

software

APPLE

Comincia ad arrivare in redazione il software dei lettori sviluppato per l'Apple IIc, segno che questo computer ha raggiunto ormai una discreta diffusione e che, nonostante l'apparenza un po' ermetica è anche lui una macchina «smantellabile». A maggior dimostrazione di cosa sia possibile fare su un IIc (e non sui predecessori) il secondo programma presenta una routine che gestisce un orologio interno in tempo reale, utilizzando una novità del «piccolo mostro» ancora poco conosciuta: gli interrupt.

Erano giunte in redazione molte lettere di affezionati lettori che chiedevano come utilizzare gli interrupt dell'Apple, e, finora, erano rimaste nel cassetto dato che i vecchi Apple II, II+, IIe e IIe/e non possedevano internamente una fonte di interruzioni (come invece possiede ad esempio il Commodore). Nell'Apple IIc le cose sono cambiate in modo drastico e adesso le fonti di interruzioni, interne alla macchina ed esterne, sono diventate persino troppe.

Vedremo, anche con l'aiuto dei lettori più intraprendenti, di sfruttarle al massimo; così come, fino ad ora, sono state sfruttate all'inverosimile le possibilità delle precedenti macchine.

Integrato

Sergio Abeni,
Castellare di Pesca (PT)

All'accensione il programma di HELLO carica in cima alla RAM un programma in linguaggio macchina (INTEGRATO) che integra tre funzioni distinte. Le prime due, l'autonumerazione e l'assegnazione di comandi Basic o D.O.S. ad alcuni tasti premuti insieme al tasto CONTROL, sono praticamente la copia dei rispettivi programmi pubblicati dalla vostra rivista sui nn. 14, 20 (con correzione sul n. 21) e 45 (& Linker) nell'angolo dedicato al software APPLE.

La terza invece è stata da me ottenuta sfruttando le idee contenute nel programma di Valter Di Dio comparso sul n. 20: si tratta di riassegnare alcuni tasti (senza alcun intervento del tasto CONTROL) in modo da ottenere per esempio i due punti premendo il tasto del punto esclamativo (e viceversa). Il vantaggio si rivela particolarmente importante e praticamente indispensabile per la programmazione sull'APPLE II C. In tal modo infatti si può tranquillamente battere un programma tenendo il tasto di SHIFT LOCK sempre inserito.

Con il programma INTEGRATO permanentemente in memoria, si ha a disposizione praticamente una nuova tastiera, quella rappresentata nella figura, che permette una digitazione dei programmi veloce, senza il continuo ricorso al tasto dello SHIFT, abbandonando quella fornitaci dall'APPLE II C che, come si sa, è fatta per lo Word Processing.

Nel caso in cui si volessero riassegnare tasti diversi da quelli già assegnati, basta avviare il programma TASTI, che è stato opportunamente aggiustato, con l'aggiunta di uno scambio di tasti, come spiegato nel programma.

La stessa tastiera (o quasi), a meno dei comandi BASIC o D.O.S., si ha con il programma di figura 3 che si autoinstalla nel D.O.S. (le locazioni libere le ho prese da MC n. 26, un D.O.S. amico).

Ed ora vediamo più da vicino le caratteristiche dei tre programmi, in particolare di INTEGRATO.

Si provi a battere sulla tastiera dell'APPLE II C la seguente linea:

```
100 A = BI:C(1,2) = DO - 1
```

