali.

Comando	Definizione - esempio
Indirizzo S	Esecuzione di un programma che inizia a questo indirizzo. 100 S : rinalizizzazione, 'a caldo' del BASIC.
Indirizzo L	Elenca 20 istruzioni in linguaggio macchina partendo da questo indirizzo; disassembla i codici esadecimali nei codici mnemonici del Mini-Assembler. 100 L 100 - 4C BF 7D JZF 0100F 101 - 4C 84 7D JZF 01004 ... etc.
Indirizzo 1.Indirizzo 2	Visualizza il contenuto delle posizioni di memoria partendo dall'indirizzo 1 fino all'indirizzo 2. 100.107 100 - 4C BF 7D 4C 84 7D 4C FB Gli indirizzi di inizio linea sono sempre nella forma XXXX oppure XXXX salvo eventualmente l'indirizzo 1.
Indirizzo	Se viene specificato un solo indirizzo il Monitor rinvia il contenuto della cella di memoria specificata. 107 107 - BF
Indirizzovalore "spazio" valore	Modifica o scrittura dei valori in posizioni di memoria adiacenti partendo dall'indirizzo specificato. 73a 00 20 (modifica BASIC).



# COMANDI DEL MONITOR

Comando	Definizione - esempio
Indirizzo 1 ( Indirizzo 2. Indirizzo 3 #	Spostamento di una zona di valori contenuti partendo dall'indirizzo 2 fino all'indirizzo 3, nella zona che comincia all'indirizzo 1. 4000-400.7FF # salvataggio della pagina di testo o grafica nella zona di memoria compresa tra 4000 e 407FF.
Indirizzo 1 ( Indirizzo 2. Indirizzo 3 V	Verifica l'identità tra 2 zone di memoria. Eventuali differenze sono segnalate: F100-01.00 F 0000 - 00 (00) 0000 - 00 (00) Il sottoprogramma CHECK nella sua forma originale differisce, nella forma attuale per i contenuti da 000 e 000 (operatore nel testo 00-000).
N	Visualizzazione in modo normale e separatore tra successivi comandi MONITOR.
I	Visualizzazione in modo inverso (nero su bianco).
Valore + Valore	Operazioni di addizione e sottrazione nel sistema di numerazione esadecimale (2 cifre). 3F-01 oppure 40-01 40 3F
Numero di connettore <u>CIRI-P</u>	Trasferisce una estratta in arrivo dalla periferica collegata allo slot specificato. Trasferisce il controllo dell'uscita alla periferica collegata allo slot specificato. è <u>CIRI-P</u> attiva la PROM della scheda di controllo del drive per floppy disk provocando l'inizializzazione del sistema.

Comando	Definizione - esempio
<u>CTRL-B</u>	Caricamento dell'interprete BASIC nella memoria RAM (inizializzazione "a freddo").
<u>CTRL-C</u>	Ricarica dell'interprete BASIC nella memoria RAM (uguale a RESET).
<u>CTRL-E</u>	Visualizzazione del contenuto dei registri del microprocessore. A= D= Y= P= S= Modifica di tali registri.
<u>CTRL-Y</u>	Salta al programma che comincia all'indirizzo. #170 3F0 : 4C 00 01 JMP #300 #370 dev'essere preparato per provocare il salto all'inizio del programma chiamato.
Indirizzo 1. Indirizzo 2 #	Registra su cassetta la zona di memoria posta tra l'indirizzo 1 e l'indirizzo 2 #000,000 #
Indirizzo 1. Indirizzo 2 #	Letture da cassetta. Carica in memoria RAM, partendo dall'indirizzo 1 fino all'indirizzo 2, dati e programmi registrati sulla cassetta.

# MINI-ASSEMBLER

Comando	Definizione - esempio
* Fede 8 oppure CALL-2408	Entrata nel Mini-Assembler la risposta è costituita da "I".
addr : COP <u>spazio</u> OPE	Istruzione assembler all'indirizzo " <u>addr</u> " (4 cifre esadecimali); il codice dell'istruzione è espressa in linguaggio mnemonico (3 caratteri) : <u>COP</u> consente i 56 codici operativi del 4502; la parte OPE ha un formato variabile che segue il modo di indirizzamento. /100: 104000 <u>R10000</u> restituisce 100-40 00 104000 questa istruzione consente di porre a zero il registro accumulatore.
<u>spazio</u> COP OPE	Assemble l'istruzione all'indirizzo seguente.
<u>FORMATO</u> della parte relativa all'operando OPE	Se l'indirizzo è di due cifre agisce in pagina zero.
# valore	Indirizzamento immediato.
# indirizzo	Indirizzamento assoluto (pagina zero oppure che non segue COP).
# indirizzo, X	Indirizzo indicizzato da X (pagina zero oppure che non segue COP).
# indirizzo, Y	Indirizzo indicizzato da Y (pagina zero oppure che non segue COP).
# indirizzo	Indirizzamento relativo (per le istruzioni di salto all'indirizzo).
(# indirizzo, X)	Indirizzamento indiretto pre-indicizzato da X.
(# indirizzo, Y)	Indirizzamento indiretto pre-indicizzato da Y.

Comando	Definizione - esempio
(R indirizza)	Indirizzamento indiretto (per l'istruzione di salto JMP). /100:10A 000,X 100-00 00 / spazio PCX 100-CA / spazio BNC 0100 104-00 FB
R	Chiamata al Monitor /4000.100 per verificare il contenuto di una zona assemblata.

## COMANDI RELATIVI AL SISTEMA OPERATIVO DOS

Salvastaggio - caricamento dei programmi

LOAD "NOME",D1	Carica in memoria RAM il programma "NOME" dal dischetto posto nell'unità 1.
SAVE "NOME",D2	Salva un programma BASIC che risiede in memoria RAM sul dischetto posto nell'unità 2 con il nome "NOME".
BLOAD "BINARIO"	Carica in memoria RAM il file binario "BINARIO" del dischetto all'indirizzo assoluto registrato in testa al file. Questo indirizzo e la lunghezza del file sono nella memoria RAM all'indirizzo \$AAT2 e \$AA60 prima del caricamento.
BSAVE "BINARIO",A\$300,L\$7F	Salva il programma in linguaggio macchina posto in memoria partendo dall'indirizzo assoluto \$300 per una lunghezza di 7F byte sul dischetto corrente sotto il nome di "BINARIO".
BSAVE "IMG",A\$2000,L\$1FFF	Salva l'immagine grafica in alta risoluzione contenuta nella pagina 1 sul dischetto con il nome "IMG".
BSBLOAD "IMG"	Sostituisce sullo schermo grafico ad alta risoluzione i punti registrati nel file "IMG".
RUN "NOME"	Carica in memoria RAM da disco ed esegue il programma che si chiama "NOME".
BRUN "BINARIO"	Carica in memoria RAM da disco ed esegue il programma scritto in linguaggio macchina che si chiama "BINARIO".
CHAIN "NOME"	Carica in memoria RAM da disco un programma scritto in BASIC Integer senza cancellare la zona delle variabili del programma precedente. Il programma "NOME" non deve avere variabili comuni ridimensionate.

SE=CHR\$(13)+CHR\$(14).

File sequenziali (batteria T)

PRINT @@"OPEN FI"

Apre sul disco selezionato un file denominato FI. Posiziona il puntatore all'inizio del file sequenziale.

PRINT @@"OPEN"FB",01"

Apre sul disco posto nell'unità 1 un file di nome variabile FB.

PRINT @@"READ FI"

Prepara una operazione di lettura all'inizio del file sequenziale FI aperto con il comando OPEN.

INPUT @B

Legge nel file FI una stringa di caratteri che sarà memorizzata in memoria RAM sotto la variabile @B. Il puntatore è posto all'inizio del campo dei dati seguente.

GET @C

Legge un solo carattere e sposta il puntatore di un carattere.

PRINT @@"POSITION FI,R"P

Posiziona il puntatore dopo il P-lesimo RETURN posto dopo la posizione attuale.

PRINT @@"WRITE FI"

Prepara un'operazione di scrittura sul file FI dove si trova il puntatore.

PRINT @B

Scrivo nel file FI la stringa @B.

PRINT @B

Scrivo la stringa @B separata dalla stringa @B, precedentemente registrata, da un carattere RETURN.

PRINT CHR\$(14)"CLOSE FI"

Chiude il file FI salvando su disco i parametri relativi alla operazione di OPEN effettuata sul file contenente l'ultimo settore utilizzato.

PRINT @@"APPEND FI"

Ritrova il file FI e posiziona il puntatore alla fine del file. Permette di scrivere dati e aggiunte alla fine del file.

PRINT @@"READ FI,B17"

Posiziona il puntatore al byte 17 per una successiva lettura 10 è il primo byte).

## COMANDI RELATIVI AL SISTEMA OPERATIVO DOS

**DB=CHR\$(13)+CHR\$(14)**

File ad accesso diretto (battere ?)

**PRINT DB"OPEN PIP,L31"**

Apri il file PIP su disco prevedendo record di lunghezza costante; in questo caso i 31 byte previsti comprendono i RETURN assenti come fine del record.

**PRINT DB"OPEN"PB",L"NB",B2"**

Apri un file di nome variabile PB di lunghezza uguale alla variabile NB sul disco sotto nell'unità 2.

**PRINT DB"READ PIP,R"J**

Posiziona il puntatore all'inizio dell'J-esimo record per una successiva lettura.

**INPUT AB**

Recupera in AB il contenuto dell'J-esimo record nel file PIP.

**GET CB**

Legge il carattere posto sotto il puntatore nel record I.

**PRINT DB"WRITE PIP,R"J**

Posiziona il puntatore all'inizio del J-esimo record per una successiva scrittura.

**PRINT IB**

Scrivi la stringa IB nel record J.

**PRINT PB**

Scrivi la stringa PB, separata dalla stringa precedentemente registrata, con RETURN.

**PRINT CHR\$(14)"CLOSE PIP"**

Chiude il file PIP salvando i parametri relativi.

**PRINT DB"READ PIP,R0,B"K**

Posiziona il puntatore al byte K del record 0 per una ulteriore lettura.

**PRINT CHR\$(14)**

Annulla l'effetto di un comando DOS, come READ, per esempio per dare all'istruzione INPUT il senso di una impostazione da tastiera.

# Comandi diversi

## CATALOG 02

Visualizza l'elenco dei programmi e dei file registrati sul dischetto posto nell'unità 2.

\* file protetti dalle cancellazioni accidentali.

I, A, T, B tipi di files

I = BASIC Integer

A = BASIC Applesoft

T = File TEXT sequenziali o diretti

B = File binari, dati o programmi in linguaggio macchina.

002 = Numero dei settori occupati dal file (modulo 256).

Un settore comprende 256 byte utili.

Una traccia comprende 13 o 14 settori

secondo il DOS impiegato (3.2 o 3.3).

Un dischetto comprende 35 tracce delle

quali 31 sono utilizzabili dall'utente e 4 riservate al sistema.

Il numero massimo di nomi di file che possono essere registrati su disco è 64 in DOS 3.2 e 105 in DOS 3.3.

## REN C,1,0

Visualizza i comandi, le entrate e le uscite come sono ricevute o inviate dal sistema operativo durante l'esecuzione di un programma.

Il comando si annulla con RESET.

## RENOM C,1,0

Annulla il comando REN.

## PRINT 00\*PR0\*0

Attiva la periferica collegata allo slot "0" per l'esecuzione della successiva istruzione PRINT.

Se l'interfaccia relativa alla stampante risiede nello slot 1:

PRINT 00\*PR01\*0

## PRINT 00\*PR00\*

Disattiva la periferica di uscita lasciando in linea solo lo schermo.

## PRINT 00\*IN0\*0

Connette la periferica collegata allo slot "0" per l'acquisizione di dati da inviare al calcolatore.



## COMANDI RELATIVI AL SISTEMA OPERATIVO DOS

**PRINT DP"IMMO"**

Limita alla tastiera la possibilità di input verso il calcolatore.

**RA0FILE 4**

Prevede l'utilizzo di 4 buffer di input/output aperti contemporaneamente in parallelo corrispondenti a 4 files aperti disponibili in memoria. Ciascun buffer occupa 393 byte. Per default il sistema riserva 3 zone riservate ai buffer. Questo comando dev'essere eseguito prima del caricamento e dell'esecuzione del programma.

**VERIFY NAME**

Verifica la buona registrazione fisica del programma o del file NAME. Se un settore del dischetto utilizzato è danneggiato sarà visualizzato il messaggio I/O ERROR (la verifica avviene anche automaticamente dopo il comando SAVE).

### Elaborazione dei file

**INIT HELLO C ,V254 3**

Procedura di inizializzazione di un dischetto vergine. Il programma BASIC che si trova in memoria RAM è caricato sul dischetto con il nome HELLO insieme al sistema operativo.

Il dischetto porta un numero di volume che può essere utile per eventuali controlli.

Il formato con il quale viene inizializzato il dischetto dipende dal sistema operativo che risiede in memoria RAM.

DOS 3.2 : 13 settori per traccia

DOS 3.3 : 16 settori per traccia

Lo spazio disponibile è:

DOS 3.2 : 103.140 byte per dischetto

DOS 3.3 : 126.976 byte per dischetto

Il sistema operativo registrato sul dischetto è del tipo "chiave" e contiene solo il programma di rilocazione per adattarsi nel modo migliore alla configurazione del sistema.

**DELETE NOME** Sopprime il file NOME dal dischetto se il file non è LOCK (protetto contro la cancellazione).

**RENAME VECCHIO,NUOVO** Cambia il nome del file o di un programma.

**LOCK NOME** Protegge il file NOME contro le cancellature accidentali che possono essere provocate dai comandi DELETE, SAVE oppure WRITE.

**UNLOCK NOME** Toglie la protezione contro le cancellature dal file NOME.

# File di comando

**LIST**  
**PRINT ON"OPEN CMD"** Creazione di un file  
**PRINT ON"WRITE CMD"**  
**PRINT "PRG1"** Registrazione degli ordini  
**PRINT CHR(10)"GO N"**  
**PRINT "LIST"**  
**PRINT "PRG2"**  
**PRINT ON"CLOSE CMD"** Chiusura del file

**RUN** Realizzazione del file

**EXEC CMD** Esecuzione automatica dei comandi registrati nel file CMD nel nostro esempio stampa di un listing sulla stampante con un formato di 40 caratteri per linea.

Tutte le istruzioni e i comandi eseguibili in modo diretto possono appartenere ad un file del tipo appena esaminato.



# CARATTERI

CONVERSIONE ESADECIMALE/DECIMALE/ESADECIMALE

Dei primi 256 numeri da 000 a 1FF

000	10	1	2	3	4	5	6	7
008	18	9	10	11	12	13	14	15
010	116	17	18	19	20	21	22	23
018	124	25	26	27	28	29	30	31
020	132	33	34	35	36	37	38	39
028	140	41	42	43	44	45	46	47
030	148	49	50	51	52	53	54	55
038	156	57	58	59	60	61	62	63
040	164	65	66	67	68	69	70	71
048	172	73	74	75	76	77	78	79
050	180	81	82	83	84	85	86	87
058	188	89	90	91	92	93	94	95
060	196	97	98	99	100	101	102	103
068	1104	105	106	107	108	109	110	111
070	1112	113	114	115	116	117	118	119
078	1120	121	122	123	124	125	126	127
080	1128	129	130	131	132	133	134	135
088	1136	137	138	139	140	141	142	143
090	1144	145	146	147	148	149	150	151
098	1152	153	154	155	156	157	158	159
0A0	1160	161	162	163	164	165	166	167
0A8	1168	169	170	171	172	173	174	175
0B0	1176	177	178	179	180	181	182	183
0B8	1184	185	186	187	188	189	190	191
0C0	1192	193	194	195	196	197	198	199
0C8	1200	201	202	203	204	205	206	207
0D0	1208	209	210	211	212	213	214	215
0D8	1216	217	218	219	220	221	222	223
0E0	1224	225	226	227	228	229	230	231
0E8	1232	233	234	235	236	237	238	239
0F0	1240	241	242	243	244	245	246	247
0F8	1248	249	250	251	252	253	254	255

CARATTERI

# CODICI DI TASTIERA

SHIFT	A1	A2	A3	A4	A5	A6	A7	A8	A9	B0	BA	BB	CTRL	
	A1	A2	A3	A4	A5	A6	A7	A8	A9	B0	BA	BB		
	1	2	3	4	5	6	7	8	9	0	.	-	RESET	
	1	2	3	4	5	6	7	8	9	0	.	-		
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B0	BA	BB	CTRL	
B1	B2	B3	B4	B5	B6	B7	B8	B9	B0	BA	BB			
SHIFT	9B	91	97	05	92	94	99	95	09	0F	00		00	CTRL
	9B	91	07	C5	02	04	09	05	C9	CF	C0		00	
	ESC	B	W	E	R	T	Y	U	I	O	P	REPT	RETURN	
	9B	91	07	C5	02	04	09	05	C9	CF	00		00	
	9B	91	97	05	92	94	99	95	09	0F	90		00	CTRL
SHIFT	01	93	04	06	07	08	0A	0B	0C	0D	0E	95	CTRL	
	C1	03	C4	C6	C7	C8	C9	CC	0B	95	95			
	CTRL	A	S	D	F	B	J	K	L	.	<-	->		
		C1	03	C4	C6	C7	C8	C9	CC	0B	95	95		
		01	93	04	06	07	08	0A	0B	0C	0D	0E	95	CTRL
SHIFT	94	98	03	96	9E	9D	9B	9C	9E	9F			CTRL	
	0A	09	C3	04	C2	0F	0A	0C	0E	0F				
	SHIFT	Z	X	C	V	B	N	M	.	/	T			SHIFT
		04	08	C3	06	C2	CE	CD	AC	AE	AF			
		94	98	03	96	9E	9D	9B	9C	9E	9F			
SHIFT	A0												CTRL	
	A0													
	SPACE													
	A0													
	A0												CTRL	

I codici ASCII estesi 17 bit più uno dei tasti in tastiera in esadecimale.

# Memoria

## LIST

```
10 HOME
20 FOR I = 160 TO 191
30 PRINT CHR$(I);
40 NEXT I
50 PRINT
60 FOR I = 192 TO 223
70 PRINT CHR$(I);
80 NEXT I
```

## END

```
"*#0123456789:;C=/?
@ABCDEFGHIJKLMNPQRSTUWXYZ*\&"_
```

## CALL-151

### \*400.41F

```
0400- A0 A1 A2 A3 A4 A5 A6 A7
0408- A8 A9 AA AB AC AD AE AF
0410- B0 B1 B2 B3 B4 B5 B6 B7
0418- B8 B9 BA BB BC BD BE BF
```

1ª linea di schermo

### \*480.49F

```
0480- C0 C1 C2 C3 C4 C5 C6 C7
0488- C8 C9 CA CB CC CD CE CF
0490- D0 D1 D2 D3 D4 D5 D6 D7
0498- D8 D9 DA DB DC DD DE DF
```

2ª linea di schermo

# CODICI DI SCHEMA

## Inverso

RUN  
/ ^#478 1 1 1 \_ \_ \_ / 0123456789 r j c = 17  
@ ABCDEFGHIJ K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9 0

1° linea

2° linea

CALL-151

\*400,41F

0400- 20 21 22 23 24 25 26 27  
0408- 28 29 2A 2B 2C 2D 2E 2F  
0410- 30 31 32 33 34 35 36 37  
0418- 38 39 3A 3B 3C 3D 3E 3F

1° linea

\*480,49F

0480- 00 01 02 03 04 05 06 07  
0488- 08 09 0A 0B 0C 0D 0E 0F  
0490- 10 11 12 13 14 15 16 17  
0498- 18 19 1A 1B 1C 1D 1E 1F

2° linea

# Flash

## RUN

0a b c d e f g h i j k l m n o p q r s t u v w x y z 0 0 0  
0A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 4 0 \_

LYLL-151

\*40H, 41F

0400- 60 61 62 63 64 65 66 67  
0408- 68 69 6A 6B 6C 6D 6E 6F  
0410- 70 71 72 73 74 75 76 77  
0418- 78 79 7A 7B 7C 7D 7E 7F

1° linea

\*460, 49F

0460- 40 41 42 43 44 45 46 47  
0468- 48 49 4A 4B 4C 4D 4E 4F  
0470- 50 51 52 53 54 55 56 57  
0478- 58 59 5A 5B 5C 5D 5E 5F

2° linea

## Codici di schermo

In uscita della stampante, i codici da 840 a 87F corrispondono ai codici ASCII relativi ai caratteri minuscoli, questa è la ragione per la quale sono stati stampati in minuscolo dalla stampante quando sono stati attivati come caratteri lampeggianti.

