

Elementi di Prolog

ottava parte

L'uso delle regole

Siamo ormai in possesso dei tool di base per la redazione di un programma in Prolog. Questa puntata entrerà nei dettagli delle «rules», le regole, che sono la vera e propria ossatura del programma. Delle regole avevamo già discusso, per la verità, in diverse occasioni in precedenza, in particolare per quanto riguarda le relazioni intercorrenti tra di loro o con i fatti e le istruzioni vere e proprie

In effetti i programmi finora presentati erano piuttosto rigidi, vale a dire, nonostante l'uso delle variabili, scarsamente adattabili ad una situazione più generale; vedremo in questa puntata come attraverso le regole sia possibile costruire programmi di uso estremamente generico, e, quindi, più elastici ed efficienti nell'uso.

Presupponiamo quindi di conoscere ormai alla perfezione la sintassi specifica del Prolog, e le formalità sintattiche necessarie per definire nuove regole. Sappiamo che la regola possiede una testa ed un corpo vero e proprio, e che questi due interagiscono tra di loro durante l'esecuzione della regola. Nonostante la semplicità e la banalità delle regole finora annunciate, abbiamo visto come sia estremamente semplice (basta definire i fatti) trarre conclusioni da una serie di eventi noti e codificati secondo una serie di regole, peraltro poco rigide.

Inutile, comunque, illudersi; sebbene il fatto che il programma sia capace di trarre conclusioni da una serie di fatti sembri quasi miracoloso, finora quanto abbiamo visto risulta di ben scarsa utilità pratica. Se pensiamo che un famosissimo WP è stato redatto in Prolog ci renderemo conto che l'abisso tra il dire è il fare è ancora immenso. Ma, poiché se mai si comincia, mai si arriva, vediamo come è possibile cominciare a disegnare un programma leggermente più complesso (ma efficiente e utile) di quelli finora visti.

È possibile guardare ad una regola come ad una organizzata gerarchia di istruzioni; senza tema di ridurre le caratteristiche del costruito possiamo dire che una regola è un fatto condizionale (è vero solo se talune condizioni sono accertate). La figura A mostra alcune delle differenze maggiori fra fatti e regole in Prolog.

La forma generale per una regola, in Prolog, è:

A if B.

che può essere letta, in altri termini come

“Vero A se vero B”.

Altre implementazioni ammettono la forma:

A :- B.

Ovviamente la «certezza» di A può dipendere, oltre che da B anche da altre variabili. In questo caso occorre utilizzare statement multipli come nell'esempio seguente:

A if B

A if C

A if D.

Questa serie di ordini è abbastanza simile a quanto visto la volta scorsa relativamente al connettivo OR. In pratica A sarà «vero» se B sarà «vero»; se ciò non avviene viene testata la verità di C e successivamente quella di D. Ovviamente, se non esiste, nel programma, un adeguato sistema di «escape» al reperimento del True, la serie di regole così articolata darà True solo se l'ultima condizione verrà soddisfatta.

Le parti di una regola

Vediamo adesso quale è il look generale di una regola in Turbo Prolog; come abbiamo già accennato, una regola consiste di una testa, in un connettore condizionale, ed un corpo. La regola che vedete in figura B, tradotta in italiano significa: «Una macchina dura se il meccanico esegue i tagliandi». La testa è la prima parte della regola, e rappresenta il primo termine di paragone della bilancia dell'If. In questo caso, per amo-

re della precisione, la testa è rappresentata da [dura(macchina)].

La testa è collegata col corpo, come abbiamo più volte detto, da un connettore condizionale; in altri termini la connessione identifica i confini delle due parti messe a confronto.

Tutto quel che segue la parte condizionale è detto il «corpo» della regola; nel nostro caso esso è

`esegue(meccanico,tagliando).`

Per usare una regola in ambiente Turbo, è sufficiente battere la testa della regola come una domanda; Turbo Prolog legge il programma alla ricerca delle condizioni successive all'If e, se le condizioni sono soddisfatte, restituisce una risposta adeguata.

