

software

CASIO FX-702P

Questo mese la rubrica di software dedicata ai pocket Casio ospita un programma che probabilmente dal titolo desterà la delusione di qualche lettore, almeno finché non verrà provato sul pocket FX-702P: il Master Mind.

Ovviamente si tratta di una versione curata e particolareggiata del popolare gioco, implementato su uno dei pocket di maggior successo. Riteniamo comunque che sia di grande utilità seguire passo passo l'analisi di sviluppo del programma ed i metodi di risparmio di memoria adottati da Ravioli.

Il secondo programma rappresenta un'applicazione puramente scientifica, che mette in risalto la precisione e la velocità di calcolo del pocket FX-702P.

Master mind

di Guido Ravioli - Milano

Il programma, per il Casio FX-702P, propone il non certo nuovo gioco del "Master Mind": si tratta cioè di dedurre il numero dell'avversario grazie ad opportune "informazioni". Qui però, contrariamente al solito, il *calcolatore ha anche un ruolo attivo*. Cioè non si limita a "rispondere", ma cerca anche di dedurre il numero pensato dal giocatore umano, tramutandosi in un vero e proprio avversario.

Si è però dovuto semplificare notevolmente il gioco per ovvi motivi di capacità di memoria e di velocità (ricordiamoci che

MASTER MIND2	46
IN 2:1/10:2/TU:3/ST?	++ 1/0: 0(3)
1	
TESTA/CROCE?	2 5(3) +?
TESTA	1
10:0/TU:1?	0?
1	0
?	?
23	47
++ 0/0: 0(1)	++ 1/0: 0(4)
1 5(1) +?	2 3(4) +?
1	0
0?	0?
0	0
?	?
45	48
++ 1/0: 0(2)	++ 1/0: 0(5)
1 0(2) +?	4 5(5) +?
0	2
0?	HO VINTO!
0	IN 2:1/10:2/TU:3/ST?
?	5

Figura 1 - Esempio del programma Master Mind.

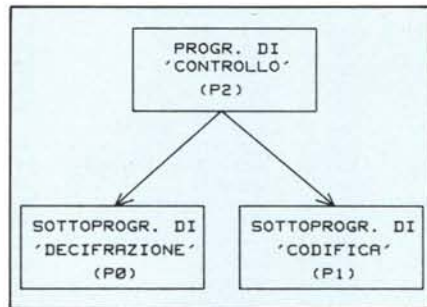


Figura 2 - Struttura fondamentale del programma Master Mind.

siamo di fronte ad un pocket-computer!), riducendo a sole 2 cifre il numero da indovinare; in compenso si è ottenuto un gioco sufficientemente veloce ed "interattivo".

Regole del gioco

Si tratta di indovinare un numero di 2 cifre, diverse fra loro. Attenzione! Vale anche avere uno "0" come prima cifra.

Esempio: numeri ammessi: 17,23,07...
num. non ammessi: 55,44,00...

Uso del programma

Il programma richiede l'espansione a 46 variabili: bisogna quindi azionare il comando "DEFM 2". Occupa 3 zone di memoria (P0,P1,P2) nonché praticamente tutti i passi di programma disponibili (è lungo 1516 step su 1520 disponibili con 46 variabili).

Si fa partire il programma con "F1 P2"; dopo l'intestazione appare il "menu" principale che offre 3 opzioni di gioco (vedi anche Fig. 1):

— "IN 2" (tasto "I"): è l'opzione più interessante, che permette di giocare in due con il computer come avversario.

— "IO" (tasto "2"): è il computer, da solo, che cerca di indovinare un numero pensato da voi.

— "TU" (tasto "3"): siete voi che dovete cercare di indovinare un numero scelto a caso dal computer.

— "ST" (STOP) permette inoltre di uscire dal programma utilizzando un qualsiasi tasto che non sia 1, 2, 3.