Il gruppo 0 1 2 ... 9 10 ... 19 ... 29 è quello delle cifre.

Il gruppo @ # \$ % & ..... 1 2 ... è quello delle lettere.

LETTERE IN MODO INVERSO	800 - 81F
CIFRE IN MODO INVERSO	820 - 83F
LETTERE IN MODO LAMPEGGIANTE	840 - 85F
CIFRE IN MODO LAMPEGGIANTE	860 - 87F

CIFRE IN MODO NORMALE	880 - 89F
LETTERE IN MODO NORMALE	900 - 91F



ASCII (POSITIVE) UNIT 7-80

Codici dei caratteri in esadecimale

00 :    " " "A "B "C "D "E "F "G

08 :    "H "I "J "K "L "M "N "O

10 :    "P "Q "R "S "T "U "V "W

18 :    "X "Y "Z esc . "SH " " .

20 :    sp :    " " " " " " " " " "

28 :    ( ) \* + , - . /

30 :    0 1 2 3 4 5 6 7

38 :    8 9 : ; < = > ?

40 :    " " "A "B "C "D "E "F "G

48 :    "H "I "J "K "L "M "N "O

50 :    "P "Q "R "S "T "U "V "W

58 :    "X "Y "Z [ \ ] ^ \_

60 :    " " "a "b "c "d "e "f "g

68 :    " " "h "i "j "k "l "m "n "o

70 :    " " "p "q "r "s "t "u "v "w

78 :    " " "x "y "z { | } ~

A per CTRL-A

"B = backspace, ←

"J = salto di linea CRLF Line Feed

"L = salto pagina CFF Form Feed

"M = ritorna carrello, RETURN

"G = NUL, →

ESC = escape

"SH= CTRL SHIFT H

"S = BELL (campanello)

Z = SHIFT Z

. = non possono essere introdotti da tastiera.

## Codici dei caratteri in esadecimale

40 : ! " # \$ % &amp; ' (

A0 : ( ) \* + , - . /

80 : 0 1 2 3 4 5 6 7

88 : 8 9 : ; &lt; = &gt; ?

C0 : @ A B C D E F G

C8 : H I J K L M N O

D0 : P Q R S T U V W

D8 : X Y Z [ \ ] ^ \_

E0 : ` a b c d e f g

E8 : h i j k l m n o

F0 : p q r s t u v w

F8 : x y z { | } ~

## Caratteri di controllo

80 : ^ \_ ` A B C D E F G

88 : H I J K L M N O

90 : P Q R S T U V W

98 : X Y Z [ \ ] ^ \_

^U = ^, ^V

^H = ^, ^B

^M = RETURN, ^D

# CONVERSIONE ESADECIMALE/DECIMALE

Numeri di 4 cifre esadecimali: H3 H2 H1 H0

CODICE	H3	H2	H1	H0
0	0	0	0	0
1	4096	256	16	1
2	8192	512	32	2
3	12288	768	48	3
4	16384	1024	64	4
5	20480	1280	80	5
6	24576	1536	96	6
7	28672	1792	112	7
8	32768	2048	128	8
9	36864	2304	144	9
A	40960	2560	160	10
B	45056	2816	176	11
C	49152	3072	192	12
D	53248	3328	208	13
E	57344	3584	224	14
F	61440	3840	240	15

## CONVERSIONE ESADECIMALE/DECIMALE

Il numero decimale è ottenuto facendo la somma dei valori presi all'intersezione della linea relativa alle cifre esadecimali e la colonna alla posizione delle cifre nel numero esadecimale.

Esempio: 84FFh diventa 40960 (8 in H3)  
+ 3840 (F in H2)  
+ 240 (F in H1)  
+ 15 (F in H0)

45045

# MESSAGGI DI ERRORE

APPLESOFT

I messaggi di errore, in un programma Applesoft, hanno la seguente forma:

? messaggio ERROR IN numero di linea

"messaggio" è il nome dell'errore.

Il numero di linea è quello relativo alla linea che contiene l'istruzione nella quale è stato riscontrato l'errore. Gli errori sono rilevati solo quando si procede all'esecuzione del programma.

Quando è riscontrato un errore, l'interprete BASIC Applesoft provoca l'arresto del programma e la visualizzazione del messaggio relativo. Le variabili e le istruzioni non vengono modificate, ma il programma non può procedere. I contatori dei cicli FOR-NEXT sono rimessi a zero e l'eventuale istruzione GOSUB è annullata.

Grazie all'istruzione ONERR GOTO e ad un sottoprogramma di trattamento degli errori un programma può malgrado tutto proseguire normalmente.

Le istruzioni date in modo diretto (senza numero di linea) possono provocare l'emissione di un messaggio di errore, in tal caso il messaggio non conterrà l'indicazione del numero di linea.

Ciascun tipo di errore è associato ad un codice che è posto, al momento dell'errore, all'indirizzo decimale 222 (oppure, in esadecimale, 00E).

Il numero di linea dove l'errore si è verificato figura negli indirizzi decimali 218 e 219 (oppure, in esadecimale, 004 e 005). Il valore del puntatore TOPR nell'istruzione errata è posto agli indirizzi decimali 220 e 221 (oppure, in esadecimale, 00C e 00D). Il valore dei puntatori relativi allo stack, al momento dell'errore, è conservato all'indirizzo decimale 223 (oppure, in esadecimale, 00F). Queste informazioni devono essere ripristinate prima di procedere al trattamento dell'errore; questo può essere fatto grazie al seguente programma:

200	40	PLA
201	40	TAY
202	40	PLA
203	A6 DF	LDX 0DF
204	94	TIX
205	40	PHA
206	98	TYA
207	40	PHA
208	60	RTS

Tale sottoprogramma può essere scritto, nella memoria del calcolatore, per mezzo delle seguenti istruzioni:

# APPLESOFT

0 PEEK 216,0 : PEEK 768,104 : PEEK 768,168 : PEEK 770,104 :  
 PEEK 771,168 : PEEK 772,223 : PEEK 773,134 : PEEK 774,72 :  
 PEEK 775,152 : PEEK 776,72 : PEEK 777,96

All'indirizzo decimale 216 figura l'indicazione di attivazione (800)  
 oppure la disattivazione (000) dell'istruzione ONERR 8070.

Segue un programma che tien conto di eventuali errori e del  
 conseguente arresto del programma:

```

0 memorizzazione del programma che segue
1 ONERR 8070 1000 : attivazione del sistema
10 normale svolgimento del programma
999 END
1000 CALL 768 : esecuzione del sottoprogramma
1010 IF PEEK(222)=0 THEN prende in considerazione l'errore 3
1020 verifica di un altro tipo di errore
1030 RESUME per tornare all'istruzione errata se necessario
  
```

Il sottoprogramma che controlla lo svolgimento del programma in caso  
 di errore è posto all'indirizzo esadecimale #B412. Provoca  
 l'esecuzione del sottoprogramma HANDLER all'indirizzo esadecimale  
 #F2E9 se ONERR è attivo. HANDLER sistema le locazioni di memoria #B4  
 e #0F.

Se ONERR non è stato utilizzato, il sottoprogramma #B412 arresta  
 l'esecuzione del programma e visualizza il messaggio di errore.

Nell'interprete Applesoft, la tabella dei messaggi di errore è  
 posizionata partendo dall'indirizzo #B260 e i loro codici  
 corrispondono alla loro posizione in questa tabella.

```

NEXT WITHOUT FORSYNTAXRETURN WITHOUT GO
SUBOUT OF DATAILLEGAL QUANTITYOVERFLOWOU
T OF MEMORYUNDEF'D STATEMENTBAD SUBSCRIP
TREDIM'D ARRAYDIVISION BY ZEROILLEGAL DI
TOO COMPLEXCAN'T CONTINUEUNDEF'D FUNCTIO
N ERROR IN
BREAK
  
```

## Elenco dei messaggi di errore in ordine alfabetico

Codice	Messaggio	Origine	Commento
107	TOO SUBSCRIPT (indice errato)	BIM	Tentativo di chiamare un elemento di una tabella di indice superiore al limite fissato con l'istruzione BIM. Esempio: BIM #151 con #1200 oppure ancora con un numero d'indice differente da quelli previsti da BIM. L'Applesoft dimensiona automaticamente a 11 le variabili indicizzate non dichiarate.
210	CAN'T CONTINUE (non posso continuare)	CONT	Impossibilità di riprendere l'esecuzione del programma con CONT. In caso di errore oppure di inserimento o modifica di una istruzione. In certi casi si può fare ripartire il programma con un GOTO ad un numero di linea.
133	DIVISION BY ZERO (divisione per zero)	/0	Può derivare da una variabile non inizializzata con un valore diverso da zero.
	TEXTS IGNORED (troppi dati) ignorati	INPUT	Se i dati, separati da virgole, sono in numero superiore a quello delle variabili previste, il programma prosegue comunque.
191	FORMULA TOO COMPLEX (formula troppo complessa)	IF "stringa di caratteri" THEN	Il test non può essere richiesto per più di due volte nello stesso programma.

# APPLESOFT

Codice	Messaggio	Origine	Commento
149	ILLEGAL DIRECT (illegale in modo diretto)	Modo diretto	Le istruzioni INPUT, SET DEF FN e GOTO non possono essere usati in modo diretto.
53	ILLEGAL QUANTITY (valore errato)	Funzione automatica	<p>I parametri forniti in una funzione superano i limiti permessi.</p> <ul style="list-style-type: none"> <li>- L'indice di una variabile è negativo</li> <li>- L'argomento di LOG è negativo o nullo</li> <li>- L'argomento di SQK è negativo. A potenza di B se B è negativo e B non è intero.</li> </ul>
93	ILLEGAL QUANTITY	HIGH LEFT RIGHT  CHR  ASC  CALL  FOR "	<p>La lunghezza o l'indice di posizionamento non è compreso tra 1 e 255.</p> <p>Il codice non è compreso tra 0 e 255.</p> <p>Il carattere è di lunghezza 0.</p> <p>L'indirizzo non è compreso tra -65535 e +65535.</p> <p>L'indirizzo non è compreso tra -65535 e +65535; il valore non è compreso tra 0 e 255.</p>
		DIMEN  HPLOT DRAW PLOT,VLIN SLIN POL	<p>L'indirizzo non è compreso tra -65535 e +65535.</p> <p>X,Y0 o X&gt;278 e Y&gt;191            X,Y0 o X&gt;278 e Y&gt;191            X,Y0 o X,7&gt;27            X0 o X&gt;255</p>

Codice	Messaggio	Origine	Commento
		HTAB VTAB SPC TAB1	100 a 10255 100 a 1024 100 a 10255 100 a 10255
		ON...GOTO ON...GOSUB	L'indice non deve superare 255 e non deve essere inferiore a 0. Se il valore dell'indice è nullo o più elevato del numero del numero di linea specificato, esecuzione procede alla linea successiva.
9	NEXT WITHOUT FOR (NEXT senza FOR)	FOR, NEXT	Cicli FOR...NEXT sono mal posizionati. Esempio: FOR I=1 TO ... FOR Y=1 TO ... PRINT I,Y NEXT I NEXT Y (scrivere NEXT Y NEXT I) Manca un FOR per un NEXT isolato.
42	TOOT OF DATA (dati mancanti)	READ	Prova di esecuzione di una istruzione READ quando tutti i DATA sono già stati letti. Prevedere un carattere per testare la fine dei dati o una variabile di conteggio oppure eseguire RESTORE per rileggere i dati all'inizio dei DATA.
		RECALL STORE	Non utilizzare variabili il cui nome comincia con RECALL o STORE.
77	TOOT OF MEMORY (memoria mancante)	GIM GOSUB HIREM:	Non superare il numero massimo di indici: 88. Non può gestire oltre 24 livelli di chiamata. Non fissare valori troppo bassi.



Codice	Messaggio	Origine	Commento
		LOREM	Non fissare valori troppo alti oppure sotto il valore attuale. Il programma è troppo grande o le variabili troppo numerose.
66	OVERFLOW ERROR (superò di capacità)	Numero reale	Risultato superiore a 1.7E38. Un numero reale è rappresentato con un byte di esponente e 4 byte per la mantissa. Se il risultato è inferiore a 2.9E-39, equivalente a 0 senza messaggio di errore.
		STR	Se il numero da convertire in stringa di caratteri è troppo grande.
		VAL	Se il valore assoluto del numero cercato è superiore a 1E38 o se il numero contiene più di 13 cifre.
120	REDIM'D ARRAY (matrice ridimensionata)	DB	Una stessa matrice non può essere ridimensionata due volte.
	REENTER (rifare l'impostazione dei dati)	INPUT	Si è eseguito un INPUT allanumerico; si devono rifare tutti i valori numerici attesi dall'istruzione INPUT.
22	RETURN WITHOUT (RETURN senza GOSUB)	RETURN	Un sottoprogramma è stato posto prima della fine logica del programma oppure è stata dimenticata l'istruzione DB. Nel trattamento di un errore, riprendere da un sottoprogramma senza eseguire GOSUB.

Codice	Messaggio	Origine	Commento
174	STRING TOO LONG (stringa di caratteri troppo lunga)	IF  LEN VAL  PRINT	Non creare una stringa di caratteri per la concatenazione la cui lunghezza superi 255. Se l'argomento è una stringa di lunghezza totale superiore a 255, APPLE ha più di 255 caratteri scrivere PRINT APPLE.
18	SYNTAX (errore di sintassi)	ASC  CONT	Istruzione incomprensibile per l'interprete Applesoft. - parentesi non chiuse - caratteri illegali - punteggiatura errata - errore di ortografia in una parola chiave  Su "CTRL-@" o "CTRL-0".  Se i dati relativi a INPUT sono interrotti con CTRL-C e si tenta di ripartire con CONT.
18	SYNTAX (errore di sintassi)	DATA	Una stringa di caratteri contenente ? non è accettata.
14	SYNTAX (errore di sintassi)	DEL  FOR...NEXT  FOR FOR2 TEXT	Dev'essere seguito da 2 numeri di linea di ordine crescente.  Non utilizzare una variabile di tipo intero (I) come indice di ciclo.  Non utilizzare queste parole chiave come i primi caratteri di una variabile (sarebbero eseguiti prima del messaggio di errore).

# APPLESOFT

Codice	Messaggio	Origine	Commento
		IF...THEN	Manca il THEN dell'IF corrispondente.
		LIST, 0	Visualizza il programma completo seguito da TERNAL.
		RECALL	Non utilizzare queste parole chiave come i primi caratteri di una variabile.
		STORE	
		UNLOAD	
		RESUME	L'istruzione viene riscontrata prima che sia prodotto un errore. Può essere un errore fatale.
		N'a'istruzione errata.	Se il carattere 0 è dato al posto del numero 0 e se la lettera I è data al posto del numero 1.
143	TYPE MISMATCH (disaccordo tra numerico e alfanumerico)	LET RICH LEFT RIGHT	- una variabile stringa non può ricevere un numerico e viceversa - errato il tipo di argomento.
224	UNDEF'D FUNCTION (funzione non definita)		Riferito ad una funzione per la quale non esiste l'istruzione DEF FN.
90	UNDEF'D STATEMENT (statement indefinito)	GOTO GOSUB ON...GOTO RUN THEN	Rinvio ad un numero di linea inesistente.

## MESSAGGI DI ERRORE RELATIVI ALLA GESTIONE DEI DISCHI

Nel sistema operativo dei dischi caricato in memoria RAM (in un sistema da 48 Kbyte), la tabella dei messaggi di errore è posta a partire dall'indirizzo 8A971. Il primo messaggio è 'RETURN', 'NELL', 'RETURN'. I seguenti sono:

```

LANGUAGE NOT AVAILABLE
RANGE ERROR
WRITE PROTECTED
END OF DATA
FILE NOT FOUND
VOLUME MISMATCH
I/O ERROR
DISK FULL
FILE LOCKED
SYNTAX ERROR
NO BUFFERS AVAILABLE
FILE TYPE MISMATCH
PROGRAM TOO LARGE
NOT DIRECT COMMAND

```

Nella rosa seguente, da 8A97F a 8A98F, sono registrate le posizioni di inizio di ciascun messaggio contenute nella tabella precedente:

0	1	25	26	34	51	62	74	91	100	109	120	132	152	170	187	DEC.
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	CODICE
824 833																
ESAS.																

Esempio: il messaggio WRITE PROTECTED è il quarto della lista, il suo codice di errore è 4 e il testo è memorizzato dall'indirizzo 8A97F+824 all'indirizzo 8A97F+832.

Una corretta analisi consentirà la traduzione in italiano dei messaggi di errore inviati dal sistema operativo.

**MESSAGGI DI ERRORE DEL SISTEMA OPERATIVO**  
(in ordine alfabetico)

Codice	Messaggio	Origine	Commento
9	DISK FULL (disco pieno)	SAVE WRITE	La directory è stata riempita oppure tutti i settori sono stati utilizzati.
3	END OF DATA (fine dei dati)	INPUT  APPEND READ  POSITION READ  EXEC F,Rr  READ F,Rr	I dati sono insufficienti per soddisfare l'istruzione INPUT. Dopo questa istruzione è ammessa solo l'istruzione WRITE. La POSITION specificata non corrisponde ad alcun dato registrato. Se r corrisponde al 2°o campo dopo la fine del file. Se r corrisponde ad una registrazione non ancora effettuata (codice 0).
10	FILE LOCKED (file o programma protetto)	SAVE DELETE ISAVE WRITE	Al file è associato un asterisco nel CATALOG e non può essere modificato in scrittura (salvo con APPEND).
4	FILE NOT FOUND (file non trovato)	LOAD RUN BLOAD BRUN DELETE	Il file chiesto non esiste sul dischetto. Verificare l'ortografia del suo nome.
12	FILE TYPE MISMATCH (disaccordo sul tipo di file)	LOAD BLOAD RUN BRUN	Un file di tipo T o B non può essere caricato con LOAD o RUN; un file di tipo I o A non può essere chiamato con BLOAD o BRUN.
13	FILE TYPE MISMATCH (disaccordo sul tipo di file)	OPEN READ POSITION WRITE APPEND EXEC CLOSE	Questi comandi sono operativi solo su file di dati (tipo T).

**MESSAGGI DI ERRORE DEL SISTEMA OPERATIVO**

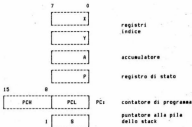
Codice	Messaggio	Origine	Commento
		CHAIN	Un programma in BASIC Applesoft non può essere 'CHAIN' con questo solo comando che riguarda solo i programmi BASIC Integer.
0	I/O ERROR (errore di Input/ Output)	Tutti i comandi	<ul style="list-style-type: none"> <li>- lettura senza disco</li> <li>- n. di slot (senza il controller dei dischi)</li> <li>- disco rovinato</li> <li>- disco non inizializzato</li> <li>- porta di lettura aperta</li> </ul>
		VERIFY	Se c'è un errore dopo la verifica che un file è registrato correttamente o meno.
1	LANGUAGE NOT AVAILABLE (interprete BASIC non presente)	LOAD PP INT	Un programma BASIC non può essere eseguito se non è presente il relativo interprete.
		APPLE II PLUS	Non ha l'interprete BASIC Integer standard.
		SCHEMA LINGUAGGI	Se la scheda madre contiene la ROM un interprete, la scheda linguaggi potrà caricare l'altro.
12	NO BUFFER AVAILABLE (troppi file aperti in memoria RAM)	MAXFILES n	Il numero massimo è 16 (il sistema ne utilizza 1 per ciascun comando). Per default 3 file.
13	NOT DIRECT COMMAND (comando diretto illegale)	OPEN READ WRITE APPEND POSITION	Non possono essere utilizzati in modo diretto. - scrivere un programma che contiene questi comandi in una PRINT.

