È disponibile, presso la redazione, il disco con i programmi pubblicati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 208.

Trucchi e routine per programmatori

di Enrico e Gianfranco Drudi,
Viserba di Rimini (Forlì)

Per semplificare il lavoro di chi intende scrivere un «programmone» per 64 o 128 e vuole evitare listati chilometrici, abbiamo studiato diverse routine che risulteranno semplici, brevi e utili, ma che purtroppo, per la solita mancanza di spazio, non abbiamo potuto pubblicare completamente.

Queste vanno semplicemente accodate al programma base (ad esempio a partire dalla riga 60000).

Quando se ne presenta la necessità, la routine va chiamata con un semplice gosub.

Tutti le variabili utilizzate sono particolari (solo a pochi «desperados» viene in mente di usare XW, XK$, ZZ$, ...); in questo modo non si creano catastrofi riutilizzando o azzerando variabili già usate dalle procedure principali.

La routine "CODICE SEGRETO" consente di precludere l'accesso al programma a coloro che non conoscono la sequenza di caratteri specificata alla riga 60006 (il codice 0123456789 è stato scelto arbitrariamente e può essere sostituito da una qualsiasi sequenza di caratteri).

Il numero di caratteri componenti di codice segreto può essere modificato intervenendo sul FOR-NEXT della riga 60003.

La routine STAMPA, anche se a molti può sembrare ridicola, è utile se nel programma vengono utilizzate molto spesso richieste di stampa di dati, elenchi, prospetti, ecc..

Dopo la chiamata (GOSUB) ritorna la variabile ZZ che conterrà il valore zero se la stampa non va effettuata, oppure il valore uno in caso contrario. La scritta "STAMPA (S/N)" compare in basso a destra e premuto il tasto "N" o "S", essa scompare. Così eventuali schermate impostate precedentemente non verranno alterate.

La routine di PRINT SCREEN, che funziona in quaranta colonne (anche nel C64), permette di stampare l'intera videata rimediando così ad una piccola manchevolezza della Commodore.

Prima del GOSUB vanno impostate le due variabili "MX" e "MY", che indicano la riga di inizio e quella di fine delimitando così la parte di schermo da stampare. Ovviamente MY deve essere maggiore od uguale a MX ed entrambe devono essere comprese tra 1 e 25 (le righe dello schermo).

Codice segreto

```
60000 REM CODICE SEGRETO
60001 ZW=0:ZW$="":ZZ$="":REM AZZERAMENTO VARIABILI
60002 ZZ=0:ZW=ZW+1:IFZW=5THENSYS65341: REM CONTA NUMERO DI TENTATIVI E RESETTA
60003 FORZZ=1TO10:REM NUMERO DI CARATTERI DI CUI E' FORMATO IL CODICE
60004 GET/ZZ$:ZW$=ZW$+ZZ$: REM RICHIEDE IMMISSIONE DI CARATTERI
60005 NEXTZZ : REM CHIUDE IL CICLO
60006 IF ZW$="0123456789"THENZW=0:ZZ=0:ZW$="":ZZ$="":RETURN : REM CONTROLLA L'ES
ATTEZZA DEL CODICE E RITORNA IN CASO AFFERMATIVO
60007 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGH
T)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)
(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RVS)CODICE ERRATO(OFF)":REM   SCRIVE
UN MESSAGGIO DI CODICE ERRATO (PER LE 80 COLONNE BISOGNA VARIARLO)
60008 =
 15 : REM ATTENDE PER PERMETTERE DI LEGGERE IL MESSAGGIO
60009 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGH
T)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)
(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)                ": REM    CANCELLA IL MES
SAGGIO APPENA SCRITTO PER NON LASCIARE 'NESSUNA TRACCIA'
60010 ZW$="":ZZ$="":GOTO60002 : REM AZZERA LE VARIABILI E CHIEDE UN ALTRO CODICE
60011 REM
60012 REM
60013 REM ************************************************************
60014 REM ELENCO DELLE VARIABILI UTILIZZATE DALLA ROUTINE 'CODICE SEGRETO'  :
60015 REM ************************************************************
60016 REM
60017 REM ZW , ZW$ , ZZ , ZZ$ .
```

