



L'Aztec C

di Raffaello De Masi

Che il Basic sia divenuto l'esperanto dell'informatica è ormai praticamente accertato e accettato. Molti linguaggi hanno tentato di scalfinare il predominio, dal moderno Pascal alle più recenti revisioni dei datati Cobol e Fortran. Ma «lui» procede indisturbato, forte della versatilità che si porta dietro e degli ultimi aggiornamenti che ne permettono anche un uso strutturato, fino a poco fa strombazzata dote cromosomica di Pascalisti ed similia. È ora la volta del «C», a sostenere il ruolo di sfidante? Non è propriamente così; il C si pone in una posizione un po' diversa rispetto a quella del Basic e degli altri linguaggi «sfidanti». È diverso da tutti, e sotto tutti gli aspetti. Non è un'alternativa, è un'altra cosa da usare, anche, in occasioni in cui il Basic non può essere usato. Non si dimentichi che è in «C» che è stato redatto il sistema operativo UNIX. Ed attualmente del «C», che nei laboratori Bell esisteva e serviva ben prima

che divenisse di uso comune, esiste sul mercato dei micro una massiccia presenza: per il Mac, per fare l'esempio che ci riguarda, ne esistono sei implementazioni. Lo stato dell'arte in questo ambito è probabilmente rappresentato dalla versione della Manx, l'Aztec C, di cui andiamo a vedere le caratteristiche.

L'Aztec C realizzato dalla Manx per il Mac è arrivato alla versione 1.06. Di esso esistono tre versioni: «Commercial», «Developer», e «Personal», in ordine decrescente di potenza e prezzo. L'ultima, dedicata ad un utente non molto sofisticato, rappresenta un mezzo d'introduzione al linguaggio: infatti costituisce un ottimo tutorial per l'utente, non familiare con il «C», che vuole imparare ad usare tale linguaggio sul Mac. Poiché molti volumi introduttivi sul «C» sono orientati a lavorare in un ambiente UNIX-like, l'ambiente SHELL costruito dal package rappresenta un'ottima palestra di allenamento. Ciò consente, inoltre, di poter utilizzare immediatamente programmi già esistenti, anche destinati a

mini e mainframe, in quanto essi vanno battuti alla tastiera esattamente come vengono letti, senza preoccuparsi di particolari feature specifiche della macchina che si sta usando.

Il Personal System, inoltre, pur rappresentando un sistema introduttivo per il «C», possiede già tutte le caratteristiche finali del pacchetto per quanto attiene allo SHELL ed al compilatore. Per quanto attiene specificatamente al Mac, già nel Personal esiste, per il programmatore, la possibilità di accedere direttamente al «toolbox» grafico del Mac, con utilizzo diretto delle oltre 700 routine descritte nell'«Inside Macintosh».

Esistono, comunque, tra i tre pacchetti, diversità, anche elevate, riguardanti l'effettiva potenza del sistema stesso. Le principali differenze si vedono solo riassunte nell'allegato schema (nel manuale occupano una decina di pagine). Comunque, nelle descrizioni successive, faremo generalmente riferimento al Personal System, e, solo eventualmente faremo notare, se è il caso, e se davvero possono interessare l'utente medio, le maggiori feature dei sistemi maggiori.

Aztec C ha una perla, un fiore all'occhiello, un particolare di tutto rispetto, che evidenzia la cura con cui tale pacchetto è stato realizzato: il manuale. E di questo ci serviremo per sviluppare la nostra prova, che è stata enormemente facilitata dalla lettura, ahimè non certo esaustiva, di tale volume.

L'entrata in Shell (per chi non lo sapesse o ricordasse, si indica con tal nome l'ambiente operativo proprio del sistema operativo UNIX e, in particolare, del nostro linguaggio) avviene in



La confezione del compilatore Aztec C 68K comprende ben 3 dischetti: Commercial, Developer, Personal ed un manuale di istruzioni di generose dimensioni.

Produttore:
MANX™ Aztec C
di Jim Goodnow II
Manx Software System
P.O. Box 55
Shrewsbury, N.J. 07781 USA

Prezzi:
Aztec c68k-c (commercial) \$499
Aztec c68k-d (developer) \$299
Aztec c68k-p (personal) \$199

bootstrap; ciò viene evidenziato dalla scomparsa del familiare schermo-scrivania-finestra, proprio di tutte le applicazioni Mac, e da uno startup screen di presentazione: pigiando il tasto del mouse compare la videata di base e ci si trova in ambiente, evidenziato da una frase di saluto «Benvenuto nel meraviglioso mondo di Aztec C» e da un menu comprendente, oltre i soliti «mela», «file», ed «edit», tre altre opzioni: «Command», che elenca i comandi principali, divisi in due categorie, «Source», che evidenzia il codice sorgente, e «Compile» che raccoglie i comandi destinati a gestire il codice sorgente. In un ambiente UNIX come questo la tentazione è troppo forte: battiamo «WHO» e ci vediamo rispondere laconicamente «Ti aspetti forse di essere in UNIX??», con i due punti interrogativi finali!

