



# Parla più FORTH

di Raffaello De Masi

## Prima parte

### Stack - RPN - Word

*È questo il primo di una serie di articoli destinati a presentare il Forth. Questo linguaggio, non ben conosciuto in Italia, presenta notevole interesse pratico. Elevata rapidità operativa, tanto da essere utilizzato normalmente per operazioni in tempo reale, estrema flessibilità, estendibilità praticamente limitata solo dalla fantasia dell'operatore, facilità di acquisizione anche da parte di principianti grazie alla sua struttura estremamente logica, ne hanno fatto il linguaggio ideale per applicazioni in campi estremamente diversi; esso è implementato infatti tanto in sistemi ad indirizzo gestionale, in cui è fondamentale la rapidità operativa, che in macchine disegnate per usi scientifici.*

*Lo schema con cui sono stati scritti gli articoli risponde all'intento di guidare il lettore ad una conoscenza graduale e logica del linguaggio. Essi sono stati redatti in modo da consentire l'apprendimento o la semplice lettura anche da parte del principiante, anche non in possesso di specifica conoscenza di un linguaggio ad alto livello, né è richiesto il possesso di un calcolatore programmabile in tale linguaggio. Ciononostante la conoscenza anche superficiale di un linguaggio di programmazione, tipicamente il Basic, ed il possesso di una macchina programmabile in Forth consentono acquisizione più rapida dei concetti ed una verifica immediata di quanto esposto.*

### Un po' di storia

Verso la metà degli anni 60, Charles H. Moore si rese conto, come lui stesso pittorescamente racconta, che con le possibilità allora offertegli dai linguaggi disponibili (Fortran, Algol) nella sua vita avrebbe, sì e no, scritto una trentina di programmi impegnativi. Le gravi difficoltà da lui incontrate nel preparare un programma applicativo di cronotografia utilizzando il Fortran lo convinsero ad analizzare la possibilità di preparare un linguaggio più flessibile e versatile e, soprattutto, più veloce e facile da usare. L'idea, partita solo con l'intento di creare nuove routine di specifica utilizzazione, lo portò, in un anno, a disporre di un notevole bagaglio di utility molto potenti. Da qui alla creazione di un nuovo linguaggio il passo fu breve. Nel 1968 si arrivò alla prima stesura di un linguaggio nuovo che Moore volle definire della quarta generazione, con chiaro riferimento ai calcolatori di allora appartenenti

alla cosiddetta terza generazione. Poiché, però, il linguaggio era implementato su una macchina IBM 1130 che accettava identificatori solo di 5 caratteri, il linguaggio della fourth generation divenne FORTH.

Tre anni dopo, nel preparare un programma di acquisizione dati da un radiotelescopio per conto del National Radio Astronomy Observatory, Moore aggiunse un compilatore al sistema, e nel 1973 si giunse alla possibilità di multiprogrammazione. In quel periodo un sistema, utilizzante routine in FORTH, pilotò un radiotelescopio alla stazione NRAO di KITT PEAK, in Arizona. In tale occasione, e grazie a tale combinazione, furono scoperti gran parte dei moduli interstellari (space dust) tuttora conosciuti.

Nello stesso anno, Moore ed alcuni suoi collaboratori lasciarono l'NRAO per formare la FORTH Inc. Sebbene inizialmente impegnata a risolvere problemi di controllo di strumentazione, in cui il FORTH rappresenta la soluzione di gran lunga più vantaggiosa, la compagnia si interessò, grazie alle notevoli richieste di mercato, anche alla soluzione di problemi diversi, primo fra tutti quelli gestionali e commerciali.

Il FORTH riscosse immediatamente un notevole successo. Già nel 1976 si fondò, in Europa, l'European Forth Users Group (EFUG) che, nel suo primo meeting inter-

nazionale, formò il FORTH Standard team, incaricato di definire le funzioni di libreria ed il vocabolario minimo da implementare in tutti i sistemi onde garantire la compatibilità. Nel 1978 in California fu fondato il FORTH INTEREST GROUP (FIG) destinato a promuovere l'uso del linguaggio tramite interscambio di idee, seminari, pubblicazioni e corsi di aggiornamento. Già nell'82 il FIG aveva 2400 membri, e alla fine dell'83 la rivista mensile FORTH DIMENSION, edita dal FIG, censiva 4000 membri.