Stampa

```
60100 REM STAMPA (S/N)
60101 ZZ=0:PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT
)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(
RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)STAMPA (S/N)":REMVISUALIZZA
 IL MESSAGGIO
60102 GET/ZZ$:IFZZ$<>"S"ANDZZ$<>"N"THEN60102 : REM RICHIEDE L'IMMISSIONE DI UN C
ARATTERE (S/N)
60103 PRINT"(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGH
T)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)
(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)              ": REM    CANCELLA I
L MESSAGGIO DI RICHIESTA
60104 IF ZZ$="S"THENZZ=1:ZZ$="":RETURN : REM IMPOSTA LA VARIABILE CHE CONTIENE
LA RISPOSTA (ZZ)
60105 ZZ$="":ZZ=0:RETURN : REM AZZERA LE VARIABILI E RITORNA
60106 REM
60107 REM
60108 REM ******************************************************
60109 REM ELENCO DELLE VARIABILI DELLA ROUTINE 'STAMPA (S/N)' :
60110 REM ******************************************************
60111 REM
60112 REM ZZ , ZZ$
```

Se questi valori non vengono impostati la routine assume per DEFAULT MX = 3 ed MY = 23; valori modificabili intervenendo nella riga 60301.

La routine di INPUT CONTROLLATO permette di inserire in variabili stringa dei dati immessi da tastiera in modo completamente controllato.

I tasti cursore e molti altri tasti che sono «fastidiosi» durante i normali INPUT, sono inutilizzabili e quindi le possibilità di errore sono limitate; inoltre proteggere un programma diventa più semplice ed efficace.

Prima di effettuare il GOSUB bisogna definire cinque variabili:

— **VJ**, che definisce il tipo di variabile desiderata.

Print Screen

```
60200 REM PRINT SCREEN
60201 IFMX=0ANDMY=0THENMX=3:MY=23:REM IMPOSTA TUTTO SCHERMO SE NON DEFINITO
60202 OPEN4,4:FORLA=MXTOMY:WX=0:FORLB=1TO39:GOSUB60204:REM APRE IL FILE SU STAMP
ANTE E LEGGI LO SCHERMO UN CARATTERE ALLA VOLTA
60203 NEXTLB:WX=1:LB=40:GOSUB60204:NEXTLA:PRINT#4:CLOSE4:MX=0:MY=0:WX=0:LB=0:LA=
0:XW=0:RETURN : REM CHIUDE I CICLI DELLA LETTURA DEL VIDEO, AZZERA ED ESCE
60204 XW=PEEK(1023+(LB+((LA-1)*40))): REM TRASFORMA I CARATTERI DELLO SCHERMO IN
 CODICI STAMPABILI
60205 IFXW>=0ANDXW<=31THENXW=XW+64:GOTO60215:REM   EFFETTUA CONTROLLI SUI
60206 IFXW>=32ANDXW<=63THEN60215:REM               CARRATTERI LETTI PER
60207 IFXW>=64ANDXW<=95THENXW=XW+32:GOTO60215:REM  POTERLI STAMPARE SU
60208 IFXW>=96ANDXW<=127THENXW=XW+64:GOTO60215:REM CARTA (DA PEEK VENGONO
60209 IFXW>=128ANDXW<=159THENXW=XW-64:REM          TRASFORMATI IN CHR$).
60210 IFXW>=160ANDXW<=191THENXW=XW-128:REM
60211 IFXW>=192ANDXW<=223THENXW=XW-96:REM
60212 IFXW>=224ANDXW<=255THENXW=XW-64:REM
60213 IFWX=1THENPRINT#4,CHR$(18)CHR$(XW)CHR$(146):RETURN:REM STAMPA IL QUARANTES
IMO CARATTERE IN MODO 'REVERSE'
60214 PRINT#4,CHR$(18)CHR$(XW)CHR$(146);:RETURN:REM STAMPA I PRIMI 39 CARATTERI
IN MODO 'REVERSE'
60215 IFWX=1THENPRINT#4,CHR$(XW):RETURN:REM STAMPA IL QUARANTESIMO CARATTERE IN
MODO NORMALE
60216 PRINT#4,CHR$(XW);:RETURN:REM STAMPA I PRIMI 39 CARATTERI IN MODO NORMALE
60217 REM
60218 REM TUTTI I VALORI DI XW COMPRESI TRA 0 E 127 VENGONO INVIATI PER LA STAMP
A ALLA RIGA 60215
60219 REM TUTTI I VALORI DI XW COMPRESI TRA 128 E 255, IN QUANTO IN 'REVERSE', V
ENGONO INVECE INVIATI ALLA RIGA 60213
60220 REM ALLE RIGHE 60213 E 60214 VENGONO INVIATI ALLA STAMPANTE IL CHR$(18) E
IL CHR$(146) (ABILITA E DISABILITA MODO 'REVERSE')
60221 REM
60222 REM
60223 REM ************************************************
60224 REM ELENCO VARIABILI DELLA ROUTINE 'PRINT SCREEN' :
60225 REM ************************************************
60226 REM LA , LB , MX , MY , WX , XW
```

