

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
1							r								
2	g														t
3		l			s			e				h		e	
4			i						d		s		e		

## Uso di zone del tabellone per routine di utilità

di Francesco Petroni

Fino a pochi anni fa l'uso del computer rispondeva a certe regole rigide, anche nell'ambito del lavoro di programmazione, in generale tendenti a minimizzare i tempi di uso della macchina, in quanto era questo il costo maggiore.

Tali regole erano quindi dettate dagli strumenti hardware e software allora in uso. E, per quanto riguarda la realizzazione di programmi, tali regole erano, in buona parte, propedeutiche all'insegnamento di questo o quel linguaggio. In pratica si tendeva, durante l'analisi a tavolino, a isolare il problema e a considerarlo indipendentemente dall'esistenza dei computer, dalla realizzazione del programma.

Queste regole comportamentali sono oggi, con l'utilizzazione generalizzata dei personal computer, quasi sempre trasgredite. Ormai sono pochissimi, forse solo quelli che provengono da esperienza mainframe quelli che, correttamente, affrontano un problema con tecniche di analisi, tutti gli altri, si mettono direttamente alla macchina, cosa che è sicuramente più di-

vertente, ma che comporta alcuni rischi.

Senza voler entrare nella polemica di quale sia il metodo più corretto per affrontare e risolvere un problema di tipo informatico, diremo solo che tale secondo metodo (quello divertente) può risultare pericoloso in quanto permette di praticare soluzioni apparentemente corrette e efficaci (che raggiungono cioè l'obiettivo) ma in modo non efficiente (ovvero in modo costoso rispetto ad altre soluzioni).

Con la diffusione dei personal computer e con la proliferazione dei venditori di macchine e quindi con l'aumento verticale della domanda di programmi applicativi, sono nate numerosissime software houses, che hanno prodotto migliaia di pacchetti di livello qualitativo il più svariato possibile, in quanto da una parte non esiste una cultura da parte degli eserciti di nuovi utenti, e dall'altra spesso non esiste una coscienza professionale da parte di chi produce il pacchetto.

Il giudizio sul software non si può infatti limitare al fatto che siano pre-

senti e che funzionino tutte le operazioni previste come necessarie all'applicazione, ma deve essere esteso a concetti di qualità. Tali concetti sono esteriori, ovvero il pacchetto deve essere di facile comprensione, non deve penalizzare eventuali errori dell'operatore ma anzi deve permettere il recupero di situazioni critiche con il minor danno possibile.

Il concetto di qualità deve essere anche interno, intendendo con tale termine la correttezza formale della programmazione, cosa di cui l'utente al limite non se ne accorge, ma che risulta evidente all'esperto che «legge» i listati.

È come se aprendo il vostro televisore, che ha sempre funzionato benissimo e del quale siete pienamente soddisfatti, vi accorgete che internamente c'è un groviglio di fili, oppure delle piastre con elementi sparsi in maniera disordinata. Da quel giorno sicuramente non avrete più tanta fiducia nel vostro apparecchio.

Nel caso di programmatori «autodidatti», in pratica lo sono tutti gli utiliz-

zatori di personal computer che lavorano in Basic, è evidente che le regole basilari della programmazione non vengono rispettate, in quanto non sono conosciute, e questo si nota, per esempio esaminando un loro listato. Successivamente, con la pratica, lo stile si affina, anche se, secondo me, ogni programmatore possiede uno stile personale, e buona parte delle regole della buona programmazione vengono in un certo senso «scoperte».

I più bravi probabilmente costruiranno proprie regole di programmazione, ad esempio arriveranno ad applicare in pieno la metodologia di programmazione strutturata, non perché l'abbiamo mai studiata, ma perché ne scoprono i vantaggi operativi.

In questo discorso entra pesantemente il linguaggio che impone le proprie caratteristiche condizionando in maniera determinante la costruzione delle proprie regole di programmazione.

Ma cosa succede quando appaiono nuovi prodotti software, non solo nuovi linguaggi, che grosso modo conservano tutte le stesse funzionalità, ma nuovi tool che introducono nuovi tipi di rapporto tra uomo e macchina.

Esistono due tipi di atteggiamento, quello che pretende di ritrovare anche nel nuovo ambiente gli stessi concetti del vecchio ambiente, magari faticosamente acquisiti, e quello che invece abbandona le vecchie «verità» per trovare al più presto la filosofia del nuovo prodotto.

Infatti si tratta di «filosofia». È necessario più della conoscenza del prodotto, che consiste nello studio di istruzioni, sintassi, ecc., un buon ambientamento. Questo vuol dire mettersi a proprio agio, capire le regole fondamentali, che permettono all'occorrenza di risolvere anche i problemini spiccioli.

Ad esempio con il vecchio COBOL occorre per ogni variabile definire la lunghezza in termini di byte, mezzi byte, ecc. con il Basic non è più necessario, si può anche non definire la variabile, in quanto assume caratteristiche di default. Se proprio si vuole si può fare, e in certi casi è meglio. In ambiente spreadsheet la variabile non esiste.

Se prendete libri di Cobol, di For-

tran, di Basic buona parte dei capitoli iniziali sono dedicati alle variabili, nei manuali dei vari Visicalc, Multiplan, Lotus 123, ecc., la variabile come entità, non è neanche citata.

Anzi è inutile andarla a cercare. Sicuramente non la va a cercare il neofita che affronta per la prima volta il Personal Computer, ma non deve andarle a cercare neanche il vecchio programmatore.

