business applications for the commodore 64

techniques and subroutines for business users

james hall



business applications

for the commodore 64

techniques and subroutines for business users

james hall

First published 1984 by:
Sunshine Books (an imprint of Scot Press Ltd.)
12–13 Little Newport Street
London WC2R 3LD

Copyright © James Hall, 1984

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording and/or otherwise, without the prior written permission of the Publishers.

British Library Cataloguing in Publication Data
Hall, James. 19---
Business applications for the Commodore 64.
1. Commodore 64 (Computer) 2. Business—Data processing
I. Title
651.8'4 HF5548.4.C6

ISBN 0 946408 12 2

Cover design by Grad Graphic Design Ltd.
Illustration by Stuart Hughes.
Typeset and printed in England by Commercial Colour Press, London E7.

CONTENTS

		Page
Introduction		1
1	First Considerations in Programming	3
2	Subroutines	11
3	Business Accounts System	71
4	Word Processing	123
5	Equipment Inventory	155
6	Society Mailing List	165
7	Program Lister and Ancillary Program	179
Appendix A:	Compilers and Programming Aids	189
Appendix B:	Subroutines Crib Sheet	193
Appendix C:	ASCII Character Set	197

Contents in detail

CHAPTER 1
First Considerations in Programming

Computers and business, facts of computer files, sequential files, records, files, fields, the limits of computer memory, disc or tape, preventing operator errors, program design.

CHAPTER 2
Subroutines

Ten subroutines, forming the bases of the programs in the rest of the book.

1. Variable Initialising and Program Aid — allocation of space to arrays, setting up string variables, program security, disc error messages, checking for presence or absence of a printer, merge, delete and renumber routines.

2. Numeric Input — data input routine which accepts numbers, decimal point, hash sign and RETURN key.

3. Alpha and Numeric — allows entry of a wider range of characters, using GET # or INPUT #.

4. Question and Answer — deals with common questions and answers.

5. Data Edit — allows file data to be manipulated, checked and edited.

6. Printers — dealing with various types of printers.

7. Numeric Calculation — deals with financial calculations, accounts.

8. Data Handling (tape or disc) — storage and recall of data.

9. Program Menu — gives choice of program functions.

10. File Construct — creates data record for a new or already existing file.

CHAPTER 3

Business Accounts System

Financial transactions related to supplying customers with goods and/or services, maintaining stock list linked to accounts system, accounting, final analysis — Customer/Supplier Balance program — Monthly Transaction program — Stock Control program — Transaction Analysis program.

CHAPTER 4

Word Processing

Creating and editing record files, edit data, creating and adding to text files, printout and special functions, underline, bold, super-script and sub-script, dot matrix and daisy wheel printers, automatic letter addressing, paragraph/chapter construct, setting up address file — Word Processing program — Address File program.

CHAPTER 5

Equipment Inventory

Creating a file of items of equipment, with information on location, purchase date, cost, current value — Equipment File program — Analysing the Inventory program.

CHAPTER 6

Society Mailing List

Creating a file of names and addresses, telephone numbers and interests, updating the file, producing an alphabetic list, producing an interest list — Mailing List program — Analysing the List program — Alphabetic Sort program.

CHAPTER 7

Program Lister and Ancillary Program

Overcoming the reproduction of dot matrix graphic symbols, manipulating data files, combining two record files together, renaming record files, transferring files stored on tape to disc and vice versa — Program Lister program — Ancillary program.

Program Notes

Lower case mode

Throughout the programs in this book, lower case mode has been used.

However, for the sake of clarity, all lower case letters have been reproduced as capitals, and all upper case as italic capitals.

When entering the programs on your Commodore 64, it is *essential* that you enter in lower case mode, retranslating all capitals to lower case and all italic capitals to capitals.

Amended listings

In most of the programs in this book, the subroutines from Chapter 2 are to be merged, and then the amended listing which follows keyed in. This contains those lines to be changed in the subroutines, plus the new program lines which are to be keyed in to make up the full program.

Quality of listings

Please be careful when entering the listings in this book. The quality of printing from a microcomputer printer means that some characters can look very similar — in particular, commas can look like full stops.

Note: In the computer listings, the £ sign appears as \. This should be noted in particular in the accounts programs.

Introduction

This book has been written to enable those thinking of introducing a computer system into their business to design and write their own programs. A Commodore 64 and Commodore cassette recorder (for storing programs and information) will be adequate for most purposes — a printer and/or disc drive can follow later if required. Obviously certain programs, such as word processing, would need a printer — or at least access to one (there are firms which will print your script for you).

The Commodore 64 has numerous business applications, all of them providing savings in time and effort in return for very little trouble on the part of the user. It can be used for accounts, VAT returns, stock control, automatic ordering, word processing, inventories, invoices, mailings and database applications — all of these described in detail in this book.

I wrote this book out of frustration at the lack of available literature when I started to write administrative and accounts programs to assist me in my own work. The intricacies of machine code are very interesting, but I wanted information on how I could apply my newly acquired knowledge of BASIC.

It is my aim to show you how to produce working programs, in such a way that lack of programming experience is not a serious disadvantage. I have attempted to write the programs in as plain and logical a manner as possible: some steps may be more long-winded than need be, but it is essential to produce working programs first — there is nothing to hinder you from modifying them at a later date, as your knowledge increases.

The main question you have to ask yourself is whether or not a computer system would be beneficial to your business. On completion of my own accounts system, I continued with my old manual system for some months as a comparison. The first discovery was the saving in time — my normal five days per month were reduced by the computer to half a day (and with a great deal more information than I had time to produce by the old method). The second discovery was that, despite the bad press which computers get, the computer produced far more accurate accounts than I did.

You may wonder if the time spent on designing and writing your own system is justified. If you require a system specific to your needs, then it is — otherwise, you may have to change your routine to suit a commercially-produced package. If you write your own, you will understand the pro-

grams in operation and will be in a position to sort out problems which may be encountered.

If you feel that you cannot spare the time required, then you could consider employing someone to do it for you. There are numerous amateur programmers who would be more than willing to put their knowledge to an applied task. Be prepared, however, to spend some time with your programmer as it is unlikely that you will manage to convey sufficient information to him/her the first time round. Hopefully this book will make the task so much easier.

Acknowledgements

I wish to thank Mr Greg Johnstone of Gate Microsystems for supplying the computer and for his valuable assistance with inital teething troubles, Mr Peter Calver of Supersoft and Mr Roy of Dataview for supplying their products for review and for their helpful suggestions. Also, my son, Alex, for the graphic work and my wife, Margaret, for help with the typing and valuable support.

CHAPTER 1
First Considerations in Programming

Before you can begin to design business programs, you need to understand just what it is that a computer, along with its supporting hardware such as printers and tape/disc storage systems, is capable of.

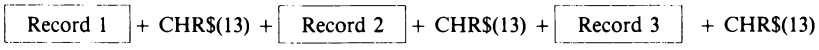
A computer can carry out mathematical functions. It can compare numbers or letters in order to decide on a course of action. It can create 'files', allowing the storage of information on magnetic tape or disc for recall at a later date. It can produce a visual display on a monitor or television screen and back up that display with a more permanent record on paper via a printer. The information stored can be analysed and manipulated so that figures can have calculations performed upon them or letters can be re-arranged in a proper format for display. If the program is good enough, anyone (even without computer experience) can be trained quickly to operate a system which is capable of dealing with most of the day-to-day running of a small business. In other words, in the computer we have a device more than adequately suited to the needs of business.

If the machine is adequate, the problem remains of the program itself. The starting point of an adequate business program is a series of subroutines designed to deal with all the repetitive functions that business applications will require. A well-tried set of reliable subroutines will enable a programmer to get on with the job of designing a program, confident in the knowledge that, when a certain subroutine is accessed, a specified function will be carried out. Freed from the necessity to worry about how each particular common task is to be performed, the programmer can concentrate on what is really important — the program as a whole.

Facts of computer files

Of all common functions of a business system, the creation, analysis and manipulation of information stored in files (on disc or tape) is probably the most important. A great deal of this book is concerned with data files; the kind of files which, on the Commodore 64, are known as 'sequential files'. In a sequential file, all the items are stored one after another on tape or disc, separated by a single carriage return character or CHR$(13), as in **Figure 1.1**.

Figure 1.1



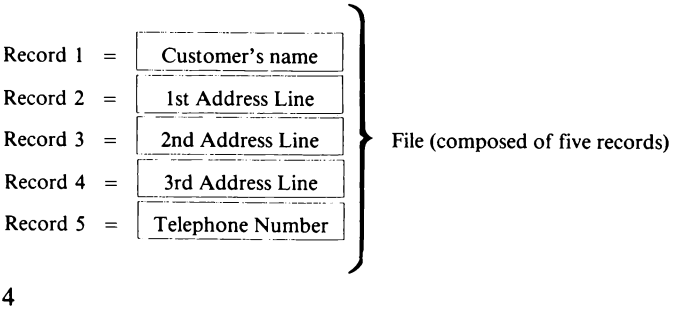
You can see from this illustration that each item is more properly called a 'record'. When data is being recalled from the file, the 64 scans the data until it finds CHR$(13) and then interprets a particular record as being complete. The records in the file are made use of by taking them into the memory using either the INPUT # or the GET # commands. INPUT # takes the whole of the record up to the carriage return character from the disc or tape, and then gives it a variable name under which it will be stored in memory. GET # picks up the contents of the record, character by character, and will not automatically recognise CHR$(13) as the end of what is to be read into memory. It will be up to the programmer to indicate that CHR$(13) is the end of the record, as in a line like:

```
GET # 1,G$: IF G$ = CHR$(13) THEN...
```

Tape as a storage medium is limited to storing data sequentially, simply because of the fact that it is a long magnetic ribbon which, while recording or replaying, travels in only one direction. Magnetic discs are more versatile, since data can be put into a variety of locations on a disc simply by moving the recording head. 'Sequential' files on a disc are not necessarily stored with records head-to-tail but, rather than delve into the intricacies of disc storage systems, it is simpler to think of them this way. In any case, no matter how the data is stored, a sequential file on a disc is treated by the 64 as if it began at the first item and went in order through to the last.

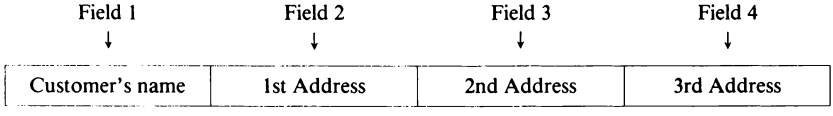
From all this, we can see how certain words are going to be used in the course of this book. 'File' will refer to a collection of records stored in order on tape or disc. Each record will be a single complete piece of information, such as the elements which go to make up the name and address of a customer, as in **Figure 1.2.**

Figure 1.2



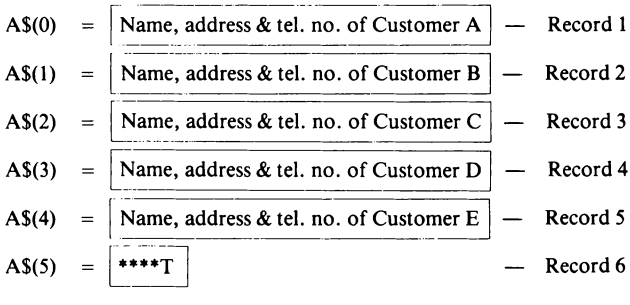
In other cases, items of information may first be joined together before being stored as a single record. In this case, each item will be known as a 'field' (see **Figure 1.3**).

Figure 1.3



When records are recalled from storage into memory, they are normally placed into elements of an array, eg A$(1), A$(2), etc. This makes it simpler for the programmer to refer to an individual record, since it will have a numbered location — record 56 will be stored in A$(56). The contents of a typical array, when used for this purpose, might look like **Figure 1.4**.

Figure 1.4



Record 6 in this figure contains what is known as the file termination code, and all the programs in this book recognise '****T' as an 'end of file marker'. There are many ways of determining when the end of a file has been reached, but the placing of a marker code which is unlikely to resemble normal data is perhaps the simplest.

Having obtained a record made up of various fields, the string handling of the 64 can be used to extract any or all of the items contained in the fields, as in the rough example in **Figure 1.5**.

Figure 1.5

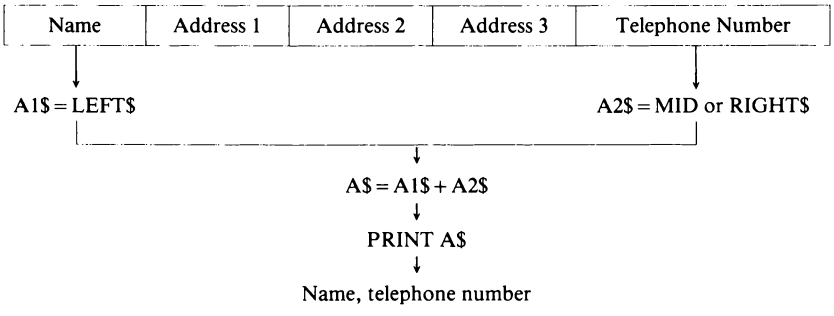
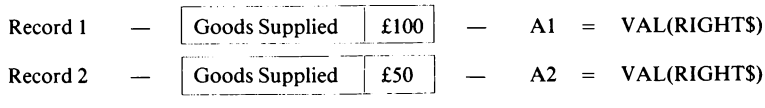


Figure 1.6



Financial records could be treated in the same way. In **Figure 1.6**, use is made of the 64's ability to transform a numeric variable, eg 100, into a string variable such as '100'. They may not look any different, but the string variable can be stored with other string variables in a way that the number could not. Thus:

A$ + A

would be illegal, but

A$ + STR$(A)

would be perfectly acceptable.

With the ability to store both numbers and text, and to contain varieties of information in an easily accessible form, it is no surprise that record files, along with record analysis and manipulation, play an important part in the programs which follow.

The limits of memory

Unfortunately for us, the memory capacity of the Commodore 64, though it is substantial, is not infinite. Every programmer has to bear in mind that, the longer a program is, the less room there is left over in memory for actually storing and handling information. For that reason, certain of the larger programs in this book are capable of being easily broken down into collections of smaller programs, each leaving more room for data in memory. The drawback here is that each time a different function is to be used, a new program has to be loaded into memory. This may be no problem if you are using discs, but the loading time required for tape may make the delays unacceptable. The criteria here must be how desperately you need the extra memory space and how often you need to access the different functions.

The larger the size of file you wish to work with, the more crucial becomes the question of saving memory space. For business programs it may become practical to consider either:

a) Compiling the program, which will reduce the length of most substantial programs and in addition make them run considerably faster, or

b) Eliminating unnecessary program lines. The subroutines given in this book are intended to be general purpose, designed to be placed into programs of your own design with the minimum of alteration. In a particular application, you may well find that some of the lines in a subroutine are never used and, if you do not intend to expand the program further, you might well be able to cut these out.

Disc or tape

By now it is clear that one of the most crucial aspects of business programming will be the speed with which information can be picked up from the storage files on which the programs themselves will work. With that in mind, it is as well to consider the facts regarding the capabilities of tape and disc for storage, before plunging into writing the programs themselves.

A system based on tape storage has two distinct disadvantages.

1) Access to data is relatively slow.

2) Only one channel of communication can be open at any one time. This means that, if information is recalled from a file, nothing can be done in terms of re-storing it without closing the original file and then re-opening it. If a file has to be edited, now matter how small the change to be made, it will be necessary to load in *all* the data in the file, make the necessary changes, and then resave the whole lot. If a program

needs to access different files for different purposes, perhaps combining information from two different sources, constant changes of tape will have to be made. On the other hand, if all that is required of a program is that it access a series of records in order, then there is no reason why a single file should not cover all of one side of a C90 cassette.

A disc drive, although relatively expensive, is faster to use, both for loading programs and data. It is also more versatile in that five channels can be open at the same time, allowing access to five different files for different purposes.

The programs in this book have been written for those who wish to begin with a tape-based system, but provision has been made for anyone who has, or intends to have, a disc drive. A simple replacement of certain clearly labelled program sections with the disc version will allow a disc-based system to be implemented at any time with the minimum of problems.

With a disc system, files can be handled much more readily and the storage capacity of the disc can even be used as an effective extension to the 64's memory.

Preventing operator errors

Operator errors are always a matter for great concern in business programs. One mistake in the entry of data to an accounting system could throw the whole of the accounts out. If your computer operator is also your secretary/typist, and familiarity with a keyboard would make that person an obvious choice, then provision for the prevention of errors must be incorporated into every program. This is not intended to convey any disrespect for secretaries, it is simply that their very training may lead them into errors. Typists, for example, make no distinction between space and shift/space in their work — the 64 quite definitely does, and to use the wrong one can corrupt an entry to a file.

Other examples of error-prevention include giving the operator a second look at what has been entered before anything irrevocable is done with it. The RUN/STOP key, which is capable of creating havoc if pressed accidentally, can be disabled. Throughout the subroutines contained in this book, you will find examples of error-prevention techniques and time will prove that they are not window-dressing, but essential parts of your system.

In addition to your straightforward 'error-trapping', as it is known, it is essential in designing your eventual system that you bear in mind the needs and weaknesses of the operator. When dealing with experienced typists, the program should attempt to make clear what is an acceptable computer input when it differs from correct practice on a typewriter. Also the screen

should never be cluttered with information or request for inputs — one question at a time will reduce errors dramatically. All in all, the operator should never be in any doubt at any point during the execution of the program just what is expected.

Program design

Designing programs requires forethought. Time spent with a pencil and paper before you begin programming can help to establish in your mind the correct sequence of subroutines for the eventual program to follow. One example might be the equipment inventory program in Chapter 5. The design of the program required first of all forgetting the capabilities of the computer itself and then concentrating on what the user would require. The objectives arrived at were as follows:

1) Production of a printed list of all items of equipment, to include a description of each item along with the serial number, room location, date purchased, purchase price and the present day value as an indication of the firm's assets and their replacement cost.

2) Production of printed lists specific to room numbers, to allow the inventory to be checked easily at regular intervals.

3) The ability to identify and list all the items purchased in a specified year.

4) The ability to search through the file and isolate an item according to a description fed in.

The task now will be to identify how much of the program material needed is already available in the form of subroutines. Reference to Chapter 5 will demonstrate that all the routine functions, such as creating a file, editing it, adding to it, printing out lists, performing financial calculations and so forth, are there waiting to be used, even if sometimes in a modified form. What matters is that when you approach the material you already have a clear idea of what will be needed and, therefore, what you are looking for in the way of subroutines.

CHAPTER 2
Subroutines

In this chapter you will find ten subroutines. In two of these (Subroutines 3 and 8), you have to choose options — these relate to two alternative data input routines and to storage on disc or tape.

In entering the subroutines, it does not matter whether you choose to enter them as one program or to enter and save them individually. Included in Subroutine 1 are three programming tools which will facilitate the combination or 'merging' of the subroutines at a later date, or the deletion of subroutines if they are not needed for a particular application. There is also a line renumber facility you may wish to use when developing your own applications, but this should *not* be applied to the subroutines themselves since they are all interdependent: they rely for their working on preservation of the line numbers given in the listings. It should also be noted that the renumber routine copes only with the line numbers themselves, GOTOs and GOSUBs in a renumbered program will have to be altered manually.

The listings given in this chapter include a special notation for 'control characters', like cursor up, cursor down, clear screen and so on. When such special characters are encountered in the listings, they should be replaced by the proper control character. The special symbols are as follows:

[CD]	Cursor down.
[CU]	Cursor up.
[CL]	Cursor left.
[CR]	Cursor right.
[HOME]	Cursor to home position.
[CLR]	Clear screen.
[RVS]	Reverse field on.
[RVO]	Reverse field off.
[SPC]	Shifted space. Note that ordinary spaces in the subroutines should be entered by use of the space bar. Where [SPC] occurs, SHIFT/SPACE *must* be used.

Note also the comments with regard to upper and lower case in the programs, given in the Program Notes at the beginning of this book.

Subroutine 1: Variable Initialising and Program Aid

This subroutine deals with the allocation of space to arrays, setting up string variables, program security, disc error messages, checking for the presence or absence of a printer, and the merge, renumber and delete routines mentioned earlier.

```
1 REM ***SUB 1.    1.8K     COMMODORE 64.*
**
2 POKE53280,11:POKE53281,12:PRINTCHR$(14
4):PRINTCHR$(14):POKE650,255:GOTO4
3 S6=1:CLOSE15:REM **CLOSE15 TO BE INCLU
DED USING DISC DRIVE ONLY**
4 W1$="[RVS]##############################
#############"
5 W2$="[CLR][RVS]PROGRAM NAME.[SPC][SPC]
[SPC][SPC][SPC] [SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][HOME][CD]"
6 OPEN15,8,15:REM **TO BE INCLUDED USING
 DISC DRIVE ONLY**
8 BL$="[SPC][SPC][SPC][SPC][SPC][SPC][SP
C][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SP
C][SPC][SPC][SPC][SPC][SPC]":REM ***20 U
PPER CASE SPACES***
9 BL$=BL$+BL$+BL$
10 DIMA5$(100),A6$(100),A7$(100),A8$(500
),A9$(500)
12 REM ***ADD COMMAND TO ENABLE OR DISAB
LE RUN/STOP KEY WITH COMPILER***
13 PRINTW2$:IFS6=1THEN1000
14 PRINT"[CD]  COPYRIGHT BELONGS TO J. D
OBERMAN.[CD][CD]":PRINTW1$
15 PRINT"[CD]  TYPE IN CODE TO PROCEED.[
CD][CD]":PRINTW1$:GOTO21
16 PRINTW2$
17 PRINT"[CD]YOU HAVE EITHER TYPED IN TH
E WRONG CODE[CD][CD][CD]"
18 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC][
SPC][SPC][SPC][SPC][SPC][SPC][SPC]     O
R[CD][CD][CD]"
19 PRINT"[SPC]     YOU ARE NOT AN AUTHORI
SED USER.[CD]"
20 PRINTW1$:FORJ=1TO3000:NEXTJ:GOTO13
```

```
21 PRINT"[CD][CD].....................
..[CD]"
22 G$="":GETG$:IFG$=""THEN22
23 IFG$=CHR$(13)THEN25
24 CO$=CO$+G$:GOTO22
25 IFCO$="DOB"THEN27
26 CO$="":GOTO16
27 PRINTW2$:PRINT"[CD][CD]DO YOU INTEND
USING THE PRINTER?[CD][CD]":PRINTW1$
28 GOSUB202:IFY$="N"THEN1000
29 PRINTW2$:PRINT"[CD][CD]IF A DEVICE NO
T PRESENT MESSAGE[CD]"
30 PRINT"IS RETURNED THEN CONNECT THE PR
INTER    AND RE-START PROGRAM.[CD][CD]"
31 OPEN4,4:PRINT#4:IFST=-128THENPRINT"[C
D][CD]DEVICE NOT PRESENT":END
32 CLOSE4:GOTO1000
9999 CLOSE15:END:REM **CLOSE15 TO BE INC
LUDED USING DISC DRIVE ONLY**
59997 REM ################################
59998 REM ***PROGRAM MERGE ROUTINE.***
59999 REM ################################
60000 PRINT"[CLR]";
60001 OPEN1,1,0,"PROGRAM NAME"
60002 POKE184,1:POKE185,96:POKE186,1:POK
E152,1:PRINT"[CLR][CD][CD]"
60003 GET#1,A$:PRINTA$;:IFSTTHEN60007
60004 IFA$<>CHR$(13)THEN60003
60005 PRINT"GOTO60002[HOME]";:POKE631,13
:POKE632,13:POKE633,13
60006 POKE198,3:END
60007 CLOSE1 : END
60997 REM ############################
60998 REM ***LINE DELETE ROUTINE.***
60999 REM ############################
61000 A1=PEEK(828):A2=PEEK(829):A3=PEEK(
830):A4=PEEK(831)
61001 A=2049:REM **2048 = START OF BASIC
**
61002 PRINT"[CLR]":INPUT"DELETE FROM (LI
NE NO.):- ";F
61003 PRINT"[CD]":INPUT"DELETE TO (LINE
NO.):- ";T
61004 IFPEEK(A+2)+256*PEEK(A+3)<FTHENA=P
```

```
EEK(A)+256*PEEK(A+1):GOTO61004
61005 POKE830,T-INT(T/256)*256:POKE831,T
/256:GOTO61007
61006 A=PEEK(828)+256*PEEK(829):T=PEEK(8
30)+256*PEEK(831)
61007 IFPEEK(A+2)+256*PEEK(A+3)>TTHEN610
12
61008 IFPEEK(A)+256*PEEK(A+1)=0THEN61012
61009 PRINT"[CLR][CD][CD][CD]"PEEK(A+2)+
256*PEEK(A+3):PRINT"GOTO61006":PRINT"[CU
][CU][CU][CU][CU][CU][CU]"
61010 POKE828,A-INT(A/256)*256:POKE829,A
/256:POKE198,2:POKE631,13:POKE632,13
61011 END
61012 POKE828,A1:POKE829,A2:POKE830,A3:P
OKE831,A4:END
61997 REM ##########################
61998 REM ***RENUMBER ROUTINE.***
61999 REM ##########################
62000 A=2049:REM **START OF BASIC = 2048
**
62001 PRINT"[CLR]LINES TO BE RENUMBERED"
:INPUT"[CD]LOWEST LINE NUMBER :- ";L
62002 INPUT"HIGHEST LINE NUMBER:- ";H
62003 INPUT"[CD]TO BE RENUMBERED FROM:-
";F
62004 INPUT"[CD]INTERVAL BETWEEN LINES:-
 ";I
62005 IFPEEK(A+2)+256*PEEK(A+3)<LTHENA=P
EEK(A)+256*PEEK(A+1):GOTO62005
62006 P1=PEEK(A)+256*PEEK(A+1):IFP1=0THE
NEND
62007 IFPEEK(A+2)+256*PEEK(A+3)>HTHENEND
62008 N2=INT(F/256):N1=INT(F-256*N2):POK
EA+2,N1:POKEA+3,N2:F=F+I:A=P1:GOTO62006
```

Commentary

Line 2: The first two POKEs set up the screen and border colours. Printing CHR$(14) puts the screen into lower case mode and the third POKE enables all the keys to repeat if pressure is maintained upon them. You might like to note that to put the screen back into upper case mode requires the printing of CHR$(142) or, alternatively, POKE 53272,23 and POKE 53272,21 for lower and upper case respectively.

Lines 3–6: The variable S6 is temporarily set to one, in order to default around the security routine in lines 14–21. The CLOSE and OPEN statements in these lines are intended to clear and re-open access to the disc 'error channel', which reports on problems with the disc drive. These should be omitted if you are not using discs.

Lines 4–5: These strings, when printed, clear the screen and then give the effect of a box which draws the attention of the operator to the current question. The printing of the program name by W2$ is a useful reminder when several programs may be in use.

Lines 8–9: B1$ is set up with sixty spaces to be used for padding out strings.

Line 10: The dimensions of these arrays can be altered to suit a particular programming need.

Line 12: If the eventual program is to be compiled with a commercial compiler for greater speed, a special instruction will be needed to disable the RUN/STOP key and this should be inserted here.

Lines 13–26: These represent the program's security system, requiring a password (specified in line 25) to be entered. In the program as listed, the password is set to 'DOB'.

Lines 22–24: GET is used to pick up characters one by one from the keyboard until RETURN (CHR$(13)) is pressed. INPUT would be simpler, but offers no control over the characters typed in. In addition, if a comma or colon were required by the password, INPUT would refuse to accept any following text. One extra thing to note in respect of entering information, either by means of INPUT or GET, is that if an entry begins with spaces they will be ignored *unless they are entered as shifted spaces.*

Clearly, as the program stands, all that would be needed to avoid this security system would be to stop the program and list it out, revealing the password. This can be avoided by altering line 25 to read:

```
25 PRINT "":IFCO$ = "DOB"THEN27:PRINT"26 DELETES"
```

The 26 'delete' characters are obtained by entering:

```
PRINT ""
```

then moving the cursor back over the second quotation mark and pressing SHIFT/INSERT 26 times. This opens up 26 spaces and the DEL key can now be pressed a sufficient number of times to fill the space. When listed, the line will now appear as:

```
25 PRINT""
```

even though the full line is actually present. This could still be overcome by starting the program with RUN 3, for instance, and it is therefore advisable to check at various points during the program that CO$ is equal to the password. A reasonable point at which to do this would be just prior to accessing a data file for recall of data. For instance, lines 701 or 703 in Subroutine 8 should be changed to:

```
701 (or 703) PRINTW2$:IFCO$< > "DOB"THENSYS64738:
    PRINT"26 DELETES
```

If the correct code has not been entered in the first place, then CO$ does not equal 'DOB' and SYS 64738 will be executed. This calls up the built-in routine in ROM which will reset the computer just as if it had been switched off and on again. Note that there are no quotation marks at the end of the 26 delete symbols in this line, otherwise the line would appear as:

```
701 PRINTW2$"
```

The routine can be toughened up even more by changing the password during the course of the program. Line 1000 in Subroutine 9 could, for instance, be altered to read:

```
1000 PRINTW2$:PRINT"TYPE OPTION NUMBER":
     CO$ = CO$ + "S":PRINT"29 DELETES"
```

The password is now changed to 'DOBS' and any tests made subsequently would be for the amended version. This method of securing a program with invisible checks makes it fairly secure against all but the most determined attempts to break into it.

One word of warning: it is wise not to include such a provision in the program until it has been finally developed and saved on tape or disc or you may accidentally lose all your work.

Lines 27–32: These lines determine whether a printer will be used and, if so, whether it is plugged in to the computer. The method used is to open a file to the printer and then to test the value of the system variable ST. If the printer is properly connected, the value of ST will be zero. If ST equals −128, this is an indication of a DEVICE NOT PRESENT error. This error should be intercepted without the program stopping, although on occasions, depending on the set-up of the printer and computer, the program *will* stop with an error. If this happens, the program must be re-run after the printer is properly connected. It is clearly better to learn that the printer is not available at this early stage, even if the program is stopped, than to perform some complicated data analysis and then lose the whole thing because the printer is not plugged in.

Line 9999: This ends the program and closes down communication with the disc drive. Once again, attaching the line to the first subroutine ensures that it will always be present, no matter what configuration of subroutines is used. The CLOSE 15 may be omitted if tape rather than disc is being used.

Lines 59997–62008: The programming tools referred to at the beginning of the chapter. They are included in the first subroutine so that it can be used to load in and merge other subsequent sections of the program. As the routines are written in BASIC, they are slow and do not compare with the commercially available machine code products, but you should find them extremely useful. They can either be included in Subroutine 1, as here, or used independently to commence the build-up of collections of stipulated subroutines.

In the following descriptions, I have only explained the workings of these routines as far as is necessary to use them. Whereas the other subroutines in this book are for you to understand and adapt to your own needs, these merge, delete and renumber routines are for you to use as they stand.

Merge

Will allow a program, which has been converted to a data file and stored on tape, to be combined with a program held in the computer memory. The program in memory can have this routine attached to it, or the routine can be used on its own. This is a merge and not an append routine and program lines in the computer memory will be overwritten if there are corresponding line numbers in the program stored on tape.

Stage 1 in the procedure is to convert the program to be merged, with the program containing the merge routine, into a data file. This is achieved by loading the program to be merged into the computer. A blank tape without any clear leader being visible is placed into the recorder. The following is entered at the keyboard followed by RETURN:

```
OPEN1,1,2,"PROGRAM NAME":CMD1:LIST
```

The instruction PRESS RECORD AND PLAY will be displayed. When the recorder has finally ceased recording, type in the following, followed by RETURN:

```
PRINT # 1:CLOSE1
```

Rewind the tape. The program in memory has now been saved as a data file.

Stage 2 is to type NEW followed by RETURN and load into the computer the program containing the merge routine. LIST line 60001 and enter the correct program name followed by RETURN. Place the program data file tape in the recorder and enter the following:

RUN60000

followed by RETURN—PRESS PLAY will be displayed. Completion of the merge will be indicated by an OUT OF DATA error message. The recorder can be switched off and, on listing the program, the merge can be confirmed. This sequence can be repeated to include as many program data files as required.

This routine imitates the addition of program lines to an existing program in memory using the keyboard; but, in this case, the keyboard entries are replaced by tape data records.

Line 60002: Will be dealt with under lines 60005–60006.

Lines 60003–60004: Each data record, or line, is printed to the screen. The end of each data record is signified by CHR$(13), and the program goes on to line 60005. Line 60003 also contains the end of data file marker, ie IF ST THEN 60007.

Lines 60005–60006: Print 'GOTO 60002' to the screen, below the data record line. The HOME command positions the cursor. Line 60005 places three RETURNs into the keyboard buffer, and line 60006 records in memory location 198 that these three key presses have still to be acted on when the program ends. The program 'ends', which 'activates' the three RETURNs: the cursor moves over the data record displayed on the screen and the GOTO 60002, and the data record is entered into the memory as if the line had been entered at the keyboard and the RETURN key pressed.

The merge routine then returns to line 60002. Normally, if a new program line was entered, the tape file would be automatically closed and so any attempt to call the next program data record in BASIC would be met with a FILE NOT OPEN error message. Line 60002 makes the computer think that the tape file has not been closed, and so the next data record can be called. This sequence goes on repeating until the end of the file is reached: the OUT OF DATA error message is displayed on the screen.

I am indebted to David Lawrence for allowing me to reproduce this merge routine from his book *The Working Commodore 64* (Sunshine Books).

Delete

Every program line begins with two 'link bytes' which indicate the place in memory where the next program line begins. Following the two link bytes are two bytes which store the actual line number of this program line. When searching for a particular line, it is possible to follow the chain of link bytes, examining line numbers along the way but without having to scan along the whole of each program line.

Deleting a program line using BASIC can only be carried out in direct mode or in simulated direct mode. Here, direct mode deletion is simulated by the simple expedient of printing a line number on to the screen, positioning the cursor on the same line and then entering a RETURN (CHR$(13)).

As this routine entails the use of END, variables will be lost, namely the highest and lowest line numbers to be deleted. We also need some method of safely recording the next line number to be deleted.

The cassette buffer memory locations (828–1019) are unused while the program is running and are therefore available as storage space where values will not be cleared. We will need two of these locations for the highest line number, and another two to hold both the lowest line number and the next line number to be deleted.

Lines 61002–61003: The lowest and highest line numbers are entered here.

Line 61004: Examines the first line of the program to see if it matches the lowest line number. If not, it passes on to the next line until the lowest line number is found.

Line 61005: The highest line number is POKEd into two of the cassette buffer memory locations.

Lines 61006–61012: Lines 61007 and 61008 ensure that the line number is not greater than the highest line number and also check for the program end. Lines 61009–61011 print the line number and 'RETURN TO LINE 61006', and use two RETURNs from the keyboard buffer. Line 61006 reads the next line number to be deleted from the cassette buffer: the process carries on repeating until either the end of the program or the highest line number is reached.

Renumber

This routine deals with renumbering stipulated program lines by stipulated increments (lines 62001 and 62002). It does not require further explanation here, but note that it will not renumber GOTOs, GOSUBs, etc, within the program lines.

Subroutine 2: Numeric Input

This is a standard data input routine which will accept only numbers, the decimal point, the # sign and the RETURN key.

```
97 REM ####################################
####################################
98 REM ***SUB 2.  0.8K   NUMERIC INPUT O
NLY VARIABLE D9$  COMMODORE 64***
99 REM ####################################
####################################
100 D9$="":PRINT"[CD]TYPE  #  (HASH SIGN
) TO END INPUT[CD]"
101 PRINTW1$:PRINT"[CD]MAXIMUM AMOUNT OF
 NUMBERS  =  ";L9:PRINT
102 FORJ=1TOL9:PRINT".";:NEXTJ:PRINT:PRI
NT"[RVS][SPC][RVO][CL]";
103 GETG$:IFG$<>""THEN103
104 G$="":GETG$:IFG$=""THEN104
105 IFG$="#"THEND9$=G$:L9=0:PRINT:RETURN
106 G=ASC(G$):IFG=13ORG=141THENPRINT" ":
GOTO114
107 IFG=20ORG=148THEN111
108 IFG=45ORG=46THEN110
109 IFG>57ORG<48THEN103
110 D9$=D9$+G$:PRINTG$;:PRINT"[RVS][SPC]
[RVO][CL]";:GOTO103
111 L=LEN(D9$):IFL<1THEND9$="":GOTO103
112 IFL<2THEND9$="":PRINT"[SPC][CL][CL][
RVS][SPC][RVO][CL]";:GOTO103
113 D9$=LEFT$(D9$,L-1):PRINT"[SPC][CL][C
L][RVS][SPC][RVO][CL]";:GOTO103
114 IFTR=1THEN117
115 IFLEN(D9$)>L9ORLEN(D9$)<1THENGOSUB20
7:GOTO100
116 G$="":L9=0:PRINT:RETURN
117 IFLEN(D9$)<>L9THENGOSUB207:GOTO100
118 GOTO116
```

Commentary

The routine is used when only numbers are required, for instance when entering pounds and pence. To access the routine from the main program, the maximum number of characters for the entry must first be specified and stored in the variable L9, then the subroutine called with GOSUB 100, eg:

```
2015 L9 = 20:GOSUB 100
```

Note when you enter this in lower case and return to it later that the variable here is L9 in lower case (l9) and not nineteen. Care must be exercised in any dealings with this subroutine to distinguish between the two.

Line 102: Prints a series of full stops across the screen to indicate the number of characters permitted.

Lines 103–110: Employ GET to allow the entry of the character to which this subroutine is limited.

Line 105: Input of '#' terminates the entry of data.

Line 106: Each character in the 64's character set has a unique code, its 'ASCII' code, and these codes can be put to good use in controlling the information entered. In this line the two characters tested for are the RETURN key and SHIFT/RETURN. Both of these are needed, since they are different characters to the 64, and typists do not usually distinguish between them. The line dictates that, if either of the two is pressed, the program moves on to receive the next item of data.

Line 107: Similar test for DELETE and SHIFT/DELETE. Deletions are carried out by lines 111–113.

Line 108: Accepts the minus sign or decimal point.

Line 109: Any characters outside the range 0–9 (ASCII 48–57) are rejected.

Line 100: Stores the characters entered in the string D9$ and then prints the latest character and follows it with a reverse space. The 'cursor left' at the end of the line places the print position back on the reverse space, which thus acts as a non-flashing cursor. This is needed to overcome the problem that the automatic flashing cursor is absent when using GET rather than INPUT. It can be reinstated by the use of POKE 204,0 (POKE 204,1 cancels it) but this is erratic in practice.

Lines 111–113: Deal with the deletion of characters.

Line 111: If there are no characters in the entry so far, the DELETE is not acted upon.

Line 112: If there is only one character in the entry so far, the DELETE results in the entry (D9$) being set to a null string.

Line 113: At this point the entry must be more than one character long. The line reduces the length of the string by one character by taking the lefthand portion of the total string to a length of L% – 1.

Line 114–118: Another optional error check. If the subroutine is called by a line which sets the variable TR equal to one, then, instead of L9 being the maximum number of characters which may be input, it becomes the only

length acceptable. Any attempt to press RETURN or SHIFT/RETURN when the entry is *not* L9 characters long will result in an error message being printed by a subsequent subroutine. Two separate error messages are provided, one for entries which are less than one character in length or greater than L9 characters and the other for entries which are between one and L9 – 1 characters long.

String handling

Deletion in this subroutine is accomplished by use of one of the 64's string-handling functions, namely LEFT$. String handling is used repeatedly in the manipulation of data and we shall take a moment to examine it.

There are three ways in which strings may be approached on the 64: either by starting from the lefthand end (LEFT$); by starting at some point which is not necessarily the lefthand end (MID$); or from the righthand end of the string (RIGHT$). Take the following line as an example:

```
10 A$ = "COMMODORE 64":PRINTLEFT$(A$,9)
```

Running this line will result in 'COMMODORE' being printed. The LEFT$ has excluded all but the first nine characters of the original string. Now try:

```
10 A$ = "COMMODORE 64":PRINTRIGHT$(A$,2)
```

The result now is '64', since all but the two rightmost characters are excluded.

```
10 A$ = "COMMODORE 64":PRINTMID$(A$,4,3)
```

will result in 'MOD'. Here the string function dictates that three characters will be printed, beginning with the fourth character.

MID$ can be used to simulate LEFT$, as in:

```
10 A$ = "COMMODORE 64":PRINTMID$(A$,1,9)
```

which produces the same result as the LEFT$ example given above. Similarly MID$ can replace RIGHT$:

```
10 A$ = "COMMODORE 64":PRINTMID$(A$,LEN(A$) – 2 + 1)
```

This produces the same result as the RIGHT$ example above. All that is necessary is to take the length of the overall string, subtract from it the number of characters to be printed and add one. Normally the '– 2' and

'+1' would be run together as '−1', but they are left separate in this case so that you can see the format which would be necessary if the number of characters were being specified by a variable. Note that there is only one parameter after the string name within the brackets, ie the starting point. When MID$ is only told the starting point of the section of the total string, and not the number of characters involved, it automatically includes the whole of the rest of the string from that point. Thus MID$(A$,1) would be the whole of A$.

These string-handling facilities can be of enormous use in data handling, as illustrated by the following example:

```
10 A$ = "60.08CR 72.08DR 100.08BALANCE"
```

To update the balance to include the 60.08 credit and the 72.08 debit, and then to change both credit and debit to 0.00, the following sequence could be used:

```
20 CR = VAL(LEFT$(A$,5))
30 DR = VAL(MID$(A$,8,5))
40 BL = VAL(MID$(A$,15,6))
50 D = DR - CR:BL = BL - D
60 CR$ = "0.00":DR$ = "0.00":BL$ = STR$(BL)
70 A$ = CR$ + MID$(A$,6,2) + DR$ + MID$(A$,13,2) + BL$ +
   RIGHT$(A$,7)
80 PRINTA$
```

The result should now be "0.00CR 0.00DR 88.08BALANCE".

Subroutine 3: Alpha and Numeric (GET #)

Another data entry routine, this time allowing the entry of a wider range of characters. The subroutine will be given in two versions, 3 and 3a. The difference between them in use will not be apparent at this stage, since they are intended to cope with the creation of files which will be recalled from tape/disc by two different methods, GET # and INPUT #. These two forms of data retrieval will be discussed in full when Subroutine 8 is presented. For the present, both of these alternative versions can be entered and stored separately so that you can decide later which is most appropriate to your particular application.

```
147 REM ####################################
####################################
148 REM **SUB 3.  0.8K  ALPHA & NUMERIC
INPUT USING GET  COMMODORE 64**
```

```
149 REM ##################################
##################################
150 D9$=""
151 PRINT"[CD]MAXIMUM NUMBER OF LETTERS
 =  ";L9:PRINT
152 FORJ=1TOL9:PRINT".";:NEXT:PRINT:PRIN
T"[RVS] [RVO][CL]";
153 GETG$:IFG$<>""THEN153
154 G$="":GETG$:IFG$=""THEN154
155 G=ASC(G$):IFG=13ORG=141THENPRINT" ":
GOTO168
156 IFG=20ORG=148THEN165
157 IFG=32ORG=160THENG=160:G$=CHR$(G):GO
TO159
158 GOTO161
159 IFG=160ANDDF=1THENG=32:G$=CHR$(G)
160 GOTO163
161 IFG=34THENPRINT"'";"[RVS] [RVO][CL]"
;:GOTO164
162 IFG>95ANDG<193ORG<33ORG>218THEN153
163 PRINTG$;:PRINT"[RVS] [RVO][CL]";
164 D9$=D9$+G$:GOTO153
165 L=LEN(D9$):IFL<1THEND9$="":GOTO153
166 IFL<2THEND9$="":PRINT" [CL][CL][RVS]
 [RVO][CL]";:GOTO153
167 D9$=LEFT$(D9$,L-1):PRINT" [CL][CL][R
VS] [RVO][CL]";:GOTO153
168 IFLEN(D9$)>L9ORLEN(D9$)<1THENGOSUB15
3:D9$="":GOTO151
169 G$="":L9=0:PRINT:RETURN
```

Commentary

This subroutine is very similar to the last, except that it allows the entry of a wider range of characters, namely all the normal printing character set, ASCII 34–95 and ASCII 193–218. You can see what all these are by referring to Appendix C. Characters outside these two ranges will be excluded by line 162.

Line 156: In addition to dealing with deletion from the righthand end of the string, this line excludes the use of the cursor control keys, codes 17, 29, 145 and 157. Cursor left could be included by amending the line to read:

```
156 IF G = 20 OR G = 148 OR G = 157 THEN 165:REM **157 =
    ASCII CODE FOR CURSOR LEFT**
```

Cursor left would now act in the same way as delete.

Line 157: This is a good example of the flexibility provided by the use of the ASCII codes for characters. Here, any leading spaces in a string are converted to SHIFT/SPACEs.

Line 159: This line has exactly the opposite effect, changing SHIFT/SPACE to ordinary SPACE if the variable DF is set to one. This is used when the name of a data file to be opened on tape/disc is being entered, since the use of SHIFT/SPACE in such a file name can cause complications.

Line 168: The length of the entry (D9$) is checked and, if less than one or greater than L9, the string is cancelled and the operator must start again. Retyping the entry can be avoided by inclusion of the following program lines:

```
168 IFLEN(D9$)< 1THENGOSUB153:GOTO150
169 IFLEN(D9$)> L9THEN171
170 G$ = "":L9 = 0:PRINT:RETURN
171 GETG$:IFG$< > ""THEN171
172 G$ = "":GETG$:IFG$ = ""THEN172
173 G = ASC(G$):IFG = 20ORG = 148ORG = 157THEN167
174 GOTO171
```

These extra lines will only accept DELETE, SHIFT/DELETE or cursor left if the length of the entry is greater than L9. The entry will not be destroyed, but RETURN will not be accepted until the length of the entry is reduced.