Bene: finiti qui, con ignominia gli esperimenti, battiamo:

```
ls
ed avremo, sullo schermo un listing dei file presenti sul dischetto. Battiamo ancora
```

```
ls -l
ed avremo un catalogo più preciso ed aggiornato, evidenziante, come negli standard dello shell «UNIX», il tipo e la data di creazione di ciascun file. Proviamo, a questo punto, a creare una nuova directory: scegliamo, dal menu «Command», o battiamo direttamente dalla tastiera (cosa più confacente all'utente UNIX) il comando:
```

```
cd/aaaa
Poiché la directory «aaaa» non esiste, lo Shell mostra il messaggio «Empty directory», per ricordare che si sta creando una nuova directory; battiamo
```

```
cat
e saremo pronti a concatenare due file come più ci aggraderà.
```

Diamo un'occhiata più da vicino allo shell. Parliamo del video di bootstrap: abbiamo già descritto il menu «mela»: «File» consente a finestre in background di ritornare in foreground contemporaneamente disattivando la finestra di Shell. «Edit» è il classico editor dell'Apple; non compare attiva-

Figura 1 - La videata iniziale, in ambiente UNIX-like, ed il catalogo del dischetto di boot; in ossequio alle regole di tale s.o., i comandi, in maiuscolo, non vengono riconosciuti, tranne poi, come viene descritto nel testo, rispondere alle spiritosamente alle nostre richieste.

```

File Edit Commands Source Compile

Welcome to the Wonderful World of Aztec C !!!

-? WHO
file not found, WHO, -1
-? who
You were expecting maybe UNIX??
-? ls -l
sys:/
TEXT flonx 01 00 84 0 ott 4 1985 .profile
<DIR> bin/
FINDER.ERIK 40 00 0 1605 giu 29 11:14 DeskTop
TEXT flonx 01 00 39 0 ott 4 1985 hello.c
PRES HURT 31 00 0 30954 giu 29 10:23 laegelriter
<DIR> lib/
FINDER.ERIK 40 00 0 1476 giu 28 09:22 Scribania
APPLE flonx 01 84 0 30930 ott 4 1985 SHELL
ZSYS IRACS 01 84 5120 48640 giu 29 11:19 System
9 files, 5243 in data, 113605 in rarc
-? █

```

to all'inizio, ovviamente, ma è utilizzabile solo quando esiste un codice sorgente (battuto direttamente alla tastiera o richiamato da disco), su cui, appunto, le operazioni di editing sono eseguibili. Il menu «Command», anch'esso già intravisto, elenca solo alcuni dei comandi eseguibili. Passiamo a «Compile»; esso comprende ancora un editor, che, a differenza del precedente, adotta la classica configurazione UNIX. Le altre utility (assemble, compile, link) sono i mezzi di base del linguaggio. Da notare che la creazione di librerie personali sono del tutto facilitate da una serie di appositi comandi, destinati a rendere del tutto maneggevole tale principale caratteristica del linguaggio.

Fin qui Aztec «C» conserva gli standard generali del linguaggio, così come definito da Kerninghan e Ritchie nel loro volume «The C Programming Language». Vediamo adesso qualche caratteristica, specifica dell'accoppiata Aztec-Mac.

Lo Shell parte automaticamente all'accensione, ed è questa la procedura da seguire sempre: se lo si lancia dal Finder si avrà un messaggio d'errore (ID = 99) al momento di entrare in ambiente, il sistema si inchioda e occorre rilanciarlo. Compare, a questo punto, la prima, particolare, caratteristica di tal linguaggio. Se si desidera che una sequenza di comandi sia eseguita ogni volta alla partenza dello Shell, è possibile inserire tali comandi (ad esempio settaggio di apparati di I/O, caricamento di librerie, definizione fissa od automatica di variabile,

operazione di «append» di testate o parti fisse di programmi, ecc.) in maniera del tutto automatica, mediante l'inserimento di opzioni, comandi, parti di codice desiderati in un file, di nome «profile» che verrà, all'autoboot del linguaggio ed all'entrata in ambiente, eseguito nella sua interezza. Per uscire dallo Shell è sufficiente chiamare il Finder; non sempre, però, tale procedura ha effetto, per problemi di memoria; a questo punto non c'è altro da fare che spegnere e riaccendere la macchina. Il manuale presenta un programma di tre righe destinato ad una meno cruenta operazione d'uscita; una volta compilato col nome che più ci aggrada (per esempio «Esci») sarà sufficiente invocarlo per ritornare nel familiare ambiente Mac.