Parleremo dettagliatamente solo della prima opzione, essendo questa la più interessante e la più "completa". Premendo "I EXE" si giocherà brevemente a testa o

croce per decidere, se lo si vuole, chi deve iniziare per primo (il computer dirà, imparzialmente, che cosa è "uscito"). Dopodiché apparirà la scritta "IO (computer)/TU (umano)?" per scegliere chi deve iniziare per primo. Supponiamo che inizi prima l'umano (con "I EXE"): bisognerà inserire il numero che si vuole "tentare" (c'è il punto interrogativo) seguito naturalmente da "EXE". Seguirà l'esito del tentativo: "+" significa cifra indovinata in posizione corretta; "0" cifra indovinata, ma posizione sbagliata. Segue il numero di cifre indovinate rispettivamente (0, 1, 2). Fra parentesi c'è il numero di tentativi eseguiti. Premendo "CONT" facciamo giocare il computer, il quale ci mostrerà dopo qualche secondo al massimo, il suo tentativo seguito dagli input "+" e "0": naturalmente dovremo inserire i numeri "0", "1", "2" a seconda dei casi (nessuna cifra indovinata, 1 o 2 cifre indovinate). Dopodiché la "mano" torna a voi e così via fino a quando uno dei due giocatori riesce ad indovinare il numero dell'avversario: dopo aver segnalato la cosa, si ritorna al menu principale. Eventuali errori di input o distrazioni da parte del giocatore vengono segnalate tramite il messaggio "DATI ERR.": schiacciando "CONT" si continua a giocare normalmente.

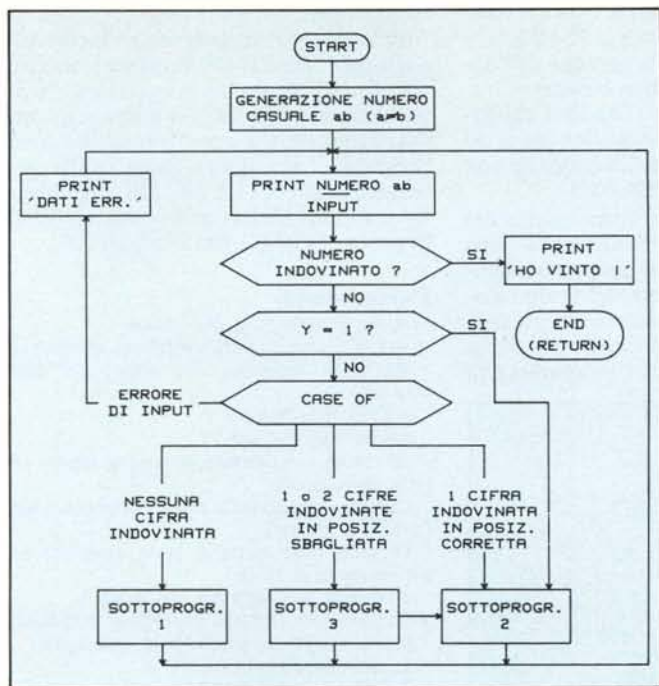
Descrizione del programma

Il programma è diviso in 3 "blocchi" principali (vedi Fig. 2): c'è un programma principale di "controllo" (nella zona di memoria P2) che gestisce opportunamente i programmi di "decifrazione" (in P0) e di "codifica" (in P1). Questi due sottoprogrammi si occupano, rispettivamente, di dedurre il numero del giocatore ("decifrazione") e di formare casualmente un numero corretto dando poi le opportune risposte al giocatore ("codifica").

Il programma di controllo e quello di codifica non presentano particolare interesse: diremo solo che il programma di controllo "pilota" i due sottoprogrammi, secondo la modalità di gioco prescelta, mediante variabili "logiche" (o meglio "flag").

Parleremo in dettaglio solo del programma di decifrazione. Lo schema fondamentale è quello di figura 3: cioè a seconda dei casi possibili (nessuna cifra indovinata, 1 o 2 cifre indovinate, ma in posizione sbagliata, 1 cifra indovinata in posizione corretta) si viene indirizzati ad un sottoprogramma opportuno che genera il nuovo tentativo in base ai tentativi precedenti. Inoltre il "flag" Y serve, come vedremo meglio fra poco, per "incanalare" comunque il flusso verso il sottoprogramma 2.

