Computer School Series

Studiare Facile

ALGEBRA

GREMESE EDITORE
Computer Games Series n. 13

Computer School Series

ALGEBRA CON COMMODORE 64

Computer Games Series
Periodico quattordicinale
N. 13 - 1° marzo 1985
Registrazione Tribunale di Roma N. 138/84
del 24 marzo 1984
Direttore Responsabile: Grazia Valci

Collana diretta da
Giancarlo Zagarese

Copertina
Antonio Dojmi

Fotocomposizione
Sistemi Tipografici Jamotti Roberto - Roma

Stampa
Nuova STE - Città di Castello

ISBN 88-7605-176-7

Computer School Series

ALGEBRA CON COMMODORE 64

Marco Sillano

GREMESE EDITORE

Computer School Series:

Volumi pubblicati:

Algebra con COMMODORE 64
Algebra con VIC 20
Algebra con ZX SPECTRUM

Di prossima pubblicazione:

Fisica con COMMODORE 64
Fisica con VIC 20
Fisica con ZX SPECTRUM
Ragioneria con COMMODORE 64
Ragioneria con VIC 20
Ragioneria con ZX SPECTRUM
L'inglese con COMMODORE 64
L'inglese con VIC 20
L'inglese con ZX SPECTRUM

Computer Games Series:

Volumi pubblicati:

Giochiamo con COMMODORE 64
Giochiamo con VIC 20
Giochiamo con ZX 81
Giochiamo con ZX SPECTRUM
Giochiamo con ATARI
Giochiamo con TRS-80
Altri giochi per il vostro COMMODORE 64
Altri giochi per il vostro VIC 20
Altri giochi per il vostro ZX SPECTRUM
Giochiamo con TI 99/4A
Altri giochi per il vostro ATARI
Altri giochi per il vostro ZX 81

Di prossima pubblicazione:

Giochiamo con APPLE
Adventure games per il COMMODORE 64
Adventure games per lo ZX SPECTRUM

INDICE

Introduzione 8
Come usare i programmi 10
Calcoli senza programmi 15
Espressioni algebriche 21
Polinomi in X 34
Polinomi ordinati 44
Equazioni di primo grado 55
Metodo di Cramer 65
Sistema X, Y implicito 73
Numeri complessi 82
Equazioni di secondo grado 89
Progressioni aritmetiche 98

Appendici

Subroutines 104
Prove colori 122

MARCO SILLANO - AUTORE

Ingegnere elettronico, si è occupato per alcuni anni di didattica della matematica moderna in un progetto UNESCO, pubblicando lavori e traduzioni. Specializzato nella progettazione di sistemi a microprocessore, insegna attualmente elettronica in una scuola di informatica. Quando non è davanti ad una tastiera lo troverete a pesca od in montagna.

GIANCARLO ZAGARESE
DIRETTORE DI COLLANA

Giancarlo Zagarese, insegnante di discipline scientifiche, è autore di oltre 250 articoli e di vari volumi nei settori dell'elettronica e dell'attività subacquea. Per GREMESE EDITORE ha già collaborato in *Il Sub per tutti* e *Il Sub in apnea* della serie "gli Abbiccì".

RINGRAZIAMENTI

L'editore e l'autore ringraziano la ditta DATAOFFICE spa, Via Sicilia 205, ROMA, tutto il personale, e in particolare il dott. Alessandro Iovenitti per la collaborazione e per aver messo a disposizione le attrezzature necessarie alla realizzazione di questo libro.

PREFAZIONE

Poco meno di un anno fa la Gremese Editore ha saputo proporre, finalmente in italiano, una dozzina di agili e divertenti volumetti sui "Giochi con il computer" nella "Computer Games Series".

Il successo, prevedibile ma confortante in questo momento un po' oscuro per l'editoria italiana, ci ha confermato che la via intrapresa era giusta e praticabile.

Oggi, con questa **"Computer School Series"**, proponiamo a quella che ormai è una sterminata falange di possessori di "piccoli" computer di voler fare un passo più avanti anche sul piano culturale.

Reputiamo, infatti, che tutto il sapere tradizionale, con tutte le sue ampie e poliedriche sfaccettature, debba essere "rivisitato" con l'ausilio dei computer.

Ogni elaboratore elettronico, per piccolo che sia, è un mezzo potente che sin d'ora, ma ancora di più nel futuro, ci sarà al fianco, momento dopo momento, come trampolino di lancio per la nostra fantasia, come amplificatore per la nostra intelligenza e come "ordinatore" per il nostro sapere.

g.c.z.

INTRODUZIONE

Il vostro COMMODORE 64, con il quale vi sarete senz'altro divertiti in molti giochi animati, ha tutte le capacità di un calcolatore, con infiniti campi di applicazione.

Anche l'algebra può essere affrontata con l'aiuto di un calcolatore come il C-64, che è in grado di effettuare per noi alcuni calcoli od operazioni ripetitive e noiose.

Con il suo aiuto possiamo evitare quegli aspetti e quelle difficoltà che talvolta l'algebra ed il suo studio presentano. L'algebra può diventare un gioco divertente...

Come un docile strumento, il calcolatore può aumentare le nostre capacità anche se è, al tempo stesso, molto stupido. Dobbiamo scegliere noi come, quando e perché usarlo. Dobbiamo essere saggi anche per lui.

Parte di questa saggezza è condensata nel programma, che guida passo passo il calcolatore, e parte in chi usa il programma, in modo giusto per i suoi scopi.

I programmi di questo libro possono aiutarvi ad affrontare alcuni problemi che l'algebra presenta: quando usarli, come applicare la potenza del vostro C-64, sarete voi a deciderlo; ciò vi porterà, ne siamo convinti, ad una migliore comprensione dei problemi di questa parte della matematica.

Non è necessario conoscere il BASIC e le tecniche di programmazione per usare i programmi di questo libro in quanto basta copiare con attenzione i listati che presentiamo: i programmi sono provati e funzioneranno.

Abbiamo scelto di presentare alcuni programmi che affrontano problemi algebrici nel modo tipico dei calcolatori, cioè calcolando. Il modo di lavorare di un calcolatore elettronico è infatti molto diverso da quello di un uomo: essenzialmente il calcolatore sa fare solo calcoli molto semplici, ma li fa molto in fretta. È il programma che combina questi calcoli tra loro fino ad ottenere dei risultati talora sorprendenti.

Occorre quindi specificare al vostro C-64 anche le operazioni più semplici. Alcune operazioni più elementari, ma che troviamo in quasi tutti i nostri programmi,

sono state raggruppate e formano un gruppo di *subroutines* riportate in appendice.

Una subroutine è un po' come un pezzo di programma prefabbricato, che può essere usato più volte là dove occorre, senza riscriverlo ogni volta.

Come un architetto dice: «Qui una porta, là due finestre, qui un'altra porta» così noi possiamo utilizzare più volte le subroutines nei nostri programmi, ogni volta che ci occorra la loro opera. Senza dover specificare ogni volta «serrature, cerniere, legno...».

Con questo metodo i programmi sono più semplici e chiari, ed è più facile seguirne il funzionamento.

Ogni programma ha un paragrafo. «COME FUNZIONA», che ne esamina la struttura; può essere molto utile se volete modificarlo o se volete creare dei vostri programmi. Non è assolutamente indispensabile leggerlo per usare i programmi: quanta gente prende un ascensore senza sapere come funziona!

Pensiamo però che prima o poi questa curiosità vi verrà: la programmazione di un calcolatore per fargli fare quello che vogliamo è una attività così affascinante che siamo certi che anche voi vorrete scrivere i vostri programmi, dopo aver usato quelli che vi presentiamo. Allora vi saranno molto utili le subroutines; per questo sono accompagnate da ampi commenti su cosa fanno e come devono essere usate in un programma.

In questa nostra fatica ci accompagnano Giorgio e Mister BIT. Giorgio è un ragazzo come voi, ma ha un calcolatore magico... nel quale abita un genietto: Mister BIT. Il vostro calcolatore non è fatato come quello di Giorgio, non apparirà quindi Mister BIT, così come noi abbiamo immaginato nel libro, né appariranno i suoi consigli (che comunque vi abbiamo trascritto...), ma con questi programmi potrà egualmente aiutarvi nella risoluzione dei vostri problemi.

Bene, un po' di entusiasmo e buon divertimento!

COME USARE I PROGRAMMI DI QUESTO LIBRO

Per utilizzare i programmi di questo libro vi occorre un C-64, un registratore CBM oppure un registratore normale con adattatore ed un televisore o monitor, meglio se a colori.

Accendete il C-64: apparirà sullo schermo una scritta con l'indicazione della quantità di memoria disponibile ed il messaggio «READY».

A questo punto siete pronti ad introdurre i vostri programmi, copiandoli fedelmente dal libro, riga per riga e premendo alla fine di ogni riga il tasto /RETURN/.

Vi potrà capitare di fare degli errori: per cancellare una riga basta scriverne il numero e usare il tasto /RETURN/; per correggerla si può riscriverla tutta oppure riscrivere soltanto le parti sbagliate, con l'uso dei tasti /DELETE/ (cancella) e /INSERT/ (inserisci). Premendo /RETURN/ la riga corretta sostituirà la vecchia.

Controllate spesso quello che avete scritto dando il comando /LIST/ e, quando il programma è così lungo da non entrare tutto nel video, usate /LIST riga iniziale - finale/ (per esempio /LIST 1000-1050/): potrete così controllare le righe comprese tra i numeri che avete indicato. Consultate il manuale del C-64 per maggiori informazioni sull'uso di queste funzioni.

CARATTERI SPECIALI

Il C-64 accetta all'interno di ogni stringa — dopo il segno / " / (virgolette) — dei caratteri speciali, che servono in genere a spostare il cursore o a definire il colore in fase di esecuzione del programma.

Questi caratteri speciali appaiono sullo schermo come piccoli disegni su fondo scuro, come peraltro già indicato nel manuale del COMMODORE.

Noi abbiamo stampato i nostri programmi con una stampante MPS-801, utilizzando un programma di stampa apposito che sostituisce ai caratteri speciali i nomi dei tasti da usare per ottenerli, scritti tra parentesi quadre.

L'esempio "STAMPA CARATTERI SPECIALI" mostra come appaiono sul video e nei listati di questo libro, i caratteri speciali usati.

```
100 REM*********************************
110 REM** STAMPA CARATTERI SPECIALI ***
120 REM*********************************
130 REM
140 REM  PER I LISTATI DEI PROGRAMMI DI
150 REM  QUESTO LIBRO E' STATA USATA LA
160 REM            LA STAMPANTE MPS-801.
170 REM       QUESTO ESEMPIO MOSTRA COME
180 REM          APPAIANO I CARATTERI DI
190 REM  CONTROLLO SUL VOSTRO SCHERMO E
200 REM        QUALI TASTI OCCORRA USARE
210 REM                     PER OTTENERLI
220 REM       SEGUENDO L'INDICAZIONI SUL
230 REM                          LISTATO.
240 REM
250 REM
260 REM  "CARATTERE     [TASTI]
270 REM
280 PRINT "▚            [HOME]
290 PRINT "▞            [SHIFT]+[HOME]
300 PRINT "▝            [CRSR↑]
310 PRINT "▗            [SHIFT]+[CRSR↑]
320 PRINT "▌            [CRSR←]
330 PRINT "▐            [SHIFT]+[CRSR←]
340 PRINT "▖            [CTRL]+[RVS ON]
350 PRINT "▘            [CTRL]+[RVS OFF]
360 REM
370 REM
380 REM*********************************
390 REM
```

```
400 REM A VOLTE NEL LISTATO I CARATTERI
410 REM        DI CONTROLLO RIPETUTI SONO
420 REM                    INDICATI COSI':
430 REM
440 REM
450 PRINT "■■■■          [3 CRSR]
460 REM
470 REM*****************************
480 REM
490 REM  NON COPIATE QUESTO PROGRAMMA
500 REM*****************************
```

COMMENTI

Nei nostri programmi abbiamo introdotto righe di commento (sono quelle che iniziano con /REM/) e righe di separazione (quelle che contengono solo i due punti /:/), per mettere in evidenza le varie parti del programma stesso. Queste righe, pur non influenzando il funzionamento del programma, occupano spazio in memoria e tempo per copiarle!

Quando copiate un programma potete quindi saltare tutte le righe di commento o di separazione. Per facilitarvi il compito, queste righe hanno un numero dispari, perciò copiate solo le righe con i numeri pari e trascurate del tutto quelle dispari.

Il seguente programma:

```
999 REM*****************************
1000 PRINT "PROVA"
1003 :
1010 END
1013 REM*****************************
```

deve essere perciò copiato così, eliminando tutte le righe dispari:

```
1000 PRINT "PROVA"
1010 END
```

in modo da risparmiare spazio in memoria e tempo nel copiarlo.

MEMORIZZARE

Dopo aver scritto un programma, per prima cosa «salvatelo» su nastro. Questo si fa con il comando /SAVE "nome"/. Scegliete per i vostri programmi dei nomi brevi e che indichino chiaramente di che programma si tratta: vi sarà più facile ricordarli ed usarli.

Salvate il programma su nastro prima ancora di provarlo: infatti un errore anche stupido nel programma può causarne la distruzione, e, se non ne avete una copia su nastro da ricaricare, dovrete ricominciare a scriverlo.

Dopo aver salvato il programma potete provarlo: date il comando /RUN/ e controllate che non ci siano indicazioni di errore e che lo schermo si presenti esattamente uguale alle figure riportate come esempio per ciascun programma: sono l'esatta riproduzione di quanto appare sullo schermo, ottenute con un programma apposito. Ricontrollate accuratamente le righe che danno origine a messaggi di errore, in quanto è facile, anche mettendoci la massima attenzione, fare una svista: basta saltare una virgola! Provate tutte le funzioni di un programma, e controllatene i risultati. Quando siete ben certi che il programma è a posto, salvatelo in maniera definitiva. Vi consigliamo di registrare i programmi almeno due volte, su due cassette diverse; non si sa mai, una cassetta potrebbe danneggiarsi e voi perdereste il vostro programma! Quindi due copie su due cassette: una da usare comunemente, l'altra da tenere da parte, di riserva.

SUBROUTINES

I primi programmi da copiare sono le subroutines dell'appendice. Saranno usate da tutti gli altri programmi. Copiatele e registratele (su due cassette) possibilmente all'inizio del nastro poiché le userete spesso: così vi sarà più facile ritrovarle.

Date loro il nome di SUBROUTINES.

Quando volete copiare un programma di questo libro o quando volete scrivere un vostro programma che le utilizzi, per prima cosa caricate le subroutines in memoria, con il comando /LOAD "SUBROUTINES"/ poi scri-

vete il programma e registrate di nuovo tutto il nome del programma che avete scritto.

Per usare in seguito questo programma ricaricatelo in memoria e quando appare la scritta /READY/, scrivete semplicemente /RUN/: il programma funzionerà.

RICORDATEVI DI:

- Caricare le subroutines
- Copiare solo le righe pari
- Fare attenzione ai caratteri speciali
- Salvare» il programma appena scritto
- Provare attentamente i programmi
- Registrare definitivamente *due volte* su *due* cassette diverse.

CALCOLI SENZA PROGRAMMI

Il nostro computer è molto più potente e più rapido di ogni comune calcolatrice tascabile. Se usato in maniera opportuna il C-64 può effettuare per noi, più velocemente e più esattamente, tutti i complessi e lunghi calcoli che ci assegnano a scuola. Non vi è neanche bisogno di saper programmare in quanto è sufficiente la conoscenza di due o tre istruzioni di BASIC per ottenere comodamente e facilmente il risultato. In BASIC, dove sono disponibili anche le «difficili» operazioni esponenziali e trigonometriche, le consuete operazioni di addizione, sottrazione, moltiplicazione, divisione, elevazione a potenza ed estrazione di radice sono proprio facili facili: basta una PRINT... Non solo il nostro computer calcolerà per noi il risultato ma l'espressione introdotta — la formula — rimarrà a nostra disposizione sul video per un controllo, per una eventuale modifica e, se ne abbiamo bisogno, per essere nuovamente impiegata.

Vediamo un po'.

Ma è proprio necessario scrivere un programma?

Giorgio è molto nervoso: proprio oggi si è rotta la piccola calcolatrice tascabile, quella che porta a scuola. Proprio oggi che deve fare tutti quei calcoli. E il padre glielo aveva detto: «Guai a te se anche questa...» non più di un mese fa.

Accende il C-64: giocare con il «Serpentone» che non deve mangiare se stesso lo calmerà, ai calcoli penserà poi...

Ecco che al posto del solito labirinto con il lungo «centipede» appare un buffo omino. Appare anche un messaggio:

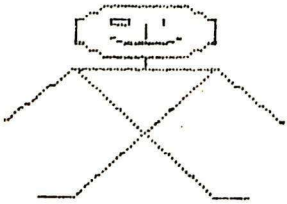
```
CIAO !
SONO MISTER BIT
VEDIAMO UN PO' QUEI
CALCOLI
CHE DEVI FARE
```

Giorgio, stupito ed obbediente, una volta tanto, prende il quaderno e comincia a scrivere sul C-64

```
375*27
```

Appena batte /RETURN/ appare la scritta:

```
NON COSI'
SE VUOI LA RISPOSTA
DEVI METTERE
UN PUNTO
INTERROGATIVO ...
```



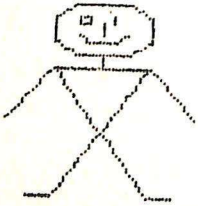
Giorgio perplesso riprova:

```
?375*27
```

e come batte /RETURN/ appare:

```
10125
```

Appare anche la figurina buffa:



```
MA QUESTO E' FACILE. SE FAI
ATTENZIONE ALLE PARENTESI
PUOI FARE CALCOLI PIU'
COMPLESSI.

PROVA !
```

Giorgio deve anche calcolare l'ipotenusa di un triangolo rettangolo, i cui lati sono lunghi 3 cm e 4 cm:

$$Ip = \sqrt{3^2 + 4^2}$$

Allora prova:

```
?SQR(3↑2+4↑2)
```

ed appare:

```
5
```

Giorgio è entusiasta: tutti i calcoli numerici gli sembrano molto semplici:

$$\frac{3 + \sqrt{25}}{4}$$

Giorgio scrive:

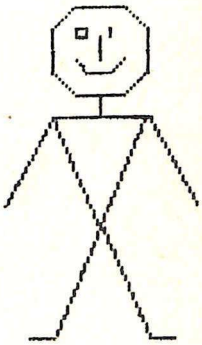
```
?(3+SQR(25))/4
```

ottenendo:

```
2
```

MISTER BIT riappare misteriosamente:

```
PUOI USARE LE SEGUENTI
OPERAZIONI

+ - * / ↑

E LE FUNZIONI

ABS( )    ATN( )
COS( )    EXP( )
INT( )    LOG( )
SGN( )    SIN( )
SQR( )    TAN( )

CONTROLLA SUL MANUALE DEL TUO C-64
E STAI MOLTO ATTENTO ALLE PARENTESI !!
```



ECCO ALCUNI ESEMPI

3+2*X

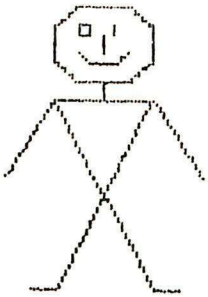
3-5/2

3*X/4

(5-3)/4

3*X↑2

(5-2)/(7+2)



COME LI SCRIVERESTI NON IN BASIC ?

Non ci sono molte difficoltà, pensa Giorgio, e traduce in modo "usuale":

$$3 + 2x$$

$$3 - \frac{5}{2}$$

$$\frac{3x}{4}$$

$$\frac{5 + 3}{4}$$

$$3x^2$$

$$\frac{5 - 2}{7 + 2}$$

Giorgio deve anche semplificare:

$$\left(\frac{2}{3}\right)^7 : \left(\frac{2}{3}\right)^2 =$$

e quindi scrive:

```
?(2/3)↑7/(2/3)↑2
```

il risultato è:

```
.131687243
```

Ma non è questo che interessa a Giorgio: non deve calcolare il valore dell'espressione, deve semplificarla.

Giorgio prova sul suo quaderno:

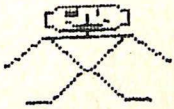
$$\left(\frac{2}{3}\right)^7 : \left(\frac{2}{3}\right)^2 = \left(\frac{2}{3}\right)^{7-2} = \left(\frac{2}{3}\right)^5$$

e guarda il C-64: non è sicuro del risultato, ma come può aiutarlo il C-64?
Sa solo calcolare, lui...

Giorgio preme alcuni tasti perplesso...

Riappare MISTER BIT:

```
PERCHE' NON VERIFICHI ?
NON HAI FATTO ERRORI
DEVONO ESSERE UGUALI !
```



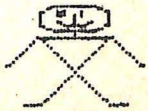
Giorgio rimane interdetto e cerca di capire che cosa gli voglia dire il buffo omino.

All'improvviso, come un lampo: «semplicissimo, se non ho fatto errori le due espressioni devono portare allo stesso risultato!» e rapidamente prova:

```
? (2/3)↑5
```

e batte /RETURN/ con un po' di ansia.

Il risultato è lo stesso:



```
BRAVO GIORGIO !
```

Fa apparire MISTER BIT che sta in un angolo dello schermo.

Giorgio si ricorda di un esercizio che gli aveva dato filo da torcere e di cui non è ancora del tutto sicuro.

Lo ritrova:

$$\frac{3}{\sqrt{7}-2-\sqrt{5}}=$$

$$\frac{3(\sqrt{7}-2+\sqrt{5})}{(\sqrt{7}-2)^2-5}=$$

$$\frac{3(\sqrt{7}-2+\sqrt{5})(3+2\sqrt{7})}{2(3-2\sqrt{7})(3+2\sqrt{7})}=$$

$$\frac{-3(\sqrt{7}-2+\sqrt{5})(3+2\sqrt{7})}{38}$$

Chissà se posso controllare con il C-64, si dice Giorgio.

E scrive con molta attenzione:

```
? 3/(SQR(7)-2-SQR(5))
```

che dà:

```
-1.88641675
```

e poi:

```
? -3*(SQR(7)-2+SQR(5))*(3+2*SQR(7))/38
```

Il risultato è lo stesso: la soluzione era quindi giusta...

Giorgio è molto soddisfatto di sé. Decide di telefonare a Massimo: «Ti devo raccontare tutto quello che ho scoperto!» esclama.

ESPRESSIONI ALGEBRICHE

```
ESPRESSIONI ALGEBRICHE

  1 = CAMBIO DI E1
  2 = CAMBIO DI E2
  3 = VARIABILI
  4 = CALCOLO DI E1
  5 = CALCOLO DI E1&E2

E1=X*X-A*A+2*A*B-B*B
E2=(X+A-B)*(X-A+B)

*******
SCEGLI CHE COSA
VUOI FARE >
```

In matematica, come nelle altre materie scientifiche quali la fisica, la chimica, la statistica, la trigonometria, spesso e volentieri ci si trova dinanzi delle formule. Spesso una formula non è nient'altro che una espressione algebrica, più o meno complessa. Utilizzando un computer come il C-64 si possono valutare due espressioni in contemporanea sostituendo, per ogni variabile, dei valori a piacere; in questo

modo si può controllare rapidamente l'esattezza di un passaggio algebrico o delle semplificazioni che siamo riusciti ad immaginare.

Non più esercizi di cui non possiamo essere sicuri dell'esattezza: il nostro computer non solo ci dirà se abbiamo sbagliato — prima di «rimediare» un brutto voto in classe — ma ci insegnerà anche qual è il metodo corretto per affrontare la verifica di una espressione algebrica utilizzando quel nuovo, raffinato strumento di calcolo che è il nostro computer.

«Ma l'algebra non è solo fatta di numeri» brontola tra sé Giorgio, un quaderno aperto davanti, il calcolatore di lato. Le espressioni algebriche contengono numeri e lettere, le semplificazioni sono spesso complicate, a volte oscure... Per esempio questa espressione:

$$x^2 - a^2 + 2ab - b^2 =$$

Giorgio ha trovato i passaggi seguenti:

$$x^2 - a^2 + 2ab - b^2 =$$

$$x^2 - (a - b)^2 =$$

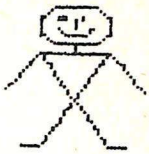
$$(x + a - b)\,(x - a + b)$$

ma non è molto soddisfatto: prova a rifarli in ordine inverso, sembra andar bene... Preme il tasto /RETURN/ del C-64: ed ecco comparire ancora MISTER BIT:

HELLO !

