

Sul numero 6 di MCmicrocomputer abbiamo pubblicato un programma che permetteva, nota una funzione  $f(x)$ , di disegnarne il grafico. A conclusione dell'articolo invitavamo i lettori ad interessarsi all'argomento, proponendo i loro programmi concernenti il tema della "rappresentazione grafica" sul PC-1211.

Dobbiamo dire che la nostra proposta è stata accolta con grande entusiasmo, e prova ne sono i due programmi pubblicati su questo numero. Il primo è relativo al problema della stampa di grafici di funzioni a due variabili: nonostante le notevoli limitazioni della stampante CE-122, il lettore Roberto Dalla Mura di Venezia è riuscito a realizzare grafici di grande effetto. Il secondo programma riguarda invece la rappresentazione grafica di dati numerici, qualora questi non fossero direttamente esprimibili attraverso una funzione.

### Grafici di funzioni a 2 variabili

di Roberto Dalla Mura - Venezia

Quando si tratta di tracciare il grafico di una funzione a 2 variabili, il problema viene risolto con un disegno in prospettiva in cui si osserva una superficie opportunamente manipolata, al fine di avere l'idea della forma del solido cui si riferisce la funzione stessa. Il problema può essere risolto anche con una calcolatrice come la Sharp PC-1211, corredata della stampante CE-122, se ci si accontenta di ottenere, anziché un disegno prospettico, una "mappa" della funzione: un disegno, cioè, che riproduca gli alti e i bassi, più o meno quando si vogliono rappresentare su una carta orografica le colline o le montagne, usando le linee altimetriche.

```

10:"A"PAUSE "AL
L CLEAR":
CLEAR
20:"B"INPUT "MA
X F(XY)=";A
(31)
30:"C"INPUT "MIN F
(XY)=";A(30)
40:"D"INPUT "IN
IZ,X=";I;U
50:INPUT "INCR,
X=";S
60:INPUT "INIZ,
Y=";V
70:INPUT "INCR,
Y=";T
75:IF T=0LET T=
1.613S
80:"D"INPUT "RI
GHE=";A(32)
90:INPUT "NASTR
I=";A(33):
IF A(32)=0
LET A(32)=
INT (.9,A(33)
)
100:INPUT "N.CAR
ATTERI?";I;Q:
R=Q
110:FOR R=36TO 3
5+0:INPUT "#
=";A(R):
NEXT R
132:INPUT "ASSE
X=";A(35):"
ASSE Y=";A(
34)
135:IF A(31)=0IF
A(30)=0GOSUB
340
139:GOSUB 400
140:"S"FOR A(29)
=1TO A(33)
145:GOSUB 410:
GOSUB 420:
GOSUB 410
150:Y=V
160:FOR A(28)=1
TO A(32)
165:IF A(35)=A(2
8)PRINT "----
-----"
GOTO 270
170:X=U
180:FOR W=1TO 16
185:IF A(34)=W+1
6A(29)-16LET
A(W)="!":
GOTO 250
190:GOSUB 500
195:PAUSE A(28),
W
200:X=X+S
210:IF Z(A(30)
LET A(W)="-"
GOTO 250
220:IF Z(A(31)
LET A(W)="+
GOTO 250
230:A(27)=INT ((
Z-A(30))/A(
31)-A(30))*C
Q-E-04)+36
240:A(W)=A*(A(2
7))
250:NEXT W
260:PRINT A(W);
C;D;E;F;G;
G;H;I;J;K;L;M;N;
O;P;Q
270:Y=Y+T
280:NEXT A(28)
290:U=U+16S
310:NEXT A(29):
GOSUB 410:
GOSUB 420:
GOSUB 410:
END
340:Y=V;X=U:
GOSUB 500:A(
31)=Z:A(30)=
Z
350:Y=V
351:FOR A(28)=1
TO A(32)/4+1
352:X=U
353:FOR W=1TO 4A
(33)
354:GOSUB 500
355:X=X+4S
356:IF A(31)<2
LET A(31)=2
357:IF A(30)>2
LET A(30)=2
358:NEXT W
359:Y=Y+4T
360:NEXT A(28)
361:RETURN
400:GOSUB 420:
GOSUB 410:
PRINT "MAX M
IN F(XY)=";
PRINT A(31),
A(30)
401:PRINT "X DA=
";U;X=U+16A(
33):S;PRINT "
";Y;Y=Y+4T
402:Y=Y+A(32):T:
PRINT "A=";Y
PRINT "INCR
=";T
404:PRINT "RIGHE
=";A(32):
PRINT "NASTR
I=";A(33):
PRINT "CARAT
TERI=";I;Q
405:FOR W=1TO A(
32)-18:PRINT
";NEXT W:
RETURN
410:PRINT " ";
PRINT " ";
PRINT " ";
RETURN
420:PRINT "+---+
---+---+---+
";A(29):
RETURN
500:Z=E4/(100+XY)
:RETURN
    
```

Figura 2 - Listing del programma "Grafici di funzioni a 2 variabili".