Subroutine 3a: Alpha and Numeric (INPUT #)

```
147 REM ####################################
####################################
148 REM ***SUB 3A.  0.8K  ALPHA & NUMERI
C INPUT USING INPUT  COMMODORE 64***
149 REM ####################################
####################################
150 D9$=""
151 PRINT"[CD]MAXIMUM NUMBER OF LETTERS
 =  ";L9
152 PRINT"[CD]":FORJ=1TOL9:PRINT".";:NEX
T:PRINT:PRINT"[RVS] [RVO][CL]";
```

```
153 GETG$:IFG$<>""THEN153
154 G$="":GETG$:IFG$=""THEN154
155 G=ASC(G$):IFG=13ORG=141THENPRINT" ":
GOTO170
156 IFG=20ORG=148THEN167
157 IFG=44THENG=172:PRINTG$;:PRINT"[RVS]
 [RVO][CL]";:G$=CHR$(G):GOTO166
158 IFG=58THENG=186:PRINTG$;:PRINT"[RVS]
 [RVO][CL]";:G$=CHR$(G):GOTO166
159 IFG=34THENG=162:G$=CHR$(G):PRINT"'";
:PRINT"[RVS] [RVO][CL]";:GOTO166
160 IFG=32ORG=160THENG=160:G$=CHR$(G):GO
TO162
161 GOTO164
162 IFG=160ANDDF=1THENG=32:G$=CHR$(G)
163 GOTO165
164 IFG>95ANDG<193ORG<33ORG>218THEN153
165 PRINTG$;:PRINT"[RVS] [RVO][CL]";
166 D9$=D9$+G$:GOTO153
167 L=LEN(D9$):IFL<1THEND9$="":GOTO153
168 IFL<2THEND9$="":PRINT" [CL][CL][RVS]
 [RVO][CL]";:GOTO153
169 D9$=LEFT$(D9$,L-1):PRINT" [CL][CL][R
VS] [RVO][CL]";:GOTO153
170 IFTR=1THEN173
171 IFLEN(D9$)>L9ORLEN(D9$)<1THENGOSUB20
7:D9$="":GOTO151
172 G$="":L9=0:PRINT:RETURN
173 IFLEN(D9$)<>L9THENGOSUB207:D9$="":GO
TO151
174 GOTO172
```

This is an alternative to Subroutine 3 for those cases where the data file being created will eventually be recalled from tape/disc by means of INPUT#. To avoid problems with the eventual use of INPUT#, certain changes have to be made to characters with which INPUT# (or INPUT) cannot deal.

The major difference between the use of INPUT# and GET# to recover data from tape/disc is the speed of access. Data brought into the computer using INPUT# is drawn in as a block, whereas GET# is obliged to bring in data character by character, building up a string until it meets a carriage return symbol (CHR$(13)), then going on to the next string. INPUT# has the limitation, however, that, when strings are being input, only 88 characters can be picked up without the program stopping with an

error. GET # can build up a string to the maximum permitted length of 255 characters before giving an error.

Another disadvantage associated with INPUT # is its inability to deal with certain characters, such as the colon and comma. Any data which follows a colon or comma in a string will be ignored by INPUT #. It is unlikely that these two characters will actually be required except in word processing applications, and their use could be excluded by incorporating a line such as:

```
157 IFG = 44ORG = 58THEN153
```

Another alternative would be to convert the ASCII codes for the two characters by adding 128 to them, creating an acceptable character. Unfortunately these transformed characters would have to be re-translated when the data was loaded back from tape and the advantages of loading whole blocks without examination of their contents would be lost. It should be noted that, when using many 'daisy wheel' type printers, the altered codes (172 and 186) will actually print a comma and colon, but, unfortunately, that this is not the case with most dot matrix printers.

To illustrate the difference in speed and the length problem, enter the following routine (this is NOT part of the main program, merely a test for your own satisfaction):

```
10  PRINT "PLACE TAPE FOR DATA IN THE RECORDER"
20  PRINT "ENSURE NO CLEAR LEADER IS SHOWING":STOP
30  OPEN 1,1,1,"X"
40  INPUT "TYPE IN INFORMATION";A$
50  PRINT #1, A$:A$ = ""
60  INPUT "TYPE IN INFORMATION";A$
70  PRINT #1, A$:A$ = ""
80  INPUT "TYPE IN INFORMATION";A$
90  PRINT #1, A$:A$ = ""
100 B$ = "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAA":PRINT #1,B$:REM 44 As
110 A$ = B$:A$ = A$ + B$:PRINT #1,A$:REM 88 As
120 A$ = A$ + B$:PRINT #1,A$:REM 132 As
130 CLOSE1:PRINT "REWIND TAPE":STOP
140 OPEN 1,1,0,"X"
150 FOR J = 1 TO 6
160 INPUT #1,A$:PRINTA$
170 NEXT J
180 CLOSE1
```

RUN the test routine and the STOP command in line 20 will cause a break in the execution to allow a tape to be placed in the recorder. When the tape has been inserted, type 'CONT' to restart the program. Now experiment by inputting different length strings. When you have INPUT three strings, the routine will output the collections of As to the tape and then stop at line 130. Rewind the tape to its starting place and type 'CONT' to start the program again. You will now see the strings you entered reprinted, followed by 44 As, 88 As and then a STRING TOO LONG error message. This illustrates both the simplicity and the limitations of INPUT# when it comes to length of data.

To see how commas and colons create problems, replace line 40 with:

```
40 G$ = "":GETG$:IF G$ = ""THEN40
41 IFG$ = CHR$(13)THENPRINT:GOTO50
42 A$ = A$ + G$:PRINTG$;:GOTO40
```

You can also replace the INPUTs at lines 60 and 80 with a similar routine, though the two GOTOs will have to be changed to match the changed line numbers. Having made these changes, you can go ahead and experiment with making entries which are longer than the two lines permitted by INPUT and which also include commas and colons. When the data is replayed from tape, you will discover that any portions which follow a comma or colon are lost.

Now you can go further and replace line 160, in the part of the routine which picks up data from tape, with the following:

```
160 G$ = "":GET#1,G$
161 IFG$ = CHR$(13)THENPRINT:GOTO170
162 A$ = A$ + G$:PRINTG$;:GOTO160
```

You will find this slower, but the limitations on character length (to 88), and with regard to commas and colons, are removed.

Commentary

Lines 157–158: Convert the comma and colon codes.

Line 159: The quotation mark (ASCII 34) can cause problems in printing. Altering the code to 162 will print a quotation mark on most daisy wheel printers. The subject of printers is dealt with more fully in Subroutine 6. The line also dictates that, if a quotation mark is entered, it is replaced on the screen by an apostrophe. Without this, printing the quotation mark followed by a semicolon and [RVS] [RVO] [CL] would corrupt the screen.

Subroutine 4: Question and Answer

This simple section is intended to deal with certain common questions and responses which would otherwise need to be repeated several times during the course of the program. Sufficient space, in terms of line numbers, has been left for you to add your own items between Subroutines 4 and 5.

```
197 REM ##################################
###
198 REM ***SUB 4.    1.1K     COMMODORE 64
***
199 REM ##################################
###
200 PRINT"[CD][CR][CR][CR][CR][CR][CR][C
R][CR][CR][CR][CR][CR][CR][RVS]#########
#####"
201 PRINT"[CR][CR][CR][CR][CR][CR][CR][C
R][CR][CR][CR][CR][CR][RVS]#[RVO]ARE YOU
 SURE[RVS]#"
202 PRINT"[CR][CR][CR][CR][CR][CR][CR][C
R][CR][CR][RVS]######################"
203 PRINT"[CR][CR][CR][CR][CR][CR][CR][C
R][CR][CR][RVS]#[RVO]Y FOR YES  N FOR NO
[RVS]#"
204 PRINT"[CR][CR][CR][CR][CR][CR][CR][C
R][CR][CR][RVS]######################"
205 Y$="":GETY$:IFY$<>"Y"ANDY$<>"N"THEN2
05
206 PRINTW2$:RETURN
207 PRINTW1$
208 PRINT"[CD]**ERROR**  WRONG INPUT   T
RY AGAIN."
209 PRINTW1$:PRINT"[CD][CD]":RETURN
210 PRINT"[CD][CR][CR][RVS]##############
########################"
211 PRINT"[CR][CR][RVS]#[RVO]****SPACE B
AR KEY TO CONTINUE****[RVS]#[RVO]"
212 PRINT"[CR][CR][RVS]##################
####################"
213 G$="":GETG$:IFG$=""THEN213
214 G=ASC(G$):IFG=32ORG=160THENPRINTW2$:
RETURN
215 GOTO213
216 PRINTW2$:PRINT"[CD][CD][CD][CR][CR][
CR][CR][CR][CR][CR][CR]DO YOU WISH TO PR
```

```
OCEED?[CD][CD][CD]"
217 PRINTW1$:GOSUB202:PRINTW2$:RETURN
218 PRINTW2$
219 PRINT"[CD][CD][CD]FINISHED ADDING DA
TA TO FILE.   DO"
220 PRINT"[CD]YOU REQUIRE THE FILE TO BE
 CLOSED?[CD][CD]":GOSUB200:PRINTW2$:RETU
RN
221 PRINTW2$:PRINT"[CD][CD]RETURN TO MEN
U PROGRAM?[CD][CD]":PRINTW1$
222 GOSUB202:IFY$="N"THENRUN3
223 PRINTW2$:PRINT"[CD][CD]PLACE MENU TA
PE IN THE RECORDER.[CD][CD]":PRINTW1$:GO
SUB210
224 CLOSE15:CLR:PRINT"[CLR]LOAD"+CHR$(34
)+"MENU"+CHR$(34):REM**CLOSE15 DISC ONLY
**
225 PRINT"[CD][CD][CD][CD]RUN":POKE198,3
:POKE631,19:POKE632,13:POKE633,13:END
226 PRINTW2$
227 PRINT"[CD][CD][CD][CR][CR][CR][CR][C
R][CR]DO YOU REQUIRE ANOTHER COPY?[CD][C
D][CD]"
228 PRINTW1$:GOSUB202:PRINTW2$:RETURN
```

Commentary

Lines 200–206: This section can be accessed either from line 200 or line 202. In the case of access at line 200, the main part of the program has asked a question of the operator and this subroutine is called up to ensure that the item entered is correct. Access at 202 allows the program to simply ask YES OR NO at any appropriate point, eg:

1000 PRINT "*D*O YOU WISH TO PROCEED":GOSUB 202

Line 205: Will not accept keyboard entries unless they are 'Y' or 'N', while line 206 clears the screen, places the program name at the top and returns to the point after the question which was specified in the main program.

Lines 210–215: Provide a halt to allow a tape to be placed in the recorder or paper in the printer. Note that the routine does not use the prompt 'Press any key to continue'. A colleague of mine routinely included this prompt in his programs until one day, during a demonstration of a program to a captive audience, the RUN/STOP key was pressed to his great embarrassment. The key to be pressed should always be specified and the space bar is the ideal one to use.

Lines 218–220: A frequently-used question when dealing with data files on tape/disc. It would be called from the main program with a line such as:

```
2525 GOSUB218:IFY$ = "Y"THENCLOSE< FILE NUMBER>
```

Lines 221–225: If you are working with several programs for different purposes, then you will need a 'menu program' which can be loaded into the computer to remind you which programs are available and to call them up.

This section deals with the question of whether this menu program is now to be called up. This question section is only used when the operator has finished with the current set of data and, if the menu program is not required, then the program will be RUN from line 3. Many programmers will decree this technique to be sloppy programming, and they may be right, but it is very useful in that all the variables and arrays are cleared and the program started afresh. RUNning from line 3 ensures that the variable S6 is set to one and the security routine is not activated for a second time.

Lines 223–225: Deal with the loading of the menu program. This is done by printing the load instruction for the program on the screen, followed by 'RUN', and then moving the cursor down over the two instructions. The cursor is moved by POKEing into the keyboard buffer the codes for [HOME] and two RETURNs, and then telling the system that there are three characters in the keyboard buffer. It is, of course, possible to load a program with a direct instruction in a line, eg:

```
1000 LOAD"MENU"
```

but this causes problems if the program to be loaded is larger than the program which calls it up.

Lines 226–228: Used by the printout sequence.

Subroutine 5: Data Edit

This subroutine allows file data to be manipulated, checked, edited and so forth. The routine is complicated but has proved time and time again that it is worth the effort of entering. The main objectives of the routine are:

a) Insertion of new records.
b) Deletion of records from the file.
c) Editing and alteration of existing records.
d) Retention of records without alteration.

Further objectives or options are:

e) Specification of which record or records require editing.
f) Examination of all records, in sequence.
g) Addition of further records to the end of the file.

```
297 REM ##################################
##############
298 REM ***SUB 5.  EDIT DATA   3.6K   CO
MMODORE 64***
299 REM ##################################
##############
300 GOSUB823:GOSUB703:F9$=D9$:D9$="":GOS
UB733:IFER=1THENER=0:GOTO300
301  I=1:P4=80:REM ***MAXIMUM STRING LENG
TH***
302 PRINTW2$:PRINT"[CD][CD][RVS]TYPE 1[R
VO]  TO EDIT SPECIFIC RECORDS.[CD][CD]"
303 PRINT"[RVS]TYPE 2[RVO]  TO EDIT ALL
RECORDS.[CD][CD]":PRINTW1$
304 G$="":GETG$:IFG$=""THEN304
305 PRINTW2$:IFG$="1"THENRT=1:DIMK$(500)
:NN=1:GOTO308
306 IFG$="2"THENRT=0:GOTO315
307 GOTO304
308 PRINTW2$
309 PRINT"[CD][CD]TYPE IN RECORD NUMBERS
 TO BE EDITED."
310 PRINT"[CD][CD]IN ASCENDING ORDER - P
RESS RETURN AFTER"
311 PRINT"[CD][CD]EACH RECORD NUMBER IS
ENTERED."
312 PRINT"[CD][CD]TYPE  #  TO END NUMBER
 ENTRY[CD][CD]":PRINTW1$
313 L9=3:GOSUB150:IFD9$="#"THENK$(NN)="*
***T":D9$="":NN=1:NL=0:PRINTW2$:GOTO315
314 K$(NN)=D9$:D9$="":NN=NN+1:GOTO308
315 GOSUB770:NL=NL+1
316 IFRT=0THEN321
317 IFA9$="****T"THENA8$(I)="****T":GOSU
B210:GOTO433
318 IFK$(NN)="****T"THENRT=0:GOTO321
319 IFVAL(K$(NN))=NLTHENNN=NN+1:GOTO321
```

```
320 A8$(I)=A9$:I=I+1:GOTO315
321 PRINTW2$:PRINTA9$
322 PRINTW1$:PRINT"[CD]**SPACE BAR**  FO
R NO CHANGE REQUIRED."
323 PRINT"[CD][CD][RVS]1[RVO]  TO FINISH
 EDITING.[CD]"
324 PRINT"[CD][CD][RVS]2[RVO]  TO EDIT,
DELETE OR ADD A NEW LINE.[CD]":PRINTW1$
325 G$="":GETG$:IFG$=""THEN325
326 IFG$=CHR$(32)ORG$=CHR$(160)THENA8$(I
)=A9$:PRINTW2$:GOTO429
327 IFG$="1"THENPRINTW2$:GOTO435
328 IFG$="2"THEN330
329 GOTO325
330 PRINTW2$:PRINTA9$:PRINTW1$
331 PRINT"[CD][RVS]1[RVO]  TO INSERT A N
EW RECORD."
332 PRINT"[CD][RVS]2[RVO]  TO DELETE REC
ORD FROM FILE."
333 PRINT"[CD][RVS]3[RVO]  TO EDIT OR CH
ANGE RECORD.":PRINTW1$
334 G$="":GETG$:IFG$=""THEN334
335 G=VAL(G$):IFG<1ORG>3THEN334
336 ONGGOTO337,347,353
337 REM ***NEW RECORD***
338 GOSUB2000
339 IFD9$="****T"THENA8$(I)="****T"
341 PRINTW2$:PRINTA8$(I)
342 PRINT"[CD][CD]IS NEW RECORD TO BE IN
CLUDED?":PRINTW1$
343 GOSUB202:IFY$="N"THENA8$(I)="":GOTO3
21
344 IFA8$(I)="****T"THEN433
345 I=I+1:GOTO321
346 REM ***REMOVE RECORD FROM DATA FILE*
**
347 PRINTW2$:PRINTA9$
348 PRINT"[CD][CD]IS RECORD TO BE DELETE
D?":PRINTW1$
349 GOSUB202:IFY$="N"THEN321
350 IFA9$="****T"THENA8$(I)="****T":GOTO
433
351 GOTO315
352 REM ***EDIT RECORD***
```

```
353 PRINTW2$
354 PRINT"MOVE CURSOR ALONG RECORD AND C
ORRECT.[CD]"
356 PRINT"PRESS RETURN KEY WHEN COMPLETE
.":PRINTW1$
362 P=1544:P1=P:L%=LEN(A9$):P2=L%:P2=P2+
P1:P3=P1+P4:REM*MAX.STRING.LEN.*
367 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD]";A9$:PRINT
368 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD][CD][CD][CD][CD][CD]":PRINTA9$
369 A=PEEK(P):PRINT"[HOME][CD][CD][CD][C
D][CD][CD][CD][CD][CD][CD][CD][CD]"
370 POKE204,0:REM ***CURSOR FLASH***
371 G$="":GETG$:IFG$=""THEN371
372 G=ASC(G$):IFG=13ORG=141THENPOKE204,1
:POKEP,A:GOTO412:REM ***RETURN***
373 IFG=29THEN375:REM ***CURSOR RIGHT***
374 GOTO378
375 IFP+1>P2THENP2=P2+1
376 IFP2>P3THENP2=P2-1:GOTO371
377 B=PEEK(P+1):PRINTG$;:POKEP,A:P=P+1:A
=B:GOTO371
378 IFG=145THEN380:REM ***CURSOR UP***
379 GOTO386
380 IFP-40<P1THEN371
381 B=PEEK(P-40):PRINTG$;:POKEP,A:P=P-40
:A=B:GOTO371
386 IFG=157THEN388:REM ***CURSOR LEFT***
387 GOTO390
388 IFP=P1THENPOKEP,A:GOTO371
389 B=PEEK(P-1):PRINTG$;:POKEP,A:P=P-1:A
=B:GOTO371
390 IFG=17THEN392:REM ***CURSOR DOWN***
391 GOTO396
392 IFP+40>P2THENP2=P2+40
393 IFP2>P3THENP2=P2-40:GOTO371
394 B=PEEK(P+40):PRINTG$;:POKEP,A:P=P+40
:A=B:GOTO371
395 IFG=34THENG$=CHR$(162):GOTO408:REM**
QUOTE MARKS**
396 IFG=44ORG=58THEN408:REM **COMMA ETC.
 ALLOWED**
397 IFG=34ORG=44ORG=58THEN371:REM **COMM
```

```
A ETC. NOT ALLOWED**
404 IFG=32THENG$=CHR$(160):GOTO408
405 IFG<33ORG>95THEN407
406 GOTO408
407 IFG<193THENPOKEP,A:GOTO371
408 B=PEEK(P+1):IFP+1>P3THEN371
409 A=B:IFP+1=>P2THENP2=P2+1
410 IFP2>P3THENP2=P3
411 PRINTG$;:P=P+1:GOTO371
412 P=P1:PRINT"[HOME][CD][CD][CD][CD][CD
][CD][CD][CD][CD][CD][CD][CD][CD][CD][CD
][CD][CD][CD][CD]PLEASE WAIT.":IFP2>P3TH
ENP2=P3
413 A=PEEK(P):IFA=32THEN421
414 IFA<32THENA=A+64:GOTO421
415 IFA=96ORA=160THENA=160:GOTO421
417 IFA>64ANDA<91THENA=A+128:GOTO421
418 IFA=34ORA=98ORA=162THENA=162:GOTO421
:REM **QUOTE MARKS**
419 IFA=44ORA=108ORA=172THENA=172:GOTO42
1:REM **COMMA**
420 IFA=58ORA=122ORA=186THENA=186:REM **
COLON**
421 A$=CHR$(A):A8$(I)=A8$(I)+A$:P=P+1
422 IFP=P2THEN424
423 GOTO413
424 L%=LEN(A8$(I)):A=ASC(RIGHT$(A8$(I),1
))
425 IFA=32THENA8$(I)=LEFT$(A8$(I),L%-1):
GOTO424
426 PRINTW2$:PRINTA8$(I)
427 PRINT"[CD][CD]ARE YOUR CORRECTIONS C
ORRECT?[CD][CD]":PRINTW1$
428 GOSUB202:IFY$="N"THENA8$(I)="":GOTO3
21
429 IFA8$(I)="****T"THEN433
430 A9$="":L%=LEN(A8$(I)):A8$(I)=A8$(I)+
LEFT$(BL$,P4+1-L%):I=I+1:GOTO315
433 GOSUB870:GOSUB210:F8$=F9$:GOTO2615:R
EM***ADD NEW DATA ROUTINE***
434 GOSUB770
435 A8$(I)=A9$:A9$="":IFA8$(I)="****T"TH
EN433
436 I=I+1:GOTO434
```

Commentary

The routine as a whole is accessed directly from the program option subroutine (Subroutine 9). It is actually called with a GOTO rather than a GOSUB and so is not strictly a subroutine.

Lines 300–314: The routine opens with this section which initialises variables and allows the operator to specify the records to be edited.

Line 300: Uses later subroutines to request the name of the file to be accessed. If an error occurs in picking up that file, then execution returns to this line with the variable ER set to one and the file name request is repeated.

Line 301: The variable I will be used to record which line of the array A8$ is being acted upon. P4 is the maximum length of a record (ie the total number of characters) and will be different for each particular program application.

Lines 302–307: Ask whether specified records are to be examined, or all of them. If the 'specified record' option is chosen, the 'trigger' variable RT is set to one and is referred to each time a record is recalled.

Lines 308–314: If specified records are to be examined, these lines allow the operator to say which ones. Numbers should be entered in ascending order. The list of records is terminated by the entry of '#'.

Lines 315–329: This section picks up records from the file one by one, checking to see whether they have been specified for editing, and then giving the operator the opportunity to leave the record unchanged, delete it, alter it, insert a new record or finish editing.

Line 315: A record is picked up from the file and stored in A9$ by this call to a later subroutine. The counter N1 is incremented by one.

Line 316: The variable RT indicates whether the 'edit specific records' option has been chosen.

Line 317: '****T' is the end of file indicator.

Lines 318–320: Used if the 'edit specific records' option has been chosen. They test the array K$, in which are stored the numbers of the records to be edited, to see whether the '****T' terminator is present and, if not, whether the number of the current record is one to be edited. If it is, they add the record to the array A8$.

Lines 321–329: Print the current record to the screen and request the operator to specify whether editing is to take place.

Lines 330–336: The menu of editing options.

Lines 337–345: Deal with the insertion of a new record by a call to a later subroutine (Subroutine 10). The variable I is incremented to indicate that an extra record is present in A8$.

Lines 346–351: Deal with deletion. If the operator answers yes to deletion, the next record is picked up from the main file without the current record being added permanently to A8$.

Lines 352–411: These lines are the first part of the 'edit record' routine. The routine is fairly complex but, in essence, the principle is simple enough. The record to be altered is printed to the screen and the operator is asked to move the cursor along the data, changing incorrect characters along the way. This is done by the use of GET to scan for the input of the cursor right key.

One major problem here is the limitation of the user-controlled cursor created by POKE 204,0. This has an irritating tendency to appear and disappear unpredictably, and also to leave behind it inverse characters on the screen when it is moved. This is a fault the 64 shares with the VIC 20 and other Commodore machines. The problem for the program is that the second part of this routine will scan across the screen and pick up the edited record from the characters displayed there. Inverse characters on the screen can therefore lead to corruption of the record being edited. The routine deals with this problem by excluding inverse characters from the acceptable set of characters.

The method used is to print the first character of the record on to the screen and then to allow the user either to confirm it by moving on or to alter it. If the character is altered, the inverse character problem will not arise. It is possible, however, that the character printed is correct but that it is inverse. In this case, the inverse character is detected, the record in memory is examined again and the correct form of the character placed on the screen before the operator is asked to deal with the next character.

Although the actual editing relies upon the use of the cursor right key to confirm characters in the record, provision is also made for the use of the other cursor keys to allow the user to move around the record on the screen.

One final problem remains. The program which will be built up from these subroutines will store data in the form of 'packed strings', that is to say strings of separate items run together into one long string and stored in that form. When the data is being analysed, the program must know exactly where to split up those strings into the separate items. This can create a problem if deletions or additions are to be made. Take the following example of a possible record in the main file:

SMITH E + three shift/spaces + 10.99

This represents a simple record containing two items. SMITH E is part of a record containing 10 character spaces with the balance being made up of shift/spaces. 10.99 is another field of five character spaces. Supposing the record had been entered incorrectly as:

SMOITH E + two shift/spaces + 10.99

Clearly the 'O' in SMOITH would need to be removed. The simplest solution would appear to be to place the cursor over the 'I' and to press delete. The record would now read:

SMITH E + two shift/spaces + 10.99

When analysed by the program, this would yield two items of data:

SMITH E 1
0.99

a state of affairs that would frighten any accountant. The delete/insert key is therefore excluded from the routine. Even so, you might consider adding to each program you build up with this system an indicator, to the operator, of the length of the items within the overall record. This could be done by adding a new line:

357 GS$ = "*NAME*.....**NO*..*"

and then adding 'PRINT GS$' to the end of line 367. The length of the fields specified would need to be altered for each different use of the routine.

Only one further point is necessary for an understanding of this first part of the editing routine, and that is to do with the commands PEEK and POKE. You will no doubt already be aware of the function of these two commands in examining the contents of memory locations or in placing numbers into memory locations. In this routine, PEEK and POKE are used to examine the contents of the screen and to place new characters on the screen since what is displayed is simply another part of the 64's memory. Thus, although characters are appearing on the screen, there are no PRINT statements in this program section.

Line 362: Sets up the parameters which will record and control the use of the cursor. P is initially equal to 1544, the position on the screen of the first character of the record. P1 also equals 1544 and remains at that value, even though P will change as the cursor moves. P2 is equal to 1544 plus the length of the string containing the record, P3 is 1544 plus the maximum number of characters allowed, this already being stored in P4, which was set in line 301.

Lines 368–372: Display the record on the screen and then begin to examine the screen memory locations containing the display. The screen codes for each character being examined are stored in the variable A. Entry of

RETURN (CHR$(13) or (141)) terminates this first part of the editing routine.

Lines 373–377: Cope with the cursor right key. Line 373 determines whether the length of the record is being increased and, if so, increments P2. Line 376 checks to see that the maximum length allowed is being exceeded. If all is well, line 377 picks up the next character in the record and moves the cursor along.

Lines 378–394: Cope with the other three cursor keys.

Line 395: If a quotation mark is present, it is altered by this line to avoid any chance of corruption of the screen display.

Lines 396–397: These two lines are interchangeable, depending on whether the comma and colon are to be acceptable characters or not (see the previous comments on the difference between INPUT # and GET #).

Lines 412–436: This is the second part of the 'edit record' routine, placing the edited record into memory and asking for confirmation of the changes.

Line 412: The cursor position variable P is reset to the beginning of the string and PLEASE WAIT is placed on the screen. If the maximum record length has been executed, the record is truncated.

Lines 413–423: Read the contents of the screen. The lines convert what is PEEKed from the screen into the ASCII values of the relevant characters (for the difference between ASCII values and SCREEN CODE values, see the tables at the back of your User Manual).

Lines 418–420: Cope with the screen codes of quotation marks, commas and colons, which require special treatment since they will not be stored as such.

Line 421: Each character picked up is added to A8$ and the cursor location moved on one.

Lines 424–425: If the record has been shortened, extra spaces may have been added to the end when P2 characters were read from the screen. These two lines strip those spaces out.

Line 430: The record is measured and, if necessary, padded out with shift/spaces.

Line 433: The file from which we are recalling data is closed and a call made

to a later subroutine to store the amended data.

Lines 434–436: The rest of the records are added to the file.

Subroutine 6: Printers

This subroutine is intended to deal with the various types of printer available for use with the 64 and it may not be necessary if you always intend to use the same printer type. The question of setting up a program for different printers is usually avoided by microcomputer books for the reason that there are such a large number of different printers, many of them having unique codes for control commands. This subroutine is not exhaustive: it deals with only some of the printers which I have used with the system developed for this book. If you have a different printer then you should be able to use the principles employed here to link it up.

The 1515 printer supplied by Commodore is a dot matrix type and is fully compatible with the 64. It can be plugged directly into the computer without the need for any kind of extra equipment. Other types of printer can be utilised, but they will generally need an 'interface' between the printer and the 64 to convert the signals produced by the 64 into a form which the printer can recognise and act upon. When buying an interface, it is essential that it is the correct one and you should either insist on a demonstration from your dealer using *exactly* the printer you intend to use or get a guarantee that the interface can be returned if it is not satisfactory. With the right interface, a wide variety of printers can be added to your 64, though the Commodore graphics set and the control characters in listings will probably not be available.

Given the potential for confusion, it is fortunate that there are a range of command codes which most manufacturers have adhered to and which are therefore common to most types of printer. To access the printer in the first place, a command such as OPEN 4,4 needs to be used, where the first of the two figures can be any valid file number and the second is the device number for the printer and must be 4 (some printers have the ability to configure themselves as device 5, but this is very rarely used). It is advisable to stick to pairs of numbers such as 4,4 wherever possible, in order to avoid using the same file number for two purposes at the same time. Thus 1,1 would normally be used for the cassette deck and 8,8 for the disc drive. Opening the same file to two different devices at the same time will stop the program with an error.

The subroutine as given here is designed to deal with six different types of printer and hopefully one of them will suit your purposes. There are several different functions built in, including upper case only, upper and lower case, line spacing, lines per page, underlining and so on.

```
443 REM ####################################
########################
444 REM ***SUB 6.  3.3K     PRINTERS VARI
OUS    COMMODORE 64***
445 REM ####################################
########################
446 REM** P1$=P1 P2 UPPER & LOWER CASE**
447 REM **P2$=P0 P1 P2 P3 LINE SPACES**
448 REM **P3$ P1 TO P6 PRINTER TYPE**P4$
=Y OR N SCREEN PRINT-OUT ONLY**
449 REM **A$ A8$ C$ CC$**U5$ PRINTER LIN
ES**TT LINES PER PAGE**
450 PRINTW2$:PRINT"[CD][RVS]WHICH OPTION
 ??"
451 PRINT"[CD][RVS]1[RVO]  IF YOUR DATA
IS OF UPPER CASE          CHARACTERS ONLY
."
452 PRINT"[CD][CD][RVS]2[RVO]  IF YOUR D
ATA CONTAINS UPPER & LOWER  CASE CHARACT
ERS.[CD][CD]"
453 PRINTW1$
454 G$="":GETG$:IFG$=""THEN454
455 G=VAL(G$):IFG<1ORG>2THEN454
456 P1$="P"+G$:RETURN
457 PRINTW2$:PRINT"[CD][RVS]WHICH OPTION
 ??"
458 PRINT"[CD][RVS]1[RVO]  FOR DOT MATRI
X PRINTER (TYPE 1)."
459 PRINT"[CD][RVS]2[RVO]  FOR DOT MATRI
X PRINTER (TYPE 2)."
460 PRINT"[CD][RVS]3[RVO]  FOR PRINTER (
TYPE 3)."
461 PRINT"[CD][RVS]4[RVO]  FOR TYPE 4 OR
 SCRIPTA II (ESW 103)."
462 PRINT"[CD][RVS]5[RVO]  FOR PRINTER (
TYPE 5)."
463 PRINT"[CD][RVS]6[RVO]  FOR A QUME SP
RINT 5 PRINTER.":PRINTW1$
464 G$="":GETG$:IFG$=""THEN464
465 G=VAL(G$):IFG<1ORG>6THEN464
466 P3$="P"+G$:IFP3$="P6"THEN468
467 PRINTW2$:RETURN
468 PRINTW2$:PRINT"[CD]PLACE AUTO SWITCH
 IN THE OFF POSITION.":PRINTW1$
```

```
469 GOSUB210:RETURN
470 I=I+1:IFP4$="Y"THENRETURN
471 IFP1$="P2"THEN479
472 REM ***UPPER CASE PRINT-OUT***
473 N=1:L=LEN(A$)
474 C$="":C$=MID$(A$,N,1)
475 C=ASC(C$):IFC=>192ANDC<219THENC=C-12
8:C$=CHR$(C)
476 CC$=CC$+C$:N=N+1:IFN=L+1THENPRINT#4,
CC$:CC$="":GOTO489
477 GOTO474
478 REM ***UPPER AND LOWER CASE***
479 IFP3$="P5"ORP3$="P6"THEN483
480 IFP3$="P3"ORP3$="P4"THEN482
481 PRINT#4,CHR$(17)A$:GOTO489
482 PRINT#4,A$:GOTO489
483 N=1:L=LEN(A$)
484 C$="":C$=MID$(A$,N,1)
485 C=ASC(C$):IFC=>192ANDC<219THENC=C-12
8:C$=CHR$(C):GOTO487
486 IFC>64ANDC<91THENC=C+128:C$=CHR$(C)
487 CC$=CC$+C$:N=N+1:IFN=L+1THENPRINT#4,
CC$:CC$="":GOTO489
488 GOTO484
489 IFP3$="P2"ORP3$="P4"ORP3$="P6"THENPR
INT#4,CHR$(10)
490 GOTO522
491 REM ***LINES FOR DOT-MATRIX  QUME  &
  SCRIPTA***
492 IFP3$="P6"THEN497
493 IFP3$="P4"THEN500
494 REM ***LINE FOR DOT-MATRIX PRINTER**
*
495 A$="":V5$=CHR$(163):FORJ=1TO80:A$=A$
+V5$:NEXT:PRINT#4,A$:I=I+1:RETURN
496 REM ***LINE FOR QUME PRINTER***
497 A$="":V5$="":FORJ=1TO80:V5$=CHR$(95)
:A$=A$+V5$:NEXTJ
498 PRINT#4,CHR$(27)CHR$(68)CHR$(27)CHR$
(68)A$:PRINT#4,CHR$(10):RETURN
499 REM ***LINE FOR SCRIPTA PRINTER***
500 A$="":V5$="":FORJ=1TO80:V5$=CHR$(95)
:A$=A$+V5$:NEXTJ
501 PRINT#4,CHR$(27)CHR$(7)CHR$(27)CHR$(
```

```
7)A$:PRINT#4,CHR$(10):RETURN
502 PRINTW2$:PRINT"[CD][CD][CD]PLACE PAP
ER IN THE PRINTER.[CD][CD][CD]"
503 PRINTW1$:GOSUB210:RETURN
504 PRINTW2$:PRINT"[CD][CD][CD]PLACE A N
EW SHEET OF PAPER IN PRINTER.[CD][CD][CD
]"
505 PRINTW1$:GOSUB210:RETURN
506 PRINTW2$
507 PRINT"[CD][CD][CD]HOW MANY SPACES BE
TWEEN LINES?[CD][CD]"
508 PRINT"0 = NO SPACES   MAXIMUM NUMBER
 = 999.[CD][CD][CD]":PRINTW1$:PRINT"[CD]
."
509 G$="":GETG$:IFG$=""THEN509
510 G=VAL(G$):IFG>3THEN509
511 P2$="P"+G$:RETURN
512 PRINTW2$
513 PRINT"[CD][CD][CD]HOW MANY LINES PER
 PAGE ARE REQUIRED?[CD][CD][CD]":PRINTW1
$
514 L9=3:GOSUB150:TT=VAL(D9$):D9$=""
515 RETURN
516 PRINTW2$:PRINT"[CD][CD][RVS]1[RVO] F
OR SCREEN DISPLAY ONLY OF DATA.[CD]"
517 PRINT"[RVS]2[RVO] FOR SCREEN DISPLAY
 AND PRINT-OUT.[CD][CD]":PRINTW1$
518 G$="":GETG$:IFG$=""THEN518
519 G=VAL(G$):IFG>2THEN518
520 P4$="Y":IFG=2THENP4$="N"
521 D9$="":RETURN
522 IFP2$="P0"THENRETURN:REM ***SPACES B
ETWEEN LINES***
523 IFP3$="P2"ORP3$="P4"ORP3$="P6"THEN52
8
524 PRINT#4:I=I+1
525 IFP2$="P2"THENPRINT#4:I=I+1:RETURN
526 IFP2$="P3"THENPRINT#4:PRINT#4:I=I+2
527 RETURN
528 PRINT#4,CHR$(10):I=I+1
529 IFP2$="P2"THENPRINT#4,CHR$(10):I=I+1
:RETURN
530 IFP2$="P3"THENPRINT#4,CHR$(10):PRINT
#4,CHR$(10):I=I+2
```

```
531 RETURN
532 REM ***DATA PRINT-OUT***
533 GOSUB823:GOSUB703:F8$=D9$:D9$=""
534 TR=0:GOSUB730:IFER=1THEN533
535 GOSUB516:GOSUB512:IFP4$="Y"THEN539
536 OPEN4,4:IFTR=1THEN538
537 GOSUB450:GOSUB457:TR=1
538 GOSUB506
539 I=0:GOSUB216:IFY$="Y"THEN542
540 GOSUB869:GOSUB901:IFP4$="N"THENCLOSE
4
541 RUN3
542 IFI=>TTTHENGOSUB210:GOTO548
543 GOSUB766:A$=A8$:A8$="":GOSUB470:IFA$
="****T"THEN545
544 GOTO542
545 GOSUB869:GOSUB901:IFP4$="N"THENCLOSE
4
546 GOSUB210:GOSUB226:IFY$="Y"THEN534
547 RUN3
548 IFP4$="N"THENCLOSE4
549 GOTO535
```

Commentary

Lines 450–456: Deal with the upper case only option. If the information to be printed is in upper and lower case, any lower case characters will first be translated into upper case. The variable P1$ is allocated the option number and will be equal to either P1 or P2.

Lines 457–467: Deal with the choice of printer: option 1 is suitable for the Commodore 1515 printer. Options 4 and 6 relate to an Olympia Scripta II (ESW 103) and a Qume Sprint 5, both daisy wheel printers which I have used extensively. Their inclusion is not intended as an indication of a personal preference on my part, but merely as an illustration of different printer types. Note that Qume, in common with several other printer types, has an on/off switch for automatic line feed. In the off position, unless the printer is given a specific command to advance the paper when a line is finished, all the information will be printed on the same line. Several other printers also need to be told when to advance the paper so, when using the Qume, you are asked to have the switch in the off position so that no special provision has to be made for this type. If you wish to print out program listings, then you will need to remember to switch the automatic line feed back on.

Line 470: A record is kept of the number of lines printed. If P4$ is equal to 'Y', then the current record is printed only to the screen, thus allowing records to be selected for printing.

Lines 472–490: This section deals with upper case only and upper/lower case printing. The material to be printed must first have been stored in the variable A$. Thus if the record recalled from a disc file is A8$, then the correct sequence for using this subroutine will be to use a line containing:

```
A$ = A8$:GOSUB470
```

Lines 472–477: Scan the record to be printed out for lower case characters and transform them into upper case by manipulating their ASCII code. The routine will work for printer types 1, 2, 5 and 6, but not for 3 and 4 since printers of this type require that any character with a code of less than 63 has 128 added to its code.

Lines 479–481: For printer types 3 to 6, line 481 is jumped over. CHR$(17) is a control code and sending it to the printer puts the printer into the correct mode for upper/lower case printing.

Line 482: The printer is directed to print out the record as received.

Lines 483–488: If the characters in the record were printed out directly, the slightly eccentric Commodore character set would mean that, on most printers, the upper case characters would come out as lower case and vice versa. The characters have, therefore, to be converted by these lines to the opposite case.

Line 489: Determines whether the special line feed character is necessary or not.

Line 490: Directs the program to the part of the routine which specifies the number of blank lines between each line of text, for instance, for double spacing.

Lines 491–501: This section deals with underlining text on the three different printer types. On a dot matrix printer the necessary control code to achieve underlining is CHR$(163), while on the Qume and Olympia Scripta it is CHR$(95).

Lines 494–495: For a dot matrix printer, the usual procedure is that a line is printed, then a line feed executed, and then the line is underlined from below. At the end of this, another line feed is executed, thus giving a blank line underneath the underlined text. If this is undesirable, the line feed after the underlining can be turned off by entering the following modification:

```
495 A$ = "":V5$ = CHR$(163):FORJ = 1TO80:A$ = A$ + V5$:NEXT:
    PRINT #4,A$CHR$(141):RETURN
```

The addition of CHR$(141) will allow the carriage return, but will suppress the line feed so that the increment to the number of lines counted has to be deleted. Underlining is seldom very satisfactory with common dot matrix printers, the end result usually being that the underline is superimposed on to the bottom edge of the current record or on to the top edge of the next. The only satisfactory solution is to have a line space between each line of text. Daisy wheel printers, usually the product of typewriter manufacturers, are equipped to cope with such difficulties, and incorporate a special underline symbol so placed as to clear the bottom edge of the text without fouling the line below.

Lines 496–501: Demonstrate the control characters necessary for dealing with daisy wheel type printers. Since the underline character on these works on the same line as the text, rather than from the line below, the line feed following the printing of the text has to be suppressed. Note that, although the code for underline is the same for the Qume and the Olympia, it need not be so for every daisy wheel printer. Refer to your printer manual for the correct code for your own printer.

Lines 502–521: This routine covers a number of self-explanatory questions which need to be answered before any text is printed.

Lines 522–531: This routine allows for up to three spaces to be specified between lines of text. The routine is accessed even if no spaces have been asked for.

Lines 532–549: This routine recalls records from the file on tape/disc in order to print them out. It is accessed from Subroutine 9, the program menu.

Lines 533–534: The operator is prompted to place the tape/disc into the recorder/drive and, if a disc drive is being used, the disc is checked to see that it is a data disc and not the program disc — if not, an error message is displayed and the operator may change the disc. The file name is requested and the file opened in readiness for reading. If the file specified is not present, an appropriate error message is displayed and the operator is given another chance. The variable ER is set to one, by Subroutine 8, to indicate errors to the current routine.

Lines 535–539: These calls deal with printer types, lines per page and so on. If the display is to be to the screen only, the instruction to open a file to the printer is bypassed. A number of lines from the record are then displayed and the questions given again so that further output may be printed. In this way the operator can scan through the records on the screen, printing only those for which hard copy is required.

Lines 540–541: Close the various files if the operator specifies that further printout is not required.

Lines 542–549: Implement the printout of the records, keeping track of lines printed and so forth. On reaching the end of the records, as indicated by the file terminator entry '****T', the various files are closed and the command RUN3 returns execution to the program menu.

Subroutine 7: Numeric Calculation

This subroutine deals with financial calculations and is essential to any application concerning accounts. It should be noted that calculations involving decimal points can sometimes produce unexpected results and that it is better to work with integers (whole numbers). Numbers with a negative value can also produce anomalous results in calculations and a special routine is included to deal with this.

```
596 REM ##################################
####################
597 REM ***SUB 7.    1.2K     CALCULATIONS
     COMMODORE 64***
598 REM ##################################
####################
599 REM ***CONVERT POUNDS AND PENCE TO P
ENCE AS AN INTEGER***
600 R=INT(R*100+.5):RETURN
601 REM ***TO ROUND OFF NUMBERS AFTER TH
E DECIMAL POINT***
602 REM ***E.G. 22.5 WOULD = 22.50***
603 REM ***E.G. 22   WOULD = 22.00***
604 REM ***E.G.   .9 WOULD = 0.90***
605 REM ***E.G.  0   WOULD = 0.00***
606 REM ***PROVISION IS MADE FOR NEGATIV
E NUMBERS***
607 IFN1$="0"THENN1$="0.00":RETURN
608 IFVAL(N1$)<0THEN616
609 N1=VAL(N1$):N1$=STR$(N1):N2=INT(N1):
N6$=".00":N7$="0":N2$=STR$(N2)
610 IFN1=N2THEN612
611 IFN1>N2THEN613
612 N1$=N1$+N6$:RETURN
613 N3=N1-N2:N3=N3+.0001
614 N3$=STR$(N3):N3$=MID$(N3$,2,3)
615 N1$=N2$+N3$:RETURN
616 N1=VAL(N1$):N1=ABS(N1):N1$=STR$(N1):
```

```
GOSUB609:L=LEN(N1$)
617 N1$="-"+RIGHT$(N1$,L-1):RETURN
618 REM **CONVERT TOTAL CREDIT & DEBITS
TO POUNDS & PENCE & PADDED STRINGS**
619 C3=0:C3=C1/100:N1$="":N1$=STR$(C3):G
OSUB607:C3$="":C3$=N1$
620 L%=0:L%=LEN(C3$):C3$=LEFT$(BL$,10-L%
)+C3$:RETURN
621 D3=0:D3=D1/100:N1$="":N1$=STR$(D3):G
OSUB607:D3$="":D3$=N1$
622 L%=0:L%=LEN(D3$):D3$=LEFT$(BL$,10-L%
)+D3$:RETURN
```

Commentary

Line 600: Converts pounds and pence to pence only, in order that we may work exclusively with whole numbers. The main program must store the value to be converted in the variable R before line 600 is accessed.

Lines 601–617: This routine standardises to two decimal places, whether or not two places were entered by the operator. Thus, if 22.3 was entered, the actual format produced by this routine would be 22.30. The routine also deals with the formatting of negative numbers. In use, the routine can be employed to standardise format before numeric items are stored in a data file or before they are printed out.

Lines 618–622: Deal with the reconversion of pence to pounds and pence. When this routine is called, the number to be acted upon is stored in C3 (in the case of credits) or D3 (in the case of debits) and the final result is made part of a 10-character string padded out with shift/spaces.

Subroutines 8t or 8d: Data Handling (*tape or disc*)

These two subroutines deal with storage and recall of data on tape or disc respectively. The line numbers of the two subroutines correspond closely and so they cannot be used in the same program. Together they allow you to make a choice of your preferred method of storage or to change the system at a later date. A disc drive capability can be added to a tape system by deleting the tape routines from a program and merging in the disc subroutine.

