

TABELLA N. 1 - COMANDI DOS

1 APPEND	6 DATE	11 FREE	16 PRINT
2 ATTRIB	7 DEBUG	12 KILL	17 PROT
3 AUTO	8 DIR	13 LIB	18 RENAME
4 BASIC	9 DUMP	14 LIST	19 TIME
5 COPY	10 FORMAT	15 LOAD	20 VERIFY

TABELLA N. 2 - COMANDI E ISTRUZIONI BASIC

1 AND	20 EOF	39 LIST (L)	58 PRINTTAB (?TAB)
2 AUTO	21 ERL	40 LLIST	59 PRINTUSING (?USING)
3 CLEAR	22 ERR	41 LOAD	60 PUT
4 CLOSE	23 ERROR	42 LOC	61 READ
5 CMD"D"	24 FIELD	43 LOF	62 REF
6 CMD"funzione Dos"	25 FOR...NEXT	44 LPRINT	63 REM (')
7 CMD"S"	26 FRE	45 LSET	64 RENUM
8 CONT	27 GET	46 MEM	65 RESTORE
9 DATA	28 GOSUB	47 MERGE	66 RESUME
10 DEFDBL	29 GOTO	48 NEW	67 RETURN
11 DEFFN	30 IF...THEN...ELSE	49 NOT	68 REVOFF
12 DEFINT	31 INKEY \$	50 ON...GOSUB	69 REVON
13 DEFSTR	32 INPUT	51 ON...GOTO	70 RSET
14 DEFUSR	33 INPUT #	52 ONERRORGOTO	71 RUN
15 DELETE (D)	34 INSTR	53 OPEN	72 SAVE
16 DIM	35 KILL	54 OR	73 STOP
17 EDIT (E)	36 LET	55 PRINT (?)	74 SYSTEM
18 END	37 LINEINPUT	56 PRINT #(?#)	75 TROFF
	38 LINEINPUT #	57 PRINT @(?@)	76 TRON

TABELLA N. 3 - FUNZIONI DI STRINGA

1 ASC	5 CVS	9 MID \$	13 RIGHT \$
2 CHR \$	6 INSTR	10 MKD \$	14 STR \$
3 CVD	7 LEFT \$	11 MKI \$	15 STRING \$
4 CVI	8 LEN	12 MKS \$	16 VAL

TABELLA N. 4 - FUNZIONI ARITMETICHE

1 &H	6 CINT	11 FIX	16 SGN
2 &O	7 COS	12 INT	17 SIN
3 ABS	8 CSNG	13 LOG	18 SQR
4 ATN	9 E	14 RANDOM	19 TAN
5 CDBL	10 EXP	15 RND	20 ↑

TABELLA N. 5 - FUNZIONI GRAFICHE

1 CLS	2 POINT	3 RESET	4 SET
-------	---------	---------	-------

TABELLA N. 6 - FUNZIONI SPECIALI

1 (tasto ^)	7 (tasti CTRL + H)	13 (tasti CTRL + T)	19 PEEK
2 (tasto □)	8 (tasti CTRL + I)	14 (tasti CTRL + X)	20 POKE
3 (tasto)	9 (tasti CTRL + J)	15 (tasti CTRL + Y)	21 POS
4 (tasto ')	10 (tasti CTRL + L)	16 (tasto ESC)	22 TIME \$
5 (tasto .)	11 (tasti CTRL + P)	17 INP	23 USR
6 (tasti CTRL + A)	12 (tasti CTRL + R)	18 OUT	24 VARPTR

TABELLA N. 7 - SIMBOLI

1 !	4 %	7 ()	10 ;
2 #	5 &	8 .	11 ?
3 \$	6 "	9 :	

ISTRUZIONI sull'uso del BASIC e del NE/DOS



Non avendo visto pubblicato sul numero precedente, nessun articolo riguardante il microcomputer, vi sarete subito allarmati credendo che i nostri tecnici se ne fossero dimenticati facendovi con questo silenzio credere di essere giunti alla fine. L'attesa di questo Basic, lo sappiamo, è stata decisamente penosa, però bisogna anche sapere che un linguaggio Basic non lo si inventa; bisogna acquistarlo e la fonte più qualificata è una sola: gli Stati Uniti.

Dopo averlo acquistato, per farlo girare occorreva ovviamente adattarlo agli indirizzi del nostro computer.

La prima volta, terminata questa operazione, ci siamo accorti che esso non risultava di nostro gradimento, quindi lo abbiamo dovuto ritornare, chiedendo in sostituzione un'edizione più completa e aggiornata.

Noi infatti non ci accontentavamo di un normale Basic ma volevamo qualcosa di più, che nulla avesse da invidiare ai migliori reperibili in commercio, tanto che, già fin dalle nostre prime richieste ci è sempre stato risposto che pretendavamo l'impossibile.

Noi volevamo un DOS studiato appositamente per Nuova Elettronica, che avesse caratteristiche e potenza comparabili al sistema operativo CP/M e cioè:

— Volevamo un Basic senza trucchi «aperto» in modo che tutti potessero liberamente duplicarlo o modificarlo senza incorrere in sanzione per il diritto internazionale del «copyright».

— Volevamo aumentare la capacità in kilobyte del dischetto portandolo a 102,4 K in quanto i 78 K precedenti, ci siamo accorti, che erano troppo pochi.

— Volevamo avere un'ampia libertà d'azione per poterlo in futuro adattare per il grafico, maiuscole, minuscole e altre imprevedibili necessità.

— Volevamo soprattutto un potente sistema operativo capace di supportare un BASIC veramente esteso. Chiariamo cosa intendiamo noi per esteso:

Dovete sapere che ogni computer è dedicato ad uno specifico campo di applicazione. Nel caso dei microcomputer (cioè quelli basati su integrati chiamati CPU), noti anche come PERSONAL-COMPUTER oppure HOME-COMPUTER, gli usi

principali sono due e cioè GESTIONALE e SCIENTIFICO.

È ovvio quindi che i vari linguaggi impiegati risultino commisurati alle applicazioni; succede allora che il Basic usato sul gestionale ha soprattutto potenza nel trattare le stringhe, mentre quello usato sullo scientifico è specializzato in calcolo.

Un esempio di questi Personal-Computer possono essere: APPLE 2, APPLE 3, COMMODORE, SHARP MZ80K e MZ80B, HP85 (HEWELETT-PACKARD), quest'ultimo prettamente scientifico.

Noi cercavamo invece un Basic che racchiudesse in sé sia le caratteristiche scientifiche che gestionali con l'aggiunta di comandi e istruzioni personalizzate dettate dalla nostra esperienza di programmazione.

Il risultato è stato veramente eccellente poiché sul calcolo, oltre alle 16 cifre significative, abbiamo ottenuto una precisione addirittura superiore di una cifra rispetto all'HP85, mentre sul gestionale una stringa può essere trattata in ogni modo possibile.

Per avere tutto questo si è impiegato molto tempo anche perché, constatato che con i telex non si riusciva ad intenderci, abbiamo dovuto recarci personalmente nelle loro sedi per trovare una soluzione ottimale dei nostri problemi.

Finalmente lo scopo che ci eravamo prefissati è stato raggiunto, e consegnandovi questo Basic possiamo preannunciarvi pure altre interessanti novità, (non chiedetecele ovviamente domattina).

Ad esempio, stiamo preparando una scheda interfaccia video grafica da 80 x 24 colonne a due tonalità di grigi, a caratteri tondeggianti, completa di maiuscolo e minuscolo.

In seguito, prepareremo un dischetto floppy utile per studiare, con l'aiuto del computer, il Basic. Inserendo tale disco nel drive, potrete scegliere tra le tante, la lezione che più vi interessa; il computer ve le presenterà sul video e sulla stampante, spiegandovi, con esempi, le soluzioni più idonee.

Utilizzando il computer quindi, sarete accompagnati per mano nel complesso mondo della programmazione fino al punto da poter scrivere da soli i programmi che vi necessitano senza difficoltà di sorta, sicuri di disporre, con la massima sicurezza, di mezzi e conoscenza appropriati per ogni tipo di problema.

1 - INTRODUZIONE

Per poter lavorare con questo BASIC + NE-DOS che occupa un totale di 26K occorre che il computer risulti così configurato:

**SCHEDA CPU,
INTERFACCIA VIDEO,
MONITOR,
INTERFACCIA FLOPPY DISK,
ALMENO UN DRIVE PER FLOPPY DISK,
MEMORIE RAM PER ALMENO 40 KILOBYTE.**

Naturalmente al posto del monitor come già sapete potrete utilizzare anche un comune televisore.

Sarebbe poi quanto mai consigliabile avere una stampante da 80 colonne ad aghi: in quanto solo con essa si possono fare bolle di accompagnamento, fatture, contabilità, comunicazioni varie, cataloghi, lettere, elenchi. Anche con una stampante termica potete fare tante cose, ma non a livello commerciale.

A - LA NUOVA EPROM

Coloro che ci hanno seguito finora nella realizzazione del computer avranno senza dubbio già apportato le modifiche indicate sul numero 76 e 77 di NUOVA ELETTRONICA per quanto riguarda la scheda CPU, l'espansione di memoria e la INTERFACCIA FLOPPY.

Assieme al disco BASIC-NEDOS avrete ricevuto pure una Eprom siglata 1.390 che dovrete sostituire nella scheda floppy LX.390 con quella già esistente siglata 390 per questi semplici motivi:

1) Con la nuova EPROM formattiamo i dischetti floppy in modo diverso dal precedente (vedi fig. 1) in modo da aumentare la capacità che da 78 kilobyte passerà così a 102,4 kilobyte.