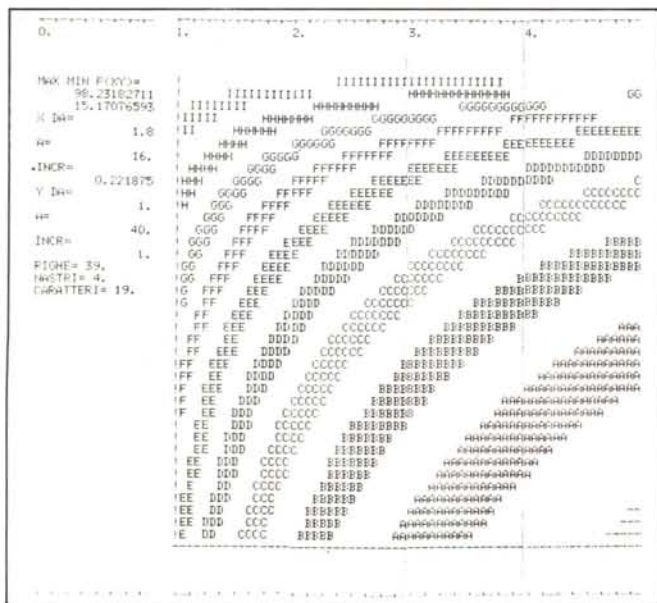


Figura 1 - Il grafico rappresenta la funzione  $z = \frac{10.000}{100 + xy}$  a cui si fa riferimento nell'esempio.

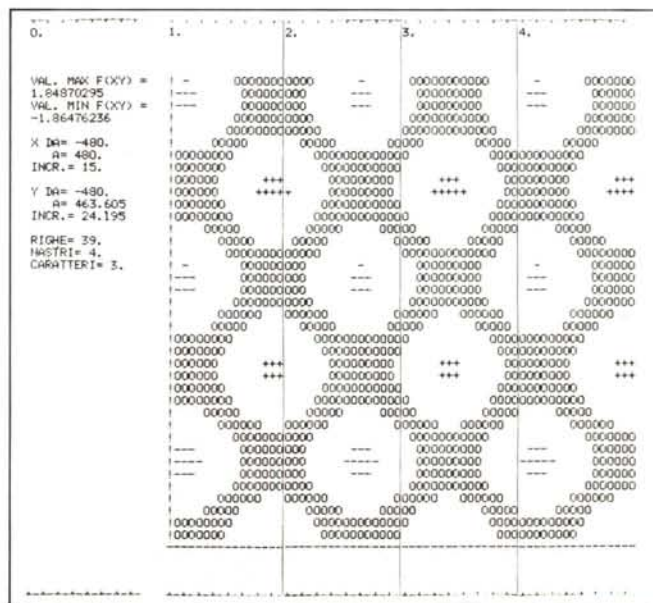


Figura 4 - Questo "ricamo" è stato costruito con  $z = \sin x + \sin y$ .

Otterremo dunque una matrice  $n \times m$  in cui l'elemento  $ij$ , risultato della nostra funzione  $Z=f(x,y)$ , sarà stampato con un certo carattere scelto opportunamente tra quelli assegnati. Sarà proprio il tipo di carattere scelto ad indicarci l'altezza della nostra funzione, cioè l'intervallo di valori in quel punto.

Supponiamo di avere una certa  $f(x,y)$  (come quella in fig. 1 per esempio); per prima cosa valutiamo entro quali limiti sarà compreso il massimo ed il minimo del nostro intervallo di campionamento, in modo tale da poter suddividere l'insieme dei valori ottenibili in tante fettine quanti sono i simboli che vorremo usare per il nostro grafico.