Andiamo avanti, per forza di cose un po' superficialmente (un manuale di un chilo e mezzo di peso, scritto in corpo 8 non è facile da riassumere). Il compilatore, invocato dal solito [cc] seguito da opzioni e dal nome del file da compilare, (si agisce da ambiente generale su file testo presenti su disco), rispetta pienamente gli standard di Kerninghan e Ritchie. Esso legge il codice sorgente dal file specificato in input, lo trasforma in linguaggio assembly e ne scrive il risultato in un altro file. In sequenza, senza dover essere appositamente invocato, entra in azione l'assemblatore che trasforma tale codice intermedio in un codice oggetto, ovviamente rilocabile, scrive i risultati su un altro file (ricordate il classico a.out dell'ambiente Unix?) e cancella il file sorgente assembly. Una

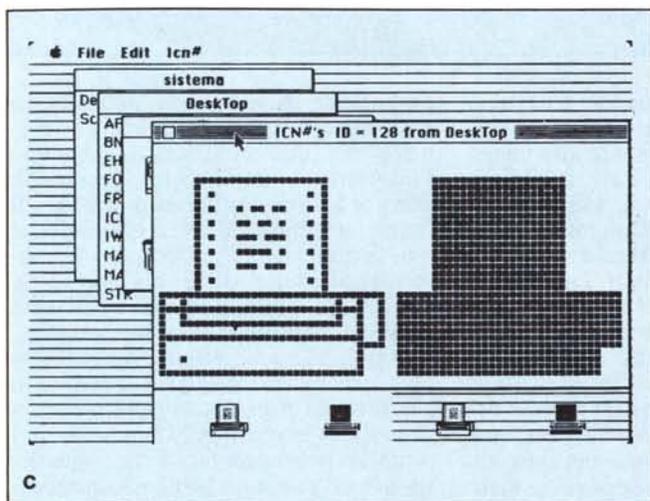
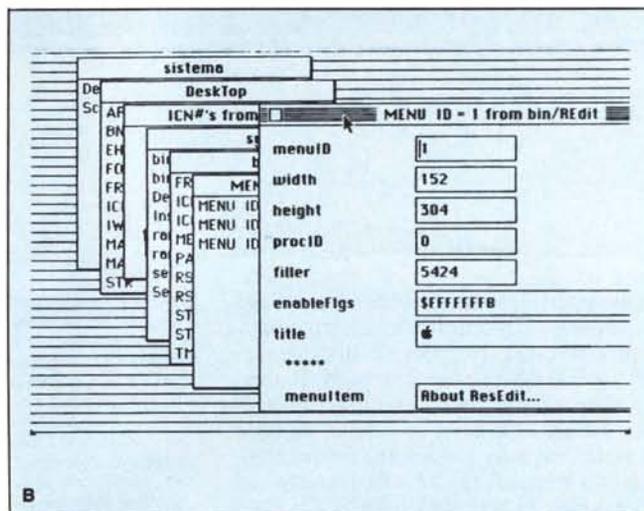
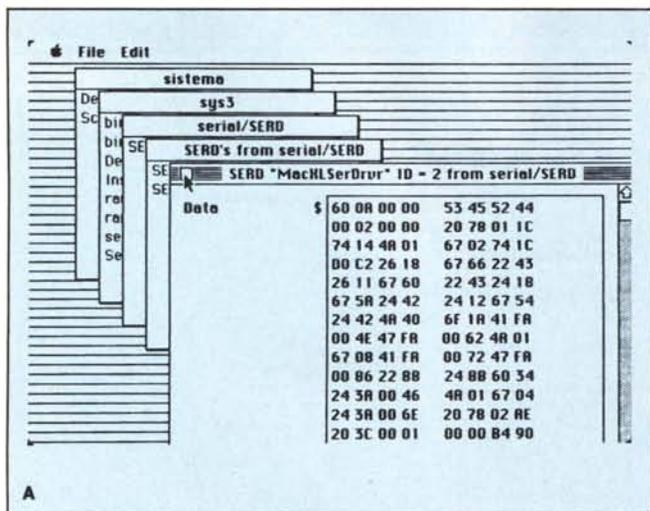


Figura 2 A, B, C
Il resource editor in azione: si noti, nella A, il protocollo di accesso ad una delle porte seriali; in B, la creazione del menu «Mela», con le ampiezze del menu e la sua lunghezza; e, infine, nella C la fase di costruzione della icona di Imagewriter.

cificate da tale comando. Ulteriori numerose opzioni (una quindicina) consentono di trasferire informazioni al preprocessore, specificare grandezze di simboli locali o di stringhe in default, definire aree (anche non limitate ad una sola directory) specifiche di ricerca per routine di libreria, o macro autodefinita dall'utente e presenti in particolari librerie.

Gran parte del manuale è specificamente dedicata alla scrittura di programmi per il Mac. A tale proposito Aztec C supporta una comodissima utility che definisce un simbolo identificante le macchine sulla quale il codice può girare. Nell'edizione attuale sono supportati i seguenti microprocessori:

- 68000 con codice MPU 68000
- 8086/8088 con codice MPU 8086
- 80186/80286 con codice MPU 80186
- 6502 con codice ecc
- 8080
- Z80

Tali codici sono molto utili in quanto possono essere inseriti in un programma, come chiamata di individuazione del microprocessore, con una serie di IF, in modo da non avere più preoccupazioni circa la macchina su cui il programma vien fatto girare.