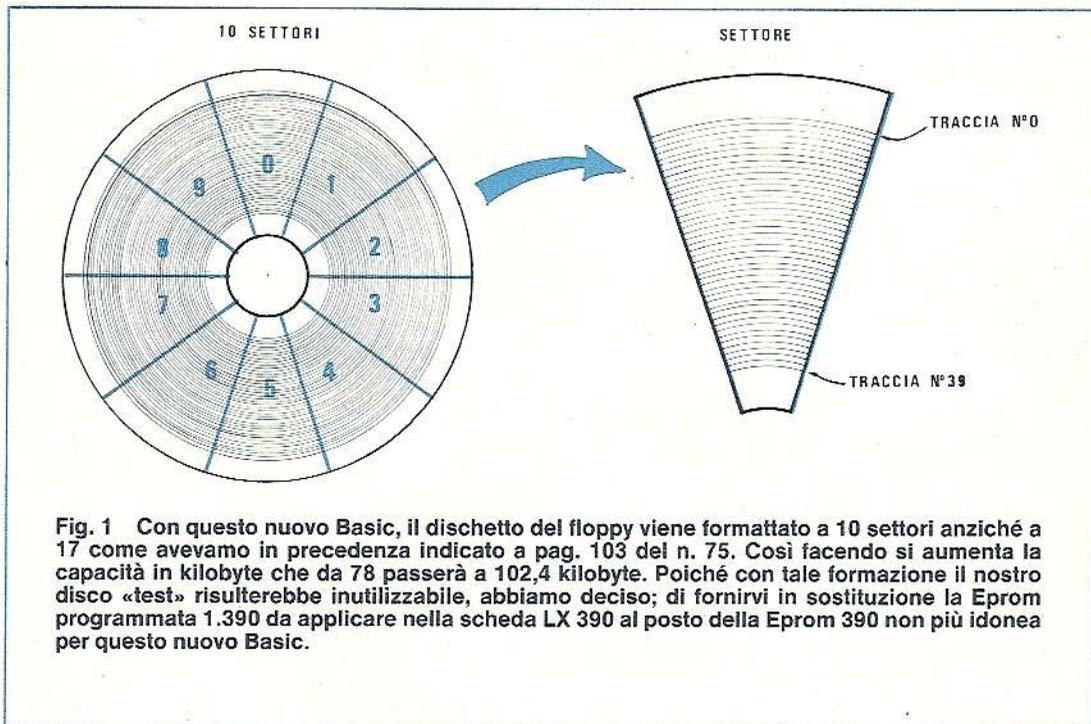


Fig. 1 Con questo nuovo Basic, il dischetto del floppy viene formattato a 10 settori anziché a 17 come avevamo in precedenza indicato a pag. 103 del n. 75. Così facendo si aumenta la capacità in kilobyte che da 78 passerà a 102,4 kilobyte. Poiché con tale formazione il nostro disco «test» risulterebbe inutilizzabile, abbiamo deciso; di fornirvi in sostituzione la Eprom programmata 1.390 da applicare nella scheda LX 390 al posto della Eprom 390 non più idonea per questo nuovo Basic.

2) Il disco BASIC NE-DOS contiene molti programmi di utilità e a questi sono stati aggiunti pure quelli di formattazione e di duplicazione. Abbiamo poi scoperto che era estremamente scomodo, dopo aver fatto partire il dischetto, vedere un lungo elenco di possibilità operative per scegliere poi quella che ci serviva. Abbiamo quindi modificato il tutto in modo che quando accendete il computer, e inserite il disco DOS-BASIC nel drive 0, e premuto il tasto dello spazio il drive parte e dopo pochi secondi il BASIC è pronto e pur se la cosa non è evidenziata, è stato caricato anche il DOS. Tutti gli altri programmi di utilità sono disponibili immediatamente portandosi a livello DOS.

Quando inserirete la nuova EPROM, nella scheda INTERFACCIA FLOPPY al posto della vecchia controllate che i piedini si siano inseriti tutti correttamente nei corrispondenti fori dello zoccolo e che la tacca di riferimento risulti rivolta come per la vecchia EPROM.

E ora iniziamo con il nostro BASIC NEDOS.

B - CARICAMENTO DEL DOS E DEL BASIC

Con la nuova EPROM inserita sulla interfaccia Floppy e con almeno 32 K di RAM montati accendete il computer e pigiate il pulsante di RESET posto sulla scheda CPU. Sul monitor apparirà la scritta:

**BOOTSTRAPPING VERSION 1.0
INSERT NE-DOS-DISK AT DRIVE 0
THEN TYPE "SPACE"**

(lancio versione 1.0)
(inserire disco NE-DOS nel drive 0)
(poi premere "spazio")

Come si vede, lo scambio di informazioni col sistema avviene in lingua Inglese, che domina sovrana in questo campo. Se non avete dimestichezza con l'inglese non preoccupatevi: le parole da conoscere sono poche e con l'uso frequente le imparerete velocemente. Ad ogni buon conto tenete sempre sott'occhio le tabelle che forniremo, in modo da poterle consultare facilmente.

Dopo che è apparsa sul video la scritta, inserite il disco NE-DOS nel DRIVE 0 quindi pigiate il tasto dello SPAZIO, cioè quello più lungo in basso sulla tastiera alfanumerica.

Dopo circa 15 secondi il computer comincia l'esecuzione del programma dimostrativo MOSTRA.

Nell'intervallo di tempo menzionato il computer ha caricato in memoria il DOS, il BASIC e per ultimo il programma MOSTRA e parte con la sua esecuzione.

Il programma MOSTRA da solo non si fermerà più, quindi dovete farlo voi battendo contemporaneamente i due tasti BREAK.

Pigiate ora il tasto DEL (serve per cancellare il video e togliere la scrittura espansa), poi il tasto RETURN e successivamente i due tasti CTRL e T contemporaneamente (serve per togliere la scrittura in negativo). Ovviamente se all'atto del BREAK non c'era scrittura espansa potete risparmiarvi la prima operazione, come potete non fare la seconda se non era in atto la scrittura in negativo (REVERSE).

Per poter proseguire dopo aver pigiato il tasto DEL ricordate sempre che occorre premere anche il tasto RETURN.

Se vi sbagliate alla partenza e non inserite il dischetto del DOS, tra le due prime scritte appare:

NO NE-DOS-SYSTEM

(non c'è il dischetto NE-DOS)

Tale scritta appare anche se esiste qualche errore di lettura dovuto magari alla testina sporca o a mancanza di schermatura sul DRIVE. Se invece avete battuto lo «spazio» senza inserire nessun disco nel drive 0, dopo poco il drive stesso si ferma e il led rimane acceso; a questo punto potete **solo** premere nuovamente il tasto RESET sulla scheda CPU e ripartire da capo.

Se in seguito ogni qualvolta ricaricate il BASIC non volete avviare anche il programma MOSTRA, dovete procedere come segue: alla partenza, invece di premere il tasto SPAZIO, premete il tasto RETURN e tenetelo premuto fino a quando il drive non si ferma; a questo punto lasciate il tasto RETURN e sul video vedrete apparire la scritta:

DOS READY

(DOS pronto)

Per passare al BASIC basta scrivere sul video la parola BASIC e pigliare il tasto RETURN: dopo circa sette secondi appare la scritta

READY

>.

E questo significa che siete a livello BASIC.

Vediamo ora brevemente cosa vuol dire LIVELLO DOS e LIVELLO BASIC.

LIVELLO DOS

(DOS = **D**ISK **O**PERATIVE **S**YSTEM).

Questo è il livello del SISTEMA OPERATIVO, che rende possibile l'uso dei Floppy Disks. Questo livello può essere raggiunto in via diretta o attraverso il BASIC.

A livello DOS si possono eseguire diversi programmi di utilità (come formattare dischetti vergini, fare copie di dischetti, ecc.).

LIVELLO BASIC

(BASIC = **B**EGINNER'S **A**LL-PURPOSE **S**YMBOLIC **I**NSTRUCTION **C**ODE).

A questo livello è possibile eseguire comandi diretti, oppure scrivere, correggere o eseguire programmi in linguaggio BASIC. Ovviamente dal livello BASIC si può passare al livello DOS o viceversa.

Non fatevi prendere dalla frenesia di voler eseguire subito dei programmi e imparare tutto in una sola volta; le nozioni sono abbastanza numerose e collegate tra di loro, ed esporle tutte assieme è abbastanza difficile; ce ne stiamo accorgendo per il libro che stiamo preparando, un «libro» veramente valido che insegni veramente a programmare in BASIC non il solito libro «INUTILE» e incomprensibile. Cercate sempre di assimilare bene un concetto prima di passare a quello successivo. Seguiteci e vedrete che poco per volta acquisterete familiarità col computer.

2 - DOS

A - INTRODUZIONE

La descrizione dettagliata di tutte le operazioni possibili porterebbe via molto spazio. Poiché pensiamo preferiate subito conoscere le nozioni principali che vi mettano in grado di programmare in BASIC, illustriamo ora ciò che è indispensabile conoscere per poter cominciare.

Ad ogni modo, per i più esperti, abbiamo riportato in Tabella n° 1 (inizio articolo) l'elenco completo dei comandi DOS. Dopo aver caricato il BASIC, è possibile passare al livello DOS scrivendo semplicemente:

CMD "S"

e battendo il tasto RETURN per rendere operativo il comando impartito.

Ricordatevi sempre che dopo aver scritto una qualsiasi istruzione per renderla effettiva, **occorre sempre pigiare il tasto RETURN.**

Dopo aver pigiato il tasto RETURN, sul monitor vi apparirà la scritta:

DOS READY

la quale ci confermerà che ci troviamo al **livello DOS**. Dopo che appare tale scritta per tornare a livello BASIC sarà sufficiente scrivere sulla tastiera.

BASIC

e ovviamente dopo averlo scritto dovremo ricordarci di battere il tasto RETURN e dopo poco sul monitor vedremo scritto:

READY

>

e questo ci conferma che siamo ritornati al livello BASIC.

Ritornando alla scritta DOS READY potremo ora imparare a formattare i dischetti vergini e a duplicarli.

B - FORMATTAZIONE DEI DISCHI VERGINI

Tutti i dischetti vergini per risultare idonei a ricevere dei dati occorre prima formattarli. Per far questo bisogna sapere se avete un solo drive, oppure due o più drive in quanto il procedimento è leggermente diverso.