È possibile altresì utilizzare regole con l'ausilio delle variabili, invece di usarle con specifici argomenti. Così, se noi aggiungiamo alla nostra base di conoscenza la regola:

`dura(volvo).`

il programma risponderà fornendo i nomi di qualunque cosa «dura». Così battendo:

`dura(Chi).`

avremo, come risposta:

`dura(volvo).`

Una applicazione in Prolog destinata a coinvolgere diverse e complesse ricerche e verifiche di condizioni e situazioni deve includere regole sofisticate coinvolgenti molte situazioni sotto le quali alcuni fatti assumono essenza di verità o viceversa. Ci ritorna in aiuto il gruppo di operatori logici [and], [or] e [not], visti in precedenza, che permettono, stavolta, in una regola, di creare sofisticate ricerche nella struttura di database. Niente di difficile nell'interpretare regole del tipo:

`dura(macchina)if
esegue(meccanico,tagliando) and
pressione(giusta,gomme) and not
correre(velocità,massima) or not
usare(olio,scadente).`

La regola «multiclausola» è uno degli approcci più utilizzati in Prolog, in generale, ed in Turbo Prolog in particolare. Ma, per rendere effettivamente partecipe l'operatore nella efficace scelta dei «requisiti» della regola, occorre qui eseguire una serie di precisazioni.

In altri termini, per utilizzare al meglio il processo di analisi intelligente di un programma in Prolog, occorre essere bene a conoscenza di alcune regole di base circa la costruzione di una regola. Vedremo di seguito quali sono i requisiti necessari per la costruzione di una regola efficace e ben scritta.

Come è noto, Turbo Prolog esige che siano definiti i tipi di ogni word utilizzata in qualsiasi parte del programma (fatti,

richieste, regole, ecc.) presente nella sezione Domains del programma. Sebbene sia una precisazione per lo meno banale, uno dei più diffusi errori nel far funzionare i programmi in Prolog è quello della dimenticanza di definizione di uno o più elementi del programma stesso.

Una regola va battuta così come l'abbiamo vista in precedenza, facendo particolare attenzione alla capitalizzazione ed alla punteggiatura (ogni regola va seguita dal punto fermo [.]).

Una regola standard potrebbe essere rappresentata nel costrutto:

`dura(Macchina) if
esegue(meccanico,tagliando) and
riscalda(motore).`

La testa della regola è:

`dura(macchina).`

Stiamo tentando in questa sezione di definire i criteri secondo cui una mac-

fatti

sempre vero, per definizione

ha un solo predicato al di fuori delle parentesi

finisce con un punto [.]

salvo specifiche diverse, "vero" solo per il fatto cui si riferisce

regole

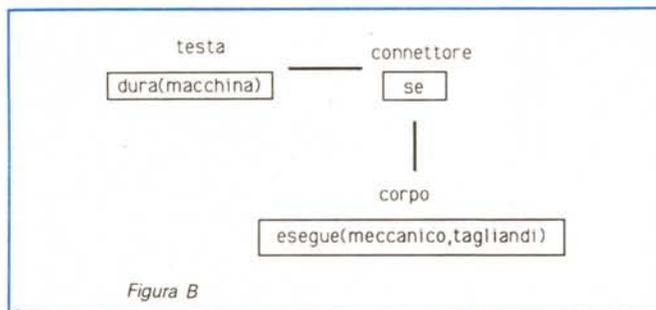
vero solo se le condizioni sono accertate

ha due distinte parti, unite da "if"

finisce con un punto [.]

istruzione di tipo generico, "vero" in tutte le occasioni

Figura A



china ha una buona durata. Occorre tenere da conto che vengono di seguito fornite indicazioni circa l'esecuzione dei tagliandi (soggetto meccanico) ed il riscaldamento del motore (in cui non esiste un soggetto eseguente). Il lettore attento si sarà reso conto che abbiamo reso «Macchina» una variabile, che sarà riempita in base alle risposte ritrovate nella base di dati. (È ovvio che il programma dovrà contenere in ogni caso una opportuna sezione di predicato, destinato a definire «dura» come un predicato speciale, e il tipo «meccanico» come un argomento).

La nostra regola, tradotta in linguaggio corrente, impone che, perché la condizione di durabilità della macchina sia rispettata, il meccanico esegua il tagliando e il motore sia riscaldato (anche se le nuove tendenze della meccanica sconsigliano tale pratica in quanto pericolosa per la salute dei pistoni-cilindri; ma per la verità Prolog sa ben poco di motori ed accetta supinamente le nostre notizie motoristiche, *n.d.r.*). Così il corpo della regola è rappresentato da due condizioni collegate da un connettivo [and].

Ognuna di queste condizioni abbisogna di un predicato differente; [esegue] e [riscalda], ognuno dei quali segue il formato rappresentato nella sezione di definizione precedente.