IL FORTH Standard Team realizzò e pubblicò, nell'ottobre dell'80, la prima edizione dello standard FORTH. Questo, che non ha ancora subito aggiornamenti, viene chiamato FORTH 79 Standard. L'altro dialetto esistente, il FIG FORTH, si serve dello standard descritto del Fig Forth Installation Manual, redatto da William Ragsdale. La presenza di due dialetti diversi (principali, perché ne esistono ancora altri, sovente finalizzati alle macchine su cui sono implementati, come ad esempio l'MMSFORTH implementato sulle macchine della Tandy Radio Shack) farebbe pensare ad un'altra occasione perduta di unicità di linguaggio. La differenza, però, è più apparente che reale, grazie alla estrema flessibilità del linguaggio che consente, con facilità, di implementare nell'uno comandi presenti nell'altro.

Ad esempio il Forth 79 ha una libreria più ricca mentre il FIG Forth possiede più numerose routine elementari, utili per la costruzione di comandi mancanti, ma più versatili e comode, quindi con semplicità operativa maggiore.

Nei nostri articoli, ove non diversamente citato, ci si riferisce a comandi, routine, e strutture logiche presenti in ambedue i dialetti. Comunque, è il caso di ripeterlo, quanto diremo è normalmente utilizzabile su qualsiasi sistema dotato di linguaggio FORTH, a qualunque dialetto appartenga.

### La torre di Babele

Nel 1954 J. W. Backus, ricercatore dell'IBM, propose il primo vero linguaggio ad alto livello, il FORTRAN, destinato a trattare algoritmi di tipo matematico (FORMula TRANslator — traduttore di formule). La prima revisione generale del linguaggio portò al FORTRAN II, cui seguirono il III, abortito prima della nascita, ed il IV, il più diffuso. La versione più aggiornata, il FORTRAN 77, fu proposta appunto in tale anno.

Il FORTRAN, comunque, presentava grosse lacune nel campo della rapida gestione dei file dati. Il governo USA, caso unico tra i governanti, resosi conto del problema e dimostrando grande preveggenza circa l'importanza che avrebbero assunto i linguaggi di programmazione, costituì una commissione per lo studio di un nuovo linguaggio. La CODASYL (CONference on DATA SYstem Language) partì con il mai soddisfatto proposito di preparare un linguaggio universale, destinato tanto ad applicazioni scientifiche che commerciali e gestionali. Resasi conto delle enormi difficoltà cui andava incontro, e visto che il FORTRAN funzionava benissimo in

campo scientifico, ci si orientò verso la soluzione di problemi gestionali e commerciali. Ebbe così vita il COBOL (Computer Business Oriented Language), anche esso destinato ad enorme successo.

Nel frattempo, ad opera di gruppi di ricercatori isolati, comparivano (1958) l'ALGOL (ALGORitmic Language) ed il LISP, il primo dei quali elaborato da matematici europei. Questo è il primo esempio di linguaggio strutturato ed aprirà la strada al PASCAL ed ad altri.

Il problema che si trovò a risolvere il CODASYL non è stato, ahimè, mai del tutto risolto. L'ADA (che prende il nome da Ada Lovelace, nipote di Lord Byron), il C (dalla rapidità entusiasmante) ed ultimo in ordine di tempo il MODULA-2 hanno tentato, come anche il Basic, di colmare l'abisso tra i due indirizzi fondamentali dell'informatica senza mai centrare effettivamente l'obiettivo, tant'è che sussiste ancora la cosiddetta vocazione del linguaggio, cui purtroppo non sfugge neppure il FORTH, la cui destinazione principe è il controllo di sistemi di strumentazione.