AVETE UN SOLO DRIVE

Dopo la comparsa della scritta DOS READY scrivete sulla tastiera

FORMAT

(non dimenticatevi di pigiare il tasto RETURN. **Importante:** d'ora in poi questa operazione «pigiare il tasto RETURN» anche se verrà omessa per non complicare inutilmente la descrizione, la dovrete sempre eseguire)

Dopo poco appare la scritta

1 NEDOS DISK FORMATTER PROGRAM 1.0 (programma 1.0 per formattare dischi)
WHICH DRIVE IS TO BE USED? (quale drive vuoi usare?)

ossia il computer ci informa che il programma di formattazione è pronto per essere avviato e prima di iniziare ci chiede quale DRIVE vogliamo usare. A questo punto avendo un solo drive dovrete togliere dal drive il disco col DOS-BASIC e sostituirlo con quello vergine da formattare.

Ricordatevi che se avete un solo DRIVE il suo numero è 0, quindi dovrete pigiare il tasto dello 0 (zero); attenzione a non confonderlo con la lettera O.

NOTA - Se pigiate il numero 0 senza aver prima cambiato il dischetto con uno vergine, sul monitor vedrete scritto

2 WRITE PROTECTED DISKETTE! (il disco è protetto contro la scrittura)
REPLY «N» TO END FORMAT. (scrivi N se vuoi uscire dalla formattazione)
REPLY «Y» AFTER ERROR CONDITION (scrivi Y dopo aver rimediato all'errore e
CORRECTED TO EXECUTE FORMAT se vuoi proseguire nella formattazione
COMMAND AGAIN. del disco inserito)

Cosa è successo? Semplicemente che il sistema si rifiuta di formattare il disco col DOS, perché ha il nastro adesivo di protezione, e quindi non può eseguire il comando FORMAT.

Ecco perché il computer vi offre due alternative: se rispondete **N** (N = NO) uscite dal comando di formattazione, se rispondete **Y** (Y = YES) prosegue con la formattazione. Prima di rispondere Y occorre togliere il disco DOS-BASIC e inserire il disco vergine.

Conviene sempre, comunque, scrivere il numero 0 'zero' solo dopo aver cambiato il dischetto; se succede il contrario, è preferibile rispondere alla domanda successiva con N, in quanto può accadere che il sistema perda il controllo e che si debba ricominciare da capo col caricamento del DOS.

Per completare il quadro della situazione, ricordiamo che lungo il bordo del dischetto è ricavato un intaglio rettangolare di circa cinque millimetri; qualora si chiuda questa tacca con una strisciolina di adesivo non è più possibile registrare dati su quel disco. In questo caso quindi il sistema rifiuta la formattazione. (All'interno di ogni confezione di dischi si trova un adesivo per questo uso).

Scrivendo il numero 0 sul video apparirà questa scritta:

3 DISKETTE NAME? (quale nome vuoi dare al dischetto?)

cioè viene richiesto il nome da assegnare al dischetto.

Potete rispondere introducendo un nome (il vostro, ad esempio MARIO) o, come minimo, qualche lettera: NE, oppure DISKA, DISKB ecc. se non scrivete nulla e battete solo RETURN il computer vi porrà nuovamente la domanda. La lunghezza del nome non deve superare gli otto caratteri: ve ne accorgete subito perché oltre tale limite il cursore non si muove più.

Eseguita questa operazione, vedrete sul video la scritta

4 CREATION DATE (data di formattazione)
(MM/DD/YY)? (mese/giorno/anno)

Dovete cioè introdurre, la data di formattazione. Notare che questa viene richiesta nella forma anglosassone, cioè prima il mese poi il giorno e quindi l'anno. Conviene rispettare questo ordine, altrimenti può succedere che la data venga rifiutata per un'evidente incompatibilità di valore nel numero del mese. Questo succede solo quando la data deve essere introdotta nel comando di duplicazione, che vedremo tra poco; in quel caso se la prima cifra di MM non è un numero oppure è un numero uguale o maggiore di due, la data è rifiutata dal sistema. Tanto vale quindi abituarsi ad introdurla nel modo richiesto, oppure a metterne una di comodo sempre uguale, ad esempio 11/11/11.

Rispettate rigorosamente la forma, vale a dire scrivere sempre per il mese due cifre (per 1-2-3 occorre scrivere 01-02-03), poi la barra, poi altre due cifre, poi un'altra barra e successivamente le ultime due cifre. Esempio: 03/07/82 - 03/18/82 - 10/30/82.

Se non rispettate queste indicazioni il sistema non accetterà quanto introdotto e ve lo richiederà.

Dopo aver introdotto la data appare la seguente domanda

5 MASTER PASSWORD? (parola chiave principale?)

Qui valgono esattamente tutte le considerazioni fatte poc'anzi per rispondere alla domanda DISKETTE NAME. Comportatevi quindi allo stesso modo. In entrambi i casi, se volete far presto, potete limitarvi a battere uno spazio e il tasto RETURN.

Non preoccupatevi per ora di sapere esattamente a cosa servono questi dati; li vedremo andando avanti nella descrizione.

Proseguendo nella formattazione. Dopo aver risposto all'ultima domanda, sul monitor compare

6 DO YOU WANT TO LOCK OUT (vuoi inibire
ANY TRACKS? qualche traccia?)

Il computer ci chiede se volete bloccare qualche traccia. Se rispondete con Y (YES) vi verrà posta la seguente domanda

WHICH TRACKS (1-39)? (quale traccia da 1 a 39?)

La risposta deve consistere in uno o più numeri compresi tra 1 e 39, separati da un trattino (—). Rammentatevi che le tracce sono 40 numerate da 0 a 39; la traccia 0 è riservata al sistema, e per questo motivo nel dischetto è esclusa dal bloccaggio, quindi apparirà sempre 1-39.

Dopo aver dato il numero delle tracce che intendete bloccare, appare la scritta

FORMAT THE LOCKED OUT TRACKS? (vuoi formattare le tracce inibite?)

cioè il sistema chiede se deve formattare anche le piste interdette. Rispondete con una Y se è sì, e con una N se è no. Raramente se non in casi particolari si ha necessità di bloccare qualche traccia quindi alla domanda 6 rispondete sempre con un N.

Se per ipotesi avete messo nel drive un disco già formattato (con o senza dati), sullo schermo apparirà questa scritta

FORMAT REJECTED, (formattazione respinta)
DISKETTE CONTAINS DATA! (il disco contiene dati)
DISKETTE TO BE FORMATTED ANYWAY (vuoi formattare ugualmente il disco?)
(REPLY "Y" OR "N")? (rispondi "Y" oppure "N".)

cioè il computer vi informa che il disco inserito è già formattato e contiene dei dati; quindi prima di procedere vuole la conferma che effettivamente vogliamo formattarlo (il che significa anche cancellare tutti i dati) ed in questo caso scriveremo Y, se invece distrattamente abbiamo preso un disco che conteneva dei dati e che noi ritenevamo vergine, per salvarlo dovremo scrivere N.

Ritornando alla domanda 6 alla quale avremo scritto N in quanto non ci interessa bloccare nessuna traccia, il computer inizierà a formattare il dischetto, facendo apparire sul video il seguente messaggio

DIRECTORY WILL BE PLACED (l'indice sarà messo
ON TRACK 17 alla pista 17)

cioè ci informa che l'elenco dei programmi il computer li colloca nella traccia 17; poi sotto si vedrà la dicitura

FORMATTING TRACK 'n' (sto formattando la traccia n)

dove il numero 'n' varia da 0 a 39 mano a mano che l'operazione di formattazione procede. Raggiunto il numero 39, sul video apparirà questa scritta

VERIFYING TRACK 'n', SECTOR 'm' (sto verificando la traccia n, settore m)

dove 'n' varia come prima e 'm' va da 0 a 9: il computer cioè verificherà automaticamente, traccia per traccia e settore per settore controllando che la formattazione risulti corretta e che non vi siano tracce difettose nel disco.

Può succedere (anche se molto di rado) che il dischetto risulti difettoso in qualche sua parte. In tal caso potreste vedere sul monitor una dicitura come questa

7 VERIFYNG TRACK 18, SECTOR 04 (sto verificando la traccia 18, settore 04)
NOT FOUND! TRACK LOCKED OUT! (non l'ho trovata! traccia inibita!)

Cioè in sede di verifica nella formattazione il computer si è accorto che la traccia 18 del settore 04 non si è formattata correttamente, e in questo caso la considera come non disponibile per le future incisioni, e prosegue oltre.

Se non trova nessuna traccia difettosa apparirà invece

8 INITIALIZING SYSTEM INFORMATION (informazione nella inizializzazione)
FORMATTING COMPLETE (formattazione completata)
PRESS "ENTER" WHEN **SYSTEM** (premere "ENTER" quando il disco
DISKETTE MOUNTED ON DRIVE 0 **SISTEMA** è montato sul drive 0)

Cioè la formattazione è terminata quindi potete togliere dal drive il dischetto formattato e inserirvi nuovamente quello col DOS-BASIC. Premendo il tasto RETURN verrà ricaricato il DOS (se non succede niente o se vedete simboli strani, premete il RESET della scheda CPU LX.382 e ricaricate il DOS nel modo esposto all'inizio).

Viceversa, se qualcosa non è andato come di dovere, il computer segnala errore e si deve ripetere il procedimento da capo.

Se vi succede di fare un BREAK durante una qualsiasi delle fasi di formattazione, il computer non risponderà più a nessun comando. In tal caso dovrete ripartire da capo nel modo solito: pigiando il tasto RESET sulla scheda CPU e ricaricando il DOS-BASIC.

Se non riuscite a proseguire in nessun modo, controllate se siete a livello DOS oppure se il disco inserito è alla rovescia (ricordate che l'etichetta del disco va dalla parte opposta del LED).

Non lasciatevi comunque trarre in inganno dalla lunghezza di queste spiegazioni: a dirlo sembra lungo e difficile, a farlo ci si rende conto che invece è molto facile. A titolo informativo diremo che il tempo necessario per formattare un disco si aggira su 50 secondi.