Occorre adesso aggiungere il tutto alla nostra base di conoscenza, cosa che può essere fatta direttamente (in Turbo Prolog) nella finestra di editor. A questo punto abbiamo la regola saldamente presente nella base di conoscenza, e questa stessa regola può essere utilizzata, né più né meno, come una qualsiasi parte del database stesso. La testa della regola può essere considerata, a tutti gli effetti, come l'istruzione con la quale è possibile porre richieste o creare goal cui Prolog possa rispondere.

È adesso possibile porre al programma goal più articolati, del tipo:

```
Goal: dura(macchina).
True.
```

Come fa Prolog a «leggere le regole»? Semplice; Prolog reagisce ad un goal destinato all'analisi di una regola, trasformando la testa della regola stessa in un goal. Seguendo passo passo il procedimento d'analisi, Prolog legge la regola di «dura» ed esegue una istanziazione di «Macchina» (variabile); immediatamente dopo, incontrando il connettore IF (si ricordi che Prolog non esegue effettivamente nulla se non trova un punto fermo), il programma si rende conto di essere di fronte ad un per così dire «subgoal» da soddisfare («meccanico(esegue,tagliando)»); viene eseguita una verifica di «funzionalità» se così si può dire, e se viene verificata la veridicità del subgoal viene settato un marker a True. Al ritrovamento di [and], ancora, stesso procedimento; nuovo subgoal e nuovo marker da settare; fine delle operazioni; vengono verificate le rispondenze allo scopo del goal principale e viene verificato il risultato; la variabile viene istanziata e tutti i marker vengono rimossi eccetto il primo, che «scivola» giù di una posizione nella base di conoscenza.

È questa la forma più semplice di creazione di una regola, ovviamente la stessa può essere resa il più complessa possibile. Supponiamo di dover tradurre in linguaggio Prolog la seguente regola esposta di seguito in lingua corrente:

```

utente1 e utente2 differiscono
se (if)
utente1 preferisce macintosh
e (and)
utente2 sceglie pc_ibm
o (or)
utente2 desidera vectra_hp
  
```

Figura C

due utenti non sono d'accordo sulle loro preferenze se uno preferisce il macintosh e l'altro il pc ibm o il vectra hp

tradotta in schema la regola è evidenziata dalla figura C e, tradotta in Turbo Prolog, apparirà come:

```
differiscono(utente1,utente2) if
preferisce(utente1,macintosh) and
sceglie(utente2,pc) or
desidera(utente2,vectra)
```

dove le rientranze del testo sono state messe apposta per individuare la gerarchia delle condizioni poste dai connettori logici.

Prima di concludere occorre ricordare che la stessa complessità che si può introdurre in una regola è applicabile anche nella formulazione di un goal. Ad esempio è possibile battere un goal del tipo:

```
Goal:
preferisce(utente1,macintosh) and
sceglie(utente2,pc) or
desidera(utente2,vectra).
```

Chiara come regola, non vi pare? La risposta sarà data da un «True» od un «False», a seconda del risultato della ricerca nella base di dati. Ma non solo; è possibile definire una regola basandosi su un'altra regola; tanto per esemplificare avremo:

```
Clauses
applista(utente) if
sceglie(utente,macintosh) and
differiscono(utente1,utente2).
```

Quanto complessa è possibile rendere una regola; in teoria non esiste limite alla complessità di un costruito (regola od altro). Una regola può essere tranquillamente resa tanto lunga da coprire tutte le eventualità prevedibili dalla regola stessa, tanto da riflettere esattamente la situazione che si intende individuare.

Comunque è indispensabile per l'efficienza del programma e, soprattutto, per la leggibilità dello stesso, di organizzare le regole in un modo logico e sequenziale. Se due regole a definizione separata, magari nidificate, sono più leggibili nel listato, c'è anche il vantaggio di render più efficiente la gestione del programma da parte del compilatore, per un complesso gioco di marker all'interno del programma stesso. La regola principale, in Prolog come dappertutto, è quella di evitare la complessità, fin quando è possibile.

Bene, siamo giunti alla fine dell'esposizione delle parti di un programma; è giunto il momento di scriverne uno; è quello che tenteremo di fare la prossima volta.