### Dalla variabile alla cella

Se proprio si vuole cercare un parallelismo diremo che la variabile viene in tutto e per tutto sostituita dalla cella, che però ha un significato e una potenzialità ben superiore.

La cella può essere usata come variabile, semplicemente inserendovi qualcosa, sia esso un valore numerico, o una stringa, può essere usato come

-1	1	2	3	4	5	6
1	MULTIPLAN	TABELLA	PITAGORICA	ALFABETICA		
2						
3	uno	due	tre	quattro	cinque	sei
4	due	quattro	sei	otto	dieci	dodici
5	tre	sei	nove	dodici	quindici	diciotto
6	quattro	otto	dodici	sedici	venti	ventiquattro
7	cinque	dieci	quindici	venti	venticinque	trenta
8	sei	dodici	diciotto	ventiquattro	trenta	trentasei
9	sette	quattordici	ventuno	ventotto	trentacinque	quarantadue
10	otto	sedici	ventiquattro	trentadue	quaranta	quarantotto
11	nove	diciotto	ventisette	trentasei	quarantacinque	cinquantatré
12	dieci	venti	trenta	quaranta	cinquanta	sessanta
13						
14						
15	1	6	3	6		diciotto
16	2	7				
17	3	8	6	2		dodici
18	4	9				
19	5	10				
20						

COMANDO: Alfa Blocca Copia Distruggi Esterno Formato Guida Inser. Jump Libera Modifica Nome Ordina Parametri Quadrante Riposiz. Stampa Trasf. Uscita Valore Scegliere un'opzione o digitare la prima lettera del comando  
R17C6 INDICE(pitagora;R17C3;R17C4 100% lib. Multiplan: b:TABFIT

Figura 1 - Tavola Pitagorica in «Stringa». L'esempio, realizzato in Multiplan 2.0, illustra l'uso della funzione «INDICE» per mezzo della quale è possibile estrarre un elemento da una matrice rettangolare. La matrice rettangolare è posizionata in una zona del tabellone.

H14: (F3) @SCANSVER (F14; \*TRG; 1) \*@SCANSVER (@MOD (2\*E14; 360); \*TRG; 2) PRONTO

	A	B	C	D	E	F	G	H	I	
1	formula	col. B	@SEN(A10*PI/180)							
2	formula	col. C	@COS(A10*PI/180)							
3	formula	col. F	@MOD(E10; 360)							
4	formula	col. G	@SEN(E10*PI/180) * @COS(E10*PI/90)							
5	formula	col. H	@SCANSVER(F10; *TRG; 1) * @SCANSVER (@MOD (2*E10; 360); *TRG; 2)							
6	-----									
7	angolo	seno	coseno	ang	mod	sen(a)	cos(2a)			
8				ang/360						
9	-----									
10	0	0,000	1,000	0	0	0,000	0,000			
11	15	0,259	0,966	15	15	0,224	0,224			
12	30	0,500	0,866	30	30	0,250	0,250			
13	45	0,707	0,707	45	45	0,000	0,000			
14	60	0,866	0,500	60	60	-0,433	-0,433			
15	75	0,966	0,259	75	75	-0,837	-0,837			
16	90	1,000	0,000	90	90	-1,000	-1,000			
17	105	0,966	-0,259	105	105	-0,837	-0,837			
18	120	0,866	-0,500	120	120	-0,433	-0,433			
19	135	0,707	-0,707	135	135	0,000	0,000			
20	150	0,500	-0,866	150	150	0,250	0,250			
	01/01/80	00:01						CALC		

Figura 2 - Trigonometria. L'esercizio consiste nel trattare le funzioni trigonometriche come una tabella di valori precalcolati. In questo modo si possono eliminare dalle formule le funzioni Seno, Coseno, ecc. L'utilità di tale esercizio è duplice. Per gli strumenti privi di funzioni trigonometriche, costituisce una valida alternativa. Per gli strumenti che invece ne dispongono, può essere utile per velocizzare i tempi di ricalcolo.

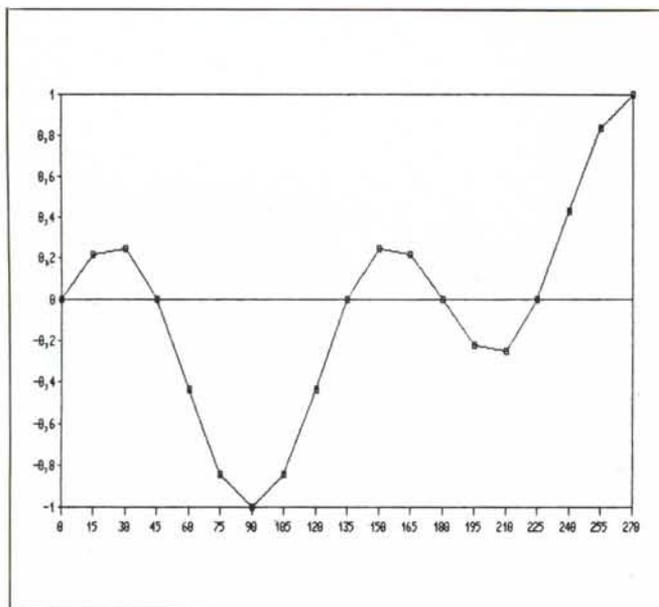


Figura 3 - Trigonometria - Intervallo (primo tempo). Nell'intervallo tra un argomento e un'altro, realizziamo la graficizzazione della curva calcolata, utilizzando i comandi minimi indispensabili per ottenere un disegno.