*** VJ = 5 : VC = 12 : VH$ = "NUMERO TELEFONICO" : VX = 19 : VY = 5: GOSUB 60400 : NT$ = VC$: VC$ = "".

Il numero telefonico risulterà contenuto nella variabile "NT$".

I tre asterischi rappresentano il numero di riga all'interno del programma.

La routine di ORDINAMENTO ALFABETICO permette di ordinare alfabeticamente (!) un certo numero di variabili stringa contenute in XL$() dimensionato. Il numero totale di variabili deve essere contenuto in XN e, chiaramente, la variabile XL$() deve essere dimensionata almeno ad XN elementi.

Al ritorno le variabili saranno ordinate alfabeticamente e contenute sempre in XL$(). Ovviamente più le variabili saranno lunghe e più tempo ci metterà la routine ad ordinarle.

La routine di CONVERSIONE DA

— **VY** e **VX**, che definiscono le coordinate (riga e colonna) in cui si posizionerà il cursore alla richiesta di immissione.

— **VC**, che definisce il totale dei caratteri che devono essere richiesti.

— **VH$**, che definisce la stringa utilizzata come domanda (spazi compresi) alla sinistra del punto in cui si posiziona il cursore alla richiesta dei dati. Il numero di caratteri di VH$ [LEN(VH$)], deve essere minore di una unità al valore di VX.

VJ può definire cinque tipi di variabili:

VJ = 1 per variabili alfanumeriche (numeri + lettere)
VJ = 2 per variabili composte da qualsiasi carattere (segni grafici compresi)
VJ = 3 per variabili solamente numeriche (numeri da 0 a 9)
VJ = 4 per variabili solo letterali
VJ = 5 per variabili che rappresentano date e numeri telefonici (numeri da 0 a 9 e il carattere /).

Una volta definite le cinque variabili si richiama la routine (GOSUB).

A immissione effettuata la stringa desiderata è contenuta nella variabile "VC$"; occorrerà quindi trasferire il suo contenuto in un'altra variabile (quella da utilizzare durante tutto il programma) e azzerare VC$ per poter riutilizzare la routine.

Segue un esempio per chiarire eventuali dubbi: supponendo che la routine inizi alla riga 60400 e si desideri richiedere un numero telefonico la riga da editare risulterà la seguente:

Input controllato

```
60300 REM CONTROLLO INPUT
60301 PRINT"(HOME)";:FORVW=1TOVX:PRINT"(DOWN)";:NEXT:REM POSIZIONA IL CURSORE
60302 PRINTVH$"<(RVS)";:FORVT=1TOVC+1:PRINTCHR$(32);:NEXT:PRINT"(OFF)>":REM DELI
MITA IL CAMPO DI INPUT
60303 FORVU=1TOVC+1: REM EFFETTUA UN CICLO PER LA LUNGHEZZA DELL'INPUT
60304 GETVB$:IFVB$=""THEN60304:REM RICHIEDE UN CARATTERE ALLA VOLTA
60305 IFVB$=CHR$(29)THEN60304:REM NON CONSIDERA CURSORE DESTRO
60306 IFVB$=CHR$(145)THEN60304:GOTOIFVB$=CHR$(27)THEN60304:REM NON CONSIDERA CUR
SORE ALTO
60307 IFVB$=CHR$(157)THEN60304:GOTOIFVB$="
"THEN60304:REM NON CONSIDERA CURSORE SINISTRO
60308 IFVB$=CHR$(148)THEN60304:GOTOIFVB$="(ENSH)"THEN60304:REM NON CONSIDERA INS
T
60309 IFVB$=CHR$(141)THEN60304:REM NON CONSIDERA SHIFT+RETURN
60310 IFVB$=CHR$(160)THEN60304:REM NON CONSIDERA SHIFT+SPACE
60311 IFVB$=CHR$(19)THEN60304:REM NON CONSIDERA HOME
60312 IFVB$=CHR$(147)THEN60304:REM NON CONSIDERA CLR
60313 IFVB$=CHR$(17)THEN60304:REM NON CONSIDERA CURSORE BASSO
60314 IFVB$=CHR$(14)THEN60304:REM NON CONSIDERA MINUSCOLO
60315 IFVB$=CHR$(20)ANDVU=1THENVB$="":GOTO60304:REM NON CONSIDERA DEL SE E' IL P
RIMO CARATTERE IMMESSO
60316 IFVB$=CHR$(13)ANDVU=1THEN60304  :REM NON CONSIDERA RETURN SE E' IL PRIMO C
ARATTERE IMMESSO (NON VUOLE STRINGHE NULLE)
60317 IFVB$=CHR$(13)THENPRINTCHR$(5):GOTO60332 :REM SCRIVE IN BIANCO SE SI PREME
 RETURN
60318 IFVB$=CHR$(20)ANDLEN(VC$)<VCTHENVC$=LEFT$(VC$,LEN(VC$)-1):PRINT"(HOME)":FO
RVW=1TOVX:PRINT"(DOWN)";:NEXT:PRINTTAB(VY)"(RVS)"VC$"(RVS)- (OFF)":VU=VU-1:GOTO6
0304:REM VEDI FONDO
60319 IFVB$=CHR$(20)ANDLEN(VC$)=VCTHENVC$=LEFT$(VC$,LEN(VC$)-1):PRINT"(HOME)":FO
RVW=1TOVX:PRINT"(DOWN)";:NEXT:PRINTTAB(VY)"(RVS)"VC$"(RVS)-(OFF)":VU=VU-1:GOTO60
304:REM VEDI FONDO
60320 IFLEN(VC$)=VCANDVB$<>CHR$(13)THEN60304 :REM RAGGIUNTA MASSIMA LUNGHEZZA ST
RINGA. NON RICEVE PIU' CARATTERI ECCETTO DEL E RETURN
60321 IFVJ=1ANDASC(VB$)=32THEN60330:REM                    CONTROLLA CHE I
60322 IFVJ=2ORVJ=4THENIFASC(VB$)=32THEN60330:REM           CARATTERI IMMESSI
60323 IFVJ=1THENIFASC(VB$)<48ORASC(VB$)>90THEN60304:REM    POSSANO APPARTENERE
60324 IFVJ=1THENIFASC(VB$)=>58ANDASC(VB$)=<64THEN60304:REM AL TIPO DI INPUT
60325 IFVJ=2THEN60330:REM                                  RICHIESTO
60326 IFVJ=3THENIFASC(VB$)<48ORASC(VB$)>57THEN60304:REM    (DATA, NUMERICO,
60327 IFVJ=4THENIFASC(VB$)<65ORASC(VB$)>90THEN60304:REM    ALFANUMERICO,
60328 IFVJ=5THENIFASC(VB$)<47ORASC(VB$)>57THEN60304:REM    ECC.).
60329 IFVJ=6THENIFASC(VB$)<45ORASC(VB$)>57THEN60304:REM
60330 VC$=VC$+VB$:PRINT"(HOME)":FORVW=1TOVX:PRINT"(DOWN)";:NEXT:IFLEN(VC$)<VCTHE
NPRINTTAB(VY)"(RVS)"VC$"-(OFF)":GOTOPRINTTAB(VY)"(RVS)"VC$"(OFF)":REM VEDI FONDO
60331 NEXT:REM CHIUDE IL CICLO
60332 FORVD=1TOVC-LEN(VC$):VC$=VC$+" ":NEXT:VC$=LEFT$(VC$,VC):RETURN:REM AGGIUNG
E OPPORTUNI SPAZI VUOTI PER COMPLETARE LUNGHEZZA STRINGA
60333 REM
60334 REM RIGA 60318 E 60319 : DIMINUISCE DI UN CARATTERE LA STRINGA E LA SCRIVE
 SU VIDEO
60335 REM
60336 REN RIGA 60330 : AGGIUNGE ALLA STRINGA DI INPUT IL CARATTERE IMMESSO E LA
VISUALIZZA SU VIDEO
60337 REM
60338 REM
60339 REM ************************************************
60340 REM ELENCO VARIABILI DELLA ROUTINE 'INPUT CONTROLLATO' :
60341 REM ************************************************
60342 REM
60343 REM VB$ , VC , VC$ , VD , VH$ , VJ , VT , VU , VW , VX , VY
```