Subroutine 8t — tape

```
697 REM ##################################
########################
```

```
698 REM ***SUB 8T.  3.2K   COMMODORE 64
  1 CHANNEL ONLY OPEN***
699 REM ##################################
##########################
700 REM ***FILE NAME FOR NEW APPEND ETC.
***
701 PRINTW2$
702 PRINT"[CD][CD][CD]TYPE IN NEW FILE N
AME[CD][CD][CD]":GOTO707
703 PRINTW2$
704 PRINT"[CD][CD][CD]NAME OF THE FILE Y
OU WISH TO ACCESS?[CD][CD][CD]":PRINTW1$
705 PRINTW1$:DF=1:L9=16:GOSUB150:GOSUB20
0:IFY$="N"THEND9$="":GOTO704
706 GOTO708
707 DF=1:L9=16:GOSUB150:GOSUB200:IFY$="N
"THEND9$="":GOTO701
708 L9=0:DF=0:RETURN:REM ***D9$ CARRIES
THE FILE NAME***
709 REM ***OPEN NEW FILE FOR DATA STORAG
E***
710 OPEN8,1,1,(F8$)
712 RETURN
720 RETURN
729 REM ***OPEN FILES F8$ & F9$ FOR DATA
 RECALL**
730 OPEN8,1,0,(F8$)
732 RETURN
733 OPEN9,1,0,(F9$)
735 RETURN
753 REM ***TO RECALL DATA USING GET STAT
EMENT***
766 G$="":GET#8,G$:IFG$=""THEN766
767  IFG$=CHR$(13)THENA8$=GG$:GG$="":PRIN
TA8$:RETURN
768  IFST<>0THEN813
769 GG$=GG$+G$:GOTO766
770 G$="":GET#9,G$:IFG$=""THEN770
771  IFG$=CHR$(13)THENA9$=GG$:GG$="":PRIN
TA9$:RETURN
772  IFST<>0THEN813
773 GG$=GG$+G$:GOTO770
774 REM ***TO RECALL DATA USING INPUT ST
ATEMENT***
```

```
781 INPUT#8,A8$:IFST<>0THEN813
782 RETURN
783 INPUT#9,A9$:IFST<>0THEN813
784 RETURN
785 REM ***STORE DATA ON TAPE FILE***
792 PRINT#8,A8$:PRINTA8$:IFST<>0THEN813
794 RETURN
795 PRINT#9,A9$:PRINTA9$:IFST<>0THEN813
797 RETURN
798 RETURN
813 IFA8$="****T"ORA9$="****T"THENRETURN
814 PRINTW2$:PRINT"[CD][CD]A TAPE ERROR
STATUS NO:- ";ST
815 PRINT"[CD]HAS BEEN RETURNED AND THE
PROGRAM WILL  NOW BE RE-STARTED."
816 PRINT"[CD][CD]PLEASE CORRECT THE ERR
OR.[CD][CD][CD]"
817 CLOSE8:CLOSE9:GOSUB210:RUN3
818 RETURN
823 PRINTW2$
824 PRINT"[CD][CD]      PLACE DATA TAPE I
N RECORDER.[CD][CD]"
825 PRINT"ENSURE NO CLEAR TAPE LEADER IS
 VISIBLE.[CD][CD]":PRINTW1$:GOSUB210
827 RETURN
828 RETURN
831 PRINTW2$
832 PRINT"[CD][CD]PLACE YOUR WORKING TAP
E IN THE RECORDER.[CD][CD]"
833 GOSUB825
836 RETURN
840 PRINTW2$:PRINT"[CD][CD]YOUR DATA HAS
 BEEN RECORDED ON FILE:-[CD]":PRINTF8$
841 PRINT"[CD][CD][CD]DO YOU REQUIRE A C
OPY TAPE OF THE DATA?[CD][CD][CD]":PRINT
W1$
842 GOSUB202:IFY$="N"THENRETURN
843 I=1:GOSUB823:GOSUB710:IFER=1THENER=0
:GOTO843
844 IFA8$(I)="****T"THENA8$="****T":GOSU
B792:GOSUB869:GOSUB901:GOSUB850:RETURN
845 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GOT
O844
850 RETURN
```

```
861 RETURN
865 CLOSE:RETURN:REM **CLOSE ALL FILES**
869 CLOSE8:RETURN
870 CLOSE9:RETURN
872 RETURN
875 RETURN
879 RETURN
882 RETURN
890 REM ***SCREEN PRINT-OUT OF DATA FILE
***
891 GOSUB823:GOSUB703:F8$=D9$:GOSUB730:I
=1:IFER=1THENER=0:GOTO891
892 GOSUB766:A8$(I)=A8$:IFA8$="****T"THE
NGOSUB869:I=1:PRINTW2$:GOTO894
893 I=I+1:GOTO892
894 PRINT"[CD][CD][CD]TO SLOW DOWN THE S
CREEN DISPLAY[CD]"
895 PRINT"PRESS THE CTRL KEY.[CD][CD][CD
]":PRINTW1$:GOSUB210:PRINTW2$
897 PRINTA8$(I):IFA8$(I)="****T"THENGOSU
B210:GOSUB901:PRINTW2$:GOTO899
898 I=I+1:GOTO897
899 PRINT"[CD][CD]REPEAT DISPLAY OF DATA
?[CD][CD]":PRINTW1$:GOSUB202:IFY$="N"THE
NRUN3
900 I=1:PRINTW2$:GOTO894
901 PRINTW2$:PRINT"[CD][CD][CD][CD][CD][C
CD]              REWIND TAPE.[CD][CD][CD]
[CD][CD][CD]":PRINTW1$:GOSUB210
902 RETURN
910 RETURN
930 REM ***TO CREATE A NEW FILE NAME USI
NG THE MONTH & YEAR***
931 DATAJANUARY,FEBRUARY,MARCH,APRIL,MAY
,JUNE,JULY,AUGUST,SEPTEMBER
932 DATAOCTOBER,NOVEMBER,DECEMBER,JANUAR
Y
933 L=LEN(M1$):M2$=LEFT$(M1$,L-3)
934 READM3$:IFM2$=M3$THENREADM4$:M2$=M4$
:GOTO936
935 M3$="":GOTO934
936 IFM2$="JANUARY"THENM5=VAL(RIGHT$(M1$
,2)):M5$=STR$(M5+1):GOTO938
937 GOTO939
```

```
938 M5$=RIGHT$(M5$,2):FF$=M2$+" "+M5$:RE
TURN
939 FF$=M2$+" "+RIGHT$(M1$,2):RETURN
```

Commentary

Lines 697–708: Establish the file name to be accessed by calling up the standard input routine entered earlier. The variable DF is set to one, which prevents the operator from entering a shift/space which would lead to a corrupt file name. For example, MARCH ACCS., if the space were actually a shift/space, would go into memory as MARCH'ACCS. and the file name would not be found on the tape — the file would be stored as MARCH. The maximum length of a file name is 16 characters and the variable L9 is set to 16 to indicate this maximum length. Depending on whether the file is for storage or recall, the file name will be allocated to F8$ or F9$.

A tape system has the disadvantage of only being able to open one communication channel at a time. Data can be recalled from a file but that file must be closed before another file can be accessed. It is therefore impossible to recall part of a file, change the tape and store that information and then recall further information from the first where it left off.

Lines 709–712: Open a file to allow the computer to store data on to tape under the file name in F8$. The logical file number is 8, which is chosen purely to correspond with the variable name.

Lines 729–735: The corresponding routine for data recall from tape. The file number in this case is 9.

Lines 766–769: We have already seen, in the section on input of data, some of the main differences between INPUT# and GET# as methods of retrieving data from tape/disc. These lines represent a routine to pick up a record from tape using GET#.

Lines 774–784: The parallel routine using INPUT#. The main program will determine which of these two is used when the program is executed. Note that it is possible to get some of the benefits of both GET# and INPUT# by splitting records which are too long to be picked up by INPUT# into two or three sections, then picking up the sections with INPUT# and concatenating them into one string. Beware, however, of creating strings longer than 255 characters, which will cause the program to stop with an error.

Lines 785–797: Print a record, contained in A8$ or A9$, to tape.

Lines 813–818: These lines deal with the situation which occurs if a 'status error' is detected in one of the foregoing program sections by means of the system variable ST.

Line 813: Here the value of ST dealt with is 64, which indicates that an end of file marker has been found. The file is checked to see whether the entry picked up is the file terminator '****T' and, if not, an error message is called up.

Lines 814–818: Print out a general-purpose error message, indicating that there is a problem with the file on tape. The routine could be expanded to take account of particular values of ST and to give appropriate error messages for them, eg IF ST = 64 THEN PRINT "ERROR MESSAGE = END OF FILE":GOSUB 210. Note that the storing of data records to tape or disc is best carried out frequently, even as frequently as re-storing every time a new record is entered. Problems with tape or disc do occur from time to time and, if they have already been saved, then even a corrupt file can be largely retrieved by use of judicious editing.

Lines 823–861: These lines deal with the recording of data, the first section ensuring that data is not lost by recording on the clear 'leader' of the cassette. The second section confirms that a file has been saved and gives the operator the option of duplicating the file. A duplicate can be made on the same tape by ignoring the instruction to rewind the tape.

Line 861: This RETURN is included to take care of future expansion of the system to use disc drives.

Lines 865–882: This program section is employed by the main program to close all or some files. The series of RETURNs at the end are to allow future disc routines to be added.

Lines 890–900: This section allows a data record to be recalled from tape and then displayed on the screen. Because of the unfortunate idiosyncrasy of the 64 which switches off the screen whenever the tape-recorder is accessed, all the information is loaded into an array before it is displayed. When the last record has been loaded, the records can then be displayed, using the CTRL key if necessary to slow the process down.

Lines 901–910: The instruction to the operator to rewind the tape.

Lines 930–939: A potentially useful routine which allows the operator to create file names based on dates, for instance for accounts programs. Provision is made for year changes, ie December 84 to January 85. The routine begins with the input of the current file name, such as 'MARCH 84', and

assumes that the format is the month name, followed by a space, followed by two digits representing the year. The month name is obtained in the variable M2$ and then compared with the month names stored in the data statements. Having found the correct month, the next month is taken from the data statements. If the new month is January then the year is incremented by one. Finally, the new month name and year digits are combined to form the new file name.

Subroutine 8d — disc

The inclusion of this subroutine will allow the tape system to be expanded to include a disc drive. Note that the two subroutines cannot be included at the same time. This book has been designed to cater first of all for a tape data system, which is where most people begin, and to allow expansion when the limitations of tape become too irksome. Those who wish to expand into a disc-based system will need to refer to their disc manual as they proceed with the entry of this subroutine since space does not permit the full exposition of the available disc commands.

```
697 REM ##################################
########################
698 REM ***SUB 8D.  COMMODORE 64  6.1K
5 CHANNELS OPEN MAX.***
699 REM ##################################
########################
700 REM ***FILE NAME FOR NEW APPEND ETC.
***
701 PRINTW2$
702 PRINT"[CD][CD]TYPE IN NEW FILE NAME.
[CD][CD]":GOTO707
703 PRINTW2$
704 PRINT"[CD][CD]NAME OF THE FILE YOU W
ISH ACCESS?[CD][CD]"
705 PRINTW1$:DF=1:L9=16:GOSUB150:GOSUB20
0:IFY$="N"THEND9$="":GOTO704
706 GOTO708
707 DF=1:L9=16:GOSUB150:GOSUB200:IFY$="N
"THEND9$="":GOTO701
708 L9=0:DF=0:RETURN:REM ***D9$ CARRIES
THE FILE NAME***
709 REM ***OPEN NEW FILE FOR DATA STORAG
E***
710 ER=0:OPEN8,8,8,"0:"+F8$+",S,W":GOSUB
```

```
910:IFDS=63THENER=1:GOTO713
711 IFDS<>0THENCLOSE8:GOSUB818:GOTO710
712 RETURN
713 CLOSE8:PRINTW2$:PRINT"[CD][CD]FILE:-
 ";F8$:GOTO715
714 PRINTW2$:PRINT"[CD][CD]FILE:- ";F9$
715 PRINT"[CD]ALREADY EXISTS AND A NEW F
ILE NAME[CD][CD]"
717 PRINT"[CR][CR][CR][CR][CR][CR][CR][C
R][CR][CR]IS REQUIRED.[CD][CD]"
718 GOSUB210:RETURN:REM ***ER=ERROR INDI
CATOR***
719 REM ***OPEN FILE FOR DATA APPEND***
720 OPEN8,8,8,"0:"+F8$+",S,R":GOSUB910:I
FDS=62ORDS=64THENER=1:CLOSE8:GOTO723
721 IFDS<>0THENCLOSE8:GOSUB818:GOTO720
722 ER=0:RETURN
723 PRINTW2$:PRINT"[CD][CD]FILE:- ";F8$:
GOTO725
724 PRINTW2$:PRINT"[CD][CD]FILE:- ";F9$
725 PRINT"[CD][CD]DOES NOT EXIST ON  YOU
R DATA DISC[CD][CD]"
726 PRINT"[CD]CHECK ENTRY OF FILE NAME I
S CORRECT.[CD]"
727 PRINT"CORRECT DISC IS IN DRIVE.[CD]"
:PRINTW1$
728 GOSUB210:RETURN.
729 REM ***OPEN FILES F8$ & F9$ FOR DATA
 RECALL**
730 OPEN8,8,8,"0:"+F8$+",S,R":GOSUB910:I
FDS=62ORDS=64THENER=1:CLOSE8:GOTO723
731 IFDS<>0THENCLOSE8:GOSUB818:GOTO730
732 ER=0:RETURN
733 OPEN9,8,9,"0:"+F9$+",S,R":GOSUB910:I
FDS=62ORDS=64THENER=1:CLOSE9:GOTO724
734 IFDS<>0THENCLOSE9:GOSUB818:GOTO733
735 ER=0:RETURN
736 REM ***OPEN TEMPORARY FILES TO STORE
 DATA***
737 F7$="T7":PRINT#15,"S:"+F7$:OPEN7,8,7
,F7$+",S,W"
738 GOSUB910:IFDS<>0THENCLOSE7:GOSUB818:
GOTO737
739 RETURN
```

```
740 F6$="T6":PRINT#15,"S:"+F6$:OPEN6,8,6
,F6$+",S,W"
741 GOSUB910:IFDS<>0THENCLOSE6:GOSUB818:
GOTO740
742 RETURN
743 F5$="T5":PRINT#15,"S:"+F5$:OPEN5,8,5
,F5$+",S,W"
744 GOSUB910:IFDS<>0THENCLOSE5:GOSUB818:
GOTO743
745 RETURN
746 REM ***OPEN TEMPORARY FILES TO RECAL
L DATA***
747 F7$="T7":OPEN7,8,7,(F7$)+",S,R":GOSU
B910:IFDS<>0THENCLOSE7:GOSUB818:GOTO747
748 RETURN
749 F6$="T6":OPEN6,8,8,(F6$)+",S,R":GOSU
B910:IFDS<>0THENCLOSE6:GOSUB818:GOTO749
750 RETURN
751 F5$="T5":OPEN5,8,5,(F5$)+",S,R":GOSU
B910:IFDS<>0THENCLOSE5:GOSUB818:GOTO751
752 RETURN
753 REM ***TO RECALL DATA FROM DISC USIN
G GET STATEMENT***
754 G$="":GET#5,G$:IFG$=""THEN754
755 GOSUB910:IFDS<>0THEN813
756 IFG$=CHR$(13)THENA5$=GG$:GG$="":PRIN
TA5$:RETURN
757 GG$=GG$+G$:GOTO754
758 G$="":GET#6,G$:IFG$=""THEN758
759 GOSUB910:IFDS<>0THEN813
760 IFG$=CHR$(13)THENA6$=GG$:GG$="":PRIN
TA6$:RETURN
761 GG$=GG$+G$:GOTO758
762 G$="":GET#7,G$:IFG$=""THEN762
763 GOSUB910:IFDS<>0THEN813
764 IFG$=CHR$(13)THENA7$=GG$:GG$="":PRIN
TA7$:RETURN
765 GG$=GG$+G$:GOTO762
766 G$="":GET#8,G$:IFG$=""THEN766
767 GOSUB910:IFDS<>0THEN813
768 IFG$=CHR$(13)THENA8$=GG$:GG$="":PRIN
TA8$:RETURN
769 GG$=GG$+G$:GOTO766
770 G$="":GET#9,G$:IFG$=""THEN770
```

```
771 GOSUB910:IFDS<>0THEN813
772 IFG$=CHR$(13)THENA9$=GG$:GG$="":PRIN
TA9$:RETURN
773 GG$=GG$+G$:GOTO770
774 REM ***TO RECALL DATA FROM DISC USIN
G INPUT STATEMENT***
775 INPUT#5,A5$:GOSUB910:IFDS<>0THEN813
776 RETURN
777 INPUT#6,A6$:GOSUB910:IFDS<>0THEN813
778 RETURN
779 INPUT#7,A7$:GOSUB910:IFDS<>0THEN813
780 RETURN
781 INPUT#8,A8$:GOSUB910:IFDS<>0THEN813
782 RETURN
783 INPUT#9,A9$:GOSUB910:IFDS<>0THEN813
784 RETURN
785 REM ***STORE DATA ON DISC FILE***
786 PRINT#5,A5$:GOSUB910:IFDS<>0THEN813
787 RETURN
788 PRINT#6,A6$:GOSUB910:IFDS<>0THEN813
789 RETURN
790 PRINT#7,A7$:GOSUB910:IFDS<>0THEN813
791 RETURN
792 PRINT#8,A8$:ER=0:GOSUB910:IFDS=72THE
NER=2:CLOSE8:GOTO798:REM*72=DISC FULL*
793 IFDS<>0THEN813
794 RETURN
795 PRINT#9,A9$:ER=0:GOSUB910:IFDS=72THE
NER=2:CLOSE9:GOTO798
796 IFDS<>0THEN813
797 RETURN
798 PRINTW2$:PRINT"[CD]THE DATA DISC IS
FULL.[CD]"
799 PRINT"REMOVE DISC FROM DRIVE & REPLA
CE WITH"
800 PRINT"[CD]A NEW DISC. REMAINING DATA
 WILL BE ON[CD]"
801 PRINT"FILE:- T4  THE DISC UTILITY PR
OGRAM WILL"
802 PRINT"[CD]COMBINE FILES USING CONCAT
 SEQUENCE.[CD]":PRINTW1$
803 GOSUB210
811 F8$="T4":GOSUB710:IFER=1THENER=0:GOT
O799
```

```
812 RETURN
813 PRINTW2$
814 PRINT"[CD][CD]DISC ERROR ";DS;" = ";
DS$
815 PRINT"[CD]THE PROGRAM WILL NOW BE RE
-STARTED."
816 PRINT"[CD][CD]PLEASE NOTE ERROR AND
CORRECT.[CD][CD]"
817 GOSUB210:RUN3
818 PRINTW2$
819 PRINT"[CD][CD]DISC ERROR ";DS;" = ";
DS$
820 PRINT"[CD]HAS BEEN RETURNED."
821 PRINT"[CD][CD]PLEASE NOTE ERROR AND
CORRECT.[CD][CD]"
822 GOSUB210:RETURN
823 PRINTW2$
824 PRINT"[CD][CD]PLACE DATA DISK IN DRI
VE.[CD][CD]":PRINTW1$:GOSUB210
825 OPEN10,8,10,"DATA DISC"+",S,R":GOSUB
910:IFDS=620RDS=64THENCLOSE10:GOTO828
826 IFDS<>0THENGOSUB818:CLOSE10:GOTO823
827 CLOSE10:RETURN
828 PRINTW2$
829 PRINT"[CD][CD]DATA DISC IS NOT IN DR
IVE AS REQUESTED.[CD][CD]"
830 PRINTW1$:GOSUB210:GOTO823
831 PRINTW2$
832 PRINT"[CD][CD]PLACE WORKING DISK IN
DRIVE.[CD][CD]"
833 PRINTW1$:GOSUB210
834 OPEN10,8,10,"WORK DISC"+",S,R":GOSUB
910:IFDS=620RDS=64THENCLOSE10:GOTO837
835 IFDS<>0THENCLOSE10:GOSUB818:GOTO831
836 CLOSE10:RETURN
837 PRINTW2$
838 PRINT"[CD][CD]WORKING DISC IS NOT IN
 DRIVE.[CD][CD]"
839 PRINTW1$:GOSUB210:GOTO831
840 PRINTW2$:PRINT"[CD][CD]YOUR DATA HAS
 BEEN RECORDED ON FILE:-[CD]":PRINTF8$
841 PRINT"[CD][CD]DO YOU REQUIRE A COPY
DISC OF THE DATA?[CD][CD]":PRINTW1$
842 GOSUB202:IFY$="N"THENRETURN
```

```
843 PRINT"[CD][CD]PLACE COPY DATA DISK I
N DRIVE.[CD][CD]":PRINTW1$:GOSUB210:PRIN
TW2$
844 OPEN10,8,10,"COPY DISC"+",S,R":GOSUB
910:IFDS=62ORDS=64THENCLOSE10:GOTO846
845 GOTO847
846 PRINT"[CD][CD]COPY DATA DISC IS NOT
IN DRIVE.[CD][CD]":PRINTW1$:GOSUB210:GOT
O840
847 I=1:GOSUB710:IFER=1THENER=0:GOTO847
848 IFA8$(I)="****T"THENA8$="****T":GOSU
B792:GOSUB869:GOSUB850:RETURN
849 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GOT
O844
850 PRINTW2$
851 PRINT"[CD][CD]REMOVE COPY DATA DISK
AND"
852 PRINT"[CD][CD]RE-PLACE WITH PROGRAM
DISK.[CD][CD]":PRINTW1$:GOSUB210
853 OPEN10,8,10,"PROG DISC"+",S,R":GOSUB
910:IFDS=62ORDS=64THENCLOSE10:GOTO857
855 IFDS<>0THENCLOSE10:GOSUB818:GOTO850
856 CLOSE10:RETURN
857 PRINTW2$
858 PRINT"[CD][CD]PROGRAM DISC IS NOT IN
 DRIVE.[CD][CD]"
859 PRINTW1$:GOSUB210:GOTO850
860 REM ***TO TEST WHETHER A DISC FILE I
S PRESENT OR NOT***
861 OPEN8,8,8,(F8$)+",S,R":GOSUB910:TR$=
"N":IFDS=63THENTR$="Y"
862 CLOSE8:RETURN
865 CLOSE15:REM ***CLOSE ALL DISC FILES*
**
866 CLOSE5:RETURN
867 CLOSE6:RETURN
868 CLOSE7:RETURN
869 CLOSE8:RETURN
870 CLOSE9:RETURN
871 REM ***RENAME A FILE***
872 PRINT#15,"R:"+(FF$)+"="+(FL$):RETURN
:REM *FL$=OLD NAME  FF$=NEW NAME*
874 REM ***COPY FILES***
875 PRINT#15,"S:"+(FL$):PRINT#15,"C:"+(F
```

```
L$)+"="+(FF$):RETURN
878 REM ***REMOVE DATA FILE***
879 PRINT#15,"S:"+(FF$):RETURN
881 REM ***COMBINE DATA FILES***
882 PRINTW2$:PRINT"[CD][CD]NAME OF FILE
TO BE ADDED TO?[CD][CD]":PRINTWI$:GOSUB7
07:F8$=D9$
883 PRINTW2$:PRINT"[CD][CD]NAME OF FILE
TO BE ADDED?[CD][CD]":PRINTWI$:GOSUB707:
F9$=D9$
884 GOSUB730:GOSUB737
885 GOSUB781:IFA8$="****T"THENGOSUB869:G
OSUB733:GOTO887
886 GOSUB790:GOTO885
887 GOSUB783:IFA9$="****T"THENGOSUB790:G
OSUB868:GOSUB870:GOTO889
888 GOSUB790:GOTO887
889 PRINT#15,"S:"+(F8$):PRINT#15,"R:"+(F
7$):PRINT#15,"S:"+(F7$):RETURN
890 REM ***SCREEN PRINT-OUT OF DISC DATA
***
891 GOSUB823:GOSUB703:F8$=D9$:GOSUB730:I
=1:IFER=1THENER=0:GOTO891
892 GOSUB766:A8$(I)=A8$:IFA8$="****T"THE
NGOSUB869:I=1:PRINTW2$:GOTO894
893 I=I+1:GOTO892
894 PRINT"[CD]TO SLOW DOWN THE SCREEN DI
SPLAY[CD]"
895 PRINT"PRESS THE CTRL KEY.[CD]":PRINT
W1$:GOSUB210
897 PRINTA8$(I):IFA8$(I)="****T"THENGOSU
B210:GOSUB901:PRINTW2$:GOTO899
898 I=I+1:GOTO897
899 PRINT"[CD][CD]REPEAT DISPLAY OF DATA
?[CD][CD]":PRINTW1$:GOSUB202:IFY$="N"THE
NRUN3
900 I=1:PRINTW2$:GOTO894
901 RETURN
910 INPUT#15,DS,DS$
930 REM ***TO CREATE A NEW FILE NAME USI
NG THE MONTH & YEAR***
931 DATAJANUARY,FEBRUARY,MARCH,APRIL,MAY
,JUNE,JULY,AUGUST,SEPTEMBER
```

```
932 DATAOCTOBER,NOVEMBER,DECEMBER,JANUAR
Y
933 L=LEN(M1$):M2$=LEFT$(M1$,L-3)
934 READM3$:IFM2$=M3$THENREADM4$:M2$=M4$
:GOTO936
935 M3$="":GOTO934
936 IFM2$="JANUARY"THENM5=VAL(RIGHT$(M1$
,2)):M5$=STR$(M5+1):GOTO938
937 GOTO939
938 M5$=RIGHT$(M5$,2):FF$=M2$+" "+M5$:RE
TURN
939 FF$=M2$+" "+RIGHT$(M1$,2):RETURN
```

Commentary

Lines 697–707: These lines mirror the similar section in Subroutine 8t except that the disc system is capable of dealing with five channels at the same time, each allocated to a different file. Thus the operator may write to one file, read from another and so on, without constantly opening and closing files.

Lines 709–728: Permit the opening of a new file to receive records or the addition of records to an existing file. The error status system for disc files is far more sophisticated than that for tapes and the greater range of error numbers allows programs to be written which will exercise more control over the operator's actions. Access to the error-checking system is achieved by including the command OPEN 15,8,15. The first 15 represents the communication channel number, the 8 is the device number of the disc and the second 15 is the secondary address of the disc 'command channel', which gives access to the disc error codes. Commodore disc drives are 'intelligent drives'; that is to say, they contain their own operating systems on read-only chips. All the work of operating the disc drive and checking on the progress of saving or loading is therefore carried out in the drive itself, unlike many other machines which use a 'disc operating system' (DOS) contained within the micro itself.

The command CLOSE 15 will close the error channel and, in doing so, any other disc file which is open at the time. For this reason, it is wise to open the error channel early in the program and to close it only at program termination (see lines 4 and 9999 in Subroutine 1 and line 224 in Subroutine 4).

Lines 708–910: Deal with the opening of a new file to receive data records. In line 710, the GOSUB 910 accesses the disc error channel and the disc

status number (variable DS) is returned. If the value of DS is 63 then the file already exists on the disc and a new file of that name cannot be created for data storage. The variable ER is set to one and the program branches to lines 713–718, where the operator is informed of the error. Other types of error are detected by line 711 which causes a branch to lines 818–822. Here the variable DS$ is printed, this being the error message supplied by the disc operating system to go with the error number contained in DS.

One thing to watch out for is the fact that the error status variable will always be set to zero immediately after a file has been closed. Thus to include a line like:

```
OPEN 8,8,8,F8$ + ",S,W" : CLOSE 8 : GOSUB 910 : IF DS = 63 THEN
713
```

would actually provide no information as to whether the file specified was already on the disc.

Lines 719–728: These lines open an existing file in preparation for the addition of further data records. A test is made for error code 62, which would indicate that the file specified is not present on the disc.

Lines 729–735: These lines open an existing file for the recall of data.

Lines 736–745: This section deals with temporary files called T5, T6 and T7. These can be useful for the storage of records when a large amount of data is being analysed. When placing data into temporary files, previous temporary files of the same name are 'scratched' from the disc.

Lines 746–752: The temporary files are opened for data record recall.

Lines 753–817: These lines are similar to the equivalent section of Subroutine 8t, except for the use of the disc error-checking routine at 910 and the variable DS rather than ST. When an error has been detected, the program will be restarted and all disc files closed.

Lines 785–812: Are concerned with the storage of data records on disc, but special provision has been made in lines 792 and 795 for a DISC FULL error. If this is returned, ie disc error number 72, the file is closed and the routine branches to lines 798–812. The operator is informed of the position and is asked to place a new (formatted) disc in the drive. The remaining data can then be recorded on the new disc under file name T4.

It is worth noting that data stored on disc will be lost unless the file is properly closed. If the disc full error is returned, enough space is reserved on the disc to close the file, but only one file. It is preferable to check the

contents of the data disc and establish whether there is enough space left, ie place the data disc in the drive and type in LOAD"$",8 followed by RETURN. The number of blocks available on the disc are then displayed by LIST.

Lines 823–839 and **lines 850–859:** Concerned with instructing the operator to place different discs into the drive, and with checking that the correct discs have been inserted. The error-checking routine is again brought into use, and plays an important part in making the system operator-compatible. For this type of system to work, each disc must have a file with the disc reference contained in its file name — eg to determine whether the data disc is in the drive, a file with the name 'DATA DISC' must be present. To accomplish this, place a formatted disc in the drive and enter the following:

OPEN8,8,8,"DATA DISC,S,W":PRINT#8,"****T":CLOSE8

and press RETURN. A file named 'DATA DISC', will be recorded on to the disc and can be used as the disc identity code. Repeat this procedure for the program disc, etc.

Lines 840–849: Will make a copy of a data file on to a copy or duplicate disc if required. It is advisable to make a copy of all data records routinely, in case a disc is accidentally damaged or lost.

Lines 865–870: These lines deal with the closing of the various disc files.

Lines 871–879: Deal with renaming files, copying a data file under a new name on to the same disc and deleting or scratching a file. The file names FF$ and FL$ must be established in the main program before calling this section.

Lines 881–889: Concerned with the combining of two data files. As both files will contain the file termination code '****T', the first step will be to place the file to be added to into a temporary file without the termination code. The data records from the file to be added or appended are recalled and appended to the temporary file and will retain the termination code. The file is closed, the original file deleted or scratched, and the temporary file is renamed with the original file name.

Lines 890–900: The same as those listed for Subroutine 8t.

Lines 930–939: Equivalent to the section on creating 'month name' files in Subroutine 8t.

Subroutine 9: Program Menu

This is the program menu, giving the operator a choice of program functions.

```
997 REM ##################################
#######
998 REM ***SUB 9.  0.6K  MENU   COMMODOR
E 64***
999 REM ##################################
#######
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO START A NEW DATA FI
LE."
1002 PRINT"[CD]2  TO ADD NEW DATA TO A F
ILE."
1003 PRINT"[CD]3  TO EDIT A DATA FILE."
1004 PRINT"[CD]4  FOR SCREEN DISPLAY OF
DATA FILE."
1005 PRINT"[CD]5  TO PRINT A DATA FILE."
1006 PRINT"[CD]6  "
1007 PRINT"[CD]7  "
1008 PRINT"[CD]8  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]9  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>9THEN1010
1012 PRINTW2$:ONGGOTO2500,2600,300,891,5
33,0000,0000,221,9999
```

Subroutine 10: File Construct

This is the subroutine which does the work of creating a data record for a new or an already existing file.

```
1996 REM ##################################
########################################
1997 REM ***SUB 10. CREATE NEW OR ADD TO
 EXISTING FILE.  3.5K  COMMODORE 64***
1998 REM ##################################
########################################
1999 PRINTW2$:PRINT"DATA ENTRY NUMBER:-
```

```
 ";K:PRINTW1$:RETURN
2000 K=1:GOSUB1999:PRINT"[CD]TYPE IN INF
ORMATION 1.[CD]"
2005 PRINT"                          OR[CD]"
2010 PRINT"#  SIGN TO FINISH ADDDING DAT
A.[CD][CD]"
2015 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2020 IFD9$="#"THEND9$="****T":RETURN
2025 B1$=D9$:L%=LEN(B1$):B1$=B1$+LEFT$(B
L$,10-L%):K=K+1
2030 GOSUB1999:PRINT"[CD]TYPE IN INFORMA
TION 2.[CD]"
2035 PRINT"                          OR[CD]"
2040 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2045 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2050 IFD9$="#"THENB2$=LEFT$(BL$,10):B3$=
LEFT$(BL$,10):K=K+2:GOTO2090
2055 B2$=D9$:L%=LEN(B2$):B2$=B2$+LEFT$(B
L$,10-L%):K=K+1
2060 GOSUB1999:PRINT"[CD]TYPE IN INFORMA
TION 3.[CD]"
2065 PRINT"                          OR[CD]"
2070 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2075 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2080 IFD9$="#"THENB3$=LEFT$(BL$,10):GOTO
2090:K=K+1
2085 B3$=D9$:L%=LEN(B3$):B3$=B3$+LEFT$(B
L$,10-L%):K=K+1
2090 GOSUB1999:PRINT"[CD]TYPE IN INFORMA
TION 4.[CD]"
2095 PRINT"                          OR[CD]"
2100 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2105 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2110 IFD9$="#"THEN2116
2115 B4$=D9$:L%=LEN(B4$):B4$=B4$+LEFT$(B
L$,10-L%):K=K+1:GOTO2120
2116 B4$=LEFT$(BL$,10):B5$=LEFT$(BL$,10)
```

```
2117 B6$=LEFT$(BL$,10):B7$=LEFT$(BL$,10)
:K=K+4:GOTO2210
2120 GOSUB1999:PRINT"[CD]TYPE IN INFORMA
TION 5.[CD]"
2125 PRINT"                    OR[CD]"
2130 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2135 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2140 IFD9$="#"THEN2146
2145 B5$=D9$:L%=LEN(B5$):B5$=B5$+LEFT$(B
L$,10-L%):K=K+1:GOTO2150
2146 B5$=LEFT$(BL$,10):B6$=LEFT$(BL$,10)
:B7$=LEFT$(BL$,10):K=K+3:GOTO2210
2150 GOSUB1999:PRINT"[CD]TYPE IN INFORMA
TION 6.[CD]"
2155 PRINT"                    OR[CD]"
2160 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2165 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2170 IFD9$="#"THENB6$=LEFT$(BL$,10):B7$=
LEFT$(BL$,10):K=K+2:GOTO2210
2175 B6$=D9$:L%=LEN(B6$):B6$=B6$+LEFT$(B
L$,10-L%):K=K+1
2180 GOSUB1999:PRINT"[CD]TYPE IN INFORMA
TION 7.[CD]"
2185 PRINT"                    OR[CD]"
2190 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2195 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2200 IFD9$="#"THENB7$=LEFT$(BL$,10):K=K+
1:GOTO2210
2205 B7$=D9$:L%=LEN(B7$):B7$=B7$+LEFT$(B
L$,10-L%):K=K+1
2210 GOSUB1999:PRINT"[CD]TYPE IN INFORMA
TION 8.[CD]"
2215 PRINT"                    OR[CD]"
2220 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2225 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2230 IFD9$="#"THENB8$=LEFT$(BL$,10):GOTO
```

```
2240
2235 B8$=D9$:L%=LEN(B8$):B8$=B8$+LEFT$(B
L$,10-L%)
2240 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]":PRINTB1$:PRINTB2$:PRINTB3
$
2245 PRINTB4$:PRINTB5$:PRINTB6$:PRINTB7$
:PRINTB8$:PRINTW1$:GOSUB200
2250 IFY$="N"THEN2490
2450 A8$(I)=B1$+B2$+B3$+B4$+B5$+B6$+B7$+
B8$
2460 RETURN
2490 B1$="":B2$="":B3$="":B4$="":B5$="":
B6$="":B7$="":B8$="":GOTO2000
2499 REM ***START A NEW FILE***
2500 I=1:GOSUB823:GOSUB701:F8$=D9$:D9$="
":GOSUB710:IFER=1THENER=0:GOTO2500
2505 GOSUB2000:IFD9$="****T"THEN2525
2510 A8$=A8$(I):GOSUB792:B1$="":B2$="":B
3$="":B4$="":B5$="":B6$=""
2515 B7$="":B8$="":GOTO2505
2525 GOSUB218:IFY$="N"THEND9$="":GOTO250
5
2530 A8$(I)="****T":A8$=A8$(I):GOSUB792:
GOSUB869:GOSUB901:IFAE$="Y"THENGOSUB879
2535 GOSUB840:RUN3
2599 REM ***ADD NEW DATA TO FILE***
2600 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO2600
2605 GOSUB781:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO2615
2610 I=I+1:GOTO2605
2615 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:AE$="Y"
2620 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO2620
2625 IFA8$(I)="****T"THENGOSUB210:GOTO26
35
2630 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO2625
2635 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO ADD MORE INFORMATION?[CD][CD]":PRINTW
1$
2640 GOSUB202:IFY$="Y"THENA8$(I)="":GOTO
```

```
2505
2645 GOTO2530:REM **CLOSE FILE**
```

Commentary

Line 1999: Keeps a tally on the number of entries per single run.

Lines 2000–2025: Deal with the entry of the first part of the data record. Provision is made to allow the operator to terminate the entry routine and close the record file using the hash (#) sign. The procedure followed, in all programs referred to, is to include the file terminator code '****T' at the end of a record file. On recall of file records, each one is examined for this code and the end of a file is easily established.

Line 2015: Allocates the number of characters allowed for the entry by stipulating the value of variable L9. This figure can be changed to suit the needs of the particular system.

As can be seen in line 2450, all eight entries are combined to form one data record or string. When this record is examined by a program dealing with the analysis of the data, it is essential to know the exact position in the record of each individual entry or field. Each field could consist of different numbers of characters, so it is necessary to build each one up to the same length with shift/spaces. Irrespective, therefore, of the number of characters entered, line 2025 will produce an entry or field of exactly 10 characters. This sequence is repeated throughout the routine and will result in a combined record of eight fields totalling 80 characters. It also means that an analysis program can pick out the correct field from a record. Provision has been made for no information being available for a particular field, by allocating 10 shift/spaces, ie LEFT$(BL$,10) to that field as in line 2080.

Lines 2240–2490: Allow the operator to check the entries and to decide whether they are correct or not.

The fields are combined to form one data string and are allocated to the subscript variable A8$(I). The records are stored on tape/disc at the time of allocation to prevent the loss of the data if a fault should arise further on in the process.

Lines 2499–2535: Control the creation of a new file and are accessed from the program menu routine.

Line 2500: Sets the subscript variable counter I to one, and accesses the routine to open a new file. Note the check procedure for an open file error, ie ER = 1.

Line 2505: Branches to the data entry routine and checks whether the file terminator code is returned.

Line 2510: Stores the record of eight fields, then empties the field variables before returning for the next record entry. If the code terminator is returned, the operator is given the option to cancel the entry in line 2525 and to continue adding further data.

Line 2530: Includes the terminator code and closes the file. In a tape system the operator is requested to rewind the tape, ie GOSUB901. This can be ignored to allow a duplicate of the file to be stored after the original. It is advisable, however, to make a copy on a separate tape. The variable AE$ relates to the 'add new data' routine.

Line 2535: Accesses the copy data file routine.

Lines 2599–2645: Deal with the addition of new data records to an existing file. The new records are appended to the end of the file. The principle of the routine is to load the file into the computer and to resave it.

In a tape system the operator is requested to rewind the tape and replace it with another data tape. The records are then stored on to the new tape until the file terminator code '****T' is detected. This code is not written to file, and the routine branches to the 'add further information' option.

With a disc system, the records are again loaded into the computer and the original file is renamed 'temp' in line 872. The records stored in memory are then written to a disc file using the original file name, minus the terminator code. The 'add further information' option is branched to and, if no further records are to be appended, the file terminator code is included at the end of the file and the file is closed. The disc system then deletes the 'temp' file from the disc before returning to the program menu routine. If, however, further records are required to be appended, the routine branches to line 2505. On completion, the file terminator code is included and the file is closed — the disc system will then delete the 'temp' file from the disc.

The inclusion of the 'temp' file sequence is to safeguard against a power failure or some form of accident with the inevitable loss of records held in computer memory. The fact that the original records are held in the 'temp' file will ensure that they are safe and that only those being appended will be lost. This is unnecessary with the tape system, since records are safe on their separate tapes.

CHAPTER 3
Business Accounts System

In building up the programs in Chapters 3 to 7, the procedure is as follows:

1) You will be given a list of the subroutines from Chapter 2 to be included in the program. If you have entered the subroutines already, then the merge program from Chapter 2, Subroutine 1, can be used to pick them up from tape and combine them.

2) You will then be given a list of amended lines. Not all of these lines need necessarily be altered. Refer to the text and you will see that the program can have different capabilities according to your needs. Choose the kind of capabilities you want and then refer to the text for the lines which have to be altered to conform with the listings in this chapter.

3) In certain areas, new lines must be added and these are specified in full.

The Business Accounts System deals with the financial transactions related to supplying customers with goods and/or services, maintaining a stock list linked to the accounts system, accounting for value added tax (VAT) at 15 per cent, and a final analysis including assets obtained from an equipment file. Throughout, VAT on purchases and VAT on sales are distinguished as 'credit' or 'purchase' tax and as 'debit' or 'sales' tax.

The system is divided into three main sections:

1) Day-to-day accounting.
2) Stock control.
3) Analysis of 1) and 2) to produce balance sheets, invoices, VAT comparison and the preparation of the customer/supplier balance file for the following month's accounts.

1. Day-to-day accounting

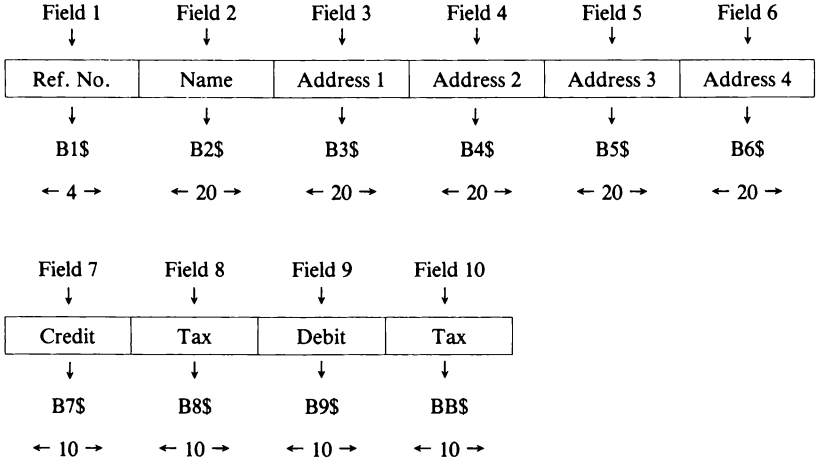
This section deals with the initial setting up of a customer and supplier reference file and with day-to-day transactions, and it consists of two programs.

Please notice in these programs that the £ sign appears as \ in the computer listings.

a. Customer/Supplier Balance

This program will produce a file containing customer or supplier reference numbers, names and addresses, and the state of each account plus VAT references. The file will be referred to for invoice and statement printouts and balance sheets — a new file will be created at the end of each month with an updated state of each account. A diagrammatic layout of a record file is illustrated in **Figure 3.1.**

Figure 3.1



I've called this program the Customer/Supplier Balance program, and the file which it produces the Customer/Supplier Balance file. The program will allow file editing, inclusion and deletion of records and a file content printout. The routines included are:

1) Subroutine 1
2) Subroutine 2
3) Subroutine 3a
4) Subroutine 4
5) Subroutine 5
6) Subroutine 6
7) Subroutine 7
8) Subroutine 8t *or* 8d
9) Subroutine 9

Modifications are listed below — Subroutine 10 may be included, but is completely changed to the listing below from line 1996 forward.