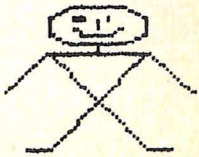
Giorgio scrive la sua espressione, preceduta dal punto interrogativo: ovviamente la risposta è 0. Non serve a nulla, pensa tra sé Giorgio, ma MISTER BIT fa apparire:

SE NON HAI FATTO ERRORI
LE TUE ESPRESSIONI SONO
UGUALI PER QUALUNQUE
VALORE NUMERICO ATTRIBUITO
ALLE LETTERE
X A B.



```
CAMBIO DI E1

E1=X*X-A*A+2*A*B-B*B
E2=(X+A-B)*(X-A+B)

*******
SCRIVI L'ESPRESSIONE
E1=
```

PROVA CON IL PROGRAMMA



```
10 A=3
20 B=13
30 X=-3
40 PRINT (X↑2-A↑2+2*A*B-B↑2)
50 PRINT ((X+A-B)*(X-A+B))
60 END
```

```
E POI PROVA A CAMBIARE
I NUMERI ATTRIBUITI A
X  A  B
NELLE RIGHE 10  20  30.
```

Giorgio scrive il programma, dà il comando /RUN/, poi cambia più volte le righe 10, 20, 30, incredulo...
MISTER BIT gli manda un messaggio:

```
CARICA LE SUBROUTINES
POI COPIA IL PROGRAMMA

'ESPRESSIONI ALGEBRICHE'
```

Giorgio ubbidisce, ha già copiato le subroutines dell'appendice. Le carica dal nastro, poi comincia a copiare con attenzione il listato. Appena finito lo salva su nastro (gli dà il nome «PROVA») e poi con un po' di apprensione dà il comando /RUN/.

Funziona! Il video appare come in figura, proponendo 5 scelte, e mostrando nel contempo le due espressioni attualmente in uso.

Giorgio sceglie la prima attività: lo schermo cambia e può scrivere un'espressione matematica con le variabili e le lettere a scelta (devono essere accettabili per il BASIC, quindi al massimo 2 caratteri, di cui il primo alfabetico) come X1, X2, TI, M, P, ecc.

Al termine dell'espressione Giorgio batte il tasto /RETURN/. Se invece di scrivere la nuova espressione, Giorgio avesse battuto subito il tasto /RETURN/, l'espressione presentata dal video non sarebbe cambiata.

Analogamente la seconda attività permette di cambiare E2, l'altra espressione.

La terza attività (VARIABILI), deve essere eseguita almeno una volta dopo aver caricato il programma, e chiede i nomi delle variabili usate (al massimo 9). Dopo aver inserito l'ultimo nome, alla domanda del nome successivo, Giorgio deve rispondere battendo solo /RETURN/.

Scegliendo una delle altre due attività lo schermo si presenta come nelle figure, chiedendo a Giorgio il valo-

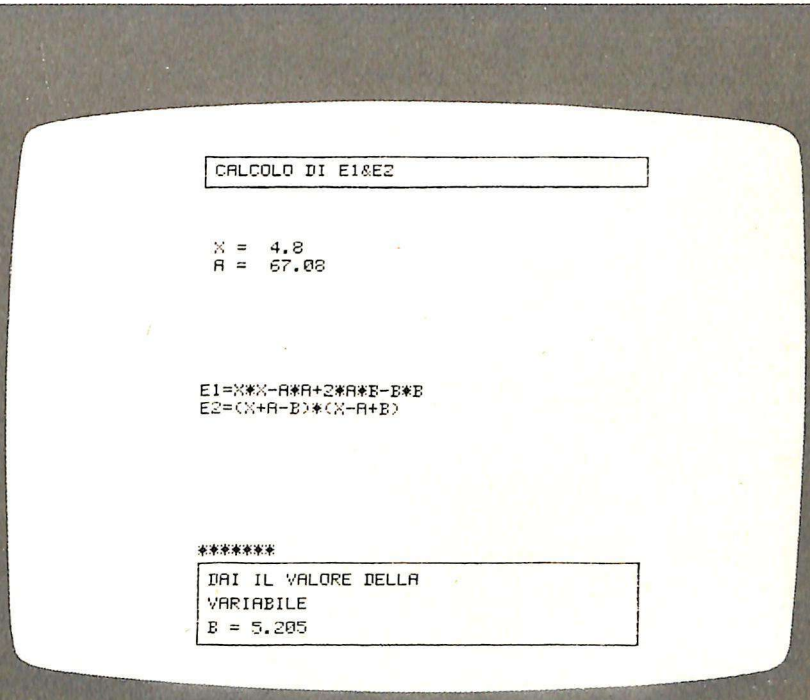
re delle variabili usate e fornendo poi i risultati calcolati in due modi diversi: come numero decimale oppure cercando di trasformarlo in frazione.

Poiché in questo tentativo il C-64 può impiegare molto tempo, Giorgio può dire al programma di saltare questa parte, rispondendo /N/ alla domanda «VUOI CHE CERCHI LA FRAZIONE».

Massimo, quando vede funzionare il programma, gli trova subito un'altra applicazione: «Ma va bene con qualsiasi formula: pensa a tutti i calcoli delle esercitazioni di fisica...».

COME FUNZIONA

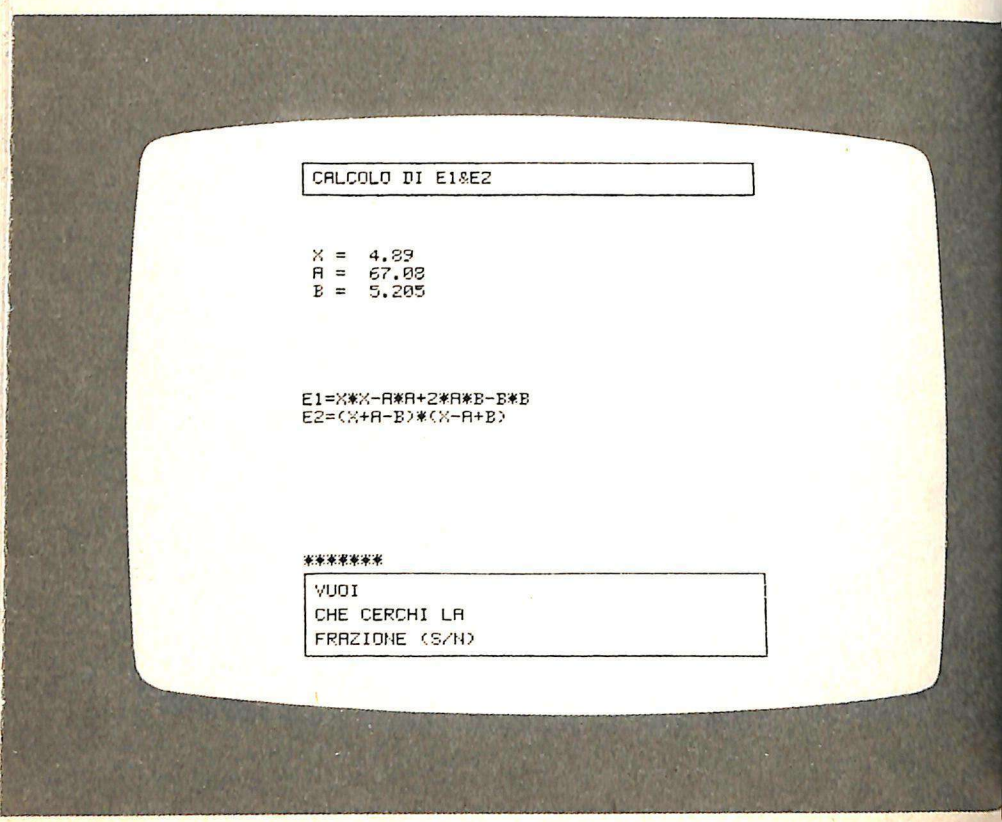
Il programma risulta costituito da cinque parti che si presentano come subroutines, scelte nelle righe 1040-



1100, tramite la SUBROUTINE 9.

È più facile cominciare a vedere il funzionamento del programma partendo dalla quarta attività, cioè quella da riga 5000 a 5830, che è la principale. Questa parte del programma calcola il valore di E1, una espressione algebrica. A riga 5000 si imposta la variabile EX a 1 (vedremo dopo a che cosa serve) e si usa la SUBROUTINE 1 per pulire lo schermo. Si usa anche una subroutine che inizia a riga 7000: è un pezzo di programma usato più volte, che quindi conviene scrivere in forma di subroutine, ma è usata solo da questo programma: è una subroutine «locale». Come si vede da riga 7040, questa subroutine fa apparire a riga 12 dello schermo la nostra espressione E1, se EX è uguale a 1, altrimenti fa apparire, oltre a E1 anche E2 (riga 7060).

Le righe 5304, 5306, 5308 definiscono gli elementi di



un vettore di stringhe: in pratica sono i nomi delle variabili usate nella nostra espressione: la prima variabile usata è *x*, e la stringa VA$(1) è proprio "X".

Questo ci serve per la riga 5420, in cui si chiede il valore delle variabili usando il loro nome. Trascuriamo per il momento la riga 5400; vedremo oltre perché è necessaria.

Usiamo la SUBROUTINE 7 per introdurre i dati, ovviamente, e così i valori numerici introdotti saranno nel vettore (di numeri, questa volta) VR( ).

Nelle righe 5504-5508 vengono attribuiti, alle variabili effettivamente usate nella nostra espressione, i valori che abbiamo introdotto e che sono nel vettore VR( ). Si salta a riga 5600, dove sono calcolate le nostre espressioni, E1, E2. A questo punto la riga 5680 usa la SUBROUTINE 6 per chiedere se si vuole che il programma

```
CALCOLO DI E1&E2

 X =  4.89
 A =  67.08
 B =  5.205


E1=X*X-A*A+2*A*B-B*B
E2=(X+A-B)*(X-A+B)

E1 =-3804.60352
E2 =-3804.60352
E1-E2= 0

*******
VUOI CALCOLARE
CON ALTRI
VALORI (S/N)?
```

effettui il tentativo di trasformare il risultato in frazione.

Se si risponde /S/, la subroutine locale che inizia a riga 7100 prova con frazioni aventi il denominatore minore di 100. Se viene trovata una frazione di cui il valore sia uguale al risultato (entro un'approssimazione di 10^{-6}- riga 7150, per eliminare gli eventuali arrotondamenti di calcolo), la frazione è stampata dopo il risultato (riga 5720).

Se EX è uguale a 1, è stampato solo il risultato E1 (righe 5700-5710), altrimenti è stampato anche E2 (riga 5720) e la differenza E1-E2 (riga 5730).

Si usa la SUBROUTINE 6 per dare l'opportunità di effettuare un altro calcolo (GOTO 5010) oppure passare ad un'altra attività (RETURN) ed il programma continua con riga 1110.

Il programma che inizia a riga 6000, CALCOLO DI E1 E E2 è praticamente quello già visto, con la differenza di riga 6000, che pone EX uguale a 2, e cambia il titolo, per poi continuare con riga 5010: praticamente è lo stesso programma, con due righe di inizio diverse (riga 5000 e riga 6000) ed un comportamento leggermente differente a seconda della variabile EX. Si sarebbe potuto continuare dopo riga 6000 all'incirca come dopo riga 5000, e in questo caso non si avrebbe avuto bisogno della variabile EX, ma il programma sarebbe stato molto più lungo. Il risultato sullo schermo lo potrete vedere in figura.

Vediamo ora le altre parti: la subroutine che inizia in riga 2000, CAMBIO DI E1, si occupa essenzialmente di modificare due righe del programma che abbiamo appena visto, precisamente la riga 5600 e la riga 7000, per permettere di modificare l'espressione E1.

Infatti comincia con la SUBROUTINE 1 (riga 2000) e la subroutine locale 7000, che preparano lo schermo. Poi le righe 2010 e 2020 chiedono E1, la prima espressione, usando la SUBROUTINE 8 - INGRESSO STRINGA -. Se si batte solo il tasto /RETURN/ non succede nulla: RI$ è uguale a "N" e il programma torna a riga 1110. Se invece si introduce un'espressione, questa è nella stringa ST$. Nelle righe 2100 e 2110 si prepara, usando ST$, la nuova riga 5600. Da notare che per inserire delle virgolette in una stringa, si deve usare la funzione CHR$(34). Nella riga 2130 si prepara anche la

stringa P3$, necessaria per rilanciare il programma da riga 1000, poi si passa alla SUBROUTINE 10, che si occupa della effettiva modifica.

In questo modo, le righe 5600 e 7000 conterranno la nuova espressione algebrica che abbiamo introdotto, e vedremo sullo schermo che E1 risulta modificata.

La subroutine che inizia a riga 3000 è in tutto analoga alla precedente, con la differenza che si occupa di E2 e modifica le righe 5610 e 7010.

Resta da esaminare il programma che inizia a riga 4000, che modifica anch'esso delle righe, ma per aggiornare i nomi delle variabili usate.

Le nostre espressioni algebriche possono usare per le variabili nomi qualsiasi di una o due lettere (come per il BASIC, vedi il manuale C-64), come X, A, T, X1, X2, FI etc... A seconda delle variabili usate occorre modificare il programma di calcolo già visto, alle righe da 5304 in poi e 5504 in poi.

Cominciamo da riga 4010: come al solito si pulisce lo schermo con la SUBROUTINE 1, poi si dà ad I il valore preso con una istruzione /PEEK (828)/.

Si deve fare così perché l'uso della SUBROUTINE 10 fa perdere il valore delle variabili. Il valore della variabile I verrà conservato nella locazione di memoria 828 che non è disturbata dalla SUBROUTINE 10 e quindi lo si ritroverà anche dopo. Quando questo programma viene usato la prima volta, provenendo da riga 1100, vediamo che a riga 4000 si dà il valore 1 alla locazione 828, che è ripreso da I in riga 4030, mentre si mette in 828 un valore incrementato, pari a 2 la prima volta. Le righe 4040, 4050, e 4060 preparano la domanda che sarà presentata dalla SUBROUTINE 8.

La stringa con il nome di una variabile sarà in ST$, I è incrementato, cioè vale ora 2, e il programma continua con la riga 4200.

Alle righe 4200-4210 si prepara la nuova riga 5504, mentre alle righe 4220 e 4230 si prepara la nuova riga 5304.

Nella riga 4240 si prepara la stringa P3$, che rilancia il programma a riga 4010, poi si va alla SUBROUTINE 10. Quest'ultima modifica le due righe che deve modificare e il programma riprende a riga 4010.

Ora alla locazione 828 troviamo 2, quindi dopo la SUB-

ROUTINE 8 saranno modificate le righe 5506 e 5306 che conteranno la seconda variabile, e così via.

Vediamo che cosa succede quando abbiamo finito con i nomi delle variabili, e battiamo solo il tasto /RETURN/; in questo caso, a riga 4070 invece di continuare con la riga 4200, continuiamo con la riga 4100.

Viene modificata una sola riga, nel nostro esempio la 5510, con un /GOTO 5600/ per evitare che qualche vecchia variabile influenzi il nuovo calcolo, e P3$ contiene l'indicazione necessaria per ritornare a riga 1000.

Ricordiamoci che in 828 c'è un numero pari al numero delle variabili aumentato di 2. È usato in riga 5400 per attribuirlo a IM e per controllare, a riga 5430, se sono stati chiesti tutti i valori delle variabili o no.

```
 991 REM*********************************
1000 PR$=  "ESPRESSIONI ALGEBRICHE"
1001 REM*********************************
1003 :
1020 GOSUB 8000:GOSUB 7000
1023 :
1040 PN$(1)="CAMBIO DI E1"
1050 PN$(2)="CAMBIO DI E2"
1060 PN$(3)="VARIABILI"
1070 PN$(4)="CALCOLO DI E1"
1080 PN$(5)="CALCOLO DI E1&E2"
1090 PN=5:GOSUB 9400
1100 ON PSGOSUB 2000,3000,4000,5000,6000
1110 GOTO 1000
1113 :
2000 PR$=PN$(1):GOSUB 8000:GOSUB 7000
2003 :
2010 M1$="SCRIVI L'ESPRESSIONE"
2020 M2$="":M3$="E1= ":GOSUB 9000
2030 IF RI$="N" THEN RETURN
2033 :
2100 P1$="7000 E1$="
2110 P1$=P1$+CHR$(34)+ST$+CHR$(34)
2120 P2$="5600 E1="+ST$
2130 P3$="GOTO 1000":GOSUB 9200
```

```
2133 :
3000 PN$(2):GOSUB 8000:GOSUB 7000
3003 :
3010 M1$="SCRIVI L'ESPRESSIONE"
3020 M2$="":M3$="E2= ":GOSUB 9000
3030 IF RI$="N" THEN RETURN
3033 :
3100 P1$="7010 E2$="
3110 P1$=P1$+CHR$(34)+ST$+CHR$(34)
3120 P2$="5610 E2="+ST$
3130 P3$="GOTO 1000":GOSUB 9200
3133 :
4000 POKE 828,1
4010 PR$="VARIABILI":GOSUB 8000
4020 GOSUB 7000
4030 I=PEEK(828):POKE 828,I+1
4040 M1$="SCRIVI I NOMI DELLE"
4050 M2$="VARIABILI USATE"
4060 M3$="VAR "+STR$(I)+"> "
4070 GOSUB 9000:IF RI$="S" THEN 4200
4073 :
4100 P1$=STR$(5500+2*I)+" GOTO 5600"
4110 P2$="":P3$="GOTO 1000":GOSUB 9200
4113 :
4200 P1$=STR$(5500+2*I)+" "+ST$+"=VR("
4210 P1$=P1$+STR$(I-1)+")"
4220 P2$=STR$(5300+2*I)+"VA$("+STR$(I-1)
4230 P2$=P2$+")="+CHR$(34)+ST$+CHR$(34)
4240 P3$="GOTO 4010":GOSUB 9200
4243 :
5000 EX=1:PR$=PN$(4)
5010 GOSUB 8000:GOSUB 7000
5293 :
5304 VA$( 1)="X"
5306 VA$( 2)="A"
5308 VA$( 3)="B"
5383 :
5400 IM=PEEK(828):I=1:M2$="VARIABILE"
5410 M1$="DAI IL VALORE DELLA"
```

```
5420 M3$=VA$(I)+" = ":GOSUB 8800
5423 :
5430 IF I<(IM-1) THEN 5420
5433 :
5504 X=VR( 1)
5506 A=VR( 2)
5508 B=VR( 3)
5510 GOTO  5600
5583 :
5600 E1=X*X-A*A+2*A*B-B*B
5610 E2=(X+A-B)*(X-A+B)
5633 :
5680 M1$="VUOI":M2$="CHE CERCHI LA":M3$=
     "FRAZIONE (S/N)":GOSUB 8400:FR$=RI$
5690 IF FR$="S" THEN M1$="":M2$="":M3$="
     ATTENDERE":GOSUB 8600:POKE 646,C4
5700 RC=16:GOSUB 8100:PRINT"E1 =";E1;:N=
     E1
5710 GOSUB 7100:IF RI$="N" THEN PRINT:GO
     TO 5730
5713 :
5720 PRINT"=";NU;"/";Z
5730 IF EX=1 THEN 5800
5733 :
5740 PRINT"E2 =";E2;:N=E2
5750 GOSUB 7100:IF RI$="N" THEN PRINT:GO
     TO 5780
5760 PRINT"=";NU;"/";Z
5780 PRINT"E1-E2=";E1-E2
5800 M1$="VUOI CALCOLARE"
5810 M2$="CON ALTRI":M3$="VALORI (S/N)?"
5820 GOSUB 8400:IF RI$="N" THEN RETURN
5830 GOTO 5010
5833 :
6000 EX=2:PR$=PN$(5):GOTO 5010
6993 :
7000 E1$="X*X-A*A+2*A*B-B*B"
7010 E2$="(X+A-B)*(X-A+B)"
7023 :
```

```
7040 RC=12:GOSUB 8100:PRINT"E1=";E1$
7050 IF EX=1 THEN RETURN
7060 PRINT"E2=";E2$:RETURN
7063 :
7100 RI$="N":IF FR$="N" THEN RETURN
7110 ND=INT(N):IF ND=N  THEN RETURN
7120 Z=2
7130 K=0
7140 NU=ND*Z+K
7150 IF INT((NU/Z)*10↑6)/(10↑6)=INT(N*10
     ↑6)/(10↑6) THEN 7190
7160 K=K+1:IF K<Z THEN 7140
7170 Z=Z+1:IF Z<100 THEN 7130
7180 RETURN
7190 RI$="S":RETURN
7193 REM*****************************
```

POLINOMI IN X

```
POLINOMI IN X

  1 = POLIN. ORDINATO
  2 = ESPRESSIONE F(X)
  3 = CALCOLO P(X);F(X)
  4 = TABULATO

P(X)= 25*X↑2-74*X+49
F(X)=(5*X-1)*(5*X-1)-16*(4*X-3)

*******
SCEGLI CHE COSA
VUOI FARE >
```

Il primo scoglio che si affronta in algebra sono i polinomi.

Finché si è trattato di monomi più o meno tutto è andato bene anche perché un solo coefficiente e una sola parte letterale sono «piccoli», facili da capire e da gestire. Quando però i monomi si ammucchiano, diventano tanti, uno in fila all'altro come tanti vagoni di un lungo treno merci, si resta molto spesso allibiti e stupefatti nel vederseli passare davanti. Eppure l'espressione algebrica può contenere come variabile

«soltanto» la x — abbiamo cioè quello che, pomposamente, in algebra viene definito come polinomio in x —. Bene, il nostro C-64 non solo è in grado di paragonare i valori di due polinomi tra di loro ma può anche calcolarne il risultato per tantissimi valori che decidiamo di assegnare alla variabile x. Questo calcolo immediato e rapido del valore di un polinomio non solo ci metterà in grado di effettuare rapidi controlli e verifiche ma anche di ottenere facilmente tutti gli elementi necessari per tracciare grafici e curve! Altro che i lunghi, faticosi calcoli da eseguire con una calcolatricetta tascabile o — addirittura — a mano con la conseguente enorme possibilità di errori!

Siamo o non siamo nell'epoca dei computer? Il nostro amato schiavo si dia da fare: non solo il compito sarà fatto presto e bene ma ci rimarrà anche un mucchio di tempo per uscire.

Massimo è convinto siano più facili: «Vedi, in questi polinomi compare solo la lettera *x*: è meglio, no?». Giorgio ha invece qualche perplessità:

$$(5x - 1)^2 - 16\,(4x - 3) = 25x^2 - 74x + 49$$

È vero che contiene solo la variabile X, e quindi è meno facile fare errori, ma a Giorgio piacerebbe poter controllare ogni passaggio, per avere la soddisfazione di dire: «Ho fatto bene!». Massimo lo provoca: «Ci vorrebbe un programmino...».

Giorgio copia velocemente POLINOMI IN X, che permette di calcolare e confrontare un polinomio ordinato ed un'espressione in *x*.

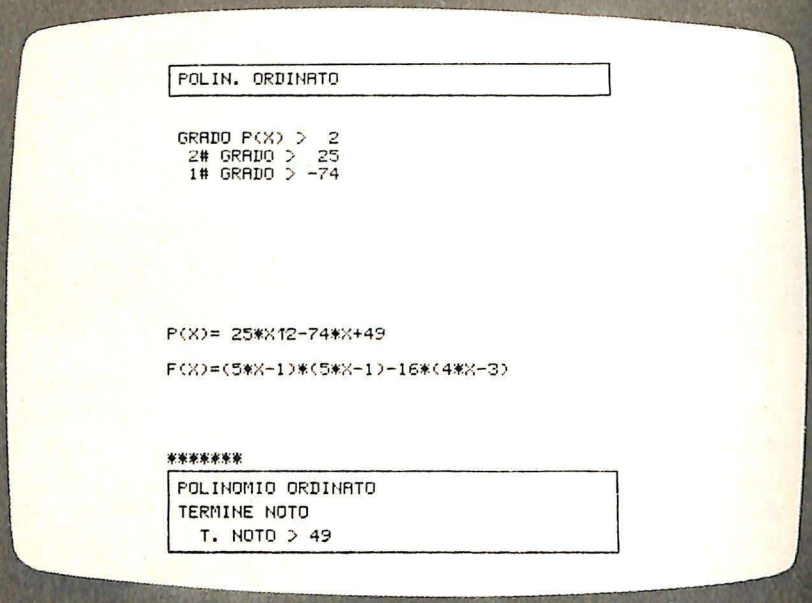
Delle quattro attività tra cui scegliere, le prime due riguardano l'introduzione di un polinomio ordinato e di un'espressione algebrica nella variabile *x*. Se il polinomio è ordinato, appaiono delle domande che chiedono a Giorgio prima il grado massimo del polinomio, poi i coefficienti in ordine decrescente fino al termine noto. Se un termine manca, Giorgio batterà solo il tasto /RETURN/. Il polinomio è poi ricostruito dal programma ed

appare sullo schermo come P(X). La funzione F(X) è invece scritta direttamente da Giorgio secondo le regole del BASIC, come già visto nel programma precedente. La terza attività permette di calcolare sia P(X) sia F(X) per un valore qualsiasi di X: è quello che fa Giorgio per controllare il suo polinomio (v. figura).

Per terminare questa attività, basta non introdurre un valore per X, ma battere solo il tasto /RETURN/: il programma ripresenta la scelta iniziale.

L'ultima possibilità (TABULATO) consiste nel calcolare automáticamente P(X) e F(X), per valori di X a partire da un minimo (START) fino ad un massimo (END), aumentando ogni volta X della stessa quantità (STEP).

Massimo è ammirato: «Questa ultima possibilità può essere sfruttata quando si deve disegnare un grafico!».