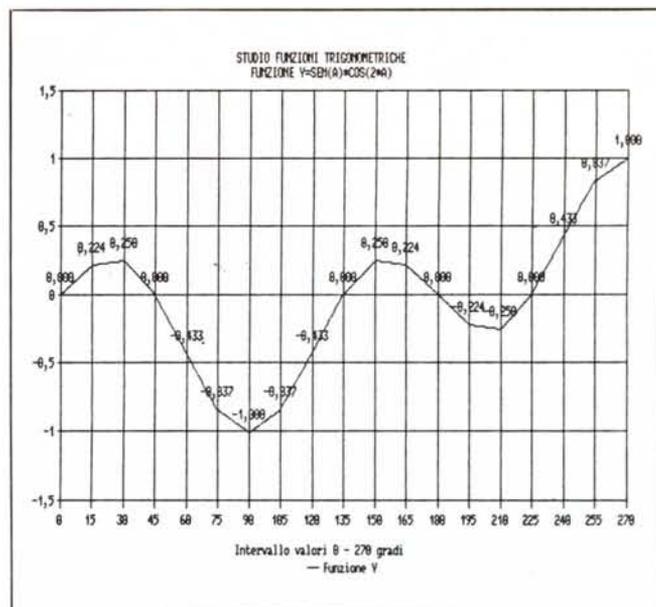


Figura 4 - Trigonometria - Intervallo (secondo tempo). Al disegno minimo indispensabile aggiungiamo una serie di abbellimenti permessi dai comandi opzionali non indispensabili.

formato di uscita, semplicemente attribuendoglielo con le apposite funzionalità.

Ma il concetto di cella è ancora più estensivo, in una cella può essere inserita una formula, logica, matematica e/o di stringa, anche di grande complessità, che in un linguaggio tradizionale può ad esempio costituire una routine di numerose righe di istruzione. La cella diventa in questo modo routine o subroutine di calcolo, aiutata in questo dall'esistenza di poderose funzioni di calcolo.

Nostro obiettivo, e più genericamente obiettivo di questa serie di articoli, è quello di esemplificare soluzioni applicative di funzionalità presenti nei vari tabelloni elettronici, eventualmente riferendoci anche alla stessa problematica risolta, diciamo, in modo tradizionale.

Ci rivolgiamo soprattutto a utenti di tabelloni agli «inizi della carriera», ma speriamo che qualche consiglio risulti utile anche ad utenti esperti.

Per quanto riguarda i prodotti, useremo tutti quelli più diffusi sul mercato, sottolineando però ancora una volta che lo spreadsheet è rappresentato da una categoria di prodotti, la cui filosofia di uso è assolutamente la stessa, e quindi una soluzione che val per uno, in genere vale per tutti gli altri.

I problemi che tratteremo in questo numero sono relativi all'uso di zone di tabellone per l'esecuzione di routine di calcolo particolari. In tutte le applicazioni infatti occorre utilizzare routine esterne per eseguire calcoli ripetitivi, oppure per memorizzare tabelle di

codifica di certi elementi ripetitivi in un archivio.

Con i linguaggi tradizionali si usano vettori e matrici, opportunamente indicizzate. Con gli spreadsheet si ricostruiscono gli stessi concetti utilizzando porzioni di tabellone. Ad esempio il contenuto di una zona di 20 per 4 caselle (80 caselle) è del tutto analogo al contenuto di una matrice 20 per 4.

Ogni tipo di tabellone elettronico infatti possiede funzioni specifiche che permettono di manipolare, per mezzo di indici, tale tabella.

### Tabella pitagorica in «Stringa»

Facciamo un primo esercizio utilizzando il Multiplan 2.0 della Microsoft, di cui abbiamo pubblicato, nel numero scorso di MC, la prova.

L'obiettivo è quello di scrivere in due caselle contigue due numeri, e di ottenere nella casella successiva il risultato dell'operazione per esteso. Ad esempio 6 per 7 quarantadue.

Va da sé che l'esercizio non ha nessuna utilità pratica, serve solo per estrarre, per mezzo di due indici, un certo elemento da una tabella rettangolare. Come si vede dalla figura 1, la tabella, che in generale deve essere posta in un'area nascosta del tabellone, riproduce la tavola pitagorica, solo che i vari numeri non sono indicati in cifre ma in lettere.

La funzione che utilizziamo è:

INDICE (AREA;n1;n2)

Dove AREA indica le coordinate di riferimento dell'ubicazione della tabella, n1 la riga e n2 la colonna.

La funzione si può usare sempre in varie maniere. La più diretta è sempre quella di indicare al suo interno coordinate dell'area e valori numerici.

In tal caso si scriverebbe:

INDICE (r1c1:r10c10;7;6)

Tale sistema non conviene perché ogni volta che si vuol cambiare un elemento si deve riscrivere, almeno parzialmente, la formula. Quindi si cerca di tirare fuori dalla formula tutti gli elementi diretti.

L'area si può battezzare tramite la funzione NOME CREA, che permette di attribuire ad una certa zona un nome proprio, molto più facile da maneggiare. Poi, invece dei valori riga e colonna, si inseriscono i riferimenti di due caselle entro le quali immettere i valori di riga e colonna desiderati.

Nel nostro caso abbiamo posto:

in R15C3 primo fattore  
in R15C4 secondo fattore  
in R15C5 la formula INDICE (PITAGORA;  
R15C3; R15C4) con il risultato

Un passo successivo consiste nell'utilizzare il NOME CREA anche per le caselle dei fattori, in tal caso la formula può perdere tutti i riferimenti diretti, diventando:

INDICE (PITAGORA; FATT1; FATT2)

Non c'è limite alla sofisticazione raggiungibile, dipende sia dal grado di conoscenza che si ha dello strumento, che dal tipo di utilizzazione che si deve fare.