Un'altra cosa: se la velocità del drive non è quella giusta, il sistema ve lo segnala con

CAN'T FORMAT, (non posso formattare)
MOTOR SPEED TOO FAST (SLOW)! (velocità del motore troppo alta o bassa)
REPLY "N" TO END FORMAT. (rispondi "N" per uscire dalla formattazione)
REPLY "S" AFTER ERROR CONDITION (rispondi "S" dopo che la condizione d'errore
CORRECTED TO EXECUTE FORMAT è stata corretta per eseguire
COMMAND AGAIN. nuovamente la formattazione)

Vale a dire: è impossibile formattare perché la velocità del motore è troppo alta (o bassa), ecc. Come vedesi abbiamo un sistema operativo molto «intelligente» che ci segnala anche i difetti del drive.

E c'è dell'altro. Se non è stato chiuso lo sportellino del drive o se non è stato inserito il dischetto, il computer segnala una o entrambe le condizioni con le scritte

DOOR NOT CLOSED ON DRIVE! (lo sportello del drive non è chiuso)
NO DISKETTE IN DRIVE! (non avete inserito il dischetto)
REPLY "N" TO (idem come sopra)

Da quanto spiegato finora potete comprendere che il DOS-BASIC che vi abbiamo procurato è quanto di meglio esista attualmente in commercio.

Per facilitare il più possibile l'operazione di formattazione, ecco un breve sunto di quello che dovete fare (i numeri tra parentesi richiamano i messaggi già menzionati).

- **PORTARSI A LIVELLO DOS**
- **DIGITARE FORMAT**
- **ALLA DOMANDA (1) RISPONDETE DIGITANDO 0**
- **ALLA DOMANDA (3) DATE UN NOME A VOSTRA SCELTA**
- **INTRODUCETE LA DATA ALLA (4) ESEMPIO 11/11/11**
- **DIGITATE UN NOME O UNO SPAZIO ALLA DOMANDA (5)**
- **RISPONDETE N ALLA DOMANDA (6)**
- **QUANDO APPARE LA (8) RIMETTETE NEL DRIVE IL DISCO BASIC. LA FORMATTAZIONE È FINITA, PREMERE POI IL TASTO RETURN E TORNERETE AL LIVELLO DOS.**

AVETE DUE O PIÙ DRIVE

Avendo due o più drive vi risparmiate di togliere il disco col DOS-BASIC dal drive 0: infatti il disco da formattare lo metterete sul drive 1 e alla prima domanda (quale drive intendete usare) risponderete pigiando il tasto del numero 1.

Tutto si ripeterà come nel caso precedente e arrivati alla fine, il sistema torna da solo al livello DOS.

E con questo crediamo di avervi proprio detto tutto sulla formattazione.

B - DUPLICAZIONE DEI DISCHETTI

Vediamo ora un'altra operazione importante che è possibile fare col sistema operativo DOS: la DUPLICAZIONE dei dischetti.

Come già sapete, non si possono fare operazioni su dischetti vergini quindi per fare una duplicazione occorre disporre di un disco già formattato.

Parleremo innanzitutto del caso in cui abbiate un solo drive.

La partenza deve essere sempre fatta dal livello DOS (vedremo poi che tutte le operazioni DOS sono fattibili anche a livello BASIC).

Se vogliamo ad esempio fare una copia del disco DOS-BASIC, cosa che del resto è estremamente consigliabile in quanto il dischetto potrebbe rovinarsi per tanti motivi (quindi meglio averne una o due copie fedeli di scorta) procederemo come segue.

Prima di partire controllate che il disco DOS-BASIC abbia il nastro adesivo di protezione contro le registrazioni. In caso contrario, per una errata manovra, potreste cancellarlo.

A livello DOS scrivete quanto segue

COPY :0 TO :0 01/25/82

È IMPORTANTISSIMO LASCIARE GLI SPAZI INDICATI: cioè COPY (spazio): 0 (spazio) TO (spazio): 0 (spazio) 01/25/82 se non li mettete o ne inserite più di uno il computer segnala errore con

BAD FILESPEC (errore di procedura)

Anche i due punti sono importanti, ovviamente. Attenzione anche a non confondere lo zero con la lettera O. Segue poi la data, da dare nel modo che abbiamo già visto per la formattazione. Quella fornita vale come esempio quindi ne potrete scegliere una anche diversa.

Se avete scritto la data in modo errato, esempio 1/25/82 o 25/01/82 la segnalazione è

BAD DATE (errore di data)

quindi occorre riscriverla. Se tutto è corretto, apparirà il seguente messaggio

9 PRESS "ENTER" WHEN SOURCE DISKETTE MOUNTED ON DRIVE 0 (premere "ENTER" quando il disco sorgente è montato è nel drive 0)

ossia premere il tasto RETURN quando il disco sorgente è nel **drive 0**.

Dato che il disco da duplicare è quello del DOS-BASIC (quindi inserito nel DRIVE 0) potete premere il tasto RETURN. Il drive partirà e sul monitor leggerete

10 READING TRACK, SECTOR n,m (sto leggendo la traccia e il settore n,m)

Il computer sta leggendo il disco sorgente da copiare, segnalando coi numeri 'n' ed 'm' a quale punto sta leggendo. Vi accorgete che ad un certo momento il drive si fermerà e sul monitor apparirà

11 PRESS "ENTER" WHEN DEST. DISKETTE MOUNTED ON DRIVE 0 (premi "ENTER" quando il disco destinatario è montato nel drive 0)

A questo punto **togliete** il disco DOS-BASIC dal drive ed **inserite** il disco formattato su cui volete fare il duplicato; poi battete il tasto RETURN.

Dopo aver eseguito anche questa istruzione, il drive riparte scrivendo sul video

12 WRITING TRACK, SECTOR n,m (cioè scrivo le tracce e i settori n e m)

che dopo poco cambierà in

VERIFYING TRACK, SECTOR n,m (verifico le tracce e i settori n e m)

Quindi, analogamente a quanto visto per la formattazione, il sistema scrive e controlla che i dati registrati siano uguali a quelli immessi.

Quando questo processo è finito riappare la scritta riportata in (9); a questo punto occorre rimettere il disco sorgente DOS-BASIC nel drive e rifare tutte le operazioni viste man mano che le domande si ripetono e **questo per 4 volte consecutive**.

In pratica avviene che la grande massa di dati da copiare quando si dispone di un solo drive viene divisa e caricata in 4 blocchi separati, quindi ciclicamente dovremo ripetere le istruzioni (9), (10), (11) e (12) **per quattro volte consecutive**.

Se non ci sono stati errori, tutto termina con la scritta

13 FULL DISKETTE COPY DONE (la copia di tutto il disco è finita)
PRESS "ENTER" WHEN SYSTEM DISKETTE MOUNTED ON DRIVE 0 (premi "ENTER" quando il disco sistema è nel drive 0)

ossia rimettete il disco sistema DOS-BASIC nel drive 0, premete il tasto RETURN per tornare al livello DOS. Se volete sincerarvi che la copia duplicata sia perfetta potrete inserirla nel drive e utilizzare questa al posto dell'originale.

Premendo, quindi, il tasto RESET presente sulla scheda CPU e provando a ricaricare il BASIC tutto dovrebbe filare liscio.

Gli unici inconvenienti per non riuscire a portare a termine una duplicazione possono essere i seguenti: avete inserito nel drive un dischetto provvisto di protezione contro le registrazioni, oppure avete lasciato nel drive il disco DOS invece di sostituirlo con quello formattato: in questo caso il sistema prova a scrivere, ma non riuscendoci ve lo segnala

14 PRESS "ENTER" WHEN **SYSTEM DISKETTE MOUNTED ON DRIVE 0**

Dopo aver rimesso il disco DOS-BASIC nel drive, si leggerà

ILLEGAL FILE NAME (manovra illecita)
DOS READY

In questo caso, eliminate la causa d'errore, ricominciate da capo e tutto girerà come precedentemente descritto.

Esiste comunque un'ultima causa che può provocare una interruzione del processo di duplicazione: ricordate il caso in cui appariva la scritta di una traccia difettosa nella formattazione? (vedi paragrafo B = formattazione dei dischi vergini)

operazione 7). Bene; allora rammenterete che tale disco aveva la traccia 18 del settore 04 difettosa. Se quindi provate a fare una duplicazione usando come destinatario proprio quel disco difettoso, il sistema ad un certo punto comunica

PRESS "ENTER" WHEN **SYSTEM
DISKETTE MOUNTED ON DRIVE 0**

(premere "ENTER" quando il disco sistema è nel drive 0)

dopo aver soddisfatto alla richiesta, si legge

PARITY ERROR DURING READ

(errore di parità durante la lettura)

Il sistema rifiuta, come è giusto che sia, di fare un duplicato su di un dischetto che abbia qualche traccia fuori uso. Facciamo presente che un disco con tracce difettose lo potremo utilizzare ugualmente per qualsiasi altro uso che non sia una duplicazione.

SE AVETE DUE DRIVE

Se avete due drive il procedimento è più semplice e veloce. Infatti basta mettere il disco DOS-BASIC nel drive 0 e quello destinatario della copia nel drive 1; e scrivere semplicemente

COPY :0 TO :1 01/25/82

Nell'esempio noi vi abbiamo messo una data 01/25/82 ma è ovvio che voi potete inserire la data che preferite; importante è non dimenticare lo spazio da: COPY (spazio): 0 (spazio) TO (spazio): 1 (spazio) data.

Il computer vi farà la domanda (9), poi la (11) presentate precedentemente (dove ovviamente al posto dello zero scriverete il numero 1); a questo punto è sufficiente premere due volte consecutivamente il tasto RETURN per dare inizio alla copiatura. IN QUESTO CASO ESSA AVVIENE IN UN'UNICA PASSATA.

Dopo circa sei minuti la copia è pronta.

Ovviamente è possibile anche effettuare delle duplicazioni di dischi che non siano quello DOS-BASIC. In questo caso bisogna prima portarsi al livello DOS, poi introdurre il comando COPY con le stesse modalità alla domanda (9).

Togliere dal drive 0 il disco sistema e mettere quello da duplicare e procedere normalmente come già spiegato. Naturalmente a duplicazione avvenuta il disco DOS-BASIC va rimesso nel drive 0.