È un continuo ricorso al tasto di SHIFT oppure un continuo inserimento e disinserimento del tasto di SHIFT LOCK. Si capisce che a lungo andare il problema diventa insostenibile. Il

```
*9400.95BC
9400- A9 00 85 73 A9 94 85 74
9408- A9 4C A2 1A A0 94 8D F5
9410- 03 8E F6 03 8C F7 03 4C
9418- 4E 94 AA F0 31 20 B1 00
9420- E0 41 F0 07 C6 88 F0 F2
9428- 4C 4E 94 20 67 DD 20 52
9430- E7 A6 50 A4 51 8E 5C 95
9438- 8C 5D 95 20 BE DE 20 67
9440- DD 20 52 E7 A6 50 A4 51
9448- 8E 5E 95 8C 5F 95 A9 59
9450- 85 38 A9 94 85 39 4C EA
9458- 03 20 1B FD C9 89 F0 3B
9460- C9 8A F0 37 C9 8B F0 33
9468- C9 9B 10 2F A0 11 D9 60
9470- 95 F0 07 88 D0 F8 20 88
9478- 94 60 88 98 0A A8 B9 72
9480- 95 85 9D B9 73 95 85 9E
9488- A0 00 B1 9D C9 80 10 E9
9490- 9D 00 02 E8 20 5C DB C8
9498- 4C 8A 94 A0 10 D9 93 95
94A0- F0 06 88 D0 F8 4C 76 94
94A8- B9 A3 95 4C 76 94 20 89
94B0- FE 20 EA 03 60 20 1B FD
94B8- E0 00 D0 47 C9 A0 D0 43
94C0- AE 5C 95 AC 5D 95 20 39
94C8- 95 A2 01 8E 60 95 86 1E
94D0- CA 86 1F A0 02 B9 17 00
94D8- 46 1E B0 06 E6 1E 4A 4A
94E0- 4A 4A 29 0F C5 1F F0 0B
94E8- C6 1F 09 B0 9D 00 02 E8
94F0- 20 ED FD A5 1E D0 DE 88
94F8- 10 DB A5 1F F0 EA A9 A0
9500- A4 24 60 C9 98 D0 04 4E
9508- 60 95 60 C9 8D D0 29 4E
9510- 60 95 90 24 18 AD 5C 95
9518- 6D 5E 95 8D 5C 95 AD 5D
9520- 95 6D 5F 95 8D 5D 95 C9
9528- FA 90 0B 86 1E 84 1F 20
9530- AE 94 A6 1E A4 1F A9 8D
9538- 60 86 1E 84 1F A2 00 86
9540- 17 86 18 86 19 A0 0F 06
9548- 1E 26 1F F8 A2 FD B5 1A
9550- 75 1A 95 1A E8 30 F7 D8
9558- 88 10 EC 60 64 00 0A 00
9560- 00 80 B1 82 84 85 86 8C
9568- 8E 8F 90 91 92 94 96 97
9570- 99 9A B4 95 D2 A8 C7 D1
9578- F3 D0 A3 D1 D3 D0 D4 D1
9580- D6 D0 93 D1 A4 D1 08 D2
9588- A9 D1 EF D1 3B D2 DE D0
9590- F6 D0 06 D2 FF 89 8A 8B
9598- DF DE A1 DB BE AD BB BA
95A0- A4 AC A5 A5 BD A2 AB A9
95A8- AD BB BA A4 AC DF DE A1
95B0- DB BE A5 A5 43 41 4C 4C
95B8- 2D 31 35 31 A0
```

Figura 1 - Programma integrato. Da \$95B4 inizia la stringa ottenibile con CTRL®. In questo caso la stringa è: CALL-151. Salvare con BSAVEINTEGRATO, A\$9400.L\$200.

programma di figura 1 risolve il problema riassegnando i tasti in modo che i simboli più usati nella programmazione (" = - () \$, ; :), attualmente disponibili senza SHIFT, possano essere ottenuti con lo SHIFT, magari in un'altra posizione, in modo che tutto il listato sia battuto col tasto di SHIFT LOCK inserito. Il risultato è raffigurato nella figura della tastiera, dove sono visibili anche alcuni comandi BASIC o D.O.S. in corrispondenza

di alcuni tasti, comandi ottenibili premendo il tasto contemporaneamente al tasto CONTROL. Quest'ultimo tipo di assegnazione dei tasti è quello già apparso su MC n. 20 (con correzione sul n. 21) al quale si rimanda chi volesse approfondire l'argomento. Il programma contiene anche l'autonumerazione già presentata sul n. 14 di MC e migliorata da Francesco Meschia sul n. 45 di MC. È bene inserire il programma seguendo i consigli dati a

pag. 108 del n. 43. Deve essere salvato con BSAVEINTEGRATO,AS9400,LS200.