L'algoritmo di risoluzione è il seguente: se non è stata indovinata nessuna cifra il sottoprogramma 1 provvede a mettere in una lista a (prima cifra) e b (seconda cifra) ed a generare un nuovo numero con cifre differenti non contenute nella lista. La lista è un vettore A(I) lungo 8 cifre al massimo (10 cifre possibili meno le 2 da indovinare):



visto che uso $A(0)=A0$ e $A(1)=A1$ per a e b il vettore andrà da $A(2)$ e $A(9)$.
Se ho indovinato 1 o 2 cifre ma in posizione sbagliata, il sottoprogramma 3 scambia la posizione di a e di b. Se avevo indovinato 2 cifre ho sicuramente vinto

(visto le regole del gioco). Se avevo indovinato solo 1 cifra passo direttamente al sottoprogramma 2, visto che a questo punto avrò sicuramente 1 cifra indovinata in posizione corretta.

Se ho indovinato una cifra in posizione

corretta passo al sottoprogramma 2 che è la parte più "intelligente" del programma di decifrazione.

Questo si preoccupa sostanzialmente di fare degli "esperimenti" in base ai quali dedurre il numero da indovinare. Più precisamente prova a generare una nuova cifra b', conservando però la vecchia cifra b e poi verifica il risultato ottenuto (si ritorna comunque al sottoprogramma 2 mettendo ad "1" il flag Y).

I casi possibili sono i seguenti (vedi Fig. 4):

- nessuna cifra indovinata (sottoprogramma 2₁): scarto a e b (mettendoli nella lista "nera"); evidentemente la cifra indovinata prima era la vecchia b: genero una nuova a e metto la vecchia cifra b nel numero da tentare (pongo $Y=0$).

- 1 cifra indovinata (sottoprogramma 2₃): deve essere per forza la nuova b. Inoltre la vecchia b deve essere "+": genero il numero $a'b_{vecchio}$ dove $a'=b_{nuovo}$ (ho trovato il numero; pongo $Y=0$).

- 1 cifra indovinata in posizione corretta (sottoprogramma 2₂): si ripete il procedimento. Le "vecchie" cifre b (già usate) le metto in un'altra lista (diversa dalla lista "nera") ottenuta sempre con $A(I)$. Questa volta si andrà da $A(10)=B0$ fino a $A(18)=B8$. Pongo $Y=1$. Si vede che, iterando, si arriva alla soluzione.

Analisi del programma (vedi listato in Fig. 5)

Area di memoria PO: sottoprogramma di "decifrazione":

righe 5-200: programma principale: generaz. num. casuale iniziale (10-70), stampa numero (80), input (90 e 110), decodifica input e chiamata ai vari sottoprogram. (100 e 150-180), flag per "collegamenti" con il

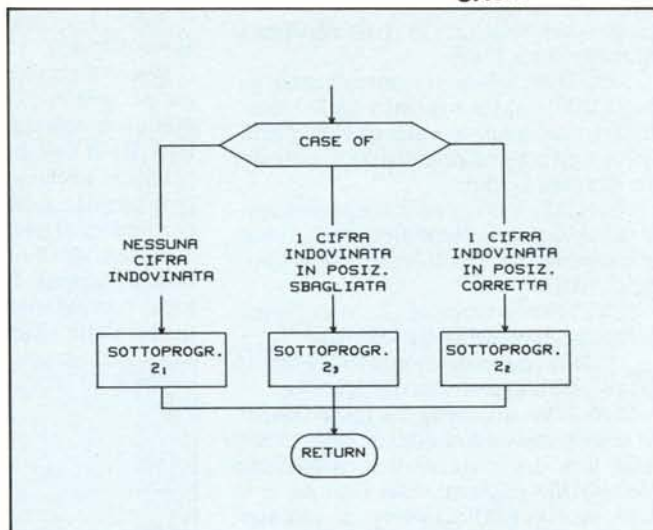


Figura 4 - Schema a blocchi del sottoprogramma 2.