Amended listing

```
1 REM ***CUSTOMER/SUPPLIER BALANCE PROGR
AM COMMODORE 64***
5 W2$="[CLR][RVS]CUSTOMER/SUPPLIER BALAN
CE PROGRAM        [HOME][CD]"
10 DIMA8$(999),A9$(999)
301 I=1:P4=144:REM ***MAXIMUM STRING LEN
GTH***
357 W5$="[RVS]NO. NAME                    AD
DRESS 1.          ADDRESS 2.             "
358 W6$="[RVS]ADDRESS 3.          ADDRES
S 4.          CREDIT    CR. TAX    "
359 W7$="[RVS]DEBIT      DR. TAX    "
360 W5$=W5$+W6$+W7$
367 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD]";W5$:PRINT
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO START A NEW BALANCE
 FILE."
1002 PRINT"[CD]2  TO ADD NEW DATA TO BAL
ANCE FILE."
1003 PRINT"[CD]3  TO EDIT BALANCE FILE."
1004 PRINT"[CD]4  FOR SCREEN DISPLAY OF
DATA FILE."
1005 PRINT"[CD]5  TO PRINT BALANCE FILE.
"
1008 PRINT"[CD]6  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]7  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>7THEN1010
1012 PRINTW2$:ONGGOTO2500,2600,300,891,5
33,221,9999
1999 PRINTW2$:PRINT"DATA ENTRY NUMBER:-
 ";K:PRINTW1$:RETURN
2000 K=1:GOSUB1999:PRINT"[CD]TYPE IN CUS
TOMER/SUPPLIER[CD]"
2002 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC]REFERENCE NUMBER.[C
D]"
2004 PRINT"#  SIGN TO FINISH ADDDING DAT
```

```
A.[CD][CD]"
2006 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=3:GOSUB100:B0$="00000"
2008 IFD9$="#"THEND9$="****T":RETURN
2010 B1$=D9$:L%=LEN(B1$):B1$=LEFT$(B0$,4
-L%)+B1$:K=K+1
2015 GOSUB1999:PRINT"[CD]TYPE IN NAME OF
 FIRM OR SURNAME.[CD]"
2020 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=15:GOSUB150
2025 B2$=D9$:K=K+1
2060 GOSUB1999:PRINT"[CD]TYPE IN TITLE O
F PERSON MR MRS ETC[CD]"
2065 PRINT"#  SIGN IF DEALING WITH A FIR
M.[CD][CD]"
2070 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=3:GOSUB150
2075 IFD9$="#"THEN2085
2080 B2$=D9$+"[SPC]"+B2$
2085 L%=LEN(B2$):B2$=B2$+LEFT$(BL$,20-L%
):GOTO2090:K=K+1
2090 GOSUB1999:PRINT"[CD]TYPE IN 1ST LIN
E OF ADDRESS[CD]"
2095 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2100 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2105 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2110 IFD9$="#"THEN2116
2115 B3$=D9$:L%=LEN(B3$):B3$=B3$+LEFT$(B
L$,20-L%):K=K+1:GOTO2120
2116 B3$=LEFT$(BL$,20):B4$=LEFT$(BL$,20)
2117 B5$=LEFT$(BL$,20):B6$=LEFT$(BL$,20)
:K=K+4:GOTO2210
2120 GOSUB1999:PRINT"[CD]TYPE IN 2ND LIN
E OF ADDRESS[CD]"
2125 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2130 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2135 PRINT"FOLLOWED BY RETURN[CD]":PRINT
```

```
W1$:L9=19:GOSUB150
2140 IFD9$="#"THEN2146
2145 B4$=D9$:L%=LEN(B4$):B4$=B4$+LEFT$(B
L$,20-L%):K=K+1:GOTO2150
2146 B4$=LEFT$(BL$,20):B5$=LEFT$(BL$,20)
:B6$=LEFT$(BL$,20):K=K+3:GOTO2210
2150 GOSUB1999:PRINT"[CD]TYPE IN 3RD LIN
E OF ADDRESS[CD]"
2155 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2160 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2165 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2170 IFD9$="#"THENB5$=LEFT$(BL$,20):B6$=
LEFT$(BL$,20):K=K+2:GOTO2210
2175 B5$=D9$:L%=LEN(B5$):B5$=B5$+LEFT$(B
L$,20-L%):K=K+1
2180 GOSUB1999:PRINT"[CD]TYPE IN 4TH LIN
E OF ADDRESS[CD]"
2185 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2190 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2195 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2200 IFD9$="#"THENB6$=LEFT$(BL$,20):K=K+
1:GOTO2210
2205 B6$=D9$:L%=LEN(B6$):B6$=B6$+LEFT$(B
L$,20-L%):K=K+1
2210 GOSUB1999:PRINT"[CD]TYPE IN CREDIT
BALANCE.[CD]"
2215 PRINT"                    OR[CD]"
2220 PRINT"#  SIGN IF BALANCE IS NIL.[CD
][CD]"
2225 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2230 IFD9$="#"ORD9$="0"THENB7$="[SPC][SP
C][SPC][SPC][SPC][SPC]0.00":B8$="[SPC][S
PC][SPC][SPC][SPC][SPC]0.00":K=K+2:GOTO2
270
2235 N1$=D9$:GOSUB607:B7$=N1$:L%=LEN(B7$
```

```
):B7$=LEFT$(BL$,10-L%)+B7$:K=K+1
2240 GOSUB1999:PRINT"[CD]TYPE IN CREDIT
TAX.[CD]"
2245 PRINT"                         OR[CD]"
2250 PRINT"#  SIGN IF BALANCE IS NIL.[CD
][CD]"
2255 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2260 IFD9$="#"ORD9$="0"THENB8$="[SPC][SP
C][SPC][SPC][SPC][SPC]0.00"::K=K+1:GOTO2
270
2265 N1$=D9$:GOSUB607:B8$=N1$:L%=LEN(B8$
):B8$=LEFT$(BL$,10-L%)+B8$:K=K+1
2270 GOSUB1999:PRINT"[CD]TYPE IN DEBIT B
ALANCE.[CD]"
2275 PRINT"                         OR[CD]"
2280 PRINT"#  SIGN IF BALANCE IS NIL.[CD
][CD]"
2285 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2290 IFD9$="#"ORD9$="0"THENB9$="[SPC][SP
C][SPC][SPC][SPC][SPC]0.00":BB$="[SPC][S
PC][SPC][SPC][SPC][SPC]0.00":GOTO2435
2295 N1$=D9$:GOSUB607:B9$=N1$:L%=LEN(B9$
):B9$=LEFT$(BL$,10-L%)+B9$:K=K+1
2300 GOSUB1999:PRINT"[CD]TYPE IN DEBIT T
AX.[CD]"
2305 PRINT"                         OR[CD]"
2310 PRINT"#  SIGN IF BALANCE IS NIL.[CD
][CD]"
2315 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2320 IFD9$="#"ORD9$="0"THENBB$="[SPC][SP
C][SPC][SPC][SPC][SPC]0.00":GOTO2435
2325 N1$=D9$:GOSUB607:BB$=N1$:L%=LEN(BB$
):BB$=LEFT$(BL$,10-L%)+BB$
2435 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]":PRINTB1$:PRINTB2$:PRINTB3
$
2440 PRINTB4$:PRINTB5$:PRINTB6$:PRINTB7$
:PRINTB8$
2441 PRINTB9$:PRINTBB$:PRINTW1$:GOSUB200
2445 IFY$="N"THEN2490
2450 A8$(I)=B1$+B2$+B3$+B4$+B5$+B6$+B7$+
```

```
B8$+B9$+BB$
2460 RETURN
2490 B1$="":B2$="":B3$="":B4$="":B5$="":
B6$="":B7$="":B8$=""
2495 B9$="":BB$="":GOTO2000
2499 REM ***START A NEW FILE***
2500 I=1:GOSUB823:GOSUB701:F8$=D9$:D9$="
":GOSUB710:IFER=1THENER=0:GOTO2500
2505 GOSUB2000:IFD9$="****T"THEN2525
2510 A8$=A8$(I):GOSUB792:B1$="":B2$="":B
3$="":B4$="":B5$="":B6$=""
2515 B7$="":B8$="":B9$="":BB$="":GOTO250
5
2525 GOSUB218:IFY$="N"THEND9$="":GOTO250
5
2530 A8$(I)="****T":A8$=A8$(I):GOSUB792:
GOSUB869:GOSUB901:IFAE$="Y"THENGOSUB879
2535 GOSUB840:RUN3
2599 REM ***ADD NEW DATA TO FILE***
2600 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO2600
2605 GOSUB766:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO2615
2610 I=I+1:GOTO2605
2615 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:AE$="Y"
2620 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO2620
2625 IFA8$(I)="****T"THENGOSUB210:GOTO26
35
2630 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO2625
2635 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO ADD MORE INFORMATION?[CD][CD]":PRINTW
1$
2640 GOSUB202:IFY$="Y"THENA8$(I)="":GOTO
2505
2645 GOTO2530:REM **CLOSE FILE**
```

Commentary

Line 301: Sets the maximum record length to 144 characters for the edit mode (ie P4 = 144).

Lines 357–367: Will display a guidance chart for the operator to adhere to when editing record fields.

The remaining program lines are modified Subroutines 9 and 10, and the principles are the same as those in Chapter 2.

The setting up of the Customer/Supplier Balance file (from now on, CSB file) is the first stage in the change-over from the existing accounts system. It is necessary to allocate a unique number to each customer and supplier account — to establish a division between the two, customer accounts will be numbered from 1 to 500 and supplier accounts from 501 to 999. These numbers will be referred to in the following program and in the analysis program, but can be altered to suit an individual system: they are used here simply for convenience. Account number 500 could be allocated for sales to customers who do not have an account and 999 for miscellaneous expenses.

The word supplier does not really describe account numbers 501 to 999, as they relate to transactions concerned with money being paid out — not only to suppliers of goods, but as mortgage payments for the business premises, interest charges to the bank, etc. In fact, any situation where outgoing cash needs to be recorded (excluding customer account credits).

The financial state of each customer's account will also have to be established and, if the CSB file is not being created at the beginning of the financial year, the total credits paid to date and the total tax paid will have to be included, as well as the total outstanding amount of tax. The same applies for the outgoing cash account numbers 501 to 999.

If you commence the file at the beginning of a financial year, you would need to include the outstanding amounts and tax from the previous year only for account numbers 1 to 499. By using the edit facility from Subroutine 5, this file can then be altered to provide a new file for the forthcoming financial year. **Figure 3.2** shows what each record should contain when the CSB file is created at the beginning of a new financial year, taking into account that:

a. Customer A still owes £100 from the previous year.

b. Customer B cleared his account.

If you create the file after the start of the financial year, the records will appear as in **Figure 3.3**, taking into account that:

a. Customer A has still not paid his outstanding account and has not purchased further items.

Figure 3.2

	Field 7 ↓	Field 8 ↓	Field 9 ↓	Field 10 ↓
Customer A	0.00	0.00	100.00	15.00
Customer B	0.00	0.00	0.00	0.00
Accs. 501–999	0.00	0.00	0.00	0.00

Figure 3.3

	Field 7 ↓	Field 8 ↓	Field 9 ↓	Field 10 ↓
Customer A	0.00	0.00	100.00	15.00
Customer B	200.00	30.00	100.00	15.00
Supplier Acc. No. 501	100.00	15.00	1000.00	150.00

b. Customer B has purchased and paid for goods amounting to £200 and has still to pay for a further £100 worth of goods.

c. Supplier Acc.No. 501 has delivered £1000 worth of goods and has been paid. £100 worth were damaged and returned and a refund has been received.

The CSB file will be updated at the end of each month, using the information contained in the file produced by the next program, which will contain the month's transactions and maintain running totals of all credits, debits and tax.

It will be necessary to establish an automatic file-name change to avoid operator entry errors when, at the end of each month, the CSB file is updated and a new file created. This may not be of paramount importance with a well-organised tape data library, but two files of the same name cannot exist on the same data disc. A convenient method would be to use the month and the last two digits of the current year, eg MARCH 84, for file names. Lines 930–939 in Subroutines 8t and 8d will change MARCH 84 to APRIL 84 or DECEMBER 84 to JANUARY 85.

Lines 701–702 and 703–704 would have to be modified to notify the operator of the format of the file name required, eg

Enter the MONTH, followed by a space, followed by the year (eg MARCH 84).

Including an error-trapping routine would, of course, minimise spelling errors. An additional safety measure would be a check to ensure that a space separated the month and the year. To check for spelling mistakes, a subroutine such as the example below could easily be incorporated, and lines 950–951 would cater for the space separator.

```
940 DATA MARCH,APRIL,MAY,JUNE,JULY,AUGUST,
    SEPTEMBER,OCTOBER,NOVEMBER
941 DATA DECEMBER,JANUARY,FEBRUARY,"****T"
945 L% = LEN(D9$)
946 READ AD$:IFAD$ = "****T"THENGOSUB207
947 IFMID$(D9$,L% - 3) = AD$THENRESTORE:RETURN
948 GOTO946
950 L% = LEN(D9$):IFMID$(D9$,L% - 2,1) = CHR$(32)THEN
    RETURN
951 GOSUB207
```

Monthly Transaction

This second program is designed to carry out two main functions:

1) Produce a record file of all financial transactions for an account month, the Monthly Transaction file.
2) Update the stock list using the entered transaction information.

Transactions may be entered at any time during the month. The final file is used at the end of the month by the Analysis program (see section 3 of this chapter) to produce ledger sheets, etc. All entries relating to the sale or return of goods from stock will be referred to the Stock Control file (produced by the program in the next section) and the stock list amended accordingly each time the program is run. By carrying out this procedure, the number of items in store will always be up-to-date. I've called this program the Monthly Transaction program and the file produced the Monthly Transaction file. **Figure 3.4** shows each record in its final state as saved on tape or disc.

Figure 3.4

Field 1	Field 2	Field 3	Field 4	Field 5	Field 6
Code	Transaction Description	Credit	Cr. Tax	Debit	Dr. Tax
← 4 →	← 36 →	← 10 →	← 10 →	← 10 →	← 10 →

The program is composed of:

1) Subroutine 1
2) Subroutine 2
3) Subroutine 3a
4) Subroutine 4
5) Subroutine 6
6) Subroutine 7
7) Subroutine 8t *or* 8d
8) Subroutine 9

The necessary modifications to Subroutines 1 and 9 are listed below, and Subroutine 10 is completely modified as shown.

Amended Listing

```
1 REM ***MONTHLY TRANSACTION PROGRAM COM
MODORE 64***
5 W2$="[CLR][RVS]MONTHLY TRANSACTION PRO
GRAM                [HOME][CD]"
8 BL$="[SPC][SPC][SPC][SPC][SPC][SPC][SP
C][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SP
C][SPC][SPC][SPC][SPC][SPC]":REM ***20 U
PPER CASE SPACES***
9 BL$=BL$+BL$+BL$:BO$="000000"
10 DIMA8$(999),A9$(999)
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO START A NEW MONTHLY
 TRANSACTION      FILE."
1002 PRINT"[CD]2  TO ADD NEW DATA TO THE
 MONTHLY          TRANSACTION FILE."
1004 PRINT"[CD]3  FOR SCREEN DISPLAY OF
FILE CONTENTS."
1005 PRINT"[CD]4  FOR PRINT-OUT OF FILE.
"
1008 PRINT"[CD]5  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]6  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>6THEN1010
1012 PRINTW2$:ONGGOTO3000,3200,891,533,2
21,9999
```

```
1999 PRINTW2$:PRINT"DATA ENTRY NUMBER:-
 ";K:PRINTW1$:RETURN
2000 K=1:GOSUB1999:PRINT"ENTER CUSTOMER/
SUPPLIER"
2005 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC]                REFEREN
CE NUMBER.[CD]"
2010 PRINT"CODE 500 = NON ACCOUNT NUMBER
"
2015 PRINT"          TRANSACTIONS RELATING
 TO STOCK."
2020 PRINT"CODE 999 = MISCELLANEOUS TRAN
SACTIONS"
2025 PRINT"                                WITH
NO CODING.[CD]"
2030 PRINT"#  SIGN TO FINISH ADDDING DAT
A.[CD]"
2035 PRINT"FOLLOWED BY RETURN":PRINTW1$:
L9=3:GOSUB100
2040 IFD9$="#"THENRETURN
2045 B1$=D9$:L%=LEN(B1$):B1$=LEFT$(B0$,4
-L%)+B1$:K=K+1
2050 IFVAL(B1$)>501THEN2370
2055 GOSUB1999:PRINT"[CD]ENTER STOCK CON
TROL NUMBER.[CD]"
2060 PRINT"                          OR[CD]"
2065 PRINT"#  SIGN FOR A TRANSACTION NOT
          INVOLVING STOCK.[CD][CD]"
2070 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=3:GOSUB100:Z=1
2075 IFD9$="#"THEN2370
2080 IFVAL(D9$)=VAL(LEFT$(A9$(Z),4))THEN
2095
2085 IFA9$(Z)="****T"THENGOSUB207:GOTO20
00
2090 Z=Z+1:GOTO2080
2095 BB$=MID$(A9$(Z),26,5)
2100 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO2100
2105 IFVAL(BB$)>0THEN2130
2110 PRINTW2$:PRINT"[CD][CD]THE STOCK LE
VEL FOR THIS ITEM IS ZERO,[CD]"
2115 PRINT"THEREFORE, NO ENTRY CAN BE MA
DE UNTIL[CD]"
```

```
2120 PRINT"ADDITIONAL STOCK HAS BEEN REC
EIVED AND[CD]"
2125 PRINT"INCLUDED IN THE STOCK CONTROL
 FILE.[CD][CD]":PRINTW1$:GOSUB210:GOTO20
00
2130 F2$=D9$:L%=LEN(F2$):F2$=LEFT$(BO$,4
-L%)+F2$:K=K+1
2135 GOSUB1999:PRINT"[CD]ENTER TOTAL NUM
BER OF STOCK ITEMS.[CD]"
2140 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=4:GOSUB100
2145 IFVAL(D9$)>VAL(BB$)THEN2155
2150 GOTO2180
2155 D9$=BB$:PRINTW2$
2160 PRINT"[CD][CD]AS THE STOCK LEVEL IS
 ONLY :- ";BB$
2165 PRINT"[CD]THE REMAINING:- ";VAL(D9$
)-VAL(BB$);" ITEMS"
2170 PRINT"[CD]WILL HAVE TO BE RE-ENTERE
D WHEN"
2175 PRINT"[CD]SUFFICIENT STOCK IS AVAIL
ABLE.[CD][CD]":PRINTW1$:GOSUB210
2180 F3$=D9$:L%=LEN(F3$):F3$=LEFT$(BO$,5
-L%)+F3$:K=K+1
2185 PRINTW2$:PRINT"[CD][CD]ARE THE ITEM
S BEING ISSUED FROM STOCK[CD]"
2190 PRINT"AND HAVE THEREFORE TO BE CHAR
GED?[CD]"
2195 GOSUB202:F4$="N":IFY$="Y"THENF4$="Y
"
2200 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]"
2205 PRINT"[CD]CUST/SUP. CODE :- ";B1$
2210 PRINT"[CD]STOCK NUMBER   :- ";F2$
2215 PRINT"[CD]TOTAL NUMBER   :- ";F3$
2220 IFF4$="N"THEN2230
2225 PRINT"[CD]CUSTOMER DEBIT":GOTO2235
2230 PRINT"[CD]CUSTOMER CREDIT"
2235 PRINTW1$:GOSUB200
2240 IFY$="N"THENB1$="":F2$="":F3$="":F4
$="":GOTO2000
2245 REM ***BUILD UP OF MONTHLY TRANSACT
ION RECORD USING STOCK RECORD***
2250 B2$=STR$(VAL(F3$))+"[SPC]"+"X"+"[SP
```

```
C]"+MID$(A9$(Z),5,16)
2255 L%=LEN(B2$):B2$=B2$+LEFT$(BL$,25-L%
):B2$=B2$+"ST.NO. "+F2$
2260 B8$=MID$(A9$(Z),61,10):B9$=MID$(A9$
(Z),71,10)
2265 IFLEFT$(B8$,1)=CHR$(160)THENB8$=MID
$(B8$,2):GOTO2265
2270 IFLEFT$(B9$,1)=CHR$(160)THENB9$=MID
$(B9$,2):GOTO2270
2275 R=VAL(B8$):GOSUB600:R=R*VAL(F3$):C1
=R:R=0:GOSUB619:B8$=C3$
2280 R=VAL(B9$):GOSUB600:R=R*VAL(F3$):C1
=R:R=0:GOSUB619:B9$=C3$
2285 A8$=B1$+B2$:IFF4$="N"THEN2295
2290 A8$=A8$+B8$+B9$+LEFT$(BL$,20):GOTO2
305:REM **ACCOUNT HOLDER DEBIT**
2291 IFVAL(LEFT$(A8$,4))<500THENA8$=A8$+
"*":GOTO2305
2295 A8$=A8$+LEFT$(BL$,20)+B8$+B9$:REM *
*ACCOUNT HOLDER CREDIT**
2296 IFVAL(LEFT$(A8$,4)<500THENA8$=A8$+"
*"
2300 REM ***UPDATE STOCK CONTROL FILE RE
CORD***
2305 B4$=MID$(A9$(Z),26,5)
2310 IFLEFT$(B4$,1)=CHR$(160)THENB4$=MID
$(B4$,2):GOTO2310
2315 B5$=MID$(A9$(Z),31,10)
2320 IFLEFT$(B5$,1)=CHR$(160)THENB5$=MID
$(B5$,2):GOTO2320
2325 IFF4$="Y"THEN2340
2330 B4$=STR$(VAL(B4$)+VAL(F3$)):REM **R
ETURN TO STOCK**
2335 B5$=STR$(VAL(B5$)-VAL(F3$)):GOTO235
0:REM **SUBTRACT FROM TOTAL STOCK SOLD**
2340 B4$=STR$(VAL(B4$)-VAL(F3$)):REM **S
UBTRACT FROM STOCK**
2345 B5$=STR$(VAL(B5$)+VAL(F3$)):REM **A
DD TO TOTAL STOCK SOLD**
2350 L%=LEN(B4$):B4$=LEFT$(BL$,5-L%)+B4$
2355 L%=LEN(B5$):B5$=LEFT$(BL$,10-L%)+B5
$
2360 A9$(Z)=LEFT$(A9$(Z),25)+B4$+B5$+MID
$(A9$(Z),41):Z=1
```

```
2365 F2$="":F3$="":F4$="":B1$="":B4$="":
B5$="":B8$="":B9$="":RETURN
2370 GOSUB1999:PRINT"[CD]TYPE IN DESCRIP
TION OF TRANSACTION.[CD]"
2375 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=35:GOSUB150
2380 B2$=D9$:L%=LEN(B2$):B2$=B2$+LEFT$(B
L$,36-L%):K=K+1
2385 GOSUB1999:PRINT"[CD]ENTER CREDIT AM
OUNT.[CD]"
2390 PRINT"                          OR[CD]"
2395 PRINT"#  SIGN OR 0 IF A DEBIT TRANS
ACTION.[CD][CD]"
2400 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2405 IFD9$="#"ORD9$="0"THENB3$=LEFT$(BL$
,10):B4$=LEFT$(BL$,10):GOTO2450
2410 N1$=D9$:GOSUB607:B3$=N1$:L%=LEN(B3$
):B3$=LEFT$(BL$,10-L%)+B3$:K=K+1
2415 GOSUB1999:PRINT"[CD]TYPE IN CREDIT
TAX.[CD]"
2420 PRINT"                          OR[CD]"
2425 PRINT"#  SIGN OR 0 IF GOODS ARE ZER
O RATED.[CD][CD]"
2430 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2435 IFD9$="#"ORD9$="0"THENB4$="[SPC][SP
C][SPC][SPC][SPC][SPC]0.00":GOTO2445
2440 N1$=D9$:GOSUB607:B8$=N1$:L%=LEN(B8$
):B8$=LEFT$(BL$,10-L%)+B8$:K=K+1
2445 B5$=LEFT$(BL$,10):B6$=LEFT$(BL$,10)
:GOTO2500
2450 GOSUB1999:PRINT"[CD]TYPE IN DEBIT A
MOUNT.[CD]"
2455 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2460 IFVAL(D9$)<.01THEN2450
2465 N1$=D9$:GOSUB607:B5$=N1$:L%=LEN(B5$
):B5$=LEFT$(BL$,10-L%)+B5$:K=K+1
2470 GOSUB1999:PRINT"[CD]TYPE IN DEBIT T
AX.[CD]"
2475 PRINT"                          OR[CD]"
2480 PRINT"#  SIGN OR 0 IF ZERO RATED.[C
D][CD]"
```

```
2485 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2490 IFD9$="#"ORD9$="0"THENB6$="[SPC][SP
C][SPC][SPC][SPC][SPC]0.00":GOTO2500
2495 N1$=D9$:GOSUB607:B6$=N1$:L%=LEN(B6$
):B6$=LEFT$(BL$,10-L%)+B6$
2500 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]"
2505 PRINT"[CD]CUST/SUP. CODE:- ";B1$
2510 PRINT"[CD]TRANSACTION   :- ";B2$
2515 PRINT"[CD]CREDIT        :- \";B3$
2520 PRINT"[CD]CREDIT TAX    :- \";B4$
2525 PRINT"[CD]DEBIT         :- \";B5$
2530 PRINT"[CD]DEBIT TAX     :- \";B6$
2535 PRINTW1$:GOSUB200
2540 IFY$="N"THEN2555
2545 A8$=B1$+B2$+B3$+B4$+B5$+B6$
2546 PRINTW2$:PRINT"[CD]IS THIS A CASH T
RANSACTION[CD]":PRINTW1$:GOSUB202
2547 IFY$="N"THENA8$=A8$+"*"
2550 B1$="":B2$="":B3$="":B4$="":B5$="":
B6$="":RETURN
2555 B1$="":B2$="":B3$="":B4$="":B5$="":
B6$="":GOTO2000
2995 REM ***START A NEW FILE***
3000 PRINTW2$:PRINT"[CD][CD]PLACE STOCK
CONTOL TAPE FILE[CD]"
3005 PRINT"                          IN THE
RECORDER.[CD][CD]":PRINTW1$:GOSUB210
3010 Z=1:F9$="STOCK CONTROL":GOSUB733:IF
ER=1THENER=0:GOTO3010
3015 GOSUB783:A9$(Z)=A9$:IFA9$="****T"TH
ENA9$="":GOSUB870:GOSUB901:GOTO3025
3020 Z=Z+1:GOTO3015
3025 Z=1:GOSUB823:GOSUB701:F8$=D9$:D9$="
":GOSUB710:IFER=1THENER=0:GOTO3025
3030 GOSUB2000:IFD9$="#"THEN3040
3035 GOSUB792:A8$="":GOTO3030
3040 GOSUB218:IFY$="N"THEND9$="":GOTO303
0
3045 A8$="****T":GOSUB792:GOSUB869:GOSUB
901:IFAE$="Y"THENGOSUB879
3050 PRINTW2$:PRINT"[CD][CD]UPDATED STOC
K CONTROL FILE WILL NOW"
```

```
3055 PRINT"[CD]BE STORED ON TAPE/DISC[CD
]"
3060 PRINT"PLACE TAPE IN RECORDER TO REC
EIVE[CD]"
3065 PRINT"UPDATED STOCK CONTROL FILE.[C
D]":GOSUB825
3070 FF$="SCTEMP":FL$=F9$:GOSUB872:F2$=F
8$:F8$=F9$:Z=1
3075 GOSUB710:IFER=1THENER=0:GOTO3075
3080 A8$=A9$(Z):GOSUB792:IFA8$="****T"TH
ENGOSUB869:A8$="":GOSUB901:GOTO3090
3085 Z=Z+1:GOTO3080
3090 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE A COPY OF YOUR DATA?[CD][CD]":PRINTW1
$:GOSUB202
3095 IFY$="N"THEN3105
3100 Z=1:GOTO3050
3105 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE A COPY OF THE MONTHLY[CD]"
3110 PRINT"TRANSACTION FILE?[CD][CD]":PR
INTW1$:GOSUB202
3115 IFY$="N"THENRUN3
3120 F8$=F2$:GOSUB823:GOSUB730:I=1:IFER=
1THENER=0:GOTO3120
3125 GOSUB781:A8$(I)=A8$:A9$(I)="":A8$="
"
3130 IFA8$(I)="****T"THENGOSUB869:GOSUB9
01:GOSUB843:RUN3
3135 I=I+1:GOTO3125
3195 REM ***ADD NEW DATA TO FILE***
3200 PRINTW2$:PRINT"[CD][CD]PLACE STOCK
CONTOL TAPE FILE[CD]"
3205 PRINT"                              IN THE
RECORDER.[CD][CD]":PRINTW1$:GOSUB210
3210 Z=1:F9$="STOCK CONTROL":GOSUB733:IF
ER=1THENER=0:GOTO3210
3215 GOSUB783:A9$(Z)=A9$:IFA9$="****T"TH
ENA9$="":GOSUB870:GOSUB901:Z=1:GOTO3225
3220 Z=Z+1:GOTO3215
3225 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO3225
3230 GOSUB781:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO3240
3235 I=I+1:GOTO3230
```

```
3240 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:AE$="Y"
3245 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO3245
3250 IFA8$(I)="****T"THENGOSUB210:GOTO32
60
3255 A8$=A8$(I):GOSUB792:A8$="":A8$(I)="
":I=I+1:GOTO3250
3260 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO ADD MORE INFORMATION?[CD][CD]":PRINTW
1$
3265 GOSUB202:IFY$="Y"THENA8$(I)="":GOTO
3030
3270 A8$=A8$(I):GOSUB792:GOSUB869:GOSUB9
01:GOSUB879:GOSUB840:RUN3
```

Commentary

Lines 1–1012: These are self-explanatory. The Edit routine (Subroutine 5) has been omitted as a great deal more programming would be necessary to cope with changes relating to the Stock Control file.

Lines 1999–2555: Concerned with the creation of a Monthly Transaction record and made up of two parts.

Part 1 (lines 1999–2075 and lines 2370–2555) relates to the creation of a record of all transactions not involved with stock items. Allowances are made as to whether the transaction is credit or debit, as well as to whether or not 'purchase' tax is applicable. The code 999 would be used for miscellaneous transactions — not suppliers or other designated people or companies with whom regular dealings are maintained, who would have an allocated code in the CSB file.

Part 2 (lines 1999–2365) is also concerned with the creation of a Monthly Transaction record, but only of those transactions directly related to the Stock Control file. The Stock Control file is loaded into the computer (lines 3000–3020 and 3200–3220) and the information contained in these records can be used to save on operator entry time.

Instead of having to enter a description of each transaction, you only need to enter the stock control code number and the total number of stock items. A simple yes/no answer (lines 2185–2195) is all that is necessary for the financial part of the transaction. Controls are incorporated to avoid the entry of stock items which are not available (lines 2095–2125 and lines 2145–2180).

To explain this rather irrational statement, the Stock Control file requires updating when the stock level reaches nil and new stock is

received. To allow for new stock price alterations, the Stock Control program will not allow new items to be included unless all the existing items have been sold. It is therefore conceivable that a customer could be given the last seven items of the old stock and three items of new stock which have a different price from the old. The person in charge of store issues is unlikely to differentiate between old and new stock and would simply inform the computer operator, that, for example, customer 1 had received 10 items of stock number 1.

Line 2010: Permits the sale of stock items to customers who do not have an account number.

Lines 2080–2090: Serve as an operator entry check. If the entered stock number does not correspond with a stock number held on file, then an error message is displayed.

Line 2100: Requires an explanation as the sequence has not been dealt with previously.

You can convert a data string containing a number to a numeric value by using the expression VAL. It is not possible to convert directly a data string number which is preceded by upper case spaces, CHR$(160), and these spaces must be removed first.

BB = VAL([SPC][SPC]65)

would allocate the value zero to BB.

By removing the two upper case spaces, as achieved in line 2100, the variable BB would equal 65.

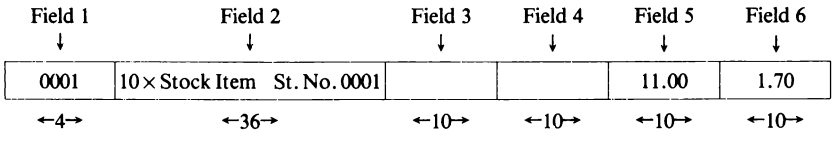
BB = VAL(65[SPC][SPC])

is acceptable and BB would equal 65.

Lines 2200–2240: Allow the operator to check the entries for accuracy (as do lines 2500–2555).

Lines 2245–2295: Convert the information entered (stock item information only) into the correct format for a Monthly Transaction record, A8$: eg information entered = Customer Account Number 1, Stock Item 1, total number of stock items involved (10), whether the items were credited or debited to the customer. The Monthly Transaction record produced from this information would read as in **Figure 3.5**, where the selling price of each item is £1.10 and the 'sales' tax is £0.17.

Figure 3.5



Lines 2300–2365: Alter the Stock Control record, A9$(Z), adding to or deducting from the stock level and also the total number of stock items sold. Assuming that the Stock Control record, A9$(Z), is composed of the information fields in **Figure 3.6**, the first and second rows show the sequence from the previous lines.

Figure 3.6

No.	Description	Min.	Stock	Sold	Pur.Price	Pur.Tax	Sale Price	Sale Tax
0001	Stock Item 1	10	35	15	1.00	0.15	1.10	0.17
0001	Stock Item 1	10	25	25	1.00	0.15	1.10	0.17

On completion of the transaction entries, the modified Stock Control records are saved on tape/disc and form a new updated Stock Control file.

These lines will be more easily understood after the next section.

Lines 3000–3135 and **lines 3195–3270:** Control the creation of a new Monthly Transaction file, and the addition of further information to an existing file, respectively. They are simply modified versions of lines 2500–2645 contained in the standard Subroutine 10.

2. Stock Control

The objectives of the listed Stock Control program are as follows:

1) To produce and add to a Stock Control file with an edit facility.
2) To incorporate new stock received into the Stock Control file.
3) To evaluate stock levels and indicate which items have to be ordered when levels fall below a set minimum.
4) To allow for two types of record printout: one which includes the stock

item number (the item description with stock level, to be used for stock-taking purposes) and the other a printout of the complete contents of each file record.

The bases of the program are:

1) Subroutine 1
2) Subroutine 2
3) Subroutine 3a
4) Subroutine 4
5) Subroutine 5
6) Subroutine 6
7) Subroutine 7
8) Subroutine 8t *or* 8d
9) Subroutine 9

A completely modified Subroutine 10 is included.

Amended listing

```
1 REM ***STOCK CONTROL PROGRAM. COMMODOR
E 64***
5 W2$="[CLR][RVS]STOCK CONTROL PROGRAM.
                    [HOME][CD]"
9 BL$=BL$+BL$+BL$:BO$="000000"
10 DIMA8$(500),A9$(500)
301 I=1:P4=80:REM ***MAXIMUM STRING LENG
TH***
357 W5$="[RVS]NO. STOCK ITEM       MIN. T
.NO.T.SOLD    PURCHASE\ TAX \      [RVO]"
358 W6$="[RVS]SALE \     SALE TAX \[RVO]"
360 W5$=W5$+W6$
367 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD]";W5$:PRINT
533 GOSUB551:GOSUB823:GOSUB703:F8$=D9$:D
9$=""
536 OPEN4,4,7:IFTR=1THEN538:REM ***OPEN4
,4,7 FOR UP/LO CHAR.ON 1515 PRINTER***
542 IFI=>TTTHENGOSUB210:GOTO549
543 GOSUB781:A$=A8$:IFA8$="****T"THENA$=
"":GOSUB210:GOTO546
544 IFSL$="1"THENA$=LEFT$(A8$,4)+"[SPC][
SPC]     "+MID$(A8$,5,16)+"[SPC][SPC]"+M
```

```
ID$(A8$,26,5)
545 GOSUB470:GOTO542
546 GOSUB869:GOSUB901:IFP4$="N"THENCLOSE
4
547 GOSUB210:GOSUB226:IFY$="Y"THEN534
548 RUN3
549 IFP4$="N"THENCLOSE4
550 GOTO535
551 PRINTW2$:PRINT"[CD][CD][RVS]1[RVO] F
OR PRINT-OUT OF STOCK LIST":PRINT"  LESS
 PRICES.[CD][CD]"
552 PRINT"[CD][CD][RVS]2[RVO] FOR PRINT-
OUT OF STOCK LIST":PRINT"  WITH PRICES.[
CD][CD]":PRINTW1$
553 L9=1:GOSUB100:IFVAL(D9$)>2THEND9$=""
:GOTO551
554 SL$=D9$:IFSL$="1"THEN556
555 GOTO559
556 GOSUB502:OPEN4,4,7
557 A$="STOCK NO.  DESCRIPTION.      IN S
TOCK.":PRINT#4,A$:A$="[SPC]"
558 PRINT#4,A$:A$="":CLOSE4:RETURN
559 GOSUB502:OPEN4,4,7
560 A$="NO. DESCRIPTION       MIN. STOCK
   SOLD. PUR.PRICE    P.TAX."
561 A$=A$+"SALE PRICE    S.TAX.":PRINT#4
,A$:A$="[SPC]"
562 PRINT#4,A$:A$="":CLOSE4:RETURN
702 D9$="STOCK CONTROL":RETURN
704 D9$="STOCK CONTROL":RETURN
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO START A STOCK CONTR
OL FILE."
1002 PRINT"[CD]2  TO ADD NEW DATA TO STO
CK FILE."
1003 PRINT"[CD]3  TO EDIT STOCK CONTROL
FILE."
1004 PRINT"[CD]4  FOR SCREEN DISPLAY OF
STOCK FILE."
1005 PRINT"[CD]5  TO PRINT STOCK FILE."
1006 PRINT"[CD]6  TO INCLUDE PURCHASED S
TOCK TO FILE."
1007 PRINT"[CD]7  TO EVALUATE STOCK BELO
```

```
W MINIMUM LEVEL"
1009 PRINT"[CD]8  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>8THEN1010
1012 PRINTW2$:ONGGOTO2500,2600,300,897,5
33,3600,4000,9999
1999 PRINTW2$:PRINT"DATA ENTRY NUMBER:-
 ";K:PRINTW1$:RETURN
2000 K=1:GOSUB1999:PRINT"[CD]ENTER STOCK
 ITEM NUMBER.[CD]"
2005 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2010 PRINT"#  SIGN TO FINISH ADDDING DAT
A.[CD][CD]"
2015 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=3:GOSUB100
2020 IFD9$="#"THEND9$="****T":RETURN
2025 B1$=D9$:L%=LEN(B1$):B1$=LEFT$(B0$,4
-L%)+B1$:K=K+1
2030 GOSUB1999:PRINT"[CD]TYPE IN STOCK D
ESCRIPTION.[CD]"
2045 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=15:GOSUB150
2055 B2$=D9$:L%=LEN(B2$):B2$=B2$+LEFT$(B
L$,16-L%):K=K+1
2060 GOSUB1999:PRINT"[CD]TYPE IN MINIMUM
 LEVEL OF STOCK ALLOWED.[CD]"
2065 PRINT"BEFORE RE-ORDERING.[CD]"
2075 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=4:GOSUB100
2085 B3$=D9$:L%=LEN(B3$):B3$=LEFT$(BL$,5
-L%)+B3$:K=K+1
2090 GOSUB1999:PRINT"[CD]ENTER THE TOTAL
 NUMBER OF STOCK ITEMS[CD]"
2095 PRINT"HELD IN STORE.[CD]"
2105 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=4:GOSUB100
2115 B4$=D9$:L%=LEN(B4$):B4$=LEFT$(BL$,5
-L%)+B4$:K=K+1
2120 GOSUB1999:PRINT"[CD]ENTER TOTAL NUM
BER SOLD TO DATE.[CD]"
2125 PRINT"                    OR[CD]"
```

```
2130 PRINT"#  SIGN FOR NONE SOLD.[CD][CD
]"
2135 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2140 IFD9$="#"THENB5$="[SPC][SPC][SPC][S
PC][SPC][SPC][SPC][SPC][SPC]0":K=K+1:GOT
O2150
2145 B5$=D9$:L%=LEN(B5$):B5$=LEFT$(BL$,1
0-L%)+B5$:K=K+1
2150 GOSUB1999:PRINT"[CD]ENTER PURCHASE
PRICE PER ITEM OR UNIT[CD]"
2155 PRINT"LESS PURCHASE TAX[CD][CD]"
2165 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100:N1$=D9$:GOSUB607
2175 B6$=N1$:L%=LEN(B6$):B6$=LEFT$(BL$,1
0-L%)+B6$:K=K+1
2180 GOSUB1999:PRINT"[CD]ENTER PURCHASE[
SPC]TAX PER ITEM OR UNIT.[CD][CD]"
2185 PRINT"                    OR[CD]"
2190 PRINT"#  SIGN FOR ZERO TAX RATE.[CD
][CD]"
2195 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
2200 IFD9$="#"ORD9$="0"THENB7$=LEFT$(BL$
,6)+"0.00":K=K+1:GOTO2210
2205 N1$=D9$:GOSUB607:B7$=N1$:L%=LEN(B7$
):B7$=LEFT$(BL$,10-L%)+B7$:K=K+1
2210 GOSUB1999:PRINT"[CD]ENTER SALE PRIC
E PER ITEM OR UNIT[CD]"
2211 PRINT"LESS SALE TAX[CD][CD]"
2215 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100:N1$=D9$:GOSUB607
2220 B8$=N1$:L%=LEN(B8$):B8$=LEFT$(BL$,1
0-L%)+B8$:K=K+1:BB$=B7$
2221 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO2221
2222 IFVAL(BB$)=0THENB9$=LEFT$(BL$,6)+"0
.00":GOTO2240
2225 GOSUB1999:PRINT"[CD]ENTER SALE TAX
PER ITEM OR UNIT.[CD][CD]"
2230 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100:N1$=D9$:GOSUB607
2235 B9$=N1$:L%=LEN(B9$):B9$=LEFT$(BL$,1
0-L%)+B9$
```

```
2240 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]":PRINTB1$:PRINTB2$:PRINTB3
$
2245 PRINTB4$:PRINTB5$:PRINTB6$:PRINTB7$
:PRINTB8$:PRINTB9$:PRINTW1$:GOSUB200
2250 IFY$="N"THEN2490
2450 A8$(I)=B1$+B2$+B3$+B4$+B5$+B6$+B7$+
B8$+B9$
2460 RETURN
2490 B1$="":B2$="":B3$="":B4$="":B5$="":
B6$="":B7$="":B8$="":B9$="":GOTO2000
2499 REM ***START A NEW FILE***
2500 I=1:GOSUB823:GOSUB701:F8$=D9$:D9$="
":GOSUB710:IFER=1THENER=0:GOTO2500
2505 GOSUB2000:IFD9$="****T"THEN2525
2510 A8$=A8$(I):GOSUB792:B1$="":B2$="":B
3$="":B4$="":B5$="":B6$=""
2515 B7$="":B8$="":B9$="":GOTO2505
2525 GOSUB218:IFY$="N"THEND9$="":GOTO250
5
2530 A8$(I)="****T":A8$=A8$(I):GOSUB792:
GOSUB869:GOSUB901:IFAE$="Y"THENGOSUB879
2535 GOSUB840:RUN3
2599 REM ***ADD NEW DATA TO FILE***
2600 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO2600
2605 GOSUB781:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO2615
2610 I=I+1:GOTO2605
2615 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:AE$="Y"
2620 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO2620
2625 IFA8$(I)="****T"THENGOSUB210:GOTO26
35
2630 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO2625
2635 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO ADD MORE INFORMATION?[CD][CD]":PRINTW
1$
2640 GOSUB202:IFY$="Y"THENA8$(I)="":GOTO
2505
2645 GOTO2530:REM **CLOSE FILE**
2999 REM ***INCLUDE PURCHASED STOCK TO E
```

```
XISTING STOCK RECORDS***
3000 PRINTW2$:I=1:PRINT"[CD]ENTER STOCK
ITEM NUMBER.[CD]"
3005 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
3010 PRINT"#  SIGN TO FINISH ADDDING DAT
A.[CD][CD]"
3015 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=3:GOSUB100
3020 IFD9$="#"THENRETURN
3025 IFVAL(D9$)=VAL(LEFT$(A8$(I),4))THEN
3050
3030 IFA8$(I)="****T"THEN3040
3035 I=I+1:GOTO3025
3040 PRINTW2$:PRINT"[CD][CD]NO STOCK ITE
M WITH NUMBER ";D9$
3045 PRINT"[CD]IS ON FILE![CD][CD]":PRIN
TW1$:GOSUB210:GOTO3000
3050 B1$=D9$:L%=LEN(B1$):B1$=LEFT$(B0$,4
-L%)+B1$
3055 PRINTW2$:PRINT"[CD]ENTER PURCHASE P
RICE PER ITEM OR UNIT[CD]"
3060 PRINT"LESS PURCHASE TAX[CD][CD]"
3065 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
3070 REM **CHECK TO COMPARE NEW PURCHASE
 PRICE WITH OLD***
3075 BB$=MID$(A8$(I),41,10)
3080 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO3080
3085 IFVAL(D9$)=VAL(BB$)THENBB$="":GOTO3
215
3090 REM **CHECK TO ESTABLISH THAT PRESE
NT STOCK IS EXHAUSTED**
3095 BB$="":BB$=MID$(A8$(I),26,5)
3100 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO3100
3105 IFVAL(BB$)=0THENBB$="":GOTO3130
3110 BB$="":PRINTW2$:PRINT"[CD][CD]PLEAS
E ADD THESE ITEMS WHEN THE PRESENT"
3115 PRINT"STOCK HAS REACHED ZERO.  THE
EDIT MODE"
3120 PRINT"WILL ALLOW FOR THIS ADDITION
```

```
IF REQUIRED.[CD][CD]":PRINTW1$
3125 GOSUB210:GOTO3000
3130 N1$=D9$:GOSUB607:B6$=N1$:L%=LEN(B6$
):B6$=LEFT$(BL$,10-L%)+B6$
3135 PRINTW2$:PRINT"[CD]ENTER PURCHASE[S
PC]TAX PER ITEM OR UNIT.[CD][CD]"
3140 PRINT"                          OR[CD]"
3145 PRINT"#  SIGN FOR ZERO TAX RATE.[CD
][CD]"
3150 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100
3155 IFD9$="#"ORD9$="0"THENB7$=LEFT$(BL$
,6)+"0.00":GOTO2210
3160 N1$=D9$:GOSUB607:B7$=N1$:L%=LEN(B7$
):B7$=LEFT$(BL$,10-L%)+B7$:K=K+1
3165 PRINTW2$:PRINT"[CD]ENTER SALE PRICE
 PER ITEM OR UNIT[CD]"
3170 PRINT"LESS SALE TAX[CD][CD]"
3175 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100:N1$=D9$:GOSUB607
3180 B8$=N1$:L%=LEN(B8$):B8$=LEFT$(BL$,1
0-L%)+B8$
3185 BB$=B7$
3190 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO3190
3195 IFVAL(BB$)=0THENB9$=LEFT$(BL$,6)+"0
.00":BB$="":GOTO3215
3200 BB$="":PRINTW2$:PRINT"[CD]ENTER SAL
E TAX PER ITEM OR UNIT.[CD][CD]"
3205 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB100:N1$=D9$:GOSUB607
3210 B9$=N1$:L%=LEN(B9$):B9$=LEFT$(BL$,1
0-L%)+B9$
3215 PRINTW2$:PRINT"[CD]ENTER THE TOTAL
NUMBER OF ITEMS[CD]"
3220 PRINT"TO BE ADDED TO STOCK.[CD]"
3225 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=4:GOSUB100
3230 B4$=D9$:L%=LEN(B4$):B4$=LEFT$(BL$,5
-L%)+B4$:B4=VAL(D9$)
3235 BB$=MID$(A8$(I),26,5)
3240 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO3240
3245 IFVAL(BB$)=0THENBB$="":GOTO3295
```

```
3250 BB$="":PRINTW2$:PRINT"[CD]PLEASE CH
ECK YOUR ENTRIES[CD]":PRINTW1$
3255 PRINT"[CD][CD]STOCK ITEM:-  ";B1$:P
RINT"[CD]ADDITION:-     ";B4$:PRINTW1$:GO
SUB200
3260 IFY$="N"THENB1$="":B4$="":GOTO3000
3265 BB$=MID$(A8$(I),26,5)
3270 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO3270
3275 B4=B4+VAL(BB$):B4$=STR$(B4):BB$="":
B4=0
3280 L%=LEN(B4$):IFL%=5THEN3290
3285 B4$=LEFT$(BL$,5-L%)+B4$
3290 A8$(I)=LEFT$(A8$(I),25)+B4$+MID$(A8
$(I),31):B1$="":B4$="":GOTO3000
3295 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]":PRINTW1$
3300 PRINT"[CD]STOCK ITEM     :- ";B1$:PR
INT"[CD]ADDITION       :- ";B4$
3305 PRINT"[CD]PURCHASE PRICE:- ";B6$:PR
INT"[CD]PURCHASE TAX  :- ";B7$
3310 PRINT"[CD]SALE PRICE     :- ";B8$:PR
INT"[CD]SALE TAX       :- ";B9$
3315 PRINTW1$:GOSUB200
3320 IFY$="N"THENB1$="":B4$="":B6$="":B7
$="":B8$="":B9$="":GOTO3000
3325 A8$(I)=LEFT$(A8$(I),25)+B4$+MID$(A8
$(I),31,10)+B6$+B7$+B8$+B9$
3330 B1$="":B4$="":B6$="":B7$="":B8$="":
B9$="":GOTO3000
3600 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO3600
3605 GOSUB781:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO3615
3610 I=I+1:GOTO3605
3615 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:GOSUB3000
3620 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO3620
3625 IFA8$(I)="****T"THENA8$=A8$(I):GOSU
B792:GOTO3635
3630 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO3625
3635 GOSUB869:GOSUB901:GOSUB879:GOSUB840
```

```
:RUN3
3999 REM ***EVALUATE STOCK BELOW MINIMUM
 LEVEL FOR RE-ORDERING***
4000 Z=1:GOSUB823:GOSUB703:F8$=D9$:D9$="
"
4005 TR=0:GOSUB730:IFER=1THENER=0:GOTO40
00
4010 GOSUB781:IFA8$="****T"THENA8$(Z)=A8
$:GOTO4045
4015 B1$=MID$(A8$,26,5):B2$=MID$(A8$,21,
5)
4020 IFLEFT$(B1$,1)=CHR$(160)THENB1$=MID
$(B1$,2):GOTO4020
4025 IFLEFT$(B2$,1)=CHR$(160)THENB2$=MID
$(B2$,2):GOTO4025
4030 IFVAL(B1$)<VAL(B2$)THEN4040
4035 A8$="":GOTO4010
4040 A8$(Z)=A8$:Z=Z+1:GOTO4010
4045 GOSUB869:GOSUB901:Z=1:IFA8$(Z)="***
*T"THEN4055
4050 GOTO4065
4055 PRINTW2$:PRINT"[CD][CD]ALL STOCK AR
E EQUAL TO OR ABOVE THE[CD]"
4060 PRINT"MINIMUM LEVEL.[CD][CD]":PRINT
W1$:GOSUB210:RUN3
4065 GOSUB516:GOSUB512:IFP4$="Y"THEN4085
4070 OPEN4,4,7:IFTR=1THEN4080
4075 GOSUB450:GOSUB457:TR=1
4080 GOSUB506
4085 I=0:GOSUB216:IFY$="Y"THEN4100
4090 IFP4$="N"THENCLOSE4
4095 RUN3
4100 IFA8$(Z)="****T"THEN4115
4105 IFI=>TTTHENGOSUB210:GOTO4130
4110 A$=LEFT$(A8$(Z),30):GOSUB470:Z=Z+1:
GOTO4100
4115 IFP4$="N"THENCLOSE4
4120 GOSUB210:GOSUB226:IFY$="Y"THENZ=1:T
R=0:GOTO4065
4125 RUN3
4130 IFP4$="N"THENCLOSE4
4135 GOTO4065
```

Commentary

Lines 533–562: A modification of the standard file record printout sequence in Subroutine 6; can produce two different types of printout. **Table 3.1** (a sample of a tape/disc file) is an example of each record in its entirety and the sample in **Table 3.2** can be chosen and used for stock-taking.