Esso va quindi a disporsi nella parte alta della RAM e si autoprottegge. Si può disinserire con IN£0 (o CONTROL RESET) e reinserire con CALL 37888.

È consigliabile porre nel programma di HELLO la linea PRINT CHR\$(4) «BRUN INTEGRATO» che permette l'entrata in funzione automa-

```

80 CT = 38241:IN = 38258:SO = 38292:DE = 38308:ST = 38324
90 HOME : PRINT "PROGRAMMA DI DEFINIZIONE TASTI": PRINT
95 PRINT "PER CAMBIARE I TASTI CON CTRL PREMI C"
100 PRINT "PER SCAMBIARE DUE TASTI PREMI T"
105 PRINT "PER ASSEGNARE LA FRASE PREMI R"
110 PRINT "PER FINIRE PREMI F": GET WR#
113 PRINT
115 IF WR# = "T" THEN 600
120 IF WR# = "R" THEN 450
125 IF WR# = "F" THEN END
130 INPUT "QUALE TASTO?";A$
140 CA = ASC (A$) + 64
150 FOR P1 = CT + 1 TO CT + 16
160 IF PEEK (P1) = (CA) THEN 190
170 NEXT
180 PRINT : PRINT "NON E' DISPONIBILE ": GOTO 130
190 P2 = (P1 - (CT + 1)) * 2
200 INPUT "QUALE ISTRUZIONE VUOI ASSEGNARE ?";I$
210 S1 = 53456:S2 = 53854:S = 0
220 FOR K = S1 TO S2
230 X = PEEK (K)
240 IF X > 128 THEN 310
250 X$ = X$ + CHR$ (X)
260 NEXT
280 IF S THEN 300
290 S = 1:S1 = 43140:S2 = 43370: GOTO 220
300 PRINT : PRINT "NON E' APPLESOFT !": PRINT "NON E' D.O.S. !": PRINT :
    GOTO 200
310 X = X - 128
320 X$ = X$ + CHR$ (X)
330 IF X$ = I$ THEN 360
340 X$ = ""
350 GOTO 260
360 K = K - LEN (X$) + 1
370 KH = INT (K / 256)
380 KL = K - KH * 256
390 POKE IN + 2 + P2,KL
400 POKE IN + 3 + P2,KH
410 PRINT : GOTO 95
450 INPUT "FRASE DA ASSEGNARE? ";F$
460 IF F$ = "" OR LEN (F$) > 75 THEN 520
470 FOR I = 1 TO LEN (F$)
480 POKE ST - 1 + I, ASC ( MID$ (F$,I,1))
490 NEXT
500 POKE ST - 1 + I,160
520 PRINT : GOTO 95
600 PRINT "ORA SI EFFETTUA LO SCAMBIO TRA DUE TASTI"
620 PRINT "SE VUOI ANNULLARE UNO SCAMBIO DEVI PREMERE LO STESSO TASTO SI
A ALLA RICHIESTA TASTO SORGENTE CHE ALLA RICHIESTA TASTO DESTINAZION
E": PRINT : PRINT
650 INPUT "TASTO SORGENTE? ";A$
660 INPUT "TASTO DESTINAZIONE? ";B$
670 CA = ASC (A$) + 128
680 CB = ASC (B$) + 128
690 POKE SO + 15,CA: POKE DE + 15,CB: POKE SO + 14,CB: POKE DE + 14,CA
700 PRINT : GOTO 95

```

Figura 2 - Programma tasti.