Calcoleremo quindi il valore di  $Z$ , progressivamente in ogni punto della matrice, incrementando di volta in volta il valore della  $x$  e della  $y$  fino ad esaurimento delle righe e delle colonne. Ogni  $Z$  subirà una trasformazione (lineare in questo caso, ma adattabile secondo le esigenze) in un valore che indicherà di quale "fettina" si tratta, tale valore servirà da indirizzamento per pescare la memoria che contiene il simbolo interessato.

Il contenuto di tale memoria verrà assegnato

ad un buffer di stampa che, raggiunti i 16 caratteri, stamperà la riga. Il problema di una matrice con  $n$  maggiore di 16 (visto che la CE-122 è capace di sole 16 colonne) è risolto vincolando  $n$  ad un multiplo di 16, in modo tale da ottenere un grafico di qualunque dimensione si voglia, purché si abbia la pazienza di incollare le strisce di carta che di seguito vengono stampate, una accanto all'altra. È come se, di tutta la matrice, la stampante ci fornisce di volta in volta un pacchetto di colonne da assemblare poi in ordine.

La prima label individua quella parte del programma in cui si azzerano tutte le memorie. Seguono una serie di input in cui si richiedono i dati su cui lavorare; alcuni di questi vengono assegnati automaticamente, nel caso l'input manchi. Per esempio si richiedono il massimo ed il minimo della funzione: se entrambi non vengono assegnati, alla fine dell'input viene attivata una subroutine per il loro calcolo e assegnazione. Così per l'incremento della  $y$  e che, salvo diversa indicazione, viene assegnato con un valore tale, rispetto all'incremento della  $x$ , da rendere il grafico abbastanza proporzionato, nonostante la difformità di misura tra un certo numero di colonne e lo stesso numero di righe.

Analogo discorso quando chiede il numero di righe da stampare: se non gli si dice niente le assegna sulla base del numero di nastri che chiede subito dopo, con lo stesso principio seguito per l'incremento della  $y$ .

Terminato l'input esegue una subroutine per stampare i dati introdotti e non, lasciando così una traccia per la lettura del grafico e per eventuali future necessità. La lunghezza di tale primo nastro è calcolata sulla base del numero di righe che verranno poi stampate come grafico, in modo che anche questo possa essere incollato a fianco dei successivi.

Il cuore del programma è una serie di 3 loop a nido, di cui il primo controlla il numero di nastri da stampare, il secondo, più interno, il numero di righe, il terzo il buffer di stampa. Più precisamente il primo, per ogni nastro, reinizializza il valore della  $y$  e adegua il valore della  $x$  alla posizione che avrà alla prima colonna del nastro successivo. Il secondo reinizializza il valore della  $x$ , lo porta cioè a capo riga, esegue la stampa e incrementa il valore della  $y$  per ogni riga che sarà elaborata. Il terzo attiva la subroutine per il calcolo della funzione, incrementa il valore della  $x$  per ognuna delle 16 colonne di ogni nastro, assegna automaticamente il carattere "-" se il valore minimo della  $f(x,y)$  è maggiore di quello effettivamente calcolato in quel punto, oppure il carattere "+" se il valore massimo della  $f(x,y)$  è minore di quello effettivamente calcolato; questo serve ad indicare che i valori della funzione in quei punti sconfinano oltre i massimi e minimi assegnati. Ciò può accadere sia perché il massimo ed il minimo possiamo assegnarlo noi in modo non adeguato, sia attraverso il calcolo automatico, poiché la subroutine che svolge questo compito è concepita in modo da testare i punti della matrice uno ogni quattro colonne e uno ogni quattro righe, per abbreviare i tempi di elaborazione altrimenti esasperanti. Cosicché in questi intervalli non considerati, succede che la funzione potrebbe assumere valori più alti o più bassi di quelli previsti.

La linea 230 del programma merita un po' d'attenzione, poiché è quella che permette di trasformare il valore di  $Z$  in un numero intero compreso in un certo intervallo, che nel nostro caso va da  $36 + 36 \cdot Q$ , in cui  $Q$  è il numero di caratteri previsti nel grafico (cioè il numero di fettine in cui verrà diviso l'intervallo compreso tra il massimo ed il minimo) e 36 è il numero che rappresenta la memoria da cui iniziano ad essere depositati i caratteri.

In sintesi, dunque: si calcola  $Z$ , la si trasforma in un numero che ci guida opportunamente alla memoria in cui è depositato il carattere, segue l'assegnazione del contenuto di questa memoria al buffer di stampa.