Ordinamento alfabetico

```
60400 REM ORDINAMENTO ALFABETICO
60401 FORXI=1TOXN-1 : REM INIZIA UN CICLO DA 1 FINO AL NUMERO MASSIMO MENO UNO
60402 FORXJ=XI+1TOXN: REM INIZIA IL SECONDO CICLO IN BASE AL PRIMO
60403 IFXL$(XJ)>=XL$(XI)THEN60405 : REM CONTROLLA LE VARIABILI
60404 XQ$=XL$(XJ):XL$(XJ)=XL$(XI):XL$(XI)=XQ$:REM SOSTITUISCE
60405 NEXTXJ:NEXTXI: REM CHIUDE I DUE CICLI
60406 RETURN : REM ESCE DALLA ROUTINE
60407 REM
60408 REM
60409 REM ************************************************************
60410 REM ELENCO VARIABILI DELLA ROUTINE 'ORDINAMENTO ALFABETICO' :
60411 REM ************************************************************
60412 REM
60413 REM XI . XJ . XL$() . XN . XQ$
```

"DATA A NUMERO" consente di trasformare una data (nel formato GG/MM) in un numero (compreso tra 1 e 366).

In questo modo, se si utilizzano archivi di grosse dimensioni, si possono risparmiare preziosi caratteri.

Bisogna inserire la data (sempre nel formato GG/MM) nella variabile TD$ e poi chiamare la routine (con un gosub).

Per una maggiore accuratezza vi è la

Conversione da data a numero

```
60500 REM CONVERSIONE DA DATA A NUMERO
60501 TA=VAL(LEFT$(TD$,2)):TB=VAL(MID$(TD$,4,2)):REM DIVIDE LA DATA IN 2 VARIABI
LI NUMERICHE INDICANTI IL GIORNO E IL MESE
60502 IFTB>12ORTB<1THENXW=1:RETURN:REM CONTROLLA LA VALIDITA' DEL MESE
60503 IFTA>31ORTA<1THENXW=1:RETURN:REM CONTROLLA LA VALIDITA' DEL GIORNO
60504 IFTB=2ANDTA>28+XYTHENXW=1:RETURN : REM CONTROLLA LA VALIDITA' DEL GIORNO
60505 IFTA=31THENIFTB=4ORTB=11ORTB=6ORTB=9THENXW=1:RETURN:REM CONTROLLA I MESI D
A 31 GIORNI
60506 IFTB=1THENTD=TA:GOTO60518: REM        CALCOLA, A SECONDA DEI MESI,
60507 IFTB=2THENTD=31+TA:GOTO60518:REM      IL NUMERO DA SOMMARE ALLA
60508 IFTB=3THENTD=59+TA+XY:GOTO60518:REM   VARIABILE TD CHE RAPPRESENTA
60509 IFTB=4THENTD=90+TA+XY:GOTO60518:REM   IL NUMERO ( DA 1 A 366 )
60510 IFTB=5THENTD=120+TA+XY:GOTO60518:REM CORRISPONDENTE ALLA DATA
60511 IFTB=6THENTD=151+TA+XY:GOTO60518:REM RICEVUTA.
60512 IFTB=7THENTD=181+TA+XY:GOTO60518:REM 'XY' VALE 1 SE L'ANNO
60513 IFTB=8THENTD=212+TA+XY:GOTO60518:REM E' BISESTILE,
60514 IFTB=9THENTD=243+TA+XY:GOTO60518:REM ALTRIMENTI VALE 0.
60515 IFTB=10THENTD=273+TA+XY:GOTO60518:REM
60516 IFTB=11THENTD=304+TA+XY:GOTO60518:REM
60517 TD=334+TA+XY:REM
60518 TA=0:TB=0:XY=0:XW=0:RETURN : REM AZZERA LE VARIABILI E FINISCE
60519 REM
60520 REM
60521 REM ************************************************************
60522 REM ELENCO VARIABILI DELLA ROUTINE 'CONVERSIONE DA DATA A NUMERO' :
60523 REM ************************************************************
60524 REM
60525 REM
60526 REM TA . TB . TD . TD$ . XW . XY
```