# MESSAGGI DI ERRORE DEL SISTEMA OPERATIVO

Codice	Messaggio	Origine	Commento
14	PROGRAM TOO LARGE (programma troppo grande)	LOAD RUN	MINEM è troppo bassa (il sistema operativo compara il numero dei settori del programma con il byte più significativo di MINEM).
2, 3	RANGE ERROR (valore errato)	V D S L R B A MAXFILES	Min - Max Volume 0 - 245 Drive 0 - 2 Slot 1 - 7 Lunghezza 1 - 32767 Numero 0 - 32767 Numero 0 - 32767 Indirizzo 0 - 65535 Numero dei file aperti 1 - 16
11	SYNTAX ERROR (errore di sintassi in un comando di DOS)	INT EXEC da LI INMa PAGa	Comando senza parametri istruzione BASIC non valido. a è negativo i > 65535 s non può essere superiore a 7.
7	VOLUME MISMATCH (disaccordo con il numero di volume)	V	Il volume del disco corrente è diverso da quello del disco richiesto. Se la richiesta è fatta con VD, la verifica del volume non potrà essere effettuata.
4	WRITE PROTECTED (protetto in scrittura)		La faccia sul disco è coperta, il file è accessibile solo in lettura. Il disco può essere inserito al contrario. Il dischetto SYSTEM MASTER è sempre protetto.

# IL LINGUAGGIO MACCHINA

## REGISTRI INTERNI DEL 6502




### Dettaglio sul registro di stato P



- |   |                  |   |               |
|---|------------------|---|---------------|
| N | segno            | D | modo decimale |
| V | overflow         | I | interrupt     |
| D | inutilizzato     | I | zero          |
| B | comando di Break | C | carry         |



## SET DI ISTRUZIONI DEL 4302

- ADC** : Addizione con riporto (Add with Carry) :  $A \leftarrow A + M + C$   
 Si aggiunge all'accumulatore la memoria specificata più il bit di riporto.  
 Opera in modo binario e decimale. Agisce su  $N$ ,  $V$ ,  $I$ ,  $C$ .
- AND** : And logica (AND) :  $A \leftarrow A \wedge M$   
 Esegue l'and bit per bit nell'accumulatore e nella memoria.  
 Agisce su  $N$ ,  $I$ .
- ASL** : Spostamento verso sinistra (Arithmetic Shift Left)  
  
 Sposta a sinistra il contenuto dell'accumulatore o di una memoria. Agisce su  $N$ ,  $I$ ,  $C$ .
- BCC** : Esegue il salto se il carry è uguale a 0 (Branch on Carry Clear). Se il bit  $C = 0$ , si salta all'istruzione indicata; in caso contrario si procede in sequenza.
- BCS** : Esegue il salto se il carry è uguale a 1 (Branch on Carry Set). Se il bit  $C = 1$ , si salta all'istruzione indicata; in caso contrario si procede in sequenza.
- BEQ** : Esegue il salto se il risultato = 0 (Branch on Equal)  
 Se il bit  $Z = 1$  (vale a dire se l'ultimo risultato è 0 o se l'ultima comparazione è risultata uguale), si passa all'istruzione specificata; in caso contrario si procede in sequenza.
- BIF** : Test del bit (Bit Test)  $Z \leftarrow \neg \bigwedge_{i=0}^{N-1} A_i \wedge M_i$ ,  $N \leftarrow N7$ ,  $V \leftarrow M6$ .  
 Esegue l'AND virtuale dell'accumulatore e della memoria specificata a posizione  $I$  di conseguenza. Inoltre i bit 7 e 6 della memoria sono copiati rispettivamente in  $N$  e  $V$ .
- BNI** : Esegue il salto se negativo (Branch on Minus).  
 Se il bit  $N = 1$ , si salta all'istruzione indicata; in caso contrario si procede in sequenza.
- BNE** : Esegue il salto se non uguale a 0 (Branch on Not Equal).  
 Se il bit  $Z = 0$  (vale a dire se l'ultimo risultato è diverso da 0 o se l'ultima comparazione è risultata diversa), si salta all'istruzione indicata; in caso contrario si procede in sequenza.
- BPL** : Esegue il salto se positivo o nullo (Branch on Not Equal)  
 Se il bit  $N = 0$ , si salta all'istruzione indicata; in caso contrario si procede in sequenza.

- BRK** : Interrupt software (BRake)  
 Pone il bit B a 1 e simula un'interruzione (salta all'indirizzo contenuto in PFFF, PFFF).
- BVC** : Esegue il salto se overflow è 0 (Branch on overflow Clear).  
 Se il bit V = 0, si salta all'istruzione indicata; in caso contrario si procede in sequenza.
- BVS** : Esegue il salto se overflow è 1 (Branch on overflow Set).  
 Se il bit V = 1, si salta all'istruzione indicata; in caso contrario si procede in sequenza.
- CLC** : Annulla il carry (CLear Carry).  
 Forza a 0 il bit di carry.
- CLD** : Annulla il modo decimale (CLear Decimal mode).  
 Forza a 0 il bit D per impostare il modo binario.
- CLI** : Autorizza le interruzioni (CLear Interrupt inhibit flag).  
 Forza a 0 il bit I d'inibizione delle interruzioni.
- CLV** : Annulla l'indicatore di overflow (CLear overflow flag).  
 Forza a 0 il bit V.
- CMF** : Comparazione con l'accumulatore (CoMPare accumulator) A ← M.  
 Effettua la sottrazione virtuale registro A - memoria e posizione, conseguentemente, gli indicatori N, I e C.
- CPI** : Comparazione con I (CoMPare with I) I ← M.  
 Effettua la sottrazione virtuale registro I - memoria e posizione, conseguentemente, gli indicatori N, I e C.
- CPT** : Comparazione con T (CoMPare with T) T ← M.  
 Effettua la sottrazione virtuale registro T - memoria e posizione, conseguentemente, gli indicatori N, I e C.
- DEC** : Decremento in memoria (DECrement memory) M ← M - 1.  
 Diminuisce di 1 il contenuto della memoria indicata. Agisce su M e I.
- DEX** : Decremento X (DECrement X) X ← X - 1.  
 Diminuisce di 1 il contenuto del registro X. Agisce su M e I.
- DEY** : Decremento Y (DECrement Y) Y ← Y - 1.  
 Diminuisce di 1 il contenuto del registro Y. Agisce su M e I.

# SET DI ISTRUZIONE DEL 4502

**ODR** : OR esclusivo (Exclusive OR).

Effettua l'OR esclusivo tra l'accumulatore e la memoria specificata. Agisce su R e I.

**INC** : Incremento della memoria (INCRement memory).

Aumenta di 1 il contenuto della memoria indicata. Agisce su R e su I.

**INI** : Incremento del registro I (INCRement I).

Aumenta di 1 il contenuto del registro I. Agisce su R e I.

**INY** : Incremento del registro Y (INCRement Y).

Aumenta di 1 il contenuto del registro Y. Agisce su R e I.

**JMP** : Salto incondizionato (JUMP)  $PC \leftarrow$  Indirizzo

Salta all'indirizzo specificato.

**JSR** : Chiama ad un sottoprogramma (Jump to Sub-Routine)  $PC \leftarrow$

$Ad$ .

Salva il PC (Program Counter) nella pila (indirizzo di ritorno) poi salta all'indirizzo specificato.

**LDA** : Carica l'accumulatore (Load Accumulator)  $A \leftarrow R$ .

Pone nell'accumulatore il contenuto della memoria specificata. Agisce su R e I.

**LRI** : Carica il registro I (Load I register)  $I \leftarrow R$ .

Pone nel registro I il contenuto della memoria specificata. Agisce su R e I.

**LDY** : Carica il registro Y (Load Y register)  $Y \leftarrow R$ .

Pone nel registro Y il contenuto della memoria specificata. Agisce su R e I.

**LSR** : Spostamento a destra (Logical Shift Right)

Scala di un posto a destra l'accumulatore

e la memoria specificata. Agisce su R, I e C.



**NOP** : Nessuna operazione (No Operation)  $PC \leftarrow PC + 1$ .

Istruzione nulla. L'esecuzione richiede 2 cicli macchina.

**ORA** : OR logico (OR Accumulator)  $A \leftarrow A \vee R$ .

Effettua l'OR logico tra l'accumulatore e la memoria specificata. Agisce su R e I.

**PMA** : Impila A (Push Accumulator)  $A \leftarrow (S) \leftarrow A$  ;  $S \leftarrow S - 1$ .

Pone l'accumulatore in cima alla pila e il pointer viene decrementato di uno.

**PSP** : Impila P (Push Processor status register).

$P \rightarrow (SP) \leftarrow P ; S \leftarrow S - 1$ .

Pone il registro di stato P in cima alla pila e decrements il puntatore di uno.

**PLA** : Disimpila verso A (Pull Accumulator)  $A \leftarrow (SP) ; S \leftarrow S + 1$  ;

$A \leftarrow (SP)$ .

Trasferisce verso l'accumulatore il contenuto della cima della pila e riaggiorna il puntatore. Ripristina S e Z.

**PLP** : Disimpila verso P (Pull P register)  $P \leftarrow (SP) ; S \leftarrow S + 1$  ;  $P \leftarrow (SP)$ .

Trasferisce verso il registro di stato P il contenuto della cima della pila e riaggiorna il puntatore. Ripristina tutti gli indicatori.

**ROL** : Rotazione a sinistra (Rotate Left).

Esegue la rotazione verso sinistra dell'accumulatore e di una memoria. Il vecchio valore del bit di riporto rientra a destra mentre il bit che esce alla sinistra diventa il nuovo valore di C. Ripristina N, Z e C.



**ROR** : Rotazione a destra (Rotate Right).

Esegue la rotazione verso destra dell'accumulatore e di una memoria. Il vecchio valore del bit di riporto rientra a sinistra mentre il bit che esce alla destra va a riempire C. Ripristina N, Z e C.



**RTI** : Ritorno dall'interruzione (Return from Interrupt).

$P \leftarrow PC$ .

Ritorno da una routine di interrupt: recupera sulla pila PC e P che sono stati salvati dal meccanismo d'interruzione.

**RTS** : Ritorno da un sottoprogramma (Return from Subroutine)  $PC \leftarrow$

Recupera sulla pila PC che era stato salvato dall'ultima JSR.

**SBC** : Sottrazione con riporto (Subtract with Carry).

$A \leftarrow A - M - C$ .

Si sottrae all'accumulatore la memoria specificata oltre al bit negativo di riporto. Opera in modo binario o decimale. Agisce su N, V, Z, C.

**SETC** : Pone a 1 il carry (Set Carry flag).

Forza a 1 il bit C.

**SEDC** : Pone in modo decimale (Set Decimal mode).

Forza a 1 il bit D.

## SET DI ISTRUZIONE DEL 6802

**SEI** : Inibisce le interruzioni (SEt Interrupt inhibit flag).  
Forza a 1 il bit I.

**STA** : Scrive l'accumulatore (Store Accumulator)  $M \leftarrow A$ .  
Trasferisce il contenuto dell'accumulatore nella memoria specificata.

**STX** : Scrive il registro X (Store X register)  $M \leftarrow X$ .  
Trasferisce il contenuto del registro X nella memoria specificata.

**STY** : Scrive il registro Y (Store Y register)  $M \leftarrow Y$ .  
Trasferisce il contenuto del registro Y nella memoria specificata.

**TAX** : Trasferimento di A in X.  $X \leftarrow A$ . Agisce su N e Z.

**TAY** : Trasferimento di A in Y.  $Y \leftarrow A$ . Agisce su N e Z.

**TSX** : Trasferimento di S in X.  $X \leftarrow S$ . Agisce su N e Z.

**TEA** : Trasferimento di A in S.  $A \leftarrow S$ . Agisce su N e Z.

**TSS** : Trasferimento di X in S.  $S \leftarrow X$ . Non agisce sugli indicatori.

**TEY** : Trasferimento di Y in A.  $A \leftarrow Y$ . Agisce su N e Z.

## Tabella di decodifica

Questa tabella rappresenta il contrario rispetto a quella che segue. In funzione del codice esadecimale AB, fornisce il codice mnemonico e il modo di indirizzamento corrispondente. Esempio: 8F → LDA IM (linea A, colonna 9). Nessun indirizzamento = indiretto e relativo.

AB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	BKX IMX	ORA IMX				ORA PZ,X	ASL PZ,X		RRF	ORA IMM	ASL A			ORA ABS	ASL ABS	
1	RPL IMX	ORA IMX				ORA PZ,X	ASL PZ,X		CLC	ORA ABS,Y				ORA ABS,X	ASL ABS,X	
2	JBR IMX	AND IMX			BIT PGE 2	AND PGE 2	ROL PGE 2		PLP	AND IMM	ROL A		BIT ABS	AND ABS	ROL ABS	
3	BRM IMX	AND IMX				AND PZ,X	ROL PZ,X		SEC	AND ABS,Y				AND ABS,X	ROL ABS,X	
4	RTI IMX	FOR IMX				FOR PGE 2	LSR PGE 2		PSA	FOR IMM	LSR A		JMP ABS	FOR ABS	LSR ABS	
5	BRV IMX	FOR IMX				FOR PZ,X	LSR PZ,X		CLU	FOR ABS,Y				FOR ABS,X	LSR ABS,X	
6	RTS IMX	ADC IMX				ADC PGE 2	ROR PGE 2		PLA	ADC IMM	ROR A		JMP IM	ADC ABS	ROR ABS	
7	BRZ IMX	ADC IMX				ADC PZ,X	ROR PZ,X		SEI	ADC ABS,Y				ADC ABS,X	ROR ABS,X	
8		STA IMX			STY PGE 2	STA PGE 2	STX PGE 2		DEY		TXA		STY ABS	STA ABS	STX ABS	
9	BCC IMX	STA IMX			STY PZ,X	STA PZ,X	STX PZ,X		TYA	STA ABS,Y	TXS			STA ABS,X		
A	LDY IMM	LSA IMM		LDX IMM		LDY PGE 2	LDX PGE 2		TAY	LDX IMM	TAX		LDY ABS	LDX ABS	LDX ABS	
B	BDS IMX	LSA IMX				LDY PZ,X	LDX PZ,X		CLV	LDX ABS,Y	TSX		LDY ABS,X	LDX ABS,X	LDX ABS,X	
C	CPY IMM	CMP IMX				CPY PGE 2	DEC PGE 2		INY	CMP IMM	DEX		CPY ABS	CMP ABS	DEC ABS	
D	BNE IMX	CMP IMX				CMP PZ,X	DEC PZ,X		CLD	CMP ABS,Y				CMP ABS,X	DEC ABS,X	
E	CPX IMM	SEC IMX			CPX PGE 2	SEC PGE 2	INC PGE 2		INX	SEC IMM	ROP		CPX ABS	SEC ABS	INC ABS	
F	BEG IMX	SEC IMX				SEC PZ,X	INC PZ,X		SED	SEC ABS,Y				SEC ABS,X	INC ABS,X	

## SET OF INSTANCES BEL 4303

Codici operativi / numero dei cicli macchina (ns) / numero dei byte  
processati (B):

CANDIDATE'S NAME		CANDIDATE'S PHOTO		CANDIDATE'S SIGNATURE		CANDIDATE'S ADDRESS		CANDIDATE'S CONTACT		CANDIDATE'S EDUCATION		CANDIDATE'S EXPERIENCE		CANDIDATE'S ACHIEVEMENTS		CANDIDATE'S RECOMMENDATIONS		CANDIDATE'S COMMENTS		
1	Mr. John Doe	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
2	Ms. Jane Smith	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
3	Mr. Robert Johnson	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
4	Ms. Emily Davis	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
5	Mr. Michael Brown	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
6	Ms. Sarah Wilson	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
7	Mr. David Miller	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
8	Ms. Lisa Anderson	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
9	Mr. James Taylor	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
10	Ms. Karen White	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
11	Mr. Christopher Lee	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
12	Ms. Michelle Hall	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
13	Mr. Daniel King	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
14	Ms. Rebecca Scott	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
15	Mr. Steven Adams	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St
16	Ms. Kimberly Baker	123 Main St	456 Elm St	789 Oak St	101 Pine St	202 Maple St	303 Birch St	404 Cedar St	505 Spruce St	606 Fir St	707 Willow St	808 Poplar St	909 Ash St	1010 Hickory St	1111 Walnut St	1212 Chestnut St	1313 Sycamore St	1414 Magnolia St	1515 Palm St	1616 Olive St

Codici operativi / numero dei cicli macchina (n) / numero dei byte  
accusati (B)

[illegible]

L I N G U A G I O  
M A C C H I N A





# I "COME...?"

I "COME...?"

## 1 - Posizionamento del cursore

Mantenere il cursore accosto alla domanda fino a quando la risposta non sia quella corretta.

La posizione del cursore è memorizzata prima della sua scomparsa e ripristinata per una eventuale nuova impostazione errata.

Se la risposta è errata, viene visualizzata.

Esempio

```
5 HOME
10 PRINT "RISPONDERE CON SI O NO ": GOSUB 100
20 INPUT ""
30 IF R$ = "S" THEN HTAB PH: VTAB PV: PRINT "SI"
35 IF R$ = "N" THEN HTAB PH: VTAB PV: PRINT "NO"
40 IF LEFT$(R$,1) < > "S" AND LEFT$(R$,1) < > "
   N" THEN HTAB PH: VTAB PV: CALL -RAB: GOTO 20
50 PRINT "FINE": PRINT : GOTO 10
```

```
100 PH = PEEK(36) + 1
200 PV = PEEK(37) + 1
210 RETURN
```

PH è la posizione orizzontale (colonna)

PV è la posizione verticale (riga)

CALL -RAB visualizza il resto della linea dopo il punto dove si trovava il cursore.

## 2 - Simulare INPUT

La stringa letta è iscritta da \$200 (\$12) a \$2FF (\$60)

APPLESOFT	Assemblatore
CALL-10964	JSR \$032C

Tutti i caratteri sono accettati fino al valore 325, ma CRL-1 annulla la linea e RETURN convalida l'input.

I  
C  
O  
M  
E  
.  
.  
.  
?  
.

1 "COME...?"

### 3 - Come impedire il list di un programma

POKE 2049,0 : POKE 2050,0

pone a zero il puntatore d'inizio relativo alla seconda linea di istruzioni.

Per ritrovare il suo esatto valore, cercare il primo byte 00 che indica la fine della prima linea di istruzioni e porre a 1 il suo valore.

Esempio:

LIST

```
10 REM COME N.3  
20 PRINT : END
```

POKE2049,0:POKE2050,0

LIST

CALL -151

\*800.81F

\*4 00



```
0800- 00 00 00 00 00 B2 20 43  
080B- 4F 4D 4D 45 4E 54 20 4E  
0810- 30 20 33 00 1C 08 14 00  
081B- BA 3A B0 00 00 00 0A 00
```

I comandi REM e PP annullano anche questo puntatore senza cancellare il programma.

### 4 - Caricare un programma dopo l'altro

- Caricare il primo programma in memoria RAM.
- Modificare il puntatore di inizio programma per il punto dopo il byte 00 dell'ultima linea di istruzione del primo programma.
- Caricare il programma che segue il memoria RAM.
- Modificare il puntatore di inizio programma per il punto di inizio relativo al primo programma.

TEXTAB puntatore di inizio programma 867,868

PRESENT puntatore di fine programma 86F,870

```

NEW

100 REM PG CODA

SAVE PCODA

REM PG TESTA          (a)

CALL -151

*AF:80

00AF- 15
00B0- 08
*800,815

0800- 00 12 08 0A 00 B2 20 50
0808- 47 20 44 45 20 54 45 54
0810- 45 00 00 64 0A
*67:12 08          (b)
*3D08

LOAD PCCODA          (c)
CALL -151

*67:01 08          (d)
*3D08

LIST

10 REM PG TESTA
100 REM PG CODA

```

### 3 - Come impedire l'accesso da tastiera

Oltre alla neutralizzazione del tasto **RESET** è necessario prevenire anche l'uso accidentale del tasto **CTRL-C** che provoca l'interruzione del programma in corso con il messaggio:

```
BREAK IN n. della linea di istruzione dove il programma è stato
fermato.
```

La soluzione proposta utilizza il trattamento d'errore: **CTRL-C** corrisponde al codice d'errore n. 235 e, quando è riscontrato, si annulla con l'istruzione **RESUME**:

```

1 ON ERR GOTO 1000
1000 IF PEEK(232) = 255 THEN RESUME

```

1 "COME...?"