Figura 3 - Schema a blocchi del sottoprogramma di decifrazione.

```

*** PRG LIST
VAR: 46 PRG: 1520
P0: 966 STEPS
5 IF T=0 THEN 150
10 I=2:T=1
50 A(0)=INT (RAN#10)
60 A(1)=INT (RAN#10)
70 IF A(1)=A(0) TH EN 60
80 PRT A(0):A(1):C SR 4:("T");("Y");("SR 10:");("Y");
90 INP N
100 IF N=2:WAIT 40: PRT "HO VINTO!":GOTO 190
110 INP "0";M
115 IF H=1:RET
150 IF Y=1:GSB 2000:GOTO 80
160 IF M=0:IF N=0:G SB 1000:GOTO 80
170 IF N=1:IF M=0:G SB 2000:GOTO 80
180 IF N=0:IF M=1:I F M=2:GSB 3000:GOTO 80
185 GSB 9000:GOTO 8 0
190 IF H=1:R=10
200 RET
1000 IF I>9:GSB 9000:GOTO 1120
1005 A(1)=A(0):I=I+1:A(I)=A(1):K=I
1010 FOR Z=0 TO 1
1020 FOR S=0 TO 9
1030 FOR I=2 TO K
1040 IF S=A(I) THEN 1000
1050 NEXT I
1060 FOR J=1 TO L
2610 IF A(1)=A(9+J) THEN 2640
2620 NEXT J
2630 L=L+1:A(9+L)=A(1)
2640 FOR S=0 TO 9
2650 IF S=A(0) THEN 2710
2660 FOR F=2 TO 9+L
2670 IF F<10 THEN 2690
2680 IF S=A(F) THEN 2710
2690 NEXT F
2700 A(1)=S+S=9
2710 NEXT S
2740 Y=1:T=T+1
2750 RET
2900 A(0)=A(1):A(1)=A(9+L)
2910 Y=0:T=T+1
2920 RET
3000 W=A(0):X=A(1)
3010 A(0)=X:A(1)=W
3015 IF M=1:GSB 2600:GOTO 3030
3020 T=T+1
3030 RET
9000 PRT "DATI ERR.":RET
P1: 278 STEPS
5 IF G=0 THEN 50
20 0=INT (RAN#10)
30 U=INT (RAN#10)
40 IF U=0 THEN 30
50 INP A
60 A=FRAC (A/100)*100
70 P=INT (A/10)
80 A=A-P*10
90 0=INT (A)
100 C=0
110 IF 0=P:C=C+1
120 IF U=0:C=C+1
130 IF C=2 THEN 200
140 D=0
150 IF 0=0:D=D+1
160 IF U=P:D=D+1
170 G=0+1
180 WAIT 9999:PRT "+":("C");("P");("D");("Y");("G");("Y");
185 IF H=1:RET
190 GOTO 50
200 WAIT 40:PRT "HA I VINTO!";("G+1");("Y")
210 IF H=1:R=10
220 RET
P2: 272 STEPS
10 WAIT 30:PRT "***MASTER MIND**"
100 INP "IN 2:1/10: 2/TU:3/ST";H
120 IF H=1:VAC :H=1:GSB 500:GOTO 100
130 IF H=2:VAC :H=2:GSB 80:GOTO 100
140 IF H=3:VAC :H=3:GSB #1:GOTO 100
150 END
500 PRT "TESTA/CROC E?"
510 IF INT (RAN#10)>=5:PRT "TESTA":GOTO 530
520 PRT "CROCE"
530 INP "10:0/TU:1"
550 IF B9=0:R=1
560 IF B9=1:R=0
570 GSB #B9
580 IF R=10 THEN 620
590 GSB #R
600 IF R=10 THEN 620
610 GOTO 570
620 RET

```

Figura 5 - Listato del programma Master Mind.

prog. di controllo (5, 115, 190), rilevamento errori input (185).

1000-1120: sottoprog. I: eventuali errori input (1000), aggiornamento lista "nera" (1005), ricerca nuovo num. corretto (1010-1090), aggiorn. indice lista nera e contatore dei tentativi (1100).

2000-2920: sottoprog. 2: corpo principale sottoprog. con decodifica input e chiamata sottoprog. (2000-2020), rilev. errori input (2025).

2300-2400: sottoprog. 2: rilev. errori (2300), aggiorn. lista nera (2305), $b = b_{precedente}$ (2310), scelta di un nuovo a corretto (2320-2380), aggiornam. I,T,Y (2390).