Tornando alla linea che permette la trasformazione lineare, si osserva che si tratta dell'equazione della retta  $y=ax+b$  in cui il primo rapporto fa variare  $Z$  sempre tra zero e uno, il coefficiente angolare serve a determinare la compressione o lo stiramento della  $y$ , il termine noto "solleva" tutto a cominciare da un certo valore:

$$W = \frac{Z_i - Z_0}{Z_n - Z_0} (B - A) + A$$

$Z_i$  = valore osservato

$Z_0$  = valore minimo

$Z_n$  = valore massimo

$A$  = estremo inferiore dell'intervallo

$B$  = estremo superiore dell'intervallo

Il programma così com'è non esaurisce ovviamente la possibilità di migliorarlo sotto molti aspetti, anche se ci si scontra con la capacità di memoria del PC-1211, tuttavia superabile se si

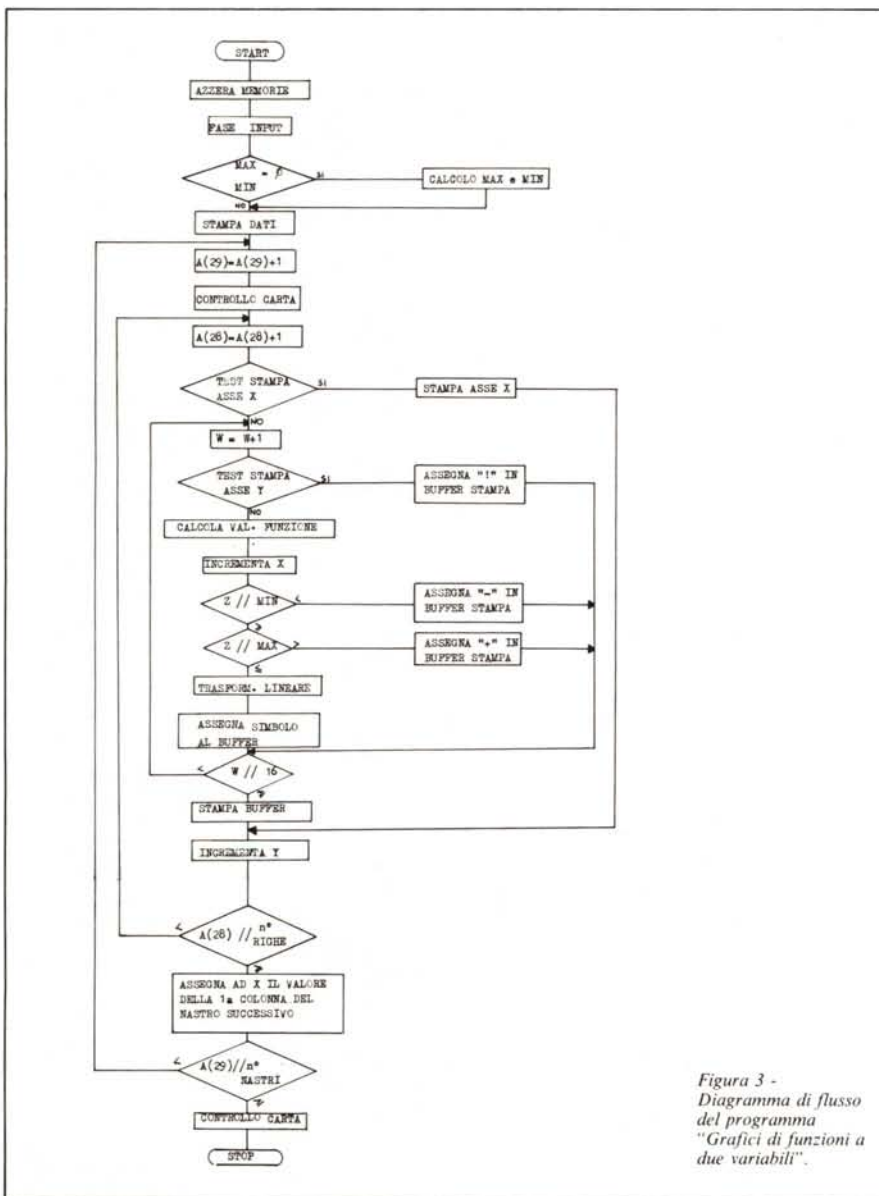


Figura 3 - Diagramma di flusso del programma "Grafici di funzioni a due variabili".

ricorre all'uso di istruzioni CHAIN.