Nel nostro caso le caselle R15C3 e R15C4, in cui si immettono i fattori di calcolo, accettano valori numerici. Si potrebbe sofisticare l'input accettando anche valori alfabetici.

Uno dei sistemi più immediati, senza così ricorrere a funzioni di stringa, è quello di creare una serie di 10 celle, contenenti i valori numerici da 1 a 10, e battezzate ciascuna con il rispettivo valore scritto in esteso.

Per cui ad esempio se nella casella R10C1 c'è il valore 3, ed è battezzata «TRE», e nella casella R20C3 digitiamo la funzionalità valore e immettiamo la stringa «TRE», la casella assume e visualizza il valore numerico 3.

La possibilità di attribuire ad una cella od a un insieme di celle dei nomi, costituisce uno dei maggiori vantaggi operativi nell'uso dei tabelloni, in quanto permette di scrivere le formule dell'applicazione riferendosi a nomi propri mnemonici anziché ad astruse coordinate.

Ricordiamo inoltre che qualsiasi operazione si esegua sul tabellone, al di là del riempimento di una singola cella, come spostamenti, inserimenti di righe e colonne, eliminazioni di righe o colonne, ecc. nessuna delle formule scritte viene «persa» in quanto è lo strumento stesso che «provvede» a modificare i riferimenti delle formule in relazione alla nuova situazione.

L'unico pericolo di provocare condizioni di errore sta nell'eliminazione di righe e colonne, nel caso che qualche formula, posta in altre parti del tabellone, faccia riferimento ad elementi poi eliminati. Ma evidentemente questo è un errore grossolano la cui colpa va attribuita all'operatore.

## Trigonometria

Supponiamo di avere un tabellone che esegua numerosi calcoli trigonometrici, e che tali calcoli riguardino sempre gli stessi angoli.

Nei linguaggi tipo BASIC è molto più rapido l'accesso ad una matrice, di quanto non sia veloce il calcolo di funzioni trigonometriche, per cui in casi del genere, per velocizzare l'esecuzione, si costruisce una matrice di valori trigonometrici precalcolati, e nelle formule si sostituiscono le funzioni trigonometriche con richiami a matrici. Ad esempio:

invece di  $H \cdot \text{SEN}(X)$

$H \cdot S(Y)$

in cui la Y è quel termine che «pesca» nella tabella il valore  $\text{SEN}(X)$ .

Realizzare relazioni di questo genere è abbastanza facile se si utilizza come indice della matrice il valore dell'angolo in gradi, per cui i valori di X e Y coincidono. Cioè  $S(45)$  è il 45.mo elemento del vettore S(I), cui corrisponde il valore del seno di 45 gradi.

Cerchiamo di fare la stessa cosa con un tabellone elettronico, in questo caso il LOTUS 123 versione 2.0 italiana. Usiamo il Lotus perché useremo una funzionalità particolare che in questo esercizio fa comodo.

Nella figura 2 vediamo l'esercizio. In alto per comodità abbiamo anche scritto le formule che stanno sotto le varie caselle, e che altrimenti non si vedono perché nella casella appare il risultato. In basso abbiamo, nelle prime tre colonne (A-C) i valori angolari in gradi, il corrispondente valore del seno e del coseno.

Per calcolare tali due funzioni, che al solito lavorano su angoli espressi in radianti, occorre tradurre l'angolo in gradi, utilizzando la solita formula in cui entra anche il valore @PI, pi greco.

La tabella (=matrice) dalla quale prelevare i valori seno e coseno è quindi costituita dalle prime tre colonne, la prima delle quali è la chiave di ricerca e le altre due comprendono i dati da ricercare. Va notato che le funzioni di ricerca estraggono il valore delle caselle, indipendentemente dal fatto che esso sia un valore diretto o il risultato di una formula.

Costruita la tabella occorre realizzare l'applicazione che la utilizzi. Nel nostro caso vogliamo calcolare la formula:

$$Y = \text{SEN}(A) \cdot \text{COS}(2 \cdot A)$$

Quindi dopo aver incolonnato una serie dei valori che ci interessa calcolare (colonna E), seguiamo le due strade di calcolo.

La prima, rappresentata dalla colonna G, è il calcolo diretto mediante inserimento nella formula delle funzioni seno e coseno. La seconda è il calcolo indiretto mediante «prelievo»

dei valori trigonometrici dalla tabella (colonne A-C, battezzata TRG).

In questo secondo caso occorre «superare» un'altra difficoltà, costituita dal fatto che mentre le funzioni trigonometriche valgono per qualsiasi angolo, anche superiore cioè a 360 gradi, non è pensabile di realizzare una tabella infinita che comprenda tutti gli angoli possibili.

La soluzione a questa difficoltà è quella di riportare qualsiasi angolo superiore a 360 gradi al valore inferiore corrispondente. Tale risultato si ottiene utilizzando la funzione @MOD (A;360) che fornisce il resto della divisione dell'angolo A per il valore 360.

Nel Lotus 2.0 esiste la funzionalità ZONA VALORI con la quale è possibile trasformare il contenuto di zone dalle formule ai corrispondenti valori. È evidente che questa funzionalità produce un notevole «risparmio» di memoria, e di velocità di ricalcolo, ma è utilizzabile solo per quelle formule che non debbono essere più modificate.

