

Seconda parte

La costruzione dei modelli

Abbiamo visto la volta scorsa la componentistica di base del programma, e ormai dovrebbero essere anche abbastanza ovvi l'uso e le interconnessioni tra le parti. Vediamo adesso come fare per costruirci i nostri modelli, in modo tale che il programma possa fornirci risposte valide e sicure.

Generalmente, quando si affronta, con carta e penna, un problema matematico (e non), cercando di schematizzarlo in un modello poi risolvibile per parti, si segue una successione di azioni che può essere, così, schematizzata:

— determinazione del procedimen-

to matematico (e, in definitiva, scrittura delle formule, funzioni, od equazioni necessarie per la soluzione del problema), e sua riduzione in forma scritta, sia esso diagramma di flusso, schema esecutivo, o, più semplicemente, stesura delle rispettive equazioni algebriche;

— risoluzione delle formule stesse, tenendo in debito conto:

le unità di misura adottate, la loro congruenza, e le relative modalità di conversione, se necessarie;

l'uso di tabelle, schede, od altre raccolte di dati (costanti) necessarie ad

ottenere la massima informazione possibile sulla variabili stesse;

la necessità di dover ricorrere a metodi iterativi per ottenere le risposte desiderate;

— esposizione dei risultati, con tipologie diverse; ad esempio, in forma numerica, grafica, tabellare, ecc.

Sembra che le eminenze grigie di Software Art abbiano a cuore gli utenti dotati di carta e matita. TK!Solver, dopo Visicalc, è organizzato proprio in questo modo, anche se lui lavora in termini di SHEET, fogli. La sequenza appena descritta può essere, in

simbolo TK!Solver	simbolo VisiCalc	comando
B	B	BLANK
C	REPLICATE	COPY
D	D	DELETE
E	E	EDIT
I	I	INSERT
L	----	LIST (non disp. in V/C)
M	M	MOVE
P	P	PRINT
Q	SO	QUIT
R	CLEAR	RESET
S	S	STORAGE
W	W	WINDOW
!	!	SOLVE
?		HELP

◀ Figura a - Analogia tra comandi TK!Solver e Visicalc.

Figura b - Finestra di Help con relative opzioni.

▼

HELP Command Symbol : ?

Opzione	Significato
?	Consente di ottenere più dettagliate istruzioni circa l'uso di HELP
topics (argomenti)	Mostra un quadro delle risorse disponibili
Topic name (nome dell'argomento)	Argomento specifico di cui occorrono spiegazioni più dettagliate

TK!Solver così riassunta, in analogia alle fasi appena descritte:

— fase iniziale:

uso di:

GLOBAL SHEET

RULE SHEET

— determinazione delle modalità di soluzione

uso di

VARIABLE SHEET e VARIABLE SUBSHEET

UNIT SHEET

FUNCTION SHEET

USER FUNCTION SHEET

LIST SHEET (nel caso si debba ricorrere a soluzioni iterative)

— determinazione del tipo di resa dei risultati

come lista

come disegno

Come si vede TK!Solver utilizza una metodologia del tutto analoga a quella manuale per la soluzione di problemi. Anzi, a dire il vero, fin da quando si inizia ad utilizzarlo, la suddivisione in fogli, lungi dall'essere complicata o d'impiccio, consente una tenuta in ordine delle diverse parti del problema, che risulta alla fine di insostituibile chiarezza. Tanto per riferire un particolare, ho provato, successivamente alla mia prima conoscenza di TK!Solver, a risolvere alcuni problemi, adottando carta e penna, dividendo il foglio in zone, destinate ad accogliere le diverse parti del problema. Ci si ritroverà con un ordine impensato, con una facilità di controllo del problema, in caso d'errore, del tutto impensabile, e con velocità di esecuzione insospettabilmente ridotta rispetto a quella dei nostri benamati fogli protocollo zeppi di calcoli disordinati. Provare per credere.

For Help, type? su questa linea, oltre questa frase, possono comparire messaggi relativi a suggerimenti, forniti dal programma, su comandi da eseguire, od informazioni associate con l'ultimo errore in cui si è incorso.

Sull'Apple Macintosh (fig. d1), ovviamente, i risultati, a livello grafico, sono incomparabilmente migliori; l'uso delle finestre e della barre di scroll permette una organizzazione dello schermo e delle informazioni più efficiente e razionale. Ciononostante la sostanza è la stessa, e le modalità d'uso non cambiano.