POSTAL COMPUTER

PC XT IBM COMPATIBILE L. 750.000

SCHEDA MADRE 6/10 MHZ, 1 DRIVE 360K, SCHEDA CGA O HERCULUS, 256K ESPANDIBILE A 640K SU PIASTRA, TASTIERA AVANZATA 101 TASTI

PC XT IBM COMPATIBILE L. 1.200.000

SCHEDA MADRE 6/10 MHZ, 1 DRIVE 360K, SCHEDA GRAFICA HERCULUS O CGA, 1 HARD DISK 20 MEGA, 256 ESPANDIBILE A 640K SU PIASTRA, TASTIERA AVANZATA 101 TASTI.

PC PHILIPS 9110

768K 1 DRIVE 5 1/4" e 1 DRIVE 3 1/2"
L. 1.230.000

NOVITÀ

CITIZEN 180 E
COMPLETA DI INTERFACCIA
IBM O COMMODORE
L. 380.000

PC AT IBM COMPATIBILE L. 1.890.000

SCHEDA MADRE 80286, 12 MHZ, 0 WAIT, 512K ESPANDIBILE A 1024K, 1 DRIVE 5,25" da 1.2 MB 1 HARD DISK DA 20 MB SCEDA HERCULES O CGA TASTIERA AVANZATA 101 TASTI.

TELEFAX MURATA M-1 L. 1.500.000

- COMPATIBILITÀ: G2 G3
- VELOCITÀ DI TRASMISSIONE 15 SECONDI
- APPARECCHIO TELEFONICO A TASTIERA INCORPORATO
- FOTOCOPIATORE
- RICEZIONE AUTOMATICA
- ROTOLO CARTA TERMICA 216 mm x 30 metri.
- OROLOGIO/CALENDARIO DIGITALE

HARD DISK SEAGATE 20 MB	L. 350.000
HARD DISK CONTROLDATA 40 MB	L. 680.000
HARD DISK CONTROLLER PER XT	L. 100.000
HARD DISK CONTROLLER PER AT	L. 220.000
SCHEDA GRAFICA SUPER E.G.A.	L. 300.000
SCHEDA MULTI I/O	L. 110.000
SCHEDA SERIALE	L. 40.000
SCHEDA PARALLELA	L. 35.000
SCHEDA PORTA JOYSTICK	L. 28.000
SCHEDA MADRE XT	L. 190.000
SCHEDA MADRE AT (12 MHZ 0 WAIT)	L. 650.000
TASTIERA AVANZATA 101 TASTI	L. 110.000
DRIVE 5,25 360KB	L. 140.000
DRIVE 5,25 1,2MB	L. 190.000
DRIVE 3,50 720KB	L. 190.000
DRIVE CONTROLLER	L. 49.000
CAVO PARALLELO	L. 15.000
DATA SWITCH A 2 PORTE	L. 60.000
MOUSE ANKO	L. 59.000
JOYSTICK I.B.M. ANKO	L. 45.000

STAMPANTI CITIZEN GRAFICA - NLQ

CITIZEN 120 D L. 360.000 120 CPS, SET. EPSON IBM 80 COL. TRATO IN TRAZIONE, FRI- ZIONE INTER. OPZIONALE IBM/COMMODORE	CITIZEN MSP 50 L. 1050.000 250/300 CAR/SEC., 80 COL.
CITIZEN LSP 100 L. 550.000 -160 cps, 80 COL.	CITIZEN MSP 55 L. 1.230.000 250/300 CAR/SEC., 136 COL.
CITIZEN MSP 10E L. 650.000 - 160 CAR/SEC., 80 COL.	CITIZEN HQP 40 L. 1.160.000 - 24 Aghi, 200 CPS ALTISSIMA QUALITÀ
CITIZEN MSP 15E L. 680.000 160 CAR/SEC., 136 COL.	CITIZEN HQP 45 L. 1.530.000 - 24 Aghi, 200 CPS ALTISSIMA QUALITÀ
CITIZEN MSP 40 L. 775.000 - 200/240 CAR/SEC., 136 COL.	CITIZEN PREMIERE 35 L. 1.250.000 - MARGHERITA PROFESSIONALE, 35 CPS
CITIZEN MSP 45 L. 950.000 - 200/240 CAR/SEC., 136 COL.	CITIZEN OVERTURE 110 * L. 3.600.000 - STAMPANTE LASER