#### 6 - Tutti i comandi sono interpretati come RUN

POKE 214,128

Un valore superiore o uguale a 128 nell'indirizzo 214 in ROM ha un effetto irreversibile su tutti i comandi o istruzioni BASIC, tutto è trasformato in RUN. Sono immuni solo i comandi d'accesso a programmi su disco.

Fare POKE per riinizializzare il sistema.

#### 7 - Inibizione del tasto RESET

L'effetto del RESET sul sistema dipende dal contenuto delle locazioni di memoria 3F2 e 3F3.

L'indirizzo contenuto in queste locazioni è quello verso il quale il sistema "salta" se viene premuto il tasto RESET.

Indirizzo		Valore di default		
Dec	Hex	Dec	Hex	
1010	3F2	191	BF	} ritorno al BASIC sotto 900 } arresto del programma
1011	3F3	187	9B	
1012	3F4	56	38	OR escl. di (1011) e (405

Il byte d'indirizzo 3F4 dev'essere ricalcolato per modificare l'effetto di RESET.

Il suo valore è ottenuto con CALL-1169 (4F86F) poi PRINT POKE (1012).

a) inibizione: il programma in corso non si arresta con RESET

\* 3F2 : 90 03 86

\* 300 : 20 EA 03     JBR 403EA (BOS)

\* 303 : 20 98 06     JBR 40898 (COMT)

\* 305 : 4C 02 07     JMF 40702 (NEXTT)

b) inibizione di fatto il sistema (dopo aver premuto RESET)

\* 300 : 4C 00 03

Il rilancio di tutto il sistema è possibile solo dopo l'interruzione.

c) il tasto RESET rilancia il sistema come quando lo si accende:

basta fare POKE 1012,0

d) disinibizione = 3F2 : 8F 9D 38

### 8 - Attesa di un carattere da tastiera

```
a) 10 X = PEEK(-16384) : IF X < 128 THEN 10
   20 POKE -16384,0 : XB = CHR$(X-128)
```

```
b) 10 WAIT -16384,128 : X = PEEK(-16384)-128 : POKE -16384,0
```

```
c) 10 GET XB
```

```
d) 10 CALL -754 (POKEY)
```

### 9 - Modificare la visualizzazione del list di un program BASIC

- POKE 13,13

La finestra di schermo è ridotta a 13 colonne di larghezza. Il comando LIST visualizza le linee di istruzione senza margine.

- TEXT annulla il comando precedente.

- Il carattere : consente di introdurre l'indentazione delle linee di istruzione.

- POKE 13,28

Facilita l'incolonnamento del ROM: la disposizione, alla memorizzazione, non è modificata dal comando LIST.

- TEXT oppure POKE 13,40 riportano al modo standard.

### 10 - GOTO calcolato utilizzando &

Si scrive l'espressione &.

Il sottoprogramma è memorizzato a partire dalla locazione #300. Dunque gli indirizzi #3F5, #3F6 e #3F7 devono essere preventivamente registrati con l'istruzione JMP #300 per rendere possibile il salto dell'interprete quando è incontrata l'espressione &.

```
& 3F5 : 4C 80 03
```

oppure:

```
POKE 1013,76 : POKE 1014,0 : POKE 1015,3
```

I "COME...?"

Sottoprogramma di valutazione dell'espressione e salto alla linea calcolata.

\* 300L

0300-	20 79 00	JGR	#0079
0303-	20 82 E7	JGR	#E782
0306-	20 1A D6	JGR	#D61A
0309-	90 03	BCC	#030E
030B-	4C 41 D9	JMP	#D941
030E-	A2 0A	LDX	#0A
0310-	4C 12 D4	JMP	#D412

#0079 FAMEVL      valutazione dell'espressione; il risultato è posto in FAC.  
#E782 SETADR      conversione di FAC in valore intero; il risultato è posto in #50, #51  
#D61A FNDLEN      cerca se la linea calcolata fa parte del programma  
#D941 #D70-      salta alla linea trovata  
#D412 ERRORR      eventuale errore con codice (0A = 90 UNDEF'D STATEMENT)

11 - Stampa con 9 decimali

a) DEF F9F(X) = INT(X\*10<sup>9</sup>)/10<sup>9</sup>

invece di stampare X, si stamperà F9F(X)

Nota: l'istruzione PRINT di un numero reale non visualizza gli zeri più a destra della parte frazionaria e neppure quelli più a sinistra della parte intera.

Se  $0,01 \leq |X| < 999\,999\,999,2$  il numero è in virgola fissa, altrimenti è sotto forma scientifica, esponente

nn.nnn nn nnEtt

n è il segno

E significa potenza di 10

nn e t sono cifre tra 0 e 9

12 arrotondare a 9 decimali

La funzione INT(X) dà come risultato il più piccolo intero inferiore a X, questo pone qualche problema se X è negativo.

Così ?INT(-0,3)

=-1

Quindi è necessario tener conto del segno di  $X$  nell'arrotondamento.

```
Caso 7INT(ABS(X)-0.5)/SGN(X)-0.5
```

```
DEF FN AR(X) = INT(ABS(X)+10^-D+.5/10^-D+SGN(X))
```

```
Fare PRINT FNAR(X)
```

c) Trascorre a  $D$  decimali con notazione flottante

```
10 XB = STR$(X)
20 FOR I = 1 TO LEN(XB): IF MID$(XB,I,1)<>"E" THEN NEXT I
30 FOR J = 1 TO I-1: IF MID$(XB,J,1)<>"." THEN NEXT J
40 IF J=0 <= I-1 THEN N=J+D:GOTO 60
50 N=I-1
60 PRINT LEFT$(XB,N)+MID$(XB,I)
```

12 - Incolloamento a destra in una zona di C caratteri

```
AB = STR$(FN AR(X))
CB = "STR$(FN AR(X)):" : REM C caratteri "spazio"
PRINT RIGHT$(CB+AB,C)
```

13 - Conoscere l'indirizzo di una variabile

Bisogna distinguere le variabili numeriche semplici dalle variabili alfanumeriche.

Per i due byte che rappresentano i primi due caratteri del nome, il sistema riserva 5 byte per conservare il valore reale o intero di una variabile numerica.

Esempio:  $A = 1$



$BI = 32767$



Il caso delle variabili alfanumeriche, dove il valore è una stringa di caratteri, è differente poiché nei 5 byte che seguono il nome della



1 "COME...?"

variabile, si trova la lunghezza della stringa e l'indirizzo di inizio della stringa.

Esempio: AA = "



L'indirizzo cercato è quello del valore di una variabile numerica dunque il puntatore del valore.

L'indirizzo di una stringa di caratteri è quello contenuto nei byte +1 e +2 in rapporto al puntatore del valore.

Questo puntatore del valore (VARPNT) è memorizzato in 803, 804 (131, 132) e contiene l'indirizzo dell'ultima variabile trattata dall'Applesoft.

e Applesoft si scrive:

```
10 X=A : REM      si ricerca dell'indirizzo A
20 A=PEEK(131) + 256*PEEK(132)
30 PRINT A
40 A=X : REM      si ristabilisce il valore di A
```

In linguaggio macchina, ci si può servire della routine PTRSET d'indirizzo 40FE3 per recuperare nell'accumulatore e nel registro Y, il byte meno significativo e quello più significativo relativo al puntatore della variabile della quale si cerca l'indirizzo.

Grazie all'operatore & seguito dal nome della variabile, si entra in un sottoprogramma in linguaggio macchina che rinvia agli indirizzi 778 (43040) e 779 (43041) il valore cercato.

\*300L

```
0300- 20 E3 DF      JSR    40FE3
0303- 80 04 03     STA    40304
0306- 8C 0B 03     STY    4030B
0309- 60           RTS
```

In una tabella di valori numerici interi, il valore occupa solo due byte per ciascuna variabile indice. Il puntatore del valore sarà

1 "COME...?"

utilizzato direttamente; l'indirizzo della variabile indice contiene il byte più significativo, seguito dal byte meno significativo.

```

1 POKE 1013,76: POKE 1014,0: POKE 1015,3
2 X = 0:L = 0:P = 0:AB = "APPLESOFT"
3 DIM A$(100):A$(1) = 32767
4  & A$(1)
5 X = PEEK (778) + PEEK (779) + 256
6 PRINT X
7 PRINT 256 + PEEK (X) + PEEK (X + 1)
8 AB = LEFT$ (AB,5)
9  & AB
10 X = PEEK (778) + PEEK (779) + 256
11 L = PEEK (X)
12 P = PEEK (X + 1) + PEEK (X + 2) + 256
13 PRINT L,P;" ";
14 FOR X = P TO P + L - 1
15 PRINT CHR$ ( PEEK (X));: NEXT X

```

Lunghezza  
puntatore

Stringa

```

RUN
2345
32767
5

```

38395 APPLE

#### 14 - Listato su stampante

Se la scheda di interfaccia della stampante è posta nel connettore A.1:

```

PR#1
LIST

```

Se le linee di istruzione superano i 30 caratteri, è necessario modificare il numero dei caratteri editati per linea sulla stampante per evitare il fastidioso classico "immagine di schermo".

```

PR#1
PRINT "[L-] 00 N" per 00 caratteri per linea.

```

Questa istruzione rende impossibile la visualizzazione sullo schermo. Per uscire da tale stato battere RESET oppure

```

PR#0

```

#### 15 - Cambiare la pagina di schermo

```

POKE -16399,0 visualizza la pagina n.1
POKE -16380,0 visualizza la pagina n.2

```

```

#0005
#0004

```

COME ?

1 "COME...?"

POKE -16364,0	visualizza in modo grafico	*C050
POKE -16363,0	visualizza in modo testo	*C051
POKE -16377,0	visualizza in modo HI-RES senza cancellare	*C057
POKE -16376,0	visualizza in modo GR senza cancellare	*C058
POKE -16362,0	grafica a tutto schermo	*C053
POKE -16361,0	grafica con 4 linee di testo	*C052

#### 14 - Modifica della finestra di schermo

TEXT regola la finestra per il suo massimo valore:

Larghezza	:	#21(33) :	WIDTH = #20(40)
Margine sinistro	:	#20(32) :	WLEFT = #0(0)
Margine superiore	:	#22(34) :	WSTOP = #0(0)
Margine inferiore	:	#23(35) :	WSTOP = #10(24)

POKE 33,larghezza	compreso tra 1 e 40
POKE 32,margine sinistro	margine sinistro + larghezza inferiore a 39
POKE 34,margine superiore	compreso tra 0 e 23
POKE 35,margine inferiore	più alto del margine superiore e inferiore a 24.

Il margine sinistro si riposiziona con un RETURN (PRINT).

#### 17 - Per far sì che PRINT stampi caratteri minuscoli

PRR (pone in linea la stampante)

```
POKE 243,32:PRINT"10 BUONGIORNO":NORMAL
10 buongiorno
```

L'indirizzo 243 o #F3 detto anche ORMASK serve nell'Applesoft per forzare il modo FLASH a gruppi di cifre (800-87F) per l'istruzione OR ORMASK.

con ORMASK = #40 in modo FLASH
= #0 in modo NORMAL e INVERSE

Le minuscole differiscono dalle maiuscole per il bit 5 del loro codice che vale 1 per le minuscole (800-87F) e 0 per le maiuscole (800-87F).

Per una operazione OR ORMASK con OR MASK = #20(32), il bit 5 è posto a 1 dalla POKE 243,32 per stampare caratteri minuscoli.

18 - Cancellazione dello schermo

HOME oppure CALL -934 o, ancora, FC38 B cancella tutto il testo dallo schermo e posiziona il cursore in alto a destra.  
 o ESC @

ESC E oppure CALL -848 o, ancora, FC9C B cancella dopo la posizione del cursore fino alla fine della linea corrente (o fino al bordo destro della finestra di schermo).

ESC F oppure CALL -938 o, ancora, FC42 B cancella dopo la posizione del cursore fino alla fine dello schermo (o della finestra).

19 - Scrivere sullo schermo dal basso verso l'alto

CALL -998 : CALL -998 inserito in ciascuna istruzione PRINT.

20 - Scrivere sullo schermo da destra a sinistra

CALL -1088 : CALL -1088 inserito in ciascuna istruzione PRINT \$.

21 - Spostare tutto il testo verso l'alto

CALL -912 oppure FC78 B

Questa operazione è realizzata automaticamente quando il cursore raggiunge la venticinquesima linea che apparirà alla ventiquattresima con lo scalare di una linea di tutto il testo verso l'alto.

22 - Proteggere un INPUT con un valore di default

Il valore di default ha lunghezza di un carattere

```
10 REM PROTEZIONE PER DEFAULT
20 DEF = "0": REM VALORE PER DEFAULT
30 PRINT "DOMANDA ? ";DEF;
40 PV = PEEK (36):PU = PEEK (37) + 1
41 IF PV > 23 THEN PV = 23: REM ATTENZIONE ALLO SCROLL
42 CALL - 1008: REM RECUPERO DI UNA POSIZIONE
50 INPUT "";REM
60 IF REM = "" THEN REM = DEF
70 HTAB PV: VTAB PU: PRINT REM
```

Quando l'INPUT è eseguita si vedrà il cursore lampeggiare sul valore di default. Se come risposta si batte RETURN il valore accettato sarà quello di default. Se si desidera impostare un altro valore, si batte tale valore su quello presentato, poi battere RETURN.

1 "COME...?"

## 23 - Prevedere la lunghezza di un programma

In generale un programma occupa tanti byte quanti sono i caratteri che lo compongono poiché esso è memorizzato tale quale, fatte salve le parole riservate che sono memorizzate con un codice di un byte.

Per ciò che concerne le variabili, ciascuna variabile numerica semplice reale o intera occupa 2 byte, ciascuna stringa occupa (7 + lunghezza) byte.

Una matrice occupa  $n \times (i + 2) \times 2$

$n$  è la dimensione della matrice (compreso lo 0)  
 $x$  = 2 (numero reale)  
 $x$  = 2 (intero)  
 $x$  = 2 (stringa di caratteri)  
 $d$  numero di dimensioni

Si guadagna spazio in memoria sopprimendo tutti gli spazi inutili, ponendo più istruzioni sulla stessa linea, evitare i REM, utilizzare variabili invece di costanti.

Utilizzare i GOSUB ai quali fare appello più volte in sequenza d'istruzioni simili.

## 24 - Fare musica con Apple

Un motivo è definito da un elenco relativo alla coppia I, J

I è l'altezza della nota e la sua frequenza

J è la durata di questa nota

Un programma, scritto in linguaggio macchina, consente di attivare l'altoparlante con LDA #C030 ad intervalli regolari:

- Il registro I è inizializzato con il valore 1, un 'bip', e diminuisce fino a 0, fino al successivo 'bip'; più I è basso, più la frequenza, dunque l'altezza, è elevata.
- Diminuisce anche il registro J e il suo passaggio a zero fa diminuire J, la durata, la quale, avvicinandosi allo zero provoca la fine dell'esecuzione di una nota.

RAMA : (con il programma seguente), I valori di I:  
255,242,230,214,204,192,182,172,162,152,144,136,128,  
120,121,110,100,102, 96, 91, 86, 83, 78, 72, 68, 64,  
64, 60, 57, 54, 50, 48, 45, 43, 40, 38, 36, 34, 32  
80L 80L LA LA SI SO SO SE SE SI FA FA 80L

0302-	AD 30 C0	LDA	#C030
0305-	80	DEY	
0306-	D0 05	BNE	#C040
0308-	CE 01 03	DEC	#C001
030B-	FD 09	BEQ	#C016

```

0300-   CA           DEL
0304-   B0 F5       BME   #0305
0310-   AE 00 03   LDI   #0306
0313-   4C 02 03   JMP   #0302
0316-   60         RTS

```

Per suonare un attivo, è necessario chiamare il sottoprogramma per ciascuna nota successiva.

In BASIC, i valori della coppia I, J sono letti da una serie di DATA fino al valore 0,0.

Il programma seguente è memorizzato all'inizio del programma con degli statement FOR E, V.

Esempio: due piccoli attivi "sintetici"

```

10 REM MUSICA
20 POKE 770,173: POKE 771,48: POKE
   772,192: POKE 773,136: POKE
   774,208: POKE 775,5: POKE 77
   6,206: POKE 777,1: POKE 778,
   3: POKE 779,240: POKE 780,9:
   POKE 781,202
30 POKE 782,208: POKE 783,245: POKE
   784,174: POKE 785,5: POKE 78
   6,3: POKE 787,76: POKE 788,2
   : POKE 789,3: POKE 790,96: POKE
   791,0: POKE 792,0
40 READ I,J: IF J = 0 THEN 70
50 POKE 768,I: POKE 769,J: CALL
   770
60 GOTO 40
70 IF F = 1 THEN END
80 F = 1: INPUT "ANCORA UNA VOLTA ? ";R#
90 GOTO 40
100 DATA 114,120,144,60,114,255
   ,1,120,128,120,144,60,128,12
   0,114,60,144,120,171,255,228
   ,255,0,0
200 DATA 0,160,128,255,152,40,1
   71,80,192,40,228,255,1,40,0,
   160,192,255,192,40,171,80,15
   2,40,128,255,0,0

```

I "COME...?"

### 25 - Disegnare, ingrandire far girare una figura

La codifica di una figura può essere semplificata utilizzando un byte per ciascun vettore elementare tracciato:

- dirigere verso l'alto → 4
- dirigere verso destra → 5
- dirigere verso il basso → 6
- dirigere verso sinistra → 7

I vettori successivi sono posti in una zona scelta dall'utilizzatore: si chiama tavola delle figure.

Questa tavola deve contenere:

- nel primo byte il numero della figura (esempio 1)
- nel terzo e nel quarto i dati per reperire l'inizio della prima figura (04 00 per una figura)
- poi i vettori della figura.

Bisogna precisare all'inizio del programma, l'indirizzo di inizio della tavola delle figure. Questo indirizzo è posto in #E0 a 232 e #E9 a 233 con la parte bassa dell'indirizzo in #E0 (senza significativi).

In BASIC, la successione dei vettori del disegno posti in DATA con 0 come fine dell'elenco, è letta e memorizzata a partire dal quinto byte della tavola (se è stata prevista una sola figura).

La figura scelta nell'esempio che segue è un petalo stilizzato.