## Le basi del FORTH

Il FORTH è la soluzione ideale per impieghi scientifici ed industriali. In pratica è quasi sempre possibile, grazie a questo linguaggio, risolvere più pulitamente, efficientemente e rapidamente la maggior parte dei problemi di tal genere che non servendosi di un linguaggio più comune, primo tra tutti il Basic.

È stato detto, appunto, che il Basic rappresenta l'utilitaria dei linguaggi: ti porta lentamente ma dappertutto. Il Pascal, nonostante tutte le sue raffinatezze, non si è rilevato poi molto migliore, tant'è che i suoi estimatori rappresentano solo una compagnia contro i corpi d'armata dei Basic. Dopo l'avvento del Fortran nel 1950 molti linguaggi ad alto livello si sono susseguiti sulla scena. Alcuni, dopo un iniziale interesse, sono scomparsi; altri anche ben congegnati, sono piuttosto complessi nella pratica, altri ancora sono dedicati a particolari problematiche e non hanno le doti per divenire di uso universale. Perché mai il Forth dovrebbe riuscire dove altri hanno fallito?

Per un buon motivo. L'approccio al problema è completamente diverso dalla prassi comune e le tecniche di programmazione sono inusuali tanto da richiedere, all'inizio, un piccolo sforzo di adattamento. Il Forth è molte cose insieme: infatti è:

- un linguaggio ad alto livello
- un assembler
- un sistema operativo

ma soprattutto è una nuova filosofia del software. Inoltre è molto più rapido dei linguaggi già nominati (alcuni sostengono anche 20 volte più del Basic) e, pur essendo più lento del 20-40% rispetto al linguaggio Assembler, elimina l'elevata noiosità di redazione. Inoltre è compatto (l'intero sistema operativo e la libreria di base, già molto potente, occupano non più di 8K) ed è facilmente trasportabile. La sua rapidità operativa lo ha fatto preferire per il controllo delle cineprese e per la creazione degli effetti speciali in tutti e tre gli episodi di "Guerre Stellari" e, ancora, per l'esecuzione delle eccezionali riprese finali del film "I predatori dell'arca perduta". Grazie alla potenza del Forth è stato possibile controllare contemporaneamente, nella già citata stazione NRAO, un telescopio straordinariamente accurato, la cupola d'osservazione, diversi monitor, memorie di massa, stampanti, continuando nel frattempo le normali routine di controllo, analisi, stampa ed immagazzinamento dei risultati delle analisi dei dati di emissione degli infrarossi dallo spazio.

Dicevamo che il Forth è inusuale, e lo è proprio fin dall'inizio. Pertanto, prima di entrare nel vivo del linguaggio è necessario evidenziare alcune particolari caratteristiche di base. Di queste le più importanti ed appariscenti sono:

lo Stack  
la RPN  
le parole (Word) ed il vocabolario.

Altre, anch'esse inusuali, come la notazione in virgola fissa, i buffer, i numeri in singola, doppia e tripla precisione, gli screen ed altre verranno trattate man mano che se ne presenterà l'occasione.

## Lo Stack

Virtualmente, tutte le operazioni del Forth coinvolgono in qualche maniera una struttura interna definita Stack. Chi ha esperienza di programmazione in Assembler conosce già questa struttura che funziona allo stesso modo in Forth, anche se è molto più utile e semplice da usare.

Chi invece non conosce lo Stack immagina una cucina di un ristorante. Su un bancone esiste un grosso foro delle dimensioni di un piatto chiuso da una piastra tenuta da una molla. Ogni volta che si aggiunge un piatto la piastra scende di un posto ed ogni volta che un piatto viene prelevato essa risale. Questa catasta (Stack) è una struttura del tipo LIFO (Last in - first out) vale a dire che i primi piatti ad essere inseriti saranno gli ultimi ad uscire e viceversa.

In pratica lo Stack è un'area nella memoria del computer dove i numeri sono temporaneamente accatastati. Ogni volta che vi si inserisce un numero questo va a depositarsi sopra quello precedente così che il numero che viene eventualmente prelevato è sempre l'ultimo appena inserito.