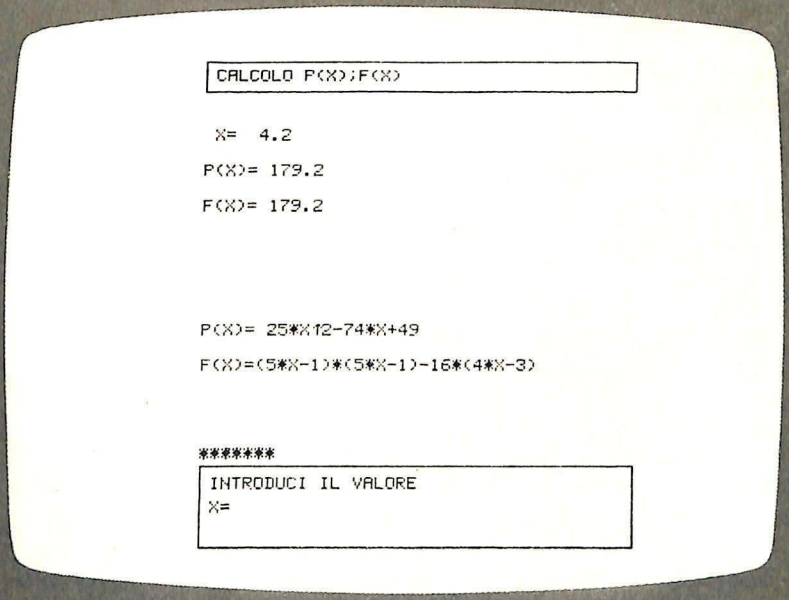
COME FUNZIONA

Per prima cosa (riga 110) si usa la SUBROUTINE 1 per pulire lo schermo, e la subroutine locale di riga 500 che fa apparire a riga 13 dello schermo sia P(X) che F(X). Preparato così lo schermo, si può scegliere tra le quattro funzioni del programma tramite la SUBROUTINE 9, a riga 300.

Lo schermo si presenterà come in figura.

Vediamo per prima la funzione principale — CALCOLO P(X); F(X) — che inizia a riga 3000. Le solite subroutines si occupano dello schermo, poi si chiede il valore della variabile X per la quale vogliamo calcolare P(X) e F(X) (righe 3010-3030), tramite la SUBROUTINE 7.

Se si introduce un valore, poiché I=0 (riga 3020), lo ritroviamo in VR (0) ed infatti nella riga 3100 si attribui-



sce ad X tale valore, poi si va ad un'altra subroutine locale, a riga 3500.

Questa subroutine a riga 3500 calcola il valore di P(X) ed a riga 3510 calcola invece quello di F(X); le righe 3520 e 3530 stampano i risultati su due righe dello schermo che prima puliscono, stampando S$, stringa di soli spazi. Con la riga 3540 si ritorna al programma chiamante.

La riga 3100, dopo aver usato la subroutine 3500, continua facendo ricominciare il programma a riga 3020.

Se però non si inserisce un valore per X, ma si batte solo /RETURN/, il programma termina a riga 3040, tornando a riga 320.

A riga 4000 comincia la funzione TABULATO del programma, al solito preparando lo schermo. Vengono poi

```
TABULATO

   START X= 10
   END   X= 40
   STEP: X= 2.5

X= 10

P(X)= 1809

F(X)= 1809

P(X)= 25*X↑2-74*X+49

F(X)=(5*X-1)*(5*X-1)-16*(4*X-3)


*******
SE VUOI
CONTINUARE
PREMI UN TASTO
```

chiesti tre valori numerici: il valore iniziale di X (righe 4010-4030), quello finale (righe 4040-4050) ed il valore dell'incremento (righe 4060-4070).

I numeri introdotti saranno rispettivamente in VR(0), VR(4), VR(2).

Cominciando dal valore iniziale (riga 4100), viene stampato a riga 6 dello schermo il valore di X ed è usata la subroutine locale a riga 3500 per calcolare e stampare i risultati (righe 4110, 4120). La riga 4130 calcola il nuovo valore di X, e la riga 4300 usa la SUBROUTINE 5 -PAUSA- per far vedere i risultati. Quando si preme un tasto, se non si è superato il valore finale, il programma presenta i nuovi valori (ricomincia da riga 4110), altrimenti termina ritornando a riga 320.

Le altre due funzioni del programma servono per definire P(X) e F(X).

Nel caso di P(X) si tratta di un polinomio ordinato, e il programma, dopo aver preparato al solito lo schermo (riga 1000), chiede il grado massimo (righe 1010-1030), tramite l'uso della SUBROUTINE 7.

Il numero introdotto è in VR (0) ed è attribuito anche alla variabile GM.

Verrà controllato che sia compreso tra 1 e 9 (riga 1040), ripresentando la domanda se GM supera tali limiti.

Le righe 1100 e 1110 preparano le domande per i coefficienti, chiesti tramite la SUBROUTINE 7, tante volte quanto è il grado del polinomio (la riga 1130 ripresenta la domanda il numero di volte necessario). Le righe 1200 e 1210 chiedono il termine noto.

A questo punto si deve costruire una stringa con il polinomio, utilizzando i coefficienti introdotti e moltiplicando per la variabile X con l'esponente appropriato. Di questo si occupano le righe da 1300 a 1400.

Ora PX$ contiene il polinomio ordinato che abbiamo introdotto. Le righe da 1500 a 1530 preparano le stringhe per usare la SUBROUTINE 10, per modificare la riga 510 e la riga 3500, riprendendo il programma alla riga 100.

Più semplice è la parte di programma che modifica F(X) e che comincia a riga 2000, pulendo lo schermo. L'espressione F(X) è chiesta direttamente come stringa (righe 2010-2030) tramite la SUBROUTINE 8. Se non si

introduce nulla il programma torna a riga 320 (riga 2040), altrimenti prosegue, preparando le stringhe per modificare le righe 520 e 3510. Poi (riga 2120) continua con la riga 1530, che prepara l'altra stringa (RUN 100) e va alla SUBROUTINE 10.

```
 99 REM*********************************
100 PR$=        "POLINOMI IN X"
103 REM*********************************
105 :
110 GOSUB 8000:GOSUB 500
113 :
200 PN$(1)="POLIN. ORDINATO
           "
```

```
ESPRESSIONE F(X)

P(X)= 25*X↑2-74*X+49

F(X)=(5*X-1)*(5*X-1)-16*(4*X-3)

*******
SCRIVI L'ESPRESSIONE
F(X)=
```

```
 210 PN$(2)="ESPRESSIONE F(X)
                 "
 220 PN$(3)="CALCOLO P(X);F(X)
                 "
 230 PN$(4)="TABULATO
                 "
 300 PN=4:GOSUB 9400
 310 ON PSGOSUB 1000,2000,3000,4000
 320 GOTO 100
 323 :
 500 RC=14:GOSUB 8100
 503 :
 510 PX$=" 25*X↑2-74*X+49"
 520 FX$="(5*X-1)*(5*X-1)-16*(4*X-3)"
 523 :
 530 PRINT "P(X)=";PX$:PRINT "[CRSR↑]F(X
     )=";FX$
 540 RETURN
 543 :
1000 PR$=PN$(1):GOSUB 8000:GOSUB 500
1003 :
1010 M1$="POLINOMIO ORDINATO":PX$=""
1020 M2$="GRADO MASSIMO":I=0
1030 M3$="GRADO P(X) > ":GOSUB 8800
1040 GM=VR(0):IF GM<1 OR GM>9 THEN 1020
1043 :
1100 M2$="COEFFICIENTE DI"
1110 M3$=STR$(GM+1-I)+"# GRADO > "
1120 GOSUB 8800
1130 IF I<GM+1 THEN 1110
1133 :
1200 M2$="TERMINE NOTO"
1210 M3$="  T. NOTO > ":GOSUB 8800
1213 :
1300 FOR K=1 TO GM+1
1310 IF VR(K)=0 THEN 1400
1320 IF PX$="" OR VR(K)<0 THEN PX$=PX$+S
     TR$(VR(K)):GOTO 1340
1330 PX$=PX$+"+"+MID$(STR$(VR(K)),2)
```

```
1340 IF K-GM=1 THEN 1400
1350 PX$=PX$+"*X"
1360 IF K-GM=0 THEN 1400
1370 PX$=PX$+"↑"+MID$(STR$(GM-K+1),2)
1400 NEXT
1403 :
1500 P1$="510 PX$="
1510 P1$=P1$+CHR$(34)+PX$+CHR$(34)
1520 P2$="3500 PX="+PX$
1530 P3$="RUN 100":GOTO 9200
1533 :
2000 PR$=PN$(2):GOSUB 8000:GOSUB 500
2003 :
2010 M1$="SCRIVI L'ESPRESSIONE"
2020 M2$="":I=0
2030 M3$="F(X)= ":GOSUB 9000
2040 IF RI$="N" THEN RETURN
2043 :
2100 P1$="520 FX$="
2110 P1$=P1$+CHR$(34)+ST$+CHR$(34)
2120 P2$="3510 FX="+ST$:GOTO 1530
2123 :
3000 PR$=PN$(3):GOSUB 8000:GOSUB 500
3003 :
3010 M1$="INTRODUCI IL VALORE"
3020 M2$="":I=0
3030 M3$="X= ":GOSUB 8800
3033 :
3040 IF RI$="N" THEN RETURN
3043 :
3100 X=VR(0):GOSUB 3500:GOTO 3820
3133 :
3500 PX= 25*X↑2-74*X+49
3510 FX=(5*X-1)*(5*X-1)-16*(4*X-3)
3513 :
3520 PRINT S$;"[SHIFT]+[CRSR↑]P(X)=";PX
3523 :
3530 PRINT S$;"[SHIFT]+[CRSR↑]F(X)=";FX
3540 RETURN
```

```
3543 :
4000 PR$=PN$(4):GOSUB 8000:GOSUB 500
4003 :
4010 M1$="INTRODUCI IL VALORE"
4020 M2$="INIZIALE DI X:":I=0
4030 M3$="  START X=":GOSUB 8800
4033 :
4040 M2$="FINALE DI X:"
4050 M3$="  END   X=":GOSUB 8800
4053 :
4060 M2$="DELL'INCREMENTO:"
4070 M3$="  STEP: X=":GOSUB 8800
4073 :
4100 X=VR(0)
4103 :
4110 RC=8:GOSUB 8100
4120 PRINT S$;"[SHIFT]+[CRSR↑]X=";X:GOSU
     B 3500
4123 :
4130 X=X+VR(2)
4300 M2$="CONTINUARE":GOSUB 8200
4303 :
4310 IF X<=VR(1) THEN 4110
4320 RETURN
4323 :
4325 REM*****************************
8000 PRINT "[SHIFT]+[HOME]":C0=0:C1=3:C2
     =6:C3=0:C4=5
8005 C5=6:C6=7:MV=1024:MC=55296
8010 POKE 53265,PEEK(53265) OR 64
8015 POKE 53280,C0:POKE 53281,C3
8020 S$="":FOR I=0 TO 39:S$=S$+" ":NEXT
```

POLINOMI ORDINATI

```
POLINOMI ORDINATI

  1 = NUOVO POLINOMIO
  2 = IN MEMORIA MX←PX
  3 = DA MEMORIA PX←MX
  4 = SOMMA      PX+QX
  5 = DIFFERENZA PX-QX
  6 = PRODOTTO   PX*QX
  7 = QUOZIENTE  PX/QX

PX= 3*X12-2*X+6

QX= 7*X13-3*X

------
MX= 7*X13-3*X

*******
SCEGLI CHE COSA
VUOI FARE >
```

Questa dei polinomi ordinati è sempre stata una iattura per tutti gli studenti intelligenti e «bravi» in matematica. In effetti le operazioni sui polinomi ordinati sono in se stesse molto semplici e, dopo un po' di studio, non dovrebbero presentare nessuna difficoltà. In pratica, però, proprio i polinomi ordinati sono la classica buccia di banana su cui si possono rischiare dei voti veramente brutti. Per giunta le ope-

razioni sui polinomi ordinati sono tanto più antipatiche in quanto questo argomento viene sviluppato dai professori — forse lo fanno apposta — proprio poco prima delle interrogazioni conclusive del quadrimestre. Siamo in pieno «rischio»; se sbagliamo una cosa ripetitiva e concettualmente facile non ce la possiamo prendere con nessuno salvo che con noi stessi e con la nostra faciloneria. Ci servirebbe un aiuto, un amico attento, pignolo, che ci permetta di non uscir fuori dall'alveo della correttezza e del rigore algebrico. Questo amico c'è ed è proprio il nostro caro C-64 che docile e gentile si trasforma in una «calcolatrice per polinomi» che, rapidamente e *senza errori*, è in grado di eseguire ognuna delle quattro operazioni fondamentali.

Perfino la «complicatissima» divisione di due polinomi può essere verificata immediatamente. Noi ed il nostro computer insieme e sicuri anche nei meandri più paludosi di questa algebra a volte bella e a volte «greve».

I polinomi si possono sommare, sottrarre, moltiplicare, dividere...

«Ma non esiste una calcolatrice per polinomi!» si arrabbia Giorgio con il suo C-64.

MISTER BIT compare in alto:

```
NE SEI SICURO ?
PROVA UN PO' CON QUESTO PROGRAMMA
```

Giorgio quindi carica il programma SUBROUTINES, poi si mette di buona lena a copiare il listato.

Completato il programma, Giorgio lo salva sul nastro, e poi prova: lo schermo appare come in figura. Scegliendo NUOVO POLINOMIO, si può inserire P(X), scegliendo ancora NUOVO POLINOMIO il polinomio P(X) diviene Q(X), e quello nuovo che si inserisce andrà in P(X). Il massimo grado ammesso è 9.

Il risultato di un'operazione si ritrova in P(X), per per-

mettere operazioni in catena, come con una calcolatrice tascabile.

Giorgio, per esempio, ha calcolato direttamente:

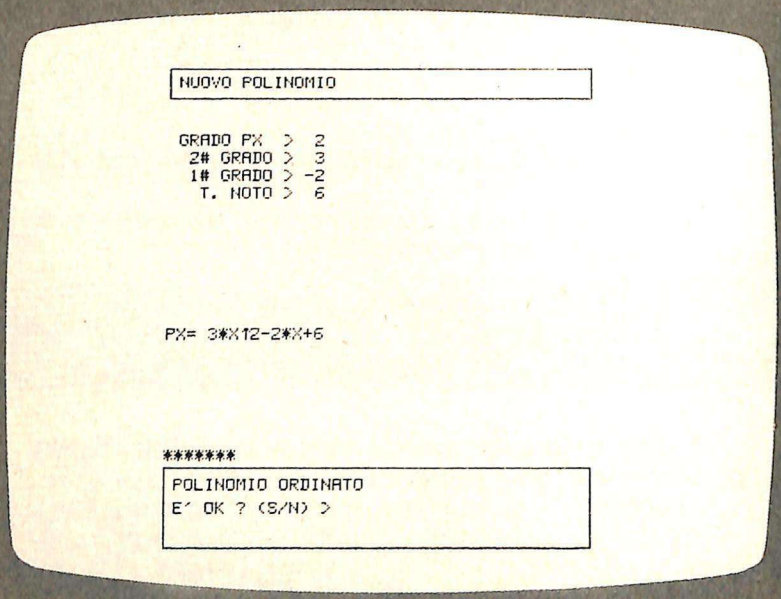
$$(3x + 5)^2 (7x - 4) + 3x^2 - 2x$$

Ha introdotto due volte *3x+5* e, facendo il prodotto, ha ottenuto in PX:

$$9x^2 + 30x + 25$$

Poi ha introdotto *7x-4*, quindi può effettuare un secondo prodotto, e poi sommare *3x-2x* al risultato:

$$63x^3 + 177x^2 + 53x - 100$$



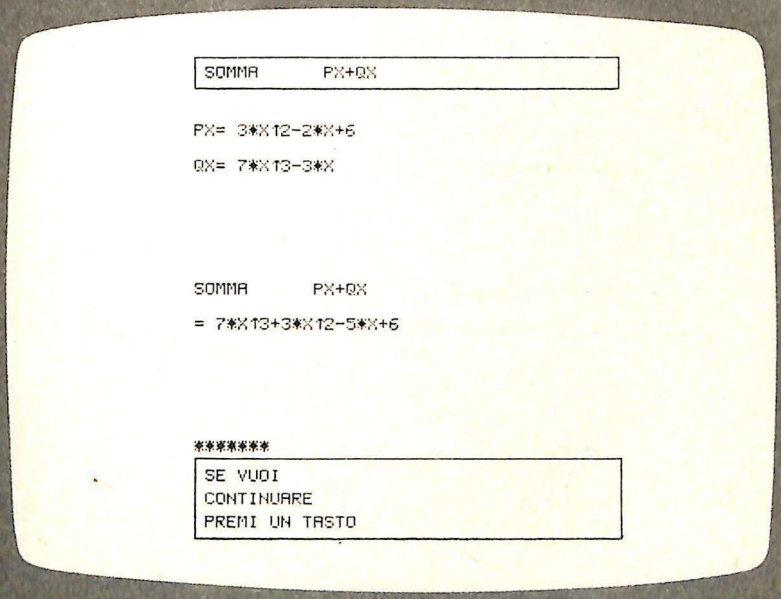
COME FUNZIONA

Anche questo programma inizia pulendo lo schermo, presentando i due polinomi (riga 110) e poi chiedendo quale operazione si intenda fare (righe 200-300).

A seconda della risposta alla SUBROUTINE 9 si continua con la subroutine appropriata.

Se si sceglie NUOVO POLINOMIO si va a riga 1000, che per prima cosa pulisce lo schermo, poi sposta in Q(X) il polinomio P(X). Questo avviene in due fasi: a riga 1000 per le stringhe, e a riga 1010 per i coefficienti. Infatti i polinomi ordinati in questo programma sono conservati sia come stringa (PX$ e QX$), sia come vettori di coefficienti (PX(K) e QX(K)).

Il programma prosegue con le righe 1100-1120, che usano la SUBROUTINE 7 per chiedere il grado del nuo-



vo polinomio, mentre la riga 1130 controlla che sia inferiore a 9, altrimenti ripete la domanda.

Vengono poi chiesti i coefficienti per i vari gradi (righe 1200-1230) e in ultimo il termine noto. Se, per esempio, il nostro polinomio è di terzo grado, ora VR(0)=3, e VR(I), VR(2), VR(3) contengono rispettivamente i coefficienti di terzo, secondo e primo grado, mentre VR(4) contiene il termine noto. Questi numeri sono trasportati nel vettore PX(K) alla riga 1320, poi si usa la subroutine locale a 2800 per ottenere dai coefficienti PX( ) il polinomio in forma di stringa, moltiplicandoli per la variabile X con gli opportuni esponenti.

La riga 1340 fa apparire il polinomio P(X) a riga 14 dello schermo, e si usa la SUBROUTINE 6 per controllare che non ci siano errori. In caso di sbaglio si può reintrodurre il polinomio P(X) ritornando a riga 1100, che ri-

```
PRODOTTO   PX*QX

PX= 3*X↑2-2*X+6

QX= 7*X↑3-3*X



PRODOTTO   PX*QX

= 21*X↑5-14*X↑4+33*X↑3+6*X↑2-18*X



*******
SE VUOI
CONTINUARE
PREMI UN TASTO
```

chiede tutti i valori, altrimenti si torna alla scelta iniziale.

Scegliendo ancora il programma 1, il polinomio già introdotto diviene Q(X) e quello che si introduce diverrà P(X).

Introdotti così due polinomi, si può passare ad eseguire operazioni su di essi. La SOMMA PX+QX inizia a riga 2000, utilizzando la subroutine locale che inizia a riga 2700. Questa subroutine pulisce lo schermo, fa apparire i due polinomi (righe 2700, 2710), e controlla che esistano entrambi, moltiplicando i gradi dei polinomi tra loro. Se manca un polinomio, il suo grado è zero, ed il prodotto è anch'esso zero: in questo caso (riga 2730) appare un messaggio di errore, e la stringa RI$ è uguale a "N". Infatti tornando al programma chiamante, vediamo che a riga 2010 si controlla RI$ e, in caso di errore, non si continua nella somma, ma si salta a riga 2400.

```
QUOZIENTE  PX/QX

PX= 1*X↑4-6*X↑3+2*X↑2-5*X+4

QX= 1*X↑2-2*X+4

RESTO

=-9*X+44

QUOZIENTE  PX/QX

= 1*X↑2-4*X-10



*******
SE VUOI
CONTINUARE
PREMI UN TASTO
```

La somma viene effettuata sommando i coefficienti di pari grado tra loro, mettendo il risultato nel vettore PX( ). Di questo si occupano le righe 2120-2220. La riga 2240 cancella il polinomio Q(X), che non serve più, sia come stringa, sia mettendo a zero il suo grado, poi si utilizza la subroutine locale per ottenere la stringa PX$, che rappresenta il polinomio somma cercato.

Le righe 2300, 2310 mostrano il risultato sullo schermo, e le righe 2400, 2410 utilizzano la SUBROUTINE 5, PAUSA, prima di tornare a riga 320.

Il programma che inizia a riga 3000, DIFFERENZA PX—QX è nelle prime due righe identico al precedente (righe 3000, 3010), dopo di che tutti i coefficienti di Q(X) sono cambiati di segno (riga 3020). Ora il programma può continuare come una somma, ed infatti continua con riga 2100 della somma.

Il programma del PRODOTTO inizia a riga 4000, anch'esso con la subroutine locale 2700.

Per effettuare il prodotto, prima si spostano i coefficienti PX( ) nel vettore VR( ), riga 4100, poi si calcola il grado del risultato (riga 4110) e si rendono nulli tutti i coefficienti di PX( ). Dopo di che si moltiplicano tutti i coefficienti di VR(K) per ogni coefficiente di QX(H) sommando il risultato al giusto posto in PX( ). Al termine PX( ) conterrà i coefficienti del risultato, e il programma può continuare a riga 2240, per le rimanenti cose da fare, che sono le stesse di quelle della somma.

La parte di programma per la DIVISIONE, oltre i controlli effettuati dalla subroutine 2700, controlla anche il grado del divisore Q(X) non sia maggiore del grado del dividendo P(X) (riga 5100), attribuito a VR(0). Infatti al termine della divisione VR( ) conterrà il risultato, e PX( ) il resto.

La riga 5210 calcola i coefficienti del quoziente, mentre la riga 5230 calcola il resto; la riga 5240 usa la subroutine di riga 2800 per trasformare il resto in stringa PX$, che è visualizzato da riga 5250.

Il quoziente viene trasferito nel vettore PX( ) dalla riga 5260, ed il programma può ora continuare come quello della moltiplicazione, andando a riga 2240.

A riga 6000 c'è la subroutine IN MEMORIA: il polinomio PX è copiato in MX, sia come coefficienti (riga 6000), sia come stringa (riga 6010).