Un fiore con un petalo come figura definita

## LIST

```

10 HGR
20 HCOLOR= 3
25 REM LA TAVOLA DELLE FIGURE E'
   ALL'INDIRIZZO #300 O 768
30 POKE 232,0: POKE 233,3
35 REM UNA SOLA FIGURA ... UN
   PETALO
40 POKE 768,1: POKE 769,0: POKE
   770,4: POKE 771,0
42 RESTORE :T = 0
43 READ D: POKE 772 + T,D: IF D =
   0 THEN 48
45 T = T + 1: GOTO 43
48 X = 140:Y = 80
50 SCALE= 3
52 REM ECCO IL FIORE
55 FOR R = 0 TO 64 STEP 4
58 ROT= R
60 DRAW 1 AT X,Y
70 NEXT R
80 END
100 DATA 4,4,4,5,4,4,4,5,4,4,4,
   5,4,4,4,5,4,5,4,5,4,5,5,
   5,4,5,5,5,4,5,5,5,4,5,5,5
200 DATA 6,6,6,7,6,6,6,7,6,6,6,
   7,6,6,6,7,6,7,6,7,6,7,7,
   7,6,7,7,7,6,7,7,7,6,7,7,7,0
300 REM LA DIREZIONE DI UN VETTORE
   ELEMENTARE VALE 4,5,6,7
   RISPETTIVAMENTE PER ALTO,
   DESTRA,BASSO,SINISTRA

```



## INDICE DEI 1 "COME...?"

ti numeri sono quelli di riferimento dei "trucchi")

- 1 - Posizionamento del cursore
- 2 - Simulare INPUT 24
- 3 - Come impedire il list di un programma
- 4 - Caricare un programma dopo l'altro
- 5 - Come impedire l'accesso alla tastiera
- 6 - Tutti i comandi sono interpretati come ROM
- 7 - Inibizione del tasto RESET
- 8 - Attesa di un carattere da tastiera
- 9 - Modificare la visualizzazione del list di un programma BASIC
- 10 - GOTO calcolato utilizzando 4
- 11 - Stampa con 2 decimali
- 12 - Incollamento a destra in una zona con 6 caratteri
- 13 - Conoscere l'indirizzo di una variabile
- 14 - Listare su stampante
- 15 - Cambiare la pagina di schermo
- 16 - Modifica della finestra di schermo
- 17 - Per far sì che PRINT stampi caratteri minuscoli
- 18 - Cancellazione dello schermo
- 19 - Scrivere sullo schermo dal basso verso l'alto
- 20 - Scrivere sullo schermo da destra a sinistra
- 21 - Spostare tutto il testo verso l'alto
- 22 - Proteggere un INPUT con un valore di default
- 23 - Prevedere la lunghezza di un programma
- 24 - Fare musica con Apple
- 25 - Disegnare, ingrandire far girare una figura

# INDIRIZZI

INDIRIZZI DEL MONITOR

Pagina zero

Dec	Hex	Nome	Funzione
0,1	00,01	LDC1,LDC2	Utilizzazioni diverse
2-31	02-1F	non utilizzati	disponibili
32,33, 34,35	20,21, 22,23	WSDL,W,T,B	i quattro parametri per la definizione della finestra
36,37	24,25	CH,CV	posizione orizzontale e verticale del cursore
38,39	26,27	GBASL,GBASH	contiene l'indirizzo di base della linea grafica calcolata da GBASCALC dopo ACC
40,43	28,29	BASDL,BASDH	indirizzo di base relativo alla linea modificata con le scroll in SCRLI
44,45	2C,2D	H2,V2	parametri di tracciamento orizzontale e verticale in grafica
46,48	2E,29	LENEM,EMMEM	codice macchina (3 caratteri in 2 byte) per il disassemblare
46	2E	MASK	OF (linee pari), FO (dispari) in colore BR
46	2E	CHRSUM	indice di errore di lettura su cassetta con READ
46	2E	FORMAT	per il disassemblaggio degli operandi
47	2F	LASTIN	indicatore di fine lettura su cassetta
47	2F	LENGTH	lunghezza di una istruzione (1, 2 o 3 byte)
48	30	COLOR	indicatore di colore per 2 linee adiacenti

# INDIRIZZI DEL MONITOR

Dec.	Hex	Nome	Funzione
49	31	MODE	conserva i caratteri dei comandi : , + - e zero
50	32	INWFLG	maschera di inversione dei caratteri da visualizzare
51	33	PROMPT	carattere 'A' che avverte la disponibilità dell'input da tastiera
52	34	YSAV	conserva il registro Y per la SETIOH
53	35	YSAVI	conserva il registro Y prima di CSUTZ
54,55	36,37	OSWL,OSWH	contiene l'indirizzo della routine di output che dev'essere usata dal sistema
56,57	38,39	ISWL,ISWH	contiene l'indirizzo della routine di input che dev'essere usata dal sistema
58,59	3A,3B	PCU,PCB	salvataggio del counter, ad esempio prima di un BREAK
60,61	3C,3D	A3L,A3H	memoria di lavoro per MOVE, VFI, READ, WRITE
62,63	3E,3F	A3L,A3H	
64,65	40,41	A3L,A3H	
66,67	42,43	A3L,A3H	
68,69	44,45	A3L,A3H	localizzazioni di salvataggio dei registri utilizzati da RESTORE e SAVE
69	46	ACC	
70	46	DNES	
71	47	FNES	
72	48	STATUS	
73	49	SPNT	
78,79	4E,4F	RNSL,RNSH	contatore di incremento durante KEYIN. Serve ad incrementare la funzione RND
74-77	4A-4D	non utilizzati	disponibili per altri sistemi
80-255	50-FF	non utilizzati	disponibili per altri sistemi

## Pagina uno

258-311	100-0FF	Stack	catena la cui cima è puntata dal registro 8
---------	---------	-------	---

## Pagina due

Dec	Hex	Nome	Funzione
512-758	200-3FA	IN	buffer di input riempito da SETLS con ADDRESS

## Pagina tre

1016-1018	3FA-3FA	USERADR	JMP all'indirizzo se CTRL-?
1019-1021	3FA-3FB	MMI	JMP all'indirizzo se interrupt non mascherato come <u>RESET</u>
1022,1023	3FB,3FB	IRQLGC	indirizzo se interruzione hardware

## Solo AUTOSTART

1008,1009	3FA,3FA	BRKV	indirizzo di ripresa se BRK e per default 39,FA (OLDBREAK)
1010,1011	3FA,3FA	SOFTCV	indirizzo di ripresa se <u>RESET</u> e per default 03,10 (ASCONF)
1012	3FA	PRESDUP MASK	byte che decide il boot a freddo o a caldo se presente <u>RESET</u>

# INDIRIZZI DI SISTEMA

Ad	Rev	Nome	Funzione
1024	400	LINE 1	inizio pagina TEXT o GR
1024-1063	400-427	linea 0 (testo)	linea 0 e 1 in GR
1064-1103	428-447	linea 8 (testo)	linea 16 e 17 in GR
1104-1143	450-477	linea 16 (testo)	linea 32 e 33 in GR
1144+8	478+8		memoria disponibile al programma per l'interfacciamento delle periferiche (1 byte per ogni connettore)
1192-1191	480-487	linea 1 TEXT	linea 2 e 3 in GR
1192-1231	488-507	linea 9 TEXT	linea 18 e 19 in GR
1232-1271	490-477	linea 17 TEXT	linea 34 e 35 in GR
1272+8	478+8		memoria RAM utilizzabile per le periferiche connesse in a da 4 a 7
1320-1319	500-527	linea 2 TEXT	linea 4 e 5 in GR
1320-1359	528-547	linea 10 TEXT	linea 20 e 21 in GR
1360-1399	550-577	linea 18 TEXT	linea 36 e 37 in GR
1400+8	578+8		memoria RAM utilizzabile per le periferiche connesse in a
1408-1447	580-5A7	linea -3 TEXT	linea 6 e 7 in GR
1448-1487	5A8-5CF	linea 11 TEXT	linea 12 e 13 in GR
1488-1527	5D0-5F7	linea 19 TEXT	linea 20 e 21 in GR
1528+8	578+8		periferiche connesse ad a
1536-1575	600-627	linea 4 TEXT	linea 8 e 9 in GR
1576-1615	628-647	linea 12 TEXT	linea 24 e 25 in GR
1616-1655	650-677	linea 20 TEXT	linea 40 e 41 in GR
1656+8	678+8		periferiche connesse ad a
1664-1703	680-6A7	linea 5 TEXT	linea 10 e 11 in GR
1704-1743	6A8-6CF	linea 13 TEXT	linea 26 e 27 in GR
1744-1783	6D0-6F7	linea 21 TEXT	linea 42 e 43 in GR

## Pagina quattro (sequito)

Dec	Hex	Nome	Funzione
1788+8	6FB+8		periferiche connesse ad 8
1792-1831	700-727	linea 6 TEXT	linee 12 e 13 in GR
1832-1871	728-74F	linea 14 TEXT	linee 28 e 29 in GR
1872-1911	750-777	linea 22 TEXT	linee 44 e 45 in GR
1912+8	778+8		periferiche connesse ad 8
1920-1959	780-7AF	linea 7 TEXT	linee 14 e 15 in GR
1960-1999	7B0-7CF	linea 15 TEXT	linee 30 e 31 in GR
2000-2039	7D0-7FF	linea 23 TEXT	linee 46 e 47 in GR
2040	7F0	SLOT 4	contiene ROM se in s vi è la
2040+8	7F8+8		periferica attualmente uti-
			lizzata. Memoria per la peri-
			ferica connessa in s

## Pagina dodici

Dec	Hex	Nome	Funzione
-14384	C000	EBD	indirizzo riservato ai carat-
			teri ricevuti da tastiera. Il
			bit 7 è posto a 1 se un tasto
			viene premuto
-14368	C010	EBOSTB	rimette a 0 il bit 7 di C000
			prima di un nuovo tasto (PEEK)
-14352	C020	TAPEOUT	uscita tutto-niente su cas-
			setta (PEEK)
-14336	C030	SPKR	attivatore tutto-niente del-
			l'altoparlante (PEEK)
-14320	C040	Utility Strobe	invia un impulso sul contatto
			3 del connettore della padio
			(PEEK)
-14304	C050	TITCLR	aziona grafico (cancellazione)
			(PEEK)
-14288	C051	TITSET	aziona testo (PEEK)
-14302	C052	RIECLR	aziona non aiuto (PEEK)
-14301	C053	RIESET	aziona aiuto (4 linee di testo)

# INDIRIZZI DI SISTEMA

Dec	Hex	Nome	Funzione
-16299	C054	LOWSCR	pagina 1 (no cancel.) (PQEE)
-16299	C055	HISCR	pagina 2 (no cancel.) (PQEE)
-16299	C056	LO-RES	grafica bassa risoluzione (PQEE)
-16299	C057	HI-RES	grafica alta risoluzione (PQEE)
-16299	C058	SETAM0	SET 0.5v (PQEE)
-16299	C059	CLRAM0	CLR 3.5v
-16299	C05A	SETAM1	AM0, AM1, AM2, AM3: 4 linee di input logici
-16299	C05C	SETAM2	
-16299	C05D	CLRAM2	
-16299	C05E	SETAM3	
-16299	C05F	CLRAM3	
-16299	C060	TAPIN	lettura del segnale su cassetta memorizzato nel bit 7 (PQEE)
-16299	C061	FB0	
-16299	C062	FB1	FB pulsante paddle, bit 7 a 1 se premuto (PEEX)
-16299	C063	FB2	
-16299	C064	PADDL0	
-16299	C065	PADDL1	PADDL paddle per i giochi (bit 7) verificata da FB0E0 (4FB0E)
-16299	C066	PADDL2	
-16299	C067	PADDL3	
-16299	C070	PTR10	inizializzazione del contatore per la lettura delle paddle (o degli ingressi analogici)

Pagina dodici (seguite)

Dec	Hex	Nome	Istruzioni	Effetto
-16256	C0B0	R	LDARCOB0	seleziona la scheda lin- guaggi solo in lettura (protetta in scrittura)
-16258	C0B1	W ENABLE ROM	LDARCOB1 LDARCOB1	disattiva la scheda lin- guaggi e disattiva la protezione in scrittura (solo scrittura)
-16254	C0B2	OFF ENABLE ROM	LDARCOB2	disattiva la scheda lin- guaggi in lettura (pre- tetta in scrittura)
-16253	C0B3	R/W	LDARCOB3 LDARCOB3	disattiva la scheda lin- guaggi in scrittura e la selezione in lettura (lettura/scrittura)
-16252, -16247	C0B4, C0B7	scheda linguag.		medesimo effetto di C0B0 - C0B3
-16246, -16245	C0B6, C0B5			medesimo effetto di C0B3 - C0B7
-16244, -16241	C0B8, C0BF	scheda linguag.		questo è il blocco n.1 che è selezionato tra B000 e BFFF quando C0B0, C0B7, si tratta del blocco n.2 ENABLE ROM: la ROM della scheda madre e della scheda linguaggi sono attivate in lettura



riepilogo della mappa di memoria tra D000 e FFFF

FFFF



scheda linguaggio  
16 Kbyte RAM

FF



scheda madre  
Integer BASIC

FF



scheda madre  
Applesoft BASIC

Pagina dodici (8C0-CFFF)

Dec	Hex	Nome	Funzione
-14240, -14220	C000, C0FF	DEV SELECT 1	periferica 1 selezionata con 14 indirizzi riservati per il dialogo
-14234,	C000-C00F	DEV SELECT 2	periferica 2
-14208,	C000-C00F	DEV SELECT 3	periferica 3
-14192,	C000-C00F	DEV SELECT 4	periferica 4
-14176,	C000-C00F	DEV SELECT 5	periferica 5
-14160,	C000-C00F	DEV SELECT 6	periferica 6
-14144	C000-C00F	DEV SELECT 7	periferica 7
	C400-C4FF	- indirizzi di sottoprogrammi in ROM posti in ciascuna scheda di interfaccia per la peri- ferica connessa allo slot 4 - ciascun sottoprogramma è limitato a 256 byte	
-14336, -12289	C800, CFFF	- estensione dello spazio di memoria ROM even- tualmente installata sulla scheda di inter- faccia si attiva DEV SELECT 6 si disattiva con CLRROM	
-12289	CFFF	CLRROM	disattiva l'eventuale espan- sione ROM 8C000 - 8CFFF

# INDIRIZZI MEMORIA ROM

Mini-Assemblatore - Sweet 16  
 ROM Integer + Monitor  
 Prog's Aid

Indirizzi		Funzione
Dec	Hex	
-2016	F500	} Mini-Assemblatore
-2430	F666	} Entrate
-2423	F689	} Sweet 16 interprete per programmare una pseudo macchina a 16 bit
-2054	F7FA	}
-12288	0000	} Prog's Aid (grafici ad alta risoluzione)
-11265	03FF	}
-11076	048C	Prog's Aid (correzione di programmi Integer)
-10955	0535	Prog's Aid (verifica la registr. su cassetta)
-10531	0600	Prog's Aid (Rinumerazione in Integer)
-10473	0717	Prog's Aid (musical)
-8192	E000	} Integer
-8189	E003	} boot "a freddo" (CTRL-B)
-2049	F3FF	} boot "a caldo" (CTRL-C)

Comandi		Indirizzi di sottoprogrammi	
Codifica	Buss	Buss	Indirizzo
8C	CTRL-C	BASCONI	FE83
82	CTRL-Y	USR	FE8A
8E	CTRL-E	REGI	FE8F
82	CTRL-Y	USR	FE8A CAUTOSTARTI
o 8B	T	o TRACE	FE82 MONITORI
8F	V	VFY	FE34
84	CTRL-X	IMPRT	FE8D
82	CTRL-Y	USR	FE83 CAUTOSTARTI
o 8C	S	o STEP2	FE84 MONITORI
89	CTRL-P	OUTPRT	FE97
8B	CTRL-B	KBASIC	FE8D
86	-	SETHODE	FE18
84	+	SETHODE	FE18
86	M	MOVE	FE2C
9B	C	LT	FE29
87	H	SETHDRM	FE84
82	I	SETINV	FE8D
85	L	LIST	FE3E
FO	M	WRITE	FECD
80	G	GO	FE84
EB	R	READ	FEFD
93	i	SETHODE	FE18
87	.	SETHODE	FE18

# INDIRIZZI DI MONITOR

Comandi		Indirizzi di sottoprogrammi	
Codifica	Nome	Nome	Indirizzo
C4	RETURN	CARON	FEF4
99	spazio	BLANK	FE04

La codifica dei caratteri di comando è come quella che appare nella tabella dei comandi: CMTBL 0FF0C-0FF02. La formula di derivazione del codice ASCII a questo codice consiste in 2 operazioni successive:

$ESR \leftarrow 80$  (vedere NOTCH in 0FFAD)  
 $ADC \leftarrow 80 \llcorner C = 1$

La tabella dei vettori relativi ai comandi (indirizzi dei sottoprogrammi SUBTBL: 0FF03-0FFF9) contiene solo la parte bassa degli indirizzi -1; la parte alta e costante è uguale a 0FE.

## Le funzioni caratteristiche del MONITOR

### I - Inizio

-155	FF65	MON	ingresso nel monitor con un 'bip'
-151	FF49	MON2	ingresso nel monitor senza un 'bip'

### I - Input dei dati (posti in 0200 su 1 caratteri)

-465	FD67	GETLINE	lettura di una linea di comando
-462	FD6A	GETLN	come GETLINE con * visualizzato
-451	FD75	WITCHAR	lettura del prossimo carattere trovato con '+'
-715	FD35	MOCHAR	lettura di un tasto con controllo di ESC
-756	FD0C	MOKEY	lettura di un tasto con cursore lasciagliante
-741	FD18	KEYIN	carattere nell'Accus. INC di 000L,R
-731	FD3F	ESC	gestione della posizione del cursore
-767	FD3D	NOTCH	controllo di CTR-L, 00 e 240 carat- teri massimi
-616	FD04	ADDIMP	aggiunta nel buffer d'ingresso 0200 fino a RETURN

3 - Analisi dei dati e interpretazione

-144	PF70	ANALISZ	Analisi del buffer B200 con MODE = 0
-141	PF73	ANITM	analisi della voce seguente
-87	PF67	SETNUM	recupero di un numero esadecimale
-83	PF60	BITCHN	invio di un carattere e decodifica
-138	PF68	DIG	ASCII verso ASCII, M di un numero
-134	PF7A	CHRSCH	ricerca del comando
-84	PF8E	TOSUB	chiamata del sottoprogramma corrispondente
-488	FE18	SETMODE	MODE="1" o "2" o "4" o "8"
-133	PF7C	LIMODE	passa a zero MODE

4 - Visualizzazione dei registri

→ -330	FE6A	PRBYTE	contenuto di A in 2 cifre esad.
-1724	FE44	PRNTX	contenuto di X
-1727	FE41	PRNTAX	contenuto di A,X in 4 cifre esad.
-1728	FE40	PRNTYX	contenuto di Y,X in 4 cifre esad.
-541	FE0E	PRNCE	passo inferiore di A in 1 cifra esad.
-321	FE8F	REGZ	tutti i registri
-1331	FE07	REGSP	sono visualizzati
-1318	FE88	REGSP1	A, X, Y, P, S

5 - Escita dei caratteri

-828	FE8E	CROUT	salto alla linea seguente
-931	FE0B	COUT	JMP (CSEL)
-938	FE0F	COUT1	visualizzazione di 1 carattere (ACC)
-932	FE06	COUT2	con salvataggio di A e Y
-1660	FE7B	VIDWAIT	sospensione della visualizzazione con CTRL-G

-1027	FEF0	VIDOUT	invia i tipi di caratteri visualizzati e posiziona il cursore
-1040	FEF0	STORABV	emissione di un 'bip'
-198	FE5A	BELL	ritarda un 'bip'
-1043	FE09	BELL1	attiva un 'bip'
-1052	FE04	BELL2	3 spazi
-1726	FE48	PEBLNK	visualizzazione in modo inverso
-384	FE80	SETINV	visualizzazione in modo normale
-388	FE84	SETNORM	

6 - Spostare del cursore

-1034	FEF4	ADVANCE	sposta di una posizione a destra
-924	FE42	CR	all'inizio della linea seguente
-932	FE46	LF	alla linea seguente
-1008	FE10	BS	una posizione a sinistra
-998	FE1A	UP	alla linea precedente
-990	FE22	VTAB	alla linea specificata in Acc CV
-1109	FE2B	TABV	alla linea specificata in Acc CV
-1087	FE01	BASCALC	calcola l'indirizzo di base BASL, BASR secondo A

# INDIRIZZI DI MONITOR











## 7 - Sezione del testo

-1225	FB39	SETTIT	modo testo
-1205	FB48	SETWAB	finestra di testo
-956	FC58	HOME	cancellazione nella finestra di testo
-958	FC42	CLREOP	cancellazione fino alla parte bassa dello schermo
-868	FC9C	CLREOL	cancellazione fino alla fine della linea
-912	FC78	SCROLL	scroll verso l'alto

## 8 - Grafici in bassa risoluzione

-1256	FB40	SETGR	modo GR misto. Cancellazione colore specificato in A COLOR
-1948	FB64	SETCOL	
-2048	FB60	PLST	} vedere dettagli nella pagina seguente
-2034	FB6E	PLSTI	
-2023	FB19	HLINE	
-2020	FB1C	HLINEL	
-2010	FB36	VLINEL	
-2008	FB28	VLINL	
-1998	FB32	CLSCB	
-1994	FB36	CLSTOP	
-1997	FB47	GBSCALE	calcola l'indirizzo base della linea grafica specificata in A, pone in A il colore del quadratino posto in Y, A
-1935	FB71	SCRN	

Comandi MONITOR

Chiamata da BASIC	Monitor	Azione	Risultato
CALL-2048	FB000	PLOT	 calcolo di GRASL, R
CALL-2034	FB000	PLOT1	 all'ordinata corrente
CALL-2023	FB190	HLINE	 A e Y sono modificati
CALL-2030	FB100	HLINE1	 alla linea corrente
CALL-2010	FB260	VLINE2	 (A)=1+(C) (X2C) A viene modificato
CALL-2008	FB280	VLINE	 (X2C)
CALL-1998	FB320	CLASCR	 cancellazione dello schermo grafico che viene posto in modo testo
CALL-1994	FB360	CLRTOP	 cancellazione dello schermo grafico. Le 4 linee in basso non vengono modificate
CALL-1992	FB380	CLRSC2	 Y cancellazione fino alla linea Y
CALL-1980	FB300	CLRSC3	 (X2C) cancellazione dello schermo nella parte in alto a sinistra