**TUTTI I PRODOTTI CITIZEN SONO COPERTI
DA CERTIFICATO DI GARANZIA DELLA VALIDITÀ DI DUE ANNI**

OFFERTA MONITOR

PHILIPS

MONITOR 8875 14" MULTISINK	L. 935.000	colore
MONITOR 8833 14" CGA	L. 450.000	colore
MONITOR 8802 14" COLORI	L. 360.000	colore
MONITOR 9043 14" EGA	L. 535.000	colore
MONITOR 9053 14" EGA	L. 595.000	colore
MONITOR 9073 14" EGA	L. 680.000	colore
MONITOR 7723 14" TTL	L. 192.000	F/A
MONITOR 7743 14" TTL	L. 205.000	F/B
MONITOR 9082 14" VGA	L. 700.000	colore

Segue PHILIPS

MONITOR 7749 14" TTL	L. 210.000	F/B
compatibile IBM sist. 2	L. 136.000	F/V
MONITOR 7513 12" TTL	L. 183.000	
MONITOR 7713 14" TTL	L. 183.000	
ANTAREX		
BOXER 14" P39 JAN DUAL	L. 190.000	F/V o F/B
BIM 12" PC DM 216B	L. 135.000	F/V
CT 9000 SHR EGA JAN	L. 670.000	colore
CT 9000/L MR14 DIM 414	L. 430.000	colore

**PREZZI
SU RICHIESTA**

GARANZIA 18 MESI

**PREZZI IVA ESCLUSA
SPESE DI SPEDIZIONE ESCLUSE**

TEL. 06/3652427/3652431

TELEFONATECI

AMSTRAD PC/IBM Comp.

8086 - 8 MHz - Drive 360K - Monitor - Interf. parallela seriale - mouse MS/DOS - 3.2 GEM - DESKTOP - GEM PAINT - BASIC 2.

Configurazione con 512K

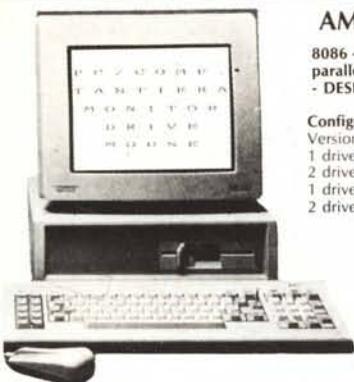
Versioni e manuali in italiano
1 drive - monitor monocrom. L. 1.210.000
2 drive monitor monocrom. L. 1.560.000
1 drive monitor graf. col. L. 1.560.000
2 drive monitor graf. col. L. 1.960.000

Configurazione con 640K

Versioni e manuali in italiano
1 drive - monitor graf. monocrom. L. 1.440.000
2 drive - monitor graf. monocrom. L. 1.860.000
1 drive - H.D. 20MB monitor graf. monocrom. L. 2.620.000

Configurazione 1640 ECD con monitor colore EGA.

1 drive L. 2.260.000
2 drive L. 2.680.000
1 drive - H.D. 20MB L. 3.410.000



Portatile Amstrad a partire da L. 1.210.000

BABY XT

Dim 26x26x8cm
8088 - 4.77/10MHz 512K - 2 drive da 3"1/2 - schede Hercules, CGA, porta ser. parall. mouse - tastiera 84 tasti - monitor monocrom. 12" doppia freq. - orologio con batt. - DOS 3.3 - man. L. 1.770.000.



COMPUTERS

NOVITÀ

BABY XT, dim. 26x26x8 1.770.000
8088, 4.77/10 MHz, 512K, 2 drive 3" 1/2, interf. ser. parall., mouse, tast. 84 tasti, mon. mono, doppia freq. Scheda Hercules e CGA.

MASTERBIT AT 3.350.000
512K, 80286, drive 1,2. HD 20 Mb, scheda EGA-MGA, monitor 12" fosfori verdi.

PC/AT 2.990.000
80286, 10 MHz, 512K esp. a 4 Mb, 1 drive 1,2 Mb. Hard 20 Mb, Hercules, CGA, EGA, Tast. 101 tasti, orologio, interf. parall.

PC 386 6.800.000
20 MHz, 512K esp. 8 Mb, 1 drive da 1,2 Mb, Hard 40 Mb/30MS, Hercules, CGA, EGA. Tasti 101 tasti, monitor mono dual.

PC ready 88 1.690.000
8088, 4.77/8 MHz, 1 drive da 5" 1/4, Hercules, Tast. 102 tasti, interf. ser. e parall., rponitor mono basculante, DOS 3.3., man. it.

PC/AT READY 3.480.000
80286, 8/12 MHz, 512K esp. 4 Mb, 1 drive da 1,2 Mb, Hard 20 Mb, Tast. 102 tasti, interf. ser. e parall., orologio, Hercules, monitor mono basculante, DOS 3.3 man. It.