Una curiosità: operando con un solo drive è possibile fare la duplicazione di un disco sullo stesso disco (a patto che non abbia il nastro adesivo di protezione). La cosa ovviamente non ha molto senso, però serve a farvi capire la «potenza» di questo sistema operativo.

Si possono anche copiare solo programmi o archivi, senza dover fare obbligatoriamente la duplicazione di tutto un disco. Per motivi di spazio affronteremo questo problema nel prossimo numero.

3 - NOZIONI PRELIMINARI SUL BASIC

A - INTRODUZIONE

Il dischetto da noi fornito contiene il DOS e il BASIC V1.0. Dopo aver effettuato il caricamento come è già stato spiegato in precedenza, il sistema può essere usato in 5 modi diversi:

- 1 - Comandi diretti
- 2 - Esecuzione di un programma
- 3 - Correzione di un programma
- 4 - Livello monitor
- 5 - Livello DOS

Vediamoli brevemente uno per uno.

ESECUZIONE DI COMANDI DIRETTI

A questo livello il computer esegue tutti i comandi introdotti da tastiera, appena questi li convalideremo premendo il tasto RETURN.

Quando sul monitor compare la scritta

READY

>

allora potete operare in COMANDI DIRETTI.

ESECUZIONE DI UN PROGRAMMA

Questo livello è reso operativo dal comando RUN: il programma presente nella memoria del computer diventa esecutivo.

CORREZIONE DI UN PROGRAMMA

Operando a questo livello è possibile fare l'EDITING, ossia modificare un programma preesistente.

LIVELLO MONITOR

In questo modo è possibile caricare in memoria dei programmi in linguaggio macchina (OBJECT FILES).

LIVELLO DOS

Questo è il livello del SISTEMA OPERATIVO, che rende possibile l'uso dei Floppy Disks. Come già detto, questo livello può essere raggiunto in via diretta (uscendo dal BASIC) o attraverso il BASIC. Vedremo l'importanza di tutto questo.

B - CONCETTI FONDAMENTALI

Iniziamo con una descrizione sommaria ma completa del BASIC, in modo che possiate già cominciare a programmare.

Nei prossimi numeri della rivista troverete una descrizione dettagliatissima di ognuna delle istruzioni BASIC e DOS, con molti esempi pratici e di difficoltà graduata.

Prima di proseguire occorre avere ben chiari alcuni concetti basilari. Le righe che seguono sono di estrema importanza per chi non conosce i metodi di programmazione in BASIC, quindi vanno seguite con molta attenzione, esempi compresi. Per sincerarvi di aver ben capito, provate a fare degli esempi vostri sullo schema di quelli qui forniti.

NOTA BENE - In tutti gli esempi che seguiranno si presuppone quanto segue:

- BASIC già caricato
- Cancellazione di programmi eventualmente già esistenti (scrivere NEW e il tasto RETURN).

PAROLE CHIAVE

Le PAROLE CHIAVE sono i comandi che il computer è in grado di riconoscere. L'elenco completo delle PAROLE CHIAVE è nelle TABELLE 2-3-4-5-6-7 riportate in prima pagina.

Usando nelle istruzioni parole diverse da quelle chiave il computer segnala errore.

VARIABILI NUMERICHE

I nomi delle variabili devono cominciare con una lettera (da A a Z); possono essere formate da un solo carattere (che quindi deve essere obbligatoriamente una lettera) o da due o più caratteri. Nel caso che si usino variabili con più di un carattere, ricordatevi che i caratteri dal secondo in poi possono essere anche numeri. Ricordatevi pure che il computer accetta definizioni di variabili con più di due caratteri, ma tiene conto solo dei primi due.

Ecco qualche esempio di variabili numeriche ammesse:

- 1 C = 32
- 2 A = C*28-3.4
- 3 S3 = K⁴/2
- 4 X0 = 3.1415
- 5 RO = 7 + (A*2.3-C)²/5
- 6 ROMANO = 7
- 7 ANNO = 1982
- 8 ROVIGO = 4.78
- 9 S3712 = 8

I simboli che non sono chiari saranno spiegati tra breve. Attenzione a non confondere il punto con la virgola: nella notazione americana non si scrive 3,4 bensì 3.4.

Per controllare gli esempi o per provare a farne di diversi basta fare come segue:

scrivere NEW e il tasto RETURN, poi scrivere un esempio e nuovamente RETURN; successivamente digitare PRINT e il nome assegnato alla variabile, e poi RETURN.

Immediatamente il computer stamperà il valore di quella variabile.

Provare quindi a scrivere l'esempio 1 facendolo seguire da RETURN; scrivete poi PRINTC (sempre seguito da RETURN). Sul monitor compare il numero 32, come era stato assegnato.

Anziché digitare PRINT si può ottenere lo stesso risultato pigiando in sua vece il punto interrogativo.

Notare che, come detto, le variabili degli esempi 5-6-8 in effetti per il computer hanno lo stesso nome 'RO' in quanto i caratteri oltre il secondo vengono ignorati. Le stesse considerazioni valgono per gli esempi 3 e 9.

Ecco ora qualche esempio di definizioni di variabili numeriche inefficaci o non ammesse:

- 10 3D = 63
- 11 8 = 2*Z
- 12 L. = 3/4
- 13 IF = 2
- 14 STORIA = 392

Se provate a controllare gli esempi 10 e 11 vedrete che in effetti quando date il comando PRINT3D oppure PRINT8 il computer vi darà rispettivamente i valori 3 e 8. È evidente quindi che non avete definito delle variabili.

Gli esempi 13 e 14 sono illeciti perché contengono istruzioni BASIC (IF la 13, TO e OR la 14).

In questo caso il monitor segnala

SYNTAX ERROR IN 15359 (errore di sintassi in 15359)

READY

UNDEFINED LINE IN 15359 (linea non specificata in 15359)

La linea 15359 la vedrete stampata spesso, quando commettete degli errori. Il numero fa riferimento al BASIC scritto in linguaggio macchina, quindi non ha nessuna relazione con ciò che state facendo.

Le variabili numeriche possono essere di tre tipi: INTERE - A SINGOLA PRECISIONE - A DOPPIA PRECISIONE.

Quelle INTERE sono seguite dal simbolo '%', quelle a SINGOLA PRECISIONE dal simbolo '!', quelle a DOPPIA PRECISIONE dal simbolo '#'.
La differenza tra variabili a SINGOLA e DOPPIA precisione consiste nel fatto che le prime sono formate da sei cifre significative e le seconde da 16 cifre significative.

Qualche esempio:

```
15 W% = 96
16 D416! = .123456
17 Q# = 3.141592653589
18 P = 16
19 P% = 1729
20 P! = 34.72
21 P# = 23.45289342
```

Negli esempi dal 18 al 23 siamo in presenza di 4 variabili ben distinte, trattate come tali dal computer nonostante inizino tutte con la stessa lettera.

VARIABILI ALFANUMERICHE

Per VARIABILI ALFANUMERICHE (dette più spesso STRINGHE) si intende un insieme di caratteri qualsiasi: lettere, numeri e simboli grafici in genere.

Si definiscono con gli stessi criteri dati per le variabili numeriche, tranne il fatto che sono contraddistinte dal simbolo '\$'. Ogni variabile alfanumerica può contenere al massimo 255 caratteri.

Qualche esempio:

```
22 A$ = "PERA"
23 CN$ = "CONSEGNA A DOMICILIO SENZA SPESE"
24 SOMMA$ = "IL RISULTATO FINALE È:"
25 F1$ = "7*A + C ="
```

Come si vede, dopo il segno '=' bisogna mettere le virgolette, altrimenti il computer segnala errore.

OPERATORI ARITMETICI

Gli OPERATORI ARITMETICI servono per eseguire i calcoli. I simboli da usare per ottenere un'operazione matematica sono i seguenti:

```
+ SOMMA
- SOTTRAZIONE
* MOLTIPLICAZIONE
/ DIVISIONE
↑ ELEVAMENTO A POTENZA.
```

OPERATORI RELAZIONALI O DI CONFRONTO

Gli OPERATORI RELAZIONALI servono quando in un programma si devono prendere decisioni in base a dei confronti. Essi sono:

```
< (MINORE DI)
> (MAGGIORE DI)
<> (DIVERSO DA)
<= (MINORE O UGUALE A)
>= (MAGGIORE O UGUALE A)
= (UGUALE A).
```

Esempio:

```
IF A > 56 PRINT "A MAGGIORE DI 56"
```

In questo caso, se la condizione $A > 56$ è verificata, il computer stampa la frase tra virgolette; in caso contrario passa alla riga di istruzione successiva.

OPERATORI LOGICI

Gli OPERATORI LOGICI accettati dal computer sono: OR, AND, NOT.

Assieme alle parole chiave IF, THEN, ELSE (che vedremo più avanti) permettono di fare confronti logici anche molto complessi e potenti.

OPERATORI DI STRINGA

Gli OPERATORI DI STRINGA sono:

- < (MINORE DI)
- > (MAGGIORE DI)
- <> (DIVERSO DA)
- <= (MINORE O UGUALE A)
- >= (MAGGIORE O UGUALE A)
- = (UGUALE A)
- + (SOMMA)

I primi sei servono per ordinare alfabeticamente le stringhe, mentre l'ultimo serve per sommarle. Vedremo esempi esplicativi più avanti.

SUNTO DELLE ISTRUZIONI BASIC

Riportiamo qui di seguito una descrizione succinta del BASIC V1.0 contenuto nel disco, in modo da permettere a tutti di poter utilizzare i vari comandi e provare le istruzioni. In futuro descriveremo approfonditamente tutte queste voci, in un volume riservato al solo computer.

Prima di proseguire rammentiamo quanto esposto al paragrafo A del capitolo 3: occorre capire bene qual'è la differenza tra i due concetti di COMANDI DIRETTI ed ESECUZIONE DI UN PROGRAMMA. A questi due modi operativi si collegano le definizioni di COMANDO e di ISTRUZIONE:

un COMANDO è impartito direttamente dalla tastiera, e diventa esecutivo appena si preme il tasto RETURN; una ISTRUZIONE è una parola chiave inserita in un programma, e che agisce solo in sede di esecuzione del programma stesso.