Table 3.1

No.	Description	Min.	Stock	Sold	Pur. Price	Pur. Tax	Sale Price	S. Tax
0001	Stock Item 1	10	35	15	1.00	0.15	1.10	0.17
0002	Stock Item 2	10	50	0	2.00	0.30	2.20	0.33
0003	Stock Item 3	5	10	5	5.00	0.75	5.50	0.83
0004	Stock Item 4	5	4	6	10.00	1.50	11.00	1.65
0005	Stock Item 5	10	40	10	2.00	0.00	2.20	0.00
0006	Stock Item 6	10	0	50	1.00	0.15	1.10	0.17

Table 3.2

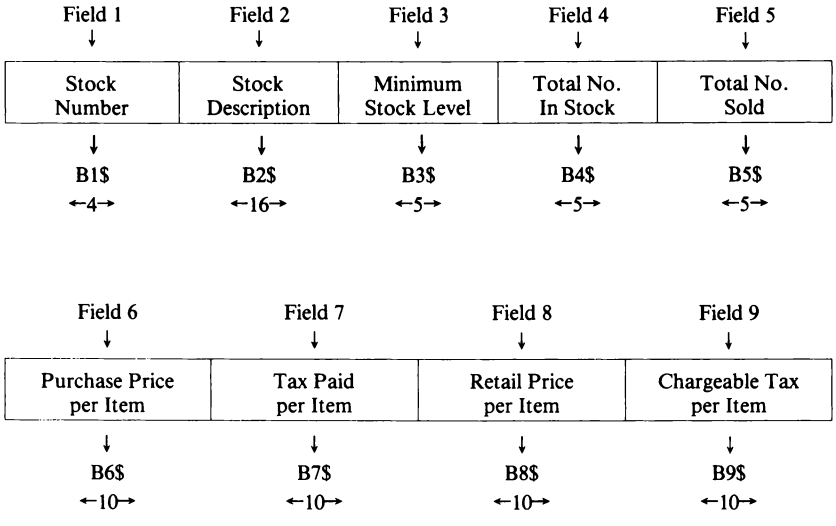
Stock No.	Description	In Stock
0001	Stock Item 1	35
0002	Stock Item 2	50
0003	Stock Item 3	10
0004	Stock Item 4	4
0005	Stock Item 5	40
0006	Stock Item 6	0

Lines 2000–2645: Cater for objective number 1 and will produce file records, each of which are composed of the fields in **Figure 3.7.**

Lines 3000–3635: Allow the stock file to be modified to include stock items which have been ordered and received. Checks are included to compensate for operator entry error, eg lines 3025–3045 establish that a stock item does exist in the file by comparing stock item numbers against the number entered.

To allow for increases or alterations in the price of the new stock, and to prevent old stock being charged at the new price, a check is placed in lines

Figure 3.7



3090–3125. New stock is not included until the old stock has been cleared and the level is zero. This is a limiting factor and the program could be modified to include a question as to whether the new stock differs in price or not. If not, then there would be no reason for the new stock not to be included. Another alternative would be to allow the new stock number additions along with the new prices and tag the necessary five extra fields on to the end of the record. By record string manipulation, these additional fields could be incorporated when the existing stock falls to zero. This, however, would necessitate the use of the GET # statement on data recall from tape or disc, resulting in a speed reduction in program operation. Each record could be split into two, recalled using INPUT # and combined to form one record for analysis.

Table 3.3

No.	Description	Min.	Stock	Sold	Pur. Price	Pur. Tax	Sale Price	S. Tax
0001	Stock Item 1	10	25	25	1.00	0.15	1.10	0.17
0002	Stock Item 2	10	50	0	2.00	0.30	2.20	0.33
0003	Stock Item 3	5	10	5	5.00	0.75	5.50	0.83
0004	Stock Item 4	5	4	6	10.00	1.50	11.00	1.65
0005	Stock Item 5	10	45	5	2.00	0.00	2.20	0.00
0006	Stock Item 6	10	20	50	2.00	0.30	2.20	0.33

Table 3.3 shows the changes resulting from the addition of 20 new items to stock item 6. Stock item 1 shows the results of ten items having been sold and stock item 5 the changes from having five items returned. These latter two alterations to the records would be brought about by the Monthly Transaction program in the first section of this chapter.

Lines 4000–4135: Deal with the evaluation of stock levels and produce a printout of any items which have fallen below the stipulated minimum level. The example below indicates those items which require re-ordering to maintain an adequate stock level:

0004	Stock Item 4	5	4
0006	Stock Item 6	10	0

3. Transaction Analysis

The main objective of this program is to produce an analysis of the financial transactions carried out during the current month. The program will deal with the following:

1) Producing monthly statements/invoices/credit notes for forwarding to customers with accounts.
2) Producing a balance sheet of the current month's transactions, including the balance carried over from the previous month and the firm's assets.
3) Estimating the amount of 'purchase' tax to be paid out, by comparing the tax paid to wholesalers/suppliers and tax received from customers.
4) Updating the Customer/Supplier Balance file and creating a new file for the coming month.

The program is composed of:

1) Subroutine 1
2) Subroutine 2
3) Subroutine 3a
4) Subroutine 4
5) Subroutine 6
6) Subroutine 7
7) Subroutine 8t *or* 8d
8) Subroutine 9

Amended listing

```
1 REM ***ACCOUNTS ANALYSIS PROGRAM COMMO
DORE 64***
```

```
5 W2$="[CLR][RVS]ACCOUNTS ANALYSIS PROGR
AM                  [HOME][CD]"
9 BL$=BL$+BL$+BL$:BO$="000000"
10 DIMA1$(999),A2$(100),A3$(100),A8$(999
),A9$(999)
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  FOR INVOICE/STATEMENT
PRINT-OUT         SEQUENCE."
1002 PRINT"[CD]2  FOR LEDGER SHEET PRINT
-OUT SEQUENCE."
1003 PRINT"[CD]3  PURCHASE TAX COMPARISO
N ANALYSIS."
1004 PRINT"[CD]4  TO UPDATE CUSTOMER/SUP
PLIER BALANCE"
1005 PRINT"         FILE FOR NEXT MONTHS
 ACCOUNTS."
1008 PRINT"[CD]5  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]6  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>6THEN1010
1012 PRINTW2$:ONGGOTO2200,4000,5000,6000
,221,9999
1999 REM ***INVOICE/STATEMENT/CREDIT NOT
E PRINT FORMAT***
2000 OPEN4,4,7:I=0:GOSUB502:GOSUB216:IFY
$="N"THENRUN3
2005 A$="J. DOBERMAN & CO. LTD.,"+LEFT$(
BL$,3)+S9$
2010 POKE53265,11:PRINT#4,CHR$(14)A$CHR$
(15):POKE53265,27
2015 REM**COMMODORE DOT MATRIX PRINTER O
NLY***
2020 A$="":L$=CHR$(163):FORJ=1TO45:A$=A$
+L$:NEXT:PRINT#4,A$:L$="":A$=""
2025 A$="22 LONDON ROAD,":GOSUB470
2030 A$="COMPUTOWN CP2 6PC.":GOSUB470
2035 A$="TELEPHONE: 09 1234567     TELEX
 98765":GOSUB470:A$="[SPC]":GOSUB470
2040 GOSUB470:A$=B2$:GOSUB470:A$=B3$:GOS
UB470:A$=B4$:GOSUB470:A$=B5$:GOSUB470
2045 A$=B6$:GOSUB470:A$="[SPC]":GOSUB470
```

```
2050 A$="ACCOUNT NUMBER  "+B1$:GOSUB470:
A$="[SPC]":GOSUB470
2055 L$=CHR$(164):FORJ=1TO80:L1$=L1$+L$:
NEXT:REM **UNDERLINE**
2060 L$=CHR$(163):FORJ=1TO80:L2$=L2$+L$:
NEXT:REM **OVERLINE**
2065 L3$=CHR$(181)+LEFT$(BL$,56)+CHR$(18
1)+LEFT$(BL$,10)+CHR$(181)
2070 L3$=L3$+LEFT$(BL$,10)+CHR$(182)
2075 L4$=LEFT$(BL$,57)+CHR$(181)+LEFT$(B
L$,10)+CHR$(181)
2080 A$=L1$:GOSUB470
2085 A$=CHR$(181)+"GOODS/SERVICES."+LEFT
$(BL$,41)+CHR$(181)
2090 A$=A$+"  AMOUNT  "+CHR$(181)+"    T
AX   "+CHR$(182):GOSUB470
2095 A$=L2$:GOSUB470:A$=L3$:GOSUB470:RET
URN
2100 A$=CHR$(181)+LEFT$(A4$,36)+LEFT$(BL
$,20)+CHR$(181)+MID$(A4$,37,10)
2105 A$=A$+CHR$(181)+RIGHT$(A4$,10)+CHR$
(182):GOSUB470:RETURN
2110 A$=L3$:GOSUB470:GOSUB470:A$=L1$:GOS
UB470:A$=L4$:GOSUB470
2115 L5$=LEFT$(BL$,45)+"TOTAL GOODS "+CH
R$(181)+LEFT$(BL$,10)+CHR$(181)
2120 A$=L5$:GOSUB470:L5$=""
2125 L5$=LEFT$(BL$,47)+"& CHARGES "+CHR$
(181)+V1$+CHR$(181):A$=L5$:GOSUB470
2130 L$=CHR$(163):FORJ=1TO12:L6$=L6$+L$:
NEXT:REM **SHORT LINE**
2135 L6$=LEFT$(BL$,57)+L6$:A$=L6$:GOSUB4
70:A$=L4$:GOSUB470
2140 L7$=LEFT$(BL$,47)+"TOTAL TAX "+CHR$
(181)+V2$+CHR$(181):A$=L7$:GOSUB470
2145 A$=L6$:GOSUB470:A$=L4$:GOSUB470
2150 L8$=LEFT$(BL$,49)+"TOTAL \ "+CHR$(1
81)+V3$+CHR$(181):A$=L8$:GOSUB470
2155 A$=L6$:GOSUB470:A$="[SPC]":GOSUB470
:GOSUB470:A$="":CLOSE4
2160 L1$="":L2$="":L3$="":L4$="":L5$="":
L6$="":L7$="":L8$="":RETURN
2199 REM ***INVOICE/STATEMENT/CREDIT NOT
E PRINT-OUT SEQUENCE***
```

```
2200 GOSUB450:GOSUB457:GOSUB512:GOSUB506
2204 PRINTW2$:PRINT"[CD][CD]ENTER THE NA
ME OF THE CURRENT CUSTOMER/[CD]"
2205 PRINT"                                SUPPLI
ER FILE.[CD][CD]":PRINTW1$:DF=1:L9=16
2210 GOSUB150:GOSUB200:IFY$="N"THEN2200
2215 DF=0:GOSUB823
2220 F9$=D9$:GOSUB733:Z=1:IFER=1THENER=0
:GOTO2220
2225 GOSUB770:A9$(Z)=A9$:A9$="":IFA9$(Z)
="****T"THENGOSUB870:GOSUB901:GOTO2235
2230 Z=Z+1:GOTO2225
2235 PRINTW2$:PRINT"[CD][CD]ENTER THE NA
ME OF THE CURRENT MONTHLY[CD]"
2240 PRINT"                              TRANSACTI
ON FILE.[CD][CD]":PRINTW1$:DF=1:L9=16
2245 GOSUB150:GOSUB200:IFY$="N"THEN2235
2250 DF=0:GOSUB823
2255 F8$=D9$:GOSUB730:X=1:IFER=1THENER=0
:GOTO2255
2260 GOSUB781:A8$(X)=A8$:A8$="":IFA8$(X)
="****T"THENGOSUB869:GOSUB901:GOTO2270
2265 X=X+1:GOTO2260
2270 Z=1:X=1:W=1
2275 IFW=500THENRUN3
2280 IFVAL(LEFT$(A9$(Z),4))=WTHEN2300
2285 IFA9$(Z)="****T"THENZ=1:GOTO2295
2290 Z=Z+1:GOTO2280
2295 W=W+1:GOTO2275
2300 B1$=LEFT$(A9$(Z),4):B2$=MID$(A9$(Z)
,5,20):REM **CUSTOMER NAME AND ADDRESS**
2305 B3$=MID$(A9$(Z),25,20):B4$=MID$(A9$
(Z),45,20):B5$=MID$(A9$(Z),65,20)
2310 B6$=MID$(A9$(Z),85,20)
2315 IFVAL(LEFT$(A8$(X),4))=WTHEN2321
2320 GOTO2340
2321 IFRIGHT$(A8$(X),1)="*"THEN2325
2322 GOTO2340
2325 BB$=MID$(A8$(X),61,10):IFRIGHT$(BB$
,1)=CHR$(160)THENX=1:GOTO2500
2330 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO2330
2335 IFVAL(BB$)=0THENX=1:GOTO2500
2340 IFA8$(X)="****T"THENX=1:GOTO2355
```

```
2345 X=X+1:GOTO2315
2350 REM **STATEMENT ANALYSIS SEQUENCE**
2355 BB$=MID$(A9$(Z),125,10):IFRIGHT$(BB
$,1)=CHR$(160)THENBB$="0":GOTO2370
2360 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO2360
2365 V1$=MID$(A9$(Z),125,10)
2370 IFVAL(BB$)=0THENZ=1:W=W+1:V1$="":GO
TO2275
2375 S9$="STATEMENT.":GOSUB2000
2380 A4$="OUTSTANDING BALANCE"+LEFT$(BL$
,17)+RIGHT$(A9$(Z),20):GOSUB2100
2385 R3=VAL(BB$):R=R3:GOSUB600:R3=R
2390 BB$=RIGHT$(A9$(Z),10):V2$=BB$:IFRIG
HT$(BB$,1)=CHR$(160)THENR4=0:GOTO2405
2395 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO2395
2400 R4=VAL(BB$)
2405 IFR4=0THEN2415
2410 R=R4:GOSUB600:R4=R
2415 R3=R3+R4:D1=R3:GOSUB621:V3$=D3$:D3$
="":GOSUB2110
2420 V1$="":V2$="":V3$="":Z=1:X=1:GOTO30
00
2499 REM ***INVOICE ANALYSIS SEQUENCE***
2500 XX=1
2505 IFA8$(X)="****T"THENA1$(XX)="****T"
:XX=1:GOTO2540
2510 IFVAL(LEFT$(A8$(X),4))=WANDRIGHT$(A
8$(X),1)="*"THEN2520
2515 X=X+1:GOTO2505
2520 IFMID$(A8$(X),69,1)=CHR$(160)THEN25
30:REM **CHECK FOR CREDIT TRANSACTION**
2525 GOTO2535
2530 X=X+1:GOTO2505
2535 A1$(XX)=A8$(X):X=X+1:XX=XX+1:GOTO25
05
2540 IFA1$(XX)="****T"THENA1$(XX)="":GOT
O3000:REM **CREDIT TRANSACTIONS**
2545 S9$="INVOICE.":GOSUB2000
2550 BB$=MID$(A9$(Z),125,10):IFRIGHT$(BB
$,1)=CHR$(160)THEN2595
2555 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO2555
```

```
2560 IFVAL(BB$)=0THEN2595
2565 R3=VAL(BB$):R=R3:GOSUB600:R3=R
2570 R4$=MID$(A9$(Z),135,10):IFRIGHT$(R4
$,1)=CHR$(160)THENR4$="0":GOTO2580
2575 IFLEFT$(R4$,1)=CHR$(160)THENR4$=MID
$(R4$,2):GOTO2575
2580 R=VAL(R4$):GOSUB600:R4=R
2585 A4$="OUTSTANDING BALANCE"+LEFT$(BL$
,17)+RIGHT$(A9$(Z),20):GOSUB2100
2590 D4=R3:R3=0:D5=R4:R4=0
2595 XX=1
2600 IFA1$(XX)="****T"THEN2650
2605 A4$=MID$(A1$(XX),5,36)+MID$(A1$(XX)
,41,20):GOSUB2100:A1$(XX)=""
2610 R3$=MID$(A4$,37,10):IFRIGHT$(R3$,1)
=CHR$(160)THENR3$="0":GOTO2620
2615 IFLEFT$(R3$,1)=CHR$(160)THENR3$=MID
$(R3$,2):GOTO2615
2620 R=VAL(R3$):GOSUB600:R3=R
2625 R4$=MID$(A4$,47,10):IFRIGHT$(R4$,1)
=CHR$(160)THENR4$="0":GOTO2635
2630 IFLEFT$(R4$,1)=CHR$(160)THENR4$=MID
$(R4$,2):GOTO2630
2635 R=VAL(R4$):GOSUB600:R4=R
2640 D4=D4+R3:R3=0
2645 D5=D5+R4:R4=0:XX=XX+1:GOTO2600
2650 D1=D4:GOSUB621:V1$=D3$:D3$=""
2655 D1=D5:GOSUB621:V2$=D3$:D3$=""
2660 IFC4>D4THENV3$=LEFT$(BL$,6)+"0.00":
GOTO2675
2665 D4=D4+D5:IFC4=0THEND1=D4:GOSUB621:V
3$=D3$:D3$="":GOTO2675
2670 C4=C4+C5:D4=D4-C4:D1=D4:GOSUB621:V3
$=D3$:D3$=""
2675 GOSUB2110:C4=0:C5=0:D4=0:D5=0:V1$="
":V2$="":V3$=""
2999 REM *** CREDIT NOTE ANALYSIS SEQUEN
CE***
3000 XX=1:X=1
3005 IFA8$(X)="****T"THENA1$(XX)="****T"
:XX=1:GOTO3030
3010 IFVAL(LEFT$(A8$(X),4))=WANDRIGHT$(A
8$(X),1)="*"THEN3020
3015 X=X+1:GOTO3005
```

```
3020 BB$=MID$(A8$(X),61,10):IFRIGHT$(BB$
,1)=CHR$(160)THENX=X+1:GOTO3005
3021 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO3021
3022 IFVAL(BB$)=0THENX=X+1:GOTO3005
3025 A1$(XX)=A8$(X):X=X+1:XX=XX+1:GOTO30
05
3030 XX=1:IFA1$(XX)="****T"THENW=W+1:X=1
:Z=1:A1$(XX)="":GOTO2275
3035 S9$="CREDIT NOTE":GOSUB2000
3040 IFA1$(XX)="****T"THEN3090
3045 A4$=MID$(A1$(XX),5,36)+MID$(A1$(XX)
,61,20):GOSUB2100:A1$(XX)=""
3050 R1$=MID$(A4$,37,10):IFRIGHT$(R1$,1)
=CHR$(160)THENR1$="0":GOTO3060
3055 IFLEFT$(R1$,1)=CHR$(160)THENR1$=MID
$(R1$,2):GOTO3055
3060 R=VAL(R1$):GOSUB600:R1=R
3065 R2$=MID$(A4$,47,10):IFRIGHT$(R2$,1)
=CHR$(160)THENR2$="0":GOTO3075
3070 IFLEFT$(R2$,1)=CHR$(160)THENR2$=MID
$(R2$,2):GOTO3070
3075 R=VAL(R2$):GOSUB600:R2=R
3080 C4=C4+R1:R1=0
3085 C5=C5+R2:R2=0:XX=XX+1:GOTO3040
3090 C1=C4:GOSUB619:V1$=C3$:C3$=""
3095 C1=C5:GOSUB619:V2$=C3$:C3$=""
3100 C4=C4+C5:C1=C4:GOSUB619:V3$=C3$:C3$
=""
3105 GOSUB2110:C4=0:C5=0:D4=0:D5=0:V1$="
":V2$="":V3$=""
3110 W=W+1:X=1:Z=1:GOTO2275
3999 REM ***LEDGER SHEET PRINT-OUT SEQUE
NCE***
4000 GOSUB450:GOSUB457:GOSUB512:GOSUB506
4005 PRINTW2$:PRINT"[CD][CD]ENTER THE NA
ME OF THE CURRENT MONTHLY[CD]"
4010 PRINT"                              TRANSACTI
ON FILE.[CD][CD]":PRINTW1$:DF=1:L9=16
4015 GOSUB150:GOSUB200:IFY$="N"THEN4005
4020 DF=0:GOSUB823
4025 F8$=D9$:GOSUB730:X=1:IFER=1THENER=0
:GOTO4025
4030 GOSUB781:A8$(X)=A8$:A8$="":IFA8$(X)
```

```
="****T"THENGOSUB869:GOSUB901:GOTO4040
4035 X=X+1:GOTO4030
4040 X=1
4045 OPEN4,4,7:I=0:GOSUB502:GOSUB216:IFY
$="N"THENRUN3
4050 A$=LEFT$(BL$,60)+"     CREDIT"+"
 DEBIT":GOSUB470:A$="":GOSUB492
4055 IFA8$(X)="****T"THEN4165
4056 IFRIGHT$(A8$(X),1)="*"THENX=X+1:GOT
O4055
4060 BB$=MID$(A8$(X),41,10):IFRIGHT$(BB$
,1)=CHR$(160)THEN4120
4065 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO4065
4070 IFVAL(BB$)=0THEN4120
4075 R1$=MID$(A8$(X),41,10)
4080 IFLEFT$(R1$,1)=CHR$(160)THENR1$=MID
$(R1$,2):GOTO4080
4085 R2$=MID$(A8$(X),51,10):IFRIGHT$(R2$
,1)=CHR$(160)THENR2$="0":GOTO4095
4090 IFLEFT$(R2$,1)=CHR$(160)THENR2$=MID
$(R2$,2):GOTO4090
4095 R1=VAL(R1$):R2=VAL(R2$):R1$="":R2$=
""
4100 R=R1:GOSUB600:R1=R:R=0:R=R2:GOSUB60
0:R2=R:R=0:R1=R1+R2:R2=0
4105 C1=R1:GOSUB621:CR=CR+R1:R1=0
4110 A$=LEFT$(A8$(X),4)+"[SPC]"+MID$(A8$
(X),5,36)+LEFT$(BL$,19)+C3$:C3$=""
4115 GOSUB470:A$="":X=X+1:GOTO4055
4120 R3$=MID$(A8$(X),61,10)
4125 IFLEFT$(R3$,1)=CHR$(160)THENR3$=MID
$(R3$,2):GOTO4125
4130 R4$=RIGHT$(A8$(X),10):IFRIGHT$(R4$,
1)=CHR$(160)THENR4$="0":GOTO4140
4135 IFLEFT$(R4$,1)=CHR$(160)THENR4$=MID
$(R4$,2):GOTO4135
4140 R3=VAL(R3$):R4=VAL(R4$):R3$="":R4$=
""
4145 R=R3:GOSUB600:R3=R:R=0:R=R4:GOSUB60
0:R4=R:R=0:R3=R3+R4:R4=0
4150 D1=R3:GOSUB619:DR=DR+R3:R3=0
4155 A$=LEFT$(A8$(X),4)+"[SPC]"+MID$(A8$
(X),5,36)+LEFT$(BL$,29)+D3$:D3$=""
```

```
4160 GOSUB470:A$="":X=X+1:GOTO4055
4165 PRINTW2$:PRINT"[CD][CD]ENTER BALANC
E FROM PREVIOUS MONTH[CD][CD]"
4170 PRINT"
         OR[CD][CD]"
4175 PRINT"# (HASH SIGN) IF NOT REQUIRED
.[CD][CD]":PRINTW1$:L9=9:GOSUB100
4180 IFD9$="#"THEN4225
4185 PRINTW2$:PRINT"[CD][CD]IS THIS AMOU
NT A CREDIT BALANCE?[CD][CD]":PRINTW1$:G
OSUB202
4190 IFY$="Y"THEN4210
4195 R=VAL(D9$):GOSUB600:R3=R:R=0:DR=DR+
R3:D1=R3:R3=0:GOSUB621
4200 A$="DEBIT BALANCE CARRIED FORWARD"+
LEFT$(BL$,41)+D3$:D3$="":GOSUB470
4205 A$="":GOTO4225
4210 R=VAL(D9$):GOSUB600:R1=R:R=0:CR=CR+
R1:C1=R1:R1=0:GOSUB619
4215 A$="CREDIT BALANCE CARRIED FORWARD"
+LEFT$(BL$,30)+C3$:C3$="":GOSUB470
4220 A$=""
4225 A$="[SPC]":GOSUB470:A$="":GOSUB492:
A$="[SPC]":GOSUB470:A$=""
4230 C1=CR:GOSUB619:D1=DR:GOSUB621
4235 A$=LEFT$(BL$,49)+"TOTALS = \ "+C3$+
D3$:GOSUB470:A$="":A$="[SPC]":GOSUB470
4240 C3$="":D3$=""
4245 IFDR>CRTHEN4260
4250 CR=CR-DR:C1=CR:GOSUB619
4255 A$=LEFT$(BL$,41)+"CREDIT BALANCE =
\ "+C3$:GOSUB470:C3$="":GOTO4270
4260 DR=DR-CR:D1=DR:GOSUB621
4265 A$=LEFT$(BL$,42)+"DEBIT BALANCE = \
 "+LEFT$(BL$,10)+D3$:D3$="":GOSUB470
4270 A$="[SPC]":GOSUB470:GOSUB470
4275 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE COMPANY ASSETS[CD]"
4280 PRINT"                                TO
 BE INCLUDED?[CD][CD]":PRINTW1$:GOSUB202
4285 IFY$="N"THENCLOSE4:RUN3
4290 PRINTW2$:PRINT"[CD][CD]PLACE EQUIPM
ENT FILE IN RECORDER.[CD][CD]":PRINTW1$:
GOSUB210
```

```
4295 F9$="EQUIPMENT":GOSUB733:CR=0:IFER=
1THENER=0:GOTO4295
4300 GOSUB783:IFA9$="****T"THEN4320
4305 R1$=RIGHT$(A9$,10):IFRIGHT$(R1$,1)=
CHR$(160)THEN4300
4310 IFLEFT$(R1$,1)=CHR$(160)THENR1$=MID
$(R1$,2):GOTO4310
4315 R1=VAL(R1$):R=R1:GOSUB600:R1=R:R=0:
CR=CR+R1:R1=0:GOTO4300
4320 C1=CR:GOSUB619
4325 A$=LEFT$(BL$,41)+"COMPANY ASSETS =
\ "+C3$:C3$="":GOSUB470
4330 A$="[SPC]":GOSUB470:A$="":CLOSE4:RU
N3
4999 REM ***PURCHASE TAX COMPARISON ANAL
YSIS***
5000 GOSUB450:GOSUB457:GOSUB512:GOSUB506
5005 PRINTW2$:PRINT"[CD][CD]ENTER THE NA
ME OF THE MOST RECENT[CD]"
5010 PRINT"CUSTOMER/SUPPLIER BALANCE FIL
E.[CD][CD]":PRINTW1$:DF=1:L9=16
5015 GOSUB150:GOSUB200:IFY$="N"THEN5005
5020 DF=0:GOSUB823
5025 F8$=D9$:GOSUB730:X=1:IFER=1THENER=0
:GOTO5025
5030 GOSUB766:A8$(X)=A8$:A8$="":IFA8$(X)
="****T"THENGOSUB869:GOSUB901:GOTO5040
5035 X=X+1:GOTO5030
5040 X=1
5045 OPEN4,4,7:I=0:GOSUB502:GOSUB216:IFY
$="N"THENCLOSE4:RUN3
5050 IFA8$(X)="****T"THEN5100
5055 R4$=RIGHT$(A8$(X),10):IFRIGHT$(R4$,
1)=CHR$(160)THENR4$="0":GOTO5065
5060 IFLEFT$(R4$,1)=CHR$(160)THENR4$=MID
$(R4$,2):GOTO5060
5065 IFVAL(R4$)=0THEN5080
5070 IFVAL(LEFT$(A8$(X),4))<500THENX=X+1
:GOTO5050
5075 R4=VAL(R4$):R=R4:GOSUB600:R4=R:R=0:
DR=DR+R4:X=X+1:R4=0:GOTO5050
5080 R2$=MID$(A8$(X),115,10):IFRIGHT$(R2
$,1)=CHR$(160)THENR2$="0":GOTO5090
5085 IFLEFT$(R2$,1)=CHR$(160)THENR2$=MID
```

```
$(R2$,2):GOTO5085
5090 IFVAL(R2$)=0THENX=X+1:R2$="":GOTO50
50
5095 R2=VAL(R2$):R=R2:GOSUB600:R2=R:R=0:
CR=CR+R2:X=X+1:R2=0:GOTO5050
5100 D1=DR:GOSUB621:C1=CR:GOSUB619
5105 A$="TOTAL TAX PAID OUT     = \ "+D3$
:GOSUB470:A$="[SPC]":GOSUB470
5110 A$="TOTAL TAX RECEIVED     = \ "+C3$
:GOSUB470:A$="[SPC]":GOSUB470
5115 IFDR>CRTHEN5130
5120 CR=CR-DR:C1=CR:GOSUB619
5125 A$="TAX DUE TO GOVERNMENT = \ "+C3$
:GOSUB470:A$="[SPC]":GOSUB470:RUN3
5130 A$="TAX PAID OUT IS GREATER THAN TA
X RECEIVED"
5135 GOSUB470:A$="[SPC]":GOSUB470:RUN3
5989 REM ***UPDATE OF CUSTOMER/SUPPLIER
TRANSACTION FILE***
5990 IFRIGHT$(BB$,1)=CHR$(160)THENBB$="0
":RETURN
5991 IFLEFT$(BB$,1)=CHR$(160)THENBB$=MID
$(BB$,2):GOTO5991
5992 RETURN
6000 PRINTW2$:PRINT"[CD][CD]ENTER THE NA
ME OF THE CURRENT CUSTOMER/[CD]"
6005 PRINT"                    SUPPLIER BALAN
CE FILE.[CD][CD]":PRINTW1$:DF=1:L9=16
6010 GOSUB150:GOSUB200:IFY$="N"THEN6000
6015 DF=0:GOSUB823
6020 F9$=D9$:GOSUB733:Z=1:IFER=1THENER=0
:GOTO6020
6025 GOSUB770:A9$(Z)=A9$:A9$="":IFA9$(Z)
="****T"THENGOSUB870:GOSUB901:GOTO6035
6030 Z=Z+1:GOTO6025
6035 PRINTW2$:PRINT"[CD][CD]ENTER THE NA
ME OF THE CURRENT MONTHLY[CD]"
6040 PRINT"                          TRANSACTI
ON FILE.[CD][CD]":PRINTW1$:DF=1:L9=16
6045 GOSUB150:GOSUB200:IFY$="N"THEN6035
6050 DF=0:GOSUB823
6055 F8$=D9$:GOSUB730:X=1:IFER=1THENER=0
:GOTO6055
6060 GOSUB781:A8$(X)=A8$:A8$="":IFA8$(X)
```

```
="****T"THENGOSUB869:GOSUB901:GOTO6070
6065 X=X+1:GOTO6060
6070 Z=1:X=1
6075 IFLEFT$(A8$(X),4)=LEFT$(A9$(Z),4)TH
EN6085
6080 IFA8$(X)="****T"THENZ=1:GOTO6525
6085 IFVAL(LEFT$(A8$(X),4))>499THEN6340
6090 BB$=MID$(A8$(X),41,10):GOSUB5990:RE
M **CUSTOMER ACCOUNT UPDATE**
6095 IFVAL(BB$)=0THEN6105
6100 GOTO6115
6105 IFRIGHT$(A8$(X),1)="*"THENX=X+1:Z=1
:GOTO6075
6110 GOTO6195
6115 R3=VAL(BB$)
6120 BB$=MID$(A8$(X),51,10):GOSUB5990
6125 R4=VAL(BB$)
6130 R=R3:GOSUB600:R3=R:IFR4=0THEN6140
6135 R=R4:GOSUB600:R4=R
6140 BB$=MID$(A9$(Z),125,10):GOSUB5990
6145 R5=VAL(BB$)
6150 BB$=MID$(A9$(Z),135,10):GOSUB5990
6155 R6=VAL(BB$)
6160 R=R5:GOSUB600:R5=R:IFR6=0THEN6170
6165 R=R6:GOSUB600:R6=R
6170 R3=R3+R5:R4=R4+R6:D1=R3:GOSUB621:DR
$=D3$:D3$=""
6175 IFR4=0THEND3$=LEFT$(BL$,10):GOTO618
5
6180 D1=R4:GOSUB621
6185 D3$=DR$+D3$
6190 A9$(Z)=LEFT$(A9$(Z),124)+D3$:D3$=""
:Z=1:X=X+1:GOTO6075
6195 BB$=MID$(A8$(X),61,10):GOSUB5990
6200 R3=VAL(BB$)
6205 BB$=MID$(A8$(X),71,10):GOSUB5990
6210 R4=VAL(BB$)
6215 R=R3:GOSUB600:R3=R:IFR4=0THEN6225
6220 R=R4:GOSUB600:R4=R
6225 BB$=MID$(A9$(Z),105,10):GOSUB5990
6230 R5=VAL(BB$)
6235 BB$=MID$(A9$(Z),115,10):GOSUB5990
6240 R6=VAL(BB$)
6245 R=R5:GOSUB600:R5=R:IFR6=0THEN6255
```

```
6250 R=R6:GOSUB600:R6=R
6255 R1=R3:R2=R4:R3=R3+R5:R4=R4+R6:D1=R3
:GOSUB621:DR$=D3$:D3$=""
6260 IFR4=0THEND3$=LEFT$(BL$,10):GOTO627
0
6265 D1=R4:GOSUB621
6270 D3$=DR$+D3$:DR$=""
6275 A9$(Z)=LEFT$(A9$(Z),104)+D3$+RIGHT$
(A9$(Z),20):D3$=""
6280 BB$=MID$(A9$(Z),125,10):GOSUB5990
6285 R5=VAL(BB$)
6290 BB$=MID$(A9$(Z),135,10):GOSUB5990
6295 R6=VAL(BB$)
6300 R=R5:GOSUB600:R5=R:IFR6=0THEN6310
6305 R=R6:GOSUB600:R6=R
6310 R3=R5-R1:R4=R6-R2:D1=R3:GOSUB621:DR
$=D3$:D3$=""
6315 IFR4=0THEND3$=LEFT$(BL$,10):GOTO632
5
6320 D1=R4:GOSUB621
6325 D3$=DR$+D3$
6330 A9$(Z)=LEFT$(A9$(Z),124)+D3$:D3$=""
:Z=1:X=X+1:GOTO6075
6335 REM ** UPDATE OF RECORDS OTHER THAN
 CUSTOMER ACCOUNTS **
6340 BB$=MID$(A8$(X),41,10):GOSUB5990
6345 IFVAL(BB$)=0THEN6355
6350 GOTO6435
6355 R3=VAL(BB$)
6360 BB$=MID$(A8$(X),51,10):GOSUB5990
6365 R4=VAL(BB$)
6370 R=R3:GOSUB600:R3=R:IFR4=0THEN6380
6375 R=R4:GOSUB600:R4=R
6380 BB$=MID$(A9$(Z),105,10):GOSUB5990
6385 R5=VAL(BB$)
6390 BB$=MID$(A9$(Z),115,10):GOSUB5990
6395 R6=VAL(BB$)
6400 R=R5:GOSUB600:R5=R:IFR6=0THEN6410
6405 R=R6:GOSUB600:R6=R
6410 R3=R3+R5:R4=R4+R6:D1=R3:GOSUB621:DR
$=D3$:D3$=""
6415 IFR4=0THEND3$=LEFT$(BL$,10):GOTO642
5
6420 D1=R4:GOSUB621
```

```
6425 D3$=DR$+D3$
6430 A9$(Z)=LEFT$(A9$(Z),104)+D3$+RIGHT$
(A9$(Z),20):D3$="":Z=1:X=X+1:GOTO6075
6435 BB$=MID$(A8$(X),61,10):GOSUB5990
6440 R3=VAL(BB$)
6445 BB$=MID$(A8$(X),71,10):GOSUB5990
6450 R4=VAL(BB$)
6455 R=R3:GOSUB600:R3=R:IFR4=0THEN6465
6460 R=R4:GOSUB600:R4=R
6465 BB$=MID$(A9$(Z),125,10):GOSUB5990
6470 R5=VAL(BB$)
6475 BB$=MID$(A9$(Z),135,10):GOSUB5990
6480 R6=VAL(BB$)
6485 R=R5:GOSUB600:R5=R:IFR6=0THEN6495
6490 R=R6:GOSUB600:R6=R
6495 R3=R3+R5:R4=R4+R6:D1=R3:GOSUB621:DR
$=D3$:D3$=""
6500 IFR4=0THEND3$=LEFT$(BL$,10):GOTO651
0
6505 D1=R4:GOSUB621
6510 D3$=DR$+D3$
6515 A9$(Z)=LEFT$(A9$(Z),124)+D3$:D3$=""
:Z=1:X=X+1:GOTO6075
6520 REM ***RECORD NEW CUSTOMER/SUPPLIER
 BALANCE FILE***
6525 M1$=F9$:GOSUB931:F8$=FF$:REM **NEW
NAME FOR RECORD FILE***
6530 PRINTW2$:PRINT"[CD][CD]PLACE DATA T
APE IN RECORDER TO[CD]"
6535 PRINT"RECORD NEW CUSTOMER/SUPPLIER
BALANCE[CD]"
6540 PRINT"
   FILE.[CD][CD]":PRINTW1$
6545 GOSUB710:Z=1:IFER=1THENER=0:GOTO654
5
6550 A8$=A9$(Z):GOSUB792:IFA8$="****T"TH
ENGOSUB869:GOSUB901:RUN3
6555 Z=Z+1:GOTO6550
```

Commentary

Lines 2000–3110: Will produce either an invoice, a statement of account or a credit note depending on the data contained in the current Monthly

Transaction file. To produce the various printouts, the Monthly Transaction file contains the following records:

Record 1

```
0001  20 × Stock Item 1      St.No. 0001          22.00      3.40
```

Record 2

```
0005   5 × Stock Item 5      St.No. 0005
       11.00            0.00
```

Record 3

```
0001  Repair to boiler                            10.00      1.50
```

In addition, the Customer/Supplier Balance file is also referred to and the contents of this file are as follows:

0001	Mr. Customer A	Address A1		Address A2		Address A3
	Address A4	0.00	0.00	100.00	15.00	
0002	Customer B	Address B1		Address B2		Address B3
		200.00	30.00	100.00	15.00	
0003	Customer C	Address C1		Address C2		
		0.00	0.00	0.00	0.00	
0004	Customer D	Address D1		Address D2		
		0.00	0.00	0.00	0.00	
0005	Customer E	Address E1		Address E2		Address E3
		100.00	15.00	100.00	15.00	
0501	Supplier A	Address S1		Address S2		Address S3
		100.00	15.00	1000.00	150.00	

Lines 2000–2160: Have been included in this program to demonstrate some of the graphic capabilities of the Commodore 1515 dot matrix printer in producing such items as invoices.

Lines 2200–2345: Recall the records from the Customer Supplier Balance file and the Monthly Transaction file. Depending upon the data in these files, the correct analysis sequence is chosen. Both the statement and the invoice sequence carry on to the credit note routine.

Figure 3.8

J. Doberman & Co. Ltd., Statement.

22 London Road,
ComPutown CP2 6PC.
TelePhone: 09 1234567 Telex 98765

Customer B.
Address B1.
Address B2.
Address B3.

Account Number 0002

Goods/Services.	Amount	Tax
Outstandin9 Balance	100.00	15.00
Total Goods & Char9es	100.00	
Total Tax	15.00	
Total £	115.00	

Lines 2350–2420: Deal with the statement sequence: produced **Figure 3.8.**

Figure 3.9

J. Doberman & Co. Ltd., Invoice.

22 London Road,
Computown CP2 6PC.
Telephone: 09 1234567 Telex 98765

Mr. Customer A.
Address A1.
Address A2.
Address A3.
Address A4.

Account Number 0001

Goods/Services.	Amount	Tax
Outstanding Balance	100.00	15.00
Repair to boiler.	10.00	1.50
Total Goods & Charges	110.00	
Total Tax	16.50	
Total £	126.50	

Lines 2500-2675: Deal with the invoice sequence (**Figure 3.9**)

Figure 3.10

J. Doberman & Co. Ltd., Credit Note

22 London Road,
ComPutown CP2 6PC.
TelePhone: 09 1234567 Telex 98765

Customer E.
Address E1.
Address E2.
Address E3.

Account Number 0005

Goods/Services.		Amount	Tax
5 X Stock Item 5.	St.No. 0005	11.00	0.00
	Total Goods & Charges	11.00	
	Total Tax	0.00	
	Total £	11.00	

Lines 3000–3110: Deal with the credit note routine **(Figure 3.10)**.

Figure 3.11

		Credit	Debit
0001 20 X Stock Item 1.	St.No. 0001	25.40	
0005 5 X Stock Item 5.	St.No. 0005		11.00
0001 Repair to boiler.		11.50	
Credit Balance Carried Forward		5000.00	
	Totals = £	5036.90	11.00
	Credit Balance = £	5025.90	
	Company Assets = £	5603.76	

Lines 4000–4160: Concerned with the analysis of the Monthly Transaction file to produce the simple balance sheet in **Figure 3.11**. The 'purchase' tax, held in separate record fields, is combined with the credit or debit transactions and included in the final figures. In this routine example, the operator is requested to enter the balance carried forward from the previous month in lines 4165–4215. The inclusion of a balance carry forward is optional. After completion of the balance totals, the Equipment file may be incorporated to indicate the firm's assets. Further programming could include stock as part of the assets and account could also be taken of outstanding debts to give a more complete picture.

Total Tax Paid Out	= 150.00
Total Tax Received	= 60.00
Tax DUE from Government	= 90.00

Although this is only a simple printout of the comparison between tax paid to suppliers, etc., and tax received from customers, it nevertheless should save a few headaches at the end of a tax year. It can also be used at any time during the financial year to indicate the state of the balance. To obtain up-to-date figures, it is essential that the Customer/Supplier Balance file is updated from the figures in the Monthly Transaction file.

Lines 5055–5095: Examine each record in the CSB file for credit tax received and debit tax paid, but will ignore the debit record of account customers. This record field is maintained for invoice/statement print-

outs and does not represent tax paid or received. The tax paid or received for customer accounts is kept up-to-date in Field 8.

Lines 6000–6555: Concerned with the updating of the Customer/Supplier Balance file and the creation of a new file for the coming month's transactions. The records created in the Monthly Transaction file are analysed to produce the required updated file. Lines 6090–6330 update the customer accounts and lines 6340–6515 update all other transactions.

Lines 6525–6555 initially access the new file name routine, ie GOSUB 931, which would change a present file name of MARCH 84 to APRIL 84, and the new Customer/Supplier Balance file is created.

CHAPTER 4
Word Processing

A word processor is becoming a common facility in companies and institutions. It can vary from a dedicated word processing machine to a microcomputer system with a word processing program as just one of its functions.

The aims of the program listed in the first section of this chapter are to enable the Commodore 64, in conjunction with a printer, to be used as a typewriter and to provide word processing facilities. The program listed in the second section will provide a bank of names and addresses which can be referred to by the first program.

This added facility will allow an unlimited, personally addressed, number of top copy letters to be produced. The programs are sufficiently sophisticated as they are to suit most purposes in the office and home. This entire book was composed using this system, so that it has been well and truly tested.

1. Word Processing

As already discussed in Chapter 1, the question as to whether to have one large program or a number of smaller programs arises with the following word processing system. As the book is primarily aimed at owners of a tape system, the following program has been divided into three parts, mainly to accommodate the paragraph construction sequence which necessitates the holding of data records in memory. Each program, however, has been numbered to allow the listings to be entered as one long program.

Owners of a disc system are not inconvenienced to any large extent, as program loading from disc is rapid, but the amount of records which can be edited can be greatly increased due to the fact that two lines or channels may be open at the same time for reading and writing data. This is also true for the chapter/paragraph construction, the only limitation being the amount of available space on a disc or discs.

a. Creating and editing record files

The aims and objectives of this program are as follows:

1) To create a new data record file.
2) To allow the addition of further records to an existing file.
3) To allow editing of the records in a file.

Start by combining the following subroutines together, using the merge facility in Subroutine 1:

1) Subroutine 3
2) Subroutine 4
3) Subroutine 6
4) Subroutine 8t *or* 8d
5) Subroutine 9

The following listing indicates the necessary changes to Subroutines 1 and 9. The name 'Super Secretary' in line 5 is, of course, purely optional.

Completely modified Subroutines 5 and 10 are given separately, later in this sub-section, under the headings 'Edit Data' and 'Creating and Adding to Text Files'. The reason for this is that these have been modified so much that it is simpler to enter them as completely new routines.

Amended listing

```
1 REM ***SUB 1.  SUPER.SEC.64 TAPE VERSI
ON  19K. COMMODORE 64***
5 W2$="[CLR][RVS]SUPER SECRETARY.[SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC]
[SPC][SPC][SPC][SPC][SPC][HOME][CD]"
10 DIMA8$(500),A9$(500)
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO START A NEW DATA FI
LE."
1002 PRINT"[CD]2  TO ADD NEW DATA TO A F
ILE."
1003 PRINT"[CD]3  TO EDIT A DATA FILE."
1004 PRINT"[CD]4  FOR SCREEN DISPLAY OF
DATA FILE."
1008 PRINT"[CD]5  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]6  TO END PROGRAM.":PRINT
W1$
```

```
1010 G=VAL(G$):IFG<10RG>6THEN1009
1011 PRINTW2$:ONGGOTO2500,2600,300,897,2
21,9999
```

Edit data

The following listing (lines 295–436) is, in the main, a repeat of the Edit routine (Subroutine 5) in Chapter 2, but incorporates additional lines and modifications to cater for the special word processor requirements.