PORTATILE HALIKAN 2.750.000
NECV20 4.77/10 MHz, 640 K, 2 drive da 3"1/2, display 640x200, uscita per monitor, RGB mono. Tast. 81 tasti, interf. ser. e parall. batt. interna, alim. borsa, DOS 3.3 GWBasic, man.

PC WORD PROCESSOR AMSTRAD 256k 980.000

512K 1.450.000

PC BONDWELL 8 - Portatile, 512K, 1 drive 720K, 3"1/2 - Scheda grafica col. 1.650.000

VIDEO WRITER PHILIPS 690.000
monitor monocrom. fosfori Ambra a 100 col. e 20 righe, tastiera 72 tasti, stampante incorporata termica a 24 aghi, cps da 25 a 50.

PC VEGAS 1.200.000
256K 1 Drive da 5,3/5 Hercules Monitor

SPECTRAVIDEO XIPRESS 16 1.350.000
256K, 8088, 2 drives, monitor 9", joystick, MS-DOS 3.2

PORTATILE SPARK 1.990.000
NEC V 20, 4.7/9.45 MHz, 384K, 1 Drive 3"1/2, interf. ser. e parall. DOS 3.3

PORTATILE TOSHIBA T 1100+ 2.760.000

PC PHILIPS 1.480.000
8088, 4.77/8MHz, 512 K 1 drive 3"1/2 Hercules CGA, orologio, interf. seriale parall. mon. monocrom.

PC PHILIPS 1.650.000
come sopra ma con 768 K e 2 drive

PC-PS/30 I.B.M. COMP. 1.870.000
8088, 4.77/10 MHz, 256K 2 drive da 3"1/2, monitor monocrom. Tastiera 101 tasti, porta ser. e parall., schede hercules e CGA

PC ASEM 3011 1.830.000
Nec V20, 10 MHz, 256K, 2 drive, Hercules, monitor mono, Tastiera Dos 3.3

TRASPORTABILE XT 2.190.000
8088, 10 MHz, 256K, 2 drive da 5"1/4 CGA, display retro illum. Tastiera.

ATARI PC3 1.660.000
8088, 640K, 2 drive 5"7/5, schede Hercules CGA, EGA, interf. ser. e parall., monitor ambra, mouse, DOS 3.2, man. it.

AT 80286 L. 2.550.000
512K, 12 MHz, 1 drive da 1,2, 1 hard disk 20 Mb, schede Hercules e CGA, interf. ser. e parall. tastiera, monitor 12" doppia freq., DOS 3.3

PORTATILE SHAR PC/4502 L. 2.760.000
80188, 7,16 MHz, 384K, 2 drive da 3"1/2, schermo retro illum., 88 tasti, porta ser. e parall., batt. ricar., DOS 3.2L.

STAMPANTI

EPSON LX 800 599.000
80cd., 150cps, NIQ, Grafica, IBM compat.

EPSON LQ 500 760.000
24 aghi, 80 cd, 150 cps, LQ, Grafica, IBM compat.

MANNESMANN TALLY MT 80 PC 499.000
80 col. 135 cps bidirez. NIQ

MANNESMANN TALLY MT 85 789.000
80 col. 180 cps NIQ bidirez. interf. parall. o seriale IBM/comp.

MANNESMANN TALLY MT 86 959.000
136 col. 180 cps NIQ bidirez. interf. parall. o ser. IBM/comp.

MANNESMANN TALLY MT 87 869.000
80 col. 200 cps NIQ grafica IBM comp.

MANNESMANN TALLY MT 290 1.870.000
132 col. 200 cps NIQ interf. parall. IBM/comp.

MANNESMANN TALLY MT81 L. 3.700.000

MANNESMANN TALLY MT 905 LASER 3.350.000

STAR LC 10 519.000
80 col., 120 cps, NIQ, grafica IBM comp.

STAR LC 10CL 559.000
come la STAR LC 10 ma a colori

STAR LC 24-10 890.000
80 col. 24 aghi, 170 cps, LQ grafica e IBM comp.

AMSTRAD DMP 3160 490.000
80 col. 160 cps NIQ Graf. IBM comp.

AMSTRAD DMP 4000 790.000
132 col. 200 cps, NIQ, Grafica IBM comp.

AMSTRAD LQ 3500 900.000
24 aghi, 80 col. 200 cps. Grafica IBM comp.