In sostanza, nel nostro caso, utilizziamo delle formule per eseguire dei calcoli, poi traduciamo i risultati nei corrispondenti valori numerici, perdendo del tutto la possibilità di risalire al modo con cui tali valori sono stati prodotti.

Prima di passare ad un nuovo argomento facciamo un intervallo costituito dalla graficizzazione della curva costituita dai valori calcolati. Vogliamo eseguire i comandi minimi indispensabili per realizzare il disegno, e vogliamo contarli.

Ci posizioniamo sulla casella E10. Premiamo il tasto comandi >/<, scegliamo il comando Grafico >G<. Non opzioniamo il tipo perché ci interessa il tipo di default (Diagramma Li-

B4: [L12] @SCANSVER(B3;PROVINCIE;1)										FRONTO
	A	B	C	D	E	F	G	H	I	
1				Tabella Provincie						
2				Sigla Città	Num.					
3	SIGLA	MI		AN	ANCONA	7				
4	PROVINCIA	MILANO		AR	AREZZO	8				
5	REGIONE	LIGURIA		BO	BOLOGNA	7				
6				FI	FIRENZE	8				
7				MI	MILANO	3				
8				GE	GENOVA	2				
9				RM	ROMA	11				
10				TO	TORINO	1				
11				AO	AOSTA	0				
12										
13										
14										
15										
16	SIGLA	MI								
17	PROVINCIA	@SCANSVER(B3;D3..F11;1)								
18	REGIONE	@SCELTA(@SCANSVER(B3;D3..F11;2);"VALLE AOSTA";"PIEMONTE";"LIGU								
19										
20										
01/01/80										

Figura 5 - Problemi di Scodifica - Scodifica Semplice. Esempificazione di come si risolve, con un tabellone elettronico, un classico problema di codifica e scodifica. L'esercizio è il solito delle sigle automobilistiche dei capoluoghi di provincia, perché comprensibile da tutti.

B5: [L12] @SCANSVER(@SCANSVER(B3;PROVINCIE;2);REGIONI;1) FRONTO

	A	B	C	D	E	F	G	H	I
1				Tabella Provincie				Tabella Regioni	
2				Sigla	Citta'	Num.		Num.Regione	
3	SIGLA	RM		AN	ANCONA	10		1 VALLE AOSTA	
4	PROVINCIA	ROMA		AR	AREZZO	9		2 PIEMONTE	
5	REGIONE	LAZIO		BO	BOLIGNA	8		3 LIGURIA	
6				FI	FIRENZE	9		4 LOMBARDIA	
7				MI	MILANO	4		5 TRENTINO	
8				PE	PESCARA	13		6 FRIULI	
9				RM	ROMA	12		7 VENETO	
10				TO	TORINO	2		8 EMILIA	
11								9 TOSCANA	
12								10 MARCHE	
13								11 UMBRIA	
14								12 LAZIO	
15								13 ABRUZZI	
16								14 MOLISE	
17	SIGLA	RM							
18	PROVINCIA	@SCANSVER(B3;D3;.F10;1)							
19	REGIONE	@SCANSVER(@SCANSVER(B3;D3;.F10;2);H3;.I16;1)							
20									
21	01/01/80	00:02							

Figura 6 - Problemi di scodifica - Scodifica Doppia. Un esempio un po' più complesso che permette di verificare la possibilità di costruire delle formule complesse, i cui parametri siano a loro volta formule complesse, e così via. Ci appoggiamo all'esempio precedente.

nea). Opzioniamo la grandezza X, e quindi premiamo >X< e poi illuminiamo la zona interessata premendo in sequenza >.< >END< >Freccia Giù< >Return<. Opzioniamo la grandezza A premendo in sequenza >A< >Freccia a destra<. >.< >END< >Freccia Giù< >Return<.

Per visualizzare premiamo >V<. Vediamo il risultato in figura 3. In pratica per ottenere il grafico minimo occorre premere una dozzina di tasti, operazione che un utente pratico esegue in uno o due secondi. In figura 4 vediamo un ulteriore abbellimento ottenuto aggiungendo ai comandi minimi indispensabile anche un po' di quelli opzionali.

### Problemi di scodifica

Altro caposaldo del trattamento dei dati è la cosiddetta codifica dei dati ripetitivi, tramite la quale si ottengono in generale due «piccioni con una fava». Il primo piccione sta nel grande risparmio di memoria, il secondo vantaggio sta nel fatto che il campo codificato diventa facilmente oggetto di statistiche, la più elementare delle quali è il conteggio della ricorrenza di ciascun codice.

Esemplifichiamo questo discorso così astruso e per entrare nello spirito dell'esercizio, supponendo di avere un archivio con migliaia di indirizzi (in Italia). Possiamo codificare la provincia utilizzando le sigle automobilistiche. In tal modo per specificare ciascuna provincia sono sufficienti due caratteri anziché quindici. Inoltre possiamo contare/estrarre con maggiore facilità gli indirizzi relativi ad una certa provincia, in quanto tutte le operazioni di confronto saranno eseguite su

stringhe di due caratteri.

In LOTUS 2.0 Italiano la funzione su cui si basa tale ricerca è la SCAN-SVER (Chiave; Area; Colonna) simile a quella vista per il Multiplan ma che accetta non due indici che specificano riga e colonna, ma una Chiave che viene ricercata nella prima colonna della tabella, e che quindi identifica la riga. Area e Colonna hanno i significati consueti.