2600-2750: sottoprog. 2: inizializzazione (2600), controllo esistenza della cifra b nella lista dei b già usati in precedenza (2605-2620), aggiorn. della lista dei b in caso negativo (2630), generaz. di una nuova cifra b con controllo di correttezza (2640-2700), aggiorn. T e flag Y = I (2740).

2900-2920: sottoprog. 2: $a = b_{attuale}$ e $b_{nuovo} = b_{prec.}$ (2900), aggiorn. Y e T (2910).

3000-3030: sottoprog. 3: scambio con b (3000-3010), vado al sottoprog. 2 nel caso di una sola cifra indovinata (3015), altrimenti aggiorn. T ed esco (3020-3030).

9000: messaggio in caso di input errati. Area di memoria P1: sottoprog. di "codifica":

flag di "controllo" (5,185,210), scelta casuale numero (20-40), input (50), decodifica input (60-90), analisi input (100-160), agg. contatore tentativi G (170), stampa risultati ottenuti (180 e 200).

Area di memoria P2: prog. principale di controllo. Intestazione (10), stampa menu

principale (100), decodifica input e chiamata sottoprog. principali (120-150).

500-610: sottoprog. per la gestione dell'opzione di gioco "1" (contro il calc.):

gioco "testa o croce?" (500-520), chi inizia per primo (530), decodifica input ed opportuno intercalare dei 2 sottoprog. con flag (R) di fine gioco (550-620).

Grazie anche alla struttura ridotta del programma rispetto al "master mind" vero (3 o 4 cifre), al posto di un'unica, estenuante partita, come nel master mind tradizionale, si potranno fare velocemente più partite in successione (un "torneo") favoriti in questo dalla struttura del programma (il

tutto magari... sotto il banco di scuola!).

Ritengo che il programma sia facilmente adattabile alla PB-100 (espansa). Inoltre, vista la modularità del programma, chi ha problemi di memoria può sempre caricare solo il sottoprogramma di decodifica come programma a sé stante. Basta infatti cancellare le righe 5, 115, 190, 200, aggiungere VAC all'inizio della riga 10 e mettere END al posto di GOTO 190 alla riga 100.

Elenco variabili

- A: input sottoprog. di codifica
- A0(A(0)): prima cifra sottoprog. decifraz.(a)
- A1(A(1)): seconda cifra sottoprog. decifraz.(b)
- A2-A9: lista "nera"
- B0-B8: lista b "usati"
- B9: input e indicatore sottoprog. scelto nel prog. di controllo
- C: num. cifre esatte in posiz. corretta nel sottoprog. di codifica
- D: num. cifre esatte in posiz. sbagliata nel sottoprog. di codifica
- F: variab. ausiliaria nel sottoprog. 2
- G: contatore tentativi (sottoprog. codifica)
- H: input opzione gioco (prog. principale)
- I: indice lista "nera"
- J: indice lista b "usati"
- K: indice max. lista nera
- L: indice max. lista dei b usati
- M,N: input sottoprog. decifraz.
- O: prima cifra sottoprog. codifica
- P,Q: prima e seconda cifra tentate (sottoprog. codifica)
- R: flag e indicatore sottoprog. scelto (prog. princ. controllo)
- S: variab. ausiliaria (sottoprog. decifraz.)
- T: contatore tentativi (sottoprog. decifraz.)
- U: seconda cifra (sottoprog. codifica)
- W,X: variab. ausiliarie (sottoprog. 3)
- Y: flag sottoprog. decifraz.
- Z: var. ausiliaria sottoprog. I

```

GRADO EQ.?
4
X? 4?
1
X? 3?
0
X? 2?
4
X? 1?
3
X? 0?
-3
R1= 0.271822931 +I 2
.19841689
R2= 0.271822931 -I 2
.19841689
R3= 0.5559822462
R4=-1.099628108
RUN
GRADO EQ.?
4
X? 4?
1
X? 3?
0
X? 2?
4
X? 1?
3
X? 0?
-1
R1=-1.324717999
R2= 0.662358993 +I
0.562279585
R3= 0.662358993 -I
0.562279585
    
```