AMSTRAD LQ 5000 di 1.310.000

132 col., 288 cps 24 aghi, grafica, I.B.M.comp. interf. ser. e parall.

OLIVETTI DM 100 580.000
80 col, 120 cps, NIQ, grafica, IBM compat.

NEC P2260 975.000
24 aghi, 80 col, 168 cps, grafica, IBM compat. 12 font resident.

PANASONIC KX-P 1081 589.000
80 col. 120 cps, NIQ, IBM comp. grafica

PANASONIC KX-P 1540 1.720.000
136 col. 240 cps, LQ, 24 aghi

PHILIPS NMS 1432 519.000
80 col. 120 cps, I.B.M. comp. graf.

MONITORS

GM 1288 D 220.000
12" doppia freq. Hercules e CGA, fosf. verdi basculante per PC

HANTAREX BOXER 12 229.000
12" fosf. verdi alta risoluzione

HANTAREX BOXER 14 doppia freq. 269.000

HANTAREX 14 499.000
14" colore standard risoluz. 80 col.

MONITOR QL 14 399.000
per QL, 85 COL., colore

MODEM

MODEM 300 baud per C64 175.000

MODEM 300 baud per RS 232 e IBM/199.000

MODEM 1200 RF 560.000

CCITT V21/V22 BELL 103/202 - 300/600-1200 Baud può allacciarsi a qualunque sistema di ritrasmissione, radiotelefonici - OM - CB.

MODEM COMMUNICATOR, 300/600/1200 e VIDEOTEL per C64/128/VIC 20 225.000

PER IBM - COMPAT. - OLIVETTI 248.000

TUTTO COME SOPRA MA CON AUTOANSWER PIU' LIRE 20.000

MODEM FULL LINK - 300/1200 FULL DUPLEX - HAYES ESTESO - INTERF. SER. E CENTRONICS - OMOLOGATO 550.000

MODEM SU SCHEDA PC INTEGRAL 300/1200 FULL DUPLEX - HAYES ESTESO - OMOLOGATO 470.000

MODEM ECLIPSE - 300/600/1200 - VIDEOTEL - INTER. SERIALE - AUTOANSWARE - OMOLOGATO 420.000

MICROSMART 339.000

V21 - V22, interf. ser. o TTL, AUTODIAL, AUTO ANSWER, HAYES esteso

AMSTRAD PC CARD 420.000
300/1200/75-1200/1200-75

JOYSTICK

DATALINE standard 9 PIN D 14.000

MASTERBIT MASTERBIT XT - DM 512

nuovo processore NEC V40, 5,5 e 8MHz, controller a 4 canali, 512 K, 2 drive, da 5"1/4, scheda Hercules e CGA, porta parallela e seriale, orologio, zoccolo per processore matematico, tastiera italiana, 4 slots, MS/DOS 3.2, GW BASIC 3.2, manuale in italiano, pulsante RESET esterno, altoparlante, monitor 12" fosfori verdi, doppia freq. basculante oppure 14" colore. Nuova Tecnologia

L. 1.570.000



PC PORTATILE OLIVETTI M15.

80C88 - 4.77/MHz - 512K - 2 drive da 3"1/2 - Display crist. Liq. 80 col., 25 righe, 640x200 - interf. ser. e parall. - Collegam. est. per drive da 5"1/4 - Batt. ricar., con 6 ore d'autonom. - tastiera 78 tasti - Aliment. con carica batt. - Borsa - 4 man. in it. - DOS 3.3. L. 1.350.000

SPECTRAVIDEO QS II plus 25.000
SPECTRAVIDEO QS IV 20.000
SPECTRAVIDEO QS IX 25.000

SINCLAIR QL

QL vers. ingl. JS 329.000
2 ROM JS (trasf. il QL da JM a JS) 60.000
CONVERTITORE RS 232 per stampante 99.000
CAVO QLR/S232 stamp. 35.000
CAVO JOYSTICK per QL 19.000
CAVO SER 1 per QL 15.000
BOX per 20 Microdrive 15.000
Copriplastiera per QL 12.000
Inter. disco - porta parallela + RAM disk + toolkit I 229.000
drive NEC singolo 259.000
drive NEC duo 229.000
doppio drive NEC unico contenitore 519.000
Orologio residente 30.000
TUTTI I PEZZI DI RICAMBIO: es. Contattiera 30.000

SINCLAIR SPECTRUM

SPECTRUM PLUS 48K 260.000
MANU.IT, 5 progr. supercop.