Per richiamare un polinomio dalla memoria occorre prima spostare PX in QX, poi MX in PX (righe 7000-7020).

```
100 DIM PX(20),QX(20),VR(20),MX(20)
101 REM*****************************
102 PR$=      "POLINOMI ORDINATI"
103 REM*****************************
107 :
110 GOSUB 8000:RC=12:GOSUB 8100
120 PRINT"PX=";PX$:PRINT"[CRSR↑]QX=";QX
    $
130 PRINT"[CRSR↑]-------":PRINT"MX=";MX$
133 :
200 PN$(1)="NUOVO POLINOMIO  "
220 PN$(2)="IN MEMORIA MX←PX "
230 PN$(3)="DA MEMORIA PX←MX "
240 PN$(4)="SOMMA      PX+QX "
250 PN$(5)="DIFFERENZA PX-QX "
260 PN$(6)="PRODOTTO   PX*QX "
270 PN$(7)="QUOZIENTE  PX/QX "
300 PN=7:GOSUB 9400
310 ON PSGOSUB 1000,6000,7000,2000,3000
    ,4000,5000
320 GOTO 102
1000 PR$=PN$(1):GOSUB 8000:QX$=PX$
1010 FOR K=0 TO PX(0)+1:QX(K)=PX(K):NEXT
1013 :
1100 M1$="POLINOMIO ORDINATO":I=0
1110 M2$="GRADO MASSIMO :"
1120 M3$="GRADO PX  > ":GOSUB 8800
1130 GM=VR(0):IF GM<1 OR GM>9 THEN 1100
1133 :
1200 M2$="COEFFICIENTE DI"
1210 M3$=STR$(GM+1-I)+"# GRADO > "
1220 GOSUB 8800
1230 IF I<GM+1 THEN 1210
1233 :
```

```
1300 M2$="TERMINE NOTO"
1310 M3$="  T. NOTO > ":GOSUB 8800
1313 :
1320 FOR K=0 TO GM+1:PX(K)=VR(K):NEXT
1330 GOSUB 2800
1333 :
1340 RC=14:GOSUB 8100:PRINT"PX=";PX$
1400 M2$="":M3$="E' OK ? (S/N) > "
1410 GOSUB 8400:IF RI$="N" THEN 1100
1413 :
1500 RETURN
1503 :
2000 PR$=PN$(4):GOSUB 2700
2010 IF RI$="N" THEN 2400
2013 :
2100 K1=PX(0):K2=QX(0)
2110 IF K2>K1 THEN 2200
2113 :
2120 FOR K=0 TO K2
2130 PX(K1+1-K)=PX(K1+1-K)+QX(K2+1-K)
2140 NEXT:GOTO 2240
2143 :
2200 FOR K=0 TO K1
2210 QX(K2+1-K)=PX(K1+1-K)+QX(K2+1-K)
2220 NEXT
2223 :
2230 FOR K=0 TO K2+1:PX(K)=QX(K):NEXT
2240 QX$="":QX(0)=0:GOSUB 2800
2243 :
2300 RC=12:GOSUB 8100
2310 PRINTPR$:PRINT"[CRSR↑]=";PX$
2313 :
2400 M2$="CONTINUARE":GOSUB 8200
2410 RETURN
2413 :
2700 GOSUB 8000:PRINT"PX=";PX$
2710 PRINT"[CRSR↑]QX=";QX$:RI$="S"
2713 :
2720 IF PX(0)*QX(0)<>0 THEN RETURN
```

```
2723 :
2730 PRINT"[3 CRSR↑]  CORREGGERE I POLIN
     OMI"
2740 RI$="N":RETURN
2743 :
2800 PX$="":GM=PX(0)
2803 :
2810 FOR K=1 TO GM+1
2820 IF PX(K)=0 THEN 2900
2830 IF PX$="" OR PX(K)<0 THEN PX$=PX$+S
     TR$(PX(K)):GOTO 2850
2840 PX$=PX$+"+"+MID$(STR$(PX(K)),2)
2850 IF K-GM=1 THEN 2900
2860 PX$=PX$+"*X"
2870 IF K-GM=0 THEN 2900
2880 PX$=PX$+"↑"+MID$(STR$(GM+1-K),2)
2900 NEXT:RETURN
2903 :
3000 PR$=PN$(5):GOSUB 2700
3010 IF RI$="N" THEN 2400
3013 :
3020 FOR K=1 TO QX(0)+1:QX(K)=(-1)*QX(K)
3030 NEXT:GOTO 2100
3033 :
4000 PR$=PN$(6):GOSUB 2700
4020 IF RI$="N" THEN 2400
4023 :
4100 FOR K=0 TO PX(0)+1:VR(K)=PX(K):NEXT
4110 PX(0)=VR(0)+QX(0)
4120 FOR K=1 TO PX(0)+1:PX(K)=0:NEXT
4123 :
4200 FOR K=1 TO VR(0)+1
4210 FOR H=1 TO QX(0)+1
4213 :
4220 PX(H+K-1)=PX(H+K-1)+VR(K)*QX(H)
4230 NEXT:NEXT:GOTO 2240
4243 :
4273 :
5000 PR$=PN$(7):GOSUB 2700
```

```
5010 IF PX(0)<QX(0) THEN GOSUB 2730
5020 IF RI$="N" THEN 2400
5023 :
5100 VR(0)=PX(0)-QX(0)
5200 FOR K=1 TO VR(0)+1
5210 VR(K)=PX(K)/QX(1)
5220 FOR H=1 TO QX(0)+1
5230 PX(H+K-1)=PX(H+K-1)-VR(K)*QX(H)
5240 NEXT:NEXT:GOSUB 2800
5243 :
5250 PRINT"[CRSR↑]RESTO":PRINT"[CRSR↑]="
     ;PX$
5260 FOR K=0 TO VR(0)+1:PX(K)=VR(K):NEXT
5270 GOTO 2240
6000 FOR K=0 TO PX(0)+1:MX(K)=PX(K):NEXT
6010 MX$=PX$:RETURN
7000 QX$=PX$:PX$=MX$
7010 FOR K=0 TO PX(0)+1:QX(K)=PX(K):NEXT
7020 FOR K=0 TO MX(0)+1:PX(K)=MX(K):NEXT
7030 RETURN
7033 REM**********************************
```

EQUAZIONI DI I° GRADO

```
EQUAZIONI DI 1# GRADO

  1 = CAMBIO F(X)

  2 = SOLUZ. F(X)=0

  3 = CALCOLO F(X)


F(X)=(2+X)/3-1/2*(3*X/2+1/3)+(3*X-2)/4-(
2*X+1)/3


*******
SCEGLI CHE COSA
VUOI FARE >
```

A questo punto chi è bravo in matematica si ribella. Nella tranquilla certezza di saper bene utilizzare quello che forse è il più potente mezzo dell'algebra e di tante altre discipline matematiche può pensare tra di sé: «Voglio proprio vedere che cosa questo computer *senza intelligenza* è in grado di insegnarmi!». E invece... Proprio per i più in gamba questo capitolo e il relativo programma costituirà una autentica sorpresa. Il modo elegante e sintetico con cui una

mente umana affronta, calcola e risolve una equazione è soltanto uno dei mezzi che la matematica ci mette a disposizione, e, per quanto elegante, non è probabilmente il più rapido, almeno se sul nostro tavolo, sul nostro banco, abbiamo il nostro fido computer in grado di lavorare per noi. Chi non è tanto bravo in matematica sfrutterà questo programma come tutti gli altri, per avere rapidamente risultati e fare bella figura con tutti, insegnanti compresi; chi è già bravo, però, troverà qualcosa in più: la conoscenza e la coscienza che la matematica, ogni campo della matematica, presenta mille diverse e scintillanti sfaccettature.

Dove la nostra mente umana trova l'alterigia delle sue capacità in un momento di sintesi, usando pochi elementi per volta, la «mente» grezza ma superbamente efficiente del computer approssima con una vera valanga di valori la soluzione finale.

Certo, sul térreno delle equazioni di primo grado lo scontro tra uomo e computer si concluderà con un uno-a-uno, in parità; ma quando gli altri campi di gara saranno resi accidentati ed impervi dalle x^2, dalle x^3, dalle x^n... è meglio imparare subito questa potente chiave di volta che il computer usa anche sulle «iniziali» equazioni di primo grado.

Giorgio ha a che fare con un'equazione di primo grado:

$$\frac{2+x}{3}-\frac{1}{2}\left(\frac{3}{2}x+\frac{1}{3}\right)+\frac{3x-2}{4}-\frac{2x+1}{3}=0$$

Con un paio di passaggi ottiene:

$$4\,(2+x)-9x-2+3\,(3x-2)-4\,(2x+1)=0$$

sviluppando e raccogliendo:

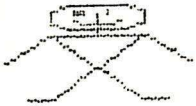
$$4x-9x+9x-8x+8-2-6-4=0$$

cioè:

$$x + 1 = 0$$

$$x = 1$$

MISTER BIT lancia un messaggio laconico nello schermo:



LO SAPEVO SUBITO!
BASTA CALCOLARE.

Giorgio scrive il programma EQUAZIONI DI I° GRADO, e scopre che in effetti il COMMODORE calcola molte volte l'espressione data, controllando il risultato: fa in modo di avvicinarsi sempre più a zero e si ferma quan-

```
CAMBIO F(X)

 INTRODUCI LA NUOVA
 EQUAZIONE DI PRIMO
      GRADO

F(X)=(2+X)/3-1/2*(3*X/2+1/3)+(3*X-2)/4-(
2*X+1)/3

*******
F(X)=
```

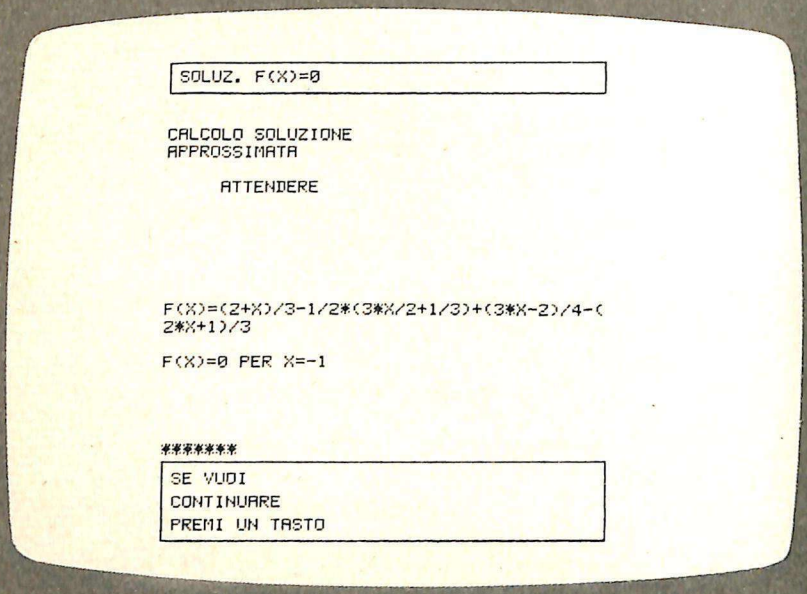
do è vicino abbastanza. Un modo inconcepibile per un uomo, ma il più facile per un calcolatore.

Il programma presenta a Giorgio tre alternative: introdurre l'equazione, il calcolo della soluzione, il calcolo dell'espressione per particolari valori di *x*.

Ovviamente la soluzione è la stessa (v. figura) di quella trovata da Giorgio, ma a volte può differire di qualche cifra decimale dalla soluzione esatta (è una soluzione approssimata). A volte il programma può non trovarla, ed allora si ferma dopo 100 tentativi; in questo caso è probabile che ci sia un errore in F(X).

COME FUNZIONA

Dopo aver pulito lo schermo, ed avervi fatto apparire

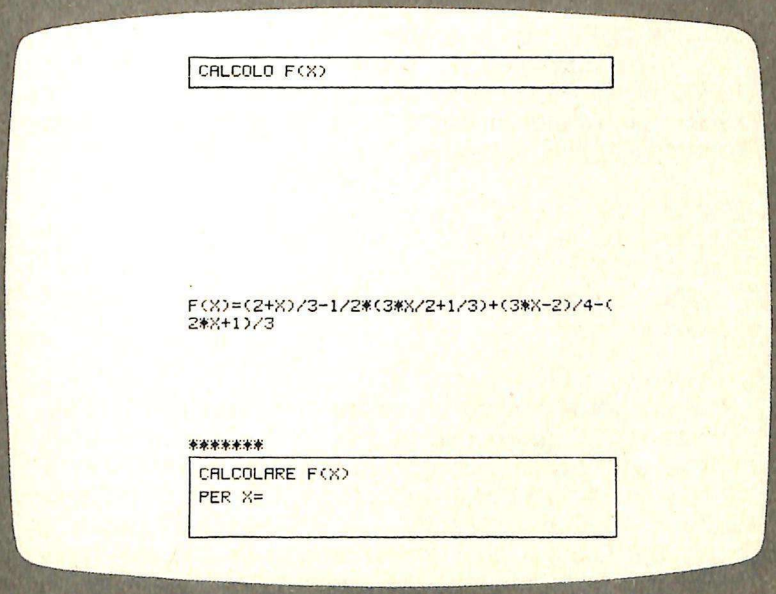


l'equazione F(X) (riga 110 e 500-530), come quasi tutti i programmi precedenti anche questo si presenta come una scelta tra alcuni sottoprogrammi che compiono operazioni diverse, tramite la SUBROUTINE 9 (righe 200-310).

La parte di programma che inizia a riga 1000, CAMBIO F(X), comincia preparando lo schermo con un messaggio (righe 1000-1030), e con l'equazione F(X) da cambiare.

Le righe 1100 e 1110 usano la SUBROUTINE 8 per introdurre la nuova equazione, che viene ribattezzata FX$ nella riga 1200.

Le righe 1210 e 1230 preparano le stringhe necessarie per modificare due righe del programma, la 520 e la 2500, compito svolto dalla SUBROUTINE 10, che rilancia il programma dall'inizio, riga 100.



Per trovare la soluzione dell'equazione F(X)=0, il programma inizia a riga 2000. Il metodo seguito è tipico del calcolo effettuato con calcolatori elettronici, ed avviene in due fasi.

Nella prima fase si tratta di trovare due valori di X, chiamati XA e XB, tali che l'equazione F(X) sia per uno positiva e per l'altro negativa.

Per calcolare F(X) per un certo valore di X, occorre prima dare a X quel valore, poi andare alla subroutine di riga 2500: la variabile FX avrà il valore cercato.

La prima fase comincia dando a X il valore 1 (riga 2100): il valore di F(X) per X=1 sarà il numero X0 (riga 2150).

Se per caso il risultato è proprio zero, la ricerca è finita, si va direttamente a riga 2400. Altrimenti si calcola (riga 2400), un nuovo valore per la variabile.

Questo nuovo valore di X serve per calcolare F(X), il risultato viene chiamto X1 (riga 2200). Anche ora se il risultato è zero si è trovata la soluzione esatta e si va direttamente a riga 2400. Se il nuovo risultato, X1, ed il vecchio, X0, sono dello stesso segno il loro prodotto è positivo (riga 2240), non si è ancora conclusa la prima fase, e si torna a riga 2200 per provare un nuovo valore di X. I valori di X che vengono calcolati nella riga 2200 sono sempre più grandi in valore assoluto, e di segno opposto. Precisamente abbiamo:

1° volta	X =	1
2° volta	X =	-2
3° volta	X =	4
4° volta	X =	-8
5° volta	X =	16

e così via.

Prima o poi si troverà un valore che rende F(X) di segno opposto a quello che ha per X=1. Se non si trova (per esempio per uno sbaglio in F(X)), la riga 2210 conta il numero di tentativi e dopo 100 prove senza trovarlo, poiché X è ora molto grande, ai limiti della capacità di calcolo del C-64, il programma smette di cercare, mandando un messaggio adatto (righe 2450-2460).

Normalmente, dopo un po' di tentativi, la prima fase si conclude felicemente, cioè vengono trovati due valori di X che fanno assumere alla equazione F(X) due valori

opposti: la riga 2300 li chiama XA e XB, ed ora inizia la seconda fase del calcolo. Il valore di X che noi cerchiamo sarà compreso tra XA e XB. La seconda fase consiste nell'avvicinare tra loro i numeri facendo in modo che la radice cercata X sia sempre compresa tra XA e XB.

Per far ciò si usa il metodo detto della "bisezione". La riga 2310 calcola un numero intermedio tra XA e XB, e poi va alla subroutine di riga 2500 per calcolare F(X) con questo valore. Se F(X) ha lo stesso segno che aveva in XA, allora la radice è compresa tra il nuovo numero e XB, viceversa in caso contrario. Il confronto avviene a riga 2330, ed il nuovo sostituisce XA (riga 2330) oppure XB (riga 2350) in modo che la radice sia sempre compresa tra XA e XB. Il programma ritorna a riga 2310 per cercare un nuovo numero tra XA e XB. È evidente che XA e XB si avvicinano sempre di più, ad ogni ciclo

```
CALCOLO F(X)

 PER X=  4.5

F(X)= -1.83333333


F(X)=(2+X)/3-1/2*(3*X/2+1/3)+(3*X-2)/4-(
2*X+1)/3


*******
CALCOLARE F(X)
PER X=
```

di calcolo, contenendo tra loro la radice incognita. La riga 2360 controlla se si è raggiunta la precisione voluta (pari a 1E-7, valore dato a K4 in riga 2100): in questo caso termina la seconda fase del calcolo, ed il programma continua con la riga 2410 che arrotonda il risultato trovato. La seconda fase si può interrompere anche se non si raggiunge la precisione richiesta dopo 100 prove: se ne occupa la riga 2320, che in questo caso fa continuare il programma a riga 2450.

La parte di programma che inizia a riga 3000 — CALCOLO DI F(X) — è molto semplice: la riga 3000 prepara lo schermo e le righe 3010, 3020 usano la SUBROUTINE 7 per chiedere il valore di X; la riga 3100 calcola il risultato e la riga 3110 lo fa apparire sullo schermo, dopo aver cancellato con S$ il vecchio. Se non si introduce il valore X, ma si batte solo /RETURN/ la riga 3030 rimanda a riga 320.

```
 99 REM*********************************
100 PR$=    "EQUAZIONI DI 1# GRADO"
107 REM*********************************
109 :
110 GOSUB 8000:GOSUB 500
113 :
200 PN$(1)="CAMBIO F(X)
            "
210 PN$(2)="SOLUZ. F(X)=0
            "
220 PN$(3)="CALCOLO F(X)
            "
300 PN=3:GOSUB 9400
310 ON PSGOSUB 1000,2000,3000
313 :
320 GOTO 100
323 :
500 RC=13:GOSUB 8100
503 :
520 FX$="(2+X)/3-1/2*(3*X/2+1/3)+(3*X-2
    )/4-(2*X+1)/3"
523 :
```

```
 530 PRINT"F(X)=";FX$:RETURN
 533 :
1000 PR$=PN$(1):GOSUB 8000
1010 PRINT" INTRODUCI LA NUOVA"
1020 PRINT" EQUAZIONE DI PRIMO"
1030 PRINT"        GRADO":GOSUB 500
1033 :
1100 M1$="":M2$="":M3$="F(X)="
1110 I=0:GOSUB 9000
1200 IF RI$="S" THEN FX$=ST$
1203 :
1210 P1$="520FX$="+CHR$(34)+FX$+CHR$(34)
1220 P2$="2500 FX="+FX$
1230 P3$="RUN100":GOTO 9200
1233 :
2000 PR$=PN$(2):GOSUB 8000
2010 PRINT" CALCOLO SOLUZIONE"
2020 PRINT" APPROSSIMATA"
2030 PRINT"[CRSR↑]      [CTRL]+[RVS ON]AT
     TENDERE[CTRL]+[RVS OFF]]":GOSUB 500
2033 :
2100 X=0:K1=1:K2=100:K3=0:K4=1E-7
2150 GOSUB 2500:X0=FX
2160 IF FX=0 THEN XA=X:XB=X:GOTO 2400
2163 :
2200 K1=K1*(-2):X=X+K1
2210 K3=K3+1:IF K3>K2 THEN 2450
2220 GOSUB 2500:X1=FX
2230 IF FX=0 THEN XA=X:XB=X:GOTO 2400
2240 IF X0*X1>0 THEN 2200
2243 :
2300 XB=X:XA=X-K1+K1/2:K3=0
2303 :
2310 X=(XA+XB)/2:GOSUB 2500
2320 K3=K3+1:IF K3>K2 THEN 2450
2330 IF FX*X1<0 THEN XA=X:GOTO 2360
2333 :
2340 IF FX=0 THEN XA=X:XB=X:GOTO 2400
2350 XB=X:X1=FX
```

```
2360 IF ABS(XA-XB)>K4 THEN 2310
2363 :
2400 PRINT:PRINT"F(X)=0 PER X=";
2410 PRINT(INT((XA+XB)/2*1E6+0.5))*1E-6
2413 :
2420 M2$="CONTINUARE":GOSUB 8200
2430 RETURN
2433 :
2450 PRINT"[CRSR↑]NON TROVATO DOPO"
2460 PRINTK2;"PROVE":GOTO 2420
2463 :
2500 FX=(2+X)/3-1/2*(3*X/2+1/3)+(3*X-2)/
     4-(2*X+1)/3
2510 RETURN
2513 :
3000 PR$=PN$(3):GOSUB 8000:GOSUB 500
3003 :
3010 M1$="CALCOLARE F(X)":M2$=""
3020 M3$="PER X= ":I=1:GOSUB 8800
3030 IF RI$="N" THEN RETURN
3033 :
3100 X=VR(1):GOSUB 2500
3110 PRINTS$;"[SHIFT]+[CRSR↑]F(X)= ";FX:
     GOTO 3020
3113 :
3117 REM*****************************
```

METODO DI CRAMER

```
SISTEMA IN X/Y (NORM.)

A1 =  2
B1 = -3
C1 =  18
A2 =  3
B2 = -4


( A1*X+B1*Y = C1
(
( A2*X+B2*Y = C2


*******
COEFFICIENTE
C2 = 25
```

Uffa, questi sistemi! All'inizio sembravano tanto semplici ma poi, con tutti i metodi di risoluzione che il professore è andato spiegando, va a finire che si perde un sacco di tempo soltanto per decidere qual è il metodo migliore per risolvere il sistema proposto dagli esercizi del libro di testo. Certamente l'autore del libro di testo, in combutta con il professore, ha fatto a bella posta, in modo che un esercizio fosse più facile da risolvere con il «confronto», quello do-

po con la «sostituzione» e quello ancora dopo chissà con che cosa... Oltretutto, proprio su questo argomento i «primi della classe» sono veramente antipatici quanto ti vengono a dire: «Ma come, non vedi che addizionando e sottraendo..». Ebbene noi, insieme al nostro computer usiamo il metodo di Cramer, una specie di ruspa, potente e stritolatrice. Il C-64 ci mostrerà tutti i passaggi pervenendo rapidamente alla soluzione esatta. Saremo proprio noi, stavolta, a controllare che tutti i virtuosismi di quelli bravi siano corretti, che veramente la loro soluzione sia quella giusta.

Un sistema di due equazioni in due incognite come questo:

$$2x - 3y = 18$$
$$3x - 4y = 25$$

si può risolvere in molti modi, e Giorgio ne conosce diversi: sostituzione, confronto, somma e sottrazione...
MISTER BIT interviene:

```
A ME PIACE IL METODO DI
CRAMER  CI SONO PIU'

CALCOLI !
```

Giorgio è un po' esterrefatto, ma copia il programma. Lo schermo gli appare come in figura, chiedendogli i coefficienti. Se fa un errore può rispondere «N» alla domanda finale. I coefficienti verranno chiesti di nuovo.
Il calcolo della *x* e della *y* si presentano in modo analogo, con tutti i passaggi, e vengono anche riconosciuti i casi particolari.
Certo che MISTER BIT... è forte!

COME FUNZIONA

Questo programma usa il vettore VR( ) non solo per immettere i dati numerici, ma anche per eseguire dei

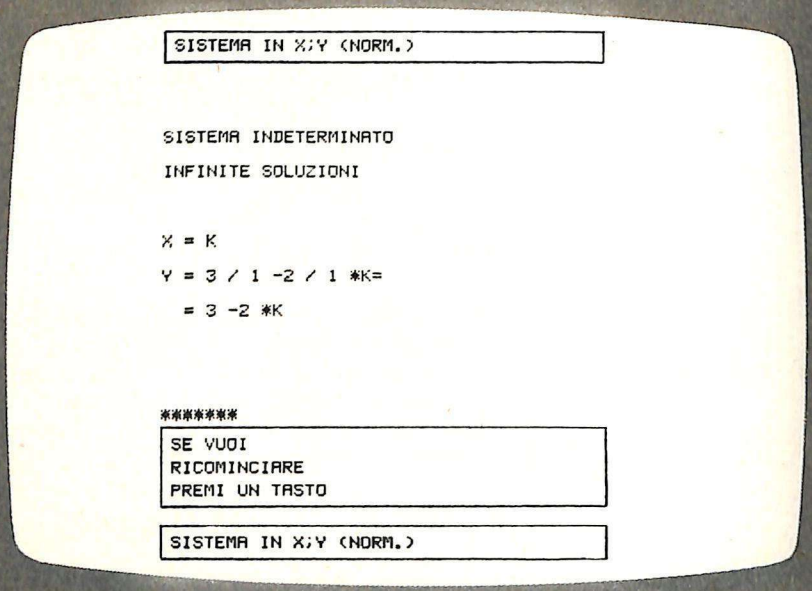
calcoli; inoltre ha bisogno di un vettore di stringhe C$( ); entrambi questi vettori conterranno 15 elementi, quindi la prima istruzione è di tipo DIM, che dimensiona i vettori e che deve essere eseguita una volta sola, all'inizio dell'esecuzione del programma. Dopo la riga 990 il programma continua come abbiamo già visto altre volte: viene pulito e preparato lo schermo (righe 1000-1030). Le righe da 1040 a 1100 usano la SUBROUTINE 7 per chiedere i valori di tutti i coefficienti e termini noti del sistema, che deve essere in forma normale.

Le righe 1110 e 1130 usano la SUBROUTINE 6 per scegliere se continuare o ripetere le domande. Se non si sono fatti errori si continua con le righe 1140-1180 che calcolano tutti i valori necessari per la soluzione del sistema, e dalla riga 1190 si va a riga 2000 se si tratta di un sistema indeterminato o impossibile.

```
METODO DI CRAMER

   ! 18      -3    !
   ! 25      -4    !   (-72)-(-75)
X =+--------------+ = -------------- =
   ! 2       -3    !   (-8)-(-9)
   ! 3       -4    !

 = 3 / 1 = 3

*******
SE VUOI
CALCOLARE Y
PREMI UN TASTO
```

Se il sistema è determinato, il programma ne calcola la soluzione con il metodo Cramer. A riga 1200 si prepara lo schermo, mentre nelle righe 1210, 1220 si trasformano tutti i risultati intermedi in stringhe di sei cifre al massimo, nel vettore C$( ).

Dalla riga 1300 alla riga 1380 il programma fa apparire sullo schermo i calcoli per la X. A riga 1390 va alla subroutine della riga 3000, che prova a semplificare una frazione di cui N è il numeratore e D il denominatore: nelle righe 3100, 3110 il programma controlla se sono entrambi divisibili per un numero intero Z: se lo sono, la frazione è semplificata e si cerca un altro divisore comune. Inoltre nelle righe 3310-3330 si sistemano i segni in modo che un eventuale segno meno appaia solo al numeratore. La riga 3340 fa apparire sullo schermo la frazione ridotta. Il programma continua poi con riga



1400, che fa apparire il valore finale di X.

La riga 1500 introduce una pausa, per mezzo della SUBROUTINE 6, prima di passare al calcolo di Y.

La presentazione del calcolo della Y è del tutto analoga a quella della X, a parte il fatto che si usano altri valori, e avviene nelle righe da 1510 a 1520. Anche ora la riga 1610 usa la subroutine di riga 3000 per provare a semplificare la frazione.

La riga 1630 introduce un'altra PAUSA prima di tornare all'inizio del programma, cioè la riga 1000 (la riga 990 deve essere eseguita una sola volta).