Quando il programma TK!Solver viene lanciato, ci troveremo di fronte ad una schermata simile a quella delle figure d) e d1). Il programma si riferisce ai fogli individuandoli come finestre, a cui accede mediante i tasti di movimento del cursore (MS-DOS) o

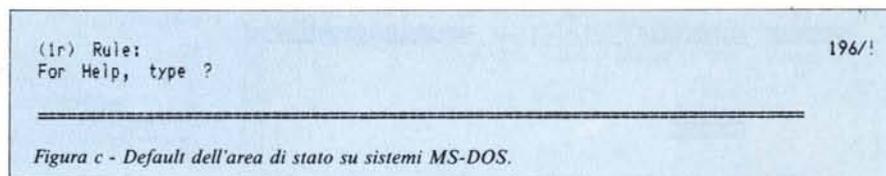


Figura c - Default dell'area di stato su sistemi MS-DOS.

tramite il mouse (su Mac). L'uso più diffuso è quello di mostrare due finestre, ognuna delle quali viene trattata indipendentemente dall'altra, e le operazioni di modifica, ridimensionamento, scrolling dell'una non influiscono in alcun modo sull'altra. Su Mac, la cosa inoltre è particolarmente facilitata dalle barre laterali di scroll. In ogni caso, comunque sia ottenuto il movimento, sia tramite l'uso dei tasti con le frecce, sia tramite il mouse, l'indicatore di posizione del cursore si aggiornerà automaticamente per indicare la nuova posizione.

Sotto questo punto di vista TK!Solver dispone di un comando molto più efficiente per spostare il cursore, che risulta molto più apprezzato quando il modello è particolarmente lungo ed ampio: il comando GOTO. Esso funziona come il classico GOTO Visicalc, ma con qualche miglioria. Vediamo le opzioni:

— quando sono presenti un numero ed una lettera (l'opzione più generale) viene indicata la precisa locazione di spostamento

— solo un numero: spostamento sulla stessa colonna, alla riga specificata (si ricordi: il numero indica la riga, la lettera la colonna, e la stessa lettera fa riferimento all'iniziale della colonna riferita; così [r] sta per [rule], o [s] sta per [status] od [st], ad esempio)

— solo lettera: spostamento sulla stessa riga, alla colonna che inizia con la lettera

— [*]: spostamento al fondo della stessa colonna

— [*, seguito da una lettera (si ricordi: le parentesi quadre sono qui inserite solo per evidenziare che si tratta di una specifica lettera da battere alla tastiera, e non vanno quindi incluse nel comando); spostamento all'estremo della stessa riga.

— ['], seguito da una serie di lettere:

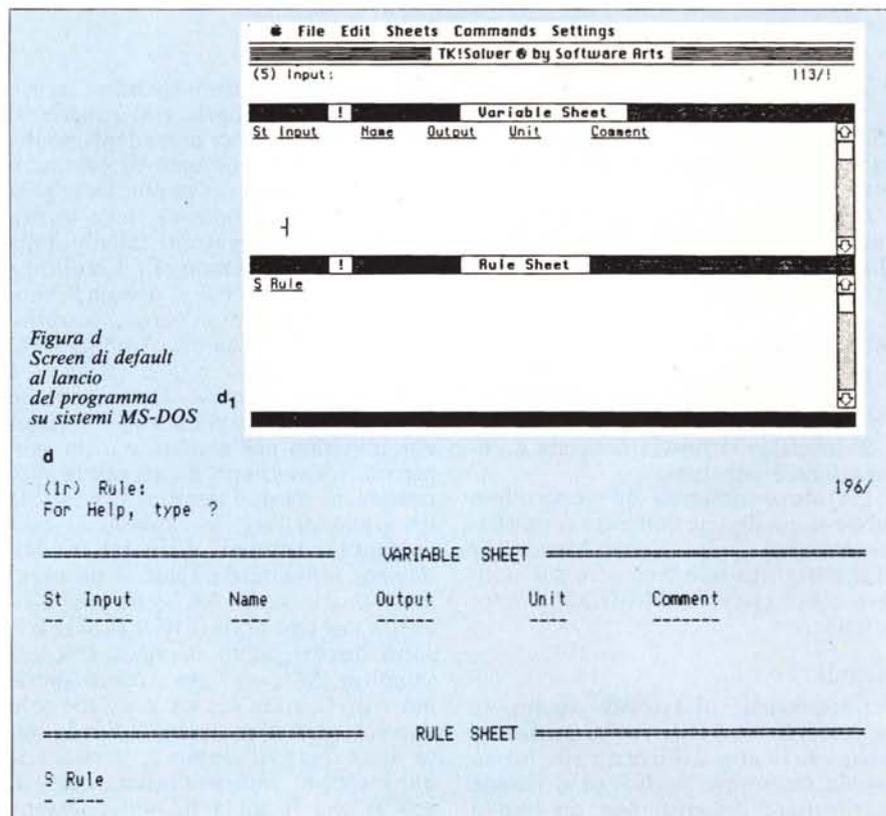


Figura d
Screen di default
al lancio
del programma
su sistemi MS-DOS

d

(1r) Rule:
For Help, type ?

196/

VARIABLE SHEET

St Input	Name	Output	Unit	Comment
---	---	---	---	---

RULE SHEET

S Rule

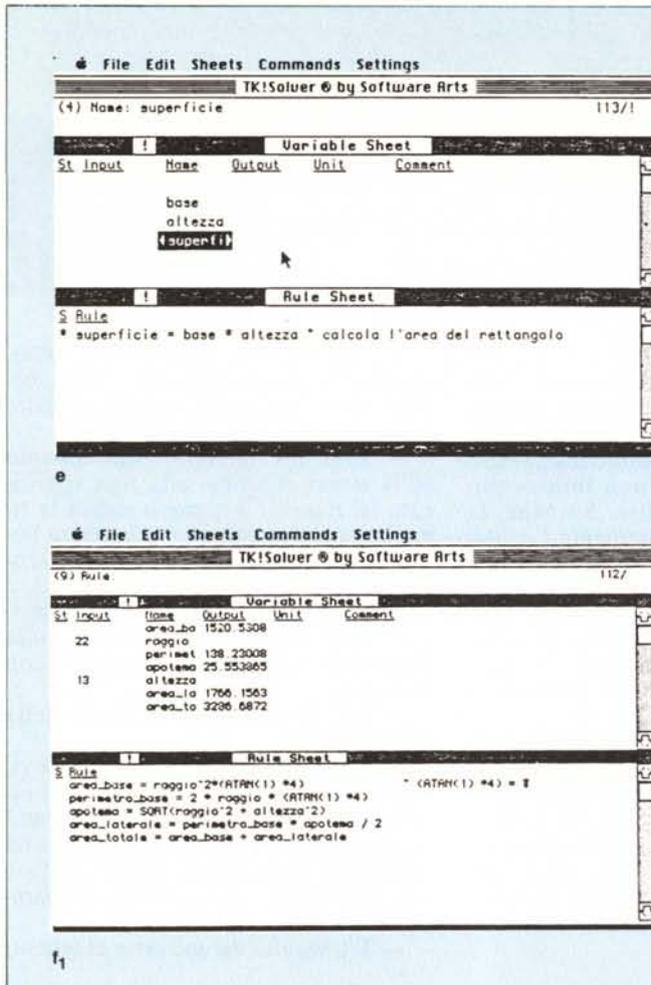


Figura e - Introduzione di una semplice formula. Si noti come, nel foglio variabili, la capienza della colonna NAME, di sole 7 lettere, «tagli» il nome della variabile, che, comunque, può essere letta utilizzando i triangoli di scrolling laterale, e che compare sempre, per intero, nella status area.

Figura f1) - Calcolo della superficie totale di un cono. Si noti che, al contrario di quanto suggerirebbero le apparenze, non è necessario esprimere le formule in forma canonica. Vale a dire che (si veda la figura F2) il programma provvede ad estrarre dalle formule fornite tutti gli elementi che gli servono per la risoluzione delle incognite, se possibile.

si tratta di un comando più sofisticato di finding e searching; consente di effettuare ricerche, sulla stessa colonna, dell'argomento seguente le virgolette.

Il comando GOTO è efficiente in qualsiasi momento, e in qualunque foglio si stia lavorando in quel momento. Basta battere:

GOTO

Il sistema risponde:

GOTO: Destination of search

Si inserisce la nostra richiesta e l'operazione è conclusa.

Un altro comando da apprendere subito è quello che consente il passaggio tra due fogli diversi. Mentre su Mac basta spostare il cursore col mouse e schiacciare, in MS-DOS occorre battere:

SWITCH

per trasferirsi sul secondo foglio, se presente. Si noti che il salto posiziona il cursore in alto a sinistra, nella prima casella del nuovo foglio: se si decide di ritornare indietro, con un nuovo

SWITCH, il sistema «ricorda» la sua precedente posizione, e si rimette al posto che occupava precedentemente.

È arrivato il momento di partire, e facciamo un esempio: imbrighiamo la tremenda potenza della nostra macchina per eseguire il calcolo della superficie di un rettangolo. Una lunga ed estenuante ricerca in alcune biblioteche universitarie ci porta a scoprire la regola di soluzione. Andiamo col cursore nel foglio regole (RULE) e battiamo la formula. Essa segue le normali regole grammaticali e sintattiche imparate nell'algebra e nella geometria. L'equazione dovrà essere rappresentata da due termini, separati da un segno di [=]; le variabili in essa contenute saranno riconosciute dall'essere separate da spazi o da segni matematici, come [+], [*], [(], [sin] e gli altri a noi ben noti. Il RULE SHEET, sotto questo punto di vista, è il più semplice da usare: esso, come abbiamo visto la volta scorsa, possiede solo due colonne; la prima [S] indica lo stato della riga: ad esempio, se contiene un asterisco indica l'inserimento di una nuova formula od un'equazione

non ancora risolta; una parentesi tonda chiusa, [)], indica che un tentativo di soluzione ha trovato un errore nell'equazione.

Nel foglio formule (RULE) l'equazione o l'espressione va inserita secondo le più pure regole e precedenze stabilite dall'algebra. La formula, accuratamente trafugata, dell'area del rettangolo sarà scritta, nella colonna [Rule] del RULE SHEET, come più ci aggraderà (si ricordi solo di lavorare anche con la finestra delle variabili aperta). Battiamo, quindi

superficie = base * altezza

Qualche piccola cosa da ricordare: si possono usare i nomi che si preferiscono, purché siano tutti interi. Perciò non va bene [superficie-del-rettangolo], ma [superficie—del—rettangolo] sì! Inoltre la lunghezza massima della formula non può superare i 200 caratteri (MS-DOS); è possibile inserire, in coda alla equazione, dei commenti, separati dalla formula stessa da virgolette ["]].

Inserita correttamente la formula (l'uso di tasti di BACKSPACE e BREAK

è del tutto analogo a quello dei Word Processor) e dato il RETURN succede qualcosa nel foglio Variabili: il programma interpreta la formula, ne estrae i nomi delle variabili e li colloca nel foglio superiore. Lo schermo si presenterà nella forma che vediamo in figura e).

Nella figura f2) vediamo come è possibile inserire, in successione, una serie di formule, tra loro collegate, atte a risolvere un problemino di geometria un po' più complicato. Non esiste alcun problema di interpretazione: il valore di (ATAN (1) *4) non è altro che il valore di π approssimato alla dodicesima cifra (1 rappresenta radianti). Il foglio variabili non presenta grandi difficoltà, tranne, forse, il significato della colonna di status [St], il cui significato è stato riassunto la volta scorsa, ma che non è male ricordare adesso, in fase di sperimentazione. Diremo, quindi che, in questa colonna, la lettera[B]lank resetta il valore della variabile, [G]uess individua un valore inserito come limite da usare in una soluzione iterativa, [I]nput «fissa» la variabile come valore da assegnare in input, con [O]utput accade il contrario, e, infine, [L]ist definisce la variabile associata con una lista. Delle altre colonne, [Input] indica variabili assegnate da chi usa il programma, [Output] indica risultati delle operazioni,

[Unit] evidenzia unità di misura e conversione (ne parleremo in seguito), ed infine vediamo la colonna [Commenti], di intuibile uso.

Nella figura f2) è rappresentata una serie di formule destinate a calcolare l'area totale di un cono: le operazioni da eseguire per impostare il disegno della maschera sono semplici ed intuitive: partendo dal RULE battiamo di seguito le formule necessarie (ogni riga può contenere una sola formula); ogni volta che si batte il [RETURN] i nomi delle nuove variabili inserite compaiono automaticamente nel foglio Variabili (è questo, oltre tutto, un buon sistema per controllare che si siano fatti errori di battitura). Esaurita la lista delle formule proviamo a schiacciare il segno [!] (che in Mac è possibile fare con il mouse). Non succede nulla, tranne un segnale di attenzione: il programma non riesce a risolvere un sistema d'equazioni in cui non conosce alcuna variabile! Passiamo al foglio variabili. Ricordi sbiaditi della scuola media e della nostra compagna di banco, oggi mamma di tre figli, ci fanno intendere che, per la soluzione del problema, occorre conoscere almeno due delle dimensioni caratteristiche del cono. Nel foglio variabili, nella colonna [Input] battiamo i relativi valori (se tentassimo di scrivere nella colonna [Output] avremmo solo un

messaggio d'errore). Riprovando la soluzione, col tipico sfarfallio, nei campi, di Visicalc, compare. Come si vede nella prima riga, nel foglio formule è possibile inserire una serie di funzioni già predefinite: l'elenco completo delle funzioni riconosciute dal sistema è in figura g).

La nostra griglia funziona, come si vede dalla congruenza dei risultati. Prima che tentativi grossolani guastino il nostro capolavoro di geometria proviamo ad usare il comando [S]ave. Questo ordine permette di salvare, assegnandogli un nome, il disegno di base che abbiamo appena costruito (incidentalmente conserva anche i dati in esso contenuti, ma si tratta di un particolare trascurabile, visto che, alla nuova riapertura ed alla successiva fase di ricalcolo i vecchi parametri spariranno per far posto ai nuovi). Esiste, in verità un altro comando per uscire dal programma [Q]uit, ma la spiacevole caratteristica di non registrare il contenuto e l'aggiornamento eventuale del modello appena costruito. Per essere più precisi il sistema informa che si sta lasciando una griglia non salvata su disco e permette di ritornare al programma normale per eseguire le desiderate registrazioni (in Mac il solito menu in finestra abbrevia e semplifica, come al solito, molto le operazioni), ma, ormai, è ben piccola barriera, come si sa, alle abitudini cattive di battere continuamente tasti in sequenza senza preoccuparsi di guardar bene cosa si fa effettivamente. A chi scrive (e penso a molti di voi) è capitato sovente di mangiarsi le mani fino ai polsi per aver risposto meccanicamente a richieste del sistema in maniera bovina e supina!

Prima di sospendere per questa puntata vedremo, ancora, un utile comando: [E]dit. Come è facilmente intuibile, si tratta di un comando che consente, come se si lavorasse con un WP od un editor di un linguaggio, di eseguire modifiche nelle finestre. In Mac la cosa è abbastanza rapida, in quanto l'operazione funziona puntando, «cliccando», e, eventualmente, con operazioni di taglio ed incollaggio. In MS-DOS e su macchine HP la cosa è un tantino più noiosa, affidata com'è al tasto di Backspace ed alle frecce direzionali. Si tratta, comunque, sempre di operazioni abbastanza intuitive, che abbisognano di poca pratica, anche perché, ovviamente, non si stanno manipolando grossi volumi di scrittura.

Anche per stavolta sospendiamo, data la lunghezza del dire! La prossima volta entreremo nelle «features» più complesse del nostro programma; a risentirci!

ABS(x)	valore assoluto
ACOS(x)	arcocoseno
ACOSH(x)	arcocoseno iperbolico
APPLY(funzione, area)	applica una funzione ad un'area
ASIN(x)	arcoseno
ATAN(x)	arcotangente
ATANH(x)	arcotangente iperbolica
COS(x)	coseno
COSH(x)	coseno iperbolico
COUNT(lista)	conta gli elementi in una lista
DOT(lista, lista)	moltiplica tra di loro gli elementi di una lista
E()	base dei logaritmi neperiani
ELEMENT()	numero degli elementi utilizzati
EXP(x)	esponenziali (E elevato alla potenza di x)
GIVEN(variabile, x, y)	valuta le variabili come valori di input
INT(x)	funzione intero
LN(x)	logaritmo naturale (neperiano)
LOG(x)	logaritmo (base 10)
MAX(serie o lista)	valore massimo nella serie
MIN(serie o lista)	valore minimo nella serie
MOD(a, b)	valore modulo (divisione intera)
NPV(x, serie)	valore presente netto
PI()	valore di pi greco
POLY(x, serie)	funzione polinomiale
SGN(x)	funzione segno
SIN(x)	funzione seno
SINH(x)	seno iperbolico
SQRT(x)	radice quadrata
STEP(x, y)	funzione passo
SUM(serie)	funzione somma elementi della serie
TAN(x)	tangente
TANH(x)	tangente iperbolica

Figura g
Elenco delle funzioni disponibili in TK!Solver; il sistema riconosce funzioni scritte in lettere sia minuscole che maiuscole, anche frammiste tra loro.