SPECTRUM 128K 299.000
2 cassette con giochi

SPECTRUM 128+2 415.000
con registratore incorporato.

Interfaccia Stampante su ROM 99.000
Interfaccia joystick tipo Kempston 1 presa 25.000
Interfaccia parlante CURRAH 60.000
Int. Ram Print. 120.000
RAM Writer incorporato + porta joystick

INTERF. DISCIPLE 199.000
interf. disco, porta parallela per stampante 2 porte joystick, 2 network, magic bottom compat. con drive da 3"1/2, 5"1/4 e interf. 1

INTERF. DRIVE con magic bottom 119.000
DRIVE NEC 3" 1/2, 720K formattati

259.000
Multifacce 1, magic bottom 105.000
Cartucce per Microdrive 5.500
Music Machin con cuffia, microtono e cassetta demo 129.000

TUTTI I PEZZI DI RICAMBIO: es. Ula 38.000

VARIE

1500 prog. per PC/comp. .
10 FLOPPY POLAROID 5"1/4 con custodia cartone 26.000
10 FLOPPY POLAROID 5"1/4 30.000

con custodia di plastica
FLOPPY POLAROID da 3"1/2 5.000
FLOPPY NEUTRI con etichetta 1.200
FLOPPY SENTINEL 3"1/2 3.500
SENTINEL 5"1/4 2.000

10 pz. con contenitore plastica nera.
VOICE GENIUS GM6 PLUS 130.000
VOICE CARD 180.000
Fa parlare il vostro PC
MOUSE PER PC 120.000
SCHEDA JOYSTICK PER PC 65.000
INTER. TRANSPCOPY PER PC 385.000
SCHERMO ANTIRIFLESSO POLAROID 120.000

HARD DISK MINISCRIBE 760.000
32 Mb con controller e cavi
HARD DISK MINISCRIBE in scheda 800.000
32 Mb con controller e cavi
HANDY SCANNER 550 mm 519.000
HRC, GCA, EGA per XT/AT/PS-2

FAX MURATA MI 1.599.000
FAX per XT e AT 869.000
Driver da 3"1/2 per PC interno 279.000
Nuovo processore NEC V20 60.000
Scheda PARADISE, compatibile 335.000
ACCESSORI e PERIFERICHE PER COMP. IBM
INTERF. per PPC AMSTRAD e TV Domestico 449.000

SCHEDA CGA/TV 449.000
Collega un XT ad un monitor CGA o ad un TV a colori o ad un VIDEOREGISTRATORE

INTERF. PER TV A COLORI CON PRESA SKART E COMPAT. CON SCHEDA COLORI CGA 110.000

GRUPPI DI CONTINUITA
BOX PER 80 FLOPPY 3"1/2 25.000
BOX PER 100 FLOPPY DA 5"1/4 25.000
KIT DI PULIZIA 5"1/4 6.500
KITI PULIZIA 3"1/2 10.000

FOTOCOPIATRICE CANON PLOTTER GRAFITEC A TAGLIO MULTIDRIVE esterno 580.000
1 drive da 5"1/4 e 1 drive da 3"1/2 per AMIGA 500, 2000, ATARI

INTERF. DIGITALIZZ. AUDIO 140.000
Regolabile con progr. e man. per AMIGA 500

PHILIPS macchine da scrivere elettr. da 290.000

COORDINA, programmi modulari contab. gen., magaz. e fatturaz. REGISTRATORE DI CASSA Misuratore fiscale

INDESIT, collegabile al PC per gestione magazzino
FLOPPY KONICA da 5"1/4 1.950
FLOPPY KONICA da 3"1/2 3.750
FLOPPY KONICA alta dens. 5"1/4 4.150
FLOPPY KONICA alta dens. 3"1/2 8.450

AVVERTENZE - Tutti i prezzi sono comprensivi di IVA e spese postali, per ordini inferiori alle 50.000 lire aggiungere L. 8.000 per contributo spese di spedizione - pagamento contrassegno al ricevimento del pacco. (E gradito il contatto telefonico).
SCONTI QUANTITA

ORDINI TELEFONICI
ORE 8.30/20.30 - Tel. 06/5621265

Garanzia 48H - la MASTERRBIT si impegna a sostituire quegli articoli riscontrati malfunzionanti entro 48H dal ricevimento, inoltre ogni articolo è fornito di regolare garanzia.
MASTERBIT Viale dei Romagnoli 35 - 00121 OSTIA LIDO RM - CAS. POST. 3016