Quando il sistema non è determinato, la riga 2010 controlla se è indeterminato o impossibile. In questo ultimo caso il programma prosegue con il messaggio delle righe 2020 e 2030 per andare alla pausa di riga 1630.

Se invece il sistema è indeterminato, le righe da 2100 a 2150 calcolano e fanno apparire la soluzione.

```
 983 REM*********************************
 985 REM        METODO DI CRAMER
 987 REM*********************************
 989 :
 990 DIM VR(15),C$(15)
1000 PR$="SISTEMA IN X;Y (NORM.)"
1010 GOSUB 8000:RC=14:GOSUB 8100
1020 PRINT" ( A1*X+B1*Y = C1"
1030 PRINT" ( ":PRINT" ( A2*X+B2*Y = C2"
1033 :
1040 M2$="":M1$="COEFFICIENTE":I=0
1050 M3$="A1 = ":GOSUB 8800
1060 M3$="B1 = ":GOSUB 8800
1070 M3$="C1 = ":GOSUB 8800
1080 M3$="A2 = ":GOSUB 8800
1090 M3$="B2 = ":GOSUB 8800
1100 M3$="C2 = ":GOSUB 8800
1103 :
1110 M1$="I COEFFICIENTI":M2$=""
1120 M3$="SONO OK ? (S/N) >":GOSUB 8400
1130 IF RI$="N" THEN 1040
```

```
1133 :
1140 VR(6)=VR(2)*VR(4):VR(7)=VR(5)*VR(1)
1150 VR(8)=VR(0)*VR(5):VR(9)=VR(3)*VR(2)
1160 VR(10)=VR(0)*VR(4):VR(11)=VR(3)*VR(
     1)
1170 VR(12)=VR(6)-VR(7):VR(13)=VR(8)-VR(
     9)
1180 VR(14)=VR(10)-VR(11)
1183 :
1190 IF VR(14)=0 THEN 2000
1200 PR$="METODO DI CRAMER":GOSUB 8000
1203 :
1210 FOR K=0 TO 14
1220 C$(K)=MID$(STR$(VR(K)),1,6):NEXT
1223 :
1300 PRINT"[3 CRSR↑]    !";C$(2); TAB(12)
     ;C$(1); TAB(18);"!"
1310 PRINT"    !";C$(5); TAB(12);C$(4); T
     AB(18);"!";
1320 PRINT TAB(21)" (";C$(6);")-(";C$(7)
     ;")"
1330 PRINT"X =+---------------+ =";
1340 PRINT TAB(22)"--------------- ="
1343 :
1350 PRINT"    !";C$(0); TAB(12);C$(1); T
     AB(18);"!";
1360 PRINT TAB(21)" (";C$(10);")-(";C$(1
     1);")"
1370 PRINT"    !";C$(3); TAB(12);C$(4); T
     AB(18);"!"
1373 :
1380 PRINT"[5 CRSR↑]  =";C$(12);" /";C$(
     14);
1383 :
1390 N=VR(12):D=VR(14):GOSUB 3000
1393 :
1400 PRINT" =";VR(12)/VR(14);
1403 :
1500 M2$="CALCOLARE Y":GOSUB 8200
```

```
1510 GOSUB 8000
1513 :
1520 PRINT"[3 CRSR↑]   !";C$(0); TAB(12)
     ;C$(2); TAB(18);"!"
1530 PRINT"   !";C$(3); TAB(12);C$(5); T
     AB(18);"!";
1540 PRINT TAB(21)" (";C$(8);")-(";C$(9)
     ;")"
1550 PRINT"Y =+---------------+ =";
1560 PRINT TAB(22)"--------------- ="
1563 :
1570 PRINT"   !";C$(0); TAB(12);C$(1); T
     AB(18);"!";
1580 PRINT TAB(21)" (";C$(10);")-(";C$(1
     1);")"
1590 PRINT"   !";C$(3); TAB(12);C$(4); T
     AB(18);"!"
1593 :
1600 PRINT"[5 CRSR↑]  =";C$(13);" /";C$(
     14);
1603 :
1610 N=VR(13):D=VR(14):GOSUB 3000
1613 :
1620 PRINT" =";VR(13)/VR(14);
1623 :
1630 M2$="RICOMINCIARE":GOSUB 8200
1640 GOTO 1000
1643 :
2000 GOSUB 8000
2010 IF VR(0)/VR(3)=VR(2)/VR(5) THEN 210
     0
2013 :
2020 PRINT"[2 CRSR↑]SISTEMA IMPOSSIBILE"
2030 PRINT"[CRSR↑]NESSUNA SOLUZIONE"
2040 GOTO 1630
2043 :
2100 PRINT"[2 CRSR↑]SISTEMA INDETERMINAT
     O"
2110 PRINT"[CRSR↑]INFINITE SOLUZIONI"
```

```
2120 PRINT"[3 CRSR↑]X = K"
2130 PRINT"[CRSR↑]Y =";:N=VR(2):D=VR(1)
2140 GOSUB 3000
2150 N1=N:D1=D
2160 N=-1*VR(0):D=VR(1)
2163 :
2170 GOSUB 3000
2180 PRINTN1;"/";D1;
2190 IF VR(0)*VR(1)<0 THEN PRINT"+";
2200 PRINTN;"/";D;"*K="
2210 PRINT"[CRSR↑]  =";VR(2)/VR(1);
2220 IF VR(0)*VR(1)<0 THEN PRINT"+";
2230 PRINT-1*VR(0)/VR(1);"*K"
2240 GOTO 1630
3000 Z=2:I=0
3003 :
3100 IF INT(N/Z)*Z<>N THEN 3200
3110 IF INT(D/Z)*Z<>D THEN 3200
3120 N=N/Z:D=D/Z:I=I+1:GOTO 3100
3123 :
3200 Z=Z+1:IF Z<N AND Z<D THEN 3100
3300 IF I=0 THEN RETURN
3303 :
3310 IF N*D<0 THEN N=-1*ABS(N):GOTO 3330
3320 N=ABS(N)
3330 D=ABS(D)
3333 :
3340 PRINT"=";N;"/";D;:RETURN
3343 :
3347 REM*****************************
```

SISTEMA X, Y IMPLICITO

```
SISTEMA X;Y IMPLICITO

  1 = CAMBIO F1

  2 = CAMBIO F2

  3 = CALCOLO X;Y


------
F1(X,Y)=(2+X+Y)/3-1/2*(2-X)-(X+Y)

F2(X,Y)=1/2*(X-Y)+1/3*(2*X-3)-3+2*Y


*******
SCEGLI CHE COSA
VUOI FARE >
```

Anche se in Algebra ogni esercizio, ogni calcolo va eseguito in modo — con la completezza dei passaggi intermedi — che il nostro insegnante si renda conto che abbiamo studiato ed imparato, qualche volta non c'è bisogno di sciorinare righe su righe, pagine su pagine dalle lente continue trasformazioni di un sistema di equazioni, fino a pervenire al risultato finale.

Se stiamo «lavorando» in un'altra materia, in fisica

o in geometria, per esempio, il sistema di equazioni è soltanto un mezzo, un qualcosa da utilizzare solo per conoscere il risultato. Se qualcuno ci può risparmiare tutti quei tediosi e lunghi passaggi tanto meglio. È soltanto il risultato quello che ci necessita? Allora meglio averlo esatto e «subito»!

Il nostro C-64 può trovare la soluzione senza fare alcun passaggio algebrico, in modo rapido ed utile. Viva il nostro C-64!

Ricompare MISTER BIT:

MA POSSO FAR DI MEGLIO !

```
CAMBIO F1

INTRODUCI LA NUOVA
EQUAZIONE IN X,Y

------
F1(X,Y)=(2+X+Y)/3-1/2*(2-X)-(X+Y)

F2(X,Y)=1/2*(X-Y)+1/3*(2*X-3)-3+2*Y

*******
F1(X,Y)=
```

```
CON IL METODO DI
CRAMER IL SISTEMA
DEVE ESSERE IN FORMA
NORMALE.

POSSO PROVARE A CALCOLARE
LA SOLUZIONE DI UN SISTEMA
NON RIDOTTO !
```

Giorgio pensa: un bel presuntuoso!

Prova il programma. Durante il calcolo questo si comporta in modo strano, mostrando i vari tentativi che ottiene via via...

Il calcolo avviene infatti in due fasi, MISTER BIT spiega brevemente come funziona. Nella prima fase si cercano quattro coppie di valori *x*, *y* tali che inseriti nelle equazioni F1 e F2 diano le quattro combinazioni possibili di segni: F1 e F2 positive, F1 negativa e F2 positiva, F1 positiva e F2 negativa, entrambe negative. Per trovare queste quattro coppie, si usa un metodo casuale (RANDOM), facendo 300 tentativi con numeri a caso sempre più grandi. Questo metodo non garantisce però che le quattro coppie vengano effettivamente trovate. La prima fase si conclude o con un messaggio di tentativi infruttuosi, ovvero si passa direttamente alla seconda fase, che consiste nel trovare, partendo dalle quattro coppie trovate nella fase precedente, altre coppie di *x* e *y* sempre più vicine alla soluzione, fino al grado di approssimazione chiesto.

Occorre che nelle due equazioni le variabili *x* e *y* non siano al denominatore.

MISTER BIT aggiunge:

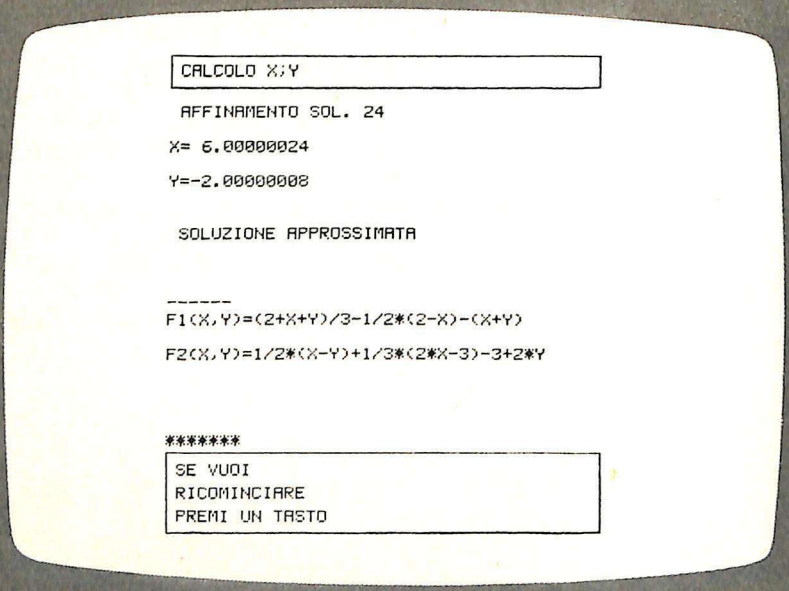
```
A VOLTE 300 PROVE NON BASTANO
ESEGUI DUE O TRE VOLTE LA PRIMA
FASE E RICORDA CHE LE SOLUZIONI
SONO APPROSSIMATE
```

A Giorgio piace questo programma: gli sembra quasi di seguire gli sforzi del calcolatore...

COME FUNZIONA

La riga 1010 pulisce lo schermo e utilizza la subroutine di riga 5000 per far apparire sullo schermo le due equazioni del sistema (righe 5000-5080). Viene poi presentata la scelta tra tre possibilità: si usa, naturalmente, la SUBROUTINE 10 (righe 1020-1060).

Per introdurre le due equazioni si usano le SUBROUTINE 8 — INGRESSO STRINGA — e 10 — MODIFICA PROGRAMMI —, ma in modo un po' diverso di come abbiamo visto in altri programmi. Le righe 2000 e 2010 preparano alcune stringhe sia per la SUBROUTINE 8 che per la SUBROUTINE 10. La riga 2100 pulisce lo schermo e prepara le rimanenti stringhe, mentre le righe 2110, 2120 mandano un messaggio sullo schermo.



Anche ora si fanno apparire le due equazioni usando la subroutine 5000 poi si usa la SUBROUTINE 8 per chiedere la prima equazione.

Nella riga 2150 si completano le stringhe per la SUBROUTINE 10, usata in riga 2160, che modificherà le righe 5000 e 6000. Si è scelta questa soluzione perché il programma che inizia a riga 3000 e che serve per modificare F2(X, Y) (si trova alle righe 5010 e 6010), dopo aver preparato le sue stringhe nelle righe 3000, 3010, 3020, può utilizzare il precedente, saltando a riga 2100.

A riga 4000 inizia il programma di calcolo vero e proprio, al solito con la pulizia dello schermo (SUBROUTINE 1) e con la subroutine locale di riga 5000, che fa apparire le due equazioni del sistema.

A riga 4010 comincia la prima fase, la ricerca RANDOM (a caso) delle prime quattro coppie di valori X e Y. Si utilizza il vettore VR( ) per conservare questi valori, secondo lo schema:

vettore	var.	SEGNO DI F1	SEGNO DI F2
VR (0)	x	+	+
VR (1)	y	+	+
VR (2)	x	−	+
VR (3)	y	−	+
VR (4)	x	+	−
VR (5)	y	+	−
VR (6)	x	−	−
VR (7)	y	−	−

La riga 4020 mette a zero tutti questi valori. Poi vengono calcolati i due valori a caso di X e Y: la riga 4100 garantisce che tali valori siano compresi tra −K1 e +K1, e inizialmente K1 vale 20 (riga 4010).

La riga 4110 va alla subroutine locale di riga 6000: questa calcola i due valori, F1 e F2, assunti dalle due equazioni per i valori X e Y in prova. A seconda del segno di F1 e F2 le righe 4120 e 4130 calcolano un numero H che indica, seguendo lo schema precedente, in quale posto andrebbero i valori X e Y. La riga 4140 con-

trolla se in quella posizione c'è uno zero: se lo trova vuol dire che quella combinazione di segno non è ancora incontrata, e X e Y sono attribuiti al vettore VR( ), al posto H e H+1.

La riga 4160 usa la subroutine locale di riga 4800 per far apparire la scritta con il numero progressivo del tentativo e i due valori X e Y in prova, cancellando i precedenti.

La riga 4200 moltiplica tra loro tutti i valori da VR(0) a VR(8). Poiché VR(8) è uguale a 1 (riga 4110), se il risultato è zero vuol dire che non sono state trovate tutte e quattro le coppie, quindi si aumenta il limite K1 per permettere valori casuali a X e Y un po' più grandi (riga 4220) e se non si sono ancora fatti 300 tentativi si torna alla riga 4100, altrimenti appare un messaggio e si usa la SUBROUTINE 5 per ricominciare (righe 4230-4410).

Se invece il prodotto di riga 4210 è diverso da 0, la prima fase è conclusa felicemente e può cominciare la seconda.

Si calcola una nuova coppia di valori X, Y facendo la media tra due coppie di valori trovati, seguendo un ordine particolare, (righe 4500, 4530) e si va alla subroutine di riga 6000 per ottenere i valori F1 e F2. A seconda del loro segno i valori X e Y sostituiscono i precedenti (righe 4540, 4550, 4560) calcolando il valore di H come abbiamo già visto nelle righe 4110-4130. La riga 4600 usa la subroutine di riga 4800 per scrivere sullo schermo il numero di affinamento e i valori di X e Y.

Per decidere se terminare il ciclo si confrontano i valori trovati con quelli precedenti, conservati in VR(8) e VR(9): se la differenza è maggiore della precisione richiesta (righe 4610, 4620) si prosegue, controllando se si sono raggiunti i 100 tentativi (riga 4630), ed in caso affermativo si esce dal ciclo andando a riga 4300. Altrimenti si aggiornano VR(8) e VR(9) con i valori appena trovati e si torna a riga 4500 per una nuova approssimazione. Quando invece due soluzioni differiscono meno della precisione richiesta il programma termina regolarmente con il messaggio di riga 4700 e va a riga 4400 per presentare la SUBROUTINE 5 — PAUSA — prima di ricominciare.

Questo metodo di calcolo non fornisce una soluzione in tutti i casi: può fallire la ricerca RANDOM (300 tentativi

non sono poi molti), ovvero per alcuni sistemi può non raggiungere una soluzione approssimata.

```
 999 REM*********************************
1000 PR$=    "SISTEMA X;Y IMPLICITO"
1003 REM*********************************
1007 :
1010 GOSUB 8000:GOSUB 5000
1013 :
1020 PN$(1)="CAMBIO F1
            "
1030 PN$(2)="CAMBIO F2
            "
1040 PN$(3)="CALCOLO X;Y
            "
1050 PN=3:GOSUB 9400
1060 ON PSGOTO 2000,3000,4000
1063 :
2000 PR$=PN$(1):P1$="5000 F1$="+CHR$(34)
2010 P2$="6000 F1=":M3$="F1(X,Y)="
2013 :
2100 GOSUB 8000:M1$="":M2$=""
2110 PRINT"[CRSR↑] INTRODUCI LA NUOVA"
2120 PRINT"[CRSR↑] EQUAZIONE IN X,Y"
2123 :
2130 GOSUB 5000:I=0:GOSUB 9000
2140 IF RI$="N" THEN 1000
2143 :
2150 P1$=P1$+ST$+CHR$(34):P2$=P2$+ST$
2160 P3$="GOTO 1000":GOTO 9200
2163 :
3000 P1$="5010 F2$="+CHR$(34)
3010 P2$="6010 F2=":M3$="F2(X,Y)="
3020 PR$=Z$+PN$(2):PR$=LEFT$(PR$,40):GOT
     O 2100
3023 :
4000 PR$=PN$(3):GOSUB 8000:GOSUB 5000
4010 ST$=" RICERCA RANDOM":K1=1:I=0
4020 FOR K=0 TO 7:VR(K)=0:NEXT
```

```
4063 :
4100 X=RND(0)*2*K1-K1:Y=RND(0)*2*K1-K1
4110 GOSUB 6000:H=0:VR(8)=1
4120 IF F1<0 THEN H=2
4130 IF F2<0 THEN H=H+4
4140 IF VR(H)*VR(H+1)<>0 THEN 4160
4150 VR(H)=X:VR(H+1)=Y
4153 :
4160 GOSUB 4800
4200 FOR K=0 TO 7:VR(8)=VR(8)*VR(K):NEXT
4210 IF VR(8)<>0 THEN I=0:GOTO 4480
4213 :
4220 K1=1.06*K1:I=I+1
4230 IF I<300 THEN 4100
4233 :
4300 PRINT"STOP:";I;"TENTATIVI"
4400 M2$="RICOMINCIARE":GOSUB 8200
4403 :
4410 GOTO 1000
4413 :
4480 ST$=" AFFINAMENTO SOL."
4490 VR(8)=VR(0):VR(9)=VR(1):K2=1E-7
4493 :
4500 FOR K=0 TO 3:H=0:IF K<2 THEN H=1
4510 IF K=1 OR K=3 THEN H=H+2
4513 :
4520 X=(VR(2*K)+VR(2*H))/2
4530 Y=(VR(2*K+1)+VR(2*H+1))/2
4540 GOSUB 6000:H=0:IF F1<0 THEN H=2
4550 IF F2<0 THEN H=H+4
4560 VR(H)=X:VR(H+1)=Y
4570 NEXT
4573 :
4580 X=(VR(0)+VR(2)+VR(4)+VR(6))/4
4590 Y=(VR(1)+VR(3)+VR(5)+VR(7))/4
4600 GOSUB 4800
4610 IF ABS(X-VR(8))>K2 THEN 4630
4613 :
4620 IF ABS(Y-VR(9))<K2 THEN 4700
```

```
4623 :
4630 I=I+1:IF I>100 THEN 4300
4633 :
4640 VR(8)=X:VR(9)=Y:GOTO 4500
4700 PRINT" [CRSR↑]SOLUZIONE APPROSSIMAT
     A"
4730 GOTO 4400
4733 :
4800 RC=2:GOSUB 8100
4810 PRINTST$;I:PRINT
4820 PRINTS$;"[SHIFT]+[CRSR↑]X=";X
4830 PRINTS$;"[SHIFT]+[CRSR↑]Y=";Y:RETUR
     N
4833 :
5000 F1$="(2+X+Y)/3-1/2*(2-X)-(X+Y)"
5010 F2$="1/2*(X-Y)+1/3*(2*X-3)-3+2*Y"
5013 :
5040 RC=13:GOSUB 8100
5050 PRINT"-------"
5060 PRINT"F1(X,Y)=";F1$
5063 :
5070 RC=16:GOSUB 8100
5080 PRINT"F2(X,Y)=";F2$:RETURN
5083 :
6000 F1=(2+X+Y)/3-1/2*(2-X)-(X+Y)
6010 F2=1/2*(X-Y)+1/3*(2*X-3)-3+2*Y
6020 RETURN
6023 :
6027 REM*****************************
```

NUMERI COMPLESSI

```
NUMERI COMPLESSI

  1 =       NUOVO X←
  2 = IN MEMORIA M←X
  3 = DA MEMORIA X←M
  4 = SCAMBIO    X Y
  5 = SOMMA      X+Y
  6 = DIFFERENZA X-Y
  7 = PRODOTTO   X*Y
  8 = QUOZIENTE  X/Y

X= 0+J0
Y= 0+J0
Z= 0+J0
-------
M= 0+J0

*******
SCEGLI CHE COSA
VUOI FARE >
```

I numeri complessi sono antipatici. Certo l'insegnante ci ha spiegato che sono molto importanti, che interi campi del sapere umano, come l'elettrotecnica, forse non esisterebbero senza questa parte della matematica... Comunque sono e restano antipatici. Intendiamoci, non è che siano difficili e, tutto sommato, le operazioni da fare sono sempre le stesse, quelle che ormai noi conosciamo bene, ma tutte quelle «i», tutti quei «j» sembra quasi che ci beffino dalle pagine

del nostro quaderno quando, sotto interrogazione, ci siamo accorti di aver fatto un grosso svarione, per giunta «cretino». Non c'è niente da fare: è proprio troppo facile sbagliare quando si debbono fare passaggi e semplificazioni su queste quantità «doppie», subdole. Ci vorrebbe un compagno, una «spalla» che ci metta in guardia e ci avvisi quando l'errore, quatto quatto, si è infiltrato nelle pagine del nostro quaderno di matematica. Il compagno c'è e manco a dirlo è il nostro C-64. I numeri complessi diventeranno semplici e facili come i numeri a cui siamo già abituati e tutti gli irridenti «immaginari» cadranno sotto i tiri serrati del cannone-laser sgranati dal C-64. Il Sommo Matematico che a suo tempo si è inventato i numeri complessi creperebbe di invidia nel vedere quello che riusciamo a fare, e con quale rapidità... A proposito: insieme al nostro C-64, e senza esagerare perché «non si sa mai», non sarebbe forse il caso di scendere in confronto anche con quel nuovo insegnante supplente, tanto giovane ma tanto mostruosamente rapido nei calcoli?

«I numeri complessi sono veramente complessi!». Giorgio non sorride neanche alla battuta infelice di Massimo: ha quasi finito di scrivere il programma.

Ecco: sullo schermo appaiono ben 8 possibilità. Il funzionamento, così gli ha garantito MISTER BIT, è facile come quello di una calcolatrice tascabile. Inserendo un nuovo numero in X, gli altri scorrono in basso, Y va in Z, e Z scompare. M, invece, rimane uguale. «È il numero in memoria» — dice Massimo. Si possono fare le quattro operazioni, ricordando che J è l'unità immaginaria (indicata anche con «i» in molti testi).

«Ma così è semplice! È un gioco con il C-64!».

COME FUNZIONA

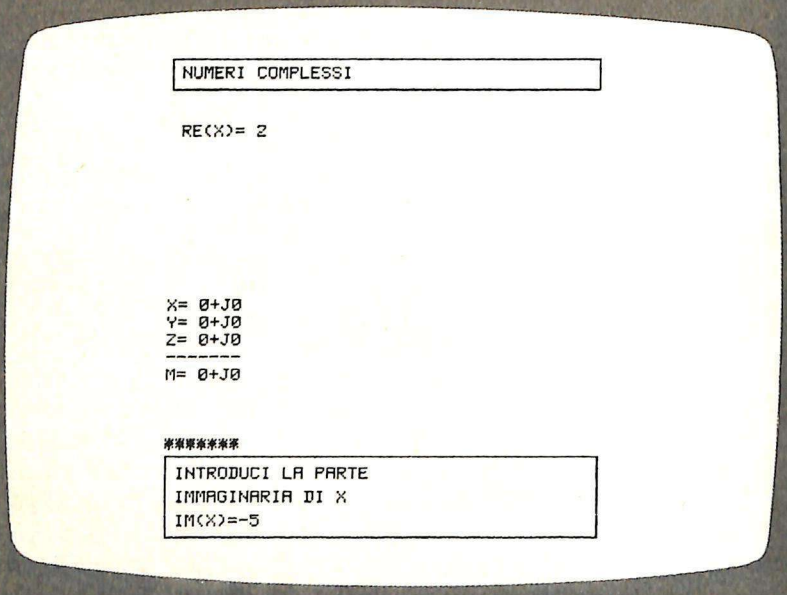
Questo programma utilizza il vettore VR( ) per conservare la parte reale e la parte immaginaria dei numeri

complessi X, Y, Z ed M, secondo lo schema:

VR(0) = Re(X) VR(1) = Im(X)
VR(2) = Re(Y) VR(3) = Im(Y)
VR(4) = Re(Z) VR(5) = Im(Z)
VR(6) = Re(M) VR(7) = Im(M)

Il programma si presenta offrendo la scelta fra 8 attività diverse (righe 1100-1180) e mostrando i valori attuali di X, Y, Z, M.