Si potrebbe, ad esempio, ampliare il numero dei dati stampati prima del grafico vero e proprio, per poter poi leggere meglio il grafico, e qualunque altra variante che si adatti meglio alle esigenze dell'utilizzatore. Ciò comunque che si voleva dimostrare è la possibilità grafica che può spaziare al di là del singolo nastro e che può rivelarsi utile in parecchi campi di applicazione.

Un fattore che però appesantisce notevolmente l'utilizzo pratico è il tempo necessario ad un'intera elaborazione, tempo che varia a seconda della complessità della funzione richiesta e soprattutto dalle dimensioni del grafico richiesto (nel caso di 4 nastri, il programma calcola ben 2394 volte il valore della funzione inserita alla riga 500). Problema parzialmente risolvibile facendo lavorare la calcolatrice di notte, dato che una volta inseriti i dati iniziali, tutto procede automaticamente fino alla fine (anche lo spegnimento), senza necessità di controlli o interventi.

### Esempio

Si abbia un obiettivo da 50 mm di focale, e si voglia ottenere un grafico che rappresenti il rapporto percentuale tra la minima distanza per la quale un'immagine è a fuoco e quella per cui è stato regolato, facendo variare sia l'apertura dell'obiettivo (da 1.8 a 16), sia la distanza tra obiettivo e immagine (da 1 a 40 metri). La relazione finale sarà la seguente:

$$Z = \frac{10000}{100 + xy}$$

in cui Z = rapporto percentuale richiesto  
x = apertura diaframma  
y = distanza (in metri)

Nell'intervallo osservato, il nostro rapporto varierà sempre tra zero e cento (più esattamente fra 13 e 98).

Allora per prima cosa inseriamo la nostra funzione alla riga 500 e passiamo al modo DEF richiamando la label A. Alla richiesta di max o min di F(x,y), premiamo direttamente enter senza inserire nulla; in questo modo i due valori verranno calcolati automaticamente. Alla richiesta del valore iniziale di x inseriamo 1.8 mentre l'incremento di x sarà (16-1.8)/(16x4) cioè l'intervallo dei possibili diaframmi, diviso per il numero di colonne della nostra matrice. Il

valore iniziale di y invece sarà 1, come pure il suo incremento. Non daremo nessuna indicazione per quanto riguarda le righe, mentre per i nastri andrà bene 4. Il PC-1211 ci chiederà quindi quanti caratteri, (possiamo darne fino a 20 in questo caso), risponderemo 19. Seguirà l'elenco dei caratteri, a questo punto, per avere un grafico chiaro, con fasce ben distinte, ci converrà alternare ogni carattere con uno spazio (blank), cioè: # = (SPC) ENTER, # = A ENTER, # = (SPC) ENTER, # = B ENTER...

Alla fine il PC-1211 chiederà su quale riga vogliamo l'asse x: rispondendo A(32) che è la memoria che contiene il numero di righe da stampare, l'asse x verrà posizionato come ultima riga. L'asse y invece lo posizioneremo sulla colonna 1.

Così con quest'ultimo ENTER partirà l'elaborazione; dopo circa 5 minuti verrà stampata la prima striscia con i dati di identificazione, e quindi ci vorranno altre 3 ore e 40 minuti perché venga stampato tutto il grafico.

Per la lettura del grafico, ricordiamo che i primi caratteri inseriti (SPC, A, SPC, B...) sono quelli che rappresentano i valori più bassi e visto che tra massimo (98) e minimo (15) ci sono 19 fasce, sarà abbastanza intuitivo capirne la logica. Il risultato è visibile in figura 1.

### Rappresentazione di dati numerici

di Domenico Finucci -  
Sesto S. Giovanni (MI)

Spesso risulta di notevole interesse la possibilità di ottenere rappresentazioni grafiche di dati numerici, non direttamente correlabili tramite formule matematiche. Il vantaggio di una rappresentazione grafica rispetto ad una numerica è, come ormai noto, una maggiore immediatezza di percezione, oltre ad un'agevole confrontabilità fra diagrammi diversi.