BA69-	A9 74	LDA	##74
BA6B-	B5 38	STA	##38
BA6D-	A9 BA	LDA	##BA
BA6F-	B5 39	STA	##39
BA71-	4C EA 03	JMP	##03EA
BA74-	20 1B FD	JSR	##FD1B
BA77-	A0 10	LDY	##10
BA79-	D9 DF BC	CMP	##BCDF, Y
BA7C-	F0 04	BEQ	##BA82
BA7E-	88	DEY	
BA7F-	D0 FB	BNE	##BA79
BA81-	60	RTS	
BA82-	B9 EF BC	LDA	##BCEF, Y
BA85-	60	RTS	
BCE0-	FF B9 BA BB DF DE A1 DB		
BCEB-	BE AD BB BA A4 AC A5 A5		
BCF0-	BD A2 AB A9 AD BB BA A4		
BCFB-	AC DF DE A1 DB BE A5 A5		

Figura 3 - Programma da inserire nelle locazioni libere del DOS per ottenere la nuova tastiera (senza comandi BASIC o DOS). È alternativo a tutti i programmi precedenti.

tica del programma all'accensione del computer.

Chi non fosse contento della tastiera così programmata (ma vi assicuro che è la più razionale) può cambiare i valori nella zona dati \$9561-\$95FE, oppure utilizzare il programma in BASIC «TASTI» (analogo a quello apparso su MC n. 20), vedi fig. 2, che consente anche lo scambio di due tasti.

Per chi fosse interessato soltanto alla riassegnazione tasti senza l'uso del CTRL e senza l'autonumerazione e volesse tenere tutta la RAM libera, è invece adatto, in alternativa, il programma in linguaggio macchina della fig. 3.

Esso, come si nota, va a installarsi all'interno del D.O.S., nelle poche zone lasciate libere dal D.O.S. stesso (vedi MC n. 26 pag. 75). Basta caricare in memoria il D.O.S., passare al Monitor