Per inserire numeri nello Stack basta scrivere il numero stesso e premere il tasto RETURN (Enter, CR, END LINE a seconda di come viene "battezzato" nel computer). Ad esempio digitando:

```
152 RETURN
il numero (decimale) 152 sarà inserito nel TOS (Top of stack - cima dello Stack).
```

È possibile inserire più numeri nello Stack sia scrivendoli e facendoli seguire dall'ENTER uno per uno, sia separandoli su una stessa riga da uno spazio. Ad esempio

```
3 325 17 441 RETURN
inserirà nello Stack i numeri nel seguente ordine
```

```
441 (TOS)
17
325
3
```

Vale a dire che i numeri vengono accatastati l'uno sull'altro da sinistra a destra.

Così come è possibile inserire è altrettanto agevole estrarre dallo Stack un numero. Per fare ciò è sufficiente digitare un (·) seguito da RETURN. Ciò estrarrà il TOS mostrandolo sul terminale e farà salire lo Stack di un posto.

Ad esempio nel caso precedente,

```
Stack
441 (TOS) 17
325
3
battendo . RETURN si avrà sullo schermo
441 OK
e lo Stack sarà così composto
17 TOS
```

325

3

Battendo diversi punti (sempre separati da uno spazio) è possibile estrarre altrettanti numeri dallo Stack.

Es.

```
... RETURN
darà sullo schermo
17 325 3 OK
```

Il messaggio OK emesso dopo ogni operazione comunica che non sono stati effettuati errori operativi. Estrarre più numeri di quanti ce ne siano nello Stack porta a condizioni d'errore. Ad esempio, sempre nel caso precedente un ulteriore

```
. RETURN
darà come risultato il messaggio
? EMPTY STACK
ossia Stack vuoto; questa situazione si chiama "Underflow".
```

Come è ovvio, le operazioni di inserimento ed estrazione (Push e Pop per gli assembleristi) possono essere combinate sulla stessa riga. Ad esempio

```
31 15 28 . . RETURN
darà come risultati
28 15 OK
```

ed il numero 31 rimarrà nello Stack come TOS.

Notare come in questo esempio si veda bene la struttura della catasta. Il numero 28 inserito per ultimo viene fuori per primo. Pertanto, anche di seguito, converremo che i numeri più a destra di una riga (da inserire) sono quelli di testa degli Stack.

L'inserimento di un numero eccessivo di elementi può riempire lo Stack, determinando una condizione di overflow cui fa seguito il messaggio (? Stack overflow)

L'area di stack è variabile a seconda delle macchine, ma generalmente tanto grande che le condizioni di overflow sono rare. Non dimentichiamo, comunque, che lo Stack è un'area di parcheggio temporaneo di dati per cui, generalmente, le sue condizioni di uso sono di molto al di sotto delle effettive capacità di immagazzinamento.

Non è questo il solo Stack presente nel Forth. Per ora comunque è sufficiente conoscere questo.

## La RPN

Nel normale uso delle calcolatrici tascabili e in molti linguaggi evoluti, tra cui il Basic, un'operazione di sottrazione tra 12 e 5 si scrive

```
12-5 RETURN (o = nelle calcolatrici)
così come la si scriverebbe facendo effettivamente l'operazione con carta e penna. Poiché, invece, il Forth esegue operazioni su numeri già presenti nello Stack è necessario che questi vengano inseriti prima dell'operatore, vale a dire
```

```
12 RETURN
5 RETURN
- RETURN
o più semplicemente
12 5 - RETURN
```

Il segno - sottrae il TOS (5) dal numero

al secondo posto dello Stack. I numeri utilizzati vengono estratti dallo Stack ed il risultato viene posto nel TOS vale a dire

STACK	- Return	Stack
5		7
12		-
-		-
-		-
-		-

Come si vede, l'operazione ha effetto distruttivo sui valori presenti originariamente nello Stack.

Questo modello operativo, di tipo post-fisso, dove il segno di operazione segue gli operandi, è chiamato RPN (Reverse Polish Notation — notazione polacca inversa) in quanto la disposizione dei numeri e degli operandi è invertita.