Figura 7 - Alcuni esempi del programma Radici di equazioni, per testarne il funzionamento.

```

VAR: 56 PRG: 1440
LIST #0
20 VAC
10 INP "GRADO EQ."
N
30 FOR I=N TO 0 ST
EP -1
40 PRT "X?";I:INP
A(N-2,I)
50 NEXT I
60 GSB 1000
70 IF N=3 THEN 130
80 IF N=4 THEN 180
90 IF A6=0:PRT "R1"
="":B4
100 IF A6=0:PRT "R2"
="":B5:END
110 PRT "R1=":B4:"
+I":A6
120 PRT "R2=":B5:"
-I":A6
130 PRT "R1=":B6
140 IF A6=0:PRT "R2"
="":B4
150 IF A6=0:PRT "R3"
="":B5:END
160 PRT "R2=":B4:"
+I":A6
170 PRT "R3=":B5:"
-I":A6:END
180 IF A6=0:PRT "R1"
="":B4
190 IF A6=0:PRT "R2"
="":B5:GOTO 220
200 PRT "R1=":B4:"
+I":A6
210 PRT "R2=":B5:"
-I":A6
220 IF A5=0:PRT "R3"
="":B6
230 IF A5=0:PRT "R4"
="":B7:END
240 PRT "R3=":B6:"
+I":A5
250 PRT "R4=":B7:"
-I":A5:END
1000 J=N-2
1010 FOR I=0 TO N-1
1020 A(J,I)=A(J,I)/A
(J,N)
1030 NEXT I
1040 IF N=2:GSB 3000
:RET
1050 IF N=3:GSB 2000
:RET
1060 A=C3:B=C2:C=C1:
D=0
1070 Q=B-(C3*A/A/8)
A=B/16)-(3*A*A/A
A/A/256)
1100 B2=Q/2
1110 B1=(Q+Q-4*S)/16
1120 B0=-(R*A/64)
1130 IF R=0 THEN 119
0
1140 IF B1=0 THEN 11
90
1150 A1=B2:A0=B1
1160 GSB 3000
1170 B6=B4
1180 GOTO 1210
1190 U=-1
1200 GSB 2000
1210 K=SQR B6
1220 IF K=0:R=SQR (Q
*Q+4*S):GOTO 12
50
1230 Q=Q+(4*B6)
1240 R=R/(2*K)
1250 L=(Q-R)/2:M=(Q+
R)/2
1260 A1=2*K:A0=L
1270 GSB 3000
1280 B6=B4-(C3/4):B7
=B5-(C3/4):H=A6
1290 A1=-2*K:A0=M
1300 GSB 3000
1310 B5=B5-(C3/4):B4
=B4-(C3/4)
1320 RET
2000 IF B0=0:Z=0:GOT
O 2190
2010 F=ABS B0
2020 FOR I=1 TO 3
2030 T=ABS A(1,I)
2040 IF F<T:F=T
2050 NEXT I
2060 IF B0/0:A=-F:1:
B=0:GOTO 2080
2070 A=0:B=F+1
2080 X=A:Y=B:A7=6S
B 4000:A4=A8
2090 A7=6S 4000:A
5=A8
2100 E=1-6
2110 IF A4=0:Z=A:GOT
O 2190
2120 IF A5=0:Z=B:GOT
O 2190
2130 Z=(X+Y)/2:A7=Z:
GSB 4000:V=A8
2140 IF V=0 THEN 219
0
2150 IF (Y-X)/E THEN
2190
2160 A7=X:GSB 4000
2170 IF A8/V>0:X=Z:6
OTO 2130
2180 Y=Z:GOTO 2130
2190 B6=Z
2200 IF V=-1:RET
2210 A1=B2+Z
2220 A0=B1+B2+Z+Z
2230 GSB 3000
2240 RET
3000 W=A1*A1-4*A8
3010 O=-A1/2
3020 IF 6S W=0:B4=0
:B5=0:A6=0:RET
3030 IF 6S W=1 THEN
3070
3040 P=SQR (-W)/2
3050 A6=P:B4=0:B5=0
3060 RET
3070 P=SQR W/2
3080 A6=0:B4=0+P:B5=
0-P
3090 RET
4000 A8=A7*A7+A7+B2*
A7+A7+B1*A7+B0
4010 RET
    
```

Figura 6 - Listato del programma Radici di equazioni.