La distinzione tra COMANDI e ISTRUZIONI è generalmente nettissima nel senso che provando ad utilizzare una parola di istruzione come se fosse un comando, il computer segnala errore (e viceversa). In effetti, però, molte istruzioni sono accettate e diventano operative anche se usate come comandi, e analogamente alcuni comandi sono operativi anche inseriti in una linea di programmazione. Affronteremo comunque questo aspetto caso per caso, in sede di spiegazione più dettagliata delle parole chiave.

Basti un solo esempio di quanto abbiamo appena detto: la parola chiave PRINT è sia COMANDO che ISTRUZIONE, vale a dire che diventa esecutiva sia introducendola direttamente da tastiera che inserendola all'interno di un programma.

Proprio per puntualizzare la differenza, almeno concettuale, tra COMANDO ed ISTRUZIONE, abbiamo adottato il seguente metodo: negli esempi che accompagnano le descrizioni delle parole chiave il simbolo > sta ad indicare che si tratta di un COMANDO, mentre se ci sono numeri di linea (ovviamente solo esemplificativi) vuol dire che la parola in questione rappresenta una ISTRUZIONE.

Esempio:

> CONT (comando)

60 NEXT (istruzione)

Ed ora passiamo pure ad illustrare in breve le parole chiave del BASIC, limitandoci per questa volta alla TABELLA 2 che è la più importante.

IL CONTENUTO DI EVENTUALI PARENTESI VICINO ALLE PAROLE CHIAVE RAPPRESENTA IL MODO IN CUI È POSSIBILE ABBREVIARE LE PAROLE STESSE.

AND

40 IF C = 3 AND H2 > B GOTO 100

Se entrambe le condizioni poste sono verificate, l'esecuzione del programma passa alla linea 100, altrimenti va alla riga successiva alla 40.

AUTO

- > AUTO Genera numeri di riga che cominciano da 10 e aumentano con passo 10
- > AUTO ,100 Genera numeri di riga che cominciano da 0 e aumentano con passo 100
- > AUTO 1250, 35 Genera numeri di riga che cominciano da 1250 e aumentano con passo 35

L'Auto viene annullato premendo contemporaneamente i tasti CTRL-A.

CLEAR

- 10 CLEAR Azzera tutte le variabili numeriche e di stringa; riserva spazio per 50 caratteri di stringa
120 CLEAR 2000 Azzera e riserva spazio per 2000 caratteri di stringa

CLOSE

- 80 CLOSE Chiude l'accesso a tutti i FILES
240 CLOSE 2,7,11 Chiude l'accesso solo ai FILES 2,7,11

CMD "D"

- > CMD"D" Fa eseguire a livello BASIC il programma di DEBUG.

CMD "funzione DOS"

- > CMD"funzione DOS" Rende possibile eseguire a LIVELLO BASIC tutte le operazioni eseguibili a LIVELLO DOS. Tra le virgolette si può mettere uno qualsiasi dei comandi DOS. Alla fine si ha il ritorno automatico al LIVELLO BASIC.

CMD "S"

- > CMD"S" Genera il passaggio dal LIVELLO BASIC al LIVELLO DOS, con perdita del BASIC e di eventuali programmi presenti in memoria centrale.

CONT

- > CONT Permette la prosecuzione del programma dopo una interruzione causata da un BREAK (tastiera) o da uno STOP (programma).

DATA

- 50 DATA 318,8.23,"CAMPER",71 Memorizza dei dati all'interno del programma. Questi dati sono accessibili esclusivamente in modo SEQUENZIALE con l'istruzione READ (vedere anche RESTORE).

DEFDBL

- 75 DEFDBL C-H Dichiara che tutte le variabili che iniziano con le lettere da C ad H saranno trattate e memorizzate in DOPPIA PRECISIONE
30 DEFDBL Q.F Idem per le lettere Q e F. NOTA: si può usare indifferentemente sia il trattino che la virgola.

I simboli % e ! annullano questa istruzione.

DEFFN

- 10 DEFFN A(X,Y) = X + 3*Y Definisce la funzione di A e di X e Y come il risultato di $X + 3*Y$
25 DEFFN C\$(A\$,I) = RIGHT\$(A\$,I) Definisce la funzione C\$ di A\$ e I come una stringa di I caratteri presi a destra della stringa C\$

DEFINT

- 20 DEFINT A-H Dichiara che tutte le variabili che cominciano con le lettere da A ad H saranno trattate e memorizzate come INTERI, eccettuato quelle che recano i simboli ! e #

DEFSNG

- 300 DEFSNG E-L,R Dichiara che tutte le variabili che cominciano con le lettere da E a L e con la lettera R saranno trattate e memorizzate in SINGOLA PRECISIONE, escluse quelle coi simboli % e #

DEFSTR

- 10 DEFSTR A,C Le variabili che iniziano con le lettere indicate sono trattate e memorizzate come STRINGHE, ad eccezione di quella coi simboli % ! e #

DEFUSR

- 120 DEFUSR 2 = &H33CD Definisce la ROUTINE espressa in LINGUAGGIO MACCHINA, memorizzata a partire dalla locazione 33CD in esadecimale. Le ROUTINES possibili sono 10 (0-9).

DELETE (D)

- > DELETE 40 Cancella la linea 40
- > DELETE 120-200 Cancella le linee dalla 120 alla 200
- > D -300 Cancella le linee dall'inizio fino alla 300
- > D. Cancella l'ultima linea generata o editata

DIM

- 10 DIM F(23) Definisce le variabili trattate come VETTORI ad una o più DIMENSIONI: F è definito come VETTORE ad una dimensione formato da 24 elementi (0-23)
- 10 DIM A S(12,58), C(6,14,E + 4,B*2,78) A S è una MATRICE a 2 dimensioni di 767 elementi, C è una MATRICE a 5 dimensioni alcune delle quali sono definite da variabili. Senza una DIM, un vettore può essere al massimo di 11 elementi.

EDIT (E)

- > EDIT 340 Fa passare dal LIVELLO DI COMANDI DIRETTI al LIVELLO DI EDITING (correzione) della linea 340
- > E 30 Fa passare dal LIVELLO DI COMANDI DIRETTI al LIVELLO DI EDITING (correzione) della linea 30

END

- 1200 END Pone termine all'esecuzione del programma. Non è indispensabile.

EOF

- 50 IF EOF(1) THEN CLOSE 1 In lettura di un FILE DATI (BUFFER 1) controlla se si è giunti alla fine del FILE; in tal caso esegue la CLOSE, altrimenti prosegue alla linea successiva.

ERL

- 90 PRINT ERL ERL è una variabile speciale riservata al sistema; assume il valore del numero di linea dove si è verificato un errore. Generalmente si usa in coppia con l'istruzione ONERRORGOTO.

ERR

- 260 PRINT ERR/2 + 1 Anche ERR è una variabile speciale riservata al sistema. Il suo valore è legato al codice dell'errore che si è verificato; la formuleta dell'esempio da' appunto il valore del codice. Normalmente si usa accoppiato con ONERRORGOTO, per ROUTINES di trattamento degli errori.

ERROR

- 60 ERROR 1 Alla linea 60 il computer si comporta come se si fosse veramente verificato l'errore di tipo 1. Serve per fare dei test sulla routine ONERRORGOTO.

FIELD

720 FIELD 3, 15 AS D\$, 10 AS CC\$, 112 AS A\$ Suddivide in CAMPI il BUFFER di un FILE RANDOM. Nell'esempio il BUFFER 3 (di 255 caratteri) è stato diviso in CAMPI in questo modo: 15 bytes per D\$, 10 per CC\$, 112 per A\$; gli ultimi 118 bytes non sono stati definiti.

FOR --- NEXT

200 FOR I = 1 TO 20 STEP 4 Queste due istruzioni danno origine ad un LOOP tra le due linee in cui compaiono.
210 PRINT A + 1 Nell'es. tutto ciò che si trova tra le linee 200 e 250 viene eseguito 5 volte. Il valore
250 NEXT I iniziale, quello finale ed il passo possono essere espressi con variabili o espres-
30 FOR E2 = A TO B*3 + C sioni (Esempio compreso tra le linee 30 e 60)
50 LPRINT "BOLOGNA"
60 NEXT

FRE

> PRINT FRE(0) È equivalente al comando MEM: dà le cellule di memoria ancora disponibili. Il numero tra parentesi può essere qualsiasi.

GET

520 GET 7,4 Serve per leggere da disco il RECORD 4 del FILE RANDOM 7. Se vengono omessi virgola e secondo numero, si ha la lettura del RECORD successivo all'ultimo letto (del primo, se non è stato letto ancora nulla).

GOSUB

90 GOSUB 8000 Causa l'interruzione del programma principale per saltare alla SUBROUTINE che inizia alla linea 8000 (vedere anche RETURN).

GOTO

15 GOTO 210 Il programma dalla linea 15 passa alla linea 210 (SALTO INCONDIZIONATO).

IF --- THEN --- ELSE

30 IF A > 34 THEN CE = 5 ELSE GOTO 100 IF permette di fare un TEST su un'espressione logica o relazionale.
30 IF A > 34 CE = 5 ELSE 100 Se il risultato del test è vero, il controllo del programma prosegue
70 IF Z THEN 340 nella linea; in caso contrario il programma passa all'istruzione ELSE
70 IF Z GOTO 340 (se esiste) o alla riga di programma successiva. Gli esempi col nu-
110 IF Q3 ELSE 315 mero di linea 30 sono equivalenti, come pure quelli numerati 70. Nel-
l'esempio 110 se il valore di Q3 è <>0 il programma prosegue con la
linea successiva, altrimenti va alla linea 315. Le istruzioni con gli stessi
numeri di linea sono equivalenti.

INKEY \$

40 AS = INKEY\$: IF A\$ = "" GOTO 40 L'istruzione INKEY \$ predisporre il computer ad accettare una stringa di un solo carattere (A\$ nell'es.) da tastiera. Se durante l'esecuzione dell'istruzione non si preme alcun tasto, il computer assegna automaticamente ad A\$ una stringa nulla (""), altrimenti assegna il carattere digitato. Nell'esempio riportato il programma si arresta alla linea 40 fino a che non si preme un tasto qualsiasi.