Questo sistema, arcinoto e caro agli utilizzatori delle calcolatrici Hewlett-Packard, cui aderiscono quasi religiosamente, è specifica anche di altre calcolatrici cosiddette "stack oriented" (es. Novus, Hasai, ecc.). Il suo uso, per lo meno strano, per chi è abituato ad operare su sistemi standard operando — operatore — operando (SOA: sistema operativo algebrico), ha però il pregio di divenire familiare dopo un uso anche relativamente breve.

Non entreremo qui nell'eterna contesa tra RPNisti e SOAisti. Ambedue i sistemi hanno i loro pregi. A favore della RPN gioca un minor uso di comandi (ad esempio mancano addirittura i segni di parentesi) ma l'uso non è altrettanto intuitivo come nel SOA. Ad esempio l'espressione  $3 + (5 \times 2 + (5 - 3 + (12 \times 4 - 5 \times 8) + 3))$  in SOA va scritta così come è seguita dal segno di =;

In RPN la stessa espressione si scriverà:  $12\ 4 \times 5\ 8 \times - 5 + 3 - 3 + 5\ 2 \times + 3 +$  per ottenere lo stesso risultato di 26.

In effetti nel secondo caso si ha un certo risparmio di tempo (vengono premuti 20 tasti invece di 27) ma occorre un piccolo sforzo mentale (per il novizio) per inserire correttamente l'espressione. La cosa generalmente sparisce dopo pochi esercizi essendo in fondo abbastanza intuitiva.

Le calcolatrici HP usano una catasta di 4 registri (registri X, Y, Z e T) in quanto quattro locazioni di memoria nello Stack sono sufficienti per un grosso numero di problemi. La lunghezza enormemente superiore dello Stack del Forth non avrebbe molto senso se non esistessero (li vedremo tra poco) una serie di comandi che manipolano direttamente il contenuto dello Stack.

È comunque importante ricordare che è necessario, quando si usa un operatore, che ci sia già qualcosa nello Stack. Tenere a mente, inoltre, che il TOS è il numero che il Forth, tramite la RPN, usa come addendo, sottraendo, moltiplicatore e divisore rispetto al secondo numero in qualsiasi operazione aritmetica.

## Le word ed il vocabolario

Ogni linguaggio possiede una sequenza di funzioni di libreria, vale a dire una serie

di termini, formati da lettere e/o numeri che, digitati da tastiera, consentono di eseguire alcune predeterminate funzioni. Queste si chiamano istruzioni (instruction) in assembler (es. ADD, JUMP), comando, ordine, dichiarazione e affermazione (statement) in Basic e Pascal (GOTO, PLOT, BEGIN), e parole (word) in Forth.

Una word (useremo la parola originale in analogia all'uso dei corrispondenti termini statement e instruction) è una sequenza di uno o più caratteri che identificano una procedura di esecuzione.

Tanto per intenderci la word + determina la somma dei due numeri in cima allo Stack e (si noti la differenza con il normale +) l'inserimento del risultato in TOS.

Ogni sistema Forth contiene già un vocabolario di parole predefinite, preinserite dal costruttore in una struttura interna del programma definita DIZIONARIO.

Questo è specifico per il dialetto di cui si è in possesso. Così, il Forth 79 contiene il Forth 79 Required Word Set; altri, come il FIG FORTH, il poliForth ecc. avranno dizionari diversi.

Tutto ciò non comporterebbe differenze apprezzabili con gli altri linguaggi se non intervenisse qui l'aspetto più entusiasmanente del Forth, che lo rende praticamente unico.

Nel Forth, infatti, è possibile definire nuove word che eseguono, mettendone insieme alcune già esistenti, funzioni ed operazioni non previste in libreria. Immaginiamo ad esempio di dover eseguire le somme

$13 + 8 + 4 - 5$

e di visualizzare il risultato.

In base a quanto detto precedentemente si dovrebbe digitare

$13\ 8 + 4 + 5 - .$  RETURN

o, il che è lo stesso, date le caratteristiche della RPN

$13\ 8\ 4\ 5 - + + .$

che danno ambedue 20 OK con Stack vuoto.