Abbiamo già visto, in uno dei precedenti articoli, l'uso di tale funzione ha due comportamenti differenti a seconda che la chiave di entrata sia numerica o alfanumerica. Se è numerica, data una chiave, la riga identificata è quella che ha il valore uguale o minore a quello ricercato, mentre la riga successiva ha un valore maggiore. In sostanza la colonna di valori da ricercare è intesa come una serie di range.

Al contrario, in caso di chiave alfabetica, la ricerca avviene per condizione di uguale. Quindi se la funzione @SCANSVER ricerca una stringa che non trova come chiave, si genera una condizione di non chiave non trovata.

Quindi se la prima colonna della tabella è alfabetica si può realizzare una vera e propria routine di scodifica utilizzando la funzione @SCANSVER ed appoggiandosi ad una matrice di scodifica.

Come al solito l'esempio (figura 5) non ha alcuna finalità pratica, se non l'apprendimento della tecnica di codifica.

Occorre, data una tabella di sigle di capoluoghi di provincia scodificare una sigla in input nel nome della città corrispondente. Il secondo obiettivo è quello di scodificare anche il codice regione, tramite il quale data una provincia è anche individuata una specifica regione.

Oltre allo Scansver, nel primo esercizio usiamo anche la funzione @SCELTA (Num;r0;r1;...;rn) con la quale si può realizzare una scodifica in una sola casella. La sintassi della Scelta accetta un valore numerico Num, intero e compreso tra 0 e N. In corrispondenza di tale valore la funzione restituisce la risposta rx, che occupa ovviamente la Num-sima posizione nella lista.

Tale funzione può essere utilizzata a condizione che il codice sia numerico e che i vari casi siano tutti inseribili in unica formula, che ha un limite superiore di 240 caratteri.

È evidente il vantaggio di poter inserire tutta la tematica di scodifica in unica casella, magari complicando la relativa formula.

Nel nostro caso abbiamo attribuito a ciascuna regione un valore numerico. Tale numero viene però inserito nella tabella provincie, come seconda colonna.

In definitiva data una sigla, con un primo Scansver viene scodificata la città relativa, con un secondo Scansver, sulla seconda colonna, viene identificato il codice numerico della regione, che non viene utilizzato direttamente ma viene inserito come chiave nella funzione Scelta (il tutto con un minimo di pratica è più semplice a farsi che a dirsi).

Nella figura 5 si notano in alto a destra la tabella delle sigle, a sinistra le caselle B2 per l'input della sigla, B3 per l'output del nome della città, e B4 per l'output del nome della regione.

In basso sono semplicemente trascritte le formule che si devono trovare «sotto» le citate caselle.

Ovviamente si potevano usare due tabelle sulle quali eseguire due operazioni di scansione collegate tra di loro (vedi figura 6). Anzi questa soluzione è da preferire quando occorre eseguire parecchie scodifiche in quanto la funzione Scansver «occupa» di meno, in termini di occupazione di spazio sul tabellone, della Scelta.

Abbiamo detto occupazione in termini di spazio anziché in termini di byte perché nel lavoro con lo spreadsheet si perde del tutto il concetto di byte, che anzi nel manuale non è neanche citato. Ovviamente da un punto di vista tecnico, è sempre possibile vedere l'applicazione in termini di mappe di memoria, puntatori, variabili, ecc. ma questo è un discorso che in genere non deve interessare l'utente medio.

Esistono comunque regole d'uso o meglio suggerimenti per l'uso, assolutamente non obbligatori, che però permettono di migliorare le prestazioni dello strumento e che tratteremo prosimamente.



## Mannesmann Tally: il fascino della semplicità.

C'è ancora qualcuno che non conosce tutta la grande famiglia delle stampanti MANNESMANN TALLY? Può darsi.

Perché nella grande famiglia di stampanti MANNESMANN TALLY le novità sono all'ordine del giorno... nuovi prodotti, versioni aggiornate, accessori che migliorano i modelli più collaudati. Oggi a SILVERSTAR potete chiedere subito qualsiasi modello di stampante MANNESMANN TALLY, dalla MT 80 alla MT 490, fino alla prestigiosa MT 660/690 e la nuovissima stampante laser MT 910 sono tutte disponibili a magazzino.

Ma non è tutto.

SILVERSTAR è in grado di fornirle, in esclusiva, molto di più: unità di emulazione che consentono di collegarsi con qualsiasi sistema.

Con questo servizio, qualunque sia il modello di stampante MANNESMANN TALLY in suo possesso, può interfacciarsi come vuole, espandersi giorno dopo giorno, secondo le crescenti esigenze della sua attività.

Ci telefoni. Per lavorare NON-STOP!

**SILVERSTAR**  
ELETTRONICA

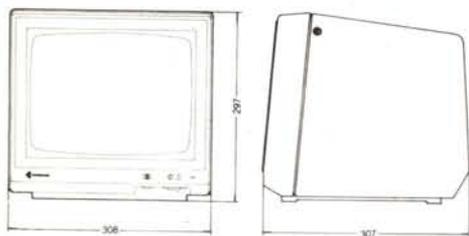
Via Dei Gracchi, 20 - 20146 MILANO - Tel. (02) 49.96

# CRYSTAL

## MONITORS



CASELLA POSTALE 142  
56025 PONTEDERA (PI)  
VIA MISERICORDIA, 84  
TEL. 0587 - 212.312



La differenza c'è!! E si vede.  
Certo!, non tutti i monitor 12'' sono uguali e per questo ti chiediamo di fare una prova confrontando la risoluzione al centro ed ai bordi di un CRYSTAL contro un altro qualsiasi monitor TTL per IBM PC.