Di questo si occupa la subroutine locale a riga 1800; dopo aver pulito lo schermo (riga 1800), a riga 1810 dà a RE e IM i valori delle componenti di X, e poi usa a sua volta la subroutine a riga 1900 la quale, dati due numeri, RE e IM, genera una stringa C$ che contiene il numero complesso pronto per essere visualizzato. Nella riga



1820 quindi si stampa sul video il valore di X.

Analogamente con Y (righe 1830, 1840), Z (righe 1850, 1860) e, dopo aver fatto apparire un separatore, con M — (righe 1870, 1880).

Esaminiamo ora le otto parti di programma selezionate dalla SUBROUTINE 9 di riga 1880. Contrariamente agli altri programmi, le 8 attività non sono realizzate con subroutines, infatti in riga 1190 abbiamo /GOTO/ e non /GOSUB/, e manca la riga successiva di rimando all'inizio del programma: le cose non cambiano molto, basta terminare ogni segmento di programma non con un /RETURN/, tipico delle subroutines, ma con un /GOTO/, seguito dal numero di riga voluto.

Cominciamo dal primo pezzo di programma che comincia a riga 1300 (NUOVO X).

Per prima cosa X e Y vengono scalati di un posto, Z si perde ed X è reso uguale a 0+J0: si usa la subroutine 1800 per far apparire questa situazione (righe 1320-1350). Viene poi usata due volte la SUBROUTINE 7, (righe 1360-1370) per la parte reale e la parte immaginaria di X. Poiché I=0, i due numeri vanno al posto giusto nel vettore VR( ).

Viene chiesta conferma dell'esattezza delle cifre introdotte (righe 1360-1370) con la SUBROUTINE 6, e se c'è un errore si torna a riga 1320, per ripresentare le domande, altrimenti si va a riga 1000 (riga 1390), per presentare la nuova situazione e chiedere che cosa fare. La parte di programma che si occupa della seconda attività — IN MEMORIA — si riduce ad una sola riga, la 1400: i valori Re(X) e Im(X) sono attribuiti anche ad M, poi si continua a riga 1000.

L'operazione inversa, dalla memoria ad X, è invece più complessa: occorre anche in questo caso, come abbiamo visto in riga 1300, scalare di un posto X e Y, poi a X possono essere attribuiti i valori di M (riga 1460).

Per scambiare tra loro X e Y si usano due variabili ausiliarie, RE ed IM, effettuando il passaggio in tre tempi, da X a RE e IM, da Y a X, terminando col passaggio da RE e IM a Y (righe 1500-1510). Poi si torna a riga 1000.

La somma di X e Y viene effettuata a riga 1550, con il risultato in RE e IM, poi si usa la subroutine locale di riga

1950 per portare Y in X, poi Z in Y, e rendere Z=0+J0, dopo di che si mette in X il risultato dell'operazione (riga 1560).

La sottrazione di riga 1600 è del tutto analoga: una volta calcolato il risultato, provvisoriamente in RE e IM, si salta a riga 1560 (da riga 1610). A riga 1650 comincia il programma per il prodotto: anche in questo caso si prepara provvisoriamente in RE e IM il risultato; poi si va a riga 1560.

Per la divisione, invece, per prima cosa (riga 1700) si prepara la variabile ausiliaria DE, che è il denominatore che si ottiene dalla razionalizzazione della frazione, e poi è calcolato il valore RE, parte reale del risultato, (riga 1710), e il valore IM, parte immaginaria del risultato (riga 1720), continuando poi al solito a riga 1560.

```
 999 REM*********************************
1000 PR$=      "NUMERI COMPLESSI"
1003 REM*********************************
1007 REM :
1010 GOSUB 1800
1013 :
1100 PN$(1)="      NUOVO X← "
1110 PN$(2)="IN MEMORIA M←X"
1120 PN$(3)="DA MEMORIA X←M"
1130 PN$(4)="SCAMBIO    X Y"
1140 PN$(5)="SOMMA      X+Y"
1150 PN$(6)="DIFFERENZA X-Y"
1160 PN$(7)="PRODOTTO   X*Y"
1170 PN$(8)="QUOZIENTE  X/Y"
1180 PN=8:GOSUB 9400
1190 ON PSGOTO 1300,1400,1450,1500,1550,
     1600,1650,1700
1193 :
1300 FOR K=0 TO 3:VR(5-K)=VR(3-K):NEXT
1310 VR(0)=0:VR(1)=0:GOSUB 1800
1313 :
1320 M1$="INTRODUCI LA PARTE":I=0
1330 M2$="REALE DI X":M3$="RE(X)="
1340 GOSUB 8800:M2$="IMMAGINARIA DI X"
```

```
1350 M3$="IM(X)=":GOSUB 8800
1353 :
1360 M1$="IL NUMERO X":M2$=""
1370 M3$="E' OK ? (S/N) >":GOSUB 8400
1380 IF RI$="N" THEN 1320
1383 :
1390 GOTO 1000
1393 :
1400 VR(6)=VR(0):VR(7)=VR(1):GOTO 1000
1403 :
1450 FOR K=0 TO 3:VR(5-K)=VR(3-K):NEXT
1460 VR(0)=VR(6):VR(1)=VR(7):GOTO 1000
1463 :
1500 RE=VR(0):IM=VR(1):VR(0)=VR(2)
1510 VR(1)=VR(3):VR(2)=RE:VR(3)=IM
1520 GOTO 1000
1523 :
1550 RE=VR(0)+VR(2):IM=VR(1)+VR(3)
1560 GOSUB 1950:VR(0)=RE:VR(1)=IM
1570 GOTO 1000
1573 :
1600 RE=VR(0)-VR(2):IM=VR(1)-VR(3)
1610 GOTO 1560
1613 :
1650 RE=VR(0)*VR(2)-VR(1)*VR(3)
1660 IM=VR(1)*VR(2)+VR(0)*VR(3)
1670 GOTO 1560
1673 :
1700 DE=VR(2)*VR(2)+VR(3)*VR(3)
1710 RE=(VR(0)*VR(2)+VR(1)*VR(3))/DE
1720 IM=(VR(1)*VR(2)-VR(0)*VR(3))/DE
1730 GOTO 1560
1733 :
1800 GOSUB 8000:RC=13:GOSUB 8100
1810 IM=VR(1):RE=VR(0):GOSUB 1900
1820 PRINT"X=";C$
1830 IM=VR(3):RE=VR(2):GOSUB 1900
1840 PRINT"Y=";C$
1850 IM=VR(5):RE=VR(4):GOSUB 1900
```

```
1860 PRINT"Z=";C$:PRINT"--------"
1870 IM=VR(7):RE=VR(6):GOSUB 1900
1880 PRINT"M=";C$:RETURN
1883 :
1900 S1$="+J":IF IM<0 THEN S1$="-J"
1910 C$=MID$(STR$(RE),1,7)+S1$+MID$(STR$
     (IM),2,7)
1920 RETURN
1923 :
1950 FOR K=0 TO 3:VR(K)=VR(K+2):NEXT
1960 VR(4)=0:VR(5)=0:RETURN
1963 :
1967 REM*********************************
```

EQUAZIONI DI II° GRADO

```
SOLUZ. EQUAZ. II GRADO

A=  1
B= -5
C=  6


F(X)= A*X↑2+B*X+C

*******
I COEFFICIENTI
SONO OK (S/N) ?
```

Certo che le equazioni di secondo grado sono belle: dall'alto di quell'esponente quadrato si incomincia a respirare un'aria più piacevolmente matematica. Ci sembra quasi di comandare a quelle due radici: voi siete reali, voi coincidenti, voi complesse... Eppure un piccolissimo dubbio viene ad offuscare tanta piacevolezza. Tutti gli esercizi del libro di testo di

Matematica danno come risultato sempre dei valori esatti, comodi, tutti i discriminanti si sciolgono come neve al sole di primavera perché sono sempre costituiti da facili quadrati perfetti che danno dei bei valori rotondi, precisi. Noi che siamo bravi in matematica abbiamo però provato ad applicare le equazioni di secondo grado nelle altre materie che studiamo: che disastro! Ogni volta sono saltati fuori dei pestiferi calcoli sotto radice con dei risultati proprio bruttarelli, zeppi di cifre e... dobbiamo ringraziare la nostra fedele calcolatricetta tascabile, perché, altrimenti, non saremmo riusciti neanche ad andare oltre la soglia del radicale. Questa volta il C-64 ci guiderà, mano nella mano, costruendo le varie soluzioni, «ragionando».

A questo punto siamo già esperti di algebra con il computer per cui possiamo toglierci il lusso di essere anche eleganti e, perché no, un pò saccenti.

«Ma secondo te ragiona?» chiede Massimo a Giorgio.

«Non credo, a volte sembra di sì, ma poi...».

Sullo schermo acceso appare MISTER BIT.

```
PUO' EFFETTUARE DELLE SCELTE
PURCHE' PREVISTE DAL PROGRAMMA
CHE DEVE CONSIDERARE TUTTI I
CASI  ALTRIMENTI NON E' UN BUON
PROGRAMMA.
```

«Per esempio, nella soluzione di un'equazione di secondo grado, si possono presentare diversi casi», dice Giorgio, «e i calcoli da eseguire sono diversi».

In effetti appena introdotti i tre coefficienti, il programma li esamina, e, se è il caso, calcola il discriminante, dal quale dipende il tipo delle radici.

Le operazioni da fare sono diverse, ma è il programma che prevede e controlla i vari casi, ottenendo risultati diversi e deve essere tale che un caso non previsto non dia risultati sbagliati, ovvero lo blocchi.

COME FUNZIONA

All'inizio questo programma chiama la subroutine locale — inizia a riga 1900 — che pulisce lo schermo, fa apparire la forma normale dell'equazione di secondo grado (righe 1900 e 1910) e riposiziona il cursore a riga 2 dello schermo.

Dopo questo vengono chiesti i tre coefficienti (righe 1100, 1110, 1120) e, se c'è stato un errore, si ripropongono le domande (righe 1130-1150). Se A è uguale a 0 c'è un errore, mentre se B oppure C sono uguali a zero si tratta di casi particolari (righe 1160-1180). Nel caso generale il programma continua, presentando il calcolo del discriminante (righe 1200-1260) ed utilizzando poi la SUBROUTINE 5 per fare una pausa (riga 1270). A seconda del valore del discriminante (la variabile V1) il

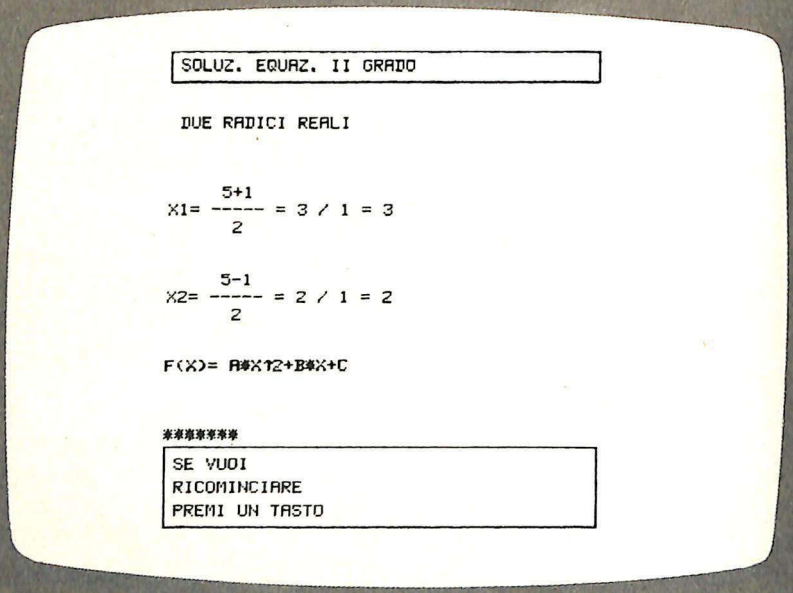
```
SOLUZ. EQUAZ. II GRADO

 A=  1
 B= -5
 C=  6

DISCRIMINANTE = B↑2-4A*C =

= 25-4* 1 * 6 =

= 1

            QUINDI ...

F(X)= A*X↑2+B*X+C


*******
SE VUOI
CONTINUARE
PREMI UN TASTO
```

programma prosegue in modo differente.

Se il discriminante è uguale a zero il programma prosegue (righe 1320-1370) presentando il caso di due radici coincidenti. A riga 1380 è di nuovo utilizzata la SUBROUTINE 5 per effettuare una pausa prima di tornare a riga 1000.

Se il discriminante è maggiore di zero si continua a riga 1400 che fa apparire una frase sullo schermo, poi (righe 1410-1440) sono calcolate le due soluzioni e trasformate in stringhe, X1$ e X2$, e si preparano altre due stringhe S1$ e S2$, prima di continuare a riga 1700. Si fa così per usare lo stesso pezzo di programma (righe 1700-1840) per presentare i risultati, che siano reali o complessi coniugati.

Infatti, nel caso di discriminante negativo, il programma continua con riga 1500, che manda due frasi sullo



schermo, poi viene calcolata sia la parte reale (X1, riga 1510) sia quella immaginaria (X2, riga 1520) della soluzione, preparandole in forma di due stringhe (come nel caso di soluzioni reali) nelle righe 1530, 1540.

Ora i due casi si fondono in uno solo (le differenze sono nel contenuto delle stringhe), preparando tre stringhe (righe 1700, 1710, 1720). Le due righe che seguono calcolano le lunghezze dei numeri che dovranno essere scritti (variabili V3, V4) per poter fare (a riga 1770) un segno di frazione della lunghezza giusta, e per centrare (riga 1790) il denominatore.

Se il discriminante è maggiore di 0, si preparano alcuni valori (riga 1782) per andare alla subroutine locale di riga 3000: questa subroutine semplifica la frazione (se è possibile, naturalmente) e la riga 1786 stampa l'eventuale frazione semplificata.

```
SOLUZ. EQUAZ. II GRADO

 DUE RADICI COMPLESSE
   (J=UNITA' IMMAG.)

     -6+J4
X1= ------ =-3+J2
       2

     -6-J4
X2= ------ =-3-J2
       2

F(X)= A*X↑2+B*X+C

*******
SE VUOI
RICOMINCIARE
PREMI UN TASTO
```

Terminata la presentazione dei risultati (a riga 1850) il programma torna alla domanda di riga 1380.

A riga 2000 comincia la parte del programma che presenta la soluzione nel caso di equazioni pure (B=0). Le due soluzioni sono opposte e sono calcolate a riga 2060, e trasformate in stringa (riga 2070). Se necessario è aggiunta l'unità immaginaria (riga 2080) e i due risultati sono presentati dalle righe 2090 e 2100. Si continua al solito a riga 1380.

Nel caso di equazione spuria occorre solo calcolare una soluzione (l'altra è uguale a 0) e questo avviene alle righe 2230-2250, poi si va alla domanda di riga 1380.

```
 999 REM*********************************
1000 PR$=  "SOLUZ. EQUAZ. II GRADO"
1003 REM*********************************
1005 :
1010 GOSUB 1900:M1$="INTRODUCI IL"
1013 :
1100 M2$="COEFFICIENTE":M3$="A= "
1110 GOSUB 8800:M3$="B= ":GOSUB 8800
1120 M3$="C= ":GOSUB 8800
1123 :
1130 M1$="I COEFFICIENTI":M2$=""
1140 M3$="SONO OK (S/N) ?":GOSUB 8400
1150 IF RI$="N" THEN 1010
1153 :
1160 IF VR(0)=0 THEN M2$="ERRORE IN A":G
     OTO 1140
1163 :
1170 IF VR(1)=0 THEN 2000
1180 IF VR(2)=0 THEN 2200
1183 :
1200 V1=VR(1)*VR(1):RC=8:GOSUB 8100
1210 PRINT"DISCRIMINANTE = ";
1220 PRINT"B↑2-4A*C ="
1230 PRINT"[CRSR↑]= ";MID$(STR$(V1),2,6)
     ;
1234 PRINT"-4*";
```

```
1240 PRINTVR(0);"*";VR(2);"="
1250 V1=V1-4*VR(0)*VR(2)
1254 PRINT"[CRSR↑]=";V1
1260 PRINT"[2 CRSR↑]              QUINDI ..
     ."
1263 :
1270 M2$="CONTINUARE":GOSUB 8200
1300 GOSUB 1900:IF V1>0 THEN 1400
1310 IF V1<0 THEN 1500
1313 :
1320 PRINT"[CRSR↑]DUE RADICI COINCIDENTI
     "
1330 PRINT"[CRSR↑]          -B"
1340 PRINT"X1=X2= ----- ="
1350 PRINT"          2*A":PRINT"[2 CRSR↑]=
     ";
1354 N=-1*VR(1):D=2*VR(0)
1360 PRINTN;"/";D;"=";
1364 GOSUB 3000:IF I=0 THEN PRINT:GOTO 1
     370
1368 PRINTN;"/";D;"="
1370 PRINT"[CRSR↑]=";N/D
1373 :
1380 M2$="RICOMINCIARE":GOSUB 8200
1390 GOTO 1000
1393 :
1400 PRINT"[CRSR↑] DUE RADICI REALI":PRI
     NT
1410 X1=(SQR(V1)-VR(1))/(2*VR(0))
1420 X1$=STR$(X1):S1$="+"
1430 X1=(SQR(V1)+VR(1))/((-2)*VR(0))
1440 X2$=STR$(X1):S2$="-":GOTO 1700
1443 :
1453 :
1500 PRINT" DUE RADICI COMPLESSE"
1506 PRINT"    (J=UNITA' IMMAG.)"
1510 V1=-1*V1:X1=-1*VR(1)/(2*VR(0))
1520 X2=SQR(V1)/(2*VR(0)):S1$="+J"
1530 S2$="-J":X1$=MID$(STR$(X1),1,6)+S1$
```

```
     +MID$(STR$(X2),2,7)
1540 X2$=MID$(STR$(X1),1,6)+S2$+MID$(STR
     $(X2),2,7)
1543 :
1700 A$=MID$(STR$(2*VR(0)),1,6)
1710 D$=MID$(STR$(SQR(V1)),2,7)
1720 B$=MID$(STR$(VR(1)*(-1)),1,6)
1730 V3=LEN(B$+D$+S1$):V4=LEN(A$)
1740 IF V4>V3 THEN V3=V4
1743 :
1750 PRINT"[2 CRSR↑]     ";B$;S1$;D$
1760 PRINT"X1= ";
1770 FOR K=0 TO V3:PRINT"-";:NEXT
1780 PRINT" =";
1781 IF LEN(S1$)>1 THEN 1788
1782 N=SQR(V1)-VR(1):D=2*VR(0):GOSUB 300
     0
1784 IF I=0 THEN 1788
1786 PRINTN;"/";D;"=";
1788 PRINTX1$
1790 PRINTSPC(4+V3/2-V4/2);A$
1793 :
1800 PRINT"[2 CRSR↑]     ";B$;S2$;D$
1810 PRINT"X2= ";
1820 FOR K=0 TO V3:PRINT"-";:NEXT
1830 PRINT" =";:
1840 IF LEN(S1$)>1 THEN 1848
1842 N=SQR(V1)+VR(1):D=-2*VR(0):GOSUB 30
     00
1844 IF I=0 THEN 1848
1846 PRINTN;"/";D;"=";
1848 PRINTX2$:PRINTSPC(4+V3/2-V4/2);A$
1850 GOTO 1380
1853 :
1900 GOSUB 8000:RC=17:GOSUB 8100
1910 PRINT"F(X)= A*X↑2+B*X+C":I=0
1920 RC=2:GOSUB 8100:RETURN
1933 :
2000 GOSUB 1900
```

```
2010 PRINT" EQUAZIONE PURA (B=0)"
2015 :
2020 RC=5:GOSUB 8100
2030 PRINT"     +      - C"
2040 PRINT"X = - SQR(---) ="
2043 :
2050 PRINT"              A"
2060 X1=SQR(ABS(VR(2)/VR(0)))
2070 X1$=MID$(STR$(X1),1,7)
2080 IF VR(2)/VR(0)>0 THEN X1$=" J"+MID$
     (X1$,2)
2090 PRINT"[2 CRSR↑]X1= +";X1$
2100 PRINT"[2 CRSR↑]X2= -";X1$:GOTO 1380
2113 :
2200 GOSUB 1900
2210 PRINT" EQUAZIONE SPURIA (C=0)"
2213 :
2215 RC=6:GOSUB 8100
2220 PRINT"X1= 0":VR(1)=-1*VR(1)
2230 PRINT"[4 CRSR↑]X2= - B/A =";
2240 PRINTVR(1);"/";VR(0);"="
2250 PRINT"[CRSR↑]  =";:N=VR(1):D=VR(0)
2252 GOSUB 3000:IF I=0 THEN 2256
2253 :
2254 PRINTN;"/";D;"=";
2256 PRINTN/D:GOTO 1380
3000 Z=2:I=0
3003 :
3100 IF INT(N/Z)*Z<>N THEN 3200
3110 IF  INT(D/Z)*Z<>D THEN 3200
3120 N=N/Z:D=D/Z:I=I+1:GOTO 3100
3200 Z=Z+1:IF Z<=N  AND Z<=D THEN 3100
3303 :
3310 IF N*D<0 THEN N=-1*ABS(N):GOTO 3330
3320 N=ABS(N)
3330 D=ABS(D):RETURN
3999 REM*********************************
```

PROGRESSIONI ARITMETICHE

```
PROGRESSIONI ARIT.

D = .5
N = 21

-------
 D =RAGIONE
 N =POSIZIONE
 A1=PRIMO TERMINE
 AN=N-ESIMO TERMINE
 SN=SOMMA

 BATTERE RETURN PER I
 VALORI DA CALCOLARE

*******
 INTRODUCI TRE
 VALORI NOTI
 A1=357
```

Quasi quasi ci dispiace di essere giunti al termine di questo libro che ci ha aperto una nuova porta, moderna e potente.

Entrare nel difficile campo dell'algebra con affianco il nostro computer è stato interessante e ci ha circondato, agli occhi dei nostri compagni ed anche del nostro professore, di un alone di «esperti», di persone che sanno, di persone a cui non solo la promozione è assicurata ma anche di persone a cui guardare

con simpatia e rispetto. Bene, bene. C'è un ultimo scoglio, ora che siamo proprio vicino all'approdo sospirato delle vacanze estive: le progressioni. Ormai non ci fanno più paura formule e formulette anche se queste progressioni aritmetiche ne hanno proprio tante. E allora, ancora una volta avanti con il nostro C-64 che, con questo programma, sceglierà per noi le formule da applicare ed eseguirà per noi i calcoli con una esattezza che solo un computer può dare, una precisione tanto più importante ora che si sta avvicinando il grande caldo e la nostra mente non riesce più tanto facilmente a concentrarsi sui calcoli matematici.

«Il programma si accorge di quali numeri hai inserito e calcola i rimanenti» spiega Giorgio. Massimo annuisce: «Inserisco solo i valori noti, per gli altri batto solo il tasto /RETURN/, e devo inserire almeno tre valori su cinque: i due rimanenti sono calcolati automaticamente».

In una progressione aritmetica interessano, in genere, il primo termine (A1), la ragione, ovvero la quantità da sommare ad ogni termine per ottenere il successivo (D), il termine di posizione N, indicato con (AN), la somma dei primi N termini, cioè dei termini da 1 a N, chiamata (SN).

MISTER BIT appare sullo schermo, un po' ironico:

```
E' FORTE IL C-64
RAGAZZI !
```

COME FUNZIONA

All'inizio questo programma, dopo aver pulito il video con la SUBROUTINE 1, fa apparire, tramite la subroutine 3000, il significato delle abbreviazioni usate (riga 1010) e scrive una avvertenza (righe 1020, 1030), e prosegue chiedendo di introdurre tre valori numerici su cinque (righe 1100-1150). Per chiedere i dati numerici non usa direttamente la SUBROUTINE 7, ma usa una

subroutine locale a 3500, che si comporta come la SUBROUTINE 7 (infatti la usa a riga 3500), ma in più, se si preme solo /RETURN/, conserva in K1 e K2 il valore che ha la variabile I. Al termine delle cinque domande K1 e K2 ci indicano quali valori sono da calcolare, e possono servire per scegliere come proseguire (righe 1200, 1210). La situazione completa è data dallo specchietto seguente:

SI DEVONO CALCOLARE	K1	K2	TOTALE K1+3 K2	RIGA PROGRAMMA
D N	*1*	*0*	*1*	*4050*
D A1	*2*	*0*	*2*	*4060*
D AN	*3*	*0*	*3*	*4070*
D SN	*4*	*0*	*4*	*4000*
N A1	*2*	*1*	*5*	*4200*
N AN	*3*	*1*	*6*	*4100*
N SN	*4*	*1*	*7*	*4010*
A1 AN	*3*	*1*	*9*	*4300*
A1 SN	*4*	*2*	*10*	*4020*
AN SN	*4*	*3*	*13*	*4030*

I dieci segmenti di programma che si occupano dei calcoli sono strutturati come subroutines locali. Effettuati i calcoli il programma prosegue presentando i risultati (righe 2000-2140) e usa a riga 2200 la SUBROUTINE 5 per effettuare una pausa prima di chiedere nuovi valori.
Alcuni calcoli, più semplici, sono raggruppati ciclicamente nelle righe 4000-4080: a seconda del punto di ingresso si calcolano prima i valori cercati, e poi, eventualmente, si ricalcolano valori già noti.