Nell'ambito di questa sezione del manuale viene dedicata molta azione ai puntatori, croce e delizia dei programmatori in «C». Ancora, è possibile richiamare funzioni Pascal, presenti in ROM, dichiarando funzioni Pascal con una istruzione della forma:

```
pascal type func() = 0 → 1234
dove:
```

— type è il tipo di valore restituito dalla funzione

apposita opzione, [-a] consente di interrompere l'operazione al primo stadio, non facendo partire l'Assembler e consentendo, a chi lo volesse, l'esame del sorgente in assembly.

Ovviamente tale operazione beneficia di tutte le facility ed opzioni di un ambiente UNIX-like. È possibile, ovviamente, chiamare, fuori dal default, codici sorgente e file presenti in altre subdirectory, directory o dischi diversi. Così, ad esempio:

```
cc dskr:db/dir 1/programma 1.c
compilerà il codice «programma 1» (si noti .c finale, secondo i classici schemi UNIX) presente nella directory «dir 1».
```

In default il codice oggetto, generato da un compilatore-Assembler, possiede un nome derivato da quello del codice sorgente, cambiando la sua estensione in [.o]. Nel caso precedente il risultato sarà «programma 1.o» e

conterrà l'applicazione finale; è altresì possibile specificare immediatamente il nome del file contenente il codice generato con la procedura:

```
cc -o programma 2.o programma 1.c.
```

Ancora, come nell'esempio precedente, se si desidera conservare il codice intermedio, sarà sufficiente inserire l'opzione [-C]. Comparirà nell'-Is, un file con l'appendice [-asm]. Così il comando:

```
cc -T programma
```

darà, dopo la compilazione, 3 file dal titolo:

- programma .c già esistente;
- programma .asm in linguaggio assembly;
- programma .o la vera e propria applicazione, in linguaggio macchina.

L'operazione di #include, fondamentale in «C», consente, ancora, di scegliere le subdirectory o le directory in cui eseguire le ricerche dei file spe-

— func è il nome della funzione
 — 0 → 1234 (o qualunque altro valore tra di essi) è il valore di aggancio della funzione (si veda Inside Macintosh).

L'Assembler è il terzo dei momenti fondamentali dell'operazione di realizzazione di un programma. Di esso diremo poco, dato che ben di rado di lui si serve intenzionalmente il programmatore, visto che, generalmente, esso riceve automaticamente il programma dal compilatore e lo trasforma in un codice rilocabile. Ciononostante il manuale fornisce chiari e precisi riferimenti alle sequenze operative specifiche della fase d'assemblaggio, con un'accurata descrizione delle opzioni presenti.

Conclude le operazioni il linker, che lega insieme parti diverse di programma, che, in formato del tutto rilocabile, dopo essere stati compilati, concorrono alla esecuzione del programma principale. Anche qui esistono specifiche precipe del Macintosh; possono essere realizzati tre tipi di programma:

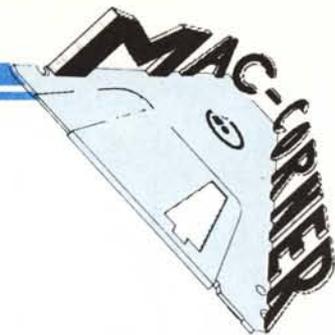
- comandi, eseguibili direttamente dalla tastiera in ambiente Shell
- driver, chiamati da altri programmi per accedere a periferiche
- accessori della scrivania, richiamabili, nel menu Mela, attraverso il menu discendente.

La maggior parte del manuale prosegue con un esame dettagliato dei comandi e delle istruzioni presenti. Si tratta di un vero e proprio tutorial, che illustra, e non poteva essere che così, anche l'accesso al ToolBox Mac. Que-

st'ultima parte è arricchita da una sezione che, attraverso alcuni chiari esempi, illustra come i programmi possano accedere facilmente a caratteristiche specifiche del Mac. Successive altre appendici introducono allo stile di programmazione «C», alle possibilità di escape da errore, (questi ultimi suddivisi in categorie, a seconda della fase in cui accadono), alla possibilità di utilizzo di routine proprie di altri linguaggi, ai criteri generali di portabilità, ecc.

Conclusioni

Aztec «C» è davvero il più completo compilatore C su cui ci sia capitato di mettere le mani. Altri ne abbiamo visti, come il Megamax ed il Desmet, anch'essi dedicati al «C», ma ci pare che questo abbia davvero tutti i numeri per essere ricordato come il campione del campo. Già la stessa simulazione d'ambiente UNIX (altri lavorano sulla più classica scrivania Mac) rappresenta un biglietto da visita di prim'ordine, ed evidenzia il gigantesco lavoro di programmazione esistente alle spalle dallo staff (capitanato da Jim Goodnow II) che ha raggiunto la release 1.06 G. Il pacchetto è arricchito dalla più recente versione del Resource Editor, di Rony Sebok e Steve Capps, già visto su altri pacchetti e dal Resource Compiler di Andy Hertzfeld, anch'esso non nuovo al programmatore impegnato. Poco male se non li si sa sfruttare appieno; già imparare completamente questa versione di «C» è una bella impresa. Saranno suf-



ficienti per cambiare qualche icona o qualche messaggio dei nostri dischetti; prima o poi impareremo ad usarli meglio.