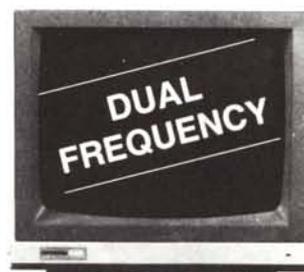
**FOSFORI VERDI (P39)**  
Freq. orizzontale 18,4 KHz  
Banda passante 16 MHz  
Ingresso: TTL  
Risoluzione: 1000 linee  
Uso: per schede IBM tipo Hercules



**FOSFORI AMBRA (PLA)**  
Freq. orizzontale 18 KHz  
Banda passante 16 MHz  
Ingresso: TTL  
Risoluzione 1000 linee  
Uso: per schede IBM tipo Hercules

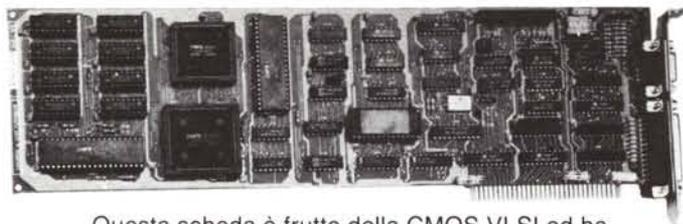


**FOSFORI BIANCHI (WD)**  
Freq. orizzontale 18,4 KHz  
Banda passante 16 MHz  
Ingresso: TTL  
Risoluzione: 1000 linee  
Uso: per schede IBM tipo Hercules



**FOSFORI P42**  
Doppia frequenza orizzontale 15,7-18,4 KHz  
Ingressi: TTL e Composito  
Risoluzione: 800 linee  
Uso: per schede tipo Hercules e Color, per Apple computer ecc.

### SUPER E.G.A. CARD



Questa scheda è frutto della CMOS-VLSI ed ha una tecnologia di larga applicazione in quanto, oltre ad emulare la Enhanced Graphics Adapter, emula anche la Color Graphics Adapter e la Hercules.

### Graphics Modes:

N° dei colori	Pixels (H x V)	Pagine (max)	Tipo di monitor*	Adattatori emulati**
16	320 x 200	8	CD/ECD	EGA
16	640 x 200	4	CD/ECD	EGA
2	640 x 350	1	MD	EGA
2	640 x 350	2	MD	EGA
4/64	640 x 350	1	ECD	EGA
16/64	640 x 350	2	ECD	EGA
4	320 x 200	1	CD/ECD	CGA, EGA
2	320 x 200	1	CD/ECD	CGA, EGA
2	640 x 200	1	CD/ECD	CGA, EGA
2	720 x 348	2	MD	HGC

### Alphanumeric Modes:

N° dei colori	Caratteri x linee	Pagine (max)	Tipo di monitor*	Adattatori emulati**
16/64	80 x 25	2	ECD	EGA
16/64	80 x 43	8	ECD	EGA
2	80 x 43	4	MD	MDA/EGA
2	40 x 25	8	CD/ECD	CGA, EGA
16	40 x 25	8	CD/ECD	CGA, EGA
2	80 x 25	8	CD/ECD	CGA, EGA
16	80 x 25	8	CD/ECD	CGA, EGA
2	80 x 25	8	MD	MDA, HGC, EGA

\* ECD = Enhanced Color Display, CD = Color Display, MD = Monochrome Display.

\*\* EGA = Enhanced Graphics Adapter, CGA = Color Graphics Adapter, MDA = Monochrome Display Adapter, HGC = Hercules Graphics Card

### MONITOR



Monitor ad alta risoluzione (DOT PITCH: 0,31 mm) consigliato con IBM PC/XT/AT per l'uso della ENHANCED GRAPHICS ADAPTER o della COLOR GRAPHICS ADAPTER.

Scanning automatico della frequenza orizzontale (15,75-21,85 KHz) per entrambi i modi grafici 640 x 350 e 640 x 200.

Nel modo 640 x 350 possono essere usati più di 64 colori per la grafica (16 alla volta).

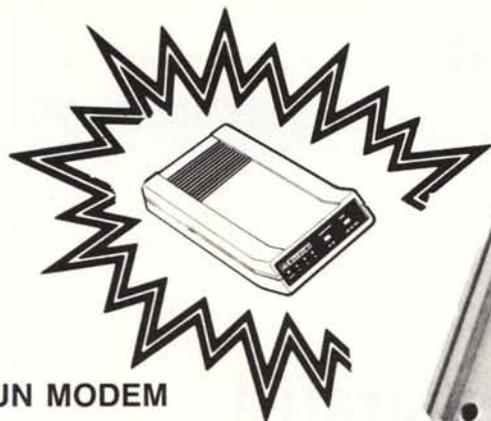
Smagnetizzazione del tubo automatica, ventilatore interno.

Possibilità di usare su tutto lo schermo solo i colori verde ed arancio (particolarmente richiesto per elaborazione di testi).