dell'anno in questione. Il numero da convertire in data deve essere contenuto nella variabile "XX" e la data ottenuta tramite la routine sarà contenuta nella variabile "TD$".

La routine "CONTROLLO DATA" permette di effettuare tutti i controlli necessari per essere sicuri della esattezza delle date con cui si lavora. Questa routine è particolarmente indispensabile prima di effettuare una chiamata alle due routine di conversione, poiché queste ultime non potrebbero lavorare correttamente utilizzando dati sbagliati (31 giugno, 30 febbraio, ecc.).

La data da controllare deve essere nel formato GG/MM e deve essere contenuta nella variabile "TD$". Alla fine del controllo la routine restituirà la variabile "XW", che conterrà il valore "1" se

variabile "XY" che contiene il valore "1" se l'anno è bisestile, il valore "0" se non lo è.

Questa variabile va «preparata» prima della chiamata alla routine.

Per non dover chiedere all'operatore se l'anno è bisestile (ogni volta che si effettua una conversione o un controllo) abbiamo pensato di utilizzare un piccolissimo file sequenziale chiamato appunto "XY" che contiene "1" se l'anno è bisestile, altrimenti "0". Non è veramente difficile costruire un'opzione che consenta di modificare tale valore nel file sequenziale (qualche PRINT, un INPUT, un OPEN-PRINT# e CLOSE).

La routine automaticamente utilizzerà il file "XY" per sapere se l'anno è bisestile.

Con questa routine, inoltre, si può calcolare la differenza in giorni tra date. Al ritorno dalla routine si potrà utilizzare la variabile "TD", che rappresenta proprio la data specificata dalla variabile "TD$", convertita però in numero.

La routine di CONVERSIONE DA "NUMERO A DATA" opera la conversione opposta alla routine precedente, trasformando un numero (da 1 a 366) in una data (GG/MM). Anche questa routine (come la precedente) utilizza il file "XY" per conoscere il numero di giorni

```
E per finire, ecco l'elenco di alcuni utili trucchi.

Blocca Run/Stop-Restore  :POKE 808,225 (237)

New finto                :POKE 2048,0:Poke 2049,0:Poke 2050,0

Repeat                   :POKE 650,128

Blocca Shift+Commodore   :POKE 657,128 (0)

Blocca Run/Stop          :POKE 788,52 (49)

List=Reset               :POKE 774,226:POKE 775,252

Restore=Reset            :POKE 792,226:POKE 793,252

Falso Ready              :POKE 808,234:?"READY.":
                          FOR I=0 TO 1 STEP 0:POKE 204,0:NEXT

Disabilita List          :POKE 775,200 (167)

Disabilita Run/Stop      :POKE 808,239 (237)

Disabilita Run/Stop-
Restore                  :POKE 808,225 (237)

Disabilita la tastiera   :POKE 649,0 (10)

Disabilita i comandi
Save e Load              :POKE 818,32 (237

List senza numeri riga   :POKE 22,35 (25)

List solo numeri riga    :POKE 774,0 (26)

Velocita' cursore        :POKE 56325,V (1 to 255) (48)

Reverse                  :POKE 199,1 (0)

Recupera List dopo New o
SYS 64738                :POKE 2050,1:SYS 42231:POKE 45,PEEK(34):
                          POKE 46,PEEK(35):CLR

Non accetta Piu' niente  :POKE 120,0 (123) (non si esce Piu'!)

Blocco totale            :? CHR$(147):POKE 1,0

Aspetta un tasto         :POKE 198,0:WAIT 198,1
```

