

Un po' di pratica con lo Spreadsheet

Vi ricordate la recente polemica sulla lunghezza dei listati dei programmi pubblicati da MCmicrocomputer? Il problema è oggi fortunatamente superato grazie alla possibilità o di richiedere direttamente i programmi su supporto magnetico o di trasferirli sul proprio computer via MC-Link. Sfogliando riviste concorrenti capita invece tuttora di trovare decine di pagine di listati inequivocabilmente pubblicati per aumentare lo spessore della rivista stessa. Si tratta spesso di programmi di scarso interesse, oppure allungati artificialmente con decine di righe inutili, oppure, e questa è la cosa peggiore, di programmi che presentano errori, come algoritmi risolutivi sbagliati

Il «bello» della programmazione non sta infatti nel risolvere comunque il problema, ma nel risolverlo nel modo migliore.

Intendendo per modo migliore quello in cui si sommino efficienza, efficacia ed eleganza della soluzione.

Se due programmi raggiungono la stessa soluzione ma uno impiega la metà del tempo dell'altro significa che il secondo è sbagliato, analogamente se il primo è più corto, in termini di numero di istruzioni, il secondo è anch'esso sbagliato.

Nel concetto, più soggettivo, di eleganza andrebbe inglobato il concetto di ordine del listato sorgente, in modo anche da renderne direttamente leggibile la logica, e il ricorso alle funzionalità evolute del linguaggio, non tanto quelle complicate, quanto quelle che concorrono a rendere ordinato il listato stesso. Per fare un esempio banale la raccolta in ben organizzate subroutine, di tutti i calcoli ripetitivi del programma.

Linguaggi o Spreadsheet

È noto che mentre l'analisi del problema da risolvere deve essere indipendente dal linguaggio utilizzato, anzi è addirittura indipendente dall'uso di un

computer, la sua trasposizione in programma non lo può essere. Ogni linguaggio ha alcune caratteristiche che gli sono proprie che condizionano la scrittura dell'algoritmo risolutivo del problema.

Va però fatta una categorizzazione in prodotti tradizionali, linguaggi che permettono una traduzione abbastanza diretta dall'algoritmo teorico al programma, e prodotti intermedi in cui la struttura organizzativa può condizionare pesantemente la stesura dell'algoritmo.

Ad esempio un programma scritto in Cobol è traducibile abbastanza facilmente in Basic, in quanto grosso modo vanno tradotte le singole istruzioni. In ambedue i linguaggi si utilizzano variabili, vettori, matrici, ecc.

Lo stesso programma non è altrettanto facilmente traducibile in foglio Spreadsheet, in quanto variano le funzioni utilizzabili ed è l'organizzazione stessa del prodotto che rende la soluzione più obbligata.

Quindi con i prodotti software di informatica individuale, in questo caso trattiamo di Spreadsheet, il modo di risolvere i problemi diventa estremamente più diretto, al punto da rendere spesso inutile la programmazione e comunque molto più rapido il lavoro di stesura del programma.

Vogliamo in questo articolo studiare alcuni problemi che abbiano la caratteristica di essere molto noti (probabilmente a tutti) e di essere molto semplici (la cui comprensione richiede pochi minuti) e vogliamo risolverli con il tabellone elettronico.

Il nostro obiettivo è quello di verificare quello che si può fare con lo spreadsheet, in che modo, e in che modo la soluzione trovata differisca da quella realizzabile con un linguaggio tradizionale. Per verificare come il tempo di sviluppo si riduca, in certi casi, di decine di volte.

Utilizzeremo, per i nostri esperimenti, il prodotto Quattro della Borland sia perché è il più recente sia perché, pur

TABELLA DI MARCIA	PREVENTIVO						CONSUNTIVO							
	h	min	med	tot	h	min	med	tot	h	min	med	tot	perc	
Roma	0	0	0	0	0	0	0	0	0	0	0	0	0	
Roma, Civ	10	18	0	25	24,0	24,0	0	57	22,2	22,2	1	2	-1,0	-7,4
Mellarese	62	72	0	55	67,6	54,0	0	58	64,1	58,8	1	5	-2,2	-5,9
Subietino	66	130	1	8	66,0	59,1	0	52	76,2	68,4	1	3	1,3	2,2
10 Casselle	33	171	0	48	49,5	57,0	0	32	61,9	68,7	1	11	3,7	6,5
11 Cass	73	244	1	15	50,4	57,4	1	9	63,5	61,5	1	17	4,1	7,1
12 Anagni	63	387	1	30	42,0	53,4	1	11	53,2	59,6	1	36	6,2	11,7
13 S. Sepolcro	37	344	0	50	44,4	52,3	0	54	41,1	56,9	1	32	4,6	8,8
14 Ferugia	66	410	1	18	56,6	52,9	1	1	64,9	58,0	1	41	5,1	9,7
15 Foligno	38	448	0	35	51,4	52,0	0	32	56,3	57,9	1	44	5,1	9,6
16 Spoleto	39	479	0	48	50,5	53,2	0	32	73,1	68,9	1	62	5,7	10,7
17 Terni	27	586	0	38	54,0	53,3	0	29	65,9	58,7	1	53	5,5	10,3
18 Viterb	30	544	0	35	65,1	54,0	0	43	53,8	58,3	1	45	4,2	8,0
19 Anagni Civ	77	621	1	5	71,1	55,6	1	11	65,1	59,8	1	39	3,4	6,2
20 Roma	12	633	0	30	24,8	54,3	0	24	30,0	50,0	1	45	3,7	6,9

Figura 1
Tabella di Marcia.
Il primo esercizio è di tipo elementare, in quanto vengono usate formule di calcolo contenenti solo le quattro operazioni. È una tabella di marcia di un viaggio automobilistico che va predisposta prima della partenza e che poi va completata con i tempi reali presi durante il percorso. Le analisi che si possono fare con i valori kilometrici e orari, preventivi e consuntivi, sono numerosissime.

Color of unprotected cells

A	B	C	D	E	F	I	J	K	L	M	P	Q	R	S	T	U	V	
1	TABELLA	PREVENTIVO															DIFFERENZE	
2	DI MARCIA																	
3	Localita'	km	km	h	min	med	med	h	min	med								
4		par	tot			par	tot											
6	Roma	0	0	0	0	0	0	0	0	0								
7	Roma CIA	18	18	0	25	24,0	24,0	0	0	0	22,2	22,2						
8	Civitanova	62	72	0	55	67,6	54,0	0	58	64,1	58							
9	Orbetello	66	138	1	0	66,0	59,1	0	52	70,2	60							
10	Crossello	33	171	0	40	49,5	57,0	0	32									
11	Sienna	73	244	1	15	58,4	57,4	1	9									
12	Arezzo	63	387	1	38	42,8	53,4	1	11									
13	S. Sepolcuro	37	344	0	58	44,4	52,3	0	54									
14	Perugia	66	418	1	18	56,6	52,9	1	1									
15	Foligno	38	448	0	35	51,4	52,0	0	32									
16	Spoleto	39	479	0	40	58,5	53,2	0	32									
17	Terzi	27	586	0	30	54,0	53,3	0	29									
18	Rieti	38	544	0	35	65,1	54,0	0	43									
19	Roma CIA	77	621	1	5	71,1	55,6	1	11									
20	Roma	12	633	0	38	24,0	54,3	0	24	30,0	58,0	1-45	3,7	6,9%				

23-Aug-88 09:05 AM

Figura 2
Scelta dei colori.
Il Borland Quattro, che utilizzeremo per i nostri esercizi, dispone di una funzionalità molto evoluta (non presente nell'1-2-3 della Lotus) che permette la gestione dei colori del tabellone. Non si possono scegliere, evidentemente, specifici colori per ogni cella, ma colori per gruppi logici di celle. I colori scelti non fanno parte del singolo lavoro ma della configurazione del prodotto.

disponendo di comode funzionalità in più, rimane compatibile con i vari standard di mercato.

Tabella di marcia

Il problema che presentiamo è quello di programmare un viaggio in automobile, prevedendo varie tappe e per ciascuna di queste, di cui è nota la lunghezza, valutando un tempo di percorrenza.

Costruiamo con questi dati una tabella in cui vi siano, per ciascuna tratta, lunghezza parziale e totale, tempi parziali e totali, e medie parziali e totali.

Sono sei le colonne di dati iniziali da inserire... prima della partenza, poi durante il viaggio vanno inseriti i tempi reali di percorrenza. Sulla base di questi vengono calcolate le medie reali, e i vari scostamenti sul preventivo.

Il problema, molto estivo (per la cronaca oggi è il 17 agosto) è affrontabile anche dal più inesperto degli utilizzatori di Spreadsheet, ed è un classico problema da risolvere con il tabellone.

Nella prima colonna (A) inseriamo le località e nella seconda, trascurando le fincature (quindi le distanze chilometri che parziali). La terza colonna, (D) è ottenuta tramite una formula che somma tutti i chilometri parziali dal primo fino a quello corrente.

Nella quarta e quinta colonna vanno immessi i tempi di percorrenza previsti, suddivisi in ore e minuti (colonne E ed F), mentre nelle due successive (I e J) vengono calcolate, con delle formule di divisione, le medie parziali, sulla tratta, e totali, sull'intero percorso fino alla tappa corrente.

Le colonne G e H non appaiono in quanto sono nascoste. Le colonne nascoste esistono a tutti gli effetti (quindi possono contenere numeri e formule, possono essere referenziate in altre zone del tabellone, ecc.) solo che non vengono visualizzate.

Nel nostro caso tali colonne sono utilizzate per calcolare i tempi di percorrenza parziali e totali in minuti, valori

necessari per il successivo calcolo delle medie.

Le colonne da L a Q contengono dati reali del viaggio, che vanno digitati, orologi alla mano, durante il viaggio. Oppure, se non disponete di un personal portatile, all'arrivo.

Nelle colonne L e M vanno immessi ore e minuti effettivi. Nelle N e O, nascoste, viene eseguito il calcolo dei tempi in minuti e nelle P e Q viene eseguito il calcolo delle due medie.

Infine dalla colonna S in poi, c'è la zona con i calcoli delle differenze e delle statistiche, tra tempi preventivi e consuntivi.

Nel nostro caso ne abbiamo immessi tre e cioè la differenza tra i tempi totali (colonna S), la differenza tra le medie (colonna T) e la differenza percentuale di quest'ultima.

Riferendoci all'esempio della foto (fig. 1), per compiere il nostro giretto di 633 km, abbiamo impiegato 45 minuti meno del previsto, migliorando la media da 54,3 km/h a 58,0 km/h, pari al 6,9 per cento.

Già da questo primo esempio emerge la semplicità d'uso del tabellone dovuta al fatto che l'organizzazione fisica dello strumento è analoga alla organizzazione logica del problema. E questa organizza-

zione non è possibile ottenerla con linguaggio, se non ricorrendo a strutture del tipo vettori o matrici, che non trovano riscontri fisici altrettanto evidenti.

L'altra caratteristica subito emergente è quella di poter facilmente riorganizzare i dati a proprio piacimento spostando ad esempio le colonne (ricordiamo che le formule in esse contenute vengono automaticamente corrette). E questa stessa facilità non è riscontrabile con nessun linguaggio tradizionale.

Altrettanto facili le implementazioni che nel nostro caso possono comportare l'aggiunta di una o più colonne con altri calcoli. Nel nostro esempio abbiamo messo solo tre colonne di statistiche, ma, ad esempio se si utilizzasse il programma in tempo reale, direttamente nel viaggio, si potrebbero inserire colonne con gli orari reali, con i tempi e le medie presunte, ecc.

L'uso del colore con il Borland Quattro

Il Borland Quattro dispone di una funzionalità molto evoluta (non presente invece nel precursore 1-2-3 della Lotus) che permette la gestione dei colori del tabellone.

I colori si scelgono via opzione default

Figura 3
Portfolio. Zona immissione dati giornalieri. Di programmi per il controllo delle tessere del Portfolio (pubblicato sul quotidiano La Repubblica) ne sono stati pubblicati, sulle riviste concorrenti, numerosi. Lo strumento che meglio si presta a tale lavoro di controllo, che in realtà consiste solo nel fare semplici somme di otto numeri e nell'eseguire un confronto, è lo Spreadsheet.

Protect a block if protection is enabled

13	9	15	28	15	51	6	51
13	28	6	29	58	18	18	58
9	6	13	45	25	18	29	48
51	12	25	11	31	18	50	19
15	15	58	28	6	47	8	25
28	6	51	29	19	8	28	48
28	51	6	13	38	9	48	19
28	15	29	45	7	18	58	19
15	29	38	8	45	28	11	58
15	51	6	12	8	25	38	48
51	6	12	38	7	25	18	58
13	6	12	25	8	48	25	58
15	28	13	25	8	45	48	18
13	28	9	58	25	9	58	19
9	28	51	51	58	28	6	28
13	15	15	8	29	8	58	13

Advanced
Values
Transpose
Protect
Unprotect
Block Name
Create
Delete
Labels
Reset
Make Table

23-Aug-88 08:31 AM

del menu principale, il che vuol dire che i colori non sono specifici del lavoro Quattro, ma della configurazione del prodotto. La scelta, pilotata dai menu a finestra, è possibile per una dozzina di tipi di celle ed avviene attraverso un palette/campionario (standard nei prodotti Borland) in cui, a vista, si sceglie l'accoppiamento tra colore di primo piano e di sfondo per una particolare zona della videata o del tabellone (fig. 2).

Nelle varie foto a corredo dell'articolo è evidente l'efficacia della funzione, che può servire, ad esempio, per evidenziare zone di immissione dati da zone protette dall'immissione, zone titoli, zone con scritte, ecc. Inoltre essendo possibile scegliere un colore di primo piano uguale al colore di sfondo, si possono «nascondere» parti della videata.

È evidente che la compatibilità del lavoro Quattro verso l'1-2-3 è conservata, in quanto il lavoro Quattro, riservato nell'1-2-3, perde praticamente i colori, non gestiti da quest'ultimo.

Portfolio Republic

Tutti i quotidiani ormai dispongono di ricchi concorsi a premio consistenti nella distribuzione di tessere (differenti tra di loro e regalate con il giornale) e nel controllo, giornaliero, da parte dei lettori di alcune tabelline numeriche o grafiche, che vanno confrontate con il contenuto delle proprie tessere.

Il quotidiano più diffuso a livello nazionale è La Repubblica che ha come gioco il Portfolio, consistente nel confronto da eseguire tra una somma di numeri variabili giornalmente, ma dipendenti dal contenuto della tessera individuale, e una cifra pubblicata anch'essa giornalmente.

Per chiarire il semplice meccanismo a chi non lo conosce, ogni tessera ha 8 numeri, variabili da 1 a 48. Ogni giorno La Repubblica pubblica una tabella con 48 valori numerati da 1 a 48, (che rappresentano le variazioni delle quotazioni di titoli di borsa ma questo ai fini del gioco è irrilevante) e un numero obiettivo chiamato dividendo. Per ciascuna tessera vanno sommati algebricamente gli otto valori corrispondenti ai numeri presenti nella tessera e se il valore risultante è uguale al valore obiettivo, la tessera, anzi il suo fortunato possessore, ha vinto.

La difficoltà della vincita sta nel fatto che il numero obiettivo in genere è molto differente dalla media dei valori somma delle varie tessere. Anche questo è irrilevante ai fini del gioco ma serve a spiegare perché pur essendoci in circolazione una decina di milioni di

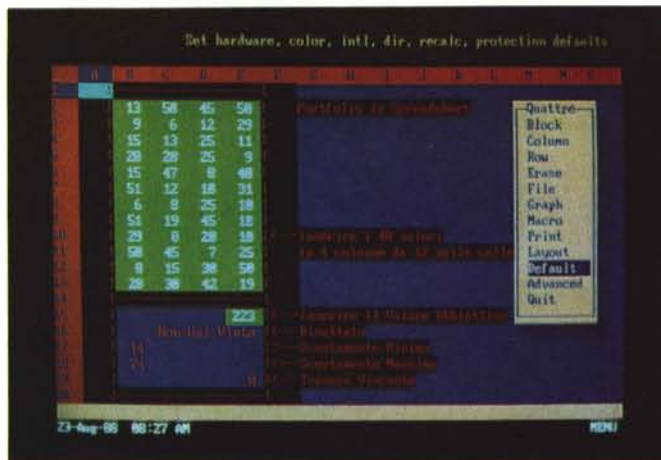


Figura 4
Portfolio. Zona tessere.

In una zona del tabellone vanno inseriti, e non più toccati, i dati iniziali delle tessere, collegati per mezzo di formule di somma e di confronto con il dividendo giornaliero. In un'altra zona vanno immessi giornalmente le 48 variazioni di quotazione e il dividendo. Nella stessa videata sono stati inseriti i messaggi di risposta e i risultati delle statistiche.

tessere i vincitori siano pochi anzi in certi giorni nessuno.

L'acquirente fisso di La Repubblica dovrebbe avere circa una decina di tessere (ogni tanto se ne trova una nel giornale), ma molti ne hanno fatto incetta e ne hanno numerose decine, per cui il controllo delle tessere diventa, se eseguito a mano, abbastanza faticoso.

Trattandosi di eseguire calcoli e confronti va da sé che il computer ben si presta a svolgere più esattamente e velocemente lo stesso lavoro.

Il controllo delle tessere con il computer comporta due vantaggi sostanziali, il primo è che una volta sviluppato il programma e memorizzato il contenuto delle tessere possedute non occorre più «portarsele appresso». Il secondo è che una volta inseriti i dati base delle tessere il lavoro di controllo ha una durata indipendente dalla loro quantità in quanto consiste nel digitare solo i 48 numeri del giorno e il dividendo e poi vedere il risultato del programma.

Il gioco Portfolio è molto seguito al punto che esistono piccole organizzazioni che eseguono il controllo delle schedine «conto terzi». Oppure in certe aziende il controllo delle schede dei dipendenti è stato centralizzato, e viene eseguito dal Centro Elaborazione Dati.

Sulle riviste di microinformatica il problema è già stato affrontato più volte. Ricordo un programma Basic il cui listato era lungo 11 pagine. Eccessivo per un problema che come calcoli richiede la somma di otto numeri e il confronto del risultato con un altro numero.

La soluzione con lo Spreadsheet

La nostra soluzione consiste nell'organizzare il tabellone in due zone, una inferiore, nascosta, in cui sono inseriti i dati di tutte le tessere e le formule di calcolo per ciascuna tessera. La zona superiore contiene le 49 celle di input, da riempire ogni giorno con i 48 valori delle quotazioni, e quello del dividendo.

Contiene inoltre una parte destinata ai

messaggi del tipo «Hai Vinto/Non Hai Vinto», e ad eventuali dati statistici, come scostamenti, medie, ecc. inseribili a volontà.

Nella zona inferiore (fig. 4), vanno inseriti i dati delle tessere (dalla colonna E alla colonna L). Se nella tessera il primo numero è il 5, nella colonna E va digitato +B6, in quanto è proprio nella cella B6 che ogni giorno viene immesso il quinto valore.

Una semplificazione può essere quella di «battezzare» le celle dove vengono immessi i valori giornalieri, ad esempio VAL01, VAL02, ecc. In tal modo nella colonna E andrebbe digitato +VAL05.

Ulteriore semplificazione potrebbe essere quella di riportare in un'unica cella tutto il contenuto della tessera e quindi la formula diventerebbe + VAL05+VAL11+VAL23...

Nel nostro caso nella colonna D, viene sommato il contenuto delle otto celle a destra (è il totale della tessera) e nella colonna C viene calcolato lo scostamento dal dividendo. Nella colonna B c'è un numero progressivo di tessera inserito per indicare, nel caso di vincita, quale abbia vinto.

Nella prima cella c'è infine una semplice formula di tipo logico @IF, che restituisce 1 se lo scostamento con il dividendo è 0 (caso vincita) oppure 0.

Questa zona del tabellone ha una larghezza di poche colonne (riducibile, complicando un poco le formule, anche ad una sola), e una lunghezza pari alle tessere possedute.

Nella parte superiore del tabellone (fig. 3) c'è la zona di immissione dati, riquadrata e bloccata in modo da impedire al cursore di posizionarsi al di fuori. Vanno inseriti 48 valori in 4 colonne da 12, e il valore del dividendo.

Nel riquadro inferiore appare il messaggio «Hai Vinto» se la somma della colonna A della zona tessere non è pari a 0. Vengono poi evidenziati, con semplici formule @MAX e @MIN, i valori massimi e minimi degli scostamenti. Interessante è infine il calcolo del numero della tessera

Figura 5
Tombola. Estrazione numero singolo.
Questo primo esercizio, il più semplice dei tre presentati, risolve il problema di calcolare un numero casuale (tra 1 e 90) e di posizionarlo nel tabellone. Ogni cella del tabellone contiene una formula che visualizza il numero estratto. In questo esercizio non è però risolto il problema di far rimanere il numero anche con le estrazioni successive e il problema, più complesso, di non far uscire di nuovo un numero già uscito.

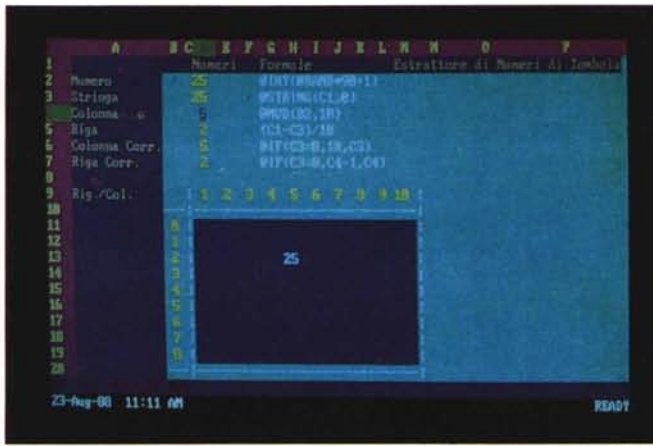


Figura 6
Borland Quattro Macro.

Le Macro nello Spreadsheet hanno vari livelli di impiego, nel senso che possono essere utilizzate per risolvere problemi sia di tipo elementare, e in questo caso sono affrontabili anche dal neofita, che di tipo complesso, per realizzare applicazioni «chiavi in mano», destinate cioè ad un utilizzatore della procedura del tutto inesperto di microinformatica. L'Help interattivo del Quattro mostra in varie videate le varie istruzioni Macro.

vincente (eventualità ahimé rara) che utilizza la formula @VLOOKUP per individuare se nella prima colonna della zona tessere c'è un 1, e in caso affermativo, a quale progressivo di tessera corrisponda.

Confrontando la nostra soluzione con lo Spreadsheet con quella realizzata in Basic (le undici pagine di listato) valutiamo un impegno di tempo e di occupazione di memoria pari a meno del cinque per cento. Il che ci permette di affermare che in questo tipo di applicazioni lo Spreadsheet è molto più adatto del linguaggio tradizionale.

Estrazione dei numeri della Tombola

Come ultimo esempio costruiamo un estrattore di numeri di Tombola. È un problema noto a tutti e per questo non lo descriveremo.

Nel trattarlo nascono numerose problematiche proprie del calcolo delle probabilità, e che trasportate su computer richiedono l'ideazione e l'utilizzo di semplici algoritmi.

L'estrazione del numero della Tombola è, per dirla con termini statistici, «senza ripetizione», nel senso che un numero, una volta estratto, non esce più. La fun-

zione di estrazione senza ripetizione va costruita in quanto lo Spreadsheet (ma anche il Basic) dispongono solo di una funzione di generazione di un numero casuale.

In una simulazione poi che preveda anche il riempimento del tabellone della Tombola con lo Spreadsheet esiste anche l'altro problema di riempire via via una zona, dapprima vuota, con dei numeri.

Tombola - primo esercizio

La soluzione più semplice prevede che in ogni cella del tabellone esista una formula che a seconda di una condizione esterna visualizzi un numero o non visualizzi nulla.

L'inconvenienza di questa soluzione sta nel fatto che una volta modificata la condizione, perché è stato estratto un altro numero, di nuovo non viene visualizzato nulla.

Nella parte alta della videata (fig. 5) sono riportati i numeri utilizzati, che sono rispettivamente il numero estratto, la sua traduzione in stringa, la colonna e la riga cui appartiene, infine la colonna e la riga corrette.

La correzione è quella che serve per portare i numeri 10, 20 ecc. sulla riga del-

la decina precedente come accade sul tabellone della Tombola.

In questo esempio, che non fa uso delle Macro, ogni cella del tabellone contiene una formula, che verifica se il numero estratto è quello cui la stessa corrisponde. Se non corrisponde, visualizza una stringa vuota, altrimenti mostra la stringa con il numero.

Questo primo esempio presenta due inconvenienti che lo rendono incompleto. Il primo è che non è stata implementata la routine di non estrazione del numero già estratto (e questo lo vedremo nel terzo esercizio).

L'inconveniente principale è invece indipendente proprio dallo Spreadsheet nel quale tutte le formule sono attive allo stesso momento, per cui delle novanta formule presenti nel tabellone solo una alla volta può evidenziare il suo numero.

Il problema si può risolvere solo con una Macro che, una volta scritto il numero, come risultato di una formula, lo traduca in un semplice valore numerico, in modo che non venga influenzato da successivi ricalcoli.

In questo caso quindi il ricorso alle Macro diventa pressoché indispensabile per risolvere un problema altrimenti non risolvibile.

Tombola - secondo esercizio

La Macro nello Spreadsheet è una sequenza di comandi di vario tipo eseguibili tutti insieme richiamando il nome della Macro.

I comandi sono oltre a quelli di menu, indicati con l'iniziale del percorso nell'albero dei menu (es. /BAV significa: /richiamo menu, B opzione Block, A opzione Advanced, V opzione Value, ecc.) comandi che indicano funzioni di tastiera (tasti funzione, freccia, ecc.), comandi di programmazione (for..., if...), ecc.

Alcuni comandi di programmazione (vedi la foto 6 con l'elenco di quelli propri delle Macro) fanno ricorso a variabili, che in uno Spreadsheet, ricordiamolo di nuovo, sono le stesse celle, che vanno riservate e destinate allo scopo. Le celle possono essere utilizzate anche come Labels per istruzioni di salto.

Nelle Macro i comandi hanno la caratteristica sintattica di essere compresi tra due parentesi graffe, mentre il tasto ENTER viene rappresentato dal carattere ~. La scrittura o l'interpretazione di una Macro sono quindi operazioni abbastanza semplici (vedi fig. 7).

Nel nostro caso esistono due Macro, la alt A, che fa uso di due subroutine la \C (che sposta il cursore a destra) e la \R (che lo sposta in giù) e la alt P che pulisce il tabellone della Tombola (il comando di menu è /BE, ovvero Block Erase della zo-



Figura 7
Tombola. Riempimento del tabellone.
Nel secondo esercizio si cominciano ad utilizzare le Macro. In questo caso il tabellone è inizialmente vuoto sia di numeri che di formule. È la Macro A, richiamabile cioè premendo alt+A, che estrae il numero e lo posiziona, facendocelo rimanere, nel tabellone. La Macro \P invece pulisce il tabellone.

Conclusioni

Gli utilizzatori dello Spreadsheet si possono dividere in tre grandi categorie.

La prima categoria è rappresentata da quelli che lo utilizzano solo per fare tabelle numeriche con somme per riga o per colonna. Si tratta di una sottutilizzazione del prodotto ma del tutto soddisfacente da parte di chi prima faceva le stesse operazioni a mano.

Appartengono a questa categoria buona parte degli autodidatti e buona parte dei neofiti del computer, che non hanno nessuna precedente esperienza con altri strumenti di Personal Computing. La seconda categoria è rappresentata dall'utente evoluto, quello che conosce bene le funzionalità e quindi riesce a risolvere problemi anche di grande complessità. Può essere il neofita evoluto, o l'utente già esperto che apprezza dello Spreadsheet la particolare rapidità d'uso. La terza categoria è quella del programmatore, che si è impadronito della filosofia d'uso dello Spreadsheet, e quindi utilizza anche il suo linguaggio di programmazione come una qualsiasi funzionalità evoluta.

Nelle applicazioni presentate abbiamo appunto esemplificato questi tre livelli.

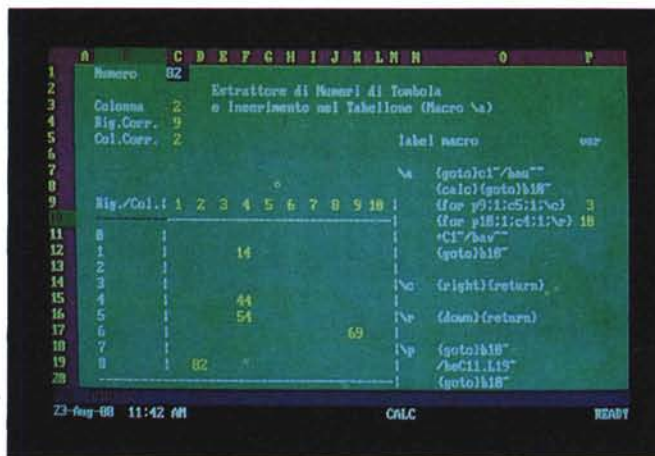
Per tornare al confronto con i linguaggi di programmazione risultano evidenti, ovviamente a chi ha un po' di dimestichezza con qualsiasi linguaggio, le notevoli differenze che proviamo ad elencare:

— *Possibilità di lavorare per passi successivi.* Con lo Spreadsheet il lavoro è sempre attivo e funzionante, sin dall'inizio e lo è anche via via che si inseriscono dati, formule, ecc. Questo permette il debug della applicazione direttamente durante la sua costruzione e non solo alla fine.

— *Modificabilità dell'applicazione.* Lo strumento Spreadsheet si presta ad essere utilizzato per sperimentare, senza rischi, implementazioni dell'applicazione. Ad esempio si può aggiungere una colonna ad una applicazione già funzionante, in cui per mezzo di una formula venga calcolato un dato... a cui non si era pensato prima. La stessa cosa in un programma scritto in Basic richiede inserimento di nuove istruzioni, e la modifica di altre, con interventi pericolosi su un programma che già «gira» bene.

— *Possibilità di frazionare il lavoro.* Con lo Spreadsheet si può affrontare immediatamente il cuore dell'applicazione, ad esempio inserendo semplicemente dati di calcolo provvisori in celle. Una volta risolta la maggiore difficoltà si può passare al completamento dell'applicazione. Con un linguaggio tradizionale, l'applicazione, per poter funzionare e quindi essere testata, deve praticamente essere completata.

Figura 8
Tombola. Estrazione senza ripetizione.
Tutti i programmi di gioco utilizzano la funzione RANDOM che genera un numero casuale tra 0 e 1 (escluso). Con semplici calcoli si traduce il numero nell'estrazione di un numero di tombola, oppure nel lancio di un dado o nella distribuzione di un mazzo di carte. Per evitare che un numero già uscito (o una carta già distribuita, ecc.) venga di nuovo estratto, occorre utilizzare il semplice algoritmo proposto nell'esercizio.



na che va da C11 a L19.

La alt A dapprima raggiunge la casella C1 dove esegue un ricalcolo del numero casuale, poi esegue due loop che spostano il cursore a destra e in basso a seconda del contenuto delle celle P9 e P10, che quindi in questo caso sono utilizzate come variabili.

Raggiunta la casella voluta vi viene copiata la formula «+C1» che quindi riproduce, al posto giusto, il numero estratto, e poi tale numero, risultato della formula, viene tradotto nel suo semplice valore numerico. Di questa operazione si occupa il comando Block Advanced Value.

Tombola Estrazione senza ripetizione

La finalità del terzo esercizio è quella di estrarre i 90 numeri senza ripetizione. Il che vuol dire che se il primo numero estratto è il 30 questo non sarà più estratto.

L'algoritmo classico è quello di estrarre un numero che rappresenta, non il numero estratto, ma un indice di un vettore di

numeri ancora non estratti.

Inizialmente il vettore (colonna A) contiene i novanta numeri da 1 a 90. Estratto il primo numero, ad esempio il 30, questo viene tolto di mezzo e sostituito da quello contenuto nell'ultimo elemento del vettore, nel nostro caso il 90.

Il secondo numero viene calcolato tra 1 e 89. Se fosse di nuovo 30 significherebbe che il numero esatto è il 90 (che occupa il trentesimo elemento del vettore), e al posto del 90 verrebbe messo l'89. Quindi l'estrazione avviene prima tra 90 numeri, poi tra 89 poi tra gli 88 rimasti, ecc.

È un meccanismo facilmente riproducibile in ogni linguaggio ed applicabile ad esempio al caso di distribuzione di un mazzo di carte. La finalità della Macro, che per mancanza di spazio non descriviamo, ma che può essere letta nella figura 7, è quella di scrivere nella colonna B i numeri via via estratti senza ripetizione. Nella foto si può notare come nel vettore in colonna A, i posti dei numeri estratti siano stati occupati dagli ultimi numeri del vettore (fig. 8).

power & compatibility



AX 8000

CPU 80386 20/25 MHZ
2-16MB RAM 0 WS
HARD DISK 40/330MB
MFM/ESDI
FLOPPY DISK
5,25"/3,5"
VIDEO MGA/EGA/VGA

AX 6000

CPU 80286 12/16 MHZ
2-4MB RAM 0 WS
HARD DISK 40/330MB
MFM/ESDI
FLOPPY DISK
5,25"/3,5"
VIDEO
CGA/MGA/EGA/VGA

PX 6000

CPU 8286 12/16 MHZ
2-16MB RAM 0 WS
HARD DISK 20/90MB
MFF/ESDI
FLOPPY DISK 5,25"/3,5"
VIDEO CGA/MGA/EGA/VGA

PX 3000

CPU 8088 10 MHZ
640K RAM
HARD DISK 20/40MB
FLOPPY DISK 5,25"/3,5"
VIDEO CGA/MGA/EGA

PX 5000

LAN WORK-STATION
CPU 80286 8/12 MHZ
512K/1MB RAM 0 WS
DISK-LESS

LAN

- ETHERNET 10MBIT/sec
- ARCNET
- STARLAN

MODELLI AX 8000/AX 6000 TOWER

Supermicro per work-station ad alte prestazioni, LAN-Server, ottimizzati per XENIX e UNIX
Hard disk con Interleaving 1:1 (transfer-rate 500K-1MB/sec)

MODELLI PX 3000/PX 5000

Personal computer da tavolo di alte prestazioni per configurazioni stand-alone o LAN work-station

LAN

Standard Ethernet, Starlan e Archnet con possibilità di remote boot-rom con work-station PX 5000 senza memoria di massa locali.