Gli altri sono un po' più complessi e vengono effettuati a partire dalle righe 4100, 4200, 4300. Nelle righe 4100-4160 si risolve un'equazione di secondo grado, che può portare a segnalazione di errore (riga 4400) se il discriminante è minore di zero (riga 4120), ovvero se entrambe le soluzioni sono negative (righe 4140, 4160).

Le formule utilizzate sono le seguenti:

4000 $$D = \frac{AN - A1}{N - 1}$$

4010 $$N = \frac{AN - A1}{D} + 1$$

4020 $$A1 = AN - D\,(N - 1)$$

4030 $$AN = A1 + (N - 1)\,D$$

4040 $$SN = \frac{A1 + AN}{2} \cdot N$$

4050 $$N = \frac{2\,SN}{AN + A1}$$

4060 $$A1 = \frac{2\,SN}{N} - AN$$

4070 $$AN = \frac{2\,SN}{N} - AN$$

4080 $$D = \frac{AN - A1}{N - 1}$$

4100 $$\frac{D - 2A1 \pm \sqrt{(D - 2A1)^2 + 8\,D \cdot SN}}{2\,D}$$

4200 $$\frac{D + 2AN \pm \sqrt{(D + 2AN)^2 - 8\,D \cdot SN}}{2\,D}$$

4300 $$AN = \frac{SN}{N} + \frac{D \cdot (N - 1)}{2}$$

```
 997 REM*********************************
1000 PR$=     "PROGRESSIONI ARIT."
1003 REM*********************************
1007 :
1010 GOSUB 8000:GOSUB 3000:K1=0:K2=0
1020 PRINT"[CRSR↑] BATTERE [CTRL]+[RVS O
     N]RETURN[CTRL]+[RVS OFF] PER I"
1030 PRINT" VALORI DA CALCOLARE"
```

```
1033 :
1100 M1$="INTRODUCI TRE"
1110 M2$="VALORI NOTI":I=0
1120 M3$="D =":GOSUB 3500
1130 M3$="N =":GOSUB 3500
1140 M3$="A1=":GOSUB 3500
1150 M3$="AN=":GOSUB 3500
1160 M3$="SN=":GOSUB 3500
1200 K1=K1+3*K2
1210 ON K1GOSUB 4050,4060,4070,4000,4200
     ,4100,4010,4400,4300,4020,4400,4400
     ,4030
1213 :
2000 GOSUB 8000:GOSUB 3000
2010 RC=3:GOSUB 8100
2100 PRINT" D =";VR(0)
2110 PRINT" N =";VR(1)
2120 PRINT" A1=";VR(2)
2130 PRINT" AN=";VR(3)
2140 PRINT" SN=";VR(4)
2143 :
2200 M2$="CALCOLARE ANCORA":GOSUB 8200
2203 :
2210 GOTO 1000
2213 :
3000 RC=11:GOSUB 8100
3010 PRINT"--------"
3020 PRINT" D =RAGIONE"
3030 PRINT" N =POSIZIONE"
3040 PRINT" A1=PRIMO TERMINE"
3050 PRINT" AN=N-ESIMO TERMINE"
3060 PRINT" SN=SOMMA":RETURN
3063 :
3500 GOSUB 8800
3510 IF RI$="N" THEN K2=K1:K1=I-1
3520 RETURN
3523 :
4000 VR(0)=(VR(3)-VR(2))/(VR(1)-1)
4010 VR(1)=(VR(3)-VR(2))/VR(0)+1
```

```
4020 VR(2)=VR(3)-(VR(1)-1)*VR(0)
4030 VR(3)=VR(2)+(VR(1)-1)*VR(0)
4040 VR(4)=(VR(2)+VR(3))*VR(1)/2
4050 VR(1)=2*VR(4)/(VR(3)+VR(2))
4060 VR(2)=2*VR(4)/VR(1)-VR(3)
4070 VR(3)=2*VR(4)/VR(1)-VR(2)
4080 VR(0)=(VR(3)-VR(2))/(VR(1)-1)
4090 RETURN
4093 :
4100 VR(5)=(VR(0)-2*VR(2))*(VR(0)-2*VR(2
     ))
4110 VR(5)=VR(5)+8*VR(0)*VR(4)
4120 IF VR(5)<0 THEN 4400
4130 VR(1)=(VR(0)-2*VR(2)-SQR(VR(5)))/(2
     *VR(0))
4140 IF VR(1)>0 THEN 4070
4150 VR(1)=(VR(0)-2*VR(2)+SQR(VR(5)))/(2
     *VR(0))
4160 IF VR(1)>0 THEN 4070
4170 GOTO 4400
4173 :
4200 VR(5)=(VR(0)+2*VR(3))*(VR(0)+2*VR(3
     ))
4210 VR(5)=VR(5)-8*VR(0)*VR(4)
4220 IF VR(5)<0 THEN 4400
4230 VR(1)=(VR(0)+2*VR(3)-SQR(VR(5)))/(2
     *VR(0))
4240 IF VR(1)>0 THEN 4060
4250 VR(1)=(VR(0)+2*VR(3)+SQR(VR(5)))/(2
     *VR(0))
4260 IF VR(1)>0 THEN 4060
4270 GOTO 4400
4300 VR(3)=VR(4)/VR(1)+VR(0)*(VR(1)-1)/2
4310 GOTO 4020
4313 :
4400 RC=10:GOSUB 8100
4410 PRINT"ERRORE"
4420 GOTO 2200
4423 :
4427 REM*****************************
```

APPENDICE

SUBROUTINES

Abbiamo riunito qui le subroutines usate da tutti i programmi di questo libro. Come abbiamo detto questo metodo evita di doverle ribattere in ogni programma e permette di ottenere programmi più semplici e chiari.

Ricordiamo che una subroutine è un pezzo di programma che termina con /RETURN/ ed è utilizzato dal programma principale con una istruzione tipo /GOSUB riga/: questa istruzione (GO=vai, SUB=subroutine) fa continuare l'elaborazione alla subroutine che RITORNA, una volta terminata, al programma principale, precisamente alla istruzione successiva al /GOSUB/.

Per ogni subroutine è indicato con esattezza che cosa fa e quali variabili usa: questo è prezioso se vorrete usarle in programmi creati da voi. L'uso delle subroutines infatti semplifica molto la programmazione ma occorre usarle con attenzione, preparando prima tutte le variabili necessarie, ed utilizzando esattamente le variabili modificate.

Come prima cosa armatevi di pazienza e copiate tutte e dieci le subroutines. Ricordate che non si devono copiare le righe dispari.

Copiate (e controllate) tutte le subroutines, salvandole su nastro — meglio se all'inizio di un nastro nuovo — con il comando /SAVE "SUBROUTINES"/.

Quando vorrete scrivere un programma che le usa, per prima cosa caricatele dal nastro, con il comando /LOAD "SUBROUTINES"/, e dopo copiate il programma. Salvate su nastro il tutto con il nome del programma che avrete scritto.

TABELLA DELLE SUBROUTINES

1 - INIZIALIZZAZIONE SCHERMO	/GOSUB 8000/
2 - POSIZIONAMENTO CURSORE	/GOSUB 8100/
3 - PRESENTAZIONE MESSAGGI	/GOSUB 8600/
4 - ATTESA DI TASTO PREMUTO	/GOSUB 8300/
5 - PAUSA	/GOSUB 8200/
6 - DOMANDA SÌ/NO	/GOSUB 8400/
7 - INGRESSO DATI NUMERICI	/GOSUB 8800/
8 - INGRESSO STRINGA	/GOSUB 9000/
9 - SCELTA PROGRAMMI	/GOSUB 9400/
10 - MODIFICA PROGRAMMI	/GOSUB 9200/

SUBROUTINE 1

INIZIALIZZAZIONE SCHERMO

Molte volte occorrre pulire lo schermo, per esempio all'inizio di un programma, oppure dopo aver presentato alcuni risultati e prima di presentarne altri. Per far ciò si può usare la SUBROUTINE 1, che pulisce tutto lo schermo e prepara sette variabili (C0, C1, C2, C3, C4, C5, C6) che definiscono i colori delle lettere e dello sfondo nelle varie zone del video. Infatti per tutti i programmi di questo libro si usa il modo colore esteso, e lo schermo è diviso in quattro zone: la prima riga che contiene il titolo, la parte centrale (zona «testo»), le ultime tre righe riservate ai messaggi e domande (zona «domande») e il bordo.

I colori scelti sono: giallo per bordo e sfondo testo, lettere nere, bianco per il titolo e viola per la zona domande: se però volete provare altri colori, dopo le subroutines troverete un programma che fa per voi!

La SUBROUTINE 1 fa anche altre cose: presenta sulla prima riga il titolo del programma (deve essere nella stringa PR$) e prepara una stringa di 22 spazi (S$) che serve per cancellare una riga sullo schermo. Inoltre prepara due valori (MC e MV) relativi allo schermo.

Il cursore è posizionato all'inizio della riga 3, pronto a scrivere nella zona testo.

```
7903 REM**********SUBROUTINE #1*********
7905 REM     INIZIALIZZAZIONE SCHERMO
7907 REM*****************************
7909 REM
7911 REM SCOPO:
7913 REM     QUESTA SUBROUTINE PULISCE LO
7915 REM      SCHERMO, PRESENTA IL TITOLO
7917 REM    DEL PROGRAMMA, PREPARA ALCUNE
7919 REM          VARIABILI USATE DA ALTRE
7921 REM                        SOBROUTINES.
7923 REM
7925 REM PRIMA DEL [GOSUB 8000]:
7927 REM  PR$ > DEVE CONTENERE IL TITOLO
7929 REM
7931 REM DOPO IL [GOSUB 8000]:
7933 REM  C0  <  DEFINISCE IL COLORE DEL
7935 REM                              BORDO
7937 REM  C1  <  DEFINISCE IL COLORE DEL
7939 REM                       FONDO TITOLO
7941 REM  C2  <  DEFINISCE IL COLORE DEL
7943 REM                    CARATTERE TITOLO
7945 REM  C3  <       DEFINISCE IL COLORE
7947 REM                    DEL FONDO TESTO
7949 REM  C4  <  DEFINISCE IL COLORE DEL
7951 REM                    CARATTERE TESTO
7953 REM  C5  <  DEFINISCE IL COLORE DEL
7955 REM                      FONDO DOMANDE
7957 REM  C6  <  DEFINISCE IL COLORE DEL
7959 REM                  CARATTERE DOMANDE
7961 REM  S$  <E'UNA STRINGA DI 40 SPAZI
7963 REM  MV  <      CONTIENE L'INDIRIZZO
7965 REM                DELLA MEMORIA VIDEO
7967 REM  MC  <      CONTIENE L'INDIRIZZO
7969 REM             DELLA MEMORIA COLORE
7971 REM
7989 REM*****************************
7999 :
8000 PRINT "[SHIFT]+[HOME]":C0=7:C1=6:C2
     =1:C3=7:C4=0
```

```
8004 C5=4:C6=0:MV=1024:MC=55296
8010 POKE 53265,PEEK(53265) OR 64
8014 POKE 53280,C0:POKE 53281,C3
8020 S$="":FOR I=0 TO 39:S$=S$+" ":NEXT
8024 POKE 53282,C1:POKE 53283,C5
8030 FOR I=0 TO 39:POKE MC+I,C2:POKE MV+
     I,96
8034 NEXT
8040 FOR I=880 TO 999:POKE MC+I,C6
8044 POKE MV+I,160:NEXT
8050 FOR K=1 TO LEN(PR$):A$=MID$(PR$,K,1
     )
8054 A=ASC(A$):IF A<64 THEN A=A+64
8060 POKE (MV+K),A:NEXT
8064 POKE 646,C4:PRINT "[HOME][2 CRSR↑]"
     :RETURN
8077 :
8079 REM*********************************
     S@"@0:8000",8
```

SUBROUTINE 2

POSIZIONAMENTO CURSORE

Spesso è necessario stampare su una certa riga del video, ma nel Basic del C-64 non esistono istruzioni adatte. Il problema è risolto da questa subroutine: basta dare come valore a RC il numero di riga su cui si vuole posizionare il cursore e usare /GOSUB 8100/: il cursore sarà posizionato all'inizio della riga voluta. Ricordate che la riga 0 è quella del titolo e che le righe 22, 23 e 24 sono riservate ai messaggi ed alle domande.

```
8061 REM**********SUBROUTINE #2*********
8063 REM      POSIZIONAMENTO CURSORE
8065 REM*********************************
8067 REM
8069 REM SCOPO:
```

```
8071 REM  QUESTA SUBROUTINE POSIZIONA IL
8073 REM   CURSORE ALL'INIZIO DELLA RIGA
8075 REM                          VOLUTA
8077 REM
8079 REM PRIMA DEL [GOSUB 8100]
8081 REM  RC  > DEVE CONTENERE IL NUMERO
8083 REM    DI RIGA SULLA QUALE SI VUOLE
8085 REM          IL CURSORE (DA 0 A 24)
8087 REM
8089 REM DOPO IL [GOSUB 8100]
8091 REM   NESSUNA VARIABILE E' ALTERATA
8093 REM
8097 REM*********************************
8099 :
8100 POKE 783,0:POKE 781,RC:POKE 782,0
8110 SYS 65520:RETURN
8197 :
8199 REM*********************************
```

SUBROUTINE 3

PRESENTAZIONE MESSAGGI

Le ultime tre righe dello schermo sono riservate a messaggi e domande: questa subroutine si occupa della loro presentazione. Per prima cosa le ultime tre righe sono pulite e colorate col colore C5, ed in fondo alla zona testo è stampata una serie di sei separatori: nelle tre righe saranno presentati i messaggi (o la domanda) preparati nelle stringhe M1$, M2$, M3$. Se il messaggio è corto non è necessario usare tutte e tre le righe: se ne può anche usare una sola (apparirà solo M3$), oppure due (appariranno M1$ e, subito sotto, M3$). Le stringhe non usate devono essere nulle, cioè non devono contenere alcun carattere.

Il cursore è posizionato subito dopo M3$, il colore è quello della zona domande: eventuali risposte possono essere scritte direttamente. Questa subroutine è usata da molte delle subroutines seguenti, che la sfruttano per compiti più specializzati.

```
8545 REM*********SUBROUTINE #3*********
8547 REM      PRESENTAZIONE MESSAGGI
8549 REM*****************************
8551 REM
8553 REM SCOPO:
8555 REM  QUESTA SUBROUTINE PRESENTA DEI
8557 REM      MESSAGGI NELLA ZONA DOMANDE
8559 REM                          DEL VIDEO
8561 REM
8563 REM PRIMA DEL [GOSUB 8600]:
8565 REM  PER AVERE TRE MESSAGGI
8567 REM  M1$ >            PRIMO MESSAGGIO
8569 REM  M2$ >          SECONDO MESSAGGIO
8571 REM  M3$ >            TERZO MESSAGGIO
8573 REM  PER AVERE DUE MESSAGGI
8575 REM  M1$ >            PRIMO MESSAGGIO
8577 REM  M3$ >          SECONDO MESSAGGIO
8579 REM  PER AVERE UN MESSAGGIO
8581 REM  M3$ >            PRIMO MESSAGGIO
8583 REM  M1$;M2$  >     SE NON USATI =""
8585 REM
8587 REM DOPO IL [GOSUB 8600]:
8589 REM  RC  <    CONTIENE IL VALORE 21
8591 REM  IL CURSORE E' POSIZIONATO DOPO
8593 REM   M3$; IL COLORE E' C6(DOMANDE)
8595 REM
8597 REM*****************************
8599 :
8600 FOR K=880 TO 999
8610 POKE (MV+K),160:NEXT
8620 RC=21:GOSUB 8100:PRINT"*******"
8630 POKE 646,C6:POKE 198,0
8640 IF M1$="" THEN 8670
8650 PRINT"[CTRL]+[RVS ON] ";M1$:IF M2$=
     "" THEN 8670
8660 PRINT"[CTRL]+[RVS ON] ";M2$
8670 PRINT"[CTRL]+[RVS ON] ";M3$;"[CTRL]
     +[RVS OFF]";:RETURN
8695 :
8697 REM*****************************
```

SUBROUTINE 4

ATTESA DI TASTO PREMUTO

Questa subroutine usa la precedente per far comparire la domanda, dopo di che ripristina il colore testo e si mette in attesa che un tasto sia premuto. Può essere usata tutte le volte in cui la risposta non deve apparire dopo le domande e quando basta sapere quale tasto è stato premuto: la stringa A$ contiene un solo carattere che corrisponde al tasto azionato.

Questa è la subroutine più importante per le comunicazioni tra il programma e chi lo usa: il programma può valutare il tasto azionato e proseguire in modi differenti.

```
8253 REM**********SUBROUTINE #4**********
8255 REM     ATTESA DI TASTO PREMUTO
8257 REM*****************************
8259 REM
8261 REM SCOPO:
8263 REM  QUESTA SUBROUTINE PRESENTA DEI
8265 REM        MESSAGGI (USA SUB#3), POI
8267 REM    RIPRISTINA IL COLORE TESTO E
8269 REM      ATTENDE CHE SIA PREMUTO UN
8271 REM                            TASTO.
8273 REM
8275 REM PRIMA DEL [GOSUB 8300]:
8277 REM  M1$;M2$;M3$ >          MESSAGGI
8279 REM                     COME PER SUB#3
8281 REM
8283 REM DOPO IL [GOSUB 8300]:
8285 REM  A$  <    CONTIENE IL CARATTERE
8287 REM                 DEL TASTO PREMUTO
8289 REM  RC  <    CONTIENE IL VALORE 21
8291 REM  IL CURSORE E' POSIZIONATO DOPO
8293 REM    M3$; IL COLORE E' C4 (TESTO)
8295 REM
8297 REM*****************************
8299 :
```

```
8300 GOSUB 8600:POKE 646,C4
8310 GET A$:IF A$="" THEN 8310
8320 RETURN
8397 :
8399 REM*****************************
```

SUBROUTINE 5

PAUSA

Questa è la prima subroutine di comunicazione specializzata: si usa quando il programma ha terminato una fase ed attende la via libera da parte di chi lo usa per continuare.

Permette quindi di avere tutto il tempo necessario per osservare i risultati ed i dati sul video.

Per semplificare l'uso, due messaggi sono fissi: M1$ è uguale a "SE VUOI", ed M3$ è uguale a "PREMI UN TASTO": basta quindi specificare solo M2$ (per esempio con "CONTINUARE", oppure "TERMINARE", o altre frasi simili).

Quando si usa questa subroutine in genere non si controlla quale tasto sia stato premuto: il programma continua nell'unico modo possibile a quel punto, dopo aver effettuato una PAUSA...

```
8147 REM*********SUBROUTINE #5*********
8149 REM                PAUSA
8151 REM*****************************
8153 REM
8155 REM SCOPO:
8157 REM   QUESTA SUBROUTINE PRESENTA UN
8159 REM      MESSAGGIO E CREA UNA PAUSA
8161 REM   NELL'ESECUZIONE DEL PROGRAMMA
8163 REM    PRINCIPALE FINO A QUANDO NON
8165 REM   SI PREME UN TASTO (USA SUB#4)
8167 REM
8169 REM PRIMA DI [GOSUB 8200]:
8171 REM  M2$ >          PARTE CENTRALE DEL
```

```
8173 REM        MESSAGGIO (M1$ E M3$ SONO
8175 REM              FISSATI DALLA SUB.)
8177 REM
8179 REM DOPO IL [GOSUB 8200]:
8181 REM  A$  <    CONTIENE IL CARATTERE
8183 REM                DEL TASTO PREMUTO
8185 REM  RC  <    CONTIENE IL VALORE 21
8187 REM  M1$ <       CONTIENE="SE VUOI"
8189 REM  M3$ <CONTIENE="PREMI UN TASTO"
8191 REM  IL CURSORE E' POSIZIONATO DOPO
8193 REM    M3$; IL COLORE E' C4 (TESTO)
8195 REM
8197 REM*************************************
8199 :
8200 M1$="SE VUOI":M3$="PREMI UN TASTO"
8297 :
8299 REM*************************************
```

SUBROUTINE 6

DOMANDA SÌ/NO

Questa è una subroutine di comunicazione più complessa della precedente, nella quale infatti si tiene conto della risposta, che però può essere solo di due tipi: o sì (tasto /S/) oppure no (tasto /N/). Come per le subroutines precedenti, sono presentati i tre messaggi e si attende l'azionamento di un tasto: se però questo è diverso da /S/ e da /N/, vengono ripresentate le tre stringhe che formano la domanda. Dopo l'uso della subroutine la stringa RI$ conterrà un solo carattere, la risposta "S" oppure "N".

Questa subroutine è utile quando un programma può continuare in due modi diversi: usandola in quel punto si chiede in pratica a chi usa il programma quale alternativa vuole, ed esaminando RI$ il programma continuerà in un modo o nell'altro.

È opportuno che in M3$ sia indicato il tipo di risposta attesa, per esempio con "SI o NO" oppure brevemente con "(S/N)".

```
8353 REM*********SUBROUTINE #6**********
8355 REM          DOMANDA SI/NO
8357 REM*********************************
8359 REM
8361 REM SCOPO:
8363 REM  QUESTA SUBROUTINE PRESENTA DEI
8365 REM  MESSAGGI (USA SUB#4) E ACCETTA
8367 REM       SOLO "S" O "N" COME TASTO
8369 REM                     DI RISPOSTA
8371 REM
8373 REM PRIMA DI [GOSUB 8400]:
8375 REM  M1$;M2$;M3$ >          MESSAGGI
8377 REM                   COME PER SUB#3
8379 REM
8381 REM DOPO IL [GOSUB 8400]:
8383 REM  RI$ < CONTIENE LA RISPOSTA "S"
8385 REM                       OPPURE "N"
8387 REM  A$  <                UGALE A RI$
8389 REM  RC  <       ASSUME IL VALORE 21
8391 REM  IL CURSORE E' POSIZIONATO DOPO
8393 REM   M3$; IL COLORE E' C4  (TESTO)
8395 REM
8397 REM*********************************
8399 :
8400 GOSUB 8300:RI$=A$
8410 IF RI$="S" OR RI$="N" THEN RETURN
8420 GOTO 8400
8497 :
8499 REM*********************************
```

SUBROUTINE 7

INGRESSO DATI NUMERICI

Spessissimo occorre che chi usa un programma introduca dei numeri che saranno poi utilizzati dal programma: di questo si occupa la SUBROUTINE 7, che è piuttosto complessa.

Vediamo come funziona: per prima cosa presenta i soliti tre messaggi che questa volta chiederanno di introdurre un valore numerico, poi si mette in attesa di una risposta. La risposta apparirà nella zona domande, dopo M3$, così come è battuta, fino a quando non viene usato il tasto /RETURN/, che termina la risposta. La stringa è tradotta in numero, e questo è stampato con M3$ nella parte testo dello schermo, in modo che chi usa il programma può ricordarsi quali numeri ha già immesso, e controllare se ha fatto qualche errore.

I numeri immessi sono resi in un vettore, cioè sono attribuiti come valore ad un insieme di variabili aventi tutte lo stesso nome (viene usato VR), e distinte tra loro da un numero, o indice (I). Occorrerà quindi preoccuparsi che I abbia il valore desiderato prima di usare questa subroutine.

Questa subroutine incrementa da sola I: se quindi in un programma è usata più volte di seguito, per introdurre variabili diverse, basta porre I=0 prima di usarla per la prima volta: il primo numero sarà così attribuito a VR(0), il secondo a VR(1), il terzo a VR(2), e così via.

Se si prevedono più di dieci valori occorre una istruzione tipo /DIM VR(max)/ per dimensionare il vettore, dove (max) è il numero massimo di valori previsto. Dall'indice dipende anche su che riga dello schermo viene scritta la domanda M3$ seguita dalla risposta: si usano le righe da 2 a 13 per presentare i primi 12 numeri, poi il tredicesimo è presentato nuovamente nella riga 2 (cancellando quello che già c'è), e così via, in modo da poter vedere sempre gli ultimi 12 numeri: questo è molto utile quando bisogna introdurre molti numeri di seguito.