Il programma presenta le seguenti caratteristiche salienti:

— consente di avere un limite inferiore dei dati diverso da zero (valore immagazzinato nella memoria W)

— consente di ottenere la sommatoria dei

dati, ed il periodo al quale si riferiscono. Segue quindi una routine di stampa per l'instestazione del grafico e l'asse delle ordinate. Per l'introduzione dei dati seguenti il primo, sul display apparirà:

OUT = 11, Y AVANTI

a significare che il numero che verrà introdotto sarà considerato come dato se diverso da 11, e darà inizio ai calcoli conclusivi se uguale a 11. A questo punto, se fosse necessario introdurre proprio il numero 11 come dato, bisognerà digitare 10.999 o 11.001 per evitare che venga considerato come il simbolo di fine lista.



Figura 6 - Output relativo all'esempio citato nel testo.

### COSA APPARIRÀ

Per rimanere in tema di disegni e rappresentazioni grafiche con il PC-1211, provate a far girare il seguente programma:

```
10:PRINT "
***:"
20:PRINT "S *
* S "
30:PRINT "H **
^ ^** H "
40:PRINT "A **
<> ** A "
50:PRINT "R *
== * R "
60:PRINT "P
** P "
70:PRINT " "
80:PRINT " P
C-1211 "
```

L'unica accortezza è nella fase di scrittura del programma in memoria: gli spazi vuoti (blank) che notate nel listing, sono fondamentali per la riuscita del disegno.

```
10:"GRAF.N": INPUT "MAX=" USING "##### 210:PRINT A#;B#;
GGTO 80 "M,"MIN=" " #;MINM
11:B#="*": W:P=14/(M-W) 130:F=0:USING : 220:J=J+1:INPUT
RETURN 90:INPUT "INIZI PRINT "+---- " "OUT=11, #AV
21:B#="*": Q=":C,"FINE ANTI = "Y:
RETURN = "D 140:J=0:INPUT "D IF Y<>11GGTO
31:B#="*": 100:PRINT " " # ATQ="Y 150
RETURN PRINT "INTER 150:F=F+Y:IF <Y< 230:PRINT "+----
41:B#="*": VALLO:" V) +(Y)M) +-----+
RETURN PRINT " "S C PRINT " " # #F=F/J
51:B#="*": " - "S D GGTO 230 240:PRINT "SOMMA
RETURN 110:PRINT " "N# 160:Y=INT (P*(Y- = "I:USING "#
61:B#="*": (M-W)/2;O=C (M) 170:W+5)+T ###.##^#F:
RETURN -W)/14:PRINT "OGNI DIVISI USING "####
71:B#="*": "ONE:"S:PRINT "MEDIA=" #
80:PRINT "RAPPR USING "##### 180:IF Y<1THEN 2 USING "####
:PRINT " " #;#;Q: 190:IF Y>7LET A# "###"Y "END"=END
:PRINT " " 120:PRINT USING "#####":Y =Y-7
GRAFICA": "#####":W: GOSUB X
```

Figura 5 - Listing del programma "Rappresentazioni grafiche".

numeri introdotti ed il valor medio degli stessi alla fine della rappresentazione.

La stampa della sommatoria è eseguita, per semplicità di lettura, in forma esponenziale; qualora comunque si desiderasse conoscere il valore esatto, esso risiede nella memoria F.

Dopo aver dato il RUN, il PC-1211 chiede l'introduzione dei valori massimo e minimo dei

Il grafico esemplificativo di figura 6 riporta i seguenti dati:

Periodo: 1979-1982

Val. max.: 2000

Val min.: 260

Dati inseriti: 500 1000 1250 1500

1750 1000 260 2000 750 905 110

1600 310



## 64 K bytes in più con sole 740.000 lire.

Se già hai un CBM 8032 tutto quello che ti serve per arrivare a 96 K bytes è una scheda 64 K bytes.

Il costo? Solo 740.000\* lire. Se invece non hai CBM 8032 prendi in considerazione il CBM 8096: significa 96 K bytes a meno di 3.000.000!\*

In ogni caso procurati un Commodore Operating System LOS:

IVA esclusa

così aggiungi 15 comandi Basic e tanti dispositivi per sviluppare programmi.

Va da sé che il LOS permette di utilizzare il nuovo potentissimo Visicalc\* 8096.

Perché aspettare? Spedisci il tagliando alla Commodore Italiana, il Rivenditore più vicino ti contatterà subito.

\*Trade mark della Visicorp

Spedire a Commodore Italiana Srl,  
via Conservatorio 22 - 20122 Milano  
tel. 74.91.126

MC

Richiesta di informazioni

nome \_\_\_\_\_

indirizzo \_\_\_\_\_

**commodore**  
COMPUTER