Lavorando con MacPaint

INGRANDIMENTI

MacWrite consente, come si sa, di ingrandire figure già realizzate utilizzando il rettangolo di selezione di cui va afferrato e tirato uno spigolo. Semplice, ma quasi mai efficace! Generalmente il risultato ottenuto è modesto: lo scritto risulta scarsamente leggibile, i contorni del disegno dentellati, e i retini formati di punteggiatura piuttosto irregolare.

Per risolvere soprattutto quest'ultimo problema risulta molto utile una tecnica, peraltro abbastanza semplice ed intuitiva, destinata a rendere più gradevole l'aspetto finale di una figura piuttosto complessa (per quelle semplici bastano, infatti, pochi tocchi sulla immagine ingrandita per risolvere,

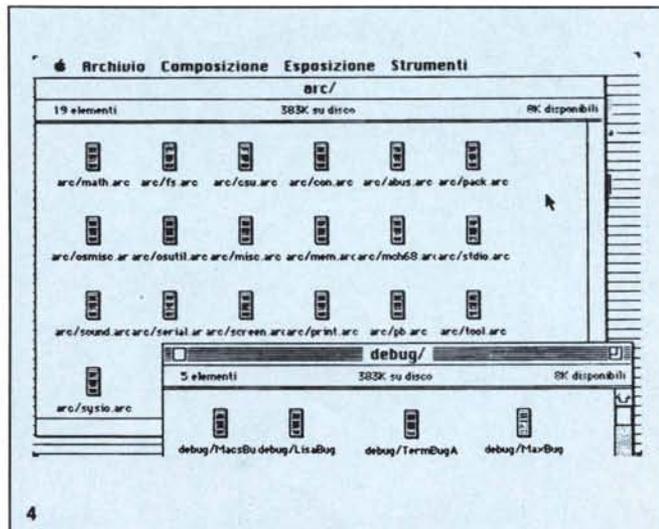
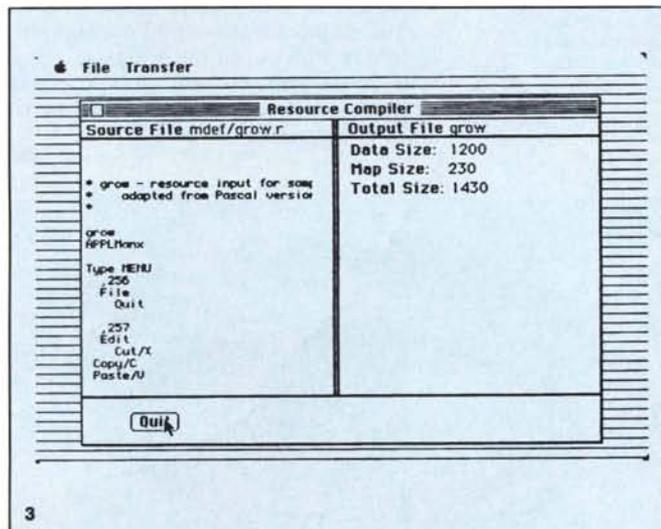


Figura 3 - L'uso del Resource Compiler ci permette di accedere più profondamente nel sistema, alla ricerca delle routine di toolbox; in figura si vede la definizione di alcuni comandi da tastiera. Figura 4 - Il contenuto del dischetto 1.3; le icone, che in default appaiono come il classico documento Macintosh (pagina bianca) sono state qui ridefinite, tramite il resource editor, a mo' di colonne di giornale.