Controllo data

```
60700 REM CONTROLLO DATA
60701 TA=VAL(LEFT$(TD$,2)):TB=VAL(MID$(TD$,4,2)) : REM DIVIDE LA DATA IN 2 VARIA
BILI NUMERICHE (MESE E GIORNO)
60702 IFMID$(TD$,3,1)<>"/"THENXW=1:RETURN : REM CONTROLLA CHE SIA PRESENTA LA BA
RRA (/)
60703 IFTB>12ORTB<1THENXW=1:RETURN : REM CONTROLLA IL MESE
60704 IFTA>31ORTA<1THENXW=1:RETURN : REM CONTROLLA IL GIORNO
60705 IFTB=2ANDTA>28+XYTHENXW=1:RETURN : REM CONTROLLA IL MESE DI FEBBRAIO
60706 IFTA=31THEN60708 : REM CONTROLLA I MESI DI 31 GIORNI
60707 XW=0:RETURN : REM SE LA DATA E' ESATTA, ESCE
60708 IFTB=4ORTB=11ORTB=6ORTB=9THENXW=1:RETURN:GOTO60707 : REM CONTROLLA I MESI
DA 31 GIORNI
60709 REM
60710 REM
60711 REM ************************************************
60712 REM ELENCO VARIABILI DELLA ROUTINE 'CONTROLLO DATA' :
60713 REM ************************************************
60714 REM
60715 REM
60716 REM TA . TB . TD$ . XW . XY
```

La routine "ORDINA 3 STRINGHE" serve per risolvere i problemi derivanti dalla necessità di mettere in ordine alfabetico 3 stringhe la cui somma in caratteri superi 255. Occorre così suddividere la stringa e aggiungere gli altri dati alle stringhe ordinate. Es.: per ordinare un archivio in base al cognome e avendo come dati di ogni cliente il cognome, il nome e l'indirizzo, se il totale dei caratteri delle 3 variabili supera i 255, si deve ricorrere all'utilizzo di questa routine. Dopo averla richiamata l'archivio sarà ordinato in base al cognome come in un normale ordinamento alfabetico, ma a fianco di ogni cognome risulteranno il nome e l'indirizzo corrispondenti.

Quindi il nome e l'indirizzo rispettivi saranno contenuti nelle variabili aventi lo stesso indice del cognome. (Se il cognome «Rossi», dopo l'ordinamento è al decimo posto, il suo nome e il suo indirizzo saranno contenuti nelle rispettive variabili al decimo posto, cioè con indice «10»).

Alla routine devono arrivare le seguenti variabili:

XT che rappresenta il totale delle stringhe da ordinare;

XA$(XT) che indica la stringa in base alla quale viene effettuato l'ordinamento alfabetico;

XB$(XT) e **XC$(XT)** che sono le rispettive variabili collegate alla prima.

Prima dell'utilizzo della routine vanno dimensionate al valore di XT le seguenti variabili: "XD$()", "XE$()", "XG$()", "XA$()", "XB$()", "XC$()", terminato l'ordinamento si possono direttamente utilizzare "XA$()", "XB$()" e "XC$()", tutte indicizzate a "XT". MC

Ordina tre stringhe

```
61000 REM ORDINAMENTO ALFABETICO DI TRE VARIABILI
61001 FOR XZ = 1 TO XT
61002 XD$(XZ) = XA$(XZ)
61003 NEXT
61004 FOR XI = 1 TO XT - 1
61005 FOR XJ = XI +1 TO XT
61006 IF XA$(XJ) >= XA$(XI) THEN 61008
61007 XQ$ = XA$(XJ) : XA$(XJ) = XA$(XI) : XA$(XI) = XQ$
61008 NEXT J : NEXT I
61009 FOR ZX = 1 TO XT
61010 FOR ZF = 1 TO XT
61011 IF XA$(XZ) = XD$(XF) THEN XE$(XZ) = XB$(XZ) : XG$(XZ) = XC$(XF) : XF = XT
61012 NEXT XF
61013 NEXT XZ
61014 FOR XZ = 1 TO XT
61015 XB$(XZ) = XE$(XZ) : XC$(XZ) = XG$(XZ) : XE$(XZ) = "" : XG$(XZ) = ""
61016 NEXT XZ
61017 REM
61018 REM
61019 REM ************************************************************
61020 REM ELENCO VARIABILI DELLA ROUTINE 'ORDINAMENTO ALFABETICO DI 3 VARIABILI:
61021 REM ************************************************************
61022 REM
61023 REM
61024 REM XA$() . XB$() . XC$() . XD$() . XE$() . XF . XG$() . XI . XJ . XQ$ . X
T . XZ . ZF . ZX
```