The following lines of the Chapter 2 Edit routine should be altered or included to correspond to the following listing: lines 337, 338, 339, 355, 357, 358, 359, 360, 361, 362, 379, 382, 383, 384, 385, 391, 396, 397, 398, 399, 400, 401, 402, 403, 416, 418, 419, 420 and 430.

Line 337 sets up the sound control commands necessary to warn the operator that the maximum line length is imminent.

Lines 354–361 display instructions for the inclusion of control characters for bold, underlining, super- and sub-script functions on printout.

Lines 396–403 allow these special characters to be included in the data record by activating the F7 function key, followed by the appropriate character control key. In line 396, allowances are made for the F7 key being activated with the shift key down, ie CHR$(136). The F7 key then becomes CHR$(140), the F8 function key. (NB. The F7 key has to be activated before each control character can be entered.)

Lines 382–385 allow the use of the delete key, as it is not vitally important to maintain record fields in word processing.

The second modification, which is optional, is the recall of the complete file from tape or disc, and its placement into the computer memory before editing begins.

The following lines will need to be altered to correspond to the following listing if this technique is to be employed: lines 295, 296, 301, 315, 317, 320, 321, 326, 330, 347, 350, 367, 368, 430, 433, 434 and 435.

These lines DO NOT need alteration if you are prepared to work on the document line by line.

Lines 295–296 recall a record file, and the data records are allocated to the subscript variable A9$(U). The delay in recalling data, record by record, during the editing process, using a tape system especially, can be irritating. There is, however, the maximum amount of memory available in the computer to be considered, since this may not be sufficient for the whole document. If the majority of files to be handled are likely to be lengthy documents, this method may not be suitable.

Full listing

```
295 GOSUB770:A9$(U)=A9$:A9$="":IFA9$(U)=
"****T"THENGOSUB870:RETURN
296 U=U+1:GOTO295
297 REM ##################################
#############
298 REM ***SUB 5T.  EDIT DATA  3.6K  COM
MODORE 64***
299 REM ##################################
#############
300 GOSUB823:GOSUB703:F9$=D9$:D9$="":GOS
UB733:IFER=1THENER=0:GOTO300
301  I=1:P4=239:U=1:GOSUB295:U=0
302 PRINTW2$:PRINT"[CD][CD][RVS]TYPE 1[R
VO]  TO EDIT SPECIFIC DATA LINES.[CD][CD
]"
303 PRINT"[RVS]TYPE 2[RVO]  TO EDIT ALL
DATA LINES.[CD][CD]":PRINTW1$
304 G$="":GETG$:IFG$=""THEN304
305 PRINTW2$:IFG$="1"THENRT=1:DIMK$(500)
:NN=1:GOTO308
306 IFG$="2"THENTR=0:GOTO315
307 GOTO304
308 PRINTW2$
309 PRINT"[CD][CD]TYPE IN LINE NUMBERS Y
OU WISH TO EDIT"
310 PRINT"[CD][CD]IN ASCENDING ORDER - P
RESS RETURN AFTER"
311 PRINT"[CD][CD]EACH LINE NUMBER IS EN
TERED."
312 PRINT"[CD][CD]TYPE  #  TO END NUMBER
 ENTRY[CD][CD]":PRINTW1$
313 L9=3:GOSUB150:IFD9$="#"THENK$(NN)="*
***T":D9$="":NN=1:NL=0:PRINTW2$:GOTO315
314 K$(NN)=D9$:D9$="":NN=NN+1:GOTO308
315 A9$(U)"":U=U+1:NL=NL+1
316 IFRT=0THEN321
317 IFA9$(U)="****T"THENA8$(I)="****T":G
OSUB210:GOTO433
318 IFK$(NN)="****T"THEN435
319 IFVAL(K$(NN))=NLTHENNN=NN+1:GOTO321
320 A8$(I)=A9$(U):I=I+1:GOTO315
321 PRINTW2$:PRINTA9$(U)
```

```
322 PRINTW1$:PRINT"[CD]**SPACE BAR**  FO
R NO CHANGE REQUIRED."
323 PRINT"[CD][CD][RVS]1[RVO]  TO FINISH
 EDITING.[CD]"
324 PRINT"[CD][CD][RVS]2[RVO]  TO EDIT,
DELETE OR ADD A NEW LINE.[CD]":PRINTW1$
325 G$="":GETG$:IFG$=""THEN325
326 IFG$=CHR$(32)ORG$=CHR$(160)THENA8$(I
)=A9$(U):PRINTW2$:GOTO429
327 IFG$="1"THENPRINTW2$:GOTO435
328 IFG$="2"THEN330
329 GOTO325
330 PRINTW2$:PRINTA9$(U):PRINTW1$
331 PRINT"[CD][RVS]1[RVO]  TO INSERT A N
EW LINE."
332 PRINT"[CD][RVS]2[RVO]  TO REMOVE LIN
E COMPLETELY."
333 PRINT"[CD][RVS]3[RVO]  TO EDIT OR CH
ANGE LINE.":PRINTW1$
334 G$="":GETG$:IFG$=""THEN334
335 G=VAL(G$):IFG<1ORG>3THEN334
336 ONGGOTO337,347,353
337 S1=54296:S2=54276:S3=54273:S4=54277:
POKES1,15:REM ***NEW LINE***
338 GOSUB2090:K=0:GOSUB2115
339 IFG=157THENA8$(I)="****T"
341 PRINTW2$:PRINTA8$(I)
342 PRINT"[CD][CD]IS THIS DATA TO BE INC
LUDED?":PRINTW1$
343 GOSUB202:IFY$="N"THENA8$(I)="":GOTO3
21
344 IFA8$(I)="****T"THEN433
345 I=I+1:GOTO321
346 REM ***REMOVE LINE FROM DATA FILE***
347 PRINTW2$:PRINTA9$(U)
348 PRINT"[CD][CD]IS THIS DATA TO BE DEL
ETED?":PRINTW1$
349 GOSUB202:IFY$="N"THEN321
350 IFA9$(U)="****T"THEN321
351 GOTO315
352 REM ***EDIT LINE***
353 PRINTW2$
354 PRINT"MOVE CURSOR ALONG LINE AND COR
RECT.[CD]"
```

```
355 PRINT"F7 KEY FOR CONTROL CHARACTER M
ODE."
356 PRINT"PRESS RETURN KEY WHEN COMPLETE
.":PRINTW1$
357 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD][CD][CD][CD][CD][CD][CD][CD][CD][CD]
[CD][CD][CD][CD]";W1$;
358 PRINT"CONTROL[SPC]MODE"
359 W3$="F1=UNDERLINE "+CHR$(191)+"  CRS
UP=SUPERSCRIPT "+CHR$(177)
360 W4$="CRSDOWN=SUBSCRIPT "+CHR$(178)+"
  F3=BOLD "+CHR$(166)
361 PRINTW3$:PRINTW4$
362 P=1544:P1=P:L%=LEN(A9$):P2=L%:P2=P2+
P1:P3=P1+240:REM** MAX.STRING.LEN.**
367 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD]";A9$(U):PRINT
368 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD][CD][CD][CD][CD][CD]":PRINTA9$(U)
369 A=PEEK(P):PRINT"[HOME][CD][CD][CD][C
D][CD][CD][CD][CD][CD][CD][CD][CD]"
370 POKE204,0
371 G$="":GETG$:IFG$=""THEN371
372 G=ASC(G$):IFG=13ORG=141THENPOKE204,1
:POKEP,A:GOTO412:REM ***RETURN***
373 IFG=29THEN375:REM ***CURSOR RIGHT***
374 GOTO378
375 IFP+1>P2THENP2=P2+1
376 IFP2>P3THENP2=P2-1:GOTO371
377 B=PEEK(P+1):PRINTG$;:POKEP,A:P=P+1:A
=B:GOTO371
378 IFG=145THEN380:REM ***CURSOR UP***
379 GOTO382
380 IFP-40<P1THEN371
381 B=PEEK(P-40):PRINTG$;:POKEP,A:P=P-40
:A=B:GOTO371
382 IFG=20THEN384:REM ***DELETE KEY***
383 GOTO386
384 IFP=P1THENPOKEP,A:GOTO371
385 PRINTG$;:P=P-1:GOTO371
386 IFG=157THEN388:REM ***CURSOR LEFT***
387 GOTO390
388 IFP=P1THENPOKEP,A:GOTO371
389 B=PEEK(P-1):PRINTG$;:POKEP,A:P=P-1:A
```

```
=B:GOTO371
390 IFG=17THEN392:REM ***CURSOR DOWN***
391 GOTO396
392 IFP+40>P2THENP2=P2+40
393 IFP2>P3THENP2=P2-40:GOTO371
394 B=PEEK(P+40):PRINTG$;:POKEP,A:P=P+40
:A=B
395 GOTO371
396 IFG=136ORG=140THEN398:REM ***F7 OR F
8 KEY***
397 GOTO404
398 G$="":GETG$:IFG$=""THEN398
399 G=ASC(G$):IFG=133ORG=137THENG=191:GO
TO403
400 IFG=134ORG=138THENG=166:GOTO403
401 IFG=145THENG=177:GOTO403
402 IFG=17THENG=178
403 G$=CHR$(G):GOTO408
404 IFG=32THENG$=CHR$(160):GOTO408
405 IFG<33ORG>95THEN407
406 GOTO408
407 IFG<193THENPOKEP,A:GOTO371
408 B=PEEK(P+1):IFP+1>P3THEN371
409 A=B:IFP+1=>P2THENP2=P2+1
410 IFP2>P3THENP2=P3
411 PRINTG$;:P=P+1:GOTO371
412 P=P1:PRINT"[HOME][CD][CD][CD][CD][CD
][CD][CD][CD][CD][CD][CD][CD][CD][CD][CD
][CD][CD][CD][CD]PLEASE WAIT.":IFP2>P3TH
ENP2=P3
413 A=PEEK(P):IFA=32THEN421
414 IFA<32THENA=A+64:GOTO421
415 IFA=96ORA=160THENA=160:GOTO421
416 IFA=102ORA=113ORA=114ORA=127THENA=A+
64:GOTO421
417 IFA>64ANDA<91THENA=A+128:GOTO421
418 IFA=44ORA=108ORA=172THENA=172:GOTO42
1
419 IFA=59ORA=123ORA=187THENA=187:GOTO42
1
420 IFA=34ORA=98ORA=162THENA=162
421 A$=CHR$(A):A8$(I)=A8$(I)+A$:P=P+1
422 IFP=P2THEN424
423 GOTO413
```

```
424 L%=LEN(A8$(I)):A=ASC(RIGHT$(A8$(I),1
))
425 IFA=32THENA8$(I)=LEFT$(A8$(I),L%-1):
GOTO424
426 PRINTW2$:PRINTA8$(I)
427 PRINT"[CD][CD]ARE YOUR CORRECTIONS C
ORRECT?[CD][CD]":PRINTW1$
428 GOSUB202:IFY$="N"THENA8$(I)="":GOTO3
21
429 IFA8$(I)="****T"THEN433
430 I=I+1:GOTO315
433 F8$=F9$:GOTO2615:REM***ADD NEW DATA
ROUTINE***
434 U=U+1
435 A8$(I)=A9$(U):A9$(U)="":IFA8$(I)="**
**T"THEN433
436 I=I+1:GOTO434
```

Creating and adding to text files

The aim of the next routine (Lines 1990–2645) is to create a new letter or text file or to add further information to an existing one. Due to the special requirements of this system, it bears little resemblance to the standard file creation routine, ie Subroutine 10, and is the reason why this was not included in the 'subroutines to be included' at the beginning of this section.

Full listing

```
1990 REM ##################################
####################
1991 REM ***SUB 10.  CREATE NEW OR ADD T
O EXISTING FILE.***
1992 REM ***WORD PROCESSOR  COMMODORE 64
***
1993 REM ##################################
####################
1999 REM ***SET MARGINS***
2000 PRINTW2$:JJ=J7:J8$="":J9$="":J8=0:J
9=0
2005 PRINT"[CD][CD]SET LEFT HAND (MARGIN
).[CD][CD]"
2010 PRINT"TYPE IN THE NUMBER OF SPACES
 OR  0.[CD]"
2015 PRINT"OR  #  TO RESET MARGINS[CD][C
D]":PRINTW1$
```

```
2020 L9=2:GOSUB150:J9$=D9$:D9$=""
2025 IFJ9$="#"THENJJ=J7:TR=0:RETURN
2030 PRINTW2$
2035 PRINT"[CD][CD]SET RIGHT HAND (MARGI
N).[CD][CD]"
2040 PRINT"TYPE IN THE NUMBER OF SPACES
 OR  0.[CD][CD]":PRINTW1$
2045 L9=2:GOSUB150:J8$=D9$:D9$="":TR=1
2050 IFJ9$="0"THENJ8=VAL(J8$):J8$=LEFT$(
BL$,J8):GOTO2070
2055 J9=VAL(J9$):J9$=LEFT$(BL$,J9)
2060 IFJ8$="0"THEN2075
2065 J8=VAL(J8$):J8$=LEFT$(BL$,J8):J6=J9
+J8:JJ=JJ-J6:TR=3:RETURN
2070 JJ=JJ-J8:TR=2:RETURN
2075 JJ=JJ-J9:TR=1:RETURN
2080 PRINTJ9$;:FORJ=1TOJJ:PRINT".";:NEXT
J:PRINT:PRINT
2085 PRINTJ9$;:RETURN
2090 PRINTW2$
2095 PRINT"[CD][CD][CD]HOW MANY CHARACTE
RS PER LINE?[CD][CD]"
2100 PRINT"NORMALLY YOU REQUIRE  70 -80.
[CD]"
2105 PRINT"BUT MAX FOR QUME=137  SCRIPTA
=123.[CD][CD]"
2110 PRINTW1$:L9=3:GOSUB150:JJ=VAL(D9$):
J7=JJ:D9$="":RETURN
2115 PRINTW2$:PRINT"LINE NUMBER:-  ";I
2120 PRINT"[CD]F1 = UNDERLINE (START & C
ANCEL)    ";CHR$(191)
2125 PRINT"F3 = BOLD  (START & CANCEL)
     ";CHR$(166)
2130 PRINT"CURSOR UP = SUPERSCRIPT
     ";CHR$(177)
2135 PRINT"CURSOR DOWN = SUBSCRIPT
     ";CHR$(178)
2140 PRINT"CURSOR RIGHT TO SET MARGINS."
2145 PRINT"CURSOR LEFT TO END LETTER.":P
RINTW1$
2150 PRINT"[CD]":IFI=2THENPRINTA8$(I-1):
GOTO2160
2155 IFI=>3THEN:PRINTA8$(I-2):PRINTA8$(I
-1)
```

```
2160 PRINT"[CD][CD]START TYPING FOLLOWED
 BY <RETURN>."
2165 IFTR=1ORTR=3THENGOSUB2080:GOTO2180
2170 FORJ=1TOJJ:PRINT".";:NEXTJ:PRINT:PR
INT:PRINT"[RVS] [RVO][CL]";
2175 GETG$:IFG$<>""THEN2175
2180 G$="":GETG$:IFG$=""THEN2180
2185 G=ASC(G$):IFG=13ORG=141ORG=20ORG=14
8THENG$="":GOTO2180
2190 IFG=157THENPRINT" ":RETURN
2195 IFG=29THENGOSUB2000:GOTO2115
2200 GOTO2215
2205 PRINT"[RVS] [RVO][CL]";
2210 G$="":GETG$:IFG$=""THEN2205
2215 G=ASC(G$):IFG=13ORG=141THENPRINTCHR
$(13):G=13:PRINT" ";:RETURN
2220 IFG=20ORG=157THEN2230
2225 GOTO2240
2230 L=LEN(A8$(I)):IFL<1THEN2175
2235 A8$(I)=LEFT$(A8$(I),L-1):PRINT" [CL
][CL][RVS] [RVO][CL]";:GOTO2210
2240 IFG=34THENPRINTCHR$(39);:GOTO2290:R
EM **QUOTE MARKS COMMA & COLON**
2245 IFG=145THENG=177:K=K+1:GOTO2285:REM
 **SUPERSCRIPT**
2250 IFG=17THENG=178:K=K+1:GOTO2285:REM
**SUBSCRIPT**
2255 IFG=133ORG=137THENG=191:K=K+1:GOTO2
285:REM **UNDERLINE**
2260 IFG=134ORG=138THENG=166:K=K+1:GOTO2
285:REM **BOLD**
2265 IFG=32THENG=160:GOTO2280:REM **SPAC
E**
2270 IFG=160THEN2280
2275 IFG>=95ANDG<193ORG<33ORG>218THEN221
0
2280 PRINTG$;:GOTO2290
2285 PRINTCHR$(G);
2290 G$=CHR$(G):A8$(I)=A8$(I)+G$
2295 IFLEN(A8$(I))-K=>JJTHEN2315
2300 IFLEN(A8$(I))-K=>JJ-5THEN2310
2305 GOTO2205
2310 POKES2,33:POKES4,10:FORS5=1TO10:POK
ES3,72:NEXT:POKES2,0:POKES3,0:GOTO2205
```

```
2315 G$="":GETG$:IFG$=""THEN2315
2320 G=ASC(G$):IFG=13ORG=141THENPRINTCHR
$(13):G=13:RETURN
2325 IFG=20ORG=157THEN2220
2330 GOTO2315
2499 REM ***TO START A NEW LETTER***
2500 I=1:GOSUB823:GOSUB701:F8$=D9$:D9$="
":GOSUB710:IFER=1THENER=0:GOTO2500
2505 S1=54296:S2=54276:S3=54273:S4=54277
:POKES1,15:GOSUB2090
2510 K=0:GOSUB2115:IFG=157THEN2525
2515 IFTR=1ORTR=3THENA8$(I)=J9$+A8$(I)
2520 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO2510
2525 GOSUB218:IFY$="N"THENA8$(I)="":GOTO
2510
2530 A8$(I)="****T":A8$=A8$(I):GOSUB792:
GOSUB869:GOSUB901:IFAE$="Y"THENGOSUB879
2535 GOSUB840:RUN3
2599 REM ***ADD NEW DATA TO LETTER FILE*
**
2600 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO2600
2605 GOSUB766:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO2615
2610 I=I+1:GOTO2605
2615 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:AE$="Y"
2620 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO2620
2625 IFA8$(I)="****T"THENGOSUB210:GOTO26
35
2630 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO2625
2635 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO ADD MORE INFORMATION?[CD][CD]":PRINTW
1$
2640 GOSUB202:IFY$="Y"THENA8$(I)="":GOTO
2505
2645 GOTO2530
```

Commentary

The routine begins either at line 2500 or 2600 by opening a tape or disc file, and the total number of characters per line is requested (lines 2090–2110).

Once established, the maximum line length cannot be altered for the duration of the particular file being created or added to.

Lines 2000–2085, 2110, 2195 and **2515:** Deal with margin setting and resetting.

Lines 2115–2170: Produce a screen display indicating that the line entry can now begin and a visual reference as to which keys should be pressed if underline, bold, super- or sub-script control characters are to be included in the record line. These control characters will be interpreted at the time of printing the records and, depending on which type of printer is being used, underlining, etc., will be carried out.

As already explained in Subroutine 6 in Chapter 2, many dot matrix printers do not lend themselves to coping with underline or super- or subscript functions. Bold, however, can be accommodated by printing the line, suppressing the line feed on carriage return, and printing the stipulated text twice. It must not be assumed, however, that all dot matrix printers are limited in this way as I can only judge by those which I have had experience of. The full potential of the system can only be realised with a printer which can backspace, advance or reverse half a line feed under program control and has a special underline key. Most daisy wheel printers have these facilities and their purchase price is not far removed from that of a good quality matrix type. The screen display also informs the operator on how to access the setting of the left and righthand margins and how to close the record file.

Line 2170: Prints a series of dots to the screen to indicate the start and finish of a line entry, the number of dots printed being related to the margin settings.

Lines 2175–2200: Deal with the closing of the file (and the margin setting) and limits these to the first keyboard entry of a new line. In other words, it is not possible to close the file or set margins in the middle of a line.

Lines 2205–2330: Control the type of entry which is acceptable, maintain a running total of the number of keyboard entries allowing for underline, bold, etc., entries (ie the variable K) which do not figure in the maximum line length and the correction of wrong entries.

Lines 2220, 2230, 2235 and 2325 allow for incorrect entries using either the delete or cursor left keys. The characters are deleted to the point where corrections are to be made and the correct text entered.

Lines 2300, 2310 and 2505 constitute the inclusion of a bleep system to warn the operator that the end of the line being entered is five character entries away, just as one would expect to find on an ordinary typewriter.

The volume control on the television monitor must be on for this bleep to function.

Lines 2499–2645: Have a similar function to those in the standard Subroutine 10. Note the inclusion of lines 2509 and 2524 to disable the RUN/STOP key during line entry and to re-enable it again at the end of the file. It is good practice to anticipate this key being pressed accidentally, especially as it is adjacent to the SHIFT/LOCK key.

Tab settings and proportional spacing have not been included, but should not prove difficult if required. Certain printers can accommodate proportional spacing by adjusting the printer manually.

b. Printout and special functions

The objectives of this routine are as follows:

1) To print a letter or text, coping with special functions, such as underline, bold, super-script and sub-script.
2) To allow for different types of printer, eg dot matrix and daisy wheel types.
3) To bring in records from an address file and allow automatic letter addressing if required.

Start by combining the following subroutines together using the merge facility in Subroutine 1:

1) Subroutine 3
2) Subroutine 4
3) Subroutine 6
4) Subroutine 8t *or* 8d
5) Subroutine 9

The routine is entered at line 3370 and incorporates routines for dot matrix and daisy wheel printers. Lines 2995–3295 are concerned with daisy wheel printers and lines 3740–3785 with dot matrix types. Only the relative routine need be included if only one type of printer is to be used.

Amended listing

```
1 REM ***WORD PROCESSOR  COMMODORE 64***
5 W2$="[CLR][RVS]SUPER SECRETARY  PRINT-
OUT[SPC][SPC][SPC][SPC][SPC][SPC][SPC][S
PC][SPC][SPC][SPC][SPC][SPC][SPC][HOME][
CD]"
```

```
10 DIMA8$(500),A9$(500)
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1005 PRINT"[CD]1  TO PRINT A LETTER FILE
 AND LABELS."
1008 PRINT"[CD]2  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]3  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1009
1011 G=VAL(G$):IFG<1ORG>3THEN1009
1012 PRINTW2$:ONGGOTO3370,221,9999
2995 REM ***PRINT-OUT QUME & SCRIPTA***
3000 A$=LEFT$(BL$,3)+A$:L=LEN(A$):N=1
3005 G$="":G$=MID$(A$,N,1):G=ASC(G$)
3010 IFG=177THEN3020
3015 GOTO3045
3020 IFP8$="Y"THENP8$="N":P7$="N":GOTO30
30
3025 P7$="Y"
3030 IFP3$="P4"THEN3040
3035 PRINT#4,CHR$(27)CHR$(68);:GOTO3125:
REM **SUPERSCRIPT**
3040 PRINT#4,CHR$(27)CHR$(7);:GOTO3125
3045 IFG=178THEN3055
3050 GOTO3080
3055 IFP7$="Y"THENP7$="N":P8$="N":GOTO30
65
3060 P8$="Y"
3065 IFP3$="P4"THEN3075
3070 PRINT#4,CHR$(27)CHR$(85);:GOTO3125:
REM **SUBSCRIPT**
3075 PRINT#4,CHR$(27)CHR$(3);:GOTO3125
3080 IFG=91ANDP3$="P4"THENG$=CHR$(219):G
OTO3115:REM**[**
3085 IFG=191THEN3245:REM **UNDERLINE**
3090 IFG=166THEN3100
3095 GOTO3110
3100 IFP5$="Y"THENP5$="N":GOTO3125
3105 P5$="Y":GOTO3125
3110 IFG=13THENPRINT#4,CHR$(10):GOTO3255
3115 PRINT#4,G$;
3120 IFP5$="Y"THENPRINT#4,CHR$(8);G$;
3125 N=N+1:IFN=L+1THENPRINT#4,CHR$(10):G
```

```
OTO3255
3130 GOTO3005
3135 G$="":G$=MID$(A$,N,1):G=ASC(G$)
3140 IFG=177THENP7$="Y":P8$="N":GOTO3150
3145 GOTO3165
3150 IFP3$="P4"THEN3160
3155 PRINT#4,CHR$(27)CHR$(68);:GOTO3245:
REM **QUME SUPERSCRIPT**
3160 PRINT#4,CHR$(27)CHR$(7);:GOTO3245:R
EM **SCRIPTA SUPERSCRIPT**
3165 IFG=178THENP7$="N":P8$="Y":GOTO3175
3170 GOTO3190
3175 IFP3$="P4"THEN3185
3180 PRINT#4,CHR$(27)CHR$(85);:GOTO3245:
REM **QUME SUBSCRIPT**
3185 PRINT#4,CHR$(27)CHR$(3);:GOTO3245:R
EM **SCRIPTA SUBSCRIPT**
3190 IFG=91ANDP3$="P4"THENG$=CHR$(219):G
OTO3225:REM **[**
3195 IFG=166THEN3205
3200 GOTO3215
3205 IFP5$="Y"THENP5$="N":GOTO3245
3210 P5$="Y":GOTO3245
3215 IFG=191THEN3125:REM **END OF UNDERL
INE**
3220 IFG=13THENPRINT#4,CHR$(10):GOTO3255
3225 PRINT#4,G$;
3230 IFP5$="Y"THENPRINT#4,CHR$(8);G$;
3235 PRINT#4,CHR$(8)CHR$(95);
3240 IFP5$="Y"THENPRINT#4,CHR$(8)CHR$(95
);
3245 N=N+1:IFN=L+1THENPRINT#4,CHR$(10):G
OTO3255
3250 GOTO3135
3255 P5$="N":IFP7$="Y"THEN3270
3260 IFP8$="Y"THEN3285
3265 RETURN
3270 IFP3$="P4"THEN3280
3275 PRINT#4,CHR$(27)CHR$(85);:P7$="N":P
8$="N":RETURN
3280 PRINT#4,CHR$(27)CHR$(3);:P7$="N":P8
$="N":RETURN
3285 IFP3$="P4"THEN3295
3290 PRINT#4,CHR$(27)CHR$(68);:P7$="N":P
```

```
8$="N":RETURN
3295 PRINT#4,CHR$(27)CHR$(7);:P7$="N":P8
$="N":RETURN
3300 REM ***ADDRESS***
3305 OPEN4,4:A$=LEFT$(BL$,54)+"J. DOBERM
AN & CO. LTD.,":GOSUB470
3310 A$=LEFT$(BL$,54)+"22 LONDON ROAD,":
GOSUB470
3315 A$=LEFT$(BL$,54)+"COMPUTOWN,":GOSUB
470
3320 A$=LEFT$(BL$,54)+"BOOKSHIRE CP2 6PC
.":GOSUB470
3325 A$="[SPC]":GOSUB470
3330 A$=LEFT$(BL$,54)+"TELEPHONE 09 1234
567":GOSUB470
3335 A$=LEFT$(BL$,54)+"TELEX      98765":
GOSUB470
3340 A$="[SPC]":GOSUB470:CLOSE4:RETURN
3345 REM ***DATE INPUT***
3350 PRINTW2$:PRINT"[CD][CD][CD]TYPE IN
THE DATE.[CD][CD][CD]":PRINTW1$:PRINT"[C
D][CD]"
3355 L9=21:GOSUB150:PRINTW2$:DT$=LEFT$(B
L$,54)+D9$:RETURN
3360 OPEN4,4:A$=DT$:GOSUB470:A$="[SPC]":
GOSUB470:GOSUB470:CLOSE4:RETURN
3365 REM ***LETTER PRINT-OUT***
3370 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE THE ADDRESS SEQUENCE?[CD][CD]":PRINTW
1$
3375 GOSUB202:Q1$=Y$
3380 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE TO PRINT YOUR ADDRESS?[CD][CD]":PRINT
W1$
3385 GOSUB202:Q2$=Y$
3390 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO INCLUDE THE DATE?[CD][CD]":PRINTW1$
3395 GOSUB202:Q3$=Y$:IFQ3$="Y"THENGOSUB3
350
3400 IFQ1$="N"THEN3470
3405 PRINTW2$:PRINT"[CD][CD][CD]PLACE TH
E  ADDRESS[CD]"
3410 PRINT"DATA TAPE IN THE RECORDER.[CD
][CD][CD]":PRINTW1$:GOSUB210
```

```
3415 F9$="ADDRESS":GOSUB733:Z=1:IFER=1TH
ENER=0:GOTO3405
3420 A9$="":GOSUB770:A9$(Z)=A9$:A9$=""
3425 IFA9$(Z)="****T"THENGOSUB870:GOSUB2
10:GOSUB901:Z=1:GOTO3460
3430 L%=LEN(A9$(Z)):N=1:GG$=""
3435 G$="":G$=MID$(A9$(Z),N,1):G=ASC(G$)
3440 IFG=162ORG=172ORG=186THENG=G-128:G$
=CHR$(G):REM **CONVERT COMMA ETC**
3445 GG$=GG$+G$:N=N+1:IFN=L%+1THENA9$(Z)
=GG$:GG$="":GOTO3455
3450 GOTO3435
3455 Z=Z+1:GOTO3420
3460 PRINTW2$:PRINT"[CD][CD]PRINT-OUT OF
 ADDRESS LABELS ONLY?[CD][CD]":PRINTW1$
3465 GOSUB202:IFY$="Y"THENZ=1:GOTO3800
3470 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:IFER=1THENER=0:GOTO3470
3475 GOSUB516:GOSUB512:I=0:IFP4$="Y"THEN
3485
3480 GOSUB457:GOSUB506:GOSUB492:GOSUB502
3485 GOSUB216:IFY$="N"THENGOSUB869:GOTO3
790
3490 IFQ2$="Y"THENGOSUB3305
3495 IFQ3$="Y"THENGOSUB3360
3500 IFQ1$="N"THEN3640
3505 OPEN4,4:REM ***ADDRESS PRINT-OUT SE
QUENCE***
3510 B1$=LEFT$(A9$(Z),20):B2$=MID$(A9$(Z
),21,20):B3$=MID$(A9$(Z),41,20)
3515 B4$=MID$(A9$(Z),61,20):B5$=MID$(A9$
(Z),81,20):B6$=MID$(A9$(Z),101,20)
3520 B7$=MID$(A9$(Z),121,20)
3525 IFLEFT$(B2$,1)=CHR$(160)THENB2$="1"
:A$=B1$:GOTO3560
3530 L%=LEN(B2$):IFRIGHT$(B2$,1)=CHR$(16
0)THENB2$=LEFT$(B2$,L%-1):GOTO3530
3535 AA$=B2$+"[SPC]"+B1$
3540 L%=LEN(AA$):IFRIGHT$(AA$,1)=CHR$(16
0)THENAA$=LEFT$(AA$,L%-1):GOTO3540
3545 IFLEFT$(B3$,1)=CHR$(160)THENA$=AA$:
GOTO3560
3550 L%=LEN(B3$):IFRIGHT$(B3$,1)=CHR$(16
0)THENB3$=LEFT$(B3$,L%-1):GOTO3550
```

```
3555 A$=B2$+"[SPC]"+B3$+"[SPC]"+B1$
3560 PRINTA$:GOSUB470
3565 IFLEFT$(B4$,1)=CHR$(160)THEN3605
3570 A$=B4$:GOSUB470
3575 IFLEFT$(B5$,1)=CHR$(160)THEN3605
3580 A$=B5$:GOSUB470
3585 IFLEFT$(B6$,1)=CHR$(160)THEN3605
3590 A$=B6$:GOSUB470
3595 IFLEFT$(B7$,1)=CHR$(160)THEN3605
3600 A$=B7$:GOSUB470
3605 A$="[SPC]":GOSUB470:GOSUB470:A$="":
Z=Z+1:IFLA$="Y"THEN3805
3610 IFB2$="1"THENB2$="":A$="DEAR SIRS,"
:GOTO3620
3615 A$="DEAR"+"[SPC]"+AA$+",":AA$=""
3620 PRINTA$:GOSUB470:A$="[SPC]":GOSUB47
0:A$="":CLOSE4:GOTO3640
3625 CLOSE4:GOSUB516:GOSUB512:I=0:IFP4$=
"Y"THEN3635
3630 GOSUB506:GOSUB492:GOSUB502
3635 GOSUB216:IFY$="N"THENGOSUB869:RUN3
3640 OPEN4,4
3645 IFI=>TTTHENGOSUB210:GOTO3625
3650 GOSUB766:I=I+1
3655 IFA8$="****T"THENGOSUB869:CLOSE4:PR
INT:GOSUB210:GOTO3770
3660 IFP4$="Y"THEN3645
3665 IFP3$="P1"ORP3$="P2"ORP3$="P3"ORP3$
="P5"THEN3740
3670 IFP3$="P6"THENL=LEN(A8$):N=1:GOTO36
80
3675 A$=A8$:A8$="":GOTO3705
3680 G$="":G$=MID$(A8$,N,1):G=ASC(G$)
3685 IFG>192ANDG<219THENG=G-128:G$=CHR$(
G):GOTO3695
3690 IFG>64ANDG<91THENG=G+128:G$=CHR$(G)
3695 A$=A$+G$:N=N+1:IFN=L+1THEN3705
3700 GOTO3680
3705 GOSUB3000:IFP6$="Y"THEN3715
3710 GOTO3735
3715 IFP3$="P4"THEN3725
3720 PRINT#4,CHR$(27)CHR$(68)CHR$(27)CHR
$(68);:GOTO3730
3725 PRINT#4,CHR$(27)CHR$(7)CHR$(27)CHR$
```

```
(7);
3730 GOSUB3000
3735 A$="":GOSUB522:GOTO3645
3740 L=LEN(A8$):N=1:REM **DOT MATRIX PRI
NTERS AND OTHERS UNSPECIFIED**
3745 G$="":G$=MID$(A8$,N,1):G=ASC(G$)
3750 IFG=162ORG=172ORG=186THENG=G-128:G$
=CHR$(G):GOTO3760
3755 IFG=166ORG=177ORG=178ORG=191THENN=N
+1:GOTO3745
3760 A$=A$+G$:N=N+1:IFN=L+1THENGOSUB470:
I=I-1:A8$="":A$="":GOTO3645
3765 GOTO3745
3770 GOSUB901:IFA9$(Z)="****T"THEN3790
3775 GOSUB226:IFY$="N"THEN3790
3780 GOSUB730:IFER=1THENER=0:GOTO3780
3785 GOTO3475
3790 PRINTW2$:PRINT"DO YOU REQUIRE ADDRE
SS LABELS?":PRINTW1$:REM **LABEL PRINT**
3795 GOSUB202:Z=1:IFY$="N"THENRUN3
3800 GOSUB457:OPEN4,4
3805 GOSUB210:IFA9$(Z)="****T"THENCLOSE4
:RUN3
3810 LA$="Y":GOTO3510
```

Commentary

Lines 2295–3295: Deal with two types of daisy wheel printer (Qume and Scripta). This routine can be modified to suit any machine by consulting the printer manual and establishing the following control codes:

1) Backspace, eg Qume and Scripta = PRINT#4, CHR$(8).
2) Underline key, eg Qume and Scripta = PRINT#4,CHR$(95).
3) Reverse half line feed, eg Scripta = PRINT#4,CHR$(27)CHR$(7).
   Qume = PRINT#4,CHR$(27)CHR$(68).
4) Advance half line feed, eg Scripta = PRINT#4,CHR$(27)CHR$(3).
   Qume = PRINT#4,CHR$(27)CHR$(85).

One curious point regarding the Scripta was that the [bracket, when entered at the keyboard, was printed out as a fraction and, to obtain the correct character, the ASCII code had to be changed from 91 to 219. Different print wheels, which are available for most printers, can give different results and it is well worthwhile establishing the characteristics of each one.

Lines 3370–3375: If required, will access the address file produced by the address program detailed in the second section of this chapter. This is a useful function if the same letter or information text is to be individually addressed and sent to a number of different people or companies. It does away with photocopies which would have to be individually addressed or having to type the same letter repeatedly.

Address labels can also be produced from the same file by placing sheets or rolls of self-adhesive labels in the printer. The exact spacing, between labels and the printing, would have to be established by trial and error and adjustments made in lines 3790–3810 and lines 3510–3605. Samples of a standard letter for three different destinations are included at the end of the chapter as an example of what can easily be achieved. The same letter could, of course, have been produced for any number of people.

Lines 3380–3385: Request whether you require your address to be printed, (lines 3305–3340). The address included is entirely fictitious and your own address, etc., should be substituted. If headed paper is to be used, then the answer to the question would be 'no', and the examples at the end of the chapter indicate the two methods.

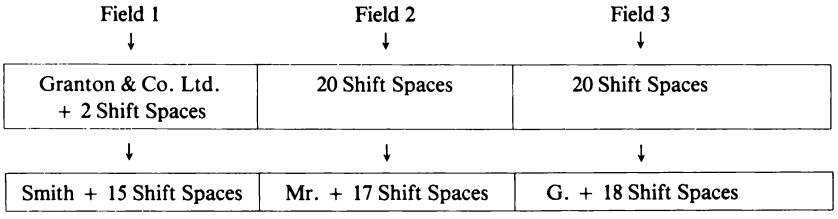
Lines 3390–3395: Deal with the inclusion of the date; its entry is carried out in lines 3345–3355.

Once your address and the date have been printed, the next procedure is the preparation of the address, in the correct format for printing. There are two distinct types of address to be dealt with: one is a company name and the other the name of an individual. The Address File Construct program initially requests the surname or the company name, which is placed in a record field of 20 characters, the balance being made up of shifted spaces. The second request is for the person's title, followed by the initials, each field again being balanced to 20 characters by shift spaces. In the case of a company, the second and third fields are entirely composed of 20 shift spaces each, the title and initials are not required. The whole record is composed of eight 20-character fields which include the address and telephone number. The first three fields of record 1, relating to the firm 'Granton & Co. Ltd' and the second record relating to 'Mr. G. Smith' would appear as in **Figure 4.1**.

Lines 3505–3620: Deal with the manipulation of the three fields to produce an acceptable format, as in the case of Smith Mr. G. to become Mr. G. Smith, followed by his address. The re-arranging also allows the start of the letter to read 'Dear Mr. Smith,' (lines 3615–3620): in the case of the firm the letter will start 'Dear Sirs,' (lines 3610–3620).

Lines 3625–3735: Relate to the letter/text records being handled, dealing

Figure 4.1



with the number of lines per page, spaces between lines and stencil-cutting mode, by referring to the standard printer Subroutine 6. Printer type allocation and screen printout only questions are dealt with, too, and an added advantage is the ability to run through the file to select specified lines for printing.

Lines 3740–3785: Relate to dot matrix printers. The control codes for underlining, etc., are removed in line 3755.

Lines 3305, 3360, 3480 and **3575:** Deal with opening a channel to a printer, but, in the lines indicated, the addition of ,7 to the normal OPEN4,4 is used specifically to place the Commodore 1515 printer in lower case mode and should not be included with other printer types. The inclusion of the ,7 solves the problems which arise when printing quotation marks. Line 481 should be altered to read 481 PRINT#4,A$:GOTO489 (the CHR$(17) can be dropped). When the 1515 is requested to print a long record, it can sometimes stop printing at the beginning of a new line and the whole process grinds to a halt. The answer to this problem is (under program control) to switch off the screen prior to printing and switch it back on again afterwards. Line 481 should now read 481POKE53265,11: PRINT#4, A$:POKE53265,27:GOTO489.

c. Paragraph/Chapter Construct

The main objective of this program is to facilitate the building up of a file of data records by accessing other files and extracting selected records.

Start by combining the following subroutines using the merge facility in Subroutine 1:

1) Subroutine 3
2) Subroutine 4
3) Subroutine 8t *or* 8d
4) Subroutine 9

Amended listing

```
1 REM ***PARAGRAPH/CHAPTER CONSTRUCT PRO
GRAM COMMODORE 64.***
5 W2$="[CLR][RVS]PARAGRAPH/CHAPTER CONST
RUCT PROGRAM.[SPC][SPC][SPC][SPC][HOME][
CD]"
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO CONSTRUCT A NEW DAT
A FILE."
1004 PRINT"[CD]2  FOR SCREEN DISPLAY OF
DATA FILE."
1008 PRINT"[CD]3  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]4  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>4THEN1010
1012 PRINTW2$:ONGGOTO4500,891,221,9999
3985 REM ######################
3990 REM ***CREATE NEW FILE***
3995 REM ######################
4000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
4005 PRINT"[CD]1  TO CONSTRUCT THE FILE
BY STIPULATED     SECTIONS."
4010 PRINT"[CD]2  TO CONSTRUCT THE FILE
LINE BY LINE."
4015 PRINT"[CD]3  FOR NEW FILE TO BE INC
LUDED    OR       END.":PRINTW1$
4020 G$="":GETG$:IFG$=""THEN4020
4025 G=VAL(G$):IFG<1ORG>3THEN4020
4030 PRINTW2$:ONGGOTO4040,4100,4035
4035 RETURN
4040 PRINT"[RVS]LINE NUMBERS TO BE INCLU
DED.[RVO]":PRINTW1$
4045 PRINT"[CD][CD]TYPE IN FIRST LINE NU
MBER.[CD][CD]":PRINTW1$
4050 L9=4:GOSUB150:K1$=D9$:D9$=""
4055 PRINTW2$:PRINT"[RVS]LINE NUMBERS TO
 BE INCLUDED.[RVO]":PRINTW1$
4060 PRINT"[CD][CD]TYPE IN LAST LINE NUM
BER.[CD][CD]":PRINTW1$
```

```
4065 L9=4:GOSUB150:K2$=D9$:D9$="":K=1
4070 GOSUB770:IFA9$="****T"THENRETURN
4075 IFK=VAL(K1$)THENPRINTW2$:PRINT"INCL
UDED LINES.":PRINTW1$:PRINTA9$:GOTO4095
4080 K=K+1:GOTO4070
4085 GOSUB770:IFA9$="****T"THENPRINTA9$:
A9$="":GOSUB210:K=1:RETURN
4090 IFK=VAL(K2$)THENA8$(I)=A9$:A9$="":I
=I+1:K=1:GOSUB210:GOTO4000
4095 A8$(I)=A9$:A9$="":I=I+1:K=K+1:GOTO4
085
4100 PRINTW2$:PRINT"[CD][CD]WHICH LINE N
UMBER DO YOU WISH TO START?[CD][CD]":PRI
NTW1$:K=1
4105 L9=4:GOSUB150:K1$=D9$:D9$=""
4110 GOSUB770:IFA9$="****T"THENRETURN
4115 IFK=VAL(K1$)THENK=1:K1$="":GOTO4125
4120 K=K+1:GOTO4110
4125 PRINTW2$:PRINT"TOTAL NUMBER OF LINE
S:- ";I-1:PRINTW1$
4130 PRINT"[CD][CD]":PRINTA9$:PRINT"[CD]
[CD]":PRINTW1$
4135 PRINT"SPACE BAR TO INCLUDE ABOVE LI
NE.[CD]"
4140 PRINT"  F7 KEY  TO REJECT  ABOVE LI
NE.[CD]"
4145 PRINT"  F1 KEY  TO RETURN TO CONSTR
UCT OPTION"
4150 G$="":GETG$:IFG$=""THEN4150
4155 G=ASC(G$):IFG=32ORG=160THEN4175
4160 IFG=136ORG=140THENA9$="":GOSUB770:G
OTO4125
4165 IFG=133ORG=137THEN4000
4170 GOTO4150
4175 A8$(I)=A9$:A9$="":I=I+1:GOSUB770:GO
TO4125
4495 REM ***START A NEW DATA FILE***
4500 I=1
4505 GOSUB823:GOSUB703:F9$=D9$:D9$="":GO
SUB733:IFER=1THENER=0:GOTO4505
4510 GOSUB4000:GOSUB870:GOSUB901
4515 PRINTW2$:PRINT"[CD][CD][CD]TOTAL NU
MBER OF LINES:- ";I-1:PRINTW1$
4520 PRINT"[CD][CD]DO YOU WISH TO ADD MO
```

```
RE INFORMATION?[CD][CD]":PRINTW1$
4525 GOSUB202:IFY$="Y"THEN4505
4530 A8$(I)="****T":I=1
4535 GOSUB823:GOSUB701:F8$=D9$:D9$="":GO
SUB710:IFER=1THENER=0:GOTO4535
4540 A8$=A8$(I):GOSUB792:IFA8$="****T"TH
ENGOSUB869:GOSUB901:GOSUB840:RUN3
4545 I=I+1:GOTO4540
```

Commentary

The principle of this routine is straightforward. Initially the operator is requested to name the first file for consideration and a choice of construction options for composing a new chapter/paragraph/file are offered, (lines 4000–4035). Option 1 allows a defined number of lines to form all of, or part of, the new file, followed by a return to the option sequence. A further number of defined records can be included, or options 2 or 3 can be chosen.

Option 2 allows the operator to run through a file, line by line, picking out records for inclusion and discarding those which are not required (lines 4100–4175). The chance to return to the option sequence is given at each stage. Rather than commence construction at record or line 1 of a file, the operator can choose which record number from which to start the selection. (N.B. Records are stored on tape/disc sequentially and, to include records which have already been passed over, it would be necessary to return to option 3, re-open the file and then include these records.)

Option 3 (lines 4515–4545) will allow the operator either to indicate that the sequence is finished and that a new file containing all the records chosen can now be saved on tape/disc, or to give the choice to carry on the construction with another file. The next file to be accessed could, of course, be the same file as already dealt with, if, for example, certain specified records from the same file are to be re-positioned after the ones already picked out.

To sum up the routine, the following two examples should give a clear picture as to its capabilities:

a) Consider a file containing 10 paragraphs where a re-shuffling of the paragraphs is necessary. Paragraph 5, for example, could be removed from its present position and placed between paragraphs 7 and 8.

b) A new file to be composed of paragraphs 1, 3 and 6 from file A, the whole of file B and the last two paragraphs of file C can easily be constructed by this routine.

2. Address File Construct

The Address File Construct program listed below will enable a file of names and addresses and telephone numbers of companies or individuals to be created for use by the word processing system. Two other facilities are offered in Chapter 6 for the manipulation of this file: the placing of the records in alphabetic order and a search routine to locate and isolate all records of clients, etc., by their surname, eg all the Smiths contained in the file or even anyone whose surname begins with S. The aims and objectives of the program are as follows:

1) To create a new file of names and addresses and telephone numbers.
2) To add further names, etc., to an existing file.
3) To provide a means of correcting or editing the file.
4) To allow the file to be displayed on to the monitor screen or printed out.