## INDIRIZZI DI MONITOR

### 9 - Input/Output

-1425	FA6F	INITAN	inizializzazione delle uscite logiche (IC000-IC005F)
-1250	FA1E	PREAD	lettura della paddle, n. in I, risultato in Y
-371	FE60	INPRT	<u>INPs</u> per altri input
-375	FE6F	SETINO	pone in linea la tastiera
-361	FE97	OUTPRT	<u>OUTPs</u> per altri output
-365	FE9C	SETVID	pone in linea il video (PS=0)
-307	FECD	WRITE	scrive su cassetta da 080 fino ad IAZ)
-259	FEFD	READ	lettura e memorizzazione tra IAI e IAZ)
-823	FE09	HEADR	scrittura su cassetta
	FA6A		verifica il controller del dischetto

### 10 - Disassemblaggio

-418	FE5E	L1ST	"L", disassembla 20 istruzioni
-413	FE63	L1ST2	disassemblaggio di IAI istruzioni
-1049	FE00	INSTOFP	una istruzione
-1918	FE02	INSDSI	il suo indirizzo
-1906	FE0C	INSDS2	il suo codice operativo
-1879	FE4F	SETTYP	ricerca il tipo di istruzione
-1769	FE53	PC60J	aggiorna il counter

### 11 - Visualizzazione delle locazioni di memoria

-589	FE05	IAM	visualizza da IAI) a IAZ)
-622	FE02	PRAI	visualizza l'indirizzo seguito da -
-586	FE06	SAIAOUT	visualizza i contenuti
-858	FE04	NTAI	incremento di IAI) fino a IAZ) (C=1)

### 12 - Spostamento e verifica del contenuto di memoria

-418	FE2C	MOVE	spostamento di IAI)-(IA2) verso IAA)
-844	FE04	NTAA	incremento di IAI),M
-438	FE36	WFT	verifica di IAI)-(IA2) con IAA)
-880	FE20	LT	trasferisce da A2 in A4 e A5

### 13 - Aritmetica esadecimale

-870	FE06	SARPM	addizione o sottrazione di IAI) e A21 con "+" o "-" nell'accumulatore. Il risultato viene visualizzato
------	------	-------	--

### 14 - Supervisore

-1472	FA40	IRB	interrupt hardware
-1447	FA39	GLDRBREA	"ERR" in linguaggio macchina
-1458	FA62	RESET	<u>RESET</u>
-1407	FA01	SEMON	invia il comando di inizializzazione
-1378	FAA6	PRULP	inizializzazione "a freddo"

-1347	FA89	SETPG3	sistemazione dei vettori 83F0 e 83F3
-1349	FA8F	SETPWRC	calcola il valore da porre in 83F4
-854	FE80	WAIT	temporizzazione
-134	FE80	IBASIC	inizializzazione del BASIC
-133	FE83	BASCONT	si procede in BASIC
-139	FE84	GO	'G'
-144	FEF4	CRMOD	solo se premuto <u>RETURN</u>
-110	FECA	USR	JMP 83F8 se premuto <u>CTRL-T</u>
-193	FF3F	RESTORE	ripristino di A, S, Y, P, S
-182	FF48	SAVE	salvataggio di A, X, Y, P, S

## Differenze

AUTOSTART		MONITOR	
		FA40-FA85	- disassembla le istruzioni
STEP	non esiste	FA85-FA86	in corso, nodo passo-passo
IRQ	FA40	FA86	
BREAK	FA4C	FA92	- considera una interruzione o un arresto
GLOBALBREAK	FA59	IBRE FA9C	- mostra il PB e i registri
RESET	FA62-FA83	non esiste	- auto-boot
APPLE II	FA60	non esiste	- mostra APPLE II all'accensione
SETPWRC	FA6F	non esiste	- memorizzazione di PAREDDP
VIGNAIT	FA78	non esiste	- interrompe e riprende la visualizzazione con <u>CTRL-S</u>
MOVEIT	FA94	non esiste	
ESCOLD	FA97	non esiste	- posizionamento del cursore con ESC I, J, K, M
ESCNEW	FA98	non esiste	
ESCNEW	FAA5	non esiste	
MULPM	non esiste	FB40-FB90	- moltiplicazione intera 16 bit
DIVPM	non esiste	FB91-FBC0	- divisione intera
TRACE	non esiste	FEC2	- modo TRACE (vedere STEP)
STEP1	FEC4	FEC4	- in AUTOSTART conduce a USR

## INDIRIZZI FONDAMENTALI SOFTEV e PWREUP

83F3, 83F1 e 83F4

La ROM AUTOSTART, il MONITOR in ROM disponibile sull'Apple II determinano il tipo di programma in caso di RESET.

Il RESET 'a caldo' è programmabile nel vettore SOFTEV e il MONITOR trattiene un byte particolare PWREUP come traccia del passaggio per un rilancio 'a freddo'. PWREUP dev'essere un esclusivo OR 845 e del contenuto di 83F3 se si desidera che RESET produca un rilancio 'a caldo'.

Valore di default:

	SOFTEV	PWREUP
senza DOS	8E003	845 (BASIC 'a caldo')
senza DOS	8908F	838

Se si inizializza il sistema 'a freddo', accendendolo, viene riconosciuta la presenza della scheda di interfaccia per la unità floppy disk, in questo caso il Monitor permette l'esecuzione del programma che carica il DOS (sistema operativo), (bootstrap) presente nella ROM della scheda di interfaccia.

Nell'insieme delle funzioni di 'nessa in marcia' del DOS, vi è l'assegnazione di un indirizzo per il rilancio 'a caldo' (in caso di RESET) che viene realizzato in SOFTEV (8908F) (parte bassa in test).

SOFTEV può essere modificato dall'utente che avrà cura di assegnare a PWREUP l'esclusivo OR di SOFTEV+1 e di 845 per far sì che RESET porti ad un programma specificato e non provochi l'equivalente di un rilancio 'a freddo'.

AMPREV

83F3, 83F4, 83F7

L'indirizzo AMPREV sarà utilizzato (indirettamente) per dare inizio all'esecuzione di un programma in linguaggio macchina da un programma in Applesoft che contiene la parola-chiave 'A'.

Esempio: 83F3 : 4C 60 03 JMP 8380

La parola-chiave 'A' porterà all'esecuzione del sottoprogramma che comincia all'indirizzo 84300.

DOSWARMSTART

8388 : 4C 8F 10

Questo indirizzo è utilizzato per tornare all'Applesoft (controllo da DOS) dopo l'utilizzo del Monitor, battere:

• 3888

]

Il programma corrente non è stato toccato.

1972 10-191

Indirizzo di ingresso al Monitor.  
Dall'Applesoft, si batte  
CALL -131  
e si viene visualizzato il

## References

1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 26

**CODE**      **OPER**      Visualizza il carattere presente nell'Accumulatore  
e sposta il cursore di una posizione.  
Considera RETURN, LF e i 2 modi Normal/Inverse.

```
OUTDB  DBASC  In AppleSoft, visualizza il carattere presente nell'
          accumulatore e tiene in considerazione i 3 modi
          di visualizzazione Normal/Reverse/Flash.
```

## Acquisizione di un carattere in un testo ASCII

```
CHARSET  #0001  Questo sottoprogramma (che si auto-modifica) punta
              in #00, #01 il carattere da prendere che sarà con-
              teneuto nell'Accumilatore Z=1 se fine dell'istruzione
              #00A o #001 C=0 se il carattere è una cifra
              (gli spazi del testo in BASIC sono saltati).
```

**CHRGST**   **NOGGT**   Il carattere è il carattere attuale e non il se-  
guente come in CHRGST.

## Lettera di un carattere battuto in battente

```

KEY      0F00C  Attende che un tasto sia premuto con il cursore
           lampeggiante. Il codice del carattere è caricato
           nell'Accumulatore.

```

**PUNTATORI FONDAMENTALI DELL'APPLESOFT**  
(indirizzi in ordine crescente)

Nome	Indirizzo		Funzione
	Hex	Dec	
TITLDB	867,868	103,104	Inizio del testo BASIC = 4001 (2049) per default
VARTAB	869,86A	105,106	Inizio delle variabili semplici, dei puntatori delle stringhe, dei puntatori delle funzioni.
ARFTAB	86B,86C	107,108	Inizio delle variabili dimensionate, puntatori alle matrici di di stringhe.
STREND	86D,86E	109,110	Inizio spazio libero
FRSTPR	86F,870	111,112	Fine spazio libero. Fine delle stringhe.
MEMB11	873,874	113,114	Inizio stringhe. Fine spazio in memoria «1. Le stringhe sono registrate dall'alto verso il basso.
CURLIN	875,876	115,116	Numero della linea in corso di esecuzione.
OLSLIN	877,878	119,120	N. di linea interrotta da CTRL-C , STOP o END.
OLSTXT	879,7A	121,122	Indirizzo dell'ultimo byte (00) della linea in corso di esecuzione.
DATLIN	87B,87C	123,124	N. della linea nella quale sono letti i DATA.
DATPTR	87D,87E	125,126	Indirizzo del primo byte relativo ai DATA da leggere.
INPPTR	87F,880	127,128	Puntatore del buffer d'ingresso da tastiera durante INPUT.
VARNAM	881,882	129,130	Contiene il nome (2 caratteri) dell'ultima variabile a cui si è fatto riferimento.

PUNTATORI FONDAMENTALI DELL'APPLESOFT  
(indirizzi in ordine crescente)

Nome	Indirizzo		Funzione
	Hex	Dec	
VARPNT	000,000	171,172	Indirizzo del valore dell'ultima variabile a cui si è fatto riferimento, e del byte relativo alla lunghezza di una stringa.
PBEND	00F,000	175,176	Fine del testo BASIC.

Riassunto		
HIGH:	stringhe	MEM012 PBETOP
	libera	STREND
	matrici	ARYTAB
	variabili	VARTAB o PBEND
LOWR: 0001	testo	TITTAB

Implementazione di un programma e delle variabili in RAM

<pre> * LIST 10 AA = 2 20 AAX = 4 30 AAB = "" 40 DIM AA(1,3) 50 DIM AA(2,1) 60 DIM AAB(3,3) 70 DEF FN AA(X) = X - 256 * INT   (X / 256) 80 PRINT FN AA(257)  RUN 1 </pre>	<p>AA variabile reale semplice AAX variabile semplice intera AAB variabile semplice stringa di caratteri AA(1,3) variabile reale dimensionata AA(2,1) variabile intera dimensionata AAB variabile stringa dimensionata FN AA(X) funzione definita dal programma</p>
<p>CALL=151</p>	
<p>*69.6A</p>	
<p>0069- 74 00 *800.873</p>	<p>Testo codificato in BASIC</p>
<pre> 0800- 00 0A 00 0A 41 41 00 0808- 32 00 14 00 14 00 41 41 0810- 25 00 34 00 1F 00 1E 00 0818- 41 41 34 00 32 32 00 00 0820- 00 28 00 8A 41 41 28 31 0828- 0C 33 29 30 1A 00 32 00 0830- 8A 41 41 28 28 32 2C 31 0838- 29 00 48 00 2C 00 8A 41 0840- 41 24 28 31 2C 32 29 00 0848- 43 00 4A 00 80 C2 41 41 0850- 28 5A 29 30 58 C9 32 33 0858- 2A 5A 63 38 58 C8 32 33 0860- 3A 22 00 71 00 50 00 8A 0868- C2 41 41 28 32 33 37 29 0870- 00 00 00 0A </pre>	<p>( ) indirizzi di inizio della linea di istruzione seguente</p> <p>[ ] numero della linea di istruzione</p> <p>( parola riservata</p> <p>00 fine linea</p> <p>00 00 fine del programma</p> <p>---- definizione di funzione</p>

**Variabili semplici****#6B.6C**2 caratteri del nome 'j'  
valore '.'

006B- 97 00

#874.89%

0874- .41 41102 00

0878- 00 00 00.C1 C1;00 04 00

0880- 00 00.41 C1;00,10 08 00

0888- 00.C1 41104 08.92 08.08.

0890- .08 00;00 00 00 00 00.

*Codifica dei due caratteri del nome*

		1° car.	2° car.	
AA	reale	P	P	P è il codice
AA1	intera	N	N	ASCII con bit 7=0
AA5	stringa	P	N	N è il codice
AA10	funzione	N	P	ASCII con bit 7=1

*Valore di una variabile su 5 byte*

	1°byte	2°byte	3°byte	4°byte	5°byte
Reale	esponente	mantissa			
Intera	parte bassa	parte alta	non utilizzati		
Stringa	lunghezza	indirizzo di inizio stringa		non utilizzati	
Funzione	indirizzo della definizione		indirizzo dell'argomento		codice P del 1° ca- rattere dopo "

Gli indirizzi sono espressi con la parte bassa sul primo byte e la parte alta sul secondo byte.



# Implementazione di variabili dimensionate

\*6D.6E

006D- 0A 07

\*097.909

0097-.41

BIN AA(1,1)

0098- 41;01 00 00 00 00 00 00 00

00A0- 00 00 00 00 00 00 00 00

00A8- 00 00 00 00 00 00 00 00

00B0- 00 00 00 00 00 00 00 00

00B8- 00 00 00 00 00 00 00 00

00C0- 00 00 00 00 00 00 00 00

00C8- 41;01 00 00 00 00 00 00 00

BIN AA(2,1)

00D0- 00 00 00 00 00 00 00 00

00D8- 00 00 00 00 00 00 00 00

BIN AA(3,1)

00E0- 00 00 00 00 00 00 00 00

00E8- 00 00 00 00 00 00 00 00

00F0- 00 00 00 00 00 00 00 00

00F8- 00 00 00 00 00 00 00 00

0900- 00 00 00 00 00 00 00 00

0908- 00 00

2 caratteri per il nome ; offset / variabile seguente

numero delle dimensioni e degli indici

numero massimo degli elementi della matrice per ciascuna dimensione dall'ultima alla prima (valore massimo dell'indice + 1).

// valore di ciascun elemento della matrice

- reale 3 byte

- intera 2 byte soltanto

- stringa 3 bytes lunghezza, indirizzo

nell'ordine AA(0,0), AA(1,0), AA(0,1), AA(1,1), AA(0,2), AA(1,2), AA(0,3), AA(1,3) l'indice più a destra aumenta più lentamente.

Esempio di implementazione di un programma e delle variabili in memoria RAM

LIST

```
10 REM APPLESOFT
20 INPUT "HOME ?";N#
30 FOR K = 1 TO LEN (N#) - 2
40 PRINT RIGHT# (N#, LEN (N#) -
    K + 1);" "; MID# (N#,K,2);"
    ";
50 NEXT K
```

CALL-151

\*69.6A

VARIAB

0069- 45 00

\*67.6B

0067- 01

0068- 00

TITAB

\*800.862

0800- 00 01 00 0A 00 02 20 41

0808- 50 50 4C 45 53 4F 46 5A

0810- 00 22 00 14 00 04 22 4C

0818- 4F 4D 45 20 3F 22 38 4E

0820- 24 00 30 00 10 00 01 40

0828- 50 31 C1 E3 28 46 24 2F

0830- C9 32 00 50 00 20 00 0A

0838- 59 28 4E 24 2C E3 28 4E

0840- 24 2F C7 40 C8 31 2F 38

0848- 22 20 22 38 0A 28 4E 24

0850- 2C 48 2C 32 2F 38 22 20

0858- 22 38 00 A0 00 32 00 8C

0860- 48 00 00

Testa BASIC codificato

( parole riservate

00 fine della linea

00 00 fine del programma

1 1 n° di linea

0) indirizzo della  
linea seguente

# APPLESOFT - ESEMPIO N.2

Esempio di implementazione di un programma e delle variabili in memoria RAM  
(segue)

```

RUN
NAME ?SERGIO                                esecuzione delle
SERGIO SE ERGIO ER RGIO RG BIO GI          istruzioni

CALL-151

*6D,4E                                     STREAM
006D- 73 0B

*8B3,870                                    variabili
0B63- 00 0A,4E,80 0A/                      -NB: lunghezza,
0B68- FA 9D,00 00,4B 00 83 20              puntatore
0B70- 00/                                  -K : valore reale

*6F,70                                     FREEST
006F- FA
0070- 95

*95FA,95FF
95FA- 53 45 52 47 49 4F                   Valore di NB

*79,7A                                    indirizzo dell'ultima
0079- 61 0B                               byte dell'ultima in-
                                          truzione eseguita

*AF,80
00AF- 65                                    fine del programma (*2)
00B0- 0B

```

Le parole riservate in ordine crescente per codice e gli indirizzi dei sottoprogrammi corrispondenti nell'interprete.

Codici da 800 a 849

Parola riservata	Codice decimale	Indirizzo decimale
END	800 120	80670 55408
FOR	801 129	80766 55142
NEXT	802 130	80CF9 54569
DATA	803 131	80FF0 53781
INPUT	804 132	80882 54343
DEL	805 133	8F331 62287
DIM	806 134	8F809 57505
READ	807 135	80882 56290
ER	808 136	8F390 62352
TEXT	809 137	8F399 62361
PR=	80A 138	8F1C5 61925
IN=	80B 139	8F1DE 61918
CALL	80C 140	8F185 61989
PLAT	80D 141	8F225 61989
KL IN	80E 142	8F232 62002
VL IN	80F 143	8F241 62017
HGR2	890 144	8F388 62424
HGR	891 145	8F3C2 62434
HCOLOR=	892 146	8F669 63289
HPLOT	893 147	8F6FE 63230
DRAW	894 148	8F769 63337
EDRAW	895 149	8F76F 63343
RTAB	896 150	8F7C7 63463
HORE	897 151	8FC58 64600
ROT=	898 152	8F721 63265
SCALE=	899 153	8F727 63271
SHLWD	89A 154	8F775 63349
TRACE	89B 155	8F360 62061
HSTRACE	89C 156	8F26F 62063
NORMAL	89D 157	8F273 62067
INVERSE	89E 158	8F277 62071
FLASH	89F 159	8F280 62080
COLOR=	8A0 160	8F24F 62031
POP	8A1 161	80960 55659
VTAB	8A2 162	8F286 62038
HTRCH	8A3 163	8F286 62066
LONGH	8A4 164	8F286 62118
ONERR	8A5 165	8F2C8 62155
RESUME	8A6 166	8F318 62232
RECALL	8A7 167	8F38C 62396
STORE	8A8 168	8F39F 62367
SPEED=	8A9 169	8F262 62050

# APPLESOFT

Le parole riservate in ordine crescente per codice e gli indirizzi dei sottoprogrammi corrispondenti nell'interprete.

Codici da 000 a 099

Parola riservata	Codice decimale	Indirizzo decimale
LET	000 170	00440 33070
GETO	000 171	00903 33010
SUM	000 172	00912 33070
IF	000 173	00909 33753
RESTORE	000 174	00049 33369
&	000 175	003F5 1013
MODUS	000 176	00921 33505
RETURN	001 177	00910 33659
REM	002 178	00900 33772
STOP	003 179	00040 33406
ON	004 180	00900 33700
WAIT	005 181	00704 39260
LOAD	006 182	00009 33497
SAVE	007 183	00000 33472
DEF	008 184	00713 30131
POKE	009 185	00770 39259
PRINT	000 186	00005 34031
CONT	000 187	00096 33446
LIST	000 188	00445 34949
CLEAR	000 189	00666 34090
SET	000 190	00040 34224
NEW	000 191	00049 34057
TAB(	000 192	
TO	001 193	
FN	002 194	
SPEC	003 195	
THEN	004 196	
AT	005 197	
NOT	006 198	00090 -0552
STEP	007 199	
*	000 200	00700 -4210
-	009 201	00707 -4233
*	000 202	00970 -5761
/	000 203	00666 -0530
^	000 204	00709 -4340
AND	000 205	00733 -0363
OR	000 206	00747 -0369
>	009 207	
*	000 208	00760 -0382
<	001 209	

Le parole riservate in ordine crescente per codice e gli indirizzi del sottoprogrammi corrispondenti nell'interprete.