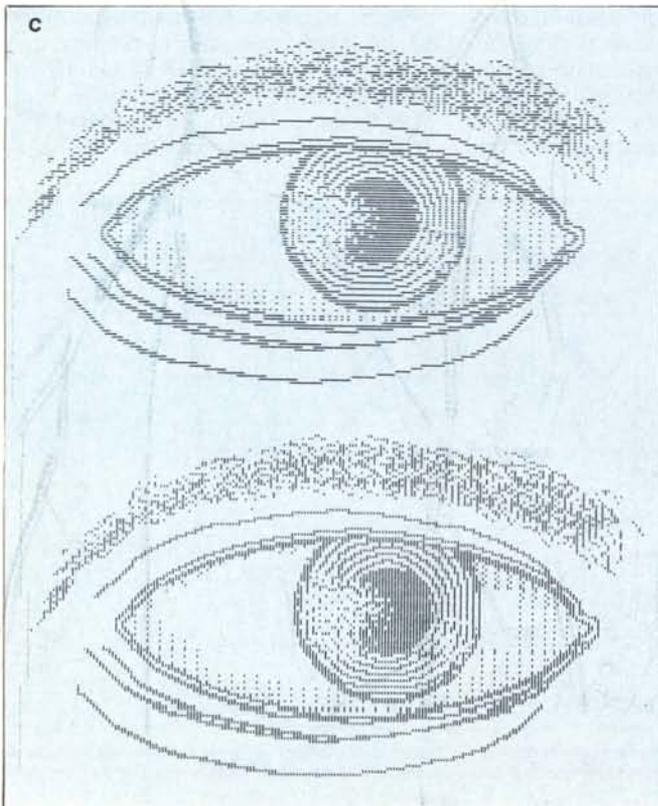
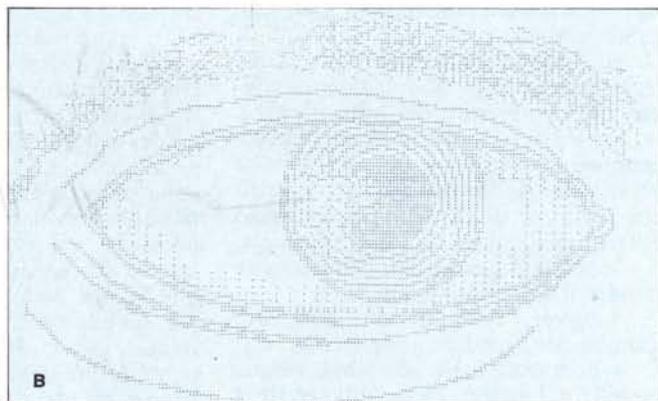
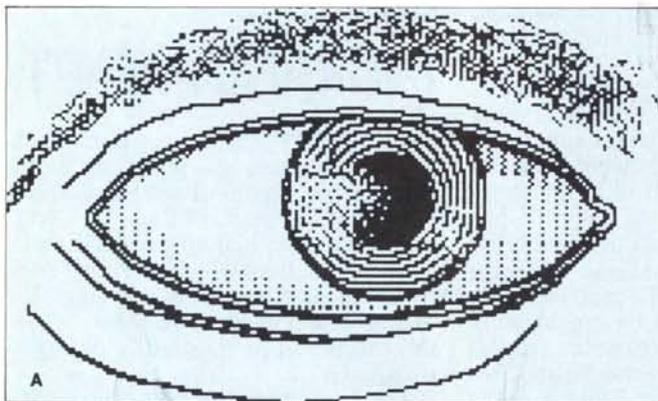
nella maggior parte dei casi, il problema).

Innanzitutto appare ovvio che è consigliabile lavorare su immagini scalate secondo valori interi, vale a dire che è preferibile creare immagini doppie, triple o comunque multiple dell'originaria. Ciò è abbastanza semplice se si tiene conto che per raddoppiare, ad

esempio, l'immagine, è sufficiente duplicare il disegno iniziale (attraverso operazioni di copia, o più semplicemente tramite il tasto Opzione) e sistemarlo in corrispondenza di uno spigolo dell'originario. Basterà poi selezionare questo e deformarlo fino a coprire anche l'altro (tasto Option Shift) per avere l'immagine perfettamente in scala (è preferibile inquadrare il disegno in un rettangolo per procedere con maggiore sicurezza).

In tal caso, essendo l'immagine di dimensioni doppie, ogni punto del retino avrà dimensioni di 4 pixel, dando

all'immagine un aspetto piuttosto rozzo e poco curato. Sarebbe meglio cancellarne 3 e partire da una immagine a tessitura piuttosto fine. Niente paura, non si lavora su immagini ingrandite! Selezionare l'intera immagine (rettangolo), scegliere inverti, selezionare la linea punteggiata, scegliere il quinto pattern della linea superiore partendo da sinistra, coprire la figura con un rettangolo pieno tenendo premuto il tasto comando, infine selezionare di nuovo l'immagine e scegliere inverti: otterremo l'effetto desiderato, con un disegno a trama finissima, ma estre-



A - Ingrandimento in scala 2:1.

B - Immagine ottenuta coprendo quella precedente con un retino in inverse (vedi testo).

C - Successive fasi di costruzione dell'immagine ombreggiata.

mamente ben dettagliato e preciso nei particolari.

L'immagine, a questo punto, è tanto fine da rendere ben poca cosa quando stampata. Niente paura: selezionare col laccio e ricopiarla più volte con sfalsamento di qualche punto: l'operazione richiede una certa attenzione ed un po' di pratica, per cui è consigliabile tenere una copia della figura originale negli appunti: un po' di prove permetteranno di ottenere il miglior risultato. Semplice, no?