INPUT

50 INPUT C,D\$ Il computer si ferma alla linea 50 e, dopo aver visualizzato un punto interrogativo, resta in attesa che venga introdotto da tastiera un numero che verrà assegnato alla variabile C; subito dopo il computer visualizza ancora? e resta in attesa di una stringa da assegnare alla variabile D\$

30 INPUT "NOME DEL CLIENTE = ";NC Ha un effetto analogo al precedente, con la differenza che il computer, prima del punto interrogativo, visualizza la frase tra virgolette.

INPUT

230 INPUT # 12, A, E\$ Nel trattamento di FILES SEQUENZIALI abilita il computer a leggere dati. Nell'esempio, dal FILE gestito dal BUFFER 12 il computer introita prima un dato numerico che assegna alla variabile A, poi una stringa che va alla variabile E\$.

INSTR

60 P = INSTR (7,A\$, "MANO") È una funzione di ricerca di una stringa all'interno di un'altra. Nell'esempio il computer va a vedere se ad iniziare dal settimo carattere della stringa A\$ c'è la stringa MANO. In caso affermativo la variabile P assume il valore 7, altrimenti il valore 0. Se non si mette il carattere d'inizio, il computer parte sempre dal primo.

KILL

> **KILL "CLIENTI"** Serve per cancellare dal disco il FILE che porta il nome specificato (CLIENTI).

LET

20 LET D = 56 Serve ad assegnare i valori alle variabili.
30 D = 56 Si può anche scrivere omettendo LET

LINEINPUT

85 LINEINPUT "TITOLO DEL FILM"; TF Simile all'istruzione INPUT, con la differenza che manca la visualizzazione del punto interrogativo ed accetta anche le virgole (le virgolette sono accettate da entrambe).

LINEINPUT

55 LINEINPUT # 9,A\$ Serve per leggere una riga di testo da disco. Es.: dal FILE del BUFFER 9 legge la prima riga di dati e la pone nella variabile A\$.

LIST (L)

> **LIST** Visualizza tutto il programma
> **LIST 30** Visualizza solo la linea 30
> **L300-400** Visualizza tutte le linee tra la 300 e la 400
> **L120-** Visualizza tutte le linee dalla 120 alla fine
> **L.** Visualizza l'ultima linea generata od editata (anche battendo solo il punto si ottiene lo stesso risultato)

LLIST

> **LLIST** Fa il listato del programma sulla stampante. Vale la casistica di LIST. (Non sono ammesse però le abbreviazioni)

LOAD

> **LOAD "PAOLO"** Carica da disco il programma il cui nome è quello specificato
> **LOAD "PAOLO",R** Nell'esempio 2, dopo il caricamento, si ha anche l'esecuzione automatica del programma caricato.

LOC

40 PRINT LOC (3) Nel trattamento dei FILES RANDOM serve per visualizzare il numero di RECORD che è stato letto per l'ultima volta con l'istruzione GET. Da' quindi la POSIZIONE ATTUALE del PUNTA-TORE

LOF

50 FOR I = 1 TO LOF (3) : INPUT #3,AS : NEXT I

Serve ad individuare la fine di un FILE DATI; LOF assume il valore del numero d'ordine dell'ultimo RECORD del FILE. Nell'esempio, dopo aver aperto il FILE DATI 3, è possibile con il LOOP indicato leggere tutti i dati del FILE anche senza conoscerne il numero totale.

LPRINT

30 LPRINT G

Serve per fare stampare dati alla stampante (nell'es. si stampa il valore di G). Vale la casistica prevista in PRINT.

LSET

430 LSET FS = "NUOVA ELETTRONICA"

Dopo un' assegnazione di FIELD, serve per mettere la stringa indicata nel CAMPO assegnatole, allineandola a SINISTRA.

MEM

> PRINT MEM

120 IF MEM > 1000 END

Nella variabile speciale MEM viene depositato automaticamente il numero di bytes di memoria ancora liberi. Gli esempi mostrano che è possibile un suo uso sia come comando che nelle istruzioni.

MERGE

> MERGE "CLIENTI"

Serve per fondere il programma in memoria centrale col programma indicato su disco (possibile solo se quest'ultimo è in FORMATO ASCII; vedere anche SAVE).

NEW

> NEW

Cancella tutte le righe di programma ed azzerava tutte le variabili.

NOT

30 IF NOT A GOTO 100

470 IF NOT D > K + 3 * F THEN 500

Se A vale zero il programma va alla linea 100
Se D non è maggiore dell'espressione assegnata, il programma prosegue alla linea 500.

ON --- GOSUB

45 ON B GOSUB 7000,8000,9000

Nel caso di B = 1 l'esecuzione del programma salta alla SUBROUTINE che inizia col numero di linea 7000; se B = 2 va alla SUBROUTINE 8000; se B = 3 va alla 9000. Al posto di B può anche esserci un'espressione. Se il valore di B non è compreso tra 1 e 3 il programma va alla linea successiva. **Se B < 0 il computer segnala errore.**

ON --- GOTO

100 ON A1*2.5 + S-3 GOTO 120,200,300,450

Funzionamento analogo all'istruzione precedente, tranne il fatto che le linee di salto non corrispondono a SUBROUTINES.

ONERRORGOTO

110 ONERRORGOTO 1000

Se si verifica un errore quando il programma è oltre la linea 110, si va alla linea 1000 (vedere anche RESUME).

OPEN

100 OPEN "E",2,"LISTA" Con questa istruzione si aprono FILES DATI sia SEQUENZIALI (SEQUENTIAL) che CASUALI (RANDOM). La lettera tra virgole imposta il metodo di accesso (I-O-E-R); il numero assegna il BUFFER al FILE specificato tra le virgolette successive.

OR

40 IF C = 3 OR H2 > B THEN 100 Se una sola delle condizioni poste (o anche entrambe) è verificata, l'esecuzione del programma passa alla linea 100, altrimenti va alla riga successiva alla 40.

PRINT (?)

30 PRINT A Stampa sul video la variabile A
70 ? "PARTITA IVA" Stampa sul video PARTITA IVA
50 PRINT Sul video il puntatore avanza di una riga

PRINT # (? #)

80 PRINT #12,A,E\$ Nel trattamento di FILES SEQUENZIALI abilita il computer a registrare dati sul FILE gestito dal BUFFER 12 (prima il numero contenuto nella variabile A e poi la stringa E\$).

PRINT @ (?@)

30 PRINT @ 330, "NUOVA ELETTRONICA" Ha funzioni analoghe a quelle di PRINT, con la differenza che visualizza quanto specificato a cominciare dalla posizione indicata dal numero prima della virgola. Questo numero deve essere compreso tra 0 (primo carattere in alto a sinistra nel monitor) e 511 (ultimo carattere in basso a destra).

PRINTTAB (?TAB)

70 ?TAB (13) "NUOVA ELETTRONICA" Consente di visualizzare quanto richiesto nella posizione di riga specificata dal numero tra parentesi (vedi 13). Questo può assumere i valori da 0 a 255. Serve per incolonnare dati sul monitor. Se il numero supera 31 (lunghezza di una riga), verranno utilizzate le righe successive.

PRINTUSING (?USING)

300 PRINTUSING " ##. ### ";U Visualizza il valore della variabile U nel FORMATO specificato dalla stringa tra virgolette (al suo posto si può richiamare una variabile di stringa definita in precedenza).

PUT

340 PUT 2,16 Scrive su disco il RECORD 16 nel FILE RANDOM 2. Deve essere preceduta dalle dichiarazioni FIELD e RSET (o LSET).

READ

20 READ X,D Legge sequenzialmente i dati contenuti nella istruzione DATA e li assegna alle variabili X e D (vedere anche RESTORE).

REF

> REF * È un comando che permette di avere dei RIFERIMENTI. Da' sul monitor l'elenco completo dei RIFERIMENTI (con \$ al posto di * l'elenco viene anche stampato)
> REF DE Da' i RIFERIMENTI della variabile DE
> REF *DE Da' i RIFERIMENTI di tutte le variabili, a cominciare da DE
> REF 2350 Da' i RIFERIMENTI del numero intero 2350. Serve per trovare i numeri di linea dove compaiono i numeri interi richiesti.

REM (')

200 REM --- ROUTINE CALCOLO TRAVE --- Serve per inserire REMARKS (COMMENTI) lungo il programma e sono da questo ignorati.

RENUM

> RENUM, RINUMERA il programma. Rinumera partendo dal numero di linea 10 e con passo 10
> RENUM 1000, Rinumera dal 1000 con passo 10
> RENUM 1000,100 Rinumera dal 1000 con passo 100.

Se ci sono errori nei rimandi del programma, non effettua la RINUMERAZIONE e segnala il numero di linea inesistente.

RESTORE

45 RESTORE Permette alla successiva istruzione READ di leggere il primo dato della prima istruzione DATA.

RESUME

12000 RESUME Va inserita nella ROUTINE di gestione degli errori (ONERRORGOTO) e serve ad indicare da quale linea deve riprendere il programma dopo che si è verificato l'errore.

Se RESUME non è seguito da numero, si torna alla linea dell'errore, altrimenti si va alla linea specificata. Se RESUME è seguito da NEXT, il programma prosegue alla linea successiva a quella d'errore.

RETURN

730 RETURN Chiude l'esecuzione di una SUBROUTINE e fa tornare il programma all'istruzione successiva alla relativa GOSUB.

REVOFF

100 REVOFF Annulla la visualizzazione in REVERSE (NEGATIVO) (vedere anche REVON).

RECON

60 RECON Inserisce la scrittura su monitor in REVERSE (NEGATIVO) (vedere anche REVOFF).

RSET

170 RSET D3 \$ = "CLIENTI IN MORA" Analoga alla LSET: serve per mettere la stringa indicata nel FIELD assegnate, allineandola a DESTRA.

RUN

> RUN Da' inizio all'esecuzione del programma cominciando dal numero di linea più basso; se dopo RUN si mette un numero, il programma parte dalla linea indicata
3450 RUN "MOSTRA" Con questa istruzione si ottiene il CONCATENAMENTO del programma in corso con quello richiamato dal disco.