Subroutines to be included are:

1) Subroutine 1
2) Subroutine 3
3) Subroutine 4
4) Subroutine 5
5) Subroutine 6
6) Subroutine 8t *or* 8d
7) Subroutine 9
8) Subroutine 10

Amended listing

```
1 REM ***SUB 1. ADDRESS.PROG.  COMMODORE
 64***
5 W2$="[CLR][RVS]ADDRESS PROGRAM.[SPC][S
PC] [SPC][SPC][SPC][SPC][SPC][SPC][SPC][
SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC][
SPC][SPC][SPC][SPC][SPC][SPC][HOME][CD]"
10 DIMA8$(999),A9$(999)
301 I=1:P4=160:REM ***MAXIMUM STRING LEN
GTH***
357 W5$="[RVS]NAME.                    TITLE.
               INITIAL.                "
358 W6$="[RVS]ADDRESS 1.               ADDRES
S 2.           ADDRESS 3.              "
359 W7$="[RVS]ADDRESS 4.               TELEPH
```

```
ONE NUMBER.    "
360 W5$=W5$+W6$+W7$
367 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD]";W5$:PRINT
702 D9$="ADDRESS":RETURN
704 D9$="ADDRESS":RETURN
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO START A NEW ADDRESS
 FILE."
1002 PRINT"[CD]2  TO ADD NEW DATA TO ADD
RESS FILE."
1003 PRINT"[CD]3  TO EDIT ADDRESS FILE."
1004 PRINT"[CD]4  FOR SCREEN DISPLAY OF
DATA FILE."
1005 PRINT"[CD]5  TO PRINT ADDRESS FILE.
"
1008 PRINT"[CD]6  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]7  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>7THEN1010
1012 PRINTW2$:ONGGOTO2500,2600,300,891,5
33,221,9999
1999 PRINTW2$:PRINT"DATA ENTRY NUMBER:-
 ";K:PRINTW1$:RETURN
2000 K=1:GOSUB1999:PRINT"[CD]TYPE IN NAM
E OF FIRM OR SURNAME.[CD]"
2005 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2010 PRINT"#  SIGN TO FINISH ADDDING DAT
A.[CD][CD]"
2015 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2020 IFD9$="#"THEND9$="****T":RETURN
2025 B1$=D9$:L%=LEN(B1$):B1$=B1$+LEFT$(B
L$,20-L%):K=K+1
2030 GOSUB1999:PRINT"[CD]TYPE IN TITLE O
F PERSON MR. MRS.ETC[CD]"
2035 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
```

```
2040 PRINT"#  SIGN IF DEALING WITH A FIR
M.[CD][CD]"
2045 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2050 IFD9$="#"THENB2$=LEFT$(BL$,20):B3$=
LEFT$(BL$,20):K=K+2:GOTO2090
2055 B2$=D9$:L%=LEN(B2$):B2$=B2$+LEFT$(B
L$,20-L%):K=K+1
2060 GOSUB1999:PRINT"[CD]TYPE IN INITIAL
S OF PERSON.[CD]"
2065 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2070 PRINT"#  SIGN IF NOT KNOWN.[CD][CD]
"
2075 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2080 IFD9$="#"THENB3$=LEFT$(BL$,20):GOTO
2090:K=K+1
2085 B3$=D9$:L%=LEN(B3$):B3$=B3$+LEFT$(B
L$,20-L%):K=K+1
2090 GOSUB1999:PRINT"[CD]TYPE IN 1ST LIN
E OF ADDRESS[CD]"
2095 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2100 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2105 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2110 IFD9$="#"THEN2116
2115 B4$=D9$:L%=LEN(B4$):B4$=B4$+LEFT$(B
L$,20-L%):K=K+1:GOTO2120
2116 B4$=LEFT$(BL$,20):B5$=LEFT$(BL$,20)
2117 B6$=LEFT$(BL$,20):B7$=LEFT$(BL$,20)
:K=K+4:GOTO2210
2120 GOSUB1999:PRINT"[CD]TYPE IN 2ND LIN
E OF ADDRESS[CD]"
2125 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2130 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2135 PRINT"FOLLOWED BY RETURN[CD]":PRINT
```

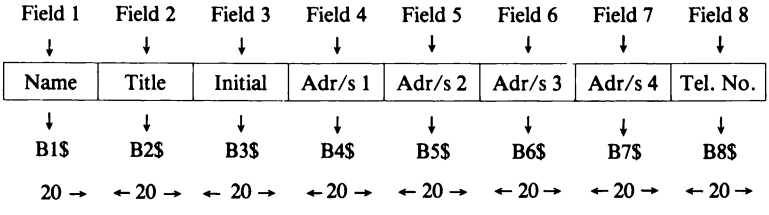
```
W1$:L9=19:GOSUB150
2140 IFD9$="#"THEN2146
2145 B5$=D9$:L%=LEN(B5$):B5$=B5$+LEFT$(B
L$,20-L%):K=K+1:GOTO2150
2146 B5$=LEFT$(BL$,20):B6$=LEFT$(BL$,20)
:B7$=LEFT$(BL$,20):K=K+3:GOTO2210
2150 GOSUB1999:PRINT"[CD]TYPE IN 3RD LIN
E OF ADDRESS[CD]"
2155 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2160 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2165 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2170 IFD9$="#"THENB6$=LEFT$(BL$,20):B7$=
LEFT$(BL$,20):K=K+2:GOTO2210
2175 B6$=D9$:L%=LEN(B6$):B6$=B6$+LEFT$(B
L$,20-L%):K=K+1
2180 GOSUB1999:PRINT"[CD]TYPE IN 4TH LIN
E OF ADDRESS[CD]"
2185 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2190 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2195 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=19:GOSUB150
2200 IFD9$="#"THENB7$=LEFT$(BL$,20):K=K+
1:GOTO2210
2205 B7$=D9$:L%=LEN(B7$):B7$=B7$+LEFT$(B
L$,20-L%):K=K+1
2210 GOSUB1999:PRINT"[CD]TYPE IN TELEPHO
NE NUMBER.[CD]"
2215 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2220 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2225 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=10:GOSUB150
2230 IFD9$="#"THENB8$=LEFT$(BL$,20):GOTO
2240
2235 B8$=D9$:L%=LEN(B8$):B8$=B8$+LEFT$(B
```

```
L$,20-L%)
2240 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]":PRINTB1$:PRINTB2$:PRINTB3
$
2245 PRINTB4$:PRINTB5$:PRINTB6$:PRINTB7$
:PRINTB8$:PRINTW1$:GOSUB200
2250 IFY$="N"THEN2490
2450 A8$(I)=B1$+B2$+B3$+B4$+B5$+B6$+B7$+
B8$
2460 RETURN
2490 B1$="":B2$="":B3$="":B4$="":B5$="":
B6$="":B7$="":B8$="":GOTO2000
2499 REM ***START A NEW FILE***
2500 I=1:GOSUB823:GOSUB701:F8$=D9$:D9$="
":GOSUB710:IFER=1THENER=0:GOTO2500
2505 GOSUB2000:IFD9$="****T"THEN2525
2510 A8$=A8$(I):GOSUB792:B1$="":B2$="":B
3$="":B4$="":B5$="":B6$=""
2515 B7$="":B8$="":GOTO2505
2525 GOSUB218:IFY$="N"THEND9$="":GOTO250
5
2530 A8$(I)="****T":A8$=A8$(I):GOSUB792:
GOSUB869:GOSUB901:IFAE$="Y"THENGOSUB879
2535 GOSUB840:RUN3
2599 REM ***ADD NEW DATA TO FILE***
2600 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO2600
2605 GOSUB766:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO2615
2610 I=I+1:GOTO2605
2615 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:AE$="Y"
2620 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO2620
2625 IFA8$(I)="****T"THENGOSUB210:GOTO26
35
2630 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO2625
2635 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO ADD MORE INFORMATION?[CD][CD]":PRINTW
1$
2640 GOSUB202:IFY$="Y"THENA8$(I)="":GOTO
2505
2645 GOTO2530:REM **CLOSE FILE**
```

Commentary

Each record produced will consist of eight fields of 20 characters, making a total record of 160 characters. Line 301 of the Edit routine is therefore altered accordingly. The inclusion of lines 357–360 and the altered line 367 will give a visual guide on field parameters to the operator when editing. As this is a dedicated program to produce an address file, the file name can therefore be 'address' and lines 702 and 704 and Subroutine 8t or 8d altered as listed. If this file title should be altered in any way, a corresponding change would be necessary in the word processing program, ie line 3415. The modified Subroutine 10 will produce a file of address records, as shown in **Figure 4.2.**

Figure 4.2



Allowing a 20-character field for the title and initials is far more than is likely to be required and may seem wasteful when considering storage space on tape or disc. These field sizes can be reduced and the word processing program modified to suit. The field sizes were chosen for simplicity and ease in calculating string field handling, but the telephone number field size is deliberate, as this routine will be modified slightly to suit the file creation Mailing List program in Chapter 6. Three names and addresses have been used to illustrate the full programs from both sections, and three letters are printed at the end of this chapter. With the following modification to the printer routine, the three records are shown below.

```
543 GOSUB766:IFA8$ = "****T"THEN545
544 A$ = LEFT$(A8$,80):GOSUB470:A$ = MID$(A8$,81):GOSUB
    470:A$ = "[SPC]":GOSUB470:GOTO542
```

This routine could be modified to produce a useful directory of names, addresses and telephone numbers in the correct format.

Smith	Mr.	G.	76, Brighton Road
Rington	Blinkshire, RI1 2LZ		01 1111111
Granton & Co. Ltd.			Brickworks Street
Industrial Estate	Shington	Kremshire SH2 6ZL	01 2222222
Black	Mrs.	J.	'Dunroamin'
6, Provost Wynd	Tremingly	Tripshire, TR6 7SS	01 3333333

15th Au9ust 1983.

Mr. G. Smith
76, Bri9hton Road
Rin9ton
Blinkshire, RI1 2LZ

Dear Mr. Smith,

I am haPPy to inform you that we have now obtained a new source of suPPly of the TX Domestic Water Heater and are in a Position to deliver by return on receiPt of your order. I must aPolo9ise for the delay, but as our ori9inal suPPlier failed to meet their commitment to us, it has taken time to find a new source of suPPly.

Our aPolo9ies for any inconvenience caused.

Yours sincerely,

J. Doberman.
(Mana9in9 Director).

15th Au9ust 1983.

Granton & Co. Ltd.
Brickworks Street
Industrial Estate
Shin9ton
Kremshire SH2 6ZL

Dear Sirs,

I am haPPy to inform you that we have now obtained a new source of suPPly of the TX Domestic Water Heater and are in a Position to deliver by return on receiPt of your order. I must aPolo9ise for the delay, but as our ori9inal suPPlier failed to meet their commitment to us, it has taken time to find a new source of suPPly.

Our aPolo9ies for any inconvenience caused.

Yours sincerely,

J. Doberman.
(Mana9in9 Director).

J. Doberman & Co. Ltd.,
22 London Road,
Computown,
Bookshire CP2 6PC.

Telephone 09 1234567
Telex 98765

15th August 1983.

Mrs. J. Black
'Dunroamin'
6, Provost Wynd
Tremingly
Tripshire, TR6 7SS

Dear Mrs. Black,

I am happy to inform you that we have now obtained a new source of supply of the TX Domestic Water Heater and are in a position to deliver by return on receipt of your order. I must apologise for the delay, but as our original supplier failed to meet their commitment to us, it has taken time to find a new source of supply.

Our apologies for any inconvenience caused.

Yours sincerely,

J. Doberman.
(Managing Director).

Mr. G. Smith
76, Brighton Road
Rington
Blinkshire, RI1 2LZ

Granton & Co. Ltd.
Brickworks Street
Industrial Estate
Shington
Kremshire SH2 6ZL

Mrs. J. Black
'Dunroamin'
6, Provost Wynd
Tremingly
Tripshire, TR6 7SS

CHAPTER 5
Equipment Inventory

The purpose of this program is to create a tape/disc file of items of equipment which can be manipulated to produce various lists. These can be included in the accounts system in Chapter 3 as part of a firm's assets. The objectives of the system are:

1) To create a file which will include a description of the equipment, the serial number, room situation, purchase date, cost and value today. This can be based on replacement cost or its re-saleable value.
2) To update the file, eg alter the valuation, alter the room situation, delete equipment or add new purchases.
3) To produce a master list in printed form.
4) Produce printed sheets of equipment listed under their room numbers.
5) Provide file-search facilities for specified items and a listing of all equipment purchased in a given year.
6) Production of a file, the value of which can be included in an accounts system as part of a firm's assets.
7) The placing of the equipment in alphabetic order.

Objective 6 has already been covered in Chapter 3.

1. Equipment File

This section deals with the setting up of the file, file editing, file printout and the inclusion of new purchases. The following subroutines form the basis of the program and should be combined using the merge facility in Subroutine 1.

1) Subroutine 2
2) Subroutine 3a
3) Subroutine 4
4) Subroutine 5
5) Subroutine 6
6) Subroutine 7
7) Subroutine 8t *or* 8d
8) Subroutine 9
9) Subroutine 10

Subroutine 10 has been included initially to aid program entry, but the final program should contain only those lines of this subroutine (from line 2499 forwards) listed below.

Amended listing

```
1 REM ***EQUIPMENT FILE.  COMMODORE 64**
*
5 W2$="[CLR][RVS]EQUIPMENT FILE.
                    [HOME][CD]"
10 DIMA8$(999),A9$(999)
301 I=1:P4=80:REM ***MAXIMUM STRING LENG
TH***
315 GOSUB783:NL=NL+1
357 W5$="[RVS]EQUIPMENT                  S
ER.NO.   ROOM DATE       INITIAL COST   "
358 W6$="VALUE           "
360 W5$=W5$+W6$
367 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD]";W5$:PRINT
702 D9$="EQUIPMENT":RETURN
704 D9$="EQUIPMENT":RETURN
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  TO START A NEW EQUIPME
NT FILE."
1002 PRINT"[CD]2  TO ADD NEW DATA TO EQU
IPMENT FILE."
1003 PRINT"[CD]3  TO EDIT EQUIPMENT FILE
."
1004 PRINT"[CD]4  FOR SCREEN DISPLAY OF
DATA FILE."
1005 PRINT"[CD]5  TO PRINT EQUIPMENT FIL
E."
1006 PRINT"[CD]6  FOR EQUIPMENT ANALYSIS
 ROUTINE."
1008 PRINT"[CD]7  RETURN TO MENU PROGRAM
."
1009 PRINT"[CD]8  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>8THEN1010
1012 PRINTW2$:ONGGOTO2500,2600,300,891,5
33,3000,221,9999
```

```
1996 REM ##################################
######################################
1997 REM ***SUB 10. CREATE NEW FILE OR A
DD TO EXISTING FILE.  COMMODORE 64***
1998 REM ##################################
######################################
1999 PRINTW2$:PRINT"DATA ENTRY NUMBER:-
 ";K:PRINTW1$:RETURN
2000 K=1:GOSUB1999:PRINT"[CD]TYPE IN DES
CRIPTION OF EQUIPMENT.[CD]"
2005 PRINT"                        OR[CD]"
2010 PRINT"#  SIGN TO FINISH ADDDING DAT
A.[CD][CD]"
2015 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=24:GOSUB150
2020 IFD9$="#"THEND9$="****T":RETURN
2025 B1$=D9$:L%=LEN(B1$):B1$=B1$+LEFT$(B
L$,25-L%):K=K+1
2030 GOSUB1999:PRINT"[CD]TYPE IN EQUIPME
NT SERIAL NUMBER.[CD]"
2035 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC]OR[CD]"
2040 PRINT"#  SIGN IF NONE AVAILABLE.[CD
][CD]"
2045 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=9:GOSUB150
2050 IFD9$="#"THENB2$=LEFT$(BL$,1)+"0000
00000":K=K+1:GOTO2060
2055 B2$=D9$:L%=LEN(B2$):B2$=B2$+LEFT$(B
L$,10-L%):K=K+1
2060 GOSUB1999:PRINT"[CD]TYPE IN THE ROO
M NUMBER"
2061 PRINT"[CD]THE EQUIPMENT IS SITUATED
.[CD]"
2065 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC]OR[CD]"
2070 PRINT"#  SIGN IF NOT KNOWN.[CD][CD]
"
2075 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=4:GOSUB150
2080 IFD9$="#"THENB3$=LEFT$(BL$,1)+"0.00
":GOTO2090:K=K+1
```

```
2085 B3$=D9$:L%=LEN(B3$):B3$=B3$+LEFT$(B
L$,5-L%):K=K+1
2090 GOSUB1999:PRINT"[CD]TYPE IN DATE PU
RCHASED"
2091 PRINT"[CD]E.G. 12/7/83[CD]"
2095 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2100 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.[CD][CD]"
2105 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=8:GOSUB150
2110 IFD9$="#"THENB4$=LEFT$(BL$,10):K=K+
1:GOTO2120
2115 B4$=D9$:L%=LEN(B4$):B4$=LEFT$(BL$,1
0-L%)+B4$:K=K+1
2120 GOSUB1999:PRINT"[CD]TYPE IN TOTAL C
OST OF EQUIPMENT.[CD]"
2135 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=14:GOSUB150:N1$=D9$:GOSUB607
2145 B5$=N1$:L%=LEN(B5$):B5$=LEFT$(BL$,1
5-L%)+B5$:K=K+1
2150 GOSUB1999:PRINT"[CD]TYPE IN TO-DAY'
S VALUE.[CD]"
2155 PRINT"[SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SPC
][SPC][SPC][SPC][SPC][SPC]OR[CD]"
2160 PRINT"#  SIGN FOR NOT KNOWN OR THE
SAME AS"
2161 PRINT"TO-DAY'S VALUE.[CD][CD]"
2165 PRINT"FOLLOWED BY RETURN[CD]":PRINT
W1$:L9=14:GOSUB150
2170 IFD9$="#"THENB6$=B5$:GOTO2240
2175 N1$=D9$:GOSUB607:B6$=N1$:L%=LEN(B6$
):B6$=LEFT$(BL$,15-L%)+B6$
2240 PRINTW2$:PRINT"[CD]PLEASE CHECK YOU
R ENTRIES[CD]":PRINTB1$:PRINTB2$:PRINTB3
$
2245 PRINTB4$:PRINTB5$:PRINTB6$:PRINTW1$
:GOSUB200
2250 IFY$="N"THEN2490
2450 A8$(I)=B1$+B2$+B3$+B4$+B5$+B6$
2460 RETURN
2490 B1$="":B2$="":B3$="":B4$="":B5$="":
```

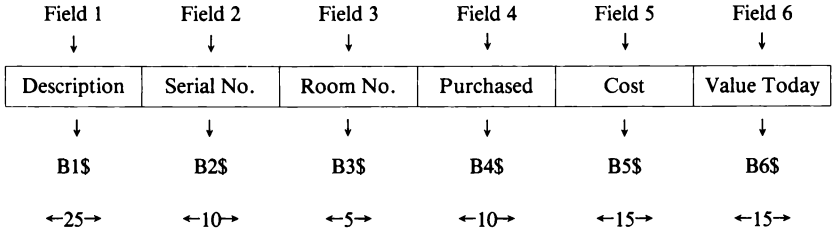
```
B6$="":GOTO2000
2499 REM ***START A NEW FILE***
2500 I=1:GOSUB823:GOSUB701:F8$=D9$:D9$="
":GOSUB710:IFER=1THENER=0:GOTO2500
2505 GOSUB2000:IFD9$="****T"THEN2525
2510 A8$=A8$(I):GOSUB792:B1$="":B2$="":B
3$="":B4$="":B5$="":B6$=""
2515 GOTO2505
2525 GOSUB218:IFY$="N"THEND9$="":GOTO250
5
2530 A8$(I)="****T":A8$=A8$(I):GOSUB792:
GOSUB869:GOSUB901:IFAE$="Y"THENGOSUB879
2535 GOSUB840:RUN3
2599 REM ***ADD NEW DATA TO FILE***
2600 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO2600
2605 GOSUB766:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB210:GOTO2615
2610 I=I+1:GOTO2605
2615 GOSUB901:FF$="TEMP":FL$=F8$:GOSUB87
2:AE$="Y"
2620 I=1:GOSUB823:GOSUB710:IFER=1THENER=
0:GOTO2620
2625 IFA8$(I)="****T"THENGOSUB210:GOTO26
35
2630 A8$=A8$(I):GOSUB792:A8$="":I=I+1:GO
TO2625
2635 PRINTW2$:PRINT"[CD][CD]DO YOU WISH
TO ADD MORE INFORMATION?[CD][CD]":PRINTW
1$
2640 GOSUB202:IFY$="Y"THENA8$(I)="":GOTO
2505
2645 GOTO2530:REM **CLOSE FILE**
```

You should note that the file records can be recalled using the INPUT# command. Each record will be made up of six fields, as shown in **Figure 5.1**.

2. Analysing the Inventory

This section deals with the analysis of the records. It can be added to the previous program or, if a large equipment file is envisaged, it might be

Figure 5.1



prudent to construct a second program. The subroutines to be included would therefore be:

1) Subroutine 1
2) Subroutine 3a
3) Subroutine 4
4) Subroutine 8t *or* 8d
5) Subroutine 9

Amended listing

```
2999 REM ***EQUIPMENT ANALYSIS PROGRAM**
*
3000 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:I=1:IFER=1THENER=0:GOTO3000
3005 GOSUB781:A8$(I)=A8$:A8$="":IFA8$(I)
="****T"THENGOSUB869:GOSUB901:GOTO3020
3010 I=I+1:GOTO3005
3020 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
3022 PRINT"[CD]1  TO SEARCH FOR EQUIPMEN
T BY NAME."
3024 PRINT"[CD]2  TO SEARCH FOR EQUIPMEN
T BY ROOM NO."
3026 PRINT"[CD]3  TO SEARCH FOR EQUIPMEN
T BY YEAR."
3028 PRINT"[CD]4  TO END PROGRAM.":PRINT
W1$
3030 G$="":GETG$:IFG$=""THEN3030
3032 G=VAL(G$):IFG<1ORG>4THEN3030
3034 PRINTW2$:ONGGOTO3200,3300,3400,9999
```

```
3049 REM ***EQUIPMENT PRINT-OUT SEQUENCE
***
3050 A$="EQUIPMENT NAME.            SER. N
O.  ROOM  PURCHASED            COST "
3055 A$=A$+"[SPC] VALUE TO-DAY"
3060 GOSUB516:GOSUB512:IFP4$="Y"THEN3080
3065 OPEN4,4,7:IFRR=1THEN3075
3070 GOSUB450:GOSUB457:RR=1
3075 GOSUB506
3080 I=0:GOSUB216:IFY$="Y"THEN3105
3085 IFP4$="N"THENCLOSE4
3090 RETURN
3095 IFI=>TTTHENGOSUB210:GOTO3125
3100 A$=A9$(Z):Z=Z+1:IFA$="****T"THENA$=
"[SPC]":GOSUB470:GOSUB470:GOTO3110
3105 GOSUB470:GOTO3095
3110 IFP4$="N"THENCLOSE4
3115 GOSUB210:GOSUB226:IFY$="Y"THENZ=1:G
OTO3060
3120 RETURN
3125 IF94$="N"THENCLOSE4
3130 GOTO3060
3199 REM ***SEARCH FOR EQUIPMENT BY NAME
***
3200 PRINTW2$:PRINT"[CD][CD]ENTER THE NA
ME OF THE EQUIPMENT TO BE"
3205 PRINT"IDENTIFIED.  TO ENTER THE FIR
ST LETTER"
3210 PRINT"ONLY, OR PART OF THE NAME, WI
LL"
3215 PRINT"SEARCH ALL ITEMS OF EQUIPMENT
"
3220 PRINT"BEGINNING WITH THE KEYBOARD E
NTRY.[CD][CD]":PRINTW1$
3225 L9=24:GOSUB150:L9=0:I=1:L%=LEN(D9$)
:Z=1
3230 IFA8$(I)="****T"THENA9$(Z)="****T":
GOSUB210:GOTO3245
3235 IFLEFT$(A8$(I),L%)=D9$THENA9$(Z)=A8
$(I):PRINTA9$(Z):Z=Z+1
3240 I=I+1:GOTO3230
3245 Z=1:GOSUB3050:I=1:Z=1:GOTO3020
3299 REM ***SEARCH FOR EQUIPMENT BY ROOM
 NUMBER***
```

```
3300 PRINTW2$:PRINT"[CD][CD]ENTER THE RO
OM NUMBER FOR ALL ITEMS"
3305 PRINT"OF EQUIPMENT CONTAINED THERE-
IN.  BY"
3310 PRINT"ENTERING THE FIRST NUMBER ONL
Y, A"
3315 PRINT"SEARCH FOR ALL ITEMS OF EQUIP
MENT"
3320 PRINT"RELATED TO THAT NUMBER OR FLO
OR"
3321 PRINT"WILL BE MADE.[CD][CD]":PRINTW
1$
3325 L9=4:GOSUB150:L9=0:I=1:L%=LEN(D9$):
Z=1
3330 IFA8$(I)="****T"THENA9$(Z)="****T":
GOSUB210:GOTO3345
3335 IFMID$(A8$(I),36,L%)=D9$THENA9$(Z)=
A8$(I):PRINTA9$(Z):Z=Z+1
3340 I=I+1:GOTO3330
3345 Z=1:GOSUB3050:I=1:Z=1:GOTO3020
3399 REM ***SEARCH FOR EQUIPMENT BY YEAR
 OF PURCHASE***
3400 PRINTW2$:PRINT"[CD][CD]ENTER THE LA
ST TWO NUMBERS OF THE"
3405 PRINT"YEAR TO INSTIGATE A SEARCH FO
R ALL"
3410 PRINT"ITEMS OF EQUIPMENT PURCHASED"
3415 PRINT"DURING THAT YEAR.[CD][CD]"
3420 PRINT"[CD]E.G.  83[CD][CD]":PRINTW1
$
3425 L9=2:GOSUB150:L9=0:I=1:L%=LEN(D9$):
Z=1
3430 IFA8$(I)="****T"THENA9$(Z)="****T":
GOSUB210:GOTO3445
3435 IFMID$(A8$(I),49,L%)=D9$THENA9$(Z)=
A8$(I):PRINTA9$(Z):Z=Z+1
3440 I=I+1:GOTO3430
3445 Z=1:GOSUB3050:I=1:Z=1:GOTO3020
```

Commentary

Lines 2999–3130: Deal with the recall of the equipment file records, the program menu and a printout routine, which includes a column heading in lines 3050–3055. The example file created by the previous program is com-

posed of the following records:

Equipment Item No. 1	12345	1.22	22/3/82	100.90	133.00
Equipment Item No. 2	6789	2.22	22/1/83	5000.76	5000.76
Equipment Item No. 3	13579	1.49	3/6/82	342.39	350.00
Equipment Item No. 4	98765	4.66	1/11/79	231.22	120.00

Lines 3199–3245: Will search the file for a stipulated piece of equipment and print out, either on screen or on paper, the result of the search. In the example below, equipment item no. 3 was chosen:

Equipment Name	Ser. No.	Room	Purchased	Cost	Value today
Equipment Item No. 3	13579	1.49	3/6/82	342.39	350.00

Lines 3299–3345: Will search for and display all equipment situated in a specified room. Equipment per floor can also be achieved by entering the first number only. In the example below, the operator stipulated all items on Floor 1.

Equipment Name	Ser. No.	Room	Purchased	Cost	Value today
Equipment Item No. 1	12345	1.22	22/3/82	100.90	133.00
Equipment Item No. 3	13579	1.49	3/6/82	342.39	350.00

Lines 3399–3445: Examine the file and identify all items purchased in a specified year. The example chosen was for the year 1982:

Equipment Name	Ser. No.	Room	Purchased	Cost	Value today
Equipment Item No. 1	12345	1.22	22/3/82	100.90	133.00
Equipment Item No. 3	13579	1.49	3/6/82	342.39	350.00

Objectives 4 and 5 are met by this routine, objective 6 has been dealt with in Chapter 3 and the placing of the records in alphabetic order can be accomplished by the following modifications to the sort routine, fully dealt within the third section of Chapter 6:

Line 3: Change 'address' to 'equipment'.

Lines 36, 37 and 38: Change the figure '20' to '25'.

CHAPTER 6
Society Mailing List

The aim of the programs contained in this chapter is to simplify the administrative paperwork involved in the maintenance of a society membership list or any other type of mailing list.

The objectives of this system are:

1) To create a file containing members'/clients'/customers' names, addresses, telephone numbers and pertinent interests.
2) To have the ability to update this file by removing records no longer relevant, by making record corrections and adding new records.
3) To be able to produce a printed list of all records, exactly as they are held on file, for checking.
4) To produce a list of the records, in alphabetical order, and formatted in such a way that the resulting list can act as a quick reference directory.
5) To produce a printed list of all members, indicating their particular interests, for circulation to members, annually or whenever necessary.
6) To produce printed membership lists appertaining to specified interests.
7) To have the facility to produce newsletters, individually addressed, with address labels, for all members or interest-specified members.

At first glance, this would appear to be a formidable list of objectives which would require considerable programming. However, it is not as bad as it looks. To produce newsletters, the word processing system would be required and, after this had been done, a method of modification and cannibalisation of this system can be instigated. The only other program to be entered would be the Alphabetic Sort routine, later in this chapter.

1. The Mailing List

Use the Address File program already detailed in the second section of Chapter 4, with the modifications listed below, to meet objectives 1 to 3.

Amended listing

```
1 REM ***SOCIETY OR BUSINESS MAILING LIS
T. COMMODORE 64***
5 W2$="[CLR][RVS]SOCIETY MEMBERSHIP OR B
USINESS MAIL LIST[HOME][CD]"
301 I=1:P4=160:REM ***MAXIMUM STRING LEN
GTH***
357 W5$="[RVS]NAME.                    TITLE.
                INITIAL.               "
358 W6$="[RVS]ADDRESS 1.               ADDRES
S 2.            ADDRESS 3.             "
359 W7$="[RVS]ADDRESS 4.               TELE.N
O.  INTERESTS."
360 W5$=W5$+W6$+W7$
367 PRINT"[HOME][CD][CD][CD][CD][CD][CD]
[CD]";W5$:PRINT
543 GOSUB766:IFA8$="****T"THEN545
544 A$=LEFT$(A8$,80):GOSUB470:A$=MID$(A8
$,81):GOSUB470:A$="[SPC]":GOSUB470:GOTO5
42
702 D9$="ADDRESS":RETURN
704 D9$="ADDRESS":RETURN
2226 IFD9$="#"THENB8$=LEFT$(BL$,10):K=K+
1:GOTO2228
2227 B8$=D9$:L%=LEN(B8$):K=K+1:IFL%<10TH
ENB8$=B8$+LEFT$(BL$,10-L%)
2228 PRINTW2$:PRINT"A = INTEREST 1":PRIN
T"B = INTEREST 2":PRINT"C = INTEREST 3"
2229 PRINT"D = INTEREST 4":PRINT"E = INT
EREST 5":PRINT"F = INTEREST 6"
2230 PRINT"G = INTEREST 7":PRINT"H = INT
EREST 8":PRINT"I = INTEREST 9"
2231 PRINT"J = INTEREST 10":PRINTW1$
2232 PRINT"TYPE IN MEMBERS INTERESTS  OR
"
2234 PRINT"#  SIGN FOR NO ENTRY REQUIRED
.":PRINTW1$:L9=10:GOSUB150
2236 IFD9$="#"THENB8$=B8$+LEFT$(BL$,10):
D9$="":GOTO2240
2238 L%=LEN(D9$):IFL%<10THEND9$=D9$+LEFT
$(BL$,10-L%)
2239 B8$=B8$+D9$:D9$=""
```

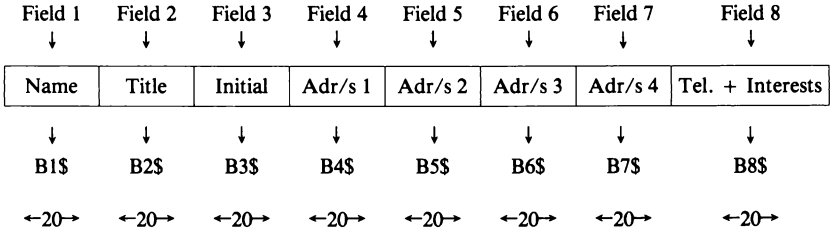
Commentary

As this is a dedicated program to produce a file of names and addresses plus interests, the file produced, for convenience, is named 'address' and therefore lines 702 and 704 should read as follows:

```
702 D9$ = "ADDRESS":RETURN
704 D9$ = "ADDRESS":RETURN
```

There is no reason why this name should be adhered to, but any alteration requires a corresponding one to line 3415 in the word processing system. Each record produced by this program is composed of eight fields, whose storage on tape/disc is shown in **Figure 6.1**.

Figure 6.1



An example file, containing three records, is printed out below. This is achieved by altering the standard printer routine, in lines 543 and 544, to break up the 160 character records into two parts to accommodate paper width.

```
543 GOSUB766:IFA8$ = "****T"THEN545
544 A$ = LEFT$(A8$,80):GOSUB470:A$ = MID$(A8$,81):GOSUB
    470:A$ = "[SPC]":GOSUB470:GOTO542
```

The alterations will print out the records in the format shown below:

```
Smith,              Mr.          G.          76, Brighton Road,
Rington,            Blinkshire, RI1 2LZ      01  1111111ABCDEFGHIJ

Granton & Co. Ltd.,                          Brickworks Street,
Industrial Estate,  Shington,    Kremshire, SH2 6ZL 01 2222222CJ

Black,              Mrs.         J.          'Dunroamin',
6, Provost Wynd,    Tremingly,   Tripshire, TR6 7SS 01 3333333BDGI
```

The records show that Mr. Smith has ten interests (codes A to J), Granton & Co. Ltd. have two (C and J) and Mrs. Black has four (B, D, G and I). These coded interests were tagged on to the telephone number input, B8$,

in lines 2228–2239: the programmer should include the allocation of these interests in lines 2228–2231.

2. Analysing the list

The following program listing is designed to cater for the analysis of the records and to fulfil objectives 4, 5 and 6. Placing the records into alphabetical order will be dealt with in the next section of this chapter.

The program should initially contain the following subroutines, combined together by the merge facility in Subroutine 1:

1) Subroutine 3
2) Subroutine 4
3) Subroutine 6
4) Subroutine 8t *or* 8d
5) Subroutine 9

Amended Listing

```
1 REM ***SOCIETY OR BUSINESS MAILING LIS
T. COMMODORE 64***
5 W2$="[CLR][RVS]SOCIETY MEMBERSHIP OR B
USINESS MAIL LIST[HOME][CD]"
10 DIMA8$(999),A9$(999)
702 D9$="ADDRESS":RETURN
704 D9$="ADDRESS":RETURN
979 REM ***RECALL ADDRESS FILE RECORDS A
ND PLACE A8$(I) SUBSCRIPT ARRAY***
980 GOSUB823:GOSUB703:F8$=D9$:GOSUB730:I
FER=1THENER=0:GOTO980
985 Z=1:PRINTW2$
990 GOSUB766:IFA8$="****T"THENGOSUB869:G
OSUB901:A8$(Z)="****T":Z=1:GOTO1000
995 A8$(Z)=A8$:Z=Z+1:A8$="":GOTO990
1000 PRINTW2$:PRINT"TYPE NUMBER FOR OPTI
ON."
1001 PRINT"[CD]1  QUICK REFERENCE DIRECT
ORY OF MEMBERS."
1002 PRINT"[CD]2  FULL MEMBERSHIP LIST P
RINT-OUT."
1003 PRINT"[CD]3  MEMBERSHIP LIST CORRES
PONDING TO A      SPECIFIED INTEREST."
1004 PRINT"[CD]4  FOR SCREEN DISPLAY OF
```

```
DATA FILE."
1005 PRINT"[CD]5  TO PRINT ADDRESS FILE.
"
1008 PRINT"[CD]6  RETURN TO MENU."
1009 PRINT"[CD]7  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>7THEN1010
1012 PRINTW2$:ONGGOTO4000,5000,6000,894,
533,221,9999
2995 REM ***PLACE RECORDS IN ACCEPTABLE
FORMAT***
3000 B1$=LEFT$(A8$(Z),20):B2$=MID$(A8$(Z
),21,20):B3$=MID$(A8$(Z),41,20)
3005 B4$=MID$(A8$(Z),61,20):B5$=MID$(A8$
(Z),81,20):B6$=MID$(A8$(Z),101,20)
3010 B7$=MID$(A8$(Z),121,20):B8$=MID$(A8
$(Z),141,10):B9$=RIGHT$(A8$(Z),10)
3015 IFLEFT$(B2$,1)=CHR$(160)THEN3040
3020 L%=LEN(B1$):IFRIGHT$(B1$,1)=CHR$(16
0)THENB1$=LEFT$(B1$,L%-1):GOTO3020
3025 L%=LEN(B2$):IFRIGHT$(B2$,1)=CHR$(16
0)THENB2$=LEFT$(B2$,L%-1):GOTO3025
3030 L%=LEN(B3$):IFRIGHT$(B3$,1)=CHR$(16
0)THENB3$=LEFT$(B3$,L%-1):GOTO3030
3035 B1$=B2$+"[SPC]"+B3$+"[SPC]"+B1$:RET
URN
3040 L%=LEN(B1$):IFRIGHT$(B1$,1)=CHR$(16
0)THENB1$=LEFT$(B1$,L%-1):GOTO3040
3045 RETURN
3499 REM ***RECORD PRINT-OUT SEQUENCE***
3500 GOSUB516:GOSUB512:IFP4$="Y"THEN3520
3505 OPEN4,4,7:IFRR=1THEN3515
3510 GOSUB450:GOSUB457:RR=1
3515 GOSUB506
3520 I=0:GOSUB216:IFY$="Y"THEN3535
3525 GOSUB869:GOSUB901:IFP4$="N"THENCLOS
E4
3530 RETURN
3535 IFI=>TTTHENGOSUB210:GOTO3565
3540 A$=A9$(X):IFA$="****T"THEN3550
3545 GOSUB470:X=X+1:GOTO3535
3550 IFP4$="N"THENCLOSE4
3555 GOSUB210:GOSUB226:IFY$="Y"THENX=1:G
```

```
OTO3500
3560 RETURN
3565 IFP4$="N"THENCLOSE4
3570 GOTO3500
3999 REM ***QUICK REFERENCE DIRECTORY OF
 MEMBERS
4000 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE NAME AND TELEPHONE NUMBER[CD]"
4005 PRINT"                  DIRECTORY ONLY?
[CD][CD]":PRINTW1$:GOSUB202:IFY$="N"THEN
4500
4010 Z=1:X=1
4015 IFA8$(Z)="****T"THENA9$(X)="****T":
GOTO4030
4020 GOSUB3000:L%=LEN(B1$):B1$=B1$+LEFT$
(BL$,40-L%):A9$(X)=B1$+B8$
4025 Z=Z+1:X=X+1:GOTO4015
4030 X=1:GOSUB3500:GOTO1000
4495 REM ***FULL ADDRESS DIRECTORY***
4500 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE NAME, ADDRESS AND[CD]"
4505 PRINT"TELEPHONE NUMBER DIRECTORY?[C
D][CD]":PRINTW1$:GOSUB202:IFY$="N"THEN48
00
4510 Z=1:X=1
4515 IFA8$(Z)="****T"THENA9$(X)="****T":
GOTO4580
4520 GOSUB3000:A9$(X)=B1$:X=X+1
4525 IFLEFT$(B4$,1)=CHR$(160)THEN4565
4530 A9$(X)=B4$:X=X+1
4535 IFLEFT$(B5$,1)=CHR$(160)THEN4565
4540 A9$(X)=B5$:X=X+1
4545 IFLEFT$(B6$,1)=CHR$(160)THEN4565
4550 A9$(X)=B6$:X=X+1
4555 IFLEFT$(B7$,1)=CHR$(160)THEN4565
4560 A9$(X)=B7$:X=X+1
4565 IFLEFT$(B8$,1)=CHR$(160)THEN4575
4570 A9$(X)=B8$:X=X+1
4575 A9$(X)="[SPC]":X=X+1:Z=Z+1:GOTO4515
4580 X=1:GOSUB3500:GOTO1000
4795 REM ***QUICK REFERENCE OF MEMBERS C
ONTAINED IN LIST BY SURNAME***
4800 PRINTW2$
4805 PRINT"[CD]THE FOLLOWING SEQUENCE AL
```

```
LOWS THE"
4810 PRINT"CHECKING OF THE MEMBERSHIP LI
ST FOR ALL"
4815 PRINT"MEMBERS WITH THE SAME SURNAME
.  BY"
4820 PRINT"ENTERING SMITH - ALL MEMBERS
WITH THE"
4825 PRINT"THE NAME SMITH WILL BE IDENTI
FIED"
4830 PRINT"AND PRINTED OUT."
4835 PRINT"BY ENTERING S ONLY, ALL MEMBE
RS"
4840 PRINT"WHOSE SURNAME BEGINS WITH  S"
4845 PRINT"WILL BE IDENTIFIED AND PRINTE
D-OUT.":PRINTW1$:GOSUB210
4850 PRINTW2$
4855 PRINT"[CD][CD]ENTER THE SURNAME OF
THE MEMBERS TO BE"
4860 PRINT"IDENTIFIED        OR[CD]"
4865 PRINT"THE FIRST LETTER, (WHICH MUST
 BE A"
4870 PRINT"CAPITAL LETTER), AND FURTHER
SEQUENTIAL"
4875 PRINT"LETTERS MAY BE INCLUDED TO EN
ABLE THE"
4880 PRINT"SEARCH TO BE MORE SPECIFIC.[C
D]":PRINTW1$
4885 L9=19:GOSUB150:GOSUB201:IFY$="N"THE
N4850
4890 X=1:Z=1:J%=LEN(D9$)
4895 IFA8$(Z)="****T"THENA9$(X)="****T":
GOTO4920
4900 IFLEFT$(A8$(Z),J%)=D9$THEN4910
4905 Z=Z+1:GOTO4895
4910 GOSUB3000:A9$(X)=B1$+"[SPC]"+B4$+B5
$+B6$ :X=X+1
4915 A9$(X)=B7$+B8$+"[SPC]"+"INTERESTS:-
[SPC]"+B9$:X=X+1:A9$(X)="[SPC]":X=X+1:Z=
Z+1:GOTO4895
4920 X=1:GOSUB3500:GOTO1000
4999 REM ***FULL MEMBERSHIP LIST WITH IN
TERESTS***
5000 DATA"A  =  INTEREST A.","B  =  INTE
REST B.","C  =  INTEREST C."
```

```
5005 DATA"D  =  INTEREST D.","E  =  INTE
REST E.","F  =  INTEREST F."
5010 DATA"G  =  INTEREST G.","H  =  INTE
REST H.","I  =  INTEREST I."
5015 DATA"J  =  INTEREST J.","****T":X=1
:Z=1
5016 A9$(X)="MEMBERSHIP LIST FOR THE DOB
ERMAN SOCIETY":X=X+1:A9$(X)="[SPC]":X=X+
1
5020 READC1$:IFC1$="****T"THENRESTORE:GO
TO5030
5025 A9$(X)=C1$:X=X+1:GOTO5020
5030 A9$(X)="[SPC]":X=X+1
5035 IFA8$(Z)="****T"THENA9$(X)="****T":
GOTO5050
5040 GOSUB3000:A9$(X)=B1$+"[SPC]"+B4$+B5
$+B6$ :X=X+1
5045 A9$(X)=B7$+B8$+"[SPC]"+"INTERESTS:-
[SPC]"+B9$:X=X+1:A9$(X)="[SPC]":X=X+1:Z=
Z+1:GOTO5035
5050 X=1:GOSUB3500:GOTO1000
5995 REM ***MEMBERSHIP LIST FOR A SPECIF
IED INTEREST***
6000 X=1:Z=1:PRINTW2$
6005 A9$(X)="MEMBERSHIP LIST FOR THE DOB
ERMAN SOCIETY":X=X+1:A9$(X)="[SPC]":X=X+
1
6010 READC1$:IFC1$="****T"THENRESTORE:GO
TO6020
6015 PRINTC1$:GOTO6010
6020 PRINTW1$
6025 PRINT"PLEASE SPECIFY WHICH INTEREST
 YOU WISH  THE LIST TO RELATE TO."
6030 PRINTW1$:L9=1:GOSUB150:GOSUB201:IFY
$="N"THEN6000
6035 IFD9$<"A"ORD9$>"J"THENGOSUB207:GOSU
B210:PRINTW2$:GOTO6000
6040 READC1$:IFD9$=LEFT$(C1$,1)THENRESTO
RE:GOTO6050
6045 GOTO6040
6050 A9$(X)="THIS LIST APPERTAINS TO MEM
BERS WHO HAVE AN INTEREST IN":X=X+1
6055 A9$(X)=MID$(C1$,7):X=X+1:A9$(X)="[S
PC]":X=X+1
```

```
6060 J%=151
6065 IFA8$(Z)="****T"THENA9$(X)="****T":
X=1:GOTO6095
6070 IFJ%=161ORMID$(A8$(Z),J%,1)=CHR$(16
0)THENJ%=151:Z=Z+1:GOTO6065
6075 IFMID$(A8$(Z),J%,1)=D9$THENJ%=151:G
OTO6085
6080 J%=J%+1:GOTO6070
6085 GOSUB3000:A9$(X)=B1$+"[SPC]"+B4$+B5
$+B6$ :X=X+1
6090 A9$(X)=B7$+B8$:X=X+1:A9$(X)="[SPC]"
:X=X+1:Z=Z+1:GOTO6065
6095 X=1:GOSUB3500:GOTO1000
```

Commentary

Lines 2995–3045: Deal with the re-arranging of record fields 1, 2 and 3 into an acceptable format, eg Smith Mr. G. to read Mr. G. Smith.

Lines 3499–3570: Accommodate the printing of the records after analysis.