Codici da 802 a 85A

Parola riservata	Codice decimale	Indirizzo decimale
SEN	802 210	8000 40304
INT	803 211	8001 40401
ABS	804 212	8002 40328
USR	805 213	8003 10
FRE	806 214	8004 50078
SCRNI	807 215	8012 50290
PEL	808 216	80F0 57393
POS	809 217	802FF 58111
SQR	80A 218	80000 61069
RND	80B 219	80FAE 61338
LOG	80C 220	80941 59713
EXP	80D 221	80FA9 61193
COS	80E 222	80FEA 61918
SIN	80F 223	80FF1 61425
TAN	809 224	8083A 61990
ATH	801 225	8009C 61398
FREE	802 226	80764 59336
LEN	803 227	80686 59094
STR	804 228	803C5 58369
VAL	805 229	80787 59143
ASC	806 230	806E3 59109
CHR	807 231	80646 58950
LEFT	808 232	8065A 58970
RIGHT	809 233	80686 59014
MID	80A 234	80691 59025

# INDIRIZZI DELL'INTERPRETE APPLESOFT

Classificati per grandi funzioni

## 1 - Entry point (punti di entrata)

-8192	E000	BASIC	inizializzazione 'a freddo' su <u>CIRI-B</u>
-3899	F120	COLDSTART	inizializzazione 'a freddo' su <u>CIRI-B</u>
-8189	E003	BASIC2	2° ingresso detto 'a caldo' su <u>CIRI-C</u> o <u>RESET</u>
-11204	D43C	CMDSLOOP	lancio del loop principale dell'interprete (a caldo)

## 2 - Input dei dati (nel buffer #2040)

-11201	D43F	input con la visualizzazione di 3	
-10942	D53E	INLIN+3	input da tastiera di una istruzione con il come 'prompt'
-10944	D53C	INLIN	input senza 'prompt'
-10951	D539	SOBRFS	pone a zero il bit 7 di tutti i caratteri registrati da INLIN
-10925	D553	INCHR	input di un carattere nell'accumulatore e mette a zero del bit 7

## 3 - Analisi dei dati (nella zona riservata al programma)

177	B1	CHRSET	caricamento nell'accumulatore del carattere puntato da #000, #009 o TITPTR (seguente o attuale) e discriminazione del tipo C=0 per le cifre [0] per la fine di una linea o di una istruzione
183	B7	CHRSET	
-9716	B40C	LINSET	caricamento in #00, #01 o LINNUM del numero della linea puntata da #00, #09 o TITPTR e prosecuzione dell'analisi
-4411	E4F5	GETBYTC	acquisizione di un carattere con CHRSET e valutazione a partire da TITPTR per la valutazione dell'espressione puntata da TITPTR e trasferimento del risultato in in FAC per FAC+intero(255 in 1 e FACLO
-4408	E4F8	GETBYT	
-4405	E4F6	CONINT	
-10726	D614	PNOLIN	ricerca nel programma l'indirizzo dell'istruzione il cui numero è in LINNUM (#00, #01). Se C=1, il risultato in LOWTR (#00, #9C) altrimenti LOWTR punta all'istruzione di numero più elevato
-6334	E74C	COMBYTE	verifica che TITPTR punti su una virgola e continua l'analisi con GETBYT
-6330	E746	SETNUM	acquisizione di un numero per la valutazione e test della virgola sul carattere seguente

(segue)

-8057	8047	FNUM	valuta una espressione puntata da TPTB e pone il risultato in FAC assicurandosi che sia un numero
-6318	8752	GETADR	FAC ← intero (2 byte) in 450, 451
-2087	8799	GETARPT	ricerca di una variabile con il nome puntata da TPTB e ...
-8121	8F81	PTGET	... il risultato in 483, 484 o 5ARPT e ? , 4 - se non esiste già creazione - il posto del nome nella tavola delle variabili è in 478, 479 e LOWR
-8047	837D	ISLET	l'Accumulatore contiene un codice ASCII di una lettera, se C = 1
-8524	8E82	PARDR	verifica delle parentesi
-8520	8E88	CHODL	TPTB punta su 1?
-8517	8E88	CHODPH	TPTB punta su 1?
-8514	8E88	CHODCH	TPTB punta su ,?
-8512	8E8C	SYNCR	altrimenti, errore di sintassi se sì l'analisi continua
	8C4A	FIN	registra il numero flottante puntato da TPTB in FAC

## 4 - Visualizzazione dei dati

-4818	8D2E	PRNTFAC	visualizza FAC (490-491) e lo distrugge
-9414	8D34	STROUT	visualizza la stringa puntata da (?, 4)
-9411	8D38	STPRNT	visualizza la stringa puntata da OFACR0, FACLR0
-9385	8D57	OUTSP	visualizza uno spazio
-9477	84F8	CR80	ritorno carrello
-9382	8D54	OUTGT	?
-9388	8D5C	OUTD	visualizza l'accum. con i modi I, F, N
-4817	8D17	SPRT	visualizza "IN" n° di linea corrente
-4828	8D24	LINPR	visualizza un intero in I, 4
-9315	84D5	PRINT	istruzione di stampa su video

## 5 - Aritmetica e funzioni algebriche

CONSTANT: NUMERICHE

80FE	90	80	80	80	38	-2 <sup>14</sup> = -32767.0005
8064	98	3E	8C	1F	FD	99 999 999,9
80F0	9E	4E	48	27	FD	999 999 999
8014	9E	4E	48	28	00	1 000 000 000 = 10 <sup>9</sup>



# INDIRIZZI DELL'INTERPRETE APPLESOFT

(segue)

F066	81	49	0F	8A	A2	PI/2 = 1.57079633
F068	83	49	0F	8A	02	2*PI = 6.28318531
F070	7F	00	00	00	00	1/4
EE64	00	00	00	00	00	1/2
E913	81	00	00	00	00	1
E928	80	35	04	F3	34	SQR(4/3) = 0.767146781
E937	81	35	04	F3	34	SQR(3) = 1.41421356
E937	80	80	00	00	00	-1/2
E93C	80	31	72	17	FB	LOG(3) = .493147181
E939	84	20	00	00	00	10
E939	81	38	AA	38	26	LOG(10)/LOG(2) = 1.44269504

## FUNCTION

E780	FABS	OFAC) ← OFAC) + 1/2
E787	FSUB	ARG ← (Y, A) e chiama FSUBT
E78A	FSUBT	FAC ← ARG-FAC
E78E	FADD	ARG ← (Y, A) e chiama FADDT
E7C1	FADDT	FAC ← FAC + ARG
E941	LOG	FAC ← LN(FAC)
E97F	FMULT	ARG ← (Y, A) e chiama FMULTT
E982	FMULTT	ARG ← FAC * ARG
E9E3	CONVPC	ARG ← (Y, A)
EAD9	SCALE	FAC ← FAC * 10
EAD5	DIV10	FAC ← FAC / 10
E866	FDIV	ARG ← (Y, A) e chiama FDIVT
E869	FDIVT	FAC ← ARG / FAC
E880	SGN	FAC ← segno di FAC
E882	SIGN	A ← segni di FAC (1 se > 0, 0 se 0, FF se < 0)
E893	FLOAT	FAC ← A diventa flottante
E8AF	ABS	valore assoluto FAC ← FAC
EE25	INT	il più grande valore intero inferiore a FAC ← FAC
EBF2	SINT	il più grande valore intero inferiore se FAC < 32767
E10C	AYINT	il più grande valore intero inferiore nella mantissa FAC

## 4 - Funzioni sulle stringhe di caratteri

DEC HEX

-8837	887B	FRMEXL	valutazione di una espressione partendo da TITPR
-8875	8881	STRIT	TITPR → Y,X, segue chiamata a STRLIT
-7195	E3E7	STRLIT	mette un carattere alla fine della stringa in ENDCR

-7187	E3ED	STRLT2	costruisce un descrittore di stringa in DSCTMP, FACMD, LD e conduce a PUTMEM
-7126	E424	PUTMEM	pone DSCTMP in un descrittore temporaneo puntato da FACMD, LD
-7205	E3ED	STRSPA	conduce a GETSPA e pone il puntatore e la lunghezza in DSCTMP
-7084	E45C	GETSPA	libera dello spazio per una stringa spor- tando verso il basso FRESPC a FRETOP - può mettere "OUT OF MEMORY" - aggiorna DSCTMP
-6761	E397	CAI	concatenazione della stringa descritta da (FACMD, LD) e quella puntata da TITPTR +1
-6684	E3E2	MOVSTR	sposta la stringa puntata da Y, X con lun- ghezza A nella posizione puntata da FRESPC (471, 472)
-6760	E3B4	MOVINS	sposta la stringa il cui descrittore è pun- tato da STRNFI verso FRESPC
-6659	E3FD	FRESTR	verifica che FAC indirizzi una stringa e conduce a FRMFAC
-6654	E400	FRMFAC	libera lo spazio occupato per una stringa temporanea
-6403	E435	FRETRM	libera il descrittore temporaneo senza li- berare la stringa
-7054	E484	BARBAR	recupera lo spazio occupato dalle stringhe abbandonate spostando verso l'alto le altre

## VARIABILI UTILIZZATE PAGINA ZERO

DEC	HEX		
13	0	CHMAC = "	
14	6	END CHR = 60	
17	11	VALTYP = 1	se stringa in FAC
82	52	TEMPPT	puntatore temporaneo
83	53	LASTPT	puntatore temporaneo
	5E,5F	INDEX	puntatore temporaneo

# INDIRIZZI DELL'INTERPRETE APPLESOFT

DEC	HEX		
111,112	6F,70	FRETOP	zona bassa stringhe
113,114	71,72	FRESPC	fine della zona libera
133,134	85,86	FORPNT	utilizzata da COPY per liberare spazio
148,149	94,95	HIGHDS	utilizzato per BLTU
150,151	96,97	HIGHTR	per l'indirizzo di destinazione
155,156	9B,9C	LOWTR	utilizzato per BLTU
157,158,159	9D,9E,9F	DOCTMP	descrittore di stringa
160,161	99,9A	FACDS,LS	puntatore di descrittore
171,172	AB,AC	STRNG1	puntatore utilizzato da MOVING
173,174	AD,AE	STRNG2	puntatore utilizzato da STRL12

## 7 - Funzioni grafiche alta risoluzione

180181223

pagina 1	: #2000 - #3FFF	
pagina 2	: #4000 - #5FFF	
linea 0	: #2000 - #2027	140 byte / linea
linea 64	: #2028 - #204F	
linea 128	: #2050 - #2077	
linea 8	: #2080 - #20AF	
linea 72	: #20B0 - #20CF	
linea 136	: #20D0 - #20FF	
linee 14,80,144,24,88,152	: #2100 - #21F7	
linee 32,96,160,40,104,168	: #2200 - #22F7	
linee 48,112,176,56,120,184	: #2300 - #23F7	

Sia n il numero della linea e Na il suo indirizzo la linea precedentemente listata quindi:

la n+1 linea Na per indirizzo Na + #400
la n+2 linea Na per indirizzo Na + #800
la n+3 linea Na per indirizzo Na + #C00
la n+4 linea Na per indirizzo Na + #1000
la n+5 linea Na per indirizzo Na + #1400
la n+6 linea Na per indirizzo Na + #1800
la n+7 linea Na per indirizzo Na + #1C00

# INDIRIZZI DELL'INTERPRETE APPLISOFT

## VARIABILI DELLA PAGINA ZERO

DEC	HEX	NOME	FUNZIONE
26,27	#1A,#1B	SHAPE L,H	puntatore nella tabella delle figure
28	#1C	NCOLORI	dipende dalla parità dell'ascissa I di SHAPE e NCOLORO
29	#1D	COUNT H	contatore nella traccia di linee
30,31	#26,#27	HBASL,H	indirizzo d'indice di una linea
48	#30	BRAGE	#01, #02, #04, #08 #16, #1, #2, #3 #40, #80, #C0 #4, #5, #6
82	#52	DY	incremento di y per ALINE
83	#53	DBENT	angolo di rotazione per BRAG

## VARIABILI DELLA PAGINA ZERO UTILIZZATE PER LE FUNZIONI GRAFICHE

DEC	HEX	NOME	FUNZIONE
224,225 226	#E0,#E1 #E2	xH,xL y	coordinate dei punti tracciati da PLOT
228	#E4	NCOLORO	#00, #A, #5, #F nero, azz, blu, bianco #00, #A, #5, #F nero, verde, rosso, bianco
229	#E5	IS7	n° di byte in una linea per il punto di ascissa x
230	#E6	HPAB	#200: pagina 1 #400: pagina 2
231	#E7	SCALE	fattore di scala di una figura
232,233	#E8,#E9	SHPTAB	puntatore d'indice della ta- bella delle figure
234	#EA	CC	contatore dei collisioni

# INDIRIZZI DELL'INTERPRETE APPLESOFT

## FUNZIONI GRAFICHE

DEC	HEX	ASSEMBLER	FUNZIONE, RISULTATO
-3112	F308	HER2	cancellazione delle pagine
-3102	F3E2	HBR	$ER2 \rightarrow HPAG, 840 \rightarrow HPAG$
-3084	F3F4	BSND	schermo di un colore uniforme colore $\rightarrow$ Acc $\rightarrow$ HCOLOR1
	F6F6	HCOLR	colore $\rightarrow$ I $\rightarrow$ HCOLOR0
-3035	F411	HPDSN	$x \rightarrow y, x \rightarrow B \text{ e } L$ $y \rightarrow$ Acc $\rightarrow y$ calcolo di $HBASL, HJ$ HBASK e ED7 e HCOLOR1
-2971	F465	INTI	seguito da N, incremento o decre- mento di $xH, xL$ e $y$
-2861	F405	INTY	seguito da N, incremento o decre- mento di $y \rightarrow HBASL, H$
-2613	F3C8	IPDS	$HBASL, HJ$ ED7 $\rightarrow xH, xL$ e $y$
-2485	F457	HPLOT	tracciamento del punto $x \rightarrow y, x$ e $y \rightarrow A$ con l'aiuto
-2482	F445	PLOT	tracciamento del punto definito da HCOLOR1; HBASK; ED7 $\rightarrow y$ ; $HBASL, H$ seguendo le istruzioni: LDA HCOLOR1 EDR (HBASL),Y AND HBASK EDR (HBASL),Y STA (HBASL),Y
-2758	F3DA	HLINE	tracciamento di una linea in qualsunque direzione dal punto at- tuale al punto $x \rightarrow I, A$ e $y \rightarrow Y$

Nome	Indirizzi		Funzioni
	Hex	Dec	
LOMEM	04A,04B	74,75	Inizio delle variabili
HIMEM	04C,04D	76,77	Fine del testo BASIC
PP	0C6,0C9	202,203	Inizio del testo BASIC
PV	0CC,0CD	204,205	Fine delle variabili
PR	0DC,0DD	220,221	N° della linea corrente
PN	0DE,0DF	222,223	Nome dell'ultima variabile a cui si è fatto riferimento
PI	0E8,0E9	224,225	Inizio dell'istruzione corrente

## Riepilogo



# INTEGER - ESDP10

Esempio di implementazione di un programma e delle sue variabili in memoria RAM. (Vedere esempio n° 2 in Applesoft).

>LIST

```
10 REM INTEGER
15 DIM NR(20)
20 INPUT "NOME ?":NR
30 FOR K=1 TO LEN(NR)-2
40 PRINT NR(K); " ";NR(K,K+1); " "
    I
50 NEXT K
60 END
```

>CALL-151

\*4A,4B

LOWER

004A- 00 0B

\*CA,CB

00CA- 9A 9B

FF

\*4C,4D

004C- 00 9A

HIERA

\*959A,95FF

```
959A- 00 0A 00 00 00 00 00 00
95A0- CE 04 C5 C7 C8 B2 01 00
95A8- 0F 00 4E CE 40 02 B2 14
95B0- 00 02 01 00 04 00 00 00
95B8- CE CF CD C5 A0 BF 29 0A
95C0- CE 40 01 00 0E 00 00 00
95C8- 0A B1 01 00 07 0B CE 40
95D0- 72 13 B2 02 00 01 00 00
95D8- 00 01 CE 40 0A CB 02 45
95E0- 0A A0 29 45 CE 40 0A CB
95E8- 23 CB 02 B1 01 00 02 45
95F0- 0A A0 29 47 01 0A 02 00
95FB- 0F CB 01 00 0C 00 01 01
```

Testo BASIC

( parole riservate

01 fine linea

() numero di byte  
della linea -1

I I n° di linea

\*CC,CB

00CC- 00 0B

FF

Esempio di implementazione di un programma e delle sue variabili in  
memoria RAM.  
(segue)

>RUN  
NOME 7SERGIO

Esecuzione  
del programma

SERGIO SE ERGIO ER RGIO RS GIO GI

>CALL-154

\*CC.CO

PI

CCCC- 20 00

CCCC.B1F

Variabili

0800- .CE 40 00 0A 08 03 C5 D2  
0808- C7 C9 CF 1E 53 4F 46 54  
0810- 00 22 08 14 00 84 22 4E  
0818- 4F 4D .CB 00 20 08 00 00

- 08,Contatore alla  
variabile successiva)
- valore (28 caratte-  
ri)
- R,(),valore



# INDIRIZZI INTEGER

Le parole riservate per ordine crescente di codice e gli indirizzi dei sottoprogrammi corrispondenti nell'interprete Integer BASIC.  
Codici da 800 a 815.

Parola riservata	Codice		Indirizzo	
	Hex	Dec	Hex	Dec -45536
E MOREN: 3	800	0		
Fine linea	801	1		
-	802	2		
!	803	3		
LOAD	804	4	8F00F	-3873
SAVE	805	5	8F140	-3776
CON	806	6	8F30A	-3518
RUN (n° di linea)	807	7	8EFF2	-4118
RUN	808	8	8EFEC	-4116
DEL	809	9	8E34F	-7513
, (per DEL)	80A	10		
NEW	80B	11	8E44B	-4739
CLR	80C	12	8E587	-4729
AUTO	80D	13	8E7C2	-4174
, (per AUTO)	80E	14		
RAM	80F	15	8EE34	-4534
HIMEM:	810	16	8F04B	-4019
LOWMEM:	811	17	8F0C9	-3895
+	812	18	8E787	-4267
-	813	19	8E782	-4278
*	814	20	8E222	-7646
/	815	21	8EF19	-4536

Le parole riservate (segue). Codici da #16 a #20

Parola riservata	Codice		Indirizzo	
	Hex	Dec	Hex	Dec -45316
= < > <> <= < > < AND OR MOD * / ( per DIM) ) per DIM THEN (n° di linee) THEN (istruzione) , (stringa) , (numero) * (indirizzo) * (fine) ! var. B indice	#16	22		
	#17	23		
	#18	24		
	#19	25		
	#1A	26		
	#1B	27		
	#1C	28		
	#1D	29		
	#1E	30		
	#1F	31	#E27A	-7558
	#20	32	#F371	-3215
	#21	33		
	#22	34		
	#23	35		
	#24	36		
	#25	37		
	#26	38		
	#27	39		
	#28	40		
	#29	41		
	#2A	42		
	#2B	43		
	#2C	44		

# INDIRIZZI INTERIOR

Le parole riservate (seguono). Codici da 830 a 840.

Parola riservata	Codice		Indirizzo	
	Hex	Dec	Hex	Dec -10536
i var. indice	830	45		
PEEK	836	46	8E0F6	-4562
RND	83F	47	8E0F6	-4574
SEN	830	48	8E05C	-4598
ADD	831	49	8E04A	-4626
PEL	832	50	8F03B	-3269
	833	51		
i per DIM	834	52		
+ (seguono)	835	53		
- (seguono)	836	54		
NOT	837	55	8E036	-4646
i	838	56		
= comparazione	839	57		
# di stringhe	83A	58		
LENi	83B	59	8E022	-4674
ASCII	83C	60	8F01B	-3299
SCRNI	83D	61	8E08A	-7542
, in SCRNI	83E	62		
i	83F	63		
# (intrinseco)	840	64		

Le parole riservate (seguono). Codici da 001 a 056.