SAVE

> SAVE "MOSTRA",A Salva il programma presente in memoria su disco, assegnandogli il nome specificato. Dando anche la parte dopo le virgolette, si ha la registrazione in FORMATO ASCII, altrimenti essa avviene in FORMATO COMPATTATO.

STOP

250 STOP Interrompe l'esecuzione del programma e fa tornare a livello di COMANDI DIRETTI (vedere anche CONT).

SYSTEM

- > **SYSTEM** Porta il computer a LIVELLO MONITOR, permettendo così di passare al LINGUAGGIO MACCHINA.

TROFF

- > **TROFF** Elimina la funzione TRACE innescata dal comando TRON.

TRON

- > **TRON** Dà inizio alla funzione TRACE. Dopo averlo introdotto, pigiare RUN: il programma parte e sul monitor appare il FLUSSO completo del programma, col numero di riga via via eseguito visualizzato tra parentesi. Serve per analizzare l'esecuzione del programma.

DESCRIZIONE DELLE ISTRUZIONI BASIC

Come avete visto, i comandi e le funzioni BASIC da illustrare dettagliatamente sono in numero considerevole.

Inoltre per un po' di tempo sarete impegnati a mettere in pratica le istruzioni che vi abbiamo dato finora. Non vorremmo poi riempirvi la testa con troppe nozioni date tutte assieme: fareste solo una gran confusione e non riuscireste ad assimilarle e a padroneggiarle veramente. D'altra parte lo spazio già occupato da queste prime note sull'utilizzo del DOS-BASIC è notevole.

Per tutti questi motivi, di tutte le istruzioni BASIC, faremo ora un rapido cenno solo di quelle due o tre che vi permetteranno di fare i primi programmi. Vi diremo come registrarli su disco e come, successivamente, richiamarli ed eseguirli anche a distanza di tempo.

Ci preme innanzitutto fare una considerazione di carattere generale: nell'uso del BASIC gli spazi non sono necessari. Facciamo un esempio:

PRINT 23 * 102

Questo è un comando diretto che serve per vedere sul monitor il risultato della moltiplicazione di 23 per 102. Ebbene, se scrivete PRINT23*102 raggiungete lo stesso scopo.

Quanto detto vale anche all'interno delle righe di programmazione: per poter leggere più agevolmente un programma dopo averlo listato sul monitor o sulla stampante vi conviene staccare le parole le une dalle altre; ricordate però che in questo modo occupate uno spazio maggiore nella memoria RAM del computer. Sappiatevi quindi regolare: se fate un programma molto lungo, eliminando gli spazi guadagnerete in memoria.

Detto questo, cominciamo pure con il comando:

MEM

DOPO AVER CARICATO IL BASIC provate a digitare PRINT MEM (potete anche digitare ?MEM, che è lo stesso, se ricordate) e vedrete visualizzato sul monitor un numero che rappresenta le celle di memoria RAM che avete a disposizione per i vostri programmi. Da tenere presente che il DOS-BASIC occupa circa 26K BYTES; il numero che vedete è la differenza tra la memoria RAM totale che avete nelle varie schede (statiche o dinamiche che esse siano) e la memoria occupata dal DOS-BASIC.

SAVE

Dopo aver fatto un programma, se spegnete il computer sapete già che lo perdetevi irrimediabilmente. Per conservarlo utilizzate questo comando nel seguente modo: dovete dare un nome a quel programma (per esempio "FATTURE") e salvarlo su disco digitando

SAVE "FATTURE"

Per abbreviare al massimo potete anche digitare **SAVE" FATTURE"** (eliminando cioè lo spazio e le virgolette finali); otterrete lo stesso risultato.

Appena avrete premuto il tasto RETURN il drive partirà e registrerà sul disco il vostro programma. Il LED acceso segnala che sta avvenendo uno scambio di dati tra computer e floppy disk.

NON DATE UN COMANDO DI SAVE SE IL DISCO È PROTETTO CONTRO LE REGISTRAZIONI: potreste perdere il controllo del BASIC e con esso anche il vostro programma!

Normalmente questo non succede, e le rare volte che ci è capitato era dovuto ad un non perfetto funzionamento di qualche cella di memoria.

Se tutto è regolare, provando a fare la manovra che vi abbiamo sconsigliato vedrete la scritta

UN PRINTABLE ERROR IN 15359

Ad ogni buon conto controllate preventivamente qual'è il comportamento del vostro computer in questo frangente tanto importante, e se del caso fate un attento controllo delle schede di memoria. Nel nostro computer è bastato sostituire un integrato del tipo 4116 in una scheda di memoria dinamica da 16K, ed il difetto è scomparso.

Potete anche fare in questo modo: mettete nel drive un duplicato del DOS-BASIC senza l'adesivo di protezione (così se vi capita di rovinarlo avete sempre l'originale per farvene un'altra copia, come vi abbiamo già consigliato); la registrazione non è mai interdetta e l'inconveniente non può verificarsi.

Ricordate poi un'altra cosa molto importante: SE ESISTE GIÀ SUL DISCO UN PROGRAMMA CON LO STESSO NOME, ESSO VIENE CANCELLATO E SOSTITUITO DAL NUOVO. Fate quindi molta attenzione.

Vedremo in seguito che è possibile (con il comando **CMD"DIR"**) sapere i nomi dei programmi e degli archivi presenti su di un dischetto; potrete così evitare l'inconveniente appena menzionato.

LOAD

Con questo comando potete richiamare dal disco i vostri programmi e successivamente eseguirli col comando RUN. Il formato del comando è il seguente:

LOAD "FATTURE"

dove il nome tra virgolette richiama l'esempio precedente. Se provate a richiamare un programma con un nome diverso da quello con cui lo avevate salvato, il computer ovviamente non lo trova e vi segnala il fatto con

FILE NOT FOUND IN 15359

Quindi, almeno fino a che non saprete richiamare l'indice dei nomi assegnati ai programmi, segnatevi in un foglio di carta e conservatelo; vi servirà senz'altro!

Anche in questo comando si possono omettere lo spazio e le virgolette finali.

Un'ultima cosa, che interessa in modo particolare chi è in possesso di più di un drive. Se fate ad esempio in questo modo

SAVE "FATTURE":1

il computer va ad incidere il programma sul drive 1. Come sempre, potete anche digitare **SAVE" FATTURE:1** ed otterrete lo stesso risultato.

Analogamente, per il comando LOAD, aggiungendo i due punti ed un numero (che ovviamente deve essere compreso tra 0 e 3) date istruzione al sistema di mettersi in comunicazione col drive specificato.

Anche in questo caso è prevista una segnalazione di errore se la manovra è sbagliata: infatti provando a chiedere il collegamento con un drive che non c'è, il monitor visualizza

TOO MANY FILES IN 15359

Non vi resta che controllare il tutto per capirne bene il meccanismo d'uso.

RACCOMANDAZIONI IMPORTANTI

Prima di chiudere, diamo qualche importante raccomandazione.

NON ACCENDETE NÉ SPEGNETE MAI IL COMPUTER COI DISCHETTI INSERITI NEI DRIVE. Potrebbero subire delle alterazioni e non dare più i risultati ottimali. Prendete quindi la buona abitudine di estrarli prima di spegnere e di inserirli dopo aver acceso il computer e premuto il tasto RESET della scheda CPU.

NON FATE MAI UN BREAK QUANDO I DRIVE HANNO I LED ACCESI. Perderete quasi sempre il controllo del sistema, con perdita del BASIC, del DOS e di tutto quello che si trova nella memoria centrale.

SE LA STAMPANTE TERMICA NON GIRA COL COMANDO LPRINT, SCRIVERE DA BASIC

OUT 3,255

E VEDRETE CHE POI FUNZIONA SUBITO; L'INIZIALIZZAZIONE SUDETTA VA FATTA OGNIQUALVOLTA SI ACCENDE IL COMPUTER (SOLO PER LA STAMPANTE TERMICA).

Fino a quando la scheda grafica non sarà disponibile, sul monitor vedrete solo lettere maiuscole. **RICORDATE PERÒ CHE TUTTO QUELLO CHE VIENE SCRITTO COME STRINGA (CIOÈ TRA VIRGOLETTE) È VISUALIZZATO IN MAIUSCOLO, MA VIENE MEMORIZZATO MAIUSCOLO O MINUSCOLO A SECONDA DI COME ERA STATO DIGITATO.**

Vale a dire che se avete scritto una stringa senza premere il tasto SHIFT delle maiuscole, la stringa stessa viene visualizzata in maiuscolo sul monitor, ma presentata in minuscolo dalla stampante. Questa considerazione è importante anche quando si opera a livello di EDITING, come avremo modo di chiarire in seguito.

NEL DRIVE O DEVE SEMPRE STARE UN DISCO COL DOS-BASIC, altrimenti diverse funzioni operative BASIC e DOS non possono funzionare. Sarebbe quindi opportuno operare con DUE DRIVE, in modo da poter registrare e prelevare programmi e dati col secondo floppy. Proprio per questo stiamo approntando un contenitore in grado di alloggiare due drive.

Fermiamoci qui, per questa volta. Avete di che sbizzarrirvi in prove e controprove, ma non mancherete di divertirvi. Ora avete in mano uno strumento molto potente e perfezionato, che eseguirà fedelmente e con precisione i vostri programmi. I risultati che ne trarrete dipenderanno, oltre che dalla nostra chiarezza, anche dalla vostra abilità e fantasia. Per ora non ci resta che augurarvi buon lavoro.

DISCO TEST

Come alcuni lettori avranno avuto modo di scoprire, il disco Test non funziona con il «nuovo Basic NE/Dos. Infatti essendo il nuovo Basic più evoluto ed adottando per ciò una diversa formattazione del disco, il computer è impossibilitato a leggere i programmi contenuti nel disco Test.

Invece di defalcare dal costo del Basic il costo del dischetto test, ed aggiungervi il costo della nuova Eprom 1.390 da inserire nella scheda LX.390 indispensabile alla gestione del Basic stesso abbiamo deciso di fornire allo stesso prezzo il nuovo BASIC più la Eprom già programmata.