Lines 3999–4030: Produce a quick reference directory printout composed of the name and telephone number only. (NB. The records have not as yet been placed in alphabetical order.) The results of this routine are shown below:

Mr. G. Smith,	01 1111111
Granton & Co. Ltd.,	01 2222222
Mrs. J. Black,	01 3333333

Lines 4495–4580: Produce a quick reference directory composed of the name, address and telephone number of each member as below:

Mr. G. Smith,
76, Brighton Road,
Rington,
Blinkshire, RI1 2LZ
01 1111111

Granton & Co. Ltd.,
Brickworks Street,
Industrial Estate,
Shington,
Kremshire, SH2 6ZL
01 2222222

Mrs. J. Black,
'Dunroamin',
6, Provost Wynd,
Tremingly,
Tripshire, TR6 7SS
01 3333333

Lines 4795–4920: Cater for a search to be made of the membership list file held on tape or disc for specified members, eg on entering Smith when requested, all members with that surname will be identified by the search and printed out. By entering S, all surnames beginning with S will be identified, on entering Smi all surnames beginning with Smi will be identified, and so on. The example below was isolated by Bl:

Mrs. J. Black, 'Dunroamin', 6, Provost Wynd, Tremingly, Tripshire, TR6 7SS 01 3333333 Interests: BDGI

Lines 4999–5050: Deal with a full membership list printout, eg:

Membership List for the Doberman Society

A = Interest A
B = Interest B
C = Interest C
D = Interest D
E = Interest E
F = Interest F
G = Interest G
H = Interest H
I = Interest I
J = Interest J

Mr. G. Smith, 76, Brighton Road, Rington, Blinkshire, RI1 2LZ
01 1111111 Interests: ABCDEFGHIJ

Granton & Co. Ltd., Brickworks Street, Industrial Estate, Shington,
Kremshire, SH2 6ZL 01 2222222 Interests: CJ

Mrs. J. Black, 'Dunroamin', 6, Provost Wynd, Tremingly,
Tripshire, TR6 7SS 01 3333333 Interests: BDGI

Lines 5995–6095: Will give a listing of members with a stipulated interest. The stipulated interest in the example was B:

Membership List for the Doberman Society
This list appertains to members who have an interest in Interest B.

Mr. G. Smith,	76, Brighton Road, Rington, Blinkshire, RI1 2LZ 01 1111111
Mrs. J. Black, Tripshire, TR6 7SS	'Dunroamin', 6, Provost Wynd, Tremingly, 01 3333333

Newsletter

To fulfil objective 7 the standard word processing system in Chapter 3 will suffice. However, meeting the requirements of producing individually-addressed letters to all members, with address labels, linked to a specific interest, requires some modifications to the printout sequence.

Change program line 3380 to 3384

```
3375 GOSUB200:Q1$=Y$:IFQ1$="N"THEN3384:R
EM ***MODS. TO CHAPTER 4  PROGRAM B***
3376 PRINTW2$:PRINT"DO YOU REQUIRE AN AL
L MEMBER CIRCULATION?":PRINTW1$
3377 GOSUB200:IFY$="Y"THENQ4$="0":GOTO33
84
3378 PRINTW2$:PRINT"A = INTEREST A.":PRI
NT"B = INTEREST B."
3379 PRINT"C = INTEREST C.":PRINT"D = IN
TEREST D.":PRINT"E = INTEREST E."
3380 PRINT"F = INTEREST F.":PRINT"G = IN
TEREST G.":PRINT"H = INTEREST H."
3381 PRINT"I = INTEREST I.":PRINT"J = IN
TEREST J.":PRINTW1$
3382 PRINT"INDICATE WHICH INTEREST?":PRI
NTW1$:L9=1:GOSUB150
3383 Q4$=D9$:D9$=""
3426 J%=151:IFQ4$="0"THEN3430
3427 IFJ%=161ORMID$(A9$(Z),J%,1)=CHR$(16
0)THENA9$(Z)="":GOTO3420
3428 IFMID$(A9$(Z),J%,1)=Q4$THEN3430
3429 J%=J%+1:GOTO3427
```

3. Alphabetic Sort

This final section is concerned with the placing of the records into alphabetical order. To obtain the maximum amount of memory space in the computer, the standard subroutines will not be included and a listing of the entire program is shown below.

Full listing

```
1 REM ***ALPHABETIC SORT OF ADDRESS FILE
***
2 POKE53280,11:POKE53281,12:PRINTCHR$(14
4):PRINTCHR$(14):POKE650,255
3 F8$="ADDRESS":UL$="P1"
4 W1$="[RVS]##############################
#############"
5 W2$="[CLR][RVS]ALPHABETIC SORT ROUTINE
 FOR ADDRESS FILE"
6 DIMA8$(999),SL(16),SR(16):GOTO18
7 PRINT"########################"
8 PRINT"# Y FOR YES  N FOR NO #"
9 PRINT"########################"
10 Y$="":GETY$:IFY$<>"Y"ANDY$<>"N"THEN10
11 PRINTW2$:RETURN
12 PRINT"##################################
######"
13 PRINT"# ****SPACE[SPC]BAR KEY TO CONT
INUE**** #"
14 PRINT"##################################
######"
15 G$="":GETG$:IFG$<>CHR$(32)ANDG$<>CHR$
(160)THEN15
16 PRINTW2$:RETURN
17 GOTO15
18 PRINTW2$:PRINT"[CD][CD]PLACE ADDRESS
TAPE IN RECORDER[CD][CD]":PRINTW1$:GOSUB
12
19 OPEN8,1,0,F8$:PRINTW2$:I=1
20 G$="":GET#8,G$:IFG$=""THEN20
21 IFG$=CHR$(13)THEN23
22 GG$=GG$+G$:GOTO20
23 A8$(I)=GG$:GG$="":PRINTA8$(I):IFA8$(I
)="****T"THENCLOSE8:I=1:GOTO26
24 I=I+1:GOTO20
25 REM ***TO CHANGE 1ST DATA CHARACTER T
O LOWER CASE IF NECESSARY***
26 GOSUB12:PRINT"[CD][CD]SWITCH OFF TAPE
 RECORDER[CD][CD]":PRINTW1$:GOSUB12
27 IFA8$(I)="****T"THEN31
28 G=ASC(LEFT$(A8$(I),1)):IFG<159THEN30
29 G=G-128:G$=CHR$(G):A8$(I)=G$+MID$(A8$
(I),2):UL$="P2"
```

```
30 I=I+1:GOTO27
31 A8$(I)="ZZZZZZZZZZZZZZZZZZZZ":N=I:I=1
32 PRINTW2$:PRINT"[CD][CD]DO YOU WISH TO
 PROCEED?[CD][CD]":PRINTW1$:GOSUB7
33 IFY$="N"THENEND
34 S=1:SL(1)=1:SR(1)=N
35 L=SL(S):R=SR(S):S=S-1
36 I=L:J=R:X$=LEFT$(A8$(INT((L+R)/2)),20
)
37 IFLEFT$(A8$(I),20)<X$THENI=I+1:GOTO37
38 IFX$<LEFT$(A8$(J),20)THENJ=J-1:GOTO38
39 IFI>JTHEN41
40 W$=A8$(I):A8$(I)=A8$(J):A8$(J)=W$:I=I
+1:J=J-1
41 IFI<JTHEN37
42 IFI>=RTHEN44
43 S=S+1:SL(S)=I:SR(S)=R
44 R=J:IFL<RTHEN36
45 IFS<>0THEN35
46 I=1
47 IFA8$(I)="ZZZZZZZZZZZZZZZZZZZZ"THENA8
$(I)="****T":I=1:GOTO49
48 I=I+1:GOTO47
49 IFUL$="P1"THEN55
50 REM ***TO CHANGE FIRST DATA CHARACTER
 BACK TO UPPER CASE***
51 IFA8$(I)="****T"THENGOSUB12:GOTO55
52 G=ASC(LEFT$(A8$(I),1)):G=G+128:G$=CHR
$(G)
53 A8$(I)=G$+MID$(A8$(I),2):PRINTA8$(I)
54 I=I+1:GOTO51
55 PRINTW2$:PRINT"PLACE A DATA TAPE IN T
HE RECORDER.":PRINTW1$:GOSUB12
56 OPEN8,1,1,F8$:I=1
57 PRINT#8,A8$(I):IFA8$(I)="****T"THENCL
OSE8:END
58 I=I+1:GOTO57
```

Commentary

After subjecting the 'address' file to this routine, the records are placed in alphabetical order as shown:

Black,	Mrs.	J.	'Dunroamin',
6, Provost Wynd,	Tremingly,	Tripshire, TR6 7SS	01 3333333BDGI

Granton & Co. Ltd.,			Brickworks Street
, Industrial Estate,	Shington,	Kremshire, SH2 6ZL	01 2222222CJ
Smith,	Mr.	G.	76, Brighton Road
, Rington,	Blinkshire, RI1 2LZ		01 1111111ABCDEFG

As the sort routine will not work if the leading character of a record is upper case, lines 25–30 convert any offending character to lower case and line 29 records that upper case characters do exist by altering the variable UL$ to "P2". This information is acted upon in lines 50–54, where the leading character is converted back.

Line 31: Alters the file terminator to ensure that it will always be the final record after the sort routine.

Lines 31–45: Deal with the placing of the records in alphabetic order.

Modifications for disc

To alter the routine for a disc system, the program lines below show the changes necessary:

```
18 PRINTW2$:PRINT"[CD][CD]PLACE THE ADDR
ESS DISC IN THE DRIVE[CD][CD]":PRINTW1$:
GOSUB12
19 OPEN8,8,8,F8$+",S,R":PRINTW2$:I=1
26 GOSUB12
55 PRINTW2$
56 OPEN8,8,8,"T1,S,W":I=1
57 PRINT#8,A8$(I):IFA8$(I)="****T"THENCL
OSE8:GOTO59
58 I=I+1:GOTO57
59 OPEN15,8,15:PRINT#8,"S:"+F8$:CLOSE15
60 OPEN15,8,15:PRINT#8,"R:"+F8$+"=T1":CL
OSE15:END
```

The alterations to facilitate a disc system are straightforward; the final part saves the records to file 'T1', removes or scratches the original file and re-names the 'T1' file to 'address' file. This will ensure that a file will be present on disc at all times and will safeguard against a power-failure. Another method would be to scratch the 'address' file, make a copy of the 'T1' file under the 'address' name, and finally scratch the 'T1' file. This is fine provided that the records being handled do not occupy more than half the available space on the disc.

CHAPTER 7
Program Lister and Ancillary Program

1. Program Lister

The program contained in this section is a simple routine to overcome the reproduction of dot matrix graphic symbols (which has led to problems both with publishers and readers). These symbols, when they form part of a program listing, can be illustrated much more clearly by letters, as used in this book, eg replacing the control graphic symbol for CURSOR DOWN by CD.

Daisy wheel printers cannot produce control characters and are therefore not of much value in program listing.

Subroutines to be included are:

1) Subroutine 1
2) Subroutine 3
3) Subroutine 4
4) Subroutine 8t *or* 8d
5) Subroutine 9

This routine requires the program to be converted to be in the form of a data file. Load the program to be converted into the computer, place a data tape into the recorder and enter:

```
OPEN1,1,1,"FILE NAME":CMD1:LIST
```

and press RETURN.

Press record and PLAY is displayed on the screen. After the recording is complete, enter:

```
PRINT#1,"****T":PRINT#1:CLOSE1
```

and press RETURN.

After the recorder has stopped, the tape may be rewound and the next program loaded into the computer. The final data file is saved on tape, under the original file name plus .1 (full stop plus 1). Any of the printout sequences will produce a satisfactory listing.

Amended listing

```
1 REM ***PROGRAM LISTER   COMMODORE 64.*
**
5 W2$="[CLR][RVS]PROGRAM LISTER.
                    [HOME][CD]"
10 DIMA8$(999)
1000 PRINTW2$:PRINT"[CD]TYPE NUMBER FOR
OPTION."
1001 PRINT"[CD]1  TO CREATE A DATA FILE
FOR EDITING."
1004 PRINT"[CD]2  FOR SCREEN DISPLAY OF
DATA FILE."
1008 PRINT"[CD]3  TO RETURN TO MENU PROG
RAM."
1009 PRINT"[CD]4  TO END PROGRAM.":PRINT
W1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>4THEN1010
1012 PRINTW2$:ONGGOTO2500,891,221,9999
1995 REM *** LISTING ROUTINE INCLUDING C
URSOR CONTROL CHARACTERS ***
2000 PRINTW2$
2005 PRINT"[CD][CD]HAVE YOU REMEMBERED T
O CONVERT YOUR     PROGRAM TO A DATA FIL
E.[CD]"
2010 PRINTW1$:GOSUB202:IFY$="Y"THENRETUR
N
2015 PRINTW2$:PRINT"[CD][CD]TO CONVERT A
 PROGRAM TO A DATA FILE[CD][CD]":PRINTW1
$
2020 PRINT"[CD]LOAD THE PROGRAM AND PLAC
E A DATA TAPE  IN THE RECORDER."
2025 PRINT"[CD]TYPE OPEN1,1,1:CMD1:LIST
AND WHEN THE   RECORDER HAS STOPPED"
2030 PRINT"[CD]TYPE PRINT#1,"+CHR$(34)+"
****T"+CHR$(34)+":PRINT#1:CLOSE1"
2035 GOSUB210:RUN3
2040 PRINTW2$:PRINT"[CD][CD][CD]NUMBER O
F CHARACTERS PER LINE?[CD][CD][CD]"
2045 PRINTW1$:L9=3:GOSUB150:XT=VAL(D9$):
D9$="":L9=0:RETURN
2050 PRINTW2$:PRINT"[CD]DO YOU REQUIRE T
HE CONTROL HEADINGS     TO BE INCLUDED?[
CD]"
```

```
2055 PRINTW1$:GOSUB202:IFY$="N"THEN2090
2060 A8$(I)="[CD] = CURSOR DOWN   [CU] =
 CURSOR UP":PRINTA8$(I):I=I+1
2065 A8$(I)="[CR] = CURSOR RIGHT  [CL] =
 CURSOR LEFT":PRINTA8$(I):I=I+1
2070 A8$(I)="[HOME] = CURSOR TO HOME POS
ITION    [CLR] = CLEAR THE SCREEN"
2075 PRINTA8$(I):I=I+1:A8$(I)="[SPC] = S
HIFTED SPACE.":PRINTA8$(I):I=I+1
2080 A8$(I)="[RVS] = REVERSE FIELD ON
[RVO] = REVERSE FIELD OFF":PRINTA8$(I)
2085 I=I+1:A8$(I)="[SPC]":I=I+1
2090 TX=1:TP=0
2095 G$="":GET#8,G$:IFG$=""THEN2095
2100 P=ASC(G$):IFP<13THENA8$(I)="****T":
PRINTA8$(I):RETURN
2105 IFP=13THEN2250
2110 IFP=17THENG$="[CD]":TX=TX+4:GOTO220
0
2115 IFP=18THENG$="[RVS]":TX=TX+7:GOTO22
00
2120 IFP=19THENG$="[HOME]":TX=TX+6:GOTO2
200
2125 IFP=29THENG$="[CR]":TX=TX+4:GOTO220
0
2130 IFP=145THENG$="[CU]":TX=TX+4:GOTO22
00
2135 IFP=146THENG$="[RVO]":TX=TX+8:GOTO2
200
2140 IFP=147THENG$="[CLR]":TX=TX+5:GOTO2
200
2145 IFP=157THENG$="[CL]":TX=TX+4:GOTO22
00
2150 IFP=160THENG$="[SPC]":TX=TX+5:GOTO2
200
2155 IFP=133THENG$="[FN1]":TX=TX+5:GOTO2
200
2160 IFP=134THENG$="[FN3]":TX=TX+5:GOTO2
200
2165 IFP=135THENG$="[FN5]":TX=TX+5:GOTO2
200
2170 IFP=136THENG$="[FN7]":TX=TX+5:GOTO2
200
2175 IFP=137THENG$="[FN2]":TX=TX+5:GOTO2
200
```

```
2180 IFP=138THENG$="[FN4]":TX=TX+5:GOTO2
200
2185 IFP=139THENG$="[FN6]":TX=TX+5:GOTO2
200
2190 IFP=140THENG$="[FN8]":TX=TX+5:GOTO2
200
2195 TX=TX+1
2200 IFTP=1THEN2225
2205 IFTP=2THEN2235
2210 IFTP=3THEN2245
2215 IFTX>XTTHENTP=1:L=LEN(G$):TX=L+6:GO
TO2225
2220 GG$=GG$+G$:GOTO2095
2225 IFTX>XTTHENTP=2:L=LEN(G$):TX=L+6:GO
TO2235
2230 G9$=G9$+G$:GOTO2095
2235 IFTX>XTTHENTP=3:L=LEN(G$):TX=L+6:GO
TO2245
2240 G8$=G8$+G$:GOTO2095
2245 G7$=G7$+G$:GOTO2095
2250 IFGG$="****T"THENA8$(I)="****T":GOS
UB210:RETURN
2255 A8$(I)=GG$:PRINTA8$(I):I=I+1:IFTP=0
THEN2275
2260 A8$(I)="[SPC][SPC][SPC][SPC][SPC][S
PC]"+G9$:PRINTA8$(I):I=I+1:IFTP=1THEN227
5
2265 A8$(I)="[SPC][SPC][SPC][SPC][SPC][S
PC]"+G8$:PRINTA8$(I):I=I+1:IFTP=2THEN227
5
2270 A8$(I)="[SPC][SPC][SPC][SPC][SPC][S
PC]"+G7$:PRINTA8$(I):I=I+1
2275 GG$="":G9$="":G8$="":G7$="":GOTO209
0
2495 REM ***CREATE A NEW FILE***
2500 GOSUB2000
2505 GOSUB823:GOSUB703:F8$=D9$:D9$="":GO
SUB730:IFER=1THENER=0:GOTO2505
2510 I=1:GOSUB2040:GOSUB2050:GOSUB869:GO
SUB901
2515 I=1:GOSUB823:F8$=F8$+"L.":GOSUB710:
IFER=1THENER=0:GOTO2515
2520 A8$=A8$(I):IFLEFT$(A8$,1)<CHR$(33)T
HENI=I+1:GOTO2520
```

```
2525 GOSUB792:A8$="":IFA8$(I)="****T"THE
N2535
2530 I=I+1:GOTO2520
2535 GOSUB869:GOSUB210:GOSUB901:GOSUB840
:RUN3
```

Commentary

Lines 2110–2190: Produce the designated control graphic codes.

Lines 2195–2275: Responsible for the formatting of the records.

2. Ancillary program

The following program listing is concerned with the manipulation of tape and disc data files. I have called it the Ancillary or Utility program.

The objectives of the program are:

1) Screen display of a tape or disc record file.
2) Printout of a tape or disc record file.
3) Copy record files on to tape or disc.
4) Combine two record files together.
5) Rename record files.
6) Remove a record file from disc.
7) Transfer files stored on tape to disc and vice versa.
8) Format a new disc under program control.

The required subroutines to formulate the program are:

1) Subroutine 1
2) Subroutine 2
3) Subroutine 3
4) Subroutine 4
5) Subroutine 6
6) Subroutine 8t *or* 8d
7) Subroutine 9
8) Subroutine 10

Amended listing

```
1 REM ***ANCILLARY/UTILITY PROGRAM  COMM
ODORE 64***
5 W2$="[CLR][RVS]ANCILLIARY/UTILITY[SPC]
PROGRAM[SPC][SPC][SPC][SPC][SPC][SPC][SP
```

```
C][SPC][SPC][SPC][SPC][SPC][SPC][SPC][SP
C][SPC][SPC][HOME][CD]"
10 DIMA8$(999),A9$(999)
1000 PRINTW2$:PRINT"TYPE[SPC]NUMBER[SPC]
FOR[SPC]OPTION."
1001 PRINT"[CD]1[SPC][SPC]FOR[SPC]SCREEN
[SPC]DISPLAY[SPC]OF[SPC]A[SPC]DATA[SPC]F
ILE."
1002 PRINT"[CD]2[SPC][SPC]FOR[SPC]PRINT-
OUT[SPC]OF[SPC]A[SPC]DATA[SPC]FILE."
1003 PRINT"[CD]3[SPC][SPC]TO[SPC]MAKE[SP
C]A[SPC]COPY[SPC]OF[SPC]A[SPC]DATA[SPC]F
ILE."
1004 PRINT"[CD]4[SPC][SPC]TO[SPC]COMBINE
[SPC]TWO[SPC]DATA[SPC]FILES."
1005 PRINT"[CD]5[SPC][SPC]TO[SPC]RENAME[
SPC]A[SPC]TAPE[SPC]DATA[SPC]FILE."
1006 PRINT"[CD]6[SPC][SPC]FOR[SPC]OPTION
S[SPC]RELATED[SPC]TO[SPC]DISC[SPC]DRIVES
."
1008 PRINT"[CD]7[SPC][SPC]RETURN[SPC]TO[
SPC]MENU[SPC]PROGRAM."
1009 PRINT"[CD]8[SPC][SPC]TO[SPC]END[SPC
]PROGRAM.":PRINTW1$
1010 G$="":GETG$:IFG$=""THEN1010
1011 G=VAL(G$):IFG<1ORG>8THEN1010
1012 PRINTW2$:ONGGOTO891,533,2000,2500,3
000,1100,221,9999
1100 PRINTW2$:PRINT"TYPE[SPC]NUMBER[SPC]
FOR[SPC]OPTION."
1105 PRINT"[CD]1[SPC][SPC]TO[SPC]FORMAT[
SPC]A[SPC]NEW[SPC]DISC."
1110 PRINT"[CD]2[SPC][SPC]TO[SPC]RENAME[
SPC]A[SPC]DISC[SPC]FILE."
1115 PRINT"[CD]3[SPC][SPC]TO[SPC]REMOVE[
SPC]OR[SPC]SCRATCH[SPC]A[SPC]DISC[SPC]FI
LE."
1120 PRINT"[CD]4[SPC][SPC]TO[SPC]COMBINE
[SPC]DISC[SPC]FILES."
1125 PRINT"[CD]5[SPC][SPC]FOR[SPC]SCREEN
[SPC]DISPLAY[SPC]OF[SPC]A[SPC]DISC[SPC]F
ILE."
1130 PRINT"[CD]6[SPC][SPC]TO[SPC]TRANSFE
R[SPC]TAPE[SPC]DATA[SPC]TO[SPC]A[SPC]DIS
```

```
C[SPC]FILE."
1135 PRINT"[CD]7[SPC][SPC]TO[SPC]TRANSFE
R[SPC]DISC[SPC]DATA[SPC]TO[SPC]A[SPC]TAP
E[SPC]FILE."
1140 PRINT"[CD]8[SPC][SPC]FOR[SPC]PREVIO
US[SPC]OPTIONS."
1145 G$="":GETG$:IFG$=""THEN1145
1150 G=VAL(G$):IFG<1ORG>8THEN1145
1155 PRINTW2$:ONGGOTO4000,4500,5000,5500
,891,6000,6500,1000
1999 REM ***TO MAKE A COPY OF A DATA FIL
E***
2000 I=1:GOSUB823:GOSUB703:F8$=D9$:GOSUB
730:IFER=1THENER=0:GOTO2000
2005 GOSUB766:IFA8$="****T"THENA8$(I)="*
***T":GOSUB869:GOSUB901:GOTO2015
2010 A8$(I)=A8$:I=I+1:GOTO2005
2015 PRINTW2$:PRINT"[CD][CD]DO[SPC]YOU[S
PC]REQUIRE[SPC]THE[SPC]COPY[SPC]FILE[CD]
"
2020 PRINT"TO[SPC]HAVE[SPC]THE[SPC]SAME[
SPC]FILE[SPC]NAME?[CD][CD]":PRINTW2$:GOS
UB202
2025 IFY$="Y"THEN2035
2030 GOSUB701:F8$=D9$
2035 GOSUB843:RUN3
2499 REM ***TO COMBINE TWO DATA FILES***
2500 I=1:GOSUB823:GOSUB703:F8$=D9$:GOSUB
730:IFER=1THENER=0:GOTO2500
2505 GOSUB766:IFA8$="****T"THENGOSUB869:
GOSUB901:F9$=F8$:GOTO2515
2510 A8$(I)=A8$:I=I+1:GOTO2505
2515 GOSUB823:PRINTW2$:PRINT"[CD][CD]SEC
OND[SPC]FILE[SPC]TO[SPC]BE[SPC]COMBINED[
SPC]TO[SPC]FIRST"
2520 GOSUB704:F8$=D9$:GOSUB730:IFER=1THE
NER=0:GOTO2515
2525 GOSUB766:IFA8$="****T"THENA8$(I)="*
***T":GOSUB869:GOSUB901:GOTO2535
2530 A8$(I)=A8$:I=I+1:GOTO2525
2535 PRINTW2$:PRINT"[CD][CD]DO YOU REQUI
RE THE NEW FILE[CD]"
2540 PRINT"TO[SPC]HAVE[SPC]THE[SPC]SAME[
SPC]NAME[SPC]AS[SPC]THE[SPC]FIRST?[CD][C
```

```
D]":PRINTW2$:GOSUB202
2545 IFY$="Y"THENF8$=F9$:GOTO2555
2550 GOSUB701:F8$=D9$
2555 GOSUB843:GOSUB840:RUN3
2999 REM ***RENAME A TAPE DATA FILE***
3000 I=1:GOSUB823:GOSUB703:F8$=D9$:GOSUB
730:IFER=1THENER=0:GOTO3000
3005 GOSUB766:IFA8$="****T"THENA8$(I)="*
***T":GOSUB869:GOSUB901:GOTO3015
3010 A8$(I)=A8$:I=I+1:GOTO3005
3015 I=1:GOSUB823:GOSUB701:F8$=D9$:GOSUB
710:IFER=1THENER=0:GOTO3015
3020 GOSUB843:GOSUB840:RUN3
3999 REM ***FORMAT A NEW DISC***
4000 PRINTW2$:PRINT"[CD][CD][SPC][SPC][S
PC][SPC]REMOVE[SPC]DISC[SPC]FROM[SPC]DIS
C[SPC]DRIVE.[CD][CD]":PRINTW1$:GOSUB210
4005 PRINTW2$:PRINT"[CD][CD]PLACE[SPC]NE
W[SPC]DISC[SPC]TO[SPC]BE[SPC]FORMATTED[S
PC]IN[SPC]DRIVE."
4006 PRINTW1$:GOSUB210
4010 PRINTW2$:PRINT"[CD][CD]ARE[SPC]YOU[
SPC]SURE[SPC]THAT[SPC]IT[SPC]IS[SPC]A[SP
C]NEW[SPC]DISC[CD]"
4015 PRINT"[CD][CD][SPC][SPC][SPC][SPC][
SPC]WHICH[SPC]IS[SPC]PRESENT[SPC]IN[SPC]
THE[SPC]DRIVE?[CD][CD]":PRINTW1$:GOSUB20
2
4020 IFY$="N"THEN4000
4025 PRINTW2$:PRINT"[CD][CD]ENTER[SPC]TH
E[SPC]DISC[SPC]TITLE[SPC]NAME.[CD][CD]":
PRINTW1$
4030 L9=16:DF=1:GOSUB150:F8$=D9$
4035 PRINTW2$:PRINT"[CD][CD]ENTER[SPC]TH
E[SPC]DISC[SPC]NUMBER.[CD][CD]":PRINTW1$
:L9=2:GOSUB100
4040 PRINTW2$:PRINT"[CD][CD]DISC[SPC]TIT
LE[SPC][SPC]:-[SPC]";F8$
4045 PRINT"[CD][CD]DISC[SPC]NUMBER[SPC]:
-[SPC]";D9$:PRINTW1$:GOSUB200
4050 IFY$="N"THEN4025
4055 PRINT"[CLR]PRINT#15,"+CHR$(34)+"N:"
+F8$+","+D9$+CHR$(34)
4060 POKE198,3:POKE631,19:POKE632,13:POK
```

```
E633,13:END
4499 REM ***RENAME A DISC FILE***
4500 GOSUB703:FL$=D9$:GOSUB701:FF$=D9$:G
OSUB872:RUN3
4999 REM ***REMOVE OR SCRATCH A FILE FRO
M DISC***
5000 GOSUB703:FF$=D9$:GOSUB879:RUN3
5499 REM ***COMBINE TWO DISC FILES***
5500 GOSUB882:RUN3
5999 REM***TRANSFER TAPE DATA TO DISC FI
LE***
6000 I=1:PRINTW2$:PRINT"[CD][CD]PLACE[SP
C]THE[SPC]DATA[SPC]TAPE[SPC]IN[SPC]THE[S
PC]RECORDER.[CD][CD]":PRINTW1$
6005 GOSUB210:GOSUB703:OPEN8,1,0
6010 GOSUB766:IFA8$="****T"THENA8$(I)="*
***T":CLOSE8:GOTO6020
6015 A8$(I)=A8$:I=I+1:GOTO6010
6020 PRINTW2$:PRINT"[CD][CD]DO[SPC]YOU[S
PC]REQUIRE[SPC]THE[SPC]COPY[SPC]FILE[CD]
"
6025 PRINT"TO[SPC]HAVE[SPC]THE[SPC]SAME[
SPC]FILE[SPC]NAME?[CD][CD]":PRINTW2$:GOS
UB202
6030 IFY$="Y"THEN6040
6035 GOSUB701:F8$=D9$
6040 GOSUB843:GOSUB840:RUN3
6499 REM ***COPY A DISC DATA FILE TO A T
APE DATA FILE***
6500 I=1:GOSUB823:GOSUB703:F8$=D9$:GOSUB
730:IFER=1THENER=0:GOTO6500
6505 GOSUB766:IFA8$="****T"THENA8$(I)="*
***T":GOSUB869:GOSUB901:GOTO6515
6510 A8$(I)=A8$:I=I+1:GOTO6505
6515 PRINTW2$:PRINT"[CD][CD]DO[SPC]YOU[S
PC]REQUIRE[SPC]THE[SPC]COPY[SPC]FILE[CD]
"
6520 PRINT"TO[SPC]HAVE[SPC]THE[SPC]SAME[
SPC]FILE[SPC]NAME?[CD][CD]":PRINTW2$:GOS
UB202
6525 IFY$="Y"THEN6535
6530 GOSUB701:F8$=D9$
6535 PRINTW2$:PRINT"[CD]PLACE[SPC]DATA[S
PC]TAPE[SPC]IN[SPC]RECORDER.[CD][CD]":PR
```

```
INTW1$:GOSUB210
6540 I=1:OPEN8,1,1,(F8$)
6545 A8$=A8$(I):GOSUB792:IFA8$(I)="****T
"THENCLOSE8:RUN3
6550 I=I+1:GOTO6545
```

APPENDIX A
Compilers and Programming Aids

BASIC compilers

Compilers are usually in the form of a series of programs which have the ability to convert other programs written in the BASIC language (source code) into machine code (object code). By compiling such a program, various advantages are gained — probably the most important one is the increase in the speed of a running program. Although BASIC is not exactly slow, a compiled version suddenly changes from a sluggish to a much faster and improved operator-related system.

Once an operator has run a system for some time, delays can become irksome, especially a screen display offering an option as to the next procedure, where the build-up of the display can be followed. A display flashed on to the screen, as if it were instantaneous, can be achieved by a compiled program. The handling of strings and data analysis can be greatly improved and speed increases of 10 per cent plus are claimed by the manufacturers.

My own experience with compilers is certainly favourable. I have run an administrative system for some time and converting the programs changed the system from being mediocre to what can best be described as a system which has suddenly become alive and even difficult to keep up with. A good example of speed increase is in the examination of a data string, character by character, where each character is printed out before proceeding to the next analysis. The printout from an uncompiled program is extremely slow, with definite delays between the printing of each character. The compiled version treated the printout as if the individual character examination sequence did not exist and there was hardly any difference from printing the data without analysis.

Other advantages associated with compilers are:

a) A reduction in program size resulting in more memory being available for data handling. This is achieved by rewriting a program in a more efficient form and ignoring REM statements, spaces, etc. The writing of programs is easier as it is no longer necessary to conserve every byte by crowding as much as possible on to each line.
b) Compiled programs cannot be listed or altered and therefore provide an added security safeguard.

c) Compatibility with BASIC/machine code combined programs.
d) Indication of syntax errors in the BASIC program as it is being compiled.

Programming aids

Programming aids, as the name suggests, are designed to help in the writing of programs and are becoming more and more sophisticated as manufacturers are continually making additions and improvements.

They are supplied either in the form of EPROMs, cartridges, or on disc /tape. EPROMs (Erasable Programmable Read Only Memory) are in the form of a chip which plugs directly into a spare socket within the computer. Cartridges are EPROMs enclosed in a capsule, and plug into the cartridge socket at the rear of the Commodore 64, behaving in the same way as ROM (Read Only Memory) chips. Disc and tape versions are a series of programs in machine code and are loaded into memory before the program to be edited (or whatever). Facilities normally offered by these aids are as follows:

a) **Append**–Will combine or add a program on tape or disc to a program currently in the computer. The lowest line number of the new program must have a greater value than the highest line number of the program residing in the computer. Append is not a merge (see below).

b) **Auto**–Provides an automatic, stipulated numbering system when writing programs. On pressing RETURN, having entered a program line, the next line number is placed on the screen ready for the next entry.

c) **Change**–Allows changing of text in program lines, eg if T + 10 required changing to T – 10 or 'Hello Sailor' to 'Hello Soldier'. Can be implemented in all program lines bearing the relevant text or a stipulated line range.

d) **Delete**–Removes stipulated lines from a program.

e) **Dump**–After running a program, dump will display all variables having a value, but not array (sub-script) variables, and allows the changing of their values or text.

f) **Find**–Will search for text, or text within quotation marks, and display on the screen the line number containing the text, plus the text itself, eg to examine a program for all lines that contain PRINT # 8 or "Hello Sailor".

g) **Help**–Assists in the location of a program error which has caused the program to stop running. By typing HELP, the line containing the syntax error is displayed on to the screen and the error pinpointed, either by an arrow or the flashing cursor.

h) **Merge**–The true combining of two programs. Will load a program from tape or disc and combine it with a program residing in the computer memory. Instead of tacking the second program on to the end of the first, as in append, a merge will interleave the new lines (provided they do not correspond with existing line numbers) with the old.

i) **Renumber**–Probably one of the most used facilities of all. It should be capable of changing the numbers of all lines from a stipulated starting point and with stipulated increments. It should also be capable of renumbering a section of a program. All references to line numbers, such as GOTO, GOSUB, etc., should also be updated to the new locations. Certain systems will produce an error number if a line is referred to by GOSUB, etc., if that line does not exist.

   Personally I prefer a system which leaves these reference numbers as they are. I usually develop a program without the standard subroutines, in the initial stages, to save time in loading, saving and verifying.

j) **Step** and **Trace**–Allows the program to proceed or run one line at a time, either with a pause or space bar to proceed. A very useful facility in debugging programs, especially if the fault is difficult to trace.

The above list should be considered as the minimum number and type of facilities offered in a programmer's aid. Today a more sophisticated package can be expected.

In assessing some of the programming aids available, I am indebted to Dataview Ltd. and to Supersoft.

Comments on compilers were partly based on the DTL-BASIC 64 compiler, supplied by Dataview, which included all the facilities mentioned in this chapter, and more, and performed in all respects according to its specification. Dataview can be contacted in the UK at Portreeves House, East Bay, Colchester, CO1 2XB, and in the USA at Cimarron, 666 Baker Street, Suite 319, Costa Mesa, California 92626. All other interested countries should contact the UK address.

The section on programming aids is partly based on VIC-TREE and Arrow Tape Functions, both available from Supersoft. The VIC-TREE cartridge provides over 40 extra commands, while the Arrow Tape Func-

tions cartridge is capable of speeding up program (though not data file) loading and saving by tape to near-disc speeds, and includes in addition a hexadecimal calculator and machine code monitor. Supersoft's address for UK and Europe is Dept. C64, Winchester House, Canning Road, Wealdstone, Harrow, Middlesex, HA3 7SJ, and their address for the USA and all other countries is Skyles Electric Works, 231E Sth Whisman Road, Mountain View, CA 94041.

APPENDIX B
Subroutines Crib Sheet

Line No.	Description
Subroutine 2	
100	Numeric Input (input returned and held by D9$).
Subroutine 3	
150	Alpha and Numeric Input (input returned and held by D9$).
Subroutine 4	
200	Are you Sure question.
202	Yes or No question.
207	Error message (incorrect input).
210	Space Bar to Continue.
216	Do you Wish to Proceed question.
218	Finished Adding Data question.
221	Return to Menu question.
226	Do you Require Another Copy question.
Subroutine 5	
300	Edit Record Data routine.
Subroutine 6	
450	Upper and Lower Case question (for Printer or Sort routines).
457	Printer Option question.
470	Printer Printout sequence (enter with data record allocated to A$).
492	Printer Underline routine.
502	Place Paper in Printer prompt.
504	New Sheet of Paper in Printer prompt.
506	Number of Spaces between Lines question.
512	Number of Lines per Page question.
516	Screen or Printer Display of Data Records question.
533	Printout of Data Records routine (independent program).
Subroutine 7	
600	Convert Pounds to Pence (R holds amount and converted R is returned).

607 Round off Numbers after the Decimal Point (N1$ to enter and N1$ returned).
619 Convert Credit to Pounds and Pence (credit entered by C1 and C3$ returned).
621 Convert Debit to Pounds and Pence (debit entered by D1 and D3$ returned).

Subroutine 8

701 New File Name question (tape and disc).
703 Name of File to be Accessed question (tape and disc).
710 Open New File for Data Storage — F8$ (tape and disc).
720 Open File for Data Append TEST routine — F8$ (disc only).
730 Open File F8$ for Data Recall (tape and disc).
733 Open File F9$ for Data Recall (tape and disc).
737 Open T7 File (F7$) for Data Storage (disc only).
740 Open T6 File (F6$) for Data Storage (disc only).
743 Open T5 File (F5$) for Data Storage (disc only).
747 Open T7 File (F7$) for Data Recall (disc only).
749 Open T6 File (F6$) for Data Recall (disc only).
751 Open T5 File (F5$) for Data Recall (disc only).
754 Data Recall using GET # A5$ (disc only).
758 Data Recall using GET # A6$ (disc only).
762 Data Recall using GET # A7$ (disc only).
766 Data Recall using GET # A8$ (tape and disc).
770 Data Recall using GET # A9$ (tape and disc).
775 Data Recall using INPUT # A5$ (disc only)
777 Data Recall using INPUT # A6$ (disc only).
779 Data Recall using INPUT # A7$ (disc only).
781 Data Recall using INPUT # A8$ (tape and disc).
783 Data Recall using INPUT # A9$ (tape and disc).
786 Store Data using PRINT # A5$ (disc only).
788 Store Data using PRINT # A6$ (disc only).
790 Store Data using PRINT # A7$ (disc only).
792 Store Data using PRINT # A8$ (tape and disc).
795 Store Data using PRINT # A9$ (tape and disc).
823 Place Data Tape/Disc in Recorder/Drive.
831 Place your Working Tape/Disc in Recorder/Drive.
840 To make a Copy of a Data File question.
850 Remove Copy Data Disc and Replace with Program Disc.
861 To Test whether File F8$ is present on Disc or not.
865 Close all Tape/Disc Files.
866 Close Disc File 5.
867 Close Disc File 6.
868 Close Disc File 7.

869 Close Tape/Disc File 8.
870 Close Tape/Disc File 9.
872 Rename a Disc File (Newname(FF$) = Oldname(F1$)).
875 Copy Disc Data File (Copyfilename (F1$) = Originalfile (FF$)).
879 Remove or Scratch a File from Disc (FF$).
882 Combine Two Disc Files.
891 Screen Display of the Contents of a Data File (tape and disc).
901 Rewind Tape (tape only)
910 Access Disc Error-checking routine.
930 Create a New File Name using the Month and Year.

Subroutine 9
1000 Program Menu Routine.

Subroutine 10
2500 Create a New Data Record File (tape and disc).
2600 Add or Append New Data Records to an Existing File (tape and disc).

Subroutine 1
60000 Merge Two Programs together.
61000 Delete Specified Program Lines.
62000 Renumber Specified Program Lines.

[CD]	=	Cursor Down	[CU]	=	Cursor Up
[CR]	=	Cursor Right	[CL]	=	Cursor Left
[HOME]	=	Cursor to Home Position	[CLR]	=	Clear the Screen
[SPC]	=	Shifted Space			
[RVS]	=	Reverse Field On	[RVO]	=	Reverse Field Off

APPENDIX C
ASCII Character Set

Code	ASCII	Peek/ Poke	Code	ASCII	Peek/ Poke	Code	ASCII	Peek/ Poke	Code	ASCII	Peek/ Poke
0		@	31	←		62	>	>	93	]	
1		a	32	Space	Space	63	?	?	94	↑	
2		b	33	!	!	64	@		95	←	
3		c	34	"	"	65	a	A	96		
4		d	35	#	#	66	b	B	97	A	Space
5		e	36	$	$	67	c	C	98	B	
6		f	37	%	%	68	d	D	99	C	
7		g	38	&	&	69	e	E	100	D	
8		h	39	'	'	70	f	F	101	E	
9		i	40	(	(	71	g	G	102	F	
10		j	41	)	)	72	h	H	103	G	
11		k	42	*	*	73	i	I	104	H	
12		l	43	+	+	74	j	J	105	I	
13	return	m	44	,	,	75	k	K	106	J	
14	L.C.	n	45	-	-	76	l	L	107	K	
15		o	46	.	.	77	m	M	108	L	
16		p	47	/	/	78	n	N	109	M	
17	CD	q	48	0	0	79	o	O	110	N	
18	RVS	r	49	1	1	80	p	P	111	O	
19	HOME	s	50	2	2	81	q	Q	112	P	
20	DEL	t	51	3	3	82	r	R	113	Q	
21		u	52	4	4	83	s	S	114	R	
22		v	53	5	5	84	t	T	115	S	
23		w	54	6	6	85	u	U	116	T	
24		x	55	7	7	86	v	V	117	U	
25		y	56	8	8	87	w	W	118	V	
26		z	57	9	9	88	x	X	119	W	
27		[	58	:	:	89	y	Y	120	X	
28		£	59	;	;	90	z	Z	121	Y	
29	CR	]	60	<	<	91	[		122	Z	
30		↑	61	=	=	92	£		123		

Code	ASCII	Peek/ Poke	Code	ASCII	Peek/ Poke	Code	ASCII	Peek/ Poke	Code	ASCII	Peek/ Poke
124			157	CL	]	190		>	223		
125			158		↑	191		?	224	Space	Space
126			159		←	192			225		
127			160	Space	Space	193	A	A	226		
128		@	161		!	194	B	B	227		
129		a	162		"	195	C	C	228		
130		b	163		#	196	D	D	229		
131		c	164		$	197	E	E	230		
132		d	165		%	198	F	F	231		
133	f1	e	166		&	199	G	G	232		
134	f3	f	167		'	200	H	H	233		
135	f5	g	168		(	201	I	I	234		
136	f7	h	169		)	202	J	J	235		
137	f2	i	170		*	203	K	K	236		
138	f4	j	171		+	204	L	L	237		
139	f6	k	172		,	205	M	M	238		
140	f8	l	173		-	206	N	N	239		
141	Return	m	174		.	207	O	O	240		
142	V.C.	n	175		/	208	P	P	241		
143		o	176		0	209	Q	Q	242		
144		p	177		1	210	R	R	243		
145	CU	q	178		2	211	S	S	244		
146	RVO	r	179		3	212	T	T	245		
147	CLS	s	180		4	213	U	U	246		
148	Insert	t	181		5	214	V	V	247		
149		u	182		6	215	W	W	248		
150		v	183		7	216	X	X	249		
151		w	184		8	217	Y	Y	250		
152		x	185		9	218	Z	Z	251		
153		y	186		:	219			252		
154		z	187		;	220			253		
155		[	188		<	221			254		
156		£	189		=	222			255		

Other titles from Sunshine

COMMODORE 64 BOOKS

Graphic Art for the Commodore 64
Boris Allan £5.95
ISBN 0 946408 15 7

DIY Robotics and Sensors on the Commodore Computer
John Billingsley £6.95
ISBN 0 946408 30 0

Artificial Intelligence on the Commodore 64
Keith & Stephen Brain £6.95
ISBN 0 946408 29 7

Commodore 64 Adventures
Mike Grace £5.95
ISBN 0 946408 11 4

Mathematics on the Commodore 64
Czes Kosniowski £5.95
ISBN 0 946408 14 9

Advanced Programming Techniques on the Commodore 64
David Lawrence £5.95
ISBN 0 946408 23 8

The Working Commodore 64
David Lawrence £5.95
ISBN 0 946408 02 5

Commodore 64 Machine Code Master
David Lawrence & Mark England £6.95
ISBN 0 946408 05 X

DRAGON BOOKS

Advanced Sound & Graphics for the Dragon
Keith & Steven Brain £5.95
ISBN 0 946408 06 8

Dragon 32 Games Master
Keith & Steven Brain £5.95
ISBN 0 946408 03 3

The Working Dragon
David Lawrence £5.95
ISBN 0 946408 01 7

The Dragon Trainer
A handbook for beginners
Brian Lloyd £5.95
ISBN 0 946408 09 2

Sunshine also publishes

POPULAR COMPUTING WEEKLY

The first weekly magazine for home computer users. Each copy contains Top 10 charts of the best-selling software and books and up-to-the-minute details of the latest games. Other features in the magazine include regular hardware and software reviews, programming hints, computer swap, adventure corner and pages of listings for the Spectrum, Dragon, BBC, VIC 20 and 64, ZX 81 and other popular micros. Only 35p a week, a year's subscription costs £19.95 (£9.98 for six months) in the UK and £37.40 (£18.70 for six months) overseas.

DRAGON USER

The monthly magazine for all users of Dragon microcomputers. Each issue contains reviews of software and peripherals, programming advice for beginners and advanced users, program listings, a technical advisory service and all the latest news related to the Dragon. A year's subscription (12 issues) costs £10 in the UK and £16 overseas.

MICRO ADVENTURER

The monthly magazine for everyone interested in Adventure games, war gaming and simulation/role-playing games. Includes reviews of all the latest software, lists of all the software available and programming advice. A year's subscription (12 issues) costs £10 in the UK and £16 overseas.

COMMODORE HORIZONS

The monthly magazine for all users of Commodore computers. Each issue contains reviews of software and peripherals, programming advice for beginners and advanced users, program listings, a technical advisory service and all the latest news. A year's subscription costs £10 in the UK and £16 overseas.

For further information contact:
Sunshine
12–13 Little Newport Street
London WC2R 3LD
01-437 4343

Printed in England by Commercial Colour Press, London E7.

Business Applications for the Commodore 64 cuts through the software jungle. It provides a library of solid routines that can be combined to build up programs for accounts, VAT returns, stock control, automatic ordering, word processing, inventories, invoices, mailings and database applications.

All the programs are designed specifically for the Commodore 64, making full use of its facilities and the peripherals available. If you are considering applying the power of the 64 to your business, this book is the most cost-effective start you can make.

James Hall cut his teeth on the computerisation of a major university department in Scotland, developing many of the techniques presented in this book. He is now Chief Technician in the Department of Anatomy at a university in the Middle East.



ISBN 0 946408 12 2

SKOVLUNDE BOGCENTER
199.00

£5.95 net



This was brought to you

from the archives of

http://retro-commodore.eu