MC



GRUPPO DISITACO



GRUPPO DISITACO

COMMODORE

Commodore 64 NEW	440.000
Commodore 64 NEW + Registratore	499.000
Commodore 128	560.000
Commodore 128 D	1.250.000
Drive 1541	450.000
Drive 1571	620.000
Monitor 1702	499.000
Monitor 1901	730.000
Stampante 803 i Tratt.	460.000
Stampante DPS 1101 132 colonne	699.000
Stampante EPSON LX 90	799.000
Registratore compatibile 64/128	65.000
Per Accessori e Software COMMODORE telefonare	

ATARI 520 STM / 1040 ST

Tastiera 520 STM	1.110.000
Tastiera 520 ST PLUS	1.298.000
Tastiera 1040 STF	2.180.000
Drive SF 354	349.000
Drive SF 314	495.000
Monitor SM 124	349.000
Monitor Colore SC 1424	860.000

PACCHETTI ATARI

1 520 STM + 1 SF 354	1.400.000
1 520 ST PLUS + 1 SF 354 + 1 SC 1424	2.240.000
1 520 ST PLUS + 2 SF 354 + 1 SM 124	2.180.000
1 1040 STF + 1 SM 124	2.480.000

LINEA MS DOS PC COMPATIBILI

**PC COMPATIBILE
MULTITEK POPULAR AM 500**
L. 1.790.000 + IVA

**PC COMPATIBILE
MULTITEK POPULAR AM 500
MONITOR COLORE**
L. 2.250.000 + IVA

**PC AT COMPATIBILE
MULTITEK ADVANCE AM 900
COMPLETO**
L. 5.700.000 + IVA

**OLIVETTI M24
256 K + 2 DISK 360 K**
L. 3.900.000 + IVA

**OLIVETTI M24
1 DRIVE 360 K + HD 20 MB**
L. 5.500.000 + IVA

**PC COMPATIBILE
COMMODORE PC 10**
L. 1.790.000 + IVA

Tutte le configurazioni comprendono: CPU 256 K, 2 Drive 360 K, Monitor monoc., DOS, Manuali, Garanzia 1 anno.

PERIFERICHE PC COMPATIBILI

Disco Rigido Interno 10 MB Slim completo	1.200.000 + IVA
Disco Rigido Interno 20 MB Slim completo	1.450.000 + IVA
Disco Rigido Interno 30 MB Slim completo	2.600.000 + IVA
Disco Rigido Esterno 20 MB completo	1.850.000 + IVA
Back up a nastro Irwin 10 MB interno	1.699.000 + IVA
Back up a nastro Tall grass 20 MB Esterno	1.990.000 + IVA
Espansione 640 K RAM Software	250.000 telefonare

**FAVOLOSO SISTEMA
VIDEOSCRITTURA AMSTRAD**
Monitor, Drive 1 MB,
Tastiera 256 K, Stampante
L. 1.390.000 + IVA

SINCLAIR

Sinclair QL versione italiana	699.000
Stampante QL 1000	799.000
Monitor QL 14" colore	650.000
Interfaccia parallela Miracle	99.000
Disk drive per QL 1 MB con interfaccia, alimentatore e utilities	749.000
Drive 2 aggiuntivo 1 MB	449.000
Mouse per QL	210.000
Valigetta porta QL ZX Spectrum 2 (reg. incorporato) telefonare	140.000
Kit tastiera plus	95.000
Int. joystick Fempson	55.000
Int. Centronics Kerpson	99.000
Tastiera professionale SAGA	95.000
Per accessori e software Sinclair telefonare	

COMMODORE AMIGA

Commodore AMIGA 512 K - Monitor colore	2.790.000 + IVA
Drive Aggiuntivo 1 MB	telefonare
Kit simulazione MS DOS per AMIGA	telefonare

STAMPANTI

Nannesman Tally MT 80 Pus	640.000
Nannesman Tally MT 80 FC	680.000
Mannesman Tally MT 85	960.000
Mannesman Tally MT 86	1.180.000
Mannesman Tally MT 290	1.899.000
Mannesman Tally MT 20	999.000
Epson LX 80 F/T	749.000
Epson LX 80 PC	799.000
Epson FX 85	1.199.000
Epson FX 105	1.499.000
Seikosha GP 50A	199.000
Seikosha GP 50AS	260.000
Seikosha GP 500 AS	380.000
Seikosha GP 500 VC	380.000
Citizen 120B IBM	680.000
Citizen 1200 CBM/64/128	680.000

MONITORS

Mon. Hantarex X11	210.000
Mon. Hantarex B12	230.000
Mon. colore 1702	499.000
Mon. colore 1901	730.000
Mon. QL colore	650.000
Mon. Multitek RGB IBM compatibile	320.000
Mon. Hantarex colore per 128	580.000

PUNTI VENDITA GRUPPO DISITACO

COMPUTRON SHOP-TRON s.r.l.

L.go Forano, 7/8 - Tel. 8391556
(Salario, Parioli)

A.D.M. s.r.l.