Osserviamo ora la sequenza di definizione di una nuova parola (Word) che consente la sequenza (- + + .). Si voglia cioè creare la parola ROMA che esegue la differenza tra i due numeri in cima allo Stack, la sommi al terzo ed al quarto termine ed infine mostri il risultato

: ROMA - + + . ;

Si noti che tutti i termini sono separati da spazi. Il segno :, definitorie, indica al sistema che si sta creando una nuova parola (ROMA) che consente di eseguire e riassume le parole successive (- + + .). L'ultima (;) definita delimitatore, indica al sistema che l'operazione di definizione è terminata.

A questo punto riproviamo con la precedente operazione

$13\ 8\ 4\ 5\ ROMA$

il sistema restituirà

20 OK

Due o più parole possono essere riunite insieme a formarne una nuova.

Facciamo un esempio. La word EMIT, abbinata ad un numero, stampa sullo schermo il carattere ASCII corrispondente al numero (in Basic PRINT CHR\$(numero)). La word SPACES, anche essa abbinata ad un numero, stampa altrettanti spazi. CR è il conosciuto CARRIAGE RETURN (ritorno a capo). Definiamo quindi la parola AA

: AA 86 EMIT 73 EMIT; RETURN  
ed AB

: AB 86 EMIT 65 EMIT; RETURN  
che daranno due imperturbabili OK ad indicare l'accettazione da parte del vocabolario di questi due nuovi termini.

Definiamo adesso la parola AC  
: AC 77 EMIT 69 EMIT; RETURN  
Riassumiamo le parole definendo AD  
: AD AA AB 2 SPACES AC CR;  
RETURN

Eseguito AD RETURN avremo sullo schermo

VIVA ME

VIVA ME

VIVA ME

OK

La parola AD, come quelle definite precedentemente, è stata inserita nel dizionario e vi rimarrà in permanenza, effettuando sempre l'operazione descritta, fino a quando non sia tolta l'alimentazione al computer, a meno che non voglia essere conservata con procedure che verranno descritte in seguito, o desideri essere espressamente cancellata o ridefinita.

Una struttura di questo tipo (rope structure = struttura a corda, con chiara analogia con i trefoli di una fune che si uniscono tra di loro per formarne uno più grosso ed i risultanti si uniscono ancora per formare una struttura più grande che li comprende) è come si vede abbastanza comoda e, soprattutto, estremamente elastica. Poiché non esiste limite alle confluenze di word in altre word se non le disponibilità di memoria stessa, è più che evidente che al programmatore viene consentita la massima libertà d'azione nello scrivere i propri programmi. La word, alla fine, sarà sempre in forma compilata, con gli immediati vantaggi su strutture e routine interpretate (come ad esempio nel caso del Basic).

Appare subito evidente l'elevata qualità del linguaggio, estremamente flessibile ed interattivo. Esso consente al programmatore di costruirsi le routine o i comandi più comodi o, perché no, che più gli aggradano.

È come avere a disposizione una officina con una serie di utensili (word già implementate nel vocabolario di base) che consentono di costruire attrezzi specifici per la bisogna, buttandoli poi via quando non servono più.

Le parole già presenti nel dizionario non sono numerosissime (80-100 in tutto anche nei vocabolari più forniti) ma generalmente molto potenti e capaci di coprire tramite opportune combinazioni praticamente qualsiasi problema operativo. La descrizione di tali word sarà oggetto dei prossimi articoli.

# NON NASCONDIAMO NIENTE. NON TACIAMO NULLA.

L'MPFII è un calcolatore compatto, piccolo come un libro, (cm. 25x18,5x3,2) con tastiera incorporata. Nonostante le ridotte dimensioni, la memoria in ROM è di 16 K e in RAM è di 64 K.

MPFII possiede una ampia serie di interfacce residenti, lavora in bassa ed in alta risoluzione, a colori, contiene un altoparlante per la generazione di suoni e voci (via software).