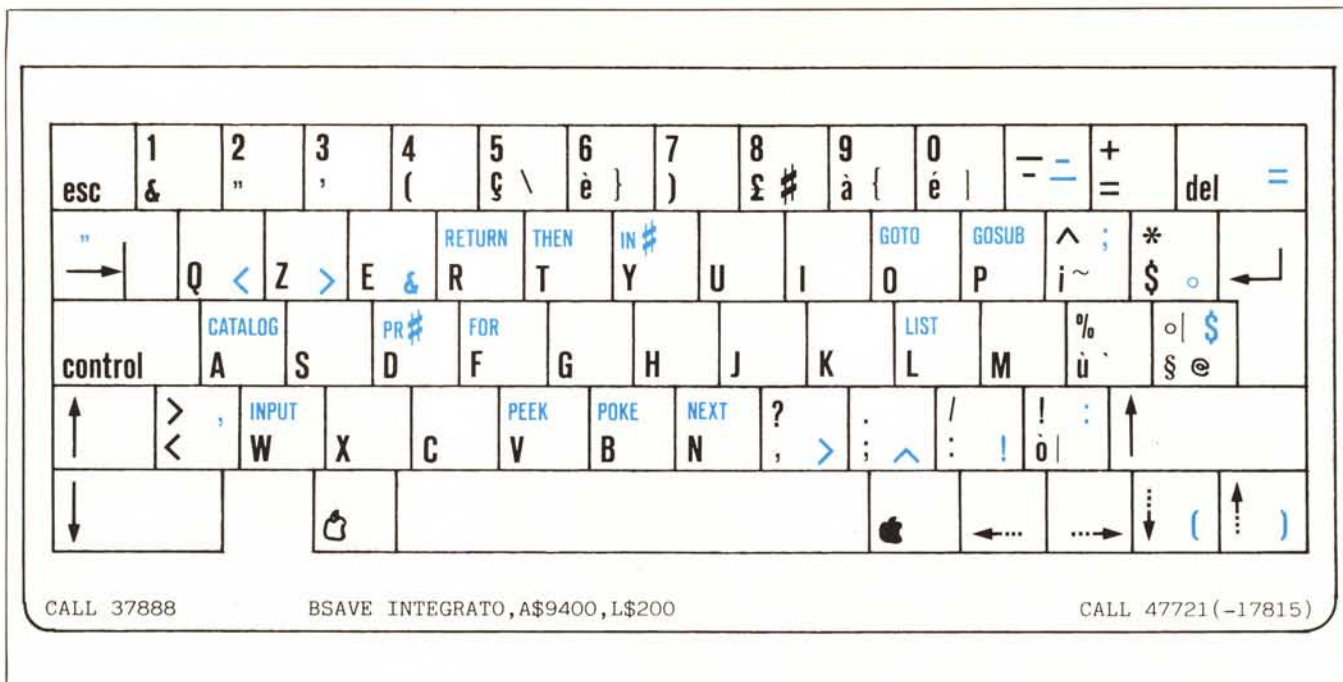


Figura 4 - Nuova tastiera ottenibile col programma integrato. In colore i comandi ottenibili con CTRL e i nuovi tasti scambiati. Per effettuare cambiamenti vedi programma tasti.

col solito CALL -151, battere il programma di figura 3, ritornare al BASIC, battere il seguente programmino di HELLO:

```
10 REM RIASSEGNAZIONE TASTI
20 CALL 47721
30 NEW
```

e poi, dopo aver inserito un disco vergine nel drive, battere INIT HELLO. Da questo momento avrete a disposizione una tastiera analoga a quella di figura 4 (senza i comandi BASIC o D.O.S.). Nel caso in cui la nuova tastiera sia disinserita, cosa che avviene quando viene premuto CTRL RESET, per reinserirla basta digitare CALL 47721.

Orologio in tempo reale

La possibilità di avere un orologio dentro al computer si rivela abbastanza comoda per dare un certo tocco professionale ai programmi applicativi; nei programmi di giochi è indispensabile per misurare i tempi dei concorrenti, infine, nell'uso scientifico, dove permette di acquisire dei dati da linee esterne (tipo pulsanti o paddle) segnalando anche il momento in cui si è verificata l'occorrenza.

Per realizzare un orologio interno è necessario avere una sorgente di clock, ovvero un qualche cosa che ad intervalli precisi generi degli impulsi rilevabili dal computer. Inoltre si deve fare in modo che il funzionamento dell'orologio non interferisca con il

normale svolgimento dei programmi (sia in Basic che in Linguaggio Macchina) e, findove possibile, viceversa.

Nei Commodore 64 la sorgente di clock è un integrato appositamente progettato (in grado oltretutto di modificare a piacere la frequenza degli impulsi), mentre nei vecchi Apple II questo integrato si può installare su una scheda apposita (VIA Versatile Interface); gli impulsi provenienti da questo integrato vanno ad attivare, su un piedino del 6502, una richiesta di interruzione (IRQ). L'interruzione dice al microprocessore di abbandonare momentaneamente quello che sta facendo ed eseguire una routine il cui indirizzo, negli Apple II, si trova in \$3FE e \$3FF (in realtà la storia è un po' più lunga, ma per ora basta questo). A questo punto, per realizzare il nostro orologio, dobbiamo solo trovare una fonte conosciuta, affidabile e costante di interruzioni.

Per la prima volta nella storia degli Apple II questa fonte di interruzioni esiste ed è già pronta all'uso, anche se i progettisti non ne prevedevano certo questa applicazione: si tratta del segnale di Video Blank.

Questo segnale compare per la prima volta nell'Apple IIe e il suo utilizzo permette di sincronizzare le operazioni di scrittura sul video con il momento in cui il pannello elettronico inizia la scansione dello schermo, questo serve per evitare che battimenti tra la frequenza di scrittura e quella di scansione rendano lampeggiante una scritta che non lo è. Provate ad esem-

pio a far girare il seguente programmino:

```
10 HOME : PRINT
20 HTAB 1: PRINT "
30 FOR D = 1 TO PDL (0): NEXT
40 HTAB 1: PRINT "MC MICROCOMPUTER";
50 GOTO 20
```

variando la posizione della paddle (0) la scritta MC MICROCOMPUTER lampeggia in modo bizzarro.