Radici di equazioni

di Andrea Quaglia - Isola d'Asti

Questo programma calcola le radici di equazioni di secondo, terzo e quarto grado, compresa la parte immaginaria (quando esiste) della radice trovata.

Le soluzioni sono date nella forma: parte reale +/— I parte immaginaria.

Il programma occupa 1429 passi di memoria (circa 2Kbyte) ai quali vanno aggiunti 240 passi per lo spazio riservato alle variabili.

Descrizione del programma:

- 10-50 input dei dati
- 60 lancio subroutine 1000 per la normalizzazione dei coefficienti
- 70-250 output risultati e fine programma
- 1000-1030 normalizzazione coefficienti
- 1040-1050 salto relativo al grado dell'equazione: se N = 2 sub. 3000, se N = 3 sub. 2000
- 1060-1320 equazione di quarto grado
- 2000-2240 equazione di terzo grado
- 3000-3090 equazione di secondo grado
- 4000-4010 funzione necessaria nel sottoprogramma 2000 per la bisezione dell'intervallo dell'equazione di terzo grado.

Ama il meglio!

32K ROM 80K RAM
Tastiera professionale a 90 tasti
Porte per monitor, TV, joysticks,
floppy disk,
cassette recorder, stampante, giochi.
*Interfaccia stampante parallela
Centronics incorporata*

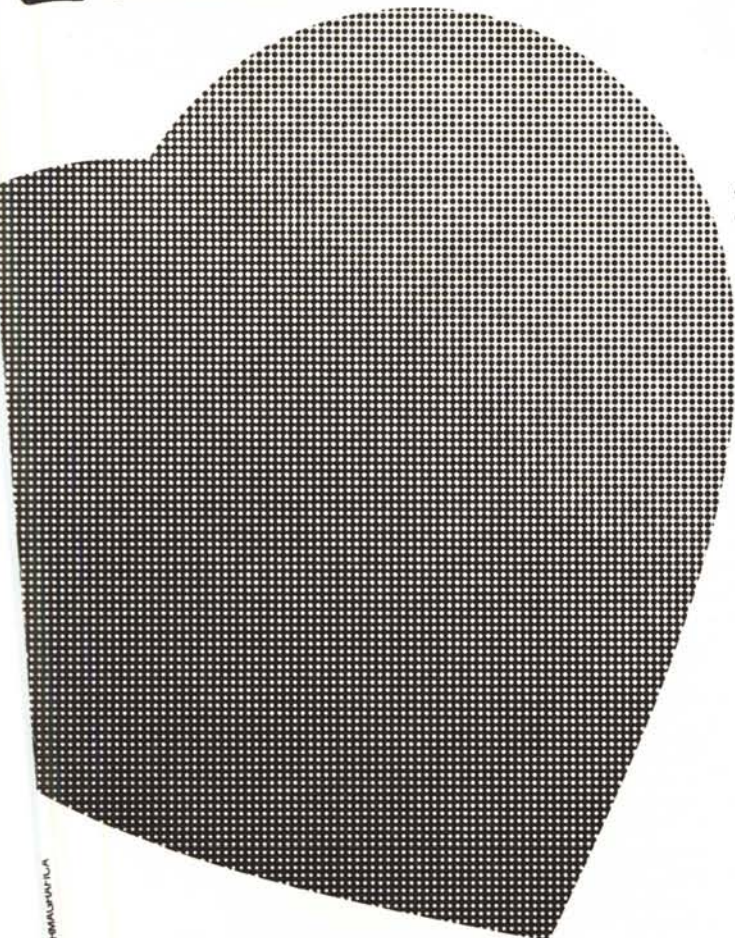
SVITM
SPECTRAVIDEO

il computer del grande standard MSX

Distributore per l'Italia
COMTRAD

Divisione Computers

Tel. (0586) 424348 TLX 623481 COMTRD I



PIRELLA GÖTTSCHE LOWE