# Bondwell™



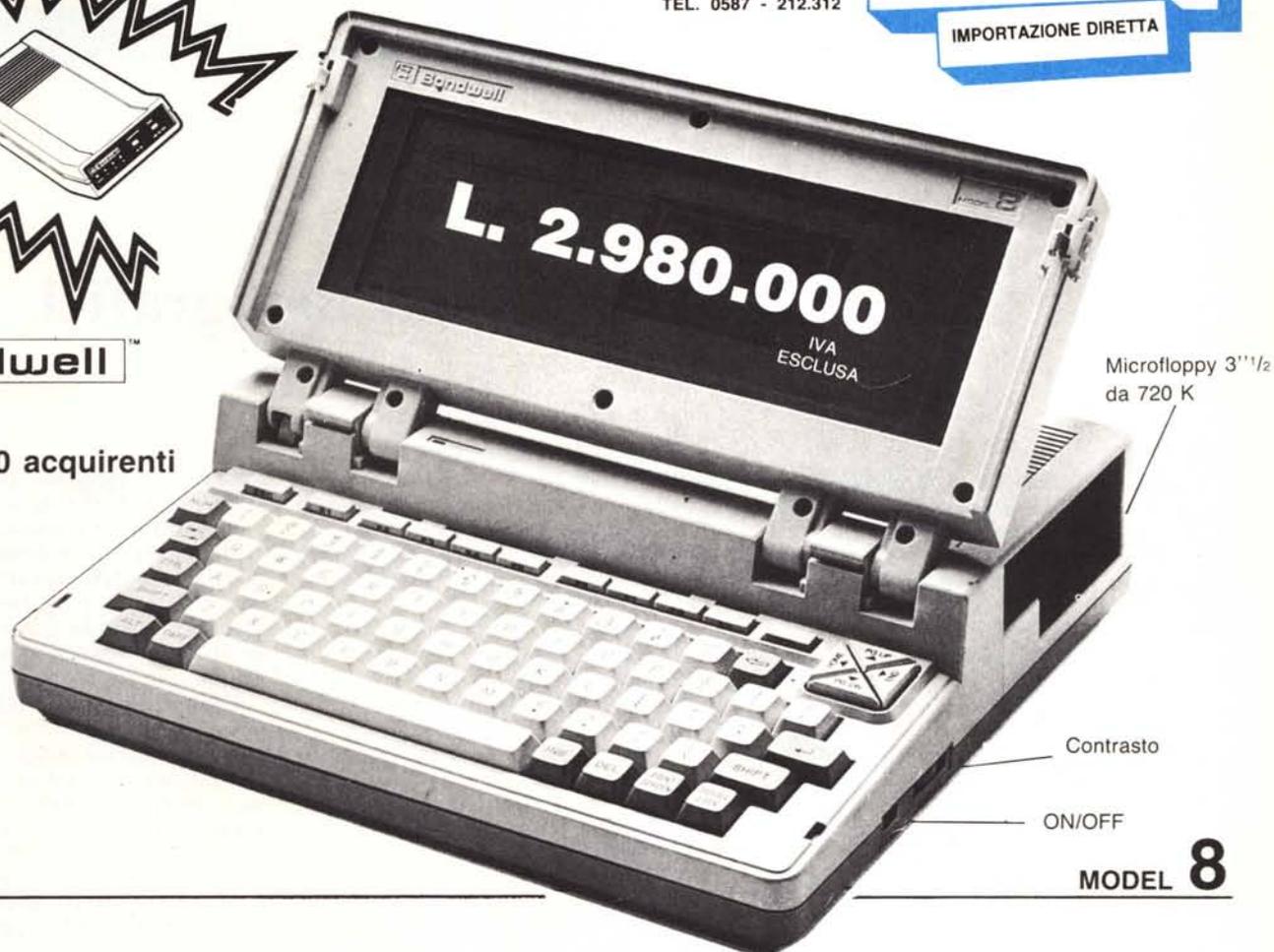
CASELLA POSTALE 142  
56025 PONTEDERA (PI)  
VIA MISERICORDIA, 84  
TEL. 0587 - 212.312



UN MODEM

**B** Bondwell™

IN REGALO  
ai primi 1.000 acquirenti



MODEL **8**

## IL PIÙ PICCOLO E POTENTE PC PORTABILE SI CHIAMA **B** Bondwell™ **8**

- Facilmente trasportabile
- Peso: Kg. 4,500
- Dimensioni: cm. 28 x 31 x 78
- IBM-PC compatibile (DOS 2.11 su licenza Microsoft)
- Dischetto con MS/DOS 2.11, GW Basic 2.0 e manuali inclusi
- Basso consumo ottenuto con l'impiego di componenti CMOS
- Microprocessore: 80C88, 4.77 MHz
- Memoria RAM: 512K
- Schermo a cristalli liquidi ad alto contrasto, illuminabile, e con risoluzione 640 x 200 (grafica), 80 x 25 (testo)
- Floppy disk interno da 3" 1/2 doppia faccia/doppia densità da 720K formattati
- Orologio/Calendario mantenuto da batterie al nichel-cadmio ricaricabili
- Batterie ricaricabili 12V-3A

- Tastiera con 76 chiavi e basso profilo, compatibile con lo standard PC/XT, dotata di funzioni del PAD numerico, 10 tasti funzione ecc. ecc.
- Porta seriale standard R-232C
- Porta parallela per stampanti
- Porta per la connessione del 2° Drive (5" 1/4 oppure 3" 1/2)
- Uscite per video RGB/TTL e video-composito.
- Led segnalatore intermittente di fine carica
- Alimentatore/Caricabatterie AC/DC
- Hard e Soft realizzati per ottenere il massimo della compatibilità IBM-PC. Possono essere eseguiti i più popolari pacchetti software come: Lotus 1-2-3, Symphony, D Base II e III, Wordstar, Flight Simulator, Framework, Jem, Sidekick, PFS serie, ...