Il controllo del segnale di Video Blank permette di conoscere da programma l'attimo in cui il pannello inizia a disegnare la pagina e, quindi, di attendere questo momento per iniziare il refresh del testo. Un'altra applicazione sfrutta il segnale per ottenere delle finestre video di testo in mezzo alla pagina grafica in HGR, infatti, conoscendo il momento in cui il pannello elettronico inizia a disegnare, si può avviare un programma che attenda un certo numero di millisecondi, poi commuti su TEXT, attenda ancora un po' e ritorni in grafica. Nel periodo tra le due commutazioni appare sul video quello che si trova, nella corrispondente pagina di testo, sotto al punto raggiunto in quel momento dal pannello elettronico.

Sull'Apple IIe il segnale di Video Blank era accessibile tramite una locazione di memoria, per capirci, allo stesso modo dei segnali di pulsante premuto provenienti dalle paddle; nel IIc, invece, questo segnale è stato collegato al sistema degli interrupt (ovviamente mascherabile). E così, ogni

volta che il video inizia un refresh della pagina, si genera un interrupt. Dal momento che la frequenza di quadro dei televisori italiani è di 50 hertz, si verificano, nel microprocessore, 50 interruzioni al secondo. Allora, se scriviamo una routine di interrupt che incrementi un contatore, questo indicherà lo scorrere del tempo in cinquantissimi di secondo.

Perché un contatore in cinquantissimi di secondo possa durare anche un giorno, deve poter arrivare almeno a 4'320'000 (24 ore) corrispondenti a 23 bit, serve quindi una routine di conteggio almeno a tre byte (24 bit), che permette di superare le 93 ore.

Tutto quello che ci occorre allora è un programmino che abiliti le interruzioni da Video Blank, e poi agganci agli interrupt il contatore a tre byte.

Questo è il programmino:

```

0300- A9 10 LDA #10
0302- 8D FE 03 STA $03FE
0305- A9 03 LDA #03
0307- 8D FF 03 STA $03FF
030A- A9 09 LDA #09
030C- 20 3D C4 JSR $C43D
030F- 60 RTS
0310- EE FB 03 INC $03FB
0313- D0 08 BNE $031D
0315- EE FC 03 INC $03FC
0318- D0 03 BNE $031D
031A- EE FD 03 INC $03FD
031D- 40 RTI

```

Adesso, facendo una CALL 768, il programma si attiva; per accorgersene basta far suonare il bip... è strano? Bene, allora la routine è attiva, e ogni cinquantesimo di secondo il microprocessore abbandona quello che sta facendo per andare ad incrementare il nostro contatore.

Naturalmente la routine viene disattivata, momentaneamente, dal DOS durante le operazioni di lettura e scrittura su disco e, definitivamente, dalla pressione del tasto RESET (questo potrebbe essere evitato ritoccando la routine di reset, ma semmai ne ripareremo).

I tre byte scelti per il contatore sono \$3FB, \$3FC e \$3FD che, nei vecchi Apple II, contenevano il salto alla routine di interrupt non mascherabile, non esistente sull'Apple IIc e quindi

inutilizzate. Ora per sapere che ore sono in un formato corretto (e non in cinquantissimi esadecimali) ci basta un semplice programma in Basic che legga le tre locazioni e le converta in ore, minuti e secondi:

```

10 POKE 1019,0: POKE 1020,0: POKE
1021,0
20 CALL 768
30 :
40 DEL 10,40
50 :
100 TI = PEEK (1021) * 65536 + PEEK
(1020) * 256 + PEEK (1019)
110 TI = INT (TI / 50)
120 T1 = INT (TI / 60)
130 SS = TI - T1 * 60
140 HH = INT (T1 / 60)
150 MM = T1 - HH * 60
160 M$ = STR$ (MM): IF LEN (M$)
= 1 THEN M$ = "0" + M$
170 H$ = STR$ (HH): IF LEN (H$)
= 1 THEN H$ = " " + H$
200 VTAB 1: HTAB 25
210 PRINT "ore: "H$": "M$": "SS" "
300 FOR D = 1 TO 200: NEXT
310 GOTO 100

```

Anche questo si potrebbe far fare alla routine di interrupt, e già che ci siamo si potrebbe scrivere direttamente il risultato dentro la variabile TIS, (c'è qualche lettore volenteroso?), ma tutto sommato mi sembra inutile affaticare la routine di interrupt con un simile aggiornamento (che verrebbe eseguito ogni 50' di secondo) per poi magari leggere TIS da Basic ogni due ore!