Via Tacito, 80 - Tel. 3612959
(Prati, S. Pietro)

COMPUTER FRIEND

Via Antonazzo Romano, 3 -
Tel. 393321

(Flaminio, Monte Mario)

2 M ELETTRONICA s.r.l.

Via Britannia, 17 - Tel. 755935
(S. Giovanni, Appio Latino)

BIT HOUSE s.r.l.

V.le Kennedy, 100 - Tel. 9005815
(Monterotondo)

DISITACO s.r.l.

Via Massacciucoli, 25/A
Tel. 8390100

(Trieste, Nomentano)

I.C.P. s.r.l.

Viale Cetra, 24-26 -
Tel. 0773-486977

(Latina)

DIREZIONE SERVIZI COMMERCIALI:

Via Arbia, 62 - Tel. 857607-8440736



* Per accessori ed altri prodotti telefonare

A. Cerqua

Tu che se un homo sapiens lo sai, i computer possono fare le cose più incredibili. Però di solito se ne stanno belli tranquilli al loro posto, senza spostarsi di un millimetro. Modulus, no.

Lui si diverte soltanto se gli fai fare quattro passi.

Modulus, infatti, non è solo cervello, ma voce, occhi, braccia, sensori e ruote: tutto quello che occorre

ad un personal robot per essere rivoluzionario.

In cosa consiste la sua rivoluzione?

Nelle sue prestazioni, innanzitutto, che erano incredibili fino ad oggi per un robot delle sue dimensioni e del suo costo. E nella sua modularità, che consente l'acquisto successivo di elementi componibili fino a raggiungere la configurazione di un android. Ognuno di questi step successivi, naturalmente, ha la sua specificità ed una sua ragion d'essere autonoma.

Se hai un Home Computer, un po' di pratica e molta fantasia, la "Versione Base" fa proprio per te. È pre-programmata e capace di muoversi a due velocità diverse rilevando gli urti e fermandosi, tracciare un percorso preordinato con precisione perfetta, disegnare con pennarelli, segnalare il suo

funzionamento tramite un display, dialogare con un Home Computer via cavo o etere grazie ad un apparecchio di comunicazione a radiofrequenza. Insomma, per dirla in due parole, la "Versione Base" sarà la tua raffinata periferica semovente.

Se ti fa piacere sapere in anticipo che stasera pioverà, allora "Service & Security Robot" è il tuo compagno ideale.

In questa configurazione Modulus dispone, oltre che della base, anche di una serie di moduli

a spicchio, ciascuno dei quali può contenere diversi componenti hard/software. Questa "torta tecnologica" rende possibile, perciò, una crescita il cui limite evolutivo non è definibile. Attualmente sono disponibili una stazione meteorologica, capace di prevedere le condizioni atmosferiche

(se messa in relazione con un computer); un sistema SONAR per



MODULUS. L'AMICO

la misura di distanze e l'individuazione di sorgenti di luce, calore, rumore e l'inseguimento di umani; un sistema VOCE per una sintesi vocale di elevata qualità e per il riconoscimento di suoni di comando; un sistema SICUREZZA per la rilevazione di fughe di gas, acqua e fumo. Inoltre una CPU a 16 bit dotata di 128 Kbyte Ram, 128 Kbyte Rom, 16 Kbyte Ram con alimentazione tampone e cartuccia Rom per i programmi applicativi, rende possibile il funzionamento di tutti i sistemi anche svincolati da qualsiasi Personal Computer.



centomila altre in più, perché ha anche due braccia, una testa e due occhioni molto, molto espressivi.

Anche se non sei uno scienziato folle, però, potrai ricavare grande piacere dalla compagnia di "Moddy", che con la sua voce

o con il suo monitor ha mille cose da raccontarti mentre ti dà una mano nelle occupazioni domestiche, nei tuoi hobbies preferiti o nelle tue attività più impegnative.

E adesso, homo sapiens, per saperne di più non ti resta che recarti nei migliori negozi di elettronica ed HiFi, oppure ritaglia il coupon qui sotto e riceverai ampio materiale illustrativo.

Ma quello che il "Service & Security Robot" ha di meglio è la possibilità dell'inserzione di un braccio. Questo, oltre ad essere caratterizzato da un'ampia possibilità di movimento, una velocità nettamente superiore a quelle fornite dai robot della precedente generazione ed una precisione elevata, dispone di una particolare controllo della forza di presa sulla mano.

Mica male! Oltre a dirti che tempo farà, Modulus ti porge anche l'ombrello!

Se poi sei nato sotto una radice quadrata, allora ne farai delle belle con "Moddy". La versione più evoluta di Modulus può fare tutto quello che fanno le precedenti e



Desidero ricevere maggiori informazioni su Modulus

Nome _____

Via _____

Cap. _____ Città _____

MC/9/86



A NEW ERA IN HOME ROBOTICS

ROBOT MODULUS È PRODOTTO E DISTRIBUITO DA SIRIUS.
MILANO FIORI PALAZZO F2 - 20094 ASSAGO (MI) ITALY - TEL. (02) 8245321 - TELEX 325584

DELL'HOMO SAPIENS.