La data controllata era errata, o il valore "0" se invece era corretta. Basterà poi un semplice "IF" per accettarsi della correttezza della data e, se necessario, visualizzare all'operatore un messaggio del tipo: "DATA ERRATA".

La routine "CTRL ERRORI DRIVE" permette di tenere sotto controllo lo status del DRIVE; scrivendo, eventualmente, il codice, il tipo, la traccia e il settore in cui l'errore è avvenuto.
Per far funzionare la routine non occorre settare alcuna variabile.

La routine è comoda e utile se fatta girare sul Commodore 64, poiché il C128 possiede già una adeguata gestione degli errori del drive.

La routine "MODIFICA CARATTERI" permette di ridefinire parte o tutto il set di caratteri del Commodore 64.

Prima di chiamare la routine vanno impostate le seguenti variabili:

XA che deve contenere il numero totale dei caratteri da modificare.

XB(XA) che deve già essere adeguatamente dimensionato e contenere i numeri (da 0 a 255) del carattere da modificare (il CHR$).

XD (XA * 8) che deve essere già dimensionato in precedenza e contenere i numeri (da 0 a 255) che specificano le posizioni dei pixel nei singoli bit di ogni carattere. Per ogni carattere si devono chiaramente specificare otto numeri (questo spiega l'indicizzazione della variabile "XD()"). I numeri da inserire in "XD()" si calcolano in binario.

Es.: se il primo bit del carattere deve essere così: ●○○○○○●●, il numero sarà: 128 + 2 + 1 = 131. Quindi nella variabile si metterà il valore 131 (la «pallina» vuota corrisponde al pixel spento, la piena a quello acceso).

Modifica caratteri

```
60900 REM MODIFICA SET DI CARATTERI
60901 PRINT CHR$(142) : REM IMPOSTA LE MAIUSCOLE
60902 POKE 52,48 : POKE 56,48 : REM RISERVA MEMORIA PER I CARATTERI
60903 POKE 56334,PEEK(56334)AND254 : REM DISATTIVA IL TIMER DI INTERRUZIONE DELL
A TASTIERA
60904 POKE 1,PEEK(1)AND251 : REM DISATTIVA I/O
60905 FOR XI = 0 TO 2047 : POKE XI +12288,PEEK(XI+53248) : NEXT : REM COPIA LA R
OMA CARATTERE IN RAM
60906 POKE 1,PEEK(1)OR4 : REM RIATTIVA I/O
60907 POKE 56334,PEEK(56334)OR1 : REM RIATTIVA LA TASTIERA
60908 POKE 53272,(PEEK(53272)AND240)+12 : REM LEGGE I CARATTERI DA RAM
60909 FOR XL = 1 TO XA : REM APRE DUE CICLI PER LA LETTURA DEI CODICI
60910 FOR XK = 0 TO 7 : REM
60911 POKE 12288 + XB(XL)*8 + XK,XD((8*(XL-1))+XK+1) : REM MODIFICA I CARATTERI
60912 NEXT XK : REM CHIUDE I DUE CICLI
60913 NEXT XL : REM
60914 RETURN : REM ESCE DALLA ROUTINE
60915 REM
60916 REM
60917 REM ************************************************************
60918 REM ELENCO VARIABILI DELLA ROUTINE 'MODIFICA SET DI CARATTERI' :
60919 REM ************************************************************
60920 REM
60921 REM
60922 REM XA . XB() . XD() . XI . XK . XL . XX
```