Due parole ancora sulla routine che attiva gli interrupt video. Questa si chiama XSETMOUSE e si trova nelle nuove ROM che gestiscono il Mouse; la routine permette di attivare i principali tipi di interrupt del IIc ed è perciò molto sofisticata. Per ora basti sapere che si possono attivare interrupt dal video, dalla tastiera, dalle interfacce seriali, dal tasto del mouse e dagli spostamenti orizzontali o verticali del mouse. Questi ultimi due segnali provengono da due pin del connettore GAME e si può scegliere di attivare l'interruzione sia sul fronte di salita (quando il segnale passa da 0 a 5 volt) che sul fronte di discesa (ovviamente da 5 a 0 volt) e questo apre una serie di grosse possibilità all'uso di un Ap-

ple IIc come controllore di processi esterni.

La tabella A mostra il valore da assegnare all'Accumulatore prima del salto alla XSETMOUSE e i relativi interrupt abilitati.

Conclusioni

Per questa volta è tutto, ma torneremo senz'altro sull'argomento interruzioni in maniera più tecnica onde far felici tutti gli smanettoni. E in tale occasione parleremo anche delle due interfacce seriali (anche loro generatrici di interrupt) e dell'uso della AUX memory.

E così, stranamente, la macchina che sembrava la meno indicata alle modifiche dei programmatori folli, rivela capacità di intervento nuove e insospettite.

Inviare i vostri programmi

Alcuni lettori ci chiedono, nelle loro lettere, come sottoporre i loro programmi a MC.

Registrate i vostri lavori su cassetta o disco (se il programma è proprio molto corto può bastare il semplice listato; certo, la cassetta non guasta mai...), corredateli dell'opportuna documentazione e spedite il tutto alla redazione, indicando magari sulla busta la rubrica interessata.

Tutti i programmi che arrivano sono esaminati ed i migliori pubblicati. Purtroppo non possiamo restituire, per ragioni organizzative, il materiale che ci viene inviato, anche in caso di mancata pubblicazione.

Ricordatevi che migliore è la documentazione, maggiore è la possibilità che il vostro lavoro venga pubblicato: spiegate quindi chiaramente il funzionamento del programma ed accludete tutto quello che pensate possa essere utile (elenco variabili e via dicendo). Soprattutto non dimenticate di indicare il computer sul quale il programma gira, né il vostro nome e indirizzo e, se possibile, il numero di telefono. Indicate anche, per la retribuzione se il programma sarà pubblicato, luogo e data di nascita, domicilio fiscale e codice fiscale (partita IVA, se la possedete).

Il compenso per i programmi pubblicati varia normalmente fra le 40 e le 150.000 lire, a seconda della qualità del lavoro inviato; eventuali programmi di particolare complessità ed interesse potranno essere valutati al di fuori di questo standard, previ accordi con la redazione.

Tabella A

Codifica del modo operativo per XSETMOU

Contenuto	Modo operativo
00	Il Mouse è disabilitato.
01	Trasparente. Le interruzioni sono unicamente utilizzate per aggiornare i parametri del Mouse, ma il programma stesso dovrà leggere questi parametri. Questo modo è l'unico disponibile in Basic.
03	Richiesta di interruzione rispetto agli spostamenti.
05	Richiesta di interruzione dal pulsante.
07	Richiesta di interruzione rispetto agli spostamenti o dal pulsante.
80	Il Mouse è disabilitato, VBLINT è abilitato.
09	Trasparente, VBLINT è abilitato.
0B	Richiesta di interruzione rispetto agli spostamenti e da VBLINT.
0D	Richiesta di interruzione dal pulsante e da VBLINT.
0F	Interruzione rispetto agli spostamenti, dal pulsante e da VBLINT.