## CARATTERISTICHE TECNICHE:

- CPU: 6502, 1 MHz
- ROM: 16 K con interprete basic apple soft, Monitor Disassembler
- RAM: 64 K
- VIDEO
  - Testo: 40x24 (2 pagine)
  - Grafica GR: 40x48 6 colori (2 pagine)
  - Grafica HGR: 280x192 6 colori (2 pagine)
- CARATTERI: 64 ASCII maiuscoli più simboli grafici e funzioni basic ONE KEY
- TASTIERA: 49 tasti multifunzioni
- INTERFACCE RESIDENTI
  - Per registratore



- Per monitor
- Per TV (color sistema Pal)
- Per tastiera esterna o joy-stick
- Per stampante (standard centronic)
- SLOT/BUS MICROPROCESSORE
  - Per interfaccia per drive-disk (FDI 1 o FDI 2)
  - Per programmi su eprom (cartridge)
  - Per schede I/O
- ALIMENTATORE ESTERNO TIPO SWITCHING
- DIMENSIONI: 25x18,5x3,2 cm.
- PESO: 780 gr.

**OPZIONALI:** Interfaccia per drive-disk (FDI) / Drive-disk Slim-Line (FDD) / Interfaccia per stampante (centronic-pik) / Interfaccia seriale RS 232 C / Tastiera esterna (originale Multitech o dattilo) / Joy-stick / Registratore / Monitor fosfori bianchi - verdi - arancio e color / Stampanti a 40 colonne termica (ST 40) o 80/132 colonne (carta comune) / Porta I/O / Bus a 3 slot / Generatore caratteri cinesi / Valigetta sistema e/o di trasporto ecc. ...

## NOTE PARTICOLARI:

- Non è una copia! La sua compatibilità si limita al basic e permette di costruire programmi accedendo alle bibliografie più complete, oggi esistenti, senza dovere "RICOMINCIARE".
- È l'unico sistema esistente in grado di ricostruire (via grafica) e visualizzare qualsiasi tipo di carattere (arabo, cirillico, greco, scandinavo, cinese, ecc.).
- La piccola tastiera, plurifunzionale, lo rende particolarmente adatto per impieghi hobbistici ed industriali (controllo di processi, controlli di sistema, robotica ecc.).
- Le ridotte dimensioni non vanno a discapito della potenza, ma facilitano gli impieghi più disparati, lasciando alla accessoristica (tastiera dattilo, valigia di contenimento ecc.), la trasformazione del computer in un Personal da scrivania.
- La possibilità di modifica del DOS operativo ne fanno, oggi, l'unico sistema, a basso costo, a più alta capacità di memoria basic.
- Il particolare indirizzamento colori in grafica, in bassa e alta risoluzione, rende possibile la creazione, tramite 6 colori di base, di innumerevoli combinazioni, nuances o gradazione degli stessi.

- Il particolare sistema operativo, su disco, DOS, gli permette di operare con 8 file in contemporanea, per una estrema operatività e velocità, sul Drive-disk.
- L'universalità del sistema, gli permette, all'uscita delle proprie interfacce, di utilizzare le periferiche più comuni, arrivando a configurazioni estremamente potenti.
- L'altoparlante, entro contenuto, permette di verificare, immediatamente, gli effetti sonori e vocali derivati o inseriti nei Vs. programmi.
- La manualistica, in italiano, in dotazione, può essere completata con ulteriore manualistica comunemente reperibile sul mercato.
- Più di 100 giochi su cassette e cartridge ed oltre 50 programmi gestionali su disco, costituiscono parte del software esistente per MPFII.

# DIGITEK COMPUTER

VIA VALLI, 28 - 42011 BAGNOLO IN PIANO (Reggio Emilia) Tel. (0522) 61623 r.a.

Per richiedere Catalogo Generale, inviare € 2000 in francobolli

Cognome \_\_\_\_\_  
Nome \_\_\_\_\_  
Via \_\_\_\_\_  
Città \_\_\_\_\_  
CAP \_\_\_\_\_  
MPFII  
MC