Parola riservata	Codice		Indirizzo	
	Res	Sec	Res	Sec -40516
	041	65		
!	042	66		
"	043	67		
#	044	68		
\$	045	69		
%	046	70		
&	047	71		
'	048	72		
(	049	73		
)	048	74		
TEXT	048	75		
@	040	76		
CALL	048	77	00000	-4448
BOR (stringhe)	048	78	00130	-7888
DIR (numeri)	047	79	00F1C	-4322
TAB	050	80	00784	-8236
END	051	81		
INPUT (stringhe)	052	82	00171	-7823
INPUT (messaggi)	053	83		
INPUT (numeri)	054	84	00000	-5206
FOR	055	85	00130	-5820
= (FOR/NEXT)	056	86		

# INDIRIZZI INTEGRATI

Le parole riservate (sequel). Codici da 837 a 868.

Parola riservata	Codice		Indirizzo	
	Hex	Dec	Hex	Dec -45314
TO (STEP)	837	87	8C950	-5808
STEP	838	88	8F27F	-3463
NEXT	839	89		
, NEXT	83A	90		
RETURN	83B	91	8E845	-5979
OSUB	83C	92	8C83C	-4084
REN	83D	93		
LET	83E	94		
GOTO	83F	95	8C85B	-4083
IF	840	96	8E82B	-6104
PRINT (stringa)	841	97	8E853	-4665
PRINT (numero)	842	98		
PRINT	843	99		
POKE	844	100		
, (POKE)	845	101		
COLOR=	846	102	8EE4E	-4530
PLOT	847	103	8EE2F	-4545
, (PLOT)	848	104		
HLIN	849	105	8EE80	-4433
, (HLIN)	84A	106		
AT (HLIN)	84B	107		

Le parole riservate (seguono). Codici da 84C a 87F.

Parola riservata	Codice		Indirizzo	
	Hex	Dec	Hex	Dec -45534
VLIN	84C	108	8EE06	-4418
• (VLIN)	84D	109		
AT (VLIN)	84E	110		
VTAB	84F	111	8EE57	-4321
• (stringa)	870	112		
• (numero)	871	113		
!	872	114		
	873	115		
LIST	874	116		
• (LIST)	875	117		
LIST	876	118		
POP	877	119	8F167	-3737
MODSP (stringa)	878	120		
MODSP (numero)	879	121		
NOTRACE	87A	122	8F176	-3722
DSP (stringa)	87B	123	8F2E0	-3360
DSP	87C	124		
TRACE	87D	125	8F171	-3727
PAB	87E	126	8F3C9	-3127
ISB	87F	127	8F41A	-3046

## DOS: INDIRIZZI DISCHETTI

BOOT: utilizzo del DOS (installazione a freddo)

Programma	Localizzazione	Occupazione	Funzione
1 - BOOT 0	FROM scheda controllo #C600	256 byte	carica BOOT 1 in RAM
2 - BOOT 1	DISCHETTO: traccia 0, settore 0 RAM: #8000-#9000	1 settore 256 byte	carica BOOT 2 e se stesso
3 - BOOT 2	DISCHETTO: traccia 0, set.da 1 a 9 MASTER SLAVE (48K) #3700-#4000 #0700-#C000	9 settori  2304 byte	contiene BIOS carica il DOS ed eventual- mente il rila- catore
BOOT 1	DISCHETTO: traccia 0, settore 0 MASTER SLAVE (48K) #3600-#36FF #0600-#06FF	1 settore  256 byte	versione del BOOT 1 dispo- nibile per l'inizializza- zione di un disco vergine
4 - DOS	DISCHETTO: traccia 2, set.da 4 a 6 traccia 1, set.da F a 0 traccia 0, set.da F a C  MASTER SLAVE (48K) #1000-#3600 #9000-#B600	25 settori   4400 byte	sistema opera- tivo del co- muni e ges- tione dello spazio sul dischetto
Rilascatore	DISCHETTO MASTER traccia 0, set. 8 e 9 RAM: #1800-#1900  (non esiste sul dis- chetto SLAVE)	2 settori 512 byte	reinstallazione del DOS al suo posto de- finitivo #9000-#C000 (48 Kbyte)

**Organizzazione del dischetto**

Qualunque sia la versione del DOS, un dischetto è costituito da 35 tracce e i dati sono divisi in settori da 256 byte.

Versione 3.2	Versione 3.3
13 settori/traccia	16 settori/traccia
453 settori/dischetto	560 settori/dischetto
di cui 403 utili	di cui 496 utili
quindi 100144 byte utili	quindi 124976 byte utili

**Occupazione di tracce e settori**

Traccia 0, 1, 2	: DOS (sistema operativo)
Traccia 011, settore 0	: VTOC (occupazione)
Traccia 011, settori da 0F a 01	: DIRECTORY (catalogo)
Tracce da 012 a 022 e da 016 a 02	: programmi e archivi : utilizzatore

IL "directory" più lungo, registrabile su un dischetto può essere costituito da circa 124000 byte).

La "directory" può contenere il nome di 105 riferimenti.

Un riferimento è un insieme di 35 caratteri che comprendono:

- l'indirizzo dell'inizio dei settori occupati (n° della traccia, n° del settore) del file referenziato
- il tipo di file A, I, F, B, e la sua eventuale protezione contro la cancellazione accidentale
- il nome del file (30 caratteri)
- la lunghezza e il numero dei settori occupati (12 byte)



# COMANDI DOS

Comandi in ordine di apparizione nella tabella dei comandi

Comandi DOS		Indirizzo di entrata	
index		Hex	DEC
000 0	INIT	0004F	42319
001 1	LOAD	00413	42003
002 2	SAVE	00397	41879
003 3	RUN	00401	42193
004 4	CHAIN	004F0	42224
005 5	DELETE	00263	41571
006 6	LOCK	00371	41983
007 7	UNLOCK	00375	41987
008 8	CLOSE	0036A	41764
009 9	READ	00318	42267
00A 10	EXEC	003C6	42438
00B 11	WRITE	00319	42256
00C 12	POSITION	00300	42441
00D 13	OPEN	002A3	41633
00E 14	APPEND	00298	41624
00F 15	RENAME	00281	41601
010 16	CATALOG	0036C	42350
011 17	MON	00233	41523
012 18	MONON	00238	41533
013 19	PRN	00229	41513
014 20	END	0022E	41518
015 21	BACKFILES	00251	41553
016 22	FP	0037A	42362
017 23	INT	0039C	42398
018 24	RSAVE	00331	41777
019 25	BLDAD	00358	41821
01A 26	BRUN	0038C	41870
01B 27	VERIFY	00378	41897

Posizione	DOS	Punto estratto
00000-00FFF	BMTS	00785
00AC9-00FFF	gestione dei comandi	00AF0
01000-0AC8	programma principale	01000
01500-0CFF	3 buffer da 512 byte	

Configurazione da 48 Kbyte di memoria RAM

RTS Head - Write - Track - Sector  
(lettura - scrittura - traccia - settore)

Sottoprogrammi di accesso ad un settore: RTS

Tabella dei parametri: IOR

L'indirizzo di IOR è caricato nel registro A (parte alta) e in Y (parte bassa) prima della chiamata di RTS:

```
Esempio: LDR #100  
          LDR #000  
          JBR #319  
          RTS
```

\* 1000 : 01 60 01 00 11 00 11 10

\* 1008 : 00 09 00 00 01

\* 1011 : 00 01 EF 00

IOR: byte n° 4 : n° di traccia (011)

byte n° 5 : n° di settore (000)

byte 6, 7 : indirizzo di DCT (01011)

byte 8, 9 : indirizzo della zona di  
trasferimento in RAM

byte C : codice del comando

00 posizionamento

01 lettura

02 scrittura

03 formattazione

DCT costanti : 00 01 EF 00 della periferica

Indirizzo Hex	Contenuto	Funzione
100	JMP #900F	inizializzazione "a caldo"
101	JMP #9004	inizializzazione "a freddo"
104	JMP #A40F	gestione dei comandi
109	JMP #B705	lettura/scrittura di un settore (RWTS)
10C	LDA #900F	ricerca dell'indirizzo relativo all'e-
	LDY #900E	lesso dei parametri per la gestione
	RTS	dei comandi
10E	LDA #A402	ricerca dell'indirizzo relativo alla
	LDY #A401	tabella IOB dei parametri dell'RWTS
	RTS	
10A	JMP #A050	per sostituire i vettori di I/O #30,
10F	JMP #FA59	#39 e #3A, #37 con i puntatori del 808
		in caso di BRK in MONITOR
3F3,3F2	#900F	indirizzo di rinvio in caso di <u>RESET</u>
		(SOFTWARE)
3F4	#30	$PARUP = (3F3) + 0A5$
3F5	JMP #FF50	in caso di 4
3F6	JMP #FF45	in caso di <u>CTRL-C</u>
3F8	JMP #FF40	se interrupt non mascherato
3FE	#FF45	se interrupt

# Indirizzi diversi del 808 in memoria RAM

- File binari caricati in RAM con BLD40
  - indirizzo in #A472, #A75
  - lunghezza in #A460, #A61
- Programma eseguito con l'inizializzazione "a freddo"
  - nome traccia 1, settore 9, byte #75...
  - type = 9E42 : 34 BRUM
  - = 9E42 : 14 EXEC
- Cessazione della pausa attivata durante CATALOG
  - #E24 : 80

## DOS: PROGRAMMI DI UTILITÀ

### SYSTEM MASTER 3.2

- |   |               |   |
|---|---------------|---|
| B | FID           | - copia di file o programmi<br>- verifica tasso di occupazione sul disco  |
| A | COPY          | - copia integrale di un disco<br>1 (con una o due unità per floppy disk)  |
| B | BOOTIS        | - inizializzazione del sistema con il DOS<br>versione 3.2 (13 settori per traccia)  |
| B | BUFFIN        | - conversione di file o programmi scritti<br>sotto DOS 3.2 in file e programmi sotto<br>DOS 3.3   |
| B | MASTER CREATE | - creazione di un disco MASTER partendo da<br>un disco già inizializzato e utilizzabile<br>come disco SLAVE<br>- possibilità di modificare il nome di un<br>programma che dev'essere eseguito al<br>BOOTSTRAP |

### DAKING programming aid's 3.3

- |             |  |
|-------------|--|
| The Patcher | visualizzazione e modifica di un qualunque settore |
| The Peeker  | lettura di un file                                 |

# Implementazione di file e di programmi sul dischetto esempio

## 1 - estratta del CATALOG

DISK VOLUME 254

```
*A 006 HELLO
*I 018 ANIMALS
*F 003 APPLE PROMS
*I 006 APPLESOFT
*I 026 APPLEVISION
*I 017 BIORHYTHM
*B 010 BOOT13
```

\* protezione in  
scrittura  
A,I,T,B tipi di  
file  
n numero dei  
settori occu-  
pati

## 2 - estratta della DIRECTORY del dischetto

Traccia 011 Settore 00F

00	00	01	0E	00	00	00	00				
08	00	00	00	13	0F	83	CB	CB	HE		
10	CC	CC	CF	A0	A0	A0	A0	A0	LLO		
18	A0	A0	A0	A0	A0	A0	A0	A0			
20	A0	A0	A0	A0	A0	A0	A0	A0			
28	A0	A0	A0	A0	0A	00	14	0F			
30	83	C1	CE	C9	CD	C1	CC	D3	ANIMALS		
38	A0	A0	A0	A0	A0	A0	A0	A0			
40	A0	A0	A0	A0	A0	A0	A0	A0			
48	A0	A0	A0	A0	A0	A0	A0	A0			
50	00	15	0F	83	C1	D0	D0	CC	APPL		
58	CB	A0	D0	D2	CF	CD	D3	A0	E PROMS		
60	A0	A0	A0	A0	A0	A0	A0	A0			
68	A0	A0	A0	A0	A0	A0	A0	A0			
70	A0	A0	03	00	14	0F	83	C1	A		
78	D0	D0	CC	C5	D3	CF	C6	D4	APPLESOFT		
80	A0	A0	A0	A0	A0	A0	A0	A0			
88	A0	A0	A0	A0	A0	A0	A0	A0			
90	A0	A0	A0	A0	A0	0A	00	17			
98	0F	83	C1	D0	D0	CC	C5	D6	APPLEV		
A0	C9	D3	C9	CF	CE	A0	A0	A0	ISION		
A8	A0	A0	A0	A0	A0	A0	A0	A0			
B0	A0	A0	A0	A0	A0	A0	A0	A0			
B8	1A	00	18	0F	83	C2	C9	CF	BIO		
C0	D2	C8	D9	D4	C8	CD	A0	A0	RHYTHM		
C8	A0	A0	A0	A0	A0	A0	A0	A0			
D0	A0	A0	A0	A0	A0	A0	A0	A0			
D8	A0	A0	A0	11	00	19	0F	83			
E0	/C2	CF	CF	D4	B1	B3	A0	A0	BOOT13		
E8	A0	A0	A0	A0	A0	A0	A0	A0			
F0	A0	A0	A0	A0	A0	A0	A0	A0			
FB	A0	A0	A0	A0	A0	A0	0A	00			

C) traccia, settore  
del seguente file

E) traccia, settore  
dell'elenco dei set-  
tori occupati da  
questo file

O tipo di file  
02 Applesoft  
01 Integer  
00 file ?  
04 binario

// nome del file  
completato per 30  
caratteri con degli  
spazi

L lunghezza del  
file e numero dei  
settori

# 000 - ESEMPIO (segue)

## 1 - estratto della FFC e tavola d'occupazione dei settori

Traccia 011

Settore 0

00 04 11 00 00 00 00 00  
 08 00 00 00 00 00 00 00  
 10 00 00 00 00 00 00 00  
 18 00 00 00 00 00 00 00  
 20 00 00 00 00 00 00 00  
 28 00 00 00 00 00 00 00  
 30 00 FF 00 00 00 00 01  
 38 00 00 00 00 00 00 00  
 40 00 00 00 00 00 00 00  
 48 00 00 00 00 00 00 00  
 50 00 00 00 00 00 00 00  
 58 00 00 00 00 00 00 00  
 60 FF FF 00 00 00 00 00  
 68 00 7F 00 00 01 FF 00 00  
 70 00 00 00 00 00 00 00  
 78 00 00 00 00 00 00 00

È 3 traccia/settore  
 del 1° settore del  
 contenuto

versione del 000  
 A(1,3) e' del valor-  
 re (254)

☐ = 122 settori  
 max. nell'elenco  
 degli indirizzi  
 (traccia/settore)  
 dei settori occupati  
 da un file

80 7F 00 00 00 00 00 00  
 88 00 00 00 00 00 00 00  
 90 00 00 00 00 00 00 00  
 98 00 00 00 00 00 00 00  
 A0 00 00 00 00 00 03 00 00  
 A8 00 00 00 00 00 00 00 00  
 B0 00 00 00 00 00 00 00 00  
 B8 00 00 00 00 00 00 00 00  
 C0 00 00 00 00 00 00 00 00

☐ = 33 tracce max.  
 per dischetto

⊞ = 16 settori per  
 traccia

— = 256 byte per  
 settore

settori occupa-  
 ti in ciascuna trac-  
 cia successiva

FE0C8A98 74543210

se un settore è li-  
 bero il corriappon-  
 dente bit è posto a  
 1

FF FF 0000 la traccia n° 10 ha tutti i suoi settori liberi  
 00 2F 0000 la traccia n° 12 ha i settori F,E,D,C,A,9,8,7 occupati

4 = estratta di un elenco di indirizzi (traccia/settore) di settori occupati da un file

Esempio: HELLO, traccia 413, settore 4E

00- 00 00 00 00 00 00 00 00 00	_____indirizzi dei
00- 00 00 00 00 00 <u>13 06</u> <u>13 06</u>	settori successi-
10- <u>13 06</u> <u>13 06</u> <u>13 06</u> 00 00	vamente occupati
10- 00 00 00 00 00 00 00 00	da HELLO

5 = estratta della coda di un programma registrata su disco

Esempio: HELLO, scritto in Applesoft

traccia 413      settore 4E

00	(71 04)19	08	0A	00	B2	20	0	2	C pò pA 2 lunghezza
08	20	20	20	20	44	4F	53	20	-- DOS del programma in nu-
10	33	2E	33	20	48	48	4C	4C	3.3 HELLO zero di byte occu-
18	4F	00/20	08	14	00	B2	20	0	2 pati in RAM
20	00/20	08	1E	00	89	3A	BA	(	11
28	00/2E	08	28	00	97	00/59		,	1 Y pbi parte bassa
30	08	32	00	BA	32	44	4F	53	2 1 "DOS pbi parte alta
38	20	56	45	52	53	49	4F	4E	VERSION
40	20	33	2E	33	20	20	20	20	3.3
48	20	20	20	20	20	20	20	20	// istruzioni di programma
50	30	38	2F	32	35	2F	38	30	08/25/80
58	22	00/88	08	3C	00	BA	3A	"	( 11
60	BA	22	41	50	50	4C	45	20	1 "APPLE
68	49	49	20	50	4C	55	53	20	11 PLUS
70	4F	52	20	52	4F	40	43	41	OR ROMCA
78	52	44	20	20	53	59	53	AD	8YS



## DOS - ESEMPIO (segue)

4 - estratto della coda di un programma binario

Esempio: BOOT13, traccia 019, settore 02

00	00 17 F0 08 20 E3 03 04	C'è già la lunghezza del programma in numero di byte in RAM
08	00 05 01 40 01 01 00 00	
10	90 17 CB 01 00 00 91 17	
18	20 5B FC 40 FF CB 0F 96	
20	17 0B 0F 00 20 ED FD 2B	Indirizzo di RAM di inizio del programma scritto in linguaggio mac- china
28	10 F3 A9 BF 05 33 20 6A	
30	FD 4D 00 02 C9 00 F0 0F	
38	C9 01 90 DC C9 00 00 00	
40	0A 0A 0A 0A 00 02 17 A9	Istruzioni di pro- gramma in linguaggio macchina
48	17 40 01 20 00 10 00 F7	
50	4D FE 16 00 0A 17 05 13	
58	E6 13 AD FF 16 4A 4A 4A	
60	05 10 A9 17 40 01 20 00	
68	1D 00 F7 EE 0A 17 EE 0A	
70	17 4D 0A 17 C5 10 F0 0A	
78	90 0B AD 02 17 AA A9 00	

BOOT13 è stato salvato sul dischetto con il comando  
SAVE BOOT13, AMP0, L0100

Implementazione di un file su dischetto

7 - estratto di un file di tipo 7

Esempio: APPLE P800, traccia 015, settore 02

00	07 05 0D C4 C5 CC 40 01	75 DEL 1	record di lunghezza fissa
08	00 00 00 AC 01 02 05 00	000,1250	
10	0D C3 C1 D6 C5 40 02 C1	SAVE RA	AC separator ", "
18	CE C4 CF CD 0D C8 CF CD	NBOM HOM	
20	C5 0D 02 05 CE 0D 00 00	E RUN	record di lunghezza fissa
28	00 C1 02 C1 CC CC C5 CC	PARALLEL	
30	40 00 02 C9 CE D4 AC 02	PRINT,2	record di lunghezza fissa
38	05 0A AC 00 AC 05 00 00	0A,0,500	
40	00 00 00 00 00 00 00 00		record di lunghezza fissa
48	00 00 00 00 00 00 00 00		
50	C3 CF C8 CD D5 CE C9 C5	COMMUNIC	record di lunghezza fissa
58	C1 D4 C9 CF CE D3 AC 02	ATIONS,2	
60	05 0A AC 00 AC 01 02 05	0A,0,125	record di lunghezza fissa
68	00 00 00 00 00 00 00 00	0	
70	00 00 00 00 00 00 00 00		record di lunghezza fissa
78	4B CE CF D4 40 C1 D6 C1	INOT AWA	









L. 15.000

Cod. 340H ISBN 88-7058-140-2

Questo libro è destinato a risiedere, in permanenza, a fianco del vostro Apple. Esso vi ricorda tutte le informazioni relative a riferimenti che potranno essere reperite velocemente: sintassi dei comandi, codici dei caratteri, messaggi di errore, linguaggio macchina, connessioni e indirizzi utili.

Le informazioni sono date senza eccessivi dettagli poiché lo scopo principale di questo libro è quello di fornire un rapido accesso alle informazioni; per una esposizione più introduttiva e completa, si potrà far riferimento ai libri "La pratica dell'Apple II" e "Apple II - Guida all'uso".

Si termina con una raccolta di "trucchi" di differenti livelli, ma tutti utili, i "come...?" sono disposti senza un ordine particolare, è comunque possibile il loro rapido reperimento grazie ad un apposito indice.