La SUBROUTINE 7 fa ancora un'altra cosa: se infatti chi usa il programma non introduce un numero, ma batte subito /RETURN/, non viene stampato nulla nella parte testo, a VR(I) viene assegnato il valore 0, e la stringa RI$, che normalmente vale "S", diventa "N". Questo è comodo quando non si sa a priori quanti sono i numeri da inserire: in questo modo chi usa il programma può segnalare di aver terminato, come se rispondesse ad un'implicita domanda di tipo SI/NO.

```
8701 REM*********SUBROUTINE #7*********
8703 REM      INGRESSO DATI NUMERICI
8705 REM*****************************
8707 REM
8709 REM SCOPO:
8711 REM  QUESTA SUBROUTINE PRESENTA DEI
8713 REM         MESSAGGI (USA SUB#3) PER
8715 REM      CHIEDERE DEI DATI NUMERICI.
8717 REM      SE SI BATTONO DELLE CIFRE E
8719 REM  [RETURN] IL NUMERO E' IN VR(I)
8721 REM    ED E' STAMPATO SULLO SCHERMO
8723 REM      INSIEME A M3$; INOLTRE I E'
8725 REM     INCREMENTATO E RI$ VALE "S".
8727 REM          SE INVECE SI BATTE SOLO
8729 REM         [RETURN], ALLORA VR(I)=0
8731 REM  NON STAMPA NULLA SULLO SCHERMO
8733 REM       MA INCREMENTA I E RI$="N".
8735 REM
8737 REM PRIMA DI [GOSUB 8800]:
8739 REM  M1$;M2$;M3$ >    MESSAGGI COME
8741 REM       PER SUB#3; M3$ E' STAMPATO
8743 REM      ANCHE NELLA PARTE SUPERIORE
8745 REM         CON IL NUMERO INTRODOTTO
8747 REM  I    >INDICA QUALE DATO CHIEDE,
8749 REM        LA PRIMA VOLTA PORRE I=0.
8751 REM       SE SI PREVEDONO PIU' DI 10
8753 REM       DATI OCCORRE UN'ISTRUZIONE
8755 REM            TIPO "DIM VR(MASSIMO)"
8757 REM
8759 REM DOPO IL [GOSUB 8800]:
8761 REM  RI$ < ="S" OPPURE "N"
8763 REM VR(I)<   CONTIENE IL DATO NUOVO
8765 REM  I   <          INCREMENTATO DI 1
8767 REM  ST$ < CONTIENE LA STRINGA COME
8769 REM                    E' STATA BATTUTA
8771 REM  RC  <      IL NUMERO DELLA RIGA
8773 REM      DOVE HA SCRITTO LA RISPOSTA
8775 REM                          [DA 3 A 13]
8777 REM    IL CURSORE E' POSIZIONATO DUE
```

```
8779 REM                    RIGHE PIU' IN BASSO
8781 REM
8791 REM*****************************************
8793 REM*********SUBROUTINE #7*********
8797 REM*****************************************
8799 :
8800 GOSUB 8890:IF RI$="N" THEN RETURN
8810 PRINTVR(I-1):RETURN
8890 GOSUB 8600:ST$="":RI$="N"
8900 GET A$:IF A$=CHR$(13) THEN 8960
8910 IF A$=CHR$(20) THEN A$=A$+"[SHIFT]+
     [CRSR←] [SHIFT]+[CRSR←]":GOTO 8940
8920 IF A$<" " THEN 8900
8930 ST$=ST$+A$:GOTO 8950
8940 ST$=MID$(ST$,1,ABS(LEN(ST$)-1))
8950 PRINT"[CTRL]+[RVS ON]";A$;"[CTRL]+[
     RVS OFF]";:GOTO 8900
8960 RC=I-(INT(I/12)*12)+3:GOSUB 8100
8970 POKE 646,C4:VR(I)=VAL(ST$):I=I+1
8980 PRINTS$;"[SHIFT]+[CRSR↑] ";:IF ST$=
     "" THEN RETURN
8990 PRINTM3$;:RI$="S":RETURN
8997 :
8999 REM*****************************************
```

SUBROUTINE 8

INGRESSO STRINGA

A volte è necessario che chi usa il programma introduca una stringa di caratteri anziché un numero: di questo si occupa la SUBROUTINE 8.

Sostanzialmente si comporta come la subroutine precedente, con alcune differenze: intanto nella zona testo apparirà la stringa M3$ seguita dalla risposta così come è stata scritta (e non tradotta in numero), poi la stringa di risposta che ci interessa sarà ST$.

Anche in questo caso però la risposta è interpretata come un numero, anche se in genere non lo è, per cui il

valore ottenuto sarà spesso 0, e tale valore verrà anche ora attribuito alla variabile VR(I),

Anche con questa subroutine occorre fare attenzione al valore di I: indica quale variabile del vettore sarà usata e su quale riga sarà stampata la risposta.

Come per la subroutine 7, anche la SUBROUTINE 8 rende la variabile RI$ uguale a "N" se come risposta si è battuto solo /RETURN/, e uguale a "S" negli altri casi.

```
8927 REM**********SUBROUTINE #8*********
8929 REM          INGRESSO STRINGA
8931 REM*****************************
8933 REM
8935 REM SCOPO:
8937 REM  QUESTA SUBROUTINE PRESENTA DEI
8939 REM          MESSAGGI (USA SUB#3) PER
8941 REM             CHIEDERE UNA STRINGA.
8943 REM          LA STRINGA E' CONVERTITA
8945 REM       IN VALORE NUMERICO IN VR(I)
8947 REM      ED E' STAMPATO SULLO SCHERMO
8949 REM       INSIEME A M3$; INOLTRE I E'
8951 REM      INCREMENTATO E RI$ VALE "S".
8953 REM           SE INVECE SI BATTE SOLO
8955 REM         [RETURN], ALLORA VR(I)=0,
8957 REM           STAMPA UNA RIGA BIANCA,
8959 REM           INCREMENTA I E RI$="N".
8961 REM
8963 REM PRIMA DI [GOSUB 8800]:
8965 REM  M1$;M2$;M3$ >     MESSAGGI COME
8967 REM        PER SUB#3; M3$ E' STAMPATO
8969 REM       ANCHE NELLA PARTE SUPERIORE
8971 REM          CON LA STRINGA INTRODOTTA
8973 REM  I    > INDICA QUALE VR(I) USARE
8975 REM
8977 REM DOPO IL [GOSUB 9000]
8979 REM  RI$ < ="S" OPPURE "N"
8981 REM VR(I)<          CONTIENE IL VALORE
8983 REM              NUMERICO DELLA STRINGA
8985 REM  I    <          INCREMENTATO DI 1
```

```
8987 REM  ST$ <        LA STRINGA BATTUTA
8989 REM  RC  <     LA RIGA DELLO SCHERMO
8991 REM      CON LA RISPOSTA (DA 3 A 13)
8993 REM    IL CURSORE E' POSIZIONATO DUE
8995 REM                RIGHE PIU' IN BASSO
8997 REM*****************************
8999 :
9000 GOSUB 8890:IF RI$="N" THEN RETURN
9010 PRINT ST$:RETURN
9087 :
9089 REM*****************************
```

SUBROUTINE 9

SCELTA PROGRAMMI

Quando un programma può continuare in due modi diversi, possiamo usare la SUB DOMANDA SI/NO, che abbiamo già visto, e a seconda della risposta il programma continuerà in un modo o nell'altro. Quando però ci sono più di due alternative, useremo questa subroutine che permette di scegliere fino a nove possibilità. Questa situazione si presenta spesso all'inizio di programmi complessi, quando occorre scegliere tra varie parti del programma che eseguono compiti diversi.

Usando questa subroutine, appaiono nella zona testo i titoli a partire dalla riga 3 numerati da 1 in poi ed appare la domanda (nella zona domande) "SCEGLI CHE COSA" "VUOI FARE".

La risposta deve essere una cifra compresa tra quelle apparse: se il tasto premuto non è un numero oppure è maggiore di quelli possibili la domanda è ripresentata.

Come detto i titoli al massimo possono essere nove, e sono contenuti in un vettore di stringhe, vale a dire in variabili tipo stringa tutte con lo stesso nome (PN$) e distinte da un numero: il primo titolo sarà in PN$(1), il secondo in PN$(2) e così via; inoltre la variabile PN dovrà indicare il numero di titoli usati. Per esempio, se ci sono tre possibilità dovrà essere /PN=3/, e /PN$(1)="primo titolo": PN$(2)="secondo titolo": PN$(3)="terzo titolo"/.

La risposta deve essere uno dei tre tasti 1, 2, 3, e nes-

sun altro tasto è accettato. Il valore corrispondente al tasto premuto si ritrova nella variabile PS, nel nostro esempio sarà 1, o 2, o 3, e deve essere usato per continuare nel modo voluto, con una istruzione BASIC del tipo /ON PS GOSUB.../ ovvero /ON PS GOTO.../.

```
9339 REM*********SUBROUTINE #9**********
9341 REM          SCELTA PROGRAMMI
9343 REM*********************************
9345 REM
9347 REM SCOPO:
9349 REM  QUESTA SUBROUTINE STAMPA SULLO
9351 REM      SCHERMO DEI TITOLI (MAX. 9)
9353 REM     CONTRADDISTINTI DA UN NUMERO
9355 REM           E CHIEDE DI SCEGLIERE.
9357 REM        IL NUMERO SCELTO E' IN PS
9359 REM
9361 REM PRIMA DEL [GOSUB 8400]:
9363 REM  PN  >       NUMERO DELLE SCELTE
9365 REM                              MAX 9
9367 REM  PN$(1)...PN$(PN) >      TITOLI,
9369 REM                    MAX 16 CARATTERI
9371 REM                 (SE 17 CHR I TITOLI
9373 REM       SONO SPAZIATI DI UNA RIGA)
9375 REM
9377 REM DOPO IL [GOSUB 9400]:
9379 REM  M1$ <   VALE:"SCEGLI CHE COSA"
9381 REM  M2$ <                  VALE:""
9383 REM  M3$ <        VALE:"VUOI FARE ?"
9385 REM  RC  <        ASSUME IL VALORE 3
9387 REM  PS  <          E' IL NUMERO DEL
9389 REM                 PROGRAMMA SCELTO
9391 REM  IL CURSORE E' POSIZIONATO DOPO
9393 REM   M3$; IL COLORE E' C4  (TESTO)
9395 REM
9397 REM*********************************
9399 :
9400 RC=3:GOSUB 8100
9410 FOR I=1 TO PN
```

```
9420 PRINT I"  ";I;"= ";PN$(I):NEXT
9430 M1$="SCEGLI CHE COSA":M2$=""
9440 M3$="VUOI FARE >":GOSUB 8300
9450 PS=ASC(A$)-48
9460 IF PS<1 OR PS>PN THEN 9440
9470 RETURN
9497 :
9499 REM*********************************
```

SUBROUTINE 10

MODIFICA PROGRAMMI

Nei programmi matematici di questo libro capita spesso di dover introdurre un'espressione matematica da calcolare, e questo si può fare bene modificando una o più righe del programma BASIC, usando questa subroutine.

Questa, in effetti non è una subroutine, per cui il suo uso richiede più attenzione, anche se è abbastanza semplice.

Innanzitutto occorre preparare due stringhe P1$ e P2$, che contengono le due nuove righe BASIC così come le vogliamo, complete del numero di riga, ed una terza stringa P3$ che contiene un comando BASIC per far riprendere un programma dal punto in cui vogliamo, tipo "RUN 2000" oppure "GOTO 100".

Poi si può usare la SUBROUTINE 10 andandoci con un /GOSUB 9200/ o anche un /GOTO 9200/.

Fate attenzione, perché dopo queste operazioni si è perso quello che c'era sullo schermo, ed anche alcuni valori delle variabili: è opportuno quindi che la riga indicata in P3$ sia una riga iniziale di un programma, perché in genere si comincia proprio pulendo lo schermo.

```
9131 REM*********SUBROUTINE #10*********
9133 REM          MODIFICA PROGRAMMI
9135 REM*********************************
9137 REM
```

```
9139 REM SCOPO:
9141 REM   QUESTA SUBROUTINE PERMETTE DI
9143 REM     AGGIUNGERE O MODIFICARE DUE
9145 REM          RIGHE DEL PROGRAMMA IN
9147 REM                        MEMORIA.
9149 REM
9151 REM ATTENZIONE:
9153 REM     NON E' UNA VERA SUBROUTINE:
9155 REM   L'USO DI QUESTA SUBROUTINE FA
9157 REM    PERDERE TUTTE LE VARIABILI E
9159 REM          NON TORNA AL PROGRAMMA
9161 REM       CHIAMANTE, MA RIPRENDE LA
9163 REM          ESECUZIONE DA UNA RIGA
9165 REM                   INDICATA IN P3$
9167 REM
9169 REM PRIMA DI [GOSUB 9200]:
9171 REM  P1$ > PRIMA RIGA DA AGGIUNGERE
9173 REM   AL PROGRAMMA, DEVE COMINCIARE
9175 REM                       CON UN NUMERO
9177 REM  P2$ >  SECONDA RIGA (COME P1$)
9179 REM  P3$ > TERZA RIGA DI ESECUZIONE
9181 REM     IMMEDIATA, TIPO "RUN 1000",
9183 REM       CHE RILANCIA IL PROGRAMMA
9185 REM
9187 REM DOPO IL [GOSUB 9200]:
9189 REM   TUTTE LE VARIABILI VALGONO 0,
9191 REM      P1$ E P2$ SONO AGGIUNTE AL
9193 REM     PROGRAMMA, P3$ E' ESEGUITA.
9195 REM   LO SCHERMO DEVE ESSERE PULITO
9197 REM*****************************
9199 :
9200 PRINT "[SHIFT]+[HOME][2 CRSR↑]";P1$
     :PRINT P2$
9210 PRINT P3$;"[HOME]";
9220 POKE 198,3:POKE 631,13:POKE 632,13
9230 POKE 633,13:END
9297 :
9299 REM*****************************
```

PROVE COLORI

Come esempio di uso delle subroutines presentate, ecco un programma che permette di modificare a volontà i colori delle varie zone dello schermo per scegliere la combinazione preferita: il programma fornisce anche i valori numerici da attribuire alle quattro variabili della riga 8000 per riottenere in seguito la combinazione scelta.

Dopo aver dato /RUN/ il programma dovrebbe presentarsi come in figura (se non avete fatto errori copiandolo, naturalmente...): la prima riga indica i valori di C0, C1, C2, C3, C4, C5, C6 attualmente in uso, le successive i colori disponibili. Nelle ultime tre righe appare la prima domanda: se la risposta è positiva appaiono le altre domande, per le varie zone dello schermo. Al termine delle domande, premendo un tasto, lo schermo assume i nuovi colori e ripresenta la domanda iniziale.

IL PROGRAMMA

Vediamo da vicino come è costruito questo programma e come sono utilizzate alcune delle subroutines precedenti.

Per prima cosa si deve pulire lo schermo e presentare il titolo: la riga 1000 definisce il titolo e la riga 1004 chiama la SUBROUTINE 1 che risolve il problema. Per non sprecare righe del video, subito sotto il titolo, in una sola riga, stampiamo i valori di C01,C1, C2, C3, C4, C5, C6 e sotto l'elenco dei colori disponibili. Questo è il momento della prima domanda: vogliamo modificare i colori?

Le righe 2000 e 2010 preparano le frasi («stringhe» direbbe Mister Bit) per la domanda, e la SUBROUTINE 6 si occupa di tutto, cioè di presentarle e attendere la risposta: se la risposta è /N/ vuol dire che non vogliamo cambiare i colori: il programma può quindi finire. Se infatti RI$="N" il programma termina a riga 2020.

La domanda successiva chiede il colore del bordo. La risposta deve essere un numero, compreso tra 0 e 15: si userà quindi la SUBROUTINE 7, che presenterà le domande e aspetterà la risposta. La riga 2060 con-

trolla anche che la risposta non superi 15: se questo accade, per sbaglio, ripresenta la domanda. Mister Bit interviene: dimentico qualche cosa. È vero, occorre ricordarsi di dare il giusto valore ad I prima di usare la SUBROUTINE 7, ed infatti nella riga 2040 ad I si dà il valore 0, inoltre l'ultima parte della domanda, con la risposta data, apparirà alla riga 2, ed anche a questo penserà la SUBROUTINE 7.

Il numero dato in risposta sarà attribuito alla variabile VR(0).

Per chiedere il colore di fondo del titolo, le prime due frasi della domanda possono restare le stesse: la riga 2070 prepara la terza, prima di usare la SUBROUTINE 7.

Non dobbiamo preoccuparci della variabile I, perché

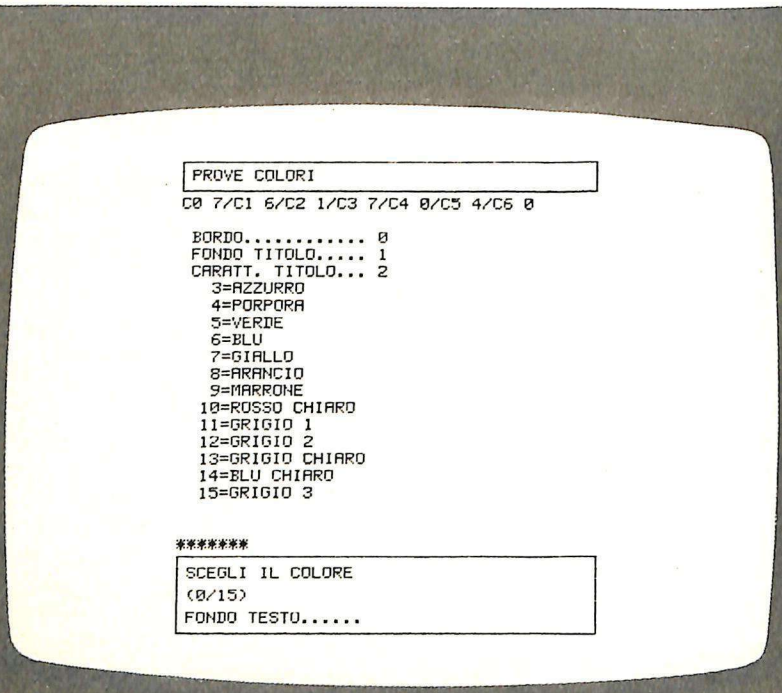
```
PROVE COLORI

C0 7/C1 6/C2 1/C3 7/C4 0/C5 4/C6 0

   0=NERO
   1=BIANCO
   2=ROSSO
   3=AZZURRO
   4=PORPORA
   5=VERDE
   6=BLU
   7=GIALLO
   8=ARANCIO
   9=MARRONE
  10=ROSSO CHIARO
  11=GRIGIO 1
  12=GRIGIO 2
  13=GRIGIO CHIARO
  14=BLU CHIARO
  15=GRIGIO 3

*******

VUOI CAMBIARE
I COLORI (S/N) ?
```

è incrementata dalla SUBROUTINE 7 automaticamente.

La riga 2080 controlla che la risposta non superi 15, altrimenti si ritorna a riga 2070, dopo aver riportato I a valore 1.

Il numero del colore scelto sarà attribuito a VR(1), ed il programma continua in modo simile chiedendo successivamente gli altri colori, per il titolo (righe 2090 e 2100) per lo sfondo e i caratteri della zona testo (righe 2150-2180).

La riga 2200 usa la SUBROUTINE 5 per effettuare una pausa: premendo un tasto qualsiasi il programma continua attribuendo a C0-C6 i nuovi valori che sono in VR(0)-VR(6) (riga 2220), ed usando poi la SUBROUTINE 1, saltandone però le prime due righe: in questo modo i colori che abbiamo scelto appariranno sullo scher-



mo. Quindi il programma ritorna a riga 1006 completando lo schermo e ripresentando la prima domanda.

Se volete usare in seguito, anche con altri programmi, una certa combinazione di colori, prendete nota delle cifre che appaiono sulla prima riga. Caricate ora il vostro programma «SUBROUTINES» e modificate la riga 8000 con i nuovi valori letti sullo schermo, e salvate di nuovo le subroutines sul nastro.

```
 999 REM*********************************
1000 PR$=        "PROVE COLORI"
1001 REM*********************************
1003 :
1004 GOSUB 8000
1006 PRINT"[2 SHIFT]+[CRSR↑]C0";C0;"[SHI
     FT]+[CRSR←]/C1";C1;"[SHIFT]+[CRSR←]
     /C2";C2;
1008 PRINT"[SHIFT]+[CRSR←]/C3";C3;"[SHIF
     T]+[CRSR←]/C4";C4;"[SHIFT]+[CRSR←]/
     C5";C5;"[SHIFT]+[CRSR←]/C6";C6
1010 PRINT"[CRSR↑]    0=NERO"
1020 PRINT"    1=BIANCO"
1030 PRINT"    2=ROSSO"
1040 PRINT"    3=AZZURRO"
1050 PRINT"    4=PORPORA"
1060 PRINT"    5=VERDE"
1070 PRINT"    6=BLU"
1080 PRINT"    7=GIALLO"
1100 PRINT"    8=ARANCIO"
1110 PRINT"    9=MARRONE"
1120 PRINT"   10=ROSSO CHIARO"
1130 PRINT"   11=GRIGIO 1"
1140 PRINT"   12=GRIGIO 2"
1150 PRINT"   13=GRIGIO CHIARO"
1160 PRINT"   14=BLU CHIARO"
1170 PRINT"   15=GRIGIO 3"
1175 :
2000 M1$="VUOI CAMBIARE":M2$=""
2010 M3$="I COLORI (S/N) ?":GOSUB 8400
```

```
2020 IF RI$="N" THEN PRINT"[SHIFT]+[HOME
     ]":END
2023 :
2040 M1$="SCEGLI IL COLORE":M2$="(0/15)"
     :I=0
2050 M3$="BORDO............":GOSUB 8800
2060 IF VR(0)>15 THEN I=0:GOTO 2050
2070 M3$="FONDO TITOLO.....":GOSUB 8800
2080 IF VR(1)>15 THEN I=1:GOTO 2070
2090 M3$="CARATT. TITOLO...":GOSUB 8800
2100 IF VR(2)>15 THEN I=2:GOTO 2090
2110 M3$="FONDO TESTO......":GOSUB 8800
2120 IF VR(3)>15 THEN I=3:GOTO 2110
2130 M3$="CARATT. TESTO....":GOSUB 8800
2140 IF VR(4)>15 THEN I=4:GOTO 2130
2150 M3$="FONDO DOMANDE....":GOSUB 8800
2160 IF VR(5)>15 THEN I=5:GOTO 2150
2170 M3$="CARATT. DOMANDE..":GOSUB 8800
2180 IF VR(6)>15 THEN I=6:GOTO 2170
2183 :
2200 M2$="PROVARE I COLORI":GOSUB 8200
2203 :
2220 C0=VR(0):C1=VR(1):C2=VR(2):C3=VR(3)
     :C4=VR(4):C5=VR(5):C6=VR(6)
2230 PRINT"[SHIFT]+[HOME]":GOSUB 8010:GO
     TO 1006
2239 REM*****************************
```

Se desiderate essere tenuti al corrente delle novità delle nostre collane compilate e spediteci in busta chiusa il tagliando sottostante. Vi terremo periodicamente informati.

Se desiderate sottoporci, per eventuale pubblicazione, vostri specifici lavori, programmi o esperienze sulle macchine più diffuse, o più recenti, inviateci degli esempi, anche parziali. Saremo lieti di esaminarli e, nel caso, di prendere contatto con Voi.

Saranno inoltre molto gradite eventuali osservazioni, segnalazioni, modifiche o aggiunte ai programmi del presente libro, e proposte e suggerimenti per pubblicazioni successive.

Desidero essere tenuto al corrente della pubblicazione delle Vs. novità.

Ho tratto questa scheda dal volume .....

.....

Acquistato presso

☐ Edicola .....

☐ Libreria .....

.....

Nome Cognome

.....

Professione

.....

Indirizzo Cap - Città

ALGEBRA — COMMODORE 64

PAGINA	RIGA	ERRATA	CORRIGE
14	1	tutto il	tutto con il
18	14	5+3	5-3
30	2010	M1$=	I=0:M1$=
31	3000	PN$(2)	PR$=PN$(2)
31	3010	M1$=	I=0:M1$=
43	8000	{ eliminare le righe>	8000-8020 }
68	figura	{ eliminare titolo in	basso nel video }
101	formula 4100		N =
101	formula 4200		N =
107	8079	{ eliminare la riga>	8079 S*'30:8000',S }
116	8910	THEN A$=A$+"[SHIFT]	THEN A$="[SHIFT]
120	9420	PRINT I" ";	PRINT " ";

MODIFICHE suggerite dai lettori:

72	3340	PRINT";N;"/";D; :RETURN	RETURN
70	1392	{ aggiungere la riga>	1392 IF I>0 THEN PRINT ";N;"/";D; }
71	1612	{ aggiungere la riga>	1612 IF I>0 THEN PRINT ";N;"/";D; }

(in questo modo si rende più chiara la presentazione dei risultati)

ALGEBRA con COMMODORE 64

Tanti fantastici programmi,
appositamente ideati e realizzati secondo gli attuali
programmi scolastici italiani
per garantirvi un migliore approccio alla materia,
una migliore comprensione,
uno studio più razionale e meno faticoso
sfruttando al massimo tutte le meravigliose possibilità
del vostro... compagno di giochi!

Tra gli argomenti trattati in questo libro:

LE ESPRESSIONI ALGEBRICHE; I POLINOMI IN X;
I POLINOMI ORDINATI; LE EQUAZIONI DI PRIMO GRADO;
IL METODO DI CRAMER; IL SISTEMA X,Y IMPLICITO;
I NUMERI COMPLESSI; LE SOLUZIONI DELLE EQUAZIONI
DI SECONDO GRADO; LE PROGRESSIONI ARITMETICHE;
E, IN APPENDICE, SUBROUTINES E PROVE COLORI.

Cambiate la vostra scuola
Migliorate il vostro rendimento
Mettete il computer al vostro servizio!

Con questo libro non solo scoprirete con amici e
insegnanti nuove prospettive, nuove dimensioni,
ma imparerete voi stessi a modificare i programmi
proposti o a realizzarne altri, sempre nuovi, personali,
per le vostre specifiche esigenze.

Studiare
col computer
È PIÙ DIVERTENTE
È PIÙ FACILE
È PIÙ...!!!

CL 006-0176-4 ISBN 88-7605-176-7

